

## Towards hybrid molecular simulations

***Citation for published version (APA):***

Markvoort, A. J. (2006). *Towards hybrid molecular simulations*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Biomedical Engineering, Eindhoven University of Technology, Mathematics and Computer Science]. Technische Universiteit Eindhoven. <https://doi.org/10.6100/IR601609>

***DOI:***

[10.6100/IR601609](https://doi.org/10.6100/IR601609)

***Document status and date:***

Published: 01/01/2006

***Document Version:***

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

***Please check the document version of this publication:***

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

***General rights***

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

***Take down policy***

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# Towards Hybrid Molecular Simulations

ISBN-10: 90-386-3037-9  
ISBN-13: 978-90-386-3037-3

The work in this thesis has been carried out under the auspices of  
the research school IPA (Institute for Programming research and  
Algorithmics)  
IPA dissertation series 2006-09



Cover design: JWL Producties / A.J. Markvoort

Printed by Printservice Technische Universiteit Eindhoven, Eindhoven, The Netherlands

# Towards Hybrid Molecular Simulations

## PROEFSCHRIFT

ter verkrijging van de graad van doctor  
aan de Technische Universiteit Eindhoven,  
op gezag van de Rector Magnificus,  
prof.dr.ir. C.J. van Duijn,  
voor een commissie aangewezen  
door het College voor Promoties  
in het openbaar te verdedigen  
op dinsdag 14 maart 2006 om 16.00 uur

door

**Albert Jan Markvoort**

geboren te Utrecht



Dit proefschrift is goedgekeurd door de promotoren:

prof.dr. P.A.J. Hilbers

en

prof.dr. M.A.J. Michels

Copromotor:

dr. R. Pino Plasencia

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Computer simulations . . . . .	2
1.2	Molecular simulations . . . . .	3
1.3	Thesis outline . . . . .	6
<b>2</b>	<b>Wavelets in electronic structure calculations</b>	<b>9</b>
2.1	Introduction . . . . .	10
2.2	Wavelets . . . . .	18
2.3	Wavelet algorithms . . . . .	30
2.4	Example calculations . . . . .	45
2.5	Application to 3D molecular systems . . . . .	57
2.6	Application to 2D quantum dots . . . . .	65
2.7	Conclusion . . . . .	72
<b>3</b>	<b>Coarse grained molecular dynamics</b>	<b>75</b>
3.1	Introduction . . . . .	76
3.2	Coarse grained lipid model . . . . .	77
3.3	Simulations of bilayer and vesicle formation . . . . .	82
3.4	Discussion . . . . .	91
3.5	Vesicle fusion . . . . .	99
3.6	Conclusion . . . . .	102
<b>4</b>	<b>Artificial Chemistry Molecular Dynamics</b>	<b>103</b>
4.1	Introduction . . . . .	104
4.2	Combining Molecular Dynamics and Artificial Chemistry . . . . .	104
4.3	Biologically inspired case studies . . . . .	107
4.4	Discussion . . . . .	113
4.5	Conclusion . . . . .	115
<b>5</b>	<b>Hybrid Molecular Dynamics - Direct Simulation Monte Carlo</b>	<b>117</b>
5.1	Introduction . . . . .	118
5.2	Heat and particle flow in nanochannels . . . . .	119
5.3	Direct simulation Monte Carlo method . . . . .	132
5.4	Hybrid molecular dynamics-Monte Carlo . . . . .	136
5.5	Conclusion . . . . .	142

## **Contents**

---

<b>6 Concluding remarks</b>	<b>143</b>
<b>Bibliography</b>	<b>148</b>
<b>Summary</b>	<b>159</b>
<b>Samenvatting</b>	<b>161</b>
<b>Dankwoord</b>	<b>163</b>
<b>Curriculum Vitae</b>	<b>165</b>
<b>Publications</b>	<b>167</b>

# 1

## Introduction

---

**C**omputer simulations are nowadays regarded as mutual, vital partners with theory and experiments in the advance of scientific knowledge and engineering practice. Such simulations enable the study of many complex systems and phenomena that would be too expensive or even impossible to study by experiment. An important class of computer simulations is formed by molecular simulations. Molecular systems are ideal subjects for simulations since most processes on a molecular level occur on such a short time as well as length scale that they are only tractable by computer simulations, whereas these microscopic processes ultimately determine all macroscopic properties of the system. The demand for larger, more detailed, and more realistic simulations requires tremendous computational power and increasingly advanced computer algorithms. This thesis is devoted both to the development and the application of several molecular simulation methods. The topics include the development of an electronic structure calculations code employing wavelets and its application to quantum dots and molecular systems, the development of a coarse grained lipid model which is applied to study the dynamics of membranes and vesicles using coarse grained molecular dynamics, the extension of molecular dynamics with artificial chemistry, the use of molecular dynamics to study the influence of surface wettability on heat and particle flow in nanochannels, and the development of a hybrid molecular dynamics - Monte Carlo method to enable the study of microchannels.

---

### 1.1 Computer simulations

**Introduction** Traditionally, research has been divided into two categories: experiment and theory. This changed with the advent of the computer. Ever since the first computers were built in the early 1940's, these computers have been used for performing simulations. Their first large scale deployment was during World War II to model the process of nuclear detonation in the Manhattan Project and to calculate trajectories of ballistic missiles. From that point on increasingly complex computer simulation techniques were developed hand-in-hand with the rapid growth of computational power. The combination of the drastic increase in computer power and the development of more and more refined simulation methods made that computer simulations became a third category of research. As such, computer simulations are not a replacement for conventional experiments or theory. Instead, the approaches may be used to complement one another in a triad of experiment, theory, and computer simulation. Experiments yield measurements of properties and behavior of real systems. Theory invokes models depending on the scale –microscopic, mesoscopic or macroscopic– at which the system should be investigated. Traditionally, the outcome of these models, when mathematically tractable, could only be validated with the outcome of experiments on the real system. Computer simulations invoke a model and then aim to follow the pseudo-real behavior of that model. In such a way, computer simulations can be used in two ways. In the first place, when models cannot be solved analytically, simulations based on the model can be used to test the correctness of the model as representation of real systems by comparing the simulated behavior with the real observed behavior from experiments. In this way, simulations can also be used to assist in the interpretation of experimental results. In the second place, closing the triad, computer simulations can, when their correctness has been established, be used to develop and validate new models in the same way as by conventional experiments. Computer experiments can be performed in cases that real experiments are too expensive, intractable or even impossible and can thus be used to provide supplementary information. Furthermore, computer simulations can be used as a predictive tool and computer experiments can have advantages over conventional experiments as they offer a very high degree of control over experimental conditions.

**Types of computer simulation** Computer simulations come in different types. The most important distinction is between discrete and stochastic simulations on one side and continuous simulations on the other. In case of discrete or stochastic simulations a queue of events is maintained sorted by the time they should occur. A discrete simulator repeatedly reads the first event from the queue and processes this event by which new events are triggered that are merged into the queue. It is also possible that not all events are processed, but that events are processed with a certain chance. Such simulations, where the events occur probabilistically, are called stochastic simulations. Continuous (analog) simulations on the other hand are based on a numerical implementation of a set of either ordinary or partial differential equations. They enable the study of differential

equations which cannot be solved analytically. Representing time by time steps, the simulator recalculates all the equations periodically and uses the results to update the state. Both of these types of computer simulations are used for molecular simulations.

## 1.2 Molecular simulations

**Introduction** The class of computer simulations involved in molecular systems is called molecular simulation. In such simulations the collective behavior of systems of atoms as a result of the interactions between the individual particles is calculated. Such simulations are very interesting because to understand the macroscopic properties of materials, it is necessary to have a firm comprehension of the microscopic interactions between the particles the material is built of, i.e., between its constituent atoms. Moreover, most processes on the atomic level occur on such a short time as well as length scale that they are intractable by conventional experiments. Contrary to conventional experiments, computer simulations can give an invaluable insight in these processes. Molecular simulations can be divided into several categories. A first division is between continuous and discrete/stochastic methods, with molecular dynamics and Monte Carlo methods respectively as important members [Fre02, All87]. In molecular dynamics simulations deterministic particle trajectories are followed by integrating equations of motion, whereas in Monte Carlo methods new configurations are generated based on probabilities. A second division is by the level of detail incorporated by the model. To obtain the highest accuracy, the interactions between the atoms should be calculated ab-initio, implying quantum mechanical calculations of the electronic structure. Such ab-initio calculations are computationally very expensive, making them unfeasible for anything but very small systems. Larger systems can be dealt with using effective potentials to describe the atomic interactions. Even larger systems can be modeled with coarse grained models. Below, these various molecular simulation methods will be discussed in more detail.

**Molecular dynamics** Molecular dynamics is a computer simulation technique where the time evolution of a set of interacting atoms is followed by integrating their equations of motion. Statistical mechanics provides the theoretical basis for extracting properties from such molecular dynamics simulations. Static properties such as structure, energy, and pressure can be obtained from pair (radial) distribution functions, whereas dynamic and transport properties, such as diffusion and heat conductivity, can be obtained from time correlation functions or from so called non-equilibrium molecular dynamics simulations. Molecular dynamics thus provides a picture of the structure and dynamics at a molecular level, which often cannot be provided by experiments.

**History** The history of molecular dynamics started in the late 1950's in the field of theoretical physics when Alder [Ald57, Ald59] studied the interactions of hard spheres.

## Introduction

---

In this case, particles move at constant velocity between perfectly elastic collisions. This allows for an exact solution of the dynamics. However, atoms are not hard spheres, but they interact with each other. In 1960 Gibson et al. [Gib60] probably performed the first molecular dynamics calculation with a continuous potential. A few years later, the first simulations using a realistic potential for liquid argon by Rahman in 1964 [Rah64] and the simulation of liquid water in 1971 by Rahman and Stillinger [Rah71] were major advances. In the mean time Verlet [Ver67, Ver68] thoroughly investigated the properties of Lennard-Jones systems, and with these successes, the method of molecular dynamics gained popularity in material science. After simulations of diatomic molecules [Har68] followed simulations of small rigid molecules [Bar73], and flexible hydrocarbons [Ryc75]. With the first simulations of proteins in 1977, on the bovine pancreatic trypsin inhibitor (BPTI) [McC77], the method also gained popularity in biochemistry and biophysics. These type of simulations were further stimulated in the 1980's when a number of general purpose force fields for water, lipids, proteins and DNA became available as well as some general purpose molecular dynamics computer programs [Bro83, Wei84, Her84, Kal99] capable of running on large parallel computers. This also enabled the simulation of the aggregation of surfactants [Smi90] and lipid membranes, first with simplified models and without water [Kox80, vdP83] and later in atomistic detail including water [Egb88]. The vast increase in computational power in recent years further enabled molecular dynamics simulations of solvated proteins [Bro88, Kar90, Dua98], lipid systems with or without proteins [Tie97, Rou04] and protein-DNA complexes [Nil98].

**Effective potentials** The molecular dynamics simulations mentioned so far are all based on effective potentials describing the interaction between the atoms. These potentials require a large number of parameters, such as parameters for non-bonded interactions, force constants, equilibrium bond lengths, angles and dihedrals, and (partial) charges. There are three ways to deal with these parameters.

The first way is to use relatively simple model potentials and parameters. As discussed above, this was the approach used in the beginning. However, such relatively simple model potentials remain interesting because the level of detail of the potentials directly influences the computational cost per particle and thus the system sizes and time scales reachable in a simulation. Such model potentials are used in coarse grained models, where a number of atoms are represented by one single particle and that thus can be used to simulate phenomena that are still out of reach of more detailed methods.

The second way is to use as accurate potentials and parameters as possible. Whereas in the beginning only model potentials were used, the focus shifted to such more and more detailed potentials with the increase of computer power available. All the parameters necessary for such detailed potentials can be estimated from either experiments or quantum mechanical calculations.

**Ab-initio molecular dynamics** The third way is to abandon effective potentials completely. Namely, notwithstanding the statistical mechanical basis of molecular dynamics, the simplified form of the potentials with estimated parameters, limits the accuracy of the method. Moreover, the classical treatment of the system also makes it impossible to deal with chemical reactions. Both limitations can be overcome by explicitly calculating the electronic structure of the atoms in the system. Namely, the interactions between atoms depend critically on the behavior of their electrons. However, electrons are no classical particles obeying Newton's laws of motion. Instead, electrons are quantum particles, obeying the laws of quantum mechanics. Accurate calculations of electronic structure are thus based on first principles (ab-initio) quantum mechanical treatment of the electrons. Resulting forces on the atoms can be derived from this electronic structure. These forces can be used, replacing the effective potentials, in the molecular dynamics scheme as pioneered by Car and Parrinello [Car87]. However, because electrons are no classical local particles, but spread out quantum particles, such ab-initio calculations are computationally very expensive, making them unfeasible for anything but very small systems and time scales.

**Monte Carlo** Another molecular simulation technique is the Monte Carlo method. This method was introduced in 1953 by Metropolis [Met53] and obtained its name because of its dependence on chances. Different from the molecular dynamics technique where precise particle trajectories are calculated, in a Monte Carlo simulation relatively large changes on the system are imposed after which it is determined whether or not the altered structure is energetically feasible at the temperature simulated. The system thus jumps abruptly from conformation to conformation, rather than evolving smoothly through time, allowing it to traverse barriers without feeling them. Many types of Monte Carlo simulations exist. For an overview see for example Ref. [Fis96].

**Method of choice** It is impossible to say that one method is better than the other. Every application has its own method that is best suited, depending both on its nature and on the level of detail desired. For example, for a fixed amount of computation time there is a trade-off between the system size and the level of detail that can be considered. Despite all computational power of even the largest modern parallel computers, the time and length scales reachable constrain the range of problems that can be addressed. With ab-initio methods, usually single molecules, interactions between two molecules, or crystals are studied. For molecular dynamics with empirical potentials the number of atoms that can be included is limited to typically  $10^7$ , corresponding to system sizes in the order of tens of nanometers. At the same time, time steps that can be made are of the order of femto-seconds, such that the very high number of  $10^9$  time steps is necessary to reach the microsecond time scale. But still, it is one or the other, small systems for 'long' time intervals or 'large' systems for short time intervals, not both at the same time. The number of particles nowadays routinely used is large compared to the first simulations



containing only tens to hundreds of atoms, but it is still very small compared to the approximately  $10^{22}$  molecules in one milliliter of water. Although this number suffices for many different applications, when larger simulations are needed, more powerful computers, more efficient simulation methods, or the combination of both are required. In this thesis several molecular simulation methods are both developed and applied on a variety of physical, chemical, mechanical and biological problems.

### 1.3 Thesis outline

As mentioned before, many macroscopic properties of materials and processes are determined at a microscopic level and molecular simulations provide a very useful tool for understanding these properties and processes. For various purposes various levels of detail are necessary. When high accuracy is needed, the interactions between the atoms should be calculated ab-initio, i.e., electronic structure calculations have to be applied. However, such ab-initio calculations are computationally very expensive, making them unfeasible for anything but very small systems.

In Chapter 2 various existing electronic structure calculation methods are discussed and a new wavelet based method is introduced in an attempt to speed up such calculations. With this wavelet based method, electronic structure can be solved efficiently and accurately by adding extra resolution locally where necessary instead of over the whole domain. This electronic structure framework is applicable for 1D, 2D and 3D problems. By applying it on several also analytically solvable cases it is proved that the method yields accurate results. Moreover, the effects of different types and orders of wavelets, different preconditioners and the advantage of pruning are shown in relation with accuracy and computational cost. Thereafter the framework is applied on a two dimensional quantum dots for the calculation of the electronic structure of quantum dot arrays and three dimensional molecular systems for the calculation of ionization potentials, electron affinities, bond properties, etc.

Such ab-initio calculations remain computationally very expensive. However, for many cases it is not necessary to calculate the exact electronic structure, but it suffices to use effective potentials to describe the atomic interactions. With highly detailed potentials many physically, chemically and biologically interesting systems can be studied in detail quantitatively. By so called coarse grained methods, in which the average behavior of several atoms is combined in a single particle and simplified potentials are used to describe the interaction between such particles, again larger systems can be considered using the same computational power.

In Chapter 3 a coarse grained model is developed for lipids, which is subsequently applied to study lipid dynamics. This includes spontaneous membrane formation from randomly dispersed lipids, spontaneous vesicle formation and vesicle fusion. These simulations have also been analyzed to determine the driving force for the vesicle formation by analyzing the potential energy landscape.

A disadvantage of molecular dynamics simulations with effective potentials is that no reactions are possible. However, many real systems, and especially biologically relevant systems, are interesting because of the reactions taking place.

In Chapter 4, therefore, the above described framework is extended with a newly developed method in which molecular dynamics is coupled with stochastic chemical reactions. Applying this new method on biologically inspired cases, that feature formation of membranes, micelles and vesicles as well growth and bursting of such vesicles, proves the feasibility and usefulness of adding such an artificial chemistry to molecular dynamics.

In Chapter 5 molecular dynamic simulations have been applied to study a completely different problem, namely the influence of the wettability of nanochannel walls on the particle and heat flow in such nanochannels. These simulations show that the influence of the gas-surface interface ultimately determines the behavior of the whole nanochannel for heat transfer as well as for flow profiles and that this influence is so important that it is dangerous to use simply boundary conditions to describe these interfaces.

However, such molecular dynamics simulations are so time consuming that only parts of nanochannels can be considered. Thus, even using effective potentials the system sizes that can be handled are sometimes too small for practical simulations. To simulate larger systems, e.g. microchannels, other techniques are required, such as other particle simulation techniques that are extensions of continuum theories. An example of such a method is the direct simulation Monte Carlo (DSMC) method, in which movements and collisions of particles occur, but where the collisions are generated stochastically with scattering rates and post-collision velocity distributions determined from the kinetic theory of gases. In Chapter 5 also this DSMC method has been applied to nanochannels. Comparison of the results with molecular dynamics simulations proves that away from the walls, i.e., in the bulk of the gas, the results are accurate, whereas the DSMC method, which uses boundary conditions to describe the walls, is not accurate enough to describe the gas-wall interaction accurately.

In case the finest details are not required in the whole domain, it is possible to combine two methods in a hybrid method, by using the more detailed method only in the area where needed and the less detailed method elsewhere, so enabling the simulation of larger systems. In the last part of Chapter 5 we couple the molecular dynamics method with the more efficient direct simulation Monte Carlo method to a hybrid approach. Molecular dynamics is used where necessary and the more efficient Monte Carlo method where possible. So in the case of our example of a microchannel, to keep the computational costs as low as possible by simulating particle behavior in the oscillation region near the wall with molecular dynamics in which the walls are also simulated accurately and the Monte Carlo technique for the particles in the bulk. In this chapter we prove that the hybrid method coupling MD and MC simulations yields accurate results while providing a speed-up of the simulations.



# 2

## Wavelets in electronic structure calculations

---

**M**ost methods for solving electronic structure calculations employ plane waves or localized bases like atomic orbitals. However, the well-known fact that electronic wave functions vary much more rapidly near the atomic nuclei than in inter-atomic regions calls for a multiresolution approach, allowing one to use low resolution and to add extra resolution only in those regions where necessary. This is provided by an alternative basis, formed of wavelets. A new electronic structure calculation method is developed that utilizes such wavelets as a basis. This method is in first instance tested on some analytically solvable problems to explain the method, to prove that it yields accurate results, to compare the effects of the application of various kinds of wavelets and preconditioners and to show the advantage of pruning. Subsequently, the method is applied to real cases, namely quantum dots and molecular systems.

Parts of this chapter are described in:

A.J. Markvoort, Ramiro Pino and P.A.J. Hilbers, **Interpolating Wavelets in Kohn-Sham Electronic Structure Calculations**, Lecture Notes in Computer Science, **2073**, 541-550 (2001).

A.J. Markvoort, P.A.J. Hilbers and Ramiro Pino, **Laterally coupled jellium-like 2D quantum dots**, Journal of Physics: Condensed Matter, **15**, 6977-6984 (2003).

---

## 2.1 Introduction

### 2.1.1 Why electronic structure calculations

To understand the macroscopic properties of a material it is necessary to have a firm comprehension of the microscopic interactions between the particles the material is built of, i.e., between its constituent atoms. Each of these atoms consists of a ‘heavy’ positively charged nucleus surrounded by very light negatively charged electrons. The interactions between the atoms depend critically on the behavior of these electrons. However, electrons are no classical particles, i.e., an electron cannot be described as a particle with a certain mass at a certain moment in time at a certain position orbiting a nucleus, obeying Newton’s laws of motion. Instead, electrons are quantum particles, obeying the laws of quantum mechanics. This means that particles are not local, but spread out. To describe such spread out electrons functions, so-called wavefunctions, are used. The wavefunction of all electrons in a molecule or a collection of atoms is also referred to as the electronic structure of that system.

The electronic structure thus ultimately determines a materials observable macroscopic properties and the chemical processes in which it is involved. Questions like what the structure of a particular molecule is, why it behaves the way it does and whether one could modify its structure and properties in a controlled, rational fashion (molecular design) deal with this electronic structure.

This electronic structure cannot be measured directly, nor can it be solved analytically for any but the very simplest systems. However, the electronic structure can be computed using numeric means, using so-called electronic structure calculations.

Accurate calculations of electronic structure are based on first principles (ab-initio) quantum mechanical treatment of the electrons. However, although the nonrelativistic equations of quantum mechanics are known since Erwin Schrödinger published them in 1925-1926, solving these equations has proved challenging. Notwithstanding the impressive computer power nowadays at our disposal, solving the basic equation of quantum mechanics, i.e. the many electron Schrödinger equation, remains a formidable task, calling for the interplay of various scientific fields, as physics, chemistry, mathematics, and computer science.

Electronic structure calculations started as an area of interest for physicists. But the fast development of computational power established the computer as a major new instrument on which experiments of chemical relevance can be performed. The continuing increase in computational power and the development of new methods and algorithms will enable the modeling of biological systems.

For this development of new methods for electronic structure calculations, thorough physical and chemical understanding of many-electron systems is required. Combining this understanding with modern mathematical concepts may lead to algorithms that exploit the peculiarities of electronic systems to yield powerful new electronic structure methods. In this drive for better methods, algorithms in which the computing time

increases linearly with respect to the number of atoms in the system are the ultimate goal.

This chapter is concerned with one of such techniques to speed up electronic structure calculations, namely with wavelets. The chapter is structured as follows. First an introduction to electronic structure calculations is given, where it is shown how the Kohn-Sham equations, that are used to calculate the electronic structure, are derived from the Schrödinger equation. Existing methods to solve these equations are shown and the reasons for using an alternative method, using wavelets, are given. In the next sections, wavelets and the way they can be used as a numeric tool are discussed. These sections, constituting mainly of a collection of theory from various book and articles cited in the text, enumerate all essential ingredients for our wavelet based electronic structure code. In the subsequent three sections this wavelets based electronic structure calculations code is presented and applied to some one, two and three dimensional systems.

## 2.1.2 Electronic structure calculations

**Schrödinger equation** Most problems in the electronic structure of matter are covered by the time-dependent Schrödinger's equation [Par89]. However, as we are concerned with atoms and molecules without time-dependent interactions, we focus here on the time-independent Schrödinger equation. For an isolated  $N$ -electron atomic or molecular system in the Born-Oppenheimer non-relativistic approximation, i.e. where the electronic and nuclear degrees of freedom are decoupled, this is given by

$$\hat{H}\Phi = E\Phi. \quad (2.1)$$

Here  $E$  is the electronic energy,  $\Phi = \Phi(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$  is the all-electron wavefunction and  $\hat{H}$  is the Hamiltonian operator

$$\hat{H} = \hat{T} + \hat{V}_{ne} + \hat{V}_{ee} + V_{nn}. \quad (2.2)$$

In this Hamiltonian

$$\hat{T} = \sum_{i=1}^N \left( -\frac{1}{2} \nabla_i^2 \right) \quad (2.3)$$

is the kinetic energy operator,

$$\hat{V}_{ne} = \sum_{i=1}^N v(\mathbf{r}_i) \quad (2.4)$$

is the nucleus-electron attraction energy operator,

$$\hat{V}_{ee} = \sum_{i < j}^N \frac{1}{r_{ij}} \quad (2.5)$$

is the electron-electron repulsion operator, and

$$V_{nn} = \sum_{\alpha < \beta}^M \frac{Z_{\alpha} Z_{\beta}}{R_{\alpha\beta}} \quad (2.6)$$

is the nucleus-nucleus repulsion energy. In Equation (2.4)  $v(\mathbf{r}_i)$  is the external potential acting on electron  $i$

$$v(\mathbf{r}_i) = - \sum_{\alpha=1}^M \frac{Z_{\alpha}}{r_{i\alpha}}, \quad (2.7)$$

the Coulombic potential due to  $M$  nuclei of charge  $Z_{\alpha}$ .

In all these equations, the coordinates  $\mathbf{x}_i$  of electron  $i$  comprise space coordinates  $\mathbf{r}_i$  and spin coordinates  $\sigma_i$ . The spatial coordinates are continuous, whereas the spin coordinates are discrete.

Equation (2.1) must be solved subject to appropriate boundary conditions.  $\Phi$  must be well-behaved everywhere, in particular decaying to zero at infinity for an atom or molecule or obeying appropriate periodic boundary conditions for a regular infinite solid.  $|\Phi|^2$  is a probability distribution function in the sense that

$$|\Phi(\mathbf{r}^N, \sigma^N)|^2 d\mathbf{r}^N \quad (2.8)$$

is the probability of finding the system with position coordinates between  $\mathbf{r}^N$  and  $\mathbf{r}^N + d\mathbf{r}^N$  and spin coordinates equal to  $\sigma^N$ , where  $\mathbf{r}^N$  stands for the set  $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N$ ,  $d\mathbf{r}^N$  for  $d\mathbf{r}_1, d\mathbf{r}_2, \dots, d\mathbf{r}_N$  and  $\sigma^N$  stands for  $\sigma_1, \sigma_2, \dots, \sigma_N$ .

There are many acceptable independent solutions of Eq. (2.1) for a given system: the eigenfunctions  $\Phi_k$  with corresponding energy eigenvalues  $E_k$ . The ground-state wavefunction and energy are denoted by  $\Phi_0$  and  $E_0$ .

Expectation values of observables are given by formulas of the type

$$\langle \hat{A} \rangle = \frac{\int d\mathbf{x}^N \Phi^* \hat{A} \Phi}{\int d\mathbf{x}^N \Phi^* \Phi} = \frac{\langle \Phi | \hat{A} | \Phi \rangle}{\langle \Phi | \Phi \rangle}, \quad (2.9)$$

where  $\int d\mathbf{x}^N$  means here the integration over  $3N$  spatial coordinates and the summation over  $N$  spin coordinates. Particular measurements give particular values of  $\hat{A}$ , but many measurements average to  $\langle \hat{A} \rangle$ .

When a system is in a state  $\Phi$  the expectation value of the total energy is given by

$$E[\Phi] = \frac{\langle \Phi | \hat{H} | \Phi \rangle}{\langle \Phi | \Phi \rangle}. \quad (2.10)$$

Since each particular measurement of the energy gives one of the eigenvalues of  $\hat{H}$ , we have

$$E[\Phi] \geq E_0, \quad (2.11)$$

i.e., the energy computed from a guessed  $\Phi$  is an upper-bound to the ground-state energy  $E_0$ . Full minimization of the functional  $E[\Phi]$  with respect to all allowed  $N$ -electron wavefunctions will give the true ground-state  $\Phi_0$  and energy  $E[\Phi_0] = E_0$ , i.e.,

$$E_0 = \min_{\Phi} E[\Phi]. \quad (2.12)$$

Every eigenstate  $\Phi$  is an extremum of the functional  $E[\Phi]$ . In other words, one may replace the Schrödinger equation with the variational principle

$$\delta E[\Phi] = 0. \quad (2.13)$$

When Eq. (2.13) is satisfied, so is Eq. (2.1), and vice versa. It is convenient to restate this equation in a way that will guarantee that the final  $\Phi$  will automatically be normalized.

This can be done by the method of Lagrange multipliers, giving

$$\delta[\langle\Phi|\hat{H}|\Phi\rangle - E\langle\Phi|\Phi\rangle] = 0. \quad (2.14)$$

Most contemporary calculations on electronic structure are done with this variational procedure, in some linear algebraic implementation.

**Density functional theory** The ground-state energy and the ground-state wavefunction are determined by the minimization of the energy functional  $E[\Phi]$ . For an  $N$ -electron system the external potential, however, completely fixes the Hamiltonian.  $N$  and  $v(\mathbf{r})$  determine thus all properties for the ground-state, which is of course not surprising since  $v(\mathbf{r})$  determines the whole nuclear frame for a molecule, which together with the number of electrons determines all the electronic properties.

The first Hohenberg-Kohn theorem [Hoh64] states that the energy can be written as a functional of the electron density only, instead. This electron density  $\rho(\mathbf{r})$  for a given state is defined as the number of electrons per unit volume in that state. Its formula in terms of  $\Phi$  is

$$\rho(\mathbf{r}_1) = N \sum_{\sigma_1} \int d\mathbf{x}_2 \cdots \int d\mathbf{x}_N |\Phi(\mathbf{x}^N)|^2. \quad (2.15)$$

The principle quantity of interest changes thus from the all-electron wavefunction to this electron density. This is very appealing since it has the computational advantage that the electron density has only three spatial coordinates, regardless of the number of electrons in the system, thus making the computation of much larger systems computationally feasible.

The ground-state energy of a many-electron system can now be obtained as the minimum of the energy functional of the density only

$$E[\rho] = \int d\mathbf{r} \rho(\mathbf{r})v(\mathbf{r}) + F[\rho] + V_{\text{nn}}, \quad (2.16)$$

where

$$F[\rho] = T[\rho] + V_{\text{ee}}[\rho] \quad (2.17)$$

and  $T[\rho]$  and  $V_{\text{ee}}[\rho]$  are the kinetic energy functional and the electron-electron interaction functional respectively. The second Hohenberg-Kohn theorem provides the energy variational principle

$$E_0 \leq E[\rho] \quad (2.18)$$

for non-negative  $\rho(\mathbf{r})$ , where  $\rho(\mathbf{r})$  is also normalized to the number of electrons. The ground-state electron density is thus the density that minimizes  $E[\rho]$  and hence satisfies the Euler equation

$$\mu = v(\mathbf{r}) + \frac{\delta F[\rho]}{\delta \rho(\mathbf{r})} \quad (2.19)$$

where  $\mu$  is the Lagrange multiplier associated with the constraint

$$\int d\mathbf{r} \rho(\mathbf{r}) = N. \quad (2.20)$$

However, the problem is that no explicit form for  $F[\rho]$  is known. Thomas-Fermi [Tho27, Fer27] and related models constitute a direct approach to solve this problem, whereby



one constructs explicit approximate forms for  $T[\rho]$  and  $V_{ee}[\rho]$ . The simplest approximation is the traditional Thomas-Fermi model where  $V_{ee}[\rho]$  is replaced by its classical part  $J[\rho]$ , which is given by

$$J[\rho] = \frac{1}{2} \int d\mathbf{r} \int d\mathbf{r}' \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}, \quad (2.21)$$

and where  $T[\rho]$  is taken from the theory of a non-interacting uniform electron gas, resulting in

$$T_{TF}[\rho] = C_F \int d\mathbf{r} \rho(\mathbf{r})^{5/3}, \quad (2.22)$$

where  $C_F = \frac{3}{10}(3\pi^2)^{2/3}$ . The Thomas-Fermi-Dirac model extends this model by adding the exchange-energy formula for a uniform electron gas as an approximation for the non-classical part of  $V_{ee}[\rho]$ .

This approach thus produces a nice simplicity as the equations involve electron density only. However, because the exact forms of the functionals are unknown and approximate functionals have to be used instead, these methods still lack high accuracy. The method improves with the development of better and better functionals, which e.g., also depend on the gradient of the density, but our efforts in this direction [Pin03b, Pin03c, Pin03d, Pin03a] will not be described here. Instead, an indirect approach to the kinetic energy functional is used, since this kinetic energy functional is of major importance.

**Kohn-Sham method** In a trade of simplicity for accuracy, Kohn and Sham invented an ingenious indirect approach to the kinetic energy functional  $T[\rho]$ , the Kohn-Sham (KS) method [Koh65], which removes the necessity of knowing the exact form of  $T[\rho]$ . Kohn and Sham proposed introducing orbitals into the problem in such a way that the kinetic energy can be computed simply to good accuracy, leaving a small residual correction to be handled separately. This is done by minimizing the kinetic energy of a non-interacting system of electrons with respect to the set of single-particle orbitals  $\Psi_i$ , which obey the orthogonality relation  $\langle \Psi_i | \Psi_j \rangle = \delta_{ij}$ . This kinetic energy is denoted by

$$T_s[\rho] = -\frac{1}{2} \sum_i \langle \Psi_i | \nabla^2 | \Psi_i \rangle. \quad (2.23)$$

To produce the desired separation out of  $T_s[\rho]$  as the kinetic energy component rewrite

$$F[\rho] = T_s[\rho] + J[\rho] + E_{xc}[\rho] \quad (2.24)$$

where

$$E_{xc}[\rho] \equiv T[\rho] - T_s[\rho] + V_{ee}[\rho] - J[\rho] \quad (2.25)$$

is called the exchange-correlation energy, containing the difference between  $T[\rho]$  and  $T_s[\rho]$  and the non-classical part of  $V_{ee}[\rho]$ . The Euler equation now becomes

$$\mu = v_{\text{eff}}(\mathbf{r}) + \frac{\delta T_s[\rho]}{\delta \rho(\mathbf{r})} \quad (2.26)$$

where the KS effective potential is defined by

$$v_{\text{eff}}(\mathbf{r}) = v(\mathbf{r}) + \int d\mathbf{r}' \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + V_{xc}(\mathbf{r}) \quad (2.27)$$

with the exchange-correlation potential

$$V_{\text{xc}}(\mathbf{r}) = \frac{\delta E_{\text{xc}}[\rho]}{\delta \rho(\mathbf{r})}. \quad (2.28)$$

Equation (2.26) with constraint (2.20) is precisely the same equation as one obtains from conventional density functional theory when one applies it to a system of non-interacting electrons moving in the external potential  $v_{\text{eff}}(\mathbf{r})$ . Therefore, for a given  $v_{\text{eff}}(\mathbf{r})$ , one obtains the density  $\rho(\mathbf{r})$  that satisfies Eq. (2.26) simply by solving the  $N$  one-electron equations

$$\left(-\frac{1}{2}\nabla^2 + v_{\text{eff}}(\mathbf{r})\right) \Psi_i(\mathbf{r}) = \varepsilon_i \Psi_i(\mathbf{r}), \quad (2.29)$$

and setting

$$\rho(\mathbf{r}) = \sum_i |\Psi_i(\mathbf{r})|^2. \quad (2.30)$$

Here,  $v_{\text{eff}}(\mathbf{r})$  depends on the density  $\rho(\mathbf{r})$  and should be solved self-consistently.

The total energy is not just the sum of the orbital energies, but can be obtained from the resultant density via

$$E[\rho] = \sum_i \int d\mathbf{r} \Psi_i^* \left(-\frac{1}{2}\nabla^2\right) \Psi_i + J[\rho] + E_{\text{xc}}[\rho] + \int d\mathbf{r} v(\mathbf{r})\rho(\mathbf{r}) + V_{\text{nn}}, \quad (2.31)$$

or equivalently from the formula

$$E[\rho] = \sum_{i=1}^N \varepsilon_i - J[\rho] + E_{\text{xc}}[\rho] - \int d\mathbf{r} V_{\text{xc}}(\mathbf{r})\rho(\mathbf{r}) + V_{\text{nn}}. \quad (2.32)$$

Through the introduction of the  $N$  orbitals  $T_s[\rho]$ , the dominant part of the true kinetic energy  $T[\rho]$ , is solved indirectly, but exactly. The price for this gain in accuracy is that there are now  $N$  equations to solve as opposed to only one equation for the total density derived from direct approximation on  $T_s[\rho]$  of the Thomas-Fermi type. The other unknown part  $E_{\text{xc}}$  remains intact. The KS equations are open for improvement with each successive better approximation to  $E_{\text{xc}}[\rho]$  and would give exact  $\rho$  and  $E$  if  $E_{\text{xc}}[\rho]$  were known precisely.

Since the effective potential  $v_{\text{eff}}(\mathbf{r})$  does not contain electron spin, the solutions  $\Psi_i(\mathbf{r})$  are doubly degenerate, one with spin up and one with spin down. A better method, as we will see later, is the spin polarized extension of the local-spin-density-approximation (LSD) which turns out to be superior to local-density-approximation (LDA).

This Kohn-Sham approach is used in this work. An explicit form for  $E_{\text{xc}}[\rho]$  is still needed to specify the Kohn-Sham equations completely, but this will be discussed later.

### 2.1.3 Why wavelets

**Basis sets** Having selected a specific electronic structure method the next choice is related to which basis set to use in order to represent the orbitals  $\Psi_i$ . In general a linear combination of basis functions is used  $\Psi_i(\mathbf{r}) = \sum_{\mathbf{k}} c_{i\mathbf{k}} f_{\mathbf{k}}(\mathbf{r}, \{\mathbf{R}_I\})$ , where the basis functions may depend on the nuclear positions  $\{\mathbf{R}_I\}$ . The variety of basis sets utilized is large. Among these bases, LCAO and plane waves are most known, but other bases

include mixed and augmented basis sets, Wannier functions, finite-elements, multigrids and last but not least wavelets.

**LCAO** In quantum chemistry, Slater-type basis functions  $f_{\mathbf{k}}^S(\mathbf{r}) = C_{\mathbf{k}}^S x^{k_x} y^{k_y} z^{k_z} e^{-\alpha_{\mathbf{k}}^S |\mathbf{r}|}$  with an exponentially decaying radial part and Gaussian type basis functions  $f_{\mathbf{k}}^G(\mathbf{r}) = C_{\mathbf{k}}^G x^{k_x} y^{k_y} z^{k_z} e^{-\alpha_{\mathbf{k}}^G |\mathbf{r}|^2}$  are widely used [Heh86]. In these functions  $C$  and  $\alpha$  are constants. These basis functions are in general centered at the positions of the nuclei, i.e., a so-called linear combination of atomic orbitals (LCAO). They have the advantage of easy (analytic) evaluation of derivatives and other matrix elements, by differentiation and integration in real-space. Furthermore, because of their shape relatively few basis functions are needed, but no fast transform exists. Another drawback arises when the atoms are allowed to move, either in geometry optimization or molecular dynamics schemes. For bases that have basis functions fixed at atoms (or bonds) so called Pulay forces [Pul69] will result. These Pulay forces represent the derivative of the basis functions with respect to the positions of the ions. This is not an unbridgeable problem, but means that these forces have to be calculated.

**Plane waves** Plane waves on the other hand originate from solid-state physics, where the ubiquitous periodicity of the underlying lattice imposes the same periodicity on the density, suggesting the use of periodic basis functions. These plane waves are defined as  $f_{\mathbf{k}}^{\text{PW}}(\mathbf{r}) = C^{\text{PW}} e^{i\mathbf{k} \cdot \mathbf{r}}$ . Plane waves form a complete and orthonormal set and the labeling of the basis functions is simply given by the vector  $\mathbf{k}$  in reciprocal space. The plane wave approach is reviewed in detail in [Pay92].

An important advantage is that plane waves are originless functions, i.e., they do not depend on the positions of the nuclei, implying that the Pulay forces vanish exactly, which tremendously facilitates force calculations. Because of this delocalization plane waves also form a very unbiased basis set, not favoring one region over others. The only way to improve the quality of the basis is to increase the energy cut-off, i.e., to increase the largest  $\mathbf{k}$ -vector that is included in the finite expansion. Another appealing feature is that derivatives in real-space are simply multiplications in reciprocal space, and both spaces can be efficiently connected via the Fast Fourier Transform (FFT), making that the total energy is found to have a particularly simple form when expressed in plane waves.

However, there are also disadvantages to the use of plane waves. The extremely high resolution required near atomic nuclei combined with the uniform resolution afforded by plane waves makes the direct application of this method prohibitive for all but the light elements. The introduction of pseudopotential theory overcomes this limitation, at the cost of an uncontrolled pseudopotential approximation. However, even in pseudopotential calculations the wavefunctions require significant resolution near ionic cores, particularly when dealing with first-row elements or transition metals. Another severe shortcoming of plane waves is the backside of the medal of being an unbiased basis

set: there is no way to shuffle more basis functions into regions in space where they are more needed than in other regions. This is particularly bad for systems with strong inhomogeneities, like a few heavy atoms in a sea of light atoms, or finite systems such as surfaces or molecules with a large vacuum region in order to allow the long-range Coulomb interactions to decay. This is often referred to as the multiple length scale deficiency of plane wave calculations.

Localized Gaussian basis functions on the one hand and plane waves on the other hand are certainly two extreme cases, but most used. There has been a tremendous effort to combine such localized and originless basis functions resulting in a rich collection of mixed and augmented basis sets.

**Real-space methods** A quite different approach is to leave conventional basis set approaches altogether and to resort to real-space methods where continuous space is replaced by discrete space. This entails that the derivative operator or the entire energy expression has to be discretized in some way. The first real-space approach devised for ab-initio molecular dynamics was based on the lowest-order finite difference approximation in conjunction with an equally spaced cubic mesh in real-space [Car87]. A variety of other more sophisticated real-space methods include multigrids [Whi89, Wes92] and finite-element methods [Tsu95]. Among the chief advantages of the real-space methods is that linear scaling approaches can be implemented in a natural way and that the multiple length scale problem can be coped with by adapting the grid. Finite-element methods also represent an efficient method for dealing with non-linear interactions by providing highly efficient rapid transforms. However, a difficulty is that each element in a finite-elements representation corresponds to the value over the region of one basis function and thus cannot be taken to be small where the electronic orbitals themselves are non-negligible. The extension to non-uniform meshes also induces Pulay forces if the mesh moves as the nuclei move.

**Wavelets** Wavelet bases [Dau92a] place functions of varying resolution on a multiresolution grid while maintaining a uniform resolution throughout all of space in the precise mathematical sense of multiresolution analysis. The mathematical regularity of the resulting basis leads to efficient fast transforms and methods to apply (differential) operators. In contrast to the expansion coefficients of a finite-elements expansion which reflect directly the values of a function, the coefficients of a multiresolution analysis separate information into different length scales. This subtle but critically important difference means that, as long as a function varies smoothly, the fine-scale coefficients will be negligible small even where the value of a function is quite large. This means that, as the atoms move, one may arrange for the changes in the basis to involve the truncation of only coefficients which are small, thereby effectively providing high resolution throughout all of space with an extremely limited number of coefficients. This also means that no particular care is needed to handle the regridding as the atoms move.

The use of wavelets in electronic structure calculations was introduced by Cho *et al.* [Cho93] who employed (Mexican hat) wavelets in solving the Schrödinger equation for Hydrogen like atoms. Later, self-consistent local density approximation (LDA) calculations on  $H_2$  and  $O_2$  using Daubechies wavelets were reported [Wei96, Tym97], followed by the introduction of interpolating wavelets by Lippert *et al.* [Lip98]. A review on wavelets in electronic structure has been given by Arias [Ari99].

### 2.1.4 Summary

Electronic structure calculations come in various kinds. This work focuses on solving electronic structure via the so-called Kohn-Sham equations.

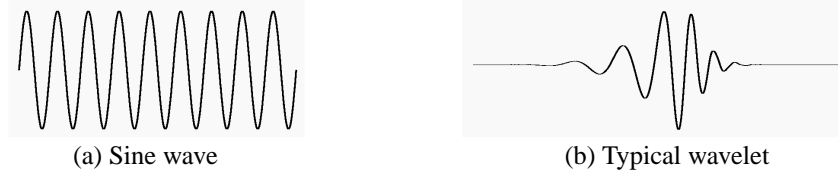
Most methods for solving electronic structure calculations employ plane waves or atomic orbitals (LCAO). Plane waves have the advantage of being orthonormal and complete, permitting systematic convergence and straightforward evaluation of forces, which is important for the extension to molecular dynamics simulations. But they are not efficient in describing localized orbitals and wavefunctions in surfaces or clusters. On the other hand localized bases, like LCAO, are usually over-complete, lack explicit convergence properties and result in difficult force calculations (Pulay forces).

However, the well-known fact that electronic wave functions vary much more rapidly near the atomic nuclei than in inter-atomic regions calls for a multiresolution approach, allowing one to use low resolution and to add extra resolution only in those regions where necessary. This is provided by an alternative basis, formed of wavelets, which allow accurate description over a range of length scales.

## 2.2 Wavelets

**What are wavelets** The name *wavelet* was devised around 1980, making it a relatively young topic. However, techniques similar to wavelets were used already from the 1930s in different fields in mathematics, physics and engineering long before they were given the status of a unified scientific field. As a consequence of these interdisciplinary origins, wavelets appeal to scientists and engineers of many different backgrounds. Besides, and most important, they are a mathematical tool with a great variety of possible applications. Nowadays wavelets are used in areas like signal processing, image processing, data compression, statistics, mathematics, physics, econometrics and computer graphics. Since wavelets are such a popular topic many books have been written on this subject within the past ten years. A number of them are mentioned in the references, e.g. [Dau92a, Bur98, Nie98b].

Literally, wavelet means *little wave*. A wave is usually defined as an oscillating function of time or space, such as a sinusoid. Wavelets are called little waves because their oscillations are localized, i.e., they have their energy concentrated. Typical examples of a wave, extending from minus infinity to plus infinity, and a wavelet, localized around a



**Figure 2.1:** Typical example of (a) a wave and (b) a wavelet. A wave extends over the whole line, whereas a wavelet has its oscillations localized.

point, are shown in Figure 2.1.

A wavelet is thus an arbitrary function that consists of some localized oscillations. Most often such wavelets are not known in closed form, but are given by means of a recursive relation, called the dilation equation. The reason for this is multiresolution theory that is discussed later.

Wavelets themselves are not that interesting as the tool they are used for, the wavelet transform. Like sinusoids, or equivalently complex exponentials, are used for Fourier analysis, which is known to be a very valuable mathematical tool, analogously, wavelets are used for the wavelet transform. Because of their locality, wavelets can be used to provide a tool for the analysis of transient, non stationary or time-varying phenomena.

The wavelet transform breaks up a function (or signal) into component pieces, which are processed further instead of the original function. The pieces the function is broken up into are the wavelets, i.e. the family of translations and dilations of one single wavelet, called the mother wavelet. These translations and dilations are denoted respectively by  $b$  and  $a$  in the following equation

$$\psi_{a,b}(x) = |a|^{-1/2} \psi\left(\frac{x-b}{a}\right). \quad (2.33)$$

The opposite process, rebuilding the original function out of its component pieces is called the inverse wavelet transform, analogous to the inverse Fourier transform.

The reasons for breaking a signal into pieces and putting it again together afterward may depend on the application. However, globally the reasons can be divided into two parts. In the first place, the original function may contain more information than can be seen at first sight, but which can be made visible using the wavelet transform. In the second place, the wavelet transform may be used to compress data. This is advantageous when the data have to be stored or when calculations or operations have to be performed on the data.

**Wavelet transform** In Fourier models functions are represented as a weighted sum of exponentials at different frequencies. The weight at each different frequency is the Fourier coefficient. Wavelet models analogously represent functions as a weighted sum of scaled and translated mother wavelets. The wavelet transform has a mother wavelet replace the exponential, scaling and translation replace the frequency shifting and a two dimensional surface of wavelet coefficients replace the one dimensional Fourier coefficients.

The formula of the well-known Fourier transform is

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dx f(x) e^{-i\omega x}. \quad (2.34)$$

Such a Fourier transform gives the frequency spectrum  $\hat{f}(\omega)$  of the function  $f(x)$ . But its disadvantage is that there is no position-frequency localization. This can be introduced with the Windowed Fourier transform

$$\hat{f}(\omega, x) = \int_{-\infty}^{\infty} dy f(y) g(y - x) e^{-i\omega y}. \quad (2.35)$$

$\hat{f}(\omega, x)$  all consist of the same envelope function  $g(y)$ , translated to the proper location and ‘filled in’ with higher frequency oscillations. The next step in this line is the wavelet transform

$$\hat{f}(a, b) = |a|^{-1/2} \int_{-\infty}^{\infty} dx f(x) \psi\left(\frac{x - b}{a}\right). \quad (2.36)$$

The wavelets  $\psi_{a,b}$  have widths adapted to their frequency. Higher frequency  $\psi_{a,b}$  are very narrow, while low frequency  $\psi_{a,b}$  are much broader. As a result, the wavelet transform is better able than the windowed Fourier transform to zoom in onto very local high frequency phenomena.

Now we have seen what wavelets are and how we can use them in the wavelet transform, we also want to be able to reconstruct our original function  $f(x)$ . Like the inverse Fourier transform

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} d\omega \hat{f}(\omega) e^{i\omega x} \quad (2.37)$$

we also want to introduce the inverse wavelet transform

$$f(x) = C_{\psi}^{-1} \int_{-\infty}^{\infty} db \int_{-\infty}^{\infty} da \frac{1}{a^2} \hat{f}(a, b) \psi_{a,b}(x), \quad (2.38)$$

where

$$C_{\psi} = 2\pi \int_{-\infty}^{\infty} d\omega |\hat{\psi}(\omega)|^2 |\omega|^{-1}. \quad (2.39)$$

This constant should exist. As a result, this is a condition our wavelet has to satisfy. Up to now we could choose any arbitrary function as our mother wavelet, but this condition means that admissible functions are those that cycle (oscillate), have finite energy and have an average value of zero.

However, the set of possible mother wavelets is still tremendously large. As a result, a valid statement for one mother wavelet can be completely invalid for another wavelet. As we will see in the next sections, we can use this freedom to choose wavelets which have the properties we are interested in.

**Discrete wavelet transform** Also analogous to the Fourier theory, the continuous wavelet transform is not employed as often as the discrete wavelet transform. The continuous transforms are primarily employed to derive properties and the discrete forms are necessary for most computer implementations. The discrete Fourier transform

$$\hat{f}_n = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dx f(x) e^{-in\omega_0 x} \quad (2.40)$$

and its inverse

$$f(x) = \frac{1}{\sqrt{2\pi}} \sum_{n=-\infty}^{\infty} \hat{f}_n e^{in\omega_0 x} \quad (2.41)$$

still have a continuous independent variable  $x$ . Only the frequencies  $\omega$  have been discretized ( $n\omega_0$ ).

The same holds for the discrete wavelet transform where the scales  $a$  and dilations  $b$  are discrete, but not the independent variable  $x$ . The discrete wavelet transform can be written as

$$d_k^j = a_0^{-j/2} \int_{-\infty}^{\infty} dx f(x) \psi\left(\frac{x - kb_0}{a_0^j}\right) \quad (2.42)$$

and its inverse as

$$f(x) = C \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} d_k^j \psi_k^j(x), \quad (2.43)$$

where  $C$  is a normalization constant and

$$\psi_k^j(x) = a_0^{-j/2} \psi\left(\frac{x - kb_0}{a_0^j}\right). \quad (2.44)$$

**Multiresolution** There exists a tremendous choice of mother wavelets, since a mother wavelet can be any function that obeys condition (2.39). This freedom can be used to impose some extra conditions. By doing this in a smart way, sets of wavelets with useful properties can be created.

The first degrees of freedom are usually used to obtain a multiresolution approach. The standard application of the multiresolution wavelet transform is to form a series of half-band filters that divide a spectrum into a high frequency band and a low frequency band. The high frequency band output is taken as the wavelet transform coefficients for a fine scale and the low frequency band output is decimated by a factor 2. This low frequency band is then split into a high and a low band, etc.

The primary constraint on the mother wavelet in such a multiresolution approach is really formulated on a different function, which is called the scaling function  $\phi$ . This is done by introducing a ladder of subspaces  $(\mathcal{V}_j)_{j \in \mathbb{Z}}$  which represent the successive resolution levels

$$(i) \quad \dots \subset \mathcal{V}_2 \subset \mathcal{V}_1 \subset \mathcal{V}_0 \subset \mathcal{V}_{-1} \subset \mathcal{V}_{-2} \subset \dots \quad (2.45a)$$

$$(ii) \quad \bigcap_{j \in \mathbb{Z}} \mathcal{V}_j = \{0\}, \quad \overline{\bigcup_{j \in \mathbb{Z}} \mathcal{V}_j} = L^2(\mathbb{R}). \quad (2.45b)$$

The multiresolution is introduced by

$$(iii) \quad f \in \mathcal{V}_j \longleftrightarrow f(2^j \cdot) \in \mathcal{V}_0 \text{ for all } j \in \mathbb{Z}, \quad (2.45c)$$

what states that all spaces are scaled versions of the central space  $\mathcal{V}_0$ . The next requirement is invariance of  $\mathcal{V}_0$  under integer translations

$$(iv) \quad f \in \mathcal{V}_0 \longrightarrow f(\cdot - k) \in \mathcal{V}_0 \text{ for all } k \in \mathbb{Z}. \quad (2.45d)$$



The final requirement is the existence of an orthonormal basis

$$(v) \quad \exists \varphi \in \mathcal{V}_0 \text{ so that the } \varphi_k^0, k \in \mathbb{Z}, \text{ constitute an orthonormal basis for } \mathcal{V}_0, \quad (2.45e)$$

where  $\varphi$  is the scaling function with translations and dilations

$$\varphi_k^j(x) = 2^{-j/2} \varphi(2^{-j}x - k). \quad (2.46)$$

Requirements (iii) and (v) together imply that  $\{\varphi_k^j : k \in \mathbb{Z}\}$  is an orthonormal basis for  $\mathcal{V}_j$  for all  $j \in \mathbb{Z}$ . The last requirement (v) can be relaxed considerably, in fact, we only need to require that the  $\{\varphi_k^0 : k \in \mathbb{Z}\}$  constitute a Riesz basis, i.e., there exist constants  $A$  and  $B$  such that for any finite set of integers  $\Lambda \subset \mathbb{Z}$  and real numbers  $\lambda_i$  with  $i \in \Lambda$  holds that

$$A \sum_{i \in \Lambda} \lambda_i^2 \leq \left\| \sum_{i \in \Lambda} \lambda_i \varphi_i^0 \right\|_2^2 \leq B \sum_{i \in \Lambda} \lambda_i^2. \quad (2.47)$$

But there are several advantages to require the scaling functions and the wavelets to be orthonormal. Namely, because of the orthogonality the projection operator on the subspace  $\mathcal{V}_j$

$$P_j : L^2(\mathbb{R}) \rightarrow \mathcal{V}_j \quad (2.48)$$

can be defined as:

$$(P_j f)(x) = \sum_{k \in \mathbb{Z}} \langle f, \varphi_k^j \rangle \varphi_k^j(x), \quad (2.49)$$

where  $\langle f, \varphi_k^j \rangle$  denotes the inner product  $\int dx f(x) \varphi_k^j(x)$ .

The beauty of this multiresolution approach is that whenever a ladder of spaces satisfies the conditions (i) to (iv), then there exists a  $\psi$  such that

$$P_{j-1}f = P_j f + \sum_{k \in \mathbb{Z}} \langle f, \psi_k^j \rangle \psi_k^j, \quad (2.50)$$

where the translations and dilations of the mother wavelet  $\psi$  are defined equivalently to Eq. (2.46) as

$$\psi_k^j(x) = 2^{-j/2} \psi(2^{-j}x - k). \quad (2.51)$$

This can be understood in the following way. Since  $\mathcal{V}_j$  is a subset of  $\mathcal{V}_{j-1}$ , the space  $\mathcal{W}_j$  that is included in  $\mathcal{V}_{j-1}$ , but not in  $\mathcal{V}_j$ , can be spanned by the orthogonal wavelets  $\psi_k^j$ . Thus,  $\mathcal{V}_j$  and  $\mathcal{W}_j$  are related by  $\mathcal{V}_j = \mathcal{V}_{j+1} \oplus \mathcal{W}_{j+1}$ . Because the spaces  $\mathcal{V}_j$  are spanned by the scaling functions, they are called the scaling spaces, whereas the  $\mathcal{W}_j$  are called the wavelet spaces because they are spanned by the wavelets. It turns out that there are many examples of such multiresolution analysis ladders, corresponding to still many orthonormal wavelet bases.

There exists an explicit recipe for the construction of this  $\psi$ . Since  $\varphi \in \mathcal{V}_0 \subset \mathcal{V}_{-1}$  and the  $\varphi_n^{-1}(x) = \sqrt{2} \varphi(2x - n)$  constitute an orthonormal basis for  $\mathcal{V}_{-1}$ , there exist  $h_n = \sqrt{2} \langle \varphi, \varphi_n^{-1} \rangle$  such that:

$$\varphi(x) = \sum_{n \in \mathbb{Z}} h_n \varphi(2x - n). \quad (2.52)$$

This latter equation is often referred to as the *dilation equation*. Also the wavelet  $\psi(x)$  can be written as a linear combination of scaling functions of one resolution higher

$$\psi(x) = \sum_{n \in \mathbb{Z}} g_n \varphi(2x - n). \quad (2.53)$$

Since  $\psi(x)$  has to be orthonormal to  $\phi(x)$ , the condition  $\sum_n g_n h_n = 0$  has to be satisfied. As a result, it then suffices to take

$$g_n = (-1)^n h_{-n+1} \quad (2.54)$$

for all  $n$ . Eq. (2.53) is denoted as the wavelet equation and the coefficients  $\{h_n\}$  and  $\{g_n\}$  as the filter coefficients. It is important to note that once either one of  $\phi(x)$ ,  $\psi(x)$ ,  $\{h_n\}$  and  $\{g_n\}$  is known, the rest are also determined via the Eqs. (2.52) - (2.54).

Combining Eq. (2.45b) and the fact that  $\mathcal{V}_j$  and  $\mathcal{W}_j$  are related by  $\mathcal{V}_j = \mathcal{V}_{j+1} \oplus \mathcal{W}_{j+1}$  yields  $\bigoplus_{j \in \mathbb{Z}} \mathcal{W}_j = L^2(\mathbb{R})$ . As a result any function  $f(x) \in L^2(\mathbb{R})$  can be written as

$$f(x) = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} d_k^j \psi_k^j(x), \quad (2.55)$$

where due to the orthogonality of the wavelets

$$d_k^j = \int_{-\infty}^{\infty} dx f(x) \psi_k^j(x). \quad (2.56)$$

In practice there is a limit to how small the smallest structures and how large the largest structures are. The smallest structures might be determined for example by the finest grid in a numerical computation or the sampling frequency of a signal. Therefore, an expansion would take place in a space  $\mathcal{V}_0$  which can represent these smallest structures. At the same time, because of the upper bound to the structure sizes, the splitting of  $\mathcal{V}_0$  in wavelet spaces  $\mathcal{W}_j$  can be truncated at level  $J$ , resulting in  $\mathcal{V}_0 = \mathcal{W}_1 \oplus \mathcal{W}_2 \oplus \mathcal{W}_3 \oplus \dots \oplus \mathcal{W}_J \oplus \mathcal{V}_J$ . The projection can thus be written in two ways

$$\begin{aligned} P_0 f(x) &= \sum_{k \in \mathbb{Z}} s_k^0 \phi_k^0(x) \\ &= \sum_{k \in \mathbb{Z}} s_k^J \phi_k^J(x) + \sum_{j=1}^J \sum_{k \in \mathbb{Z}} d_k^j \psi_k^j(x), \end{aligned} \quad (2.57)$$

where

$$s_k^j = \int_{-\infty}^{\infty} dx f(x) \phi_k^j(x) \quad (2.58)$$

and

$$d_k^j = \int_{-\infty}^{\infty} dx f(x) \psi_k^j(x). \quad (2.59)$$

The wavelet transform converts one representation to the other. These coefficients  $s_k^j$  and  $d_k^j$  are called scaling coefficients and wavelet coefficients respectively.

**Line versus interval** So far, the summations over the translations  $k$ , e.g., in Eq. (2.57), run over all integers, i.e., from  $-\infty$  to  $\infty$ . But in applications wavelets are often used on an interval instead of on the whole line. For the transition from wavelets on the line to wavelets on an interval, four different interesting solutions exist, viz.:

- adding zeros,
- periodizing,
- reflecting edges, and
- constructing special edge wavelets.

The first solution, adding zeros, is actually the same as doing nothing at all. A function  $f(x)$  supported on an interval can always be extended to the whole line by putting  $f(x) = 0$  outside the interval. There are two disadvantages to this naive approach. Firstly, such extensions may introduce discontinuities at the borders of the interval, typically resulting in large wavelet coefficients for fine scales near these edges. Secondly, extending the interval, by adding zeros, results in the use of more, i.e. too many, wavelets.

A second solution is to extend the interval on both sides with copies of itself. This is equivalent to using periodized wavelets on the original interval. For (quasi) periodic functions this is a very good solution. However, in case the function on the interval is not periodic itself this method results in the same problem of having a discontinuity as with adding zeros.

A third solution is reflecting at the edges. In this case, the function on the interval is extended by mirroring it at the edges [Coh92, Coh93]. This is equivalent to using folded wavelets on the original interval. The problem of the discontinuities has been solved now, but this method has some drawbacks too. Firstly, the discontinuity has been replaced by a discontinuity in the first derivative. Secondly, starting with an orthonormal wavelet basis, this folding typically does not lead to an orthonormal wavelet basis on the interval.

All drawbacks of the previous methods can be overcome by constructing some new special wavelets for the edge wavelets. Meyer [Coh93, Mey92] constructed such edge wavelets first, but his method still has the drawback of having more scaling functions than wavelets at a certain resolution level. A better method, focused on the Daubechies wavelets, has been suggested by Cohen *et al* [Coh93, Dau93a, Dau93b].

We have used the periodized wavelets and the special edge wavelets because they enable the expansion of a function on the interval in an orthonormal basis on that interval. The use of periodized wavelets is preferred where possible, because for the special edge wavelets also special edge transform and operator coefficients have to be used, which may result in errors on the boundaries.

**Daubechies wavelets** As mentioned above, there are many examples of multiresolution analysis ladders, corresponding to still many orthonormal wavelet bases. The remaining degrees of freedom can be used to add extra useful properties. One of the most known types of wavelets is the family of Daubechies wavelets. These Daubechies wavelets are a family of wavelets which are orthonormal, have compact support and where the other degrees of freedom are used for smoothness of the wavelets. In this paragraph we construct these wavelets by requiring consecutively:

- Multi-resolution,
- Orthonormality,
- Compact support, and
- Smoothness.

As we have seen above, multi-resolution requires that the scaling functions obey the dilation equation

$$\varphi(x) = \sum_{n \in \mathbb{Z}} h_n \varphi(2x - n). \quad (2.60)$$

Thus, the scaling function  $\varphi(x)$  is determined once the so-called filter coefficients  $\{h_n\}$  are known. Using this scaling function and the coefficients  $\{h_n\}$  again, the corresponding wavelets are defined as

$$\psi(x) = \sum_{n \in \mathbb{Z}} g_n \varphi(2x - n), \quad (2.61)$$

where the coefficients  $\{g_n\}$  are given by the relation

$$g_n = (-1)^n h_{2M-1-n}, \quad (2.62)$$

where  $2M$  is the compact support length, i.e., the length of the interval where both the wavelet and the scaling function may be non-zero. Thus, both the scaling functions and the corresponding wavelets are determined once the filter coefficients  $\{h_n\}$  are determined.

The first requirement on these filter coefficients comes from the orthonormality constraint on the scaling functions and the wavelets. This constraint consists of two parts. Firstly, the scaling function should be normalized, and secondly, the scaling functions should be orthogonal. Normalization of the scaling function means

$$\int_{-\infty}^{\infty} dx \varphi(x) = 1. \quad (2.63)$$

Integrating the dilation equation and substituting this normalization constraint yields the normalization constraint in terms of the filter coefficients

$$\sum_{n \in \mathbb{Z}} h_n = 2. \quad (2.64)$$

The other part of the requirement is that the scaling functions are orthonormal, i.e.,

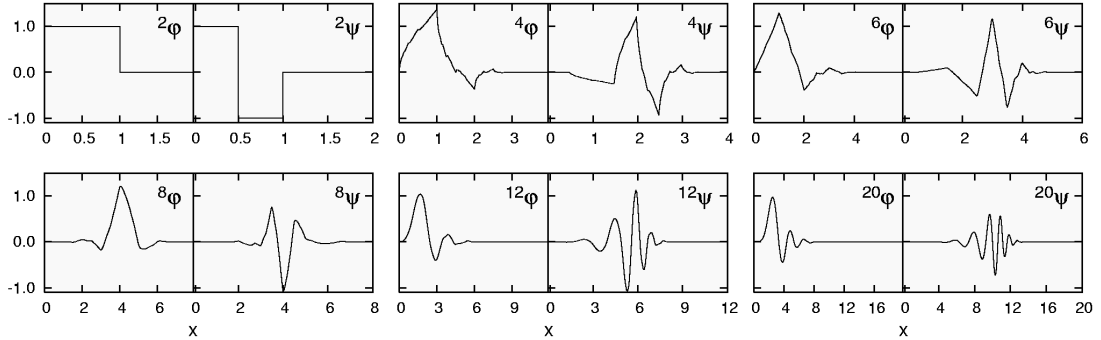
$$\int_{-\infty}^{\infty} dx \varphi(x) \varphi(x - k) = \delta_{k,0}. \quad (2.65)$$

Again, substituting the dilation equation and integrating this yields the condition in terms of the filter coefficients

$$\frac{1}{2} \sum_{n \in \mathbb{Z}} h_n h_{n-2k} = \delta_{k,0}. \quad (2.66)$$

The next requirement, that the wavelets have compact support, means that the wavelets and scaling functions are non-zero only in a finite interval. The length of this interval depends on the number of non-zero coefficients  $h_n$ , as can be seen again from the dilation equation. In this way we can construct orthonormal wavelets with compact support. For example, a compactly supported wavelet with genus  $2M$  is determined by  $2M$  non-zero filter coefficients  $\{h_n\}_{n=0}^{2M-1}$  and is non-zero on the interval  $[0, 2M]$  only. The corresponding wavelet and scaling function will be called the Daubechies  $2M$  wavelet and scaling function, i.e.,  ${}^{2M}\psi$  and  ${}^{2M}\varphi$  respectively.

Thus, if we want to construct such compactly supported orthonormal wavelets and scaling functions, we have Eq. (2.64) and Eq. (2.66) for  $k = 0..M-1$ . These are  $M+1$



**Figure 2.2:** Daubechies scaling functions and their corresponding wavelets for various filter lengths. The Daubechies 2 wavelet at the top is also known as the Haar wavelet.

equations with  $2M$  unknowns, leaving  $M - 1$  degrees of freedom. For  $M = 1$  this means that the wavelet is completely fixed. This wavelet is also known as the Haar wavelet and is shown at the top of Figure 2.2. For higher values of  $M$ , Daubechies used these degrees of freedom to make the wavelet as smooth as possible by setting the first  $M - 1$  moments of the wavelet equal to zero

$$\int_{-\infty}^{\infty} dx x^m \psi(x) = 0, \quad m = 1..M - 1. \quad (2.67)$$

In terms of the filter coefficients this requirement is given by

$$\sum_{n \in \mathbb{Z}} (-1)^n n^m h_{2M-1-n} = 0, \quad m = 1..M - 1. \quad (2.68)$$

Together with the first two requirements we now have  $2M$  equations and just as many unknowns. E.g., in case of the Daubechies 6 wavelets the following set of equations is obtained

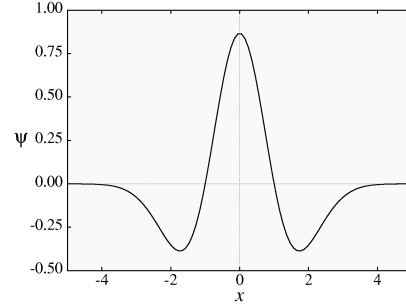
$$\begin{aligned} h_0 + h_1 + h_2 + h_3 + h_4 + h_5 &= 2 \\ h_0^2 + h_1^2 + h_2^2 + h_3^2 + h_4^2 + h_5^2 &= 2 \\ h_0 h_2 + h_1 h_3 + h_2 h_4 + h_3 h_5 &= 0 \\ h_0 h_4 + h_1 h_5 &= 0 \\ -h_1 + h_2 - h_3 + h_4 - h_5 &= 0 \\ -h_1 + 2h_2 - 3h_3 + 4h_4 - 5h_5 &= 0. \end{aligned} \quad (2.69)$$

By solving this set of quadratic equations the filter coefficients are obtained and as a result the scaling function and the wavelet are known too. The resulting filter coefficients for the first couple of Daubechies wavelets are given in Table 2.1. Plots of some Daubechies wavelets and scaling functions are given in Figure 2.2.

**Mexican hat wavelet** The next wavelet we consider is the Mexican hat wavelet. Mexican hat wavelets owe their name to their shape, which is shown in Figure 2.3. It is an example of a wavelet that is known in closed form, viz., the Mexican hat wavelet is the

M	1	2	3
$h_0$	1	$\frac{1}{4}(1+\sqrt{3})$	$\frac{1}{16}(1+\sqrt{10}+\sqrt{5+2\sqrt{10}})$
$h_1$	1	$\frac{1}{4}(3+\sqrt{3})$	$\frac{1}{16}(5+\sqrt{10}+3\sqrt{5+2\sqrt{10}})$
$h_2$	0	$\frac{1}{4}(3-\sqrt{3})$	$\frac{1}{16}(10-2\sqrt{10}+2\sqrt{5+2\sqrt{10}})$
$h_3$	0	$\frac{1}{4}(1-\sqrt{3})$	$\frac{1}{16}(10-2\sqrt{10}-2\sqrt{5+2\sqrt{10}})$
$h_4$	0	0	$\frac{1}{16}(5+\sqrt{10}-3\sqrt{5+2\sqrt{10}})$
$h_5$	0	0	$\frac{1}{16}(1+\sqrt{10}-\sqrt{5+2\sqrt{10}})$

**Table 2.1:** The filter coefficients for the 3 lowest order Daubechies wavelets.



**Figure 2.3:** The Mexican hat wavelet, owing its name to its shape.

second derivative of a Gaussian. Normalized to unity in the  $L^2$ -norm this yields

$$\psi(x) = \frac{2}{\sqrt{3}}\pi^{-1/4}(1-x^2)e^{-x^2/2}. \quad (2.70)$$

Having a closed form has the advantage that many operators on the wavelets can be calculated analytically, what is very cost effective. However, these wavelets do not have finite support. Related to that, in terms of the dilation equation such Mexican hat wavelets result in infinitely many non-zero filter coefficients for a Gaussian as scaling function. Furthermore, the Mexican hat wavelets and scaling functions do not form an orthonormal set.

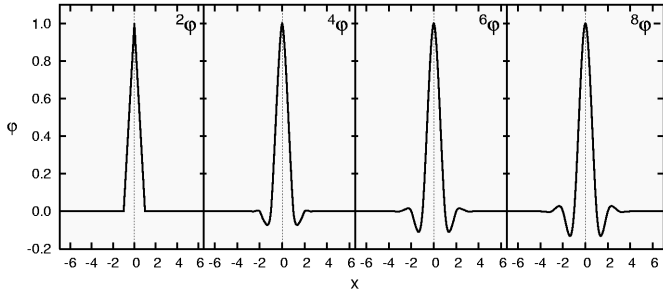
The reason that these Mexican hat wavelets are considered here is that they were one of the first wavelets to be used in electronic structure calculations [Cho93].

**Interpolets** The last type of wavelets that will be considered are the so-called interpolating wavelets or interpolets for short. The interpolating wavelets described here are based on the symmetric iterative interpolation process as introduced by Deslaurier and Dubuc [Des89]. This process uses two positive integral parameter:  $b$  (a base) and  $2M$  (an even number of moving points). A function  $g$  that is defined on the integers is extended to all integral multiples of  $1/b$ : if  $r$  is an integer between 0 and  $b$  and if  $n$  is an integer,  $g(n+r/b)$  is defined as the value  $p(n+r/b)$  where  $p$  is a Lagrange polynomial of degree smaller than  $2M$  such that  $p(k) = g(k)$  for every  $k \in [n-M+1, n+M]$ . By iterating this construction an extension  $g(t)$  is found for any rational number  $t$  whose denominator is an integral power of  $b$ . This resulting  $g(t)$  is a uniformly continuous extension on any finite interval whatever the base  $b$ , the number of moving points  $2M$  and the initial values  $g(n)$  and hence it has a unique continuous extension to the reals. By interpolating the Kronecker sequence at the integers using a base  $b = 2$  the interpolets are constructed

$$\varphi(x) = \sum_k \varphi\left(\frac{k}{2}\right)\varphi(2x-k). \quad (2.71)$$

Such interpolets were constructed by Donoho [Don92].

There is a close connection between the Daubechies wavelets and these interpolets, viz. the interpolating scaling function  $\varphi$  is the autocorrelation function of the Daubechies



2M	2	4	6	8
$h_0$	1	1	1	1
$h_1$	$\frac{1}{2}$	$\frac{9}{16}$	$\frac{75}{128}$	$\frac{1225}{2048}$
$h_2$	0	0	0	0
$h_3$	0	$-\frac{1}{16}$	$-\frac{25}{256}$	$-\frac{245}{2048}$
$h_4$	0	0	0	0
$h_5$	0	0	$\frac{3}{256}$	$\frac{49}{2048}$
$h_6$	0	0	0	0
$h_7$	0	0	0	$-\frac{5}{2048}$

**Figure 2.4:** Various interpolants, differing in the order  $2M$  of the polynomial span. **Table 2.2:** The filter coefficients for the four lowest order interpolants.

scaling function

$$\varphi(x) = \int dy \varphi^{\text{Daub}}(y) \varphi^{\text{Daub}}(x+y). \quad (2.72)$$

**Derivation** The interpolants can be derived in various ways. The first method is from their desired properties. Interpolants are functions having the following properties

1. **Dilation equation:**  $\varphi(x/2) = \sum_{k \in \mathbb{Z}} h_k \varphi(x-k)$
2. **Interpolating / Cardinality:**  $\varphi(k) = \delta_{k,0}, \forall k \in \mathbb{Z}$
3. **Symmetry:**  $\varphi(-x) = \varphi(x)$
4. **Compact support:**  $\varphi(x) = 0$ , for  $|x| > 2M-1$
5. **Polynomial span:** Any polynomial of degree  $2M-1$  can be represented exactly.

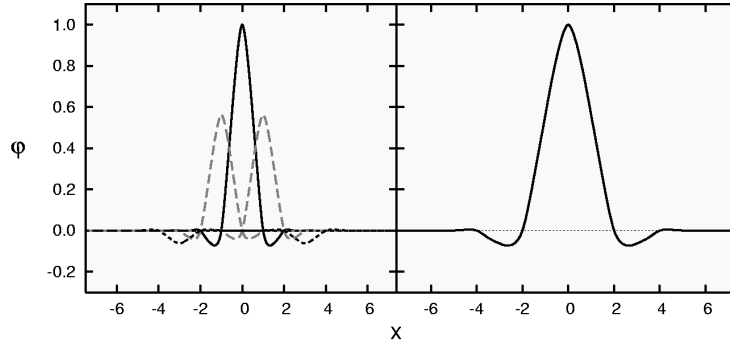
Just like for the Daubechies wavelets, the support length of the interpolants is related with the number of non-zero filter coefficients in the dilation equation. The freedom in choosing these filter coefficients is used for the cardinality and symmetry and subsequently to make them as smooth as possible, resulting in the polynomial span.

In terms of the filter coefficients the requirements yield  $h_{2k} = \delta_{k,0}$  for cardinality,  $h_{-k} = h_k$  for symmetry, and  $\sum_k k^m h_k = \delta_{m,0}$  for the polynomial span. The remaining filter coefficients  $h_k$  can thus be calculated by solving

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 3^2 & \cdots & (2M-1)^2 \\ 1 & 3^4 & \cdots & (2M-1)^4 \\ & & \ddots & \\ 1 & 3^{2M-2} & \cdots & (2M-1)^{2M-2} \end{bmatrix} \begin{bmatrix} h_1 \\ h_3 \\ \vdots \\ h_{2M-1} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (2.73)$$

for the appropriate number of non-zero filter coefficients, i.e., the number of moving points  $2M$ . For  $2M = 2, 4, 6$  and  $8$ , this yields the numbers shown in Table 2.2.

The interpolants  $2^M \varphi$  corresponding to these filter coefficients have been plotted in Figure 2.4. The  $2\varphi$  interpolant is also known as the hat function or linear B-spline.



**Figure 2.5:** Construction of an interpolant using the dilation equation. One interpolant, with  $2M = 4$ , is the weighted sum of 5 interpolants that are twice as narrow.

A second way to calculate the filter coefficients is via the Lagrange polynomials which are used in the interpolation process. For  $-n \in [-M+1, M]$  and  $r$  equal to 0 or 1, then

$$\varphi(n + \frac{r}{2}) = L_{-n}(\frac{r}{2}), \quad (2.74)$$

where  $L_{-n}$  denote the Lagrange polynomials and the filter coefficients are given via  $h_k = \varphi(k/2)$ .

A third way to obtain the same values is via the autocorrelation function. Because of Eq. (2.71) the filter coefficients can be obtained via

$$h_k = \sum_{n=-M+1}^{M-k} h_n^{\text{Daub}} h_{k+n}^{\text{Daub}}, \text{ for } k = 1, \dots, 2M-1 \quad (2.75)$$

from the Daubechies scaling filter coefficients.

So far, the interpolating scaling function has been discussed. In resemblance to the orthogonal wavelet transforms, the interpolant transform represents a function using dilations and translations of scaling functions and wavelets (Eq. (2.57)). However, the interpolating wavelets, interpolants for short, are non-orthogonal and the scaling coefficients are obtained from linear combinations of samples rather than from integrals, i.e.,  $s_k^0 = f(k)$ .

A striking difference with the Daubechies wavelets is that the filter coefficient for the wavelets  $\psi(x)$  are all zero except for  $g_0$ . Because of this the wavelets are equal to the scaling functions. And thus the interpolant transform results in a true combination of dilations and translations of one single mother function. Figure 2.5 shows how the interpolant at one level is the weighted sum of interpolants at one level lower.

In Section 2.4 the advantages and disadvantages of the different wavelets, such as accuracy and computational cost, are compared in actual electronic structure calculations. But first we focus in the following section on algorithms needed to use wavelets as a basis set for calculations.



## 2.3 Wavelet algorithms

The first two algorithms needed to use wavelets as a basis set for calculations are concerned with the transition between real space and scaling space, i.e., respectively the projection of a function onto the scaling space and the reconstruction in real space. This is followed by the algorithm forming the basis of the use of wavelets: the fast wavelet transform. This algorithm, and its inverse, transform respectively from the scaling space to the wavelet space and back and are linear in time with respect to the number of wavelets to be used. The actual calculations will be performed in this wavelet space. For this purpose algorithms are discussed for performing local operations, like addition or multiplication of two functions, and non-local operations, like differentiation, on the wavelet coefficients. Also it is shown how integration is performed by summing over (part of) the wavelet coefficients. And last but not least it is shown how the wavelet basis can be pruned without invalidating any of the previous algorithms.

### 2.3.1 Projection and reconstruction

**Projection** The first step is the projection from real space to scaling space  $\mathcal{V}_0$ . This corresponds to calculating the expansion coefficients  $s_k^0$  in Eq. (2.57), i.e.,

$$P_0 f(x) = \sum_{k=1}^N s_k^0 \phi_k^0(x). \quad (2.76)$$

This step differs for orthogonal wavelets and for interpolating wavelets. For interpolating wavelets the scaling coefficients are obtained from samples whereas for orthonormal wavelets from inner products. Because of the cardinality property of the interpolating wavelets, the scaling coefficient  $s_k^0$  is equal to the functional value at the position where the interpolet  $\phi_k^0$  is centered, i.e.,  $s_k^0 = f(k)$ . For orthogonal wavelets on the other hand, the scaling coefficients are obtained using inner products

$$s_k^0 = \int dx \phi_k^0(x) f(x). \quad (2.77)$$

However, if  $f(x)$  is not known as a continuous function but only at some discrete grid points, several methods exist to approximate the scaling coefficients. Four such methods are discussed, which will be called the direct method, the sum method, the moments method, and the inverse method respectively. At the end, the accuracy and the computational cost of the four methods are compared.

**Direct method** The simplest method is what we will call the direct method. Using this method, the values of the scaling coefficients equal the functional values of the function  $f$  at the corresponding grid point.

$$s_k^0 = f(k). \quad (2.78)$$

This is the same method as used for interpolating wavelets. However, for wavelets that do not have the cardinality property this method is usually a rather rough approximation.

**Sum method** The scaling coefficients  $s_k^0$  are given by the integral in Eq. (2.77). A well-known method of approximating an integral is to replace it by a sum. The scaling coefficients are thus approximated by the following summation

$$s_k^0 = \sum_{l=-\infty}^{\infty} \phi_k^0(l) f(l). \quad (2.79)$$

In case of wavelets with compact support the infinite sum will reduce to a finite sum with  $2M - 1$  terms, where  $2M$  is the support length of the wavelet, still resulting in a computationally efficient algorithm which is linear in the number of grid points  $N$  and in the support length of the wavelets  $M$ ,  $O(MN)$ . Moreover, it will give a better approximation than the previous (direct) method.

**Moments method** The third method uses the moments of the scaling function

$$m_k = \int_{-\infty}^{\infty} dx x^k \phi_k^0(x). \quad (2.80)$$

A quadrature matrix  $C$  is constructed using these moments in such a way that the scaling coefficients for polynomials  $p(x)$  up to order  $M - 1$  can be calculated exactly [Jam93]. Thus,

$$\int_{-\infty}^{\infty} dx p(x) \phi_k^0(x) = \sum_{l=0}^{M-1} c_l p(l+k), \quad (2.81)$$

where the coefficients  $\{c_l\}_{l=0}^{M-1}$  can be found by solving the following system of linear equations for the coefficients  $c_l$

$$\int_{-\infty}^{\infty} dx x^m \phi(x) = \sum_{l=0}^{M-1} l^m c_l, \quad (2.82)$$

for  $m = 0 \dots M - 1$ . Using these coefficients  $\{c_l\}_{l=0}^{M-1}$ , the scaling coefficients are hence approximated by

$$s_k^0 = \sum_{l=0}^{M-1} c_l f(l+k). \quad (2.83)$$

The number of terms in this sum is again proportional to the support length of the scaling functions (and wavelets) used. The transform scales also linear with the number of grid points, thus the order is again  $O(NM)$  like for the sum method.

**Inverse method** The fourth method is the most accurate for most wavelets, with the drawback of also being the computationally most expensive one. Its derivation uses Fourier-space methods [Dau92b]. Given the values of the function  $f$  on the grid one can write the Fourier series

$$\begin{aligned} \sum_{n=-\infty}^{\infty} f(n) e^{-inq} &= \sum_{n=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} s_l^0 \phi(n-l) e^{-inq} \\ &= \left( \sum_{l=-\infty}^{\infty} s_l^0 e^{-ilq} \right) \left( \sum_{m=-\infty}^{\infty} \phi(m) e^{-imq} \right). \end{aligned} \quad (2.84)$$

The  $\{s_l^0\}$  are thus the Fourier coefficients

$$s_l^0 = \frac{1}{2\pi} \int_0^{2\pi} dq \left( \sum_{n=-\infty}^{\infty} f(n) e^{-inq} \right) \left( \sum_{m=-\infty}^{\infty} \phi(m) e^{-imq} \right)^{-1}. \quad (2.85)$$

In this method the scaling coefficients are thus approximated by

$$s_k^0 = \sum_{n=-\infty}^{\infty} a_{k-n} f(n), \quad (2.86)$$

where

$$a_v = \frac{1}{2\pi} \int_0^{2\pi} dq \frac{e^{ivq}}{\sum_{m=-\infty}^{\infty} \phi(m) e^{-imq}}. \quad (2.87)$$

The difficulty now boils down to evaluating the coefficients  $a_v$ . These coefficients will typically decay as a function of  $v$ , though rather slow. The computation of the scaling coefficients thus becomes more expensive, but by using a certain threshold it remains linear in the number of grid points.

**Reconstruction** The reconstruction of the function from the scaling coefficients is straightforward. The function  $\tilde{f}$  is given by a linear combination of the scaling functions, i.e.,  $\tilde{f}(x) = \sum s_k^0 \phi_k^0(x)$ . As a result, the function values on the grid  $\tilde{f}(n)$  are given by

$$\tilde{f}(n) = \sum_{k=-\infty}^{\infty} s_k^0 \phi(n-k), \quad (2.88)$$

where we only need to know the functional value of the scaling functions on the grid points, i.e., on the integer values. For orthonormal wavelets the number of terms  $2M-1$  in the summation is thus once again related to the support length of the wavelets. Because of the cardinality of the interpolets the summation has unit length for these wavelets, i.e., the functional value is equal to the scaling coefficient of the scaling function centered at that point.

**Comparison** Which method to choose for depends on the type of wavelets and the desired accuracy. Because of the cardinality of the interpolets the reconstruction simplifies to an identity operation. Therefore, the inverse operation, i.e. the projection, should also be an identity operation, making the direct method the most appropriate for this kind of wavelets.

For the orthonormal Daubechies wavelets a comparison between the four different methods is made by means of a small numerical example for periodized wavelets. In this example we will expand two functions in terms of the Daubechies 6 scaling functions  $\phi_k^0(x)$ . The first function considered is one period of a sine, i.e.,  $f(x) = \sin(x)$  on the interval  $0 \leq x < 2\pi$ . The second function is a Gaussian,  $g(x) = \exp(-x^2)$ , on the interval  $-\sqrt{5/2} \leq x < \sqrt{5/2}$ . Contrary to  $f$ , the function  $g$  is not periodic. However, since the function is almost zero near both borders it can be extended quasi-periodic.

We will consider these functions on an equally spaced grid with 32 grid points on their respective intervals. The scale  $j=0$  is chosen such that at each grid point one scaling

	$\epsilon_f$	$\epsilon_g$	$n$
Direct method	1.13e-1	6.34e-2	1
Moments method	5.96e-4	8.04e-4	3
Sum method	1.60e-4	4.80e-4	4
Inverse method	7.73e-10	4.07e-10	15

**Table 2.3:** The errors as defined in Eq. (2.89), which show the difference in accuracy of the four different methods for two different functions compared to the computational cost of each of the methods.

function is centered. Using all four methods the scaling coefficients corresponding to these scaling functions are calculated, resulting in 32 scaling coefficients each. How accurate the various approximation methods work can be seen by reconstructing the function  $\tilde{f}$  from these scaling coefficients according to Eq. (2.88) and subtracting the result from the original function  $f$ .

A way to compare the accuracy is by defining the error as the average distance in the  $L^2$ -norm between the reconstructed function and the original function, i.e.,

$$\epsilon_f = \sqrt{\frac{1}{N} \sum_k |\tilde{f}(k) - f(k)|^2}. \quad (2.89)$$

The resulting errors are tabulated in Table 2.3. The cost of the reconstruction step is a summation of four terms per grid point, independent of the projection method. The costs of the various projection methods differ. These costs are given in the last column of the same table as the number of terms ( $n$ ) in the summation.

### 2.3.2 Fast wavelet transform

**Orthonormal wavelets** Using the projection as described in the previous section, the scaling coefficients at the finest level, i.e. level 0, were calculated. This scaling space can be decomposed into two parts,  $\mathcal{V}_1$  and  $\mathcal{W}_1$  such that  $\mathcal{V}_0 = \mathcal{V}_1 \oplus \mathcal{W}_1$ . As a result, the projection can also be written as the sum of a projection on  $\mathcal{V}_1$  and a projection on  $\mathcal{W}_1$

$$P_1 f(x) = \sum_k s_k^1 \phi_k^1(x) \quad \text{and} \quad Q_1 f(x) = \sum_k d_k^1 \psi_k^1(x). \quad (2.90)$$

Using the dilation equation, the coefficients  $s_k^1$  and  $d_k^1$  can be calculated from the coefficients  $s_k^0$  by

$$s_k^j = \sqrt{2} \sum_{l=0}^{2M-1} h_l s_{2k+l}^{j-1} \quad \text{and} \quad d_k^j = \sqrt{2} \sum_{l=0}^{2M-1} g_l s_{2k+l}^{j-1}. \quad (2.91)$$

In this way, out of the  $N$  values  $s_k^0$  on the finest scale,  $\frac{N}{2}$  scaling coefficients  $s_k^1$  and  $\frac{N}{2}$  wavelet coefficients  $d_k^1$  can be calculated. In case of a periodized wavelet with four filter

coefficients this can be depicted by the following matrix multiplication:

$$\begin{bmatrix} s_0^j \\ s_1^j \\ s_2^j \\ \vdots \\ s_{\frac{N}{2}-1}^j \\ d_0^j \\ d_1^j \\ \vdots \\ d_{\frac{N}{2}-1}^j \end{bmatrix} = \begin{bmatrix} g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 & \cdots & 0 & 0 & 0 \\ & & \vdots & & & & & \ddots & & & \vdots & \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \\ g_2 & g_3 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & g_0 & g_1 \\ h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 & \cdots & 0 & 0 & 0 \\ & & \vdots & & & & & \ddots & & & \vdots & \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & h_0 & h_1 \end{bmatrix} \begin{bmatrix} s_0^{j-1} \\ s_1^{j-1} \\ s_2^{j-1} \\ \vdots \\ s_{\frac{N}{2}-2}^{j-1} \\ s_{\frac{N}{2}-1}^{j-1} \end{bmatrix} \quad (2.92)$$

Now we introduce the following notation for the wavelet coefficients  $s_k^j$  and  $d_k^j$ . The  $N$  coefficients  $s_k^0$  are stored in a vector  $\mathbf{d}^0 (i : 0 \leq i < N)$ . After one step of the wavelet transform the  $\frac{N}{2}$  scaling coefficients  $s_k^1$  are stored together with the  $\frac{N}{2}$  wavelet coefficients  $d_k^1$  in the vector  $\mathbf{d}^1$ , where the scaling coefficients constitute the first half  $\mathbf{d}^1 (i : 0 \leq i < \frac{N}{2})$  and the wavelet coefficients the second half  $\mathbf{d}^1 (i : \frac{N}{2} \leq i < N)$ . Using this notation the above matrix multiplication can be written as  $\mathbf{d}^1 = \mathbf{T}^{(1)} \mathbf{d}^0$ .

A second step can be performed by decomposing  $\mathcal{V}_1$  into  $\mathcal{V}_2$  and  $\mathcal{W}_2$  in the same way, where  $\frac{N}{4}$  scaling coefficients  $s_k^2$  and  $\frac{N}{4}$  wavelet coefficients  $d_k^2$  are calculated from the  $\frac{N}{2}$  scaling coefficients  $s_k^1$ .

By introducing the vector  $\mathbf{d}^j$  in which the coefficients  $s_k^j, d_k^j, \dots, d_k^2$ , and  $d_k^1$  are stored in this order, the coefficients  $s_k^2$  and  $d_k^2$  can be calculated using the matrix multiplication  $\mathbf{d}^2 = \mathbf{T}^{(2)} \mathbf{d}^1$ . Only the scaling coefficients, which form the first half of the vector  $\mathbf{d}^1$ , have to be transformed whereas the wavelet coefficients in the second half should remain untouched. The matrix  $\mathbf{T}^{(2)}$  is thus related to matrix  $\mathbf{T}^{(1)}$  by

$$\mathbf{T}_n^{(j+1)} = \begin{bmatrix} \mathbf{T}_{\frac{n}{2}}^{(j)} & \mathbf{0}_{\frac{n}{2}} \\ \mathbf{0}_{\frac{n}{2}} & \mathbf{I}_{\frac{n}{2}} \end{bmatrix}. \quad (2.93)$$

Correspondingly,  $\mathcal{V}_2$  can be decomposed further by repeatedly using

$$\mathbf{d}^j = \mathbf{T}^{(j)} \mathbf{d}^{j-1}. \quad (2.94)$$

When the number of grid points  $N$  equals  $2^J$  this can be repeated  $J$  times until only one coefficient  $s_0^J$  is left, resulting in

$$\mathcal{V}_0 = \mathcal{V}_J \oplus \mathcal{W}_J \oplus \mathcal{W}_{J-1} \oplus \dots \oplus \mathcal{W}_1. \quad (2.95)$$

The total wavelet transform is the combination of all  $J$  steps, and in its matrix notation  $\mathbf{T}$  it is the product of the matrices for the separate steps

$$\mathbf{T} = \mathbf{T}^{(J)} \dots \mathbf{T}^{(2)} \mathbf{T}^{(1)}, \quad (2.96)$$

resulting in the complete transformation

$$\mathbf{d}^J = \mathbf{T} \mathbf{d}^0. \quad (2.97)$$

In algorithmic form the total wavelet transform can be written as

```

for  $j = 0$  to  $J - 1$  do
   $n = N/2^j$ 
   $\mathbf{d}(i : 0 \leq i < n) = \mathbf{T}_n^{(1)} \mathbf{d}(i : 0 \leq i < n)$ 
end for

```

Note that the transform can continue till only one scaling coefficient is left at the highest level. However, the transform can be stopped at any intermediate level as well.

The cost of one step of the wavelet transform is  $2Mn$  multiply-adds, where  $n$  is the length of the vector. In the first step the length of the vector is equal to  $N$  and in all consecutive steps this length halves. The total cost of the wavelet transform is thus,  $2M \sum_{j=0}^{J-1} 2^{-j} N$  what is bounded by  $4MN$ . The transform is thus linear in the number of grid points, accounting for the name *fast wavelet transform*. (For comparison, the fast Fourier transform is order  $N \log N$ ).

The *inverse wavelet transform* can reconstruct all scaling coefficients at the finest level  $s_k^0$  from the wavelet coefficients. For the orthonormal wavelets, this inverse wavelet transform is equal to the transpose of the wavelet transform. In formula form this yields for the coefficients

$$s_{2n}^{j-1} = \sum_{k=1}^M h_{2k} s_{n-k}^j + \sum_{k=1}^M g_{2k} d_{n-k}^j \quad (2.98)$$

and

$$s_{2n-1}^{j-1} = \sum_{k=1}^M h_{2k-1} s_{n-k}^j + \sum_{k=1}^M g_{2k-1} d_{n-k}^j. \quad (2.99)$$

**Interpolating wavelets** For interpolets the situation is very much alike. However, the absence of orthonormality implies that Eq. (2.95) does not hold any more. Still, the function can be written as

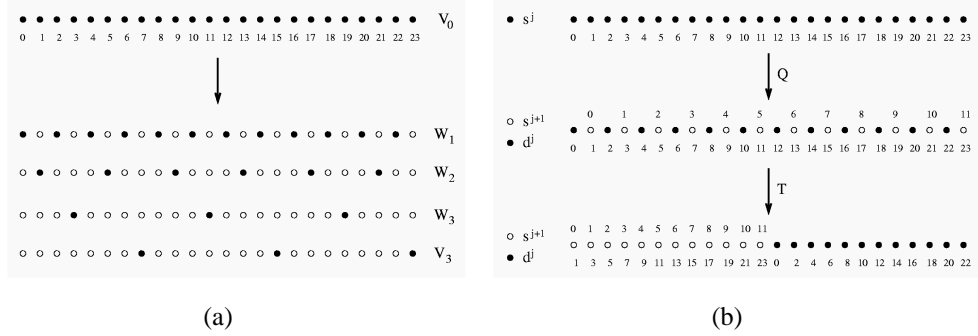
$$f(x) = \sum_k^J s_k^J \phi_k^J(x) + \sum_{j=1}^J \sum_k d_k^j \psi_k^j(x). \quad (2.100)$$

Because of the cardinality each coefficient is directly related to one grid point. Figure 2.6a shows how the grid points are assigned to a certain level. In this case the function  $f$  is known at 24 equally spaced grid points. At each grid point a scaling function  $\phi_k^0$  is centered, resulting in 24 scaling coefficients  $s_k^0$ . In every step  $j$  of the wavelet transform the even numbered points go to  $\mathcal{W}_j$  and the odd numbered points to  $\mathcal{V}_j$ .

The interpolet coefficients at level  $j$ , i.e.  $s_k^j$  and  $d_k^j$ , can be derived from the interpolet coefficients one level lower  $s_k^{j-1}$ , where the coefficients are given by

$$\begin{aligned} s_k^j &= s_{2k+1}^{j-1} \\ d_k^j &= s_{2k}^{j-1} - \sum_{l=-M}^{M-1} h_{2l+1} s_{2k+2l+1}^{j-1}. \end{aligned} \quad (2.101)$$

In vector notation, this interpolet transform can again be written in terms of matrix multiplications. But for interpolets it is easier to split one step of the transform in



**Figure 2.6:** (a) Decomposition of the grid in different levels. (b) One step of the interpolat transform is split in two parts, a computation part and a permutation part.

two parts, a calculational part  $Q$  and a permutation part  $P$ . One step of the interpolat transform is thus

$$\mathbf{T}^{(1)} = \mathbf{Q}\mathbf{P}. \quad (2.102)$$

This is demonstrated in Figure 2.6b. The calculational part is

$$\mathbf{Q}_N = \begin{bmatrix} \mathbf{I}_{\frac{N}{2}} & \mathbf{0}_{\frac{N}{2}} \\ -\mathbf{H}_{\frac{N}{2}} & \mathbf{I}_{\frac{N}{2}} \end{bmatrix} \quad (2.103)$$

where  $\mathbf{I}_n$  is the identity matrix of size  $n \times n$  and  $\mathbf{H}$  is a band diagonal matrix containing the coefficients  $h_l$  for odd  $l$

$$\mathbf{H} = \begin{bmatrix} h_1 & h_3 & 0 & \cdots & 0 & h_3 & h_1 \\ h_1 & h_1 & h_3 & 0 & \cdots & 0 & h_3 \\ h_3 & h_1 & h_1 & h_3 & 0 & \cdots & 0 \\ \vdots & & & \ddots & & & \\ 0 & \cdots & 0 & h_3 & h_1 & h_1 & h_3 \\ h_3 & 0 & \cdots & 0 & h_3 & h_1 & h_1 \end{bmatrix}. \quad (2.104)$$

The permutation matrix  $\mathbf{P}$  orders the coefficients in such a way that the scaling coefficients form the first half of the vector and the wavelet coefficients the second half.

The total interpolat transform  $\mathbf{T}$  is again the product of these matrices

$$\mathbf{T} = \mathbf{T}^{(J)} \cdots \mathbf{T}^{(2)} \mathbf{T}^{(1)}. \quad (2.105)$$

Given the interpolat transform, three other transforms are determined, viz. the inverse interpolat transform, the transpose interpolat transform and the inverse transpose interpolat transform. The latter two transforms were not discussed for the Daubechies wavelets because due to the orthonormality of these wavelets the transpose transform is equal to the inverse transform and as a result the inverse transpose transform is equal to the original transform.

One step of the inverse interpolat transform is of the form

$$(\mathbf{T}^{(1)})^{-1} = \mathbf{P}^{-1} \mathbf{Q}^{-1}. \quad (2.106)$$

$\mathbf{Q}^{-1}$  is of the same form as  $\mathbf{Q}$  but now with  $\mathbf{H}$  instead of  $-\mathbf{H}$ . The whole inverse interpolat transform consists of doing this for all levels, starting from the highest level

towards the lowest

$$\mathbf{T}^{-1} = \mathbf{T}^{(1)-1} \mathbf{T}^{(2)-1} \dots \mathbf{T}^{(J)-1}. \quad (2.107)$$

For interpolating wavelets the transpose transform is no longer equal to the inverse transform. The transpose interpolat transform is also interesting as we will see later. Whereas the interpolat uses neighboring values to update its own value, the transpose interpolat transform spreads its value to neighboring coefficients. This can be seen as follows

$$\mathbf{Q}_N^T = \begin{bmatrix} \mathbf{I}_{\frac{N}{2}} & -\mathbf{H}_{\frac{N}{2}}^T \\ \mathbf{0}_{\frac{N}{2}} & \mathbf{I}_{\frac{N}{2}} \end{bmatrix}, \quad (2.108)$$

since now coefficients from the lower half of the vector contribute to coefficients in the upper half. The whole transpose interpolat transform consists of doing this for all levels, starting from the highest level towards the lowest.

The set of transforms is completed by the inverse transpose interpolat transform

$$(\mathbf{T}^{(1)})^{-1T} = \mathbf{Q}^{-1T} \mathbf{P}, \quad (2.109)$$

since the transpose of  $\mathbf{P}$  is equal to its inverse. The whole inverse transpose interpolat transform consists of doing this for all levels, starting from the lowest level towards the highest.

Contrary to orthonormal wavelets, the filter for obtaining the scaling coefficients at a higher level has unit length. Because of this the values of the scaling coefficients do not change on going from one level to the next. This also implies that the wavelet coefficients in all levels can be calculated directly from the values of the scaling coefficients in  $\mathcal{V}_0$  with only  $4M - 2$  multiply-adds, thus simplifying parallelization.

### 2.3.3 Integration

The integral of a function over the whole space is an often used operation. The integration  $\int dx f(x)$  becomes in terms of wavelets

$$\sum_k s_k^J \int dx \phi_k^J(x) + \sum_{j=1}^J \sum_k d_k^j \int dx \psi_k^j(x). \quad (2.110)$$

The orthonormal Daubechies wavelets have the properties

$$\int dx \phi(x) = 1 \quad \text{and} \quad \int dx \psi(x) = 0, \quad (2.111)$$

reducing the integral of  $f$  to

$$2^J \sum_k s_k^J. \quad (2.112)$$

On the other hand, interpolating wavelets have the properties

$$\int dx \phi^J(x) = 2^J \quad \text{and} \quad \int dx \psi^j(x) = 2^{j-1}, \quad (2.113)$$

reducing the integral of  $f$  to

$$2^J \sum_k s_k^J + \sum_{j=1}^J 2^{j-1} \sum_k d_k^j. \quad (2.114)$$



### 2.3.4 Operators

**Local operators** Local operators only depend on the values of the functions at the point itself. Among the simplest such operators are addition and subtraction of two functions. These operators amount to simply adding or subtracting the wavelet coefficients, e.g.,

$$\begin{aligned} f(x) + g(x) &= \sum_k \left( {}^f s_k^J + {}^g s_k^J \right) \phi_k^J(x) + \sum_j \sum_k \left( {}^f d_k^j + {}^g d_k^j \right) \psi_k^j(x) \\ &= \sum_k {}^{f+g} s_k^J \phi_k^J(x) + \sum_j \sum_k {}^{f+g} d_k^j \psi_k^j(x). \end{aligned} \quad (2.115)$$

For most other local operations the functional values need to be known. These can be obtained easily for interpolants. Because of the cardinality of the interpolants the functional values are equal to the value of the scaling coefficient of the interpolant centered at those points. Thus, an inverse interpolant transform has to be performed. Subsequently the operation can be performed on the functional values, followed by the interpolant transform to obtain the result back to wavelet space.

For other wavelets these operations can also be performed in real space. However, an extra reconstruction and projection step have to be included. This is not a problem itself, but, as we will see later, this will make it impossible to prune the wavelet basis. To be able to work with a pruned wavelet basis for such wavelets, many more operators have to be dealt with in the same way as with the non-local operators.

**Non-local operators** Examples of non-local operators that we need are the overlap, the derivative and the second derivative. All these operators are handled in the same manner. Beylkin [Bey92] showed how these operators can be represented in the Daubechies wavelet basis, but this same method can also be used for the interpolants. The method will be demonstrated using the  $n$ -th derivative operator. For calculating the  $n$ -th derivative, we have to calculate the coefficients

$$L_k^0 = \langle \phi(x-k) | \frac{d^n}{dx^n} | \phi(x) \rangle = \int dx \phi(x-k) \frac{d^n}{dx^n} \phi(x). \quad (2.116)$$

By substituting the dilation equation one obtains the recursive relation

$$\begin{aligned} L_k^0 &= \int dx \phi(x-k) \frac{d^n}{dx^n} \phi(x) \\ &= \sum_l \sum_{l'} h_l h_{l'} \int dx \phi(2x-2k-l) \frac{d^n}{dx^n} \phi(2x-l') \\ &= 2^{n-1} \sum_l \sum_{l'} h_l h_{l'} \int dy \phi(y-2k-l) \frac{d^n}{dy^n} \phi(y-l') \\ &= 2^{n-1} \sum_l \sum_{l'} h_l h_{l'} L_{2k+l-l'}^0, \end{aligned} \quad (2.117)$$

and from the moments one obtains

$$\sum_k k^n L_k^0 = (-1)^n n!. \quad (2.118)$$

	I2	I4	I6	D6		I2	I4	I6
$L_0^0$	-2	$-\frac{20}{9}$	$-\frac{600888786024221}{253622166763700}$	$-\frac{295}{56}$	$L_1^+$	0	$-\frac{1}{8}$	$-\frac{24567315301808903}{140242913333655552}$
$L_1^0$	1	$\frac{9}{8}$	$\frac{18912324450745487}{15582545925961728}$	$\frac{356}{105}$	$L_3^+$	0	$\frac{1}{8}$	$\frac{45701663340631}{229155087146496}$
$L_2^0$	0	0	$\frac{840330704}{22825995008733}$	$-\frac{92}{105}$	$L_5^+$	0	0	$-\frac{413396361927385}{17530364166706944}$
$L_3^0$	0	$-\frac{1}{72}$	$-\frac{426200894323013}{11686909444471296}$	$\frac{4}{35}$	$L_7^+$	0	0	$-\frac{216039286320463}{280485826667311104}$
$L_4^0$	0	0	$\frac{4214962498727}{547823880209592}$	$\frac{3}{560}$	$L_9^+$	0	0	$\frac{24553237019}{261891528167424}$
$L_5^0$	0	0	$-\frac{1106268006449}{3682849614854400}$	0	$L_{11}^+$	0	0	$-\frac{104802013}{243477280093152}$
$L_6^0$	0	0	$-\frac{840330704}{22825995008733}$	0	$L_{13}^+$	0	0	$-\frac{9573}{81159093364384}$
$L_7^0$	0	0	$\frac{71096253079}{31165091851923456}$	0				
$L_8^0$	0	0	$-\frac{25528}{2536221667637}$	0				
$L_9^0$	0	0	$-\frac{28719}{10388363950641152}$	0				

**Table 2.4:** The second derivative coefficients for the three lowest order interpolants (I) and the Daubechies (D) 6 wavelets.

Because of the compact support of the wavelets, most of these coefficients are equal to zero. The number of elements that can be non-zero depends on the wavelet chosen. For interpolants the non-zero elements are  $L_{-4M+3}^0 \dots L_{4M-3}^0$  and for Daubechies wavelets  $L_{-2M+2}^0 \dots L_{2M-2}^0$ . Because of symmetry it also holds that  $L_{-k}^0 = (-1)^n L_k^0$ . Solving Eqs. (2.117) and (2.118) for the second derivative,  $n = 2$ , results for the three lowest order interpolants in the values shown at the left hand side in Table 2.4. For the two lowest order Daubechies wavelets no second derivative exists. For the Daubechies 6 wavelets the result is shown in the same table.

If we use only wavelets at one level, these coefficients  $L_k^0$  suffice. Namely, the matrix that represents the operator in  $\mathcal{V}_0$  is a band diagonal matrix  $\mathbf{L}^0$  with matrix elements  $l_{ij} = L_{i-j}^0$ . The action of the operator is calculated as  $\mathbf{p}^0 = \mathbf{L}^0 \mathbf{d}^0$ . However, when we have wavelets at two levels, coefficients for between the levels are needed as well, i.e.,

$$\alpha_k = \langle \psi(x-k) | \frac{d^n}{dx^n} | \psi(x) \rangle, \quad (2.119)$$

$$\beta_k = \langle \psi(x-k) | \frac{d^n}{dx^n} | \phi(x) \rangle, \text{ and} \quad (2.120)$$

$$\gamma_k = \langle \phi(x-k) | \frac{d^n}{dx^n} | \psi(x) \rangle. \quad (2.121)$$

Using the dilation equation these inter-level coefficients can be calculated from the intra-level coefficients  $L_k^0$  as

$$\alpha_k = \sum_l \sum_{l'} g_l g_{l'} L_{2k+l-l'}^0, \quad (2.122)$$

$$\beta_k = \sum_l \sum_{l'} g_l h_{l'} L_{2k+l-l'}^0, \text{ and} \quad (2.123)$$

$$\gamma_k = \sum_l \sum_{l'} h_l g_{l'} L_{2k+l-l'}^0. \quad (2.124)$$

Using these coefficients, the matrix for the  $n$ -th derivative becomes

$$\mathbf{L}_N^1 = \begin{bmatrix} 2^{1-n} \mathbf{L}_{\frac{N}{2}}^0 & \mathbf{C}_{\frac{N}{2}} \\ \mathbf{B}_{\frac{N}{2}} & \mathbf{A}_{\frac{N}{2}} \end{bmatrix}, \quad (2.125)$$

where the elements of the matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  are  $a_{i,j} = \alpha_{i-j}$ ,  $b_{i,j} = \beta_{i-j}$  and  $c_{i,j} = \gamma_{i-j}$  respectively.

For interpolets where  $\psi(x) = \phi(2x)$  this simplifies to the inter-level coefficients

$$L_k^+ = \langle \phi(x-k) | \frac{d^n}{dx^n} | \phi(2x) \rangle, \quad (2.126)$$

which can be calculated from the intra-level coefficients  $L_k^0$  as

$$L_k^+ = \sum_l h_l L_{k-l}^0 \quad (2.127)$$

and the matrix

$$\mathbf{L}_N^1 = \begin{bmatrix} 2^{1-n} \mathbf{L}_{\frac{N}{2}}^0 & \mathbf{L}_{\frac{N}{2}}^+ \\ \mathbf{L}_{\frac{N}{2}}^+ & \mathbf{L}_{\frac{N}{2}}^{even} \end{bmatrix}, \quad (2.128)$$

where  $L_{i,j}^{even} = L_{2(i-j)}^0$ . The resulting coefficients  $L_k^+$  for the second derivative are given in Table 2.4 as well for the lowest order interpolets, where  $L_{-k}^+ = L_k^+$ .

In case there are more levels, matrix coefficients for interactions between all different levels are needed. These can all be derived from the coefficients  $L_n^0$  in the same way using the dilation and the wavelet equation. However, another way to construct the matrix is by using

$$\begin{aligned} \langle \mathbf{d}^0 | \mathbf{L}^0 | \mathbf{d}^0 \rangle &= \langle \mathbf{T}^{-1} \mathbf{d}^J | \mathbf{L}^0 | \mathbf{T}^{-1} \mathbf{d}^J \rangle \\ &= \langle \mathbf{d}^J | \mathbf{T}^{-1T} \mathbf{L}^0 \mathbf{T}^{-1} | \mathbf{d}^J \rangle \\ &= \langle \mathbf{d}^J | \mathbf{L}^J | \mathbf{d}^J \rangle, \end{aligned} \quad (2.129)$$

yielding

$$\mathbf{L}^J = \mathbf{T}^{-1T} \mathbf{L}^0 \mathbf{T}^{-1}. \quad (2.130)$$

However, the more levels are used, the wider the wavelets become and as a result the more overlap with other wavelets and thus the larger the number of non-zero coefficients, resulting in a matrix for the operation that lost its sparsity, resulting in a quadratic scaling behavior.

An alternative way to calculate  $\mathbf{p}^J = \mathbf{L}^J \mathbf{d}^J$  can be constructed by splitting the wavelet transform in two parts

$$\mathbf{T} = \mathbf{R} \mathbf{T}^{(1)}, \quad (2.131)$$

where

$$\mathbf{R}_N = \mathbf{T}_N^{(J)} \cdots \mathbf{T}_N^{(2)} = \begin{bmatrix} \mathbf{T}_{\frac{N}{2}} & \mathbf{0}_{\frac{N}{2}} \\ \mathbf{0}_{\frac{N}{2}} & \mathbf{I}_{\frac{N}{2}} \end{bmatrix}, \quad (2.132)$$

Substituting this one obtains

$$\begin{aligned}
\mathbf{L}_N^J &= \mathbf{R}_N^{-1T} \mathbf{T}_N^{(1)-1T} \mathbf{L}_N^0 \mathbf{T}_N^{(1)-1} \mathbf{R}_N^{-1} \\
&= \begin{bmatrix} \mathbf{T}_{\frac{N}{2}}^{-1T} & \mathbf{0}_{\frac{N}{2}} \\ \mathbf{0}_{\frac{N}{2}} & \mathbf{I}_{\frac{N}{2}} \end{bmatrix} \begin{bmatrix} 2^{1-n} \mathbf{L}_{\frac{N}{2}}^0 & \mathbf{C}_{\frac{N}{2}} \\ \mathbf{B}_{\frac{N}{2}} & \mathbf{A}_{\frac{N}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{T}_{\frac{N}{2}}^{-1} & \mathbf{0}_{\frac{N}{2}} \\ \mathbf{0}_{\frac{N}{2}} & \mathbf{I}_{\frac{N}{2}} \end{bmatrix} \\
&= 2^{1-n} \begin{bmatrix} \mathbf{L}_{\frac{N}{2}}^{J-1} & \mathbf{0}_{\frac{N}{2}} \\ \mathbf{0}_{\frac{N}{2}} & \mathbf{0}_{\frac{N}{2}} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{\frac{N}{2}} & \mathbf{0}_{\frac{N}{2}} \\ \mathbf{0}_{\frac{N}{2}} & \mathbf{A}_{\frac{N}{2}} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{\frac{N}{2}} & \mathbf{T}_{\frac{N}{2}}^{-1T} \mathbf{C}_{\frac{N}{2}} \\ \mathbf{0}_{\frac{N}{2}} & \mathbf{0}_{\frac{N}{2}} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{\frac{N}{2}} & \mathbf{0}_{\frac{N}{2}} \\ \mathbf{B}_{\frac{N}{2}} \mathbf{T}_{\frac{N}{2}}^{-1} & \mathbf{0}_{\frac{N}{2}} \end{bmatrix}.
\end{aligned} \tag{2.133}$$

This equation can be used to create an alternative approach, which uses only interactions with neighboring levels combined with the wavelet transform, resulting in a linear scaling scheme. The whole matrix is split into four parts. The latter two are neighboring level interactions, viz., the fourth one is the towards lower interaction and the third one the towards higher interaction. The second one is the intra-level interaction for the finest wavelets, whereas the first part is the original problem but now for the first half of the wavelet coefficients. This can be used to construct the following algorithm for  $\mathbf{p}^J = \mathbf{L}^J \mathbf{d}^J$ :

```

pJ(i : 0 ≤ i < N/2J) = L0dJ(i : 0 ≤ i < N/2J)
for j = J to 1 do
  pJ(i : N/2j ≤ i < N/2j-1) = Adj(i : N/2j ≤ i < N/2j-1)
  pJ(i : N/2j ≤ i < N/2j-1) += Bdj(i : 0 ≤ i < N/2j)
  tj(i : 0 ≤ i < N/2j) = Cdj(i : N/2j ≤ i < N/2j-1)
  dj-1 = T(j)-1dj
end for
for j = 2 to J do
  tj += T(j)-1tj-1
end for
pJ += tJ

```

The special case for interpolants is obtained by substituting Eq. (2.128) instead of Eq. (2.125), i.e., substituting everywhere **B** and **C** by **L**<sup>+</sup> and substituting **A** by **L**<sup>even</sup>.

This new operator can be used directly to calculate the kinetic energy (e.g. Eq. (2.23))

$$\begin{aligned}
E_k &= \langle \mathbf{d}^0 | \mathbf{L}^0 | \mathbf{d}^0 \rangle \\
&= \langle \mathbf{d}^J | \mathbf{L}^J | \mathbf{d}^J \rangle.
\end{aligned} \tag{2.134}$$

But to use it to calculate the second derivative, an extra transform is needed. To calculate the result in wavelet space  $\mathcal{W}_1 \oplus \mathcal{W}_2 \oplus \mathcal{W}_3 \oplus \dots \oplus \mathcal{W}_J \oplus \mathcal{V}_J$  one has to calculate

$$\begin{aligned}
\mathbf{T} \mathbf{L}^0 | \mathbf{d}^0 \rangle &= \mathbf{T} \mathbf{L}^0 | \mathbf{T}^{-1} \mathbf{d}^J \rangle \\
&= \mathbf{T} \mathbf{L}^0 \mathbf{T}^{-1} | \mathbf{d}^J \rangle \\
&= \mathbf{T} \mathbf{T}^T \mathbf{L}^J | \mathbf{d}^J \rangle.
\end{aligned} \tag{2.135}$$

Thus, only for orthonormal wavelets, where  $\mathbf{T}^T = \mathbf{T}^{-1}$ ,  $\mathbf{L}^J | \mathbf{d}^J \rangle$  yields directly the result of the operator in wavelet space. For other wavelets, like the interpolants, the result first

2M	2	4	6	2	4	6	
$S_0^0$	$\frac{2}{3}$	56264 70245	67751983439382433 80022904342005702	$S_1^+$	$\frac{1}{2}$	239171 421470	1085441555078573078431 1843727716039811374080
$S_1^0$	$\frac{1}{6}$	19253 140490	10401555307753158173 87796557906657684480	$S_3^+$	0	-7487 107040	-10511478421687078501 102429317557767298560
$S_2^0$	0	-2827 70245	-10636019205366151 200057260855014255	$S_5^+$	0	16673 6743520	265604554287972939 184372771603981137408
$S_3^0$	0	6283 2247840	3885507748820613551 307287952673301895680	$S_7^+$	0	1 210735	-106114710840853271 216909143063507220480
$S_4^0$	0	-16 210735	-1423693615386019 1028865912968644740	$S_9^+$	0	0	-2186739499472653 136572423410356398080
$S_5^0$	0	-1 6743520	19214162612610613 184372771603981137408	$S_{11}^+$	0	0	-19610542057 800229043420057020
$S_6^0$	0	0	-179398920064 85738826080720395	$S_{13}^+$	0	0	448593 266743014473352340
$S_7^0$	0	0	2232277564897 24101015895945246720				
$S_8^0$	0	0	9569984 66685753618338085				
$S_9^0$	0	0	1345779 136572423410356398080				

2M	2	4	6	2	4	6	
$D_0^0$	0	0	0	$D_1^+$	$-\frac{1}{2}$	-41719 66528	-601842755183958511 919737224275161600
$D_1^0$	$-\frac{1}{2}$	-3659 5280	-21573383441499121 27870824978035200	$D_3^+$	0	1307 24640	2140389637385939 29198007119846400
$D_2^0$	0	731 6930	9557878635178 54435205035225	$D_5^+$	0	-4241 665280	-5754566124347879 367894889710064640
$D_3^0$	0	-481 73920	-701118554033057 2388927852601600	$D_7^+$	0	-1 41580	589497248444887 33444899736422400
$D_4^0$	0	4 10395	3212112203957 1026492437807100	$D_9^+$	0	0	128675668980953 3678948897100646400
$D_5^0$	0	1 665280	-666591477947 1967352351390720	$D_{11}^+$	0	0	12644177 62211662897400
$D_6^0$	0	0	2835533248 163305615105675	$D_{13}^+$	0	0	1349 48386848920200
$D_7^0$	0	0	478319328577 1226316299033548800				
$D_8^0$	0	0	43168 18145068345075				
$D_9^0$	0	0	4047 12387033323571200				

**Table 2.5:** The overlap ( $S$ ) and first derivative ( $D$ ) coefficients for the three lowest order interpolets.

has to be multiplied by  $\mathbf{T}\mathbf{T}^T$ .

Exactly in the same way the coefficients of the overlap matrix  $\mathbf{S}$  and the first derivative  $\mathbf{D}$  can be calculated, where  $n$  equals zero and unity respectively. These coefficients are shown in Table 2.5 for the lowest order interpolets. For the overlap coefficients holds  $S_{-k}^x = S_k^x$  and for the first derivative coefficients  $D_{-k}^x = -D_k^x$ .

### 2.3.5 Pruning

So far the number of wavelets used in an expansion is equal to the number of scaling functions used at the finest scale and thus equal to the number of grid points where the function  $f$  was known. However, we want the basis to be as small as possible to store the function efficiently and to be able to perform efficient calculations.

The idea of the wavelet transform is that the highest scale  $\mathcal{V}_J$  gives a rough approximation whereas the wavelet levels  $\mathcal{W}_j$  add more and more levels of detail. In regions where this detail is not needed the corresponding wavelet coefficients  $d_k^j$  will be (close to) zero. As a result these coefficients can be omitted, i.e., the wavelet expansion can be

truncated by eliminating elements of the basis. One is said to be working in a truncated basis when one works within the subspace formed by the remaining basis elements. Another word for this truncation of the expansion is pruning.

A criterion is needed to determine which elements could be eliminated. The most straightforward criterion is to determine a cut-off value. All elements smaller than this cut-off value should then be left out. This results in a compact representation of the function that can be used for example to compress images. This is used in the JPEG2000 wavelet compression (ISO 15444), which can compress images up to 200 times with no appreciable degradation in quality. By adding again the zeros and an inverse transform the original image is retained

However, when calculations have to be performed on the compressed data, the situation becomes more complex. The scale and translation of each wavelet left in the pruned basis are known, thus all the elements of the matrix representing an operator in this basis can be calculated. As shown in the previous subsection this results in a scheme that is no longer linear in the number of wavelets used.

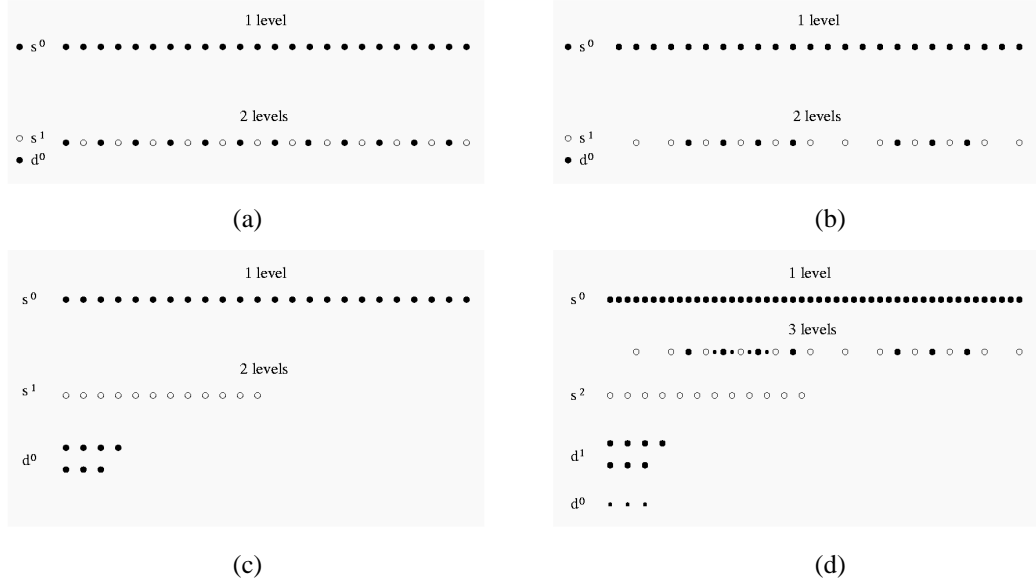
The alternative described in the same subsection uses a transform combined with only neighboring level interactions. However, if during this transform all elements that were eliminated have to be filled in again the advantage of pruning is gone. However, for interpolating wavelets this is not necessary if we lay some restrictions on the pruning.

Not all elements smaller than the cut-off can just be left out. Namely, it might be needed by some coefficients at lower levels to be able to do the inverse transform or some other operation. Because we need to know the neighboring coefficients to perform the transforms and to calculate the action of operators, it is convenient to regard regions of coefficients that we want to keep and other parts that can be neglected, instead of looking at individual coefficients, although the regions could in principle have unity width. These regions are also necessary to serve as a surrounding box if coefficients at a finer level are needed.

Thus, we get various levels, and in every level (small) arrays, containing part of the original array. This will be demonstrated first for 1D but can be extended to higher dimensions, what will be discussed later on. In Ref. [Lip98] this is called the good basis condition for synthesis and reconstruction.

The idea of pruning is shown in Figure 2.7. Part a shows first the coefficients needed if only one level is used. In case of this example  $N$  coefficients are needed. Subsequently, the coefficients needed after the wavelet transform, if two levels are used, is shown.  $N/2$  scaling coefficients are needed, the open circles, to give a rough approximation of the function. The detail is added using  $N/2$  wavelet coefficients, the filled circles. Part b of the figure shows that when the detail is not needed everywhere, wavelet coefficients can be left out, in this case leaving wavelet coefficients in two small regions.

If only one level is used, we thus need one large array to store the coefficients  $s_k^0$ . After the wavelet transform and omitting the unnecessary wavelet coefficients we have various levels, each containing one or more smaller arrays with coefficients at that level in some



**Figure 2.7:** (a) In one step of the wavelet transform  $N$  scaling coefficients are converted to  $N/2$  coefficients of wider scaling functions (open circles) and  $N/2$  wavelet coefficients (filled circles). (b) The wavelet coefficients can also be added only where necessary. (c) The coefficients are then stored in separate arrays. (d) Adding an extra level of detail somewhere boils down to adding just one extra array.

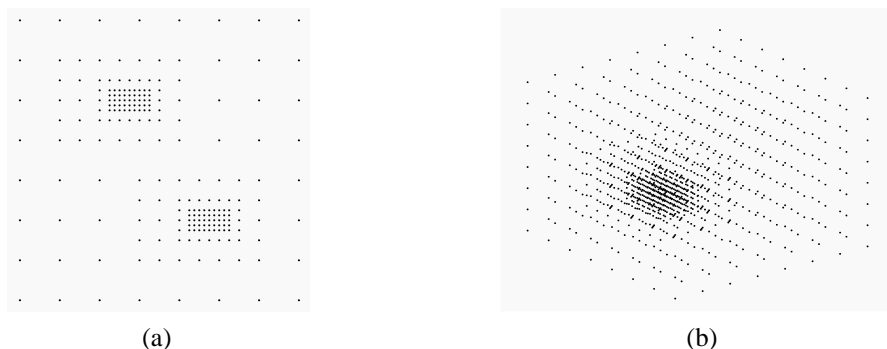
part of the space. This is shown in Figure 2.7c.

When more detail is needed in some region extra levels can be added. In the first representation, using only one level, this means that we have to take twice as narrow scaling functions, which would double the number of coefficients necessary. In the second representation this means adding an extra level and in this level only a small number of finer wavelets at the position where necessary. Thus only a small array is added as shown in Figure 2.7d.

Adding higher resolution in a limited region using an extra level thus adds only a few basis functions compared to doubling the number of basis functions when using only one level of finer wavelets. In higher dimensions the gain is even larger because in  $n$ -dimensional space the number of basis functions increases as  $2^n$ . An example of good bases for interpolants in 2D and 3D (with  $M=2$ ) are shown in Figure 2.8.

### 2.3.6 Higher dimensions

So far everything has been discussed for the one dimensional case. We are interested in such one dimensional calculations, but our main interest is in higher dimensional calculations, like 2D quantum dots and 3D molecular systems. For such calculations higher dimensional wavelets are needed. Such higher dimensional wavelets can be obtained in several ways. One way is to derive higher dimensional wavelets in a way analogous to the one dimensional wavelets we have seen above. Another way is to construct the



**Figure 2.8:** A good grid decomposition in (a) 2D and (b) 3D. In a good grid decomposition finer levels of detail are only possible if the coarser levels also exist around that position.

higher dimensional wavelets from the one dimensional wavelets by means of tensor products. In 3D, the wavelet can then be written as

$$\psi(x, y, z) = \psi(x)\psi(y)\psi(z). \quad (2.136)$$

The construction of such higher dimensional wavelets can be done in two ways; as a tensor product of 1D multiresolution analyses or as a tensor product of 1D wavelets. The main difference between these two methods is that in the former  $x$ ,  $y$  and  $z$  are dilated simultaneously, whereas in the latter  $x$ ,  $y$  and  $z$  are dilated separately. We have chosen to use the latter method, because it has the advantage that the higher dimensional wavelet transform can be performed by consecutive 1D transforms in the different directions. All algorithms discussed so far can then be used as well for these higher dimensional wavelets.

## 2.4 Example calculations

Before tackling three dimensional molecular systems and two dimensional structures called quantum dots, the wavelets will be applied first to some one dimensional case studies. These one dimensional structures are, although of physical interest, mainly artificial. However, because of their simplicity the one dimensional cases are illustrative in showing how our wavelets based electronic structure code works. The harmonic oscillator, the Poisson equation and the potential well are used as examples to explain the solution method, to show that the method yields accurate results, to compare the effects of different kinds of wavelets, to compare various preconditioners and to show the advantage of pruning.

The units that are employed throughout this chapter are atomic units. The unit of length is the Bohr radius  $a_0 (= 0.5292 \text{ \AA} = 5.292 \cdot 10^{-11} \text{ m})$ , the unit of charge is the charge of one electron  $e (= 1.602 \cdot 10^{-19} \text{ C})$ , the unit of mass is the mass of one electron  $m_e (= 9.109 \cdot 10^{-31} \text{ kg})$  and the unit of energy is the Hartree  $(= 27.211 \text{ eV} = 4.260 \cdot 10^{-18} \text{ J})$ .



### 2.4.1 Quantum harmonic oscillator

A well-known example of a one dimensional system is the quantum harmonic oscillator, which is one of the foundation problems of quantum mechanics. Because this problem is analytically solvable, it provides a good first test for the applicability of wavelets in electronic structure calculations.

**Analytical solution** The Schrödinger equation for the one dimensional harmonic oscillator is

$$\hat{H}\Psi(x) = \left( -\frac{1}{2} \frac{d^2}{dx^2} + V(x) \right) \Psi(x) = E\Psi(x), \quad (2.137)$$

where  $\hat{H}$  is the Hamiltonian,  $\Psi$  the wavefunction,  $E$  the energy of that wavefunction and  $V$  the potential. For the harmonic oscillator this potential is given by

$$V(x) = \frac{1}{2} \omega^2 x^2, \quad (2.138)$$

where  $\omega$  is the strength of the confinement. This eigenvalue problem can be solved analytically, resulting in the formula of the normalized wavefunctions

$$\Psi_n(x) = \left( \frac{\omega}{\pi} \right)^{1/4} \frac{1}{\sqrt{2^n n!}} H_n(\sqrt{\omega} x) e^{-\omega x^2/2}, \quad (2.139)$$

where  $H_n(\sqrt{\omega} x)$  is a Hermite polynomial. The corresponding energy eigenvalues, with degeneracy  $n$ , are

$$E_n = \left( n + \frac{1}{2} \right) \omega. \quad (2.140)$$

The ground-state wavefunction is thus  $\Psi_0(x) = (\omega/\pi)^{1/4} e^{-\omega x^2/2}$  with energy  $E_0 = \omega/2$ .

**Numerical solution** The problem is solved numerically by minimizing the total energy

$$E(\Psi) = \frac{\langle \Psi | \hat{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle}, \quad (2.141)$$

which is obtained by multiplying Eq. (2.137) by  $\Psi^*(x)$  and integrating over  $x$ .

The minimization can be performed by means of the steepest descent method. In this method, an arbitrary initial guess of the wavefunction  $\Psi^{(0)}$  is started from. To minimize the energy the direction in which the energy decreases the steepest is searched for. This direction is given by the local downhill gradient of the energy with respect to this wavefunction. The wavefunction is updated in this direction resulting in a better guess of the wavefunction. The new gradient is calculated, etc. This process is repeated till it converged to a minimum, at which the gradient is zero.

The gradient of the energy with respect to the wavefunction can be calculated straightforwardly as [Sti89]

$$\frac{\delta E(\Psi)}{\delta \Psi^*} = \hat{H}\Psi. \quad (2.142)$$

The wavefunction is updated in the direction of this gradient

$$\Psi^{(n+1)} = \Psi^{(n)} + \lambda^{(n)} g^{(n)}, \quad (2.143)$$

where the gradient is denoted by  $g^{(n)}$  and  $\lambda$  is the step size in the update direction. This

step size can be chosen freely. A simple method is to take a fixed (small) step size. However, probably this does not minimize the energy in the search direction and as a result the next search direction might be almost the same. Instead,  $\lambda$  could also be chosen such that the energy is minimized in the search direction, i.e., by means of a line minimization. This optimal value for  $\lambda^{(n)}$  can be derived by substituting Equation (2.143) in Equation (2.141)

$$E^{(n+1)} = \frac{\langle \Psi^{(n)} + \lambda^{(n)} g^{(n)} | \hat{H} | \Psi^{(n)} + \lambda^{(n)} g^{(n)} \rangle}{\langle \Psi^{(n)} + \lambda^{(n)} g^{(n)} | \Psi^{(n)} + \lambda^{(n)} g^{(n)} \rangle}. \quad (2.144)$$

Rewriting this equation gives

$$E^{(n+1)} = \frac{d + e\lambda + f\lambda^2}{a + b\lambda + c\lambda^2}, \quad (2.145)$$

where

$$\begin{aligned} a &= \langle \Psi^{(n)} | \Psi^{(n)} \rangle & d &= \langle \Psi^{(n)} | \hat{H} | \Psi^{(n)} \rangle \\ b &= \langle \Psi^{(n)} | g^{(n)} \rangle + \langle g^{(n)} | \Psi^{(n)} \rangle & e &= \langle \Psi^{(n)} | \hat{H} | g^{(n)} \rangle + \langle g^{(n)} | \hat{H} | \Psi^{(n)} \rangle \\ c &= \langle g^{(n)} | g^{(n)} \rangle & f &= \langle g^{(n)} | \hat{H} | g^{(n)} \rangle. \end{aligned} \quad (2.146)$$

Given these coefficients the value for  $\lambda$  minimizing the energy in Eq. (2.145) can be solved analytically. The optimal step size is given by

$$\lambda^{(n)} = \frac{dc - fa + \sqrt{d^2c^2 - 2dcfa + f^2a^2 - ecdb + e^2ca + fb^2d - fbea}}{fb - ec}. \quad (2.147)$$

To calculate the coefficients of Equation (2.146), eight integrals have to be evaluated. However, this number can be reduced to two since for properly normalized orbitals  $a$  equals unity,  $d$  is the energy for the old wavefunction  $\Psi^{(n)}$ ,  $b$  can also be derived from this old energy and the two contributions to  $e$  are equal, resulting in

$$\begin{aligned} a &= 1 & d &= E^{(n)} \\ b &= 2E^{(n)} & e &= 2c \\ c &= \langle g^{(n)} | g^{(n)} \rangle & f &= \langle g^{(n)} | \hat{H} | g^{(n)} \rangle. \end{aligned} \quad (2.148)$$

The equation for the optimal  $\lambda$  simplifies as a result to

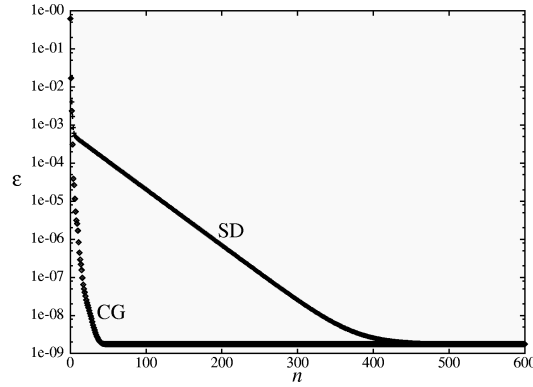
$$\lambda^{(n)} = \frac{cE - f + \sqrt{-3c^2E^2 - 6cEf + f^2 + 4c^3 + 4fE^3}}{2(fE - c^2)}. \quad (2.149)$$

To perform the calculations, a basis set is needed. The basis used is formed of wavelets. To start with, the wavefunction  $\Psi(x)$  is expanded in terms of one level of wavelets only

$$\Psi(x) = \sum_l s_l \phi_l^0(x). \quad (2.150)$$

For the implementation these wavelet coefficients are stored in a vector  $\mathbf{s}$ . Operator actions are then matrix multiplications with this vector. The above scheme is thus translated as follows: an initial guess  $\mathbf{s}^{(0)}$ , gradients  $\mathbf{g}^{(n)} = \mathbf{H}\mathbf{s}^{(n)}$ , coefficients  $c = \mathbf{g}^{(n)T} \mathbf{g}^{(n)}$  and  $f = \mathbf{g}^{(n)T} \mathbf{H} \mathbf{g}^{(n)}$ , and updated wavefunctions  $\mathbf{s}^{(n+1)} = \mathbf{s}^{(n)} + \lambda^{(n)} \mathbf{g}^{(n)}$ , where  $\mathbf{H}$  is a band diagonal matrix representing the Hamiltonian in this wavelet space.

The precision of the wavefunction can be expressed by the difference in energy of the calculated and the analytical solution. This error  $\varepsilon$  has been plotted in Figure 2.9 as a



**Figure 2.9:** The error  $\varepsilon$  in calculating the ground-state energy of the one dimensional harmonic oscillator as a function of the number of iterations  $n$  for interpolant  $M = 2$  and  $N=128$ , both using the steepest descent and the conjugate gradient method.

function of the number of iterations.

Minimizing in the direction of the gradient seems a natural way to work. However, it has been proved that it is much more efficient to regard also the search directions of previous steps. This is what is done in the conjugate gradient method [Pol71]. Here the search direction is  $h^{(n)}$  instead of  $g^{(n)}$  where this new search direction is related to the gradient via

$$h^{(n)} = \begin{cases} g^{(n)} & n = 0 \\ g^{(n)} + \gamma^{(n-1)}h^{(n-1)} & n > 0 \end{cases}, \quad (2.151)$$

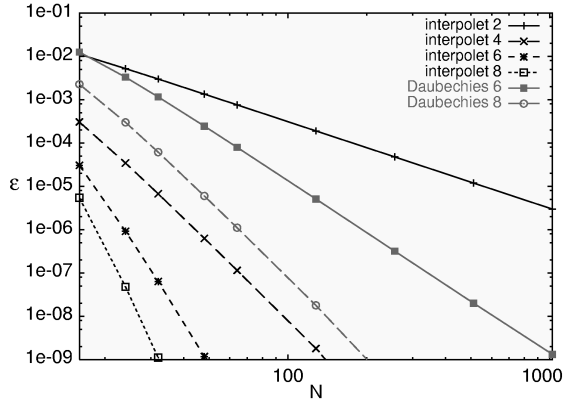
where

$$\gamma^{(n-1)} = \frac{\langle g^{(n)} | g^{(n)} \rangle}{\langle g^{(n-1)} | g^{(n-1)} \rangle}. \quad (2.152)$$

For a comparison of the convergence of conjugate gradients with the steepest descent method, the error as a function of the number of iterations for conjugate gradients is given in Figure 2.9 as well. As can be seen the convergence of the conjugate gradient method is superior to that of the steepest descent method.

**Comparison of various wavelets** Using the scheme described the ground-state energy of the harmonic oscillator has been calculated using various types of wavelets, viz., using the Daubechies 6 and 8 wavelets and using various interpolating wavelets. The properties of the wavelets and the number of wavelets determine the accuracy that is reached in the calculation. This accuracy can be quantified by  $\varepsilon$  as the difference between the numerical and the theoretical result for the ground-state energy. This error has been plotted in Figure 2.10 for the various wavelets as a function of the number of wavelets used. In these calculations the choice  $\omega = 1$  has been used, resulting in a theoretical energy of 0.5.

From this figure it is clear that the error decreases with the number of wavelets used for all types of wavelets. Since this figure is on a double logarithmic scale, the lines in the figure correspond to functions  $f(x) = cx^{-n}$ . Their respective powers  $n$  are given in



**Figure 2.10:** The error in calculating the ground-state energy of the one dimensional harmonic oscillator as a function of the number of wavelets used for various types of wavelets.

**Table 2.6:** Comparison of the slopes ( $n$ ) of the lines in Figure 2.10 with the number of zero moments (z.m.) of the corresponding wavelets.

wavelet	$n$	z.m.
Daubechies 6	4	2
Daubechies 8	6	3
interpolet 2	2	1
interpolet 4	6	3
interpolet 6	10	5
interpolet 8	14	7

Table 2.6. In the same table the number of zero moments of the corresponding wavelets are shown.

Thus except from decreasing with the number of wavelets used, the error also decreases with the order of the wavelets. As a result less wavelets of higher order are necessary to obtain the same error. However, higher order wavelets have more non-zero filter coefficients and thus more expensive calculations per wavelet. Thus resulting in a trade-off between the number of wavelets and the order of the wavelets. For the calculations of the molecular systems and quantum dots which are described later the interpolets 6 have been used.

## 2.4.2 Poisson equation

An important part of electronic structure calculations is to solve integrals like

$$V_H(\mathbf{r}) = \int d\mathbf{r}' \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}. \quad (2.153)$$

However, such integrations are very costly and the singularity may raise problems. Another way to calculate this same potential  $V_H(\mathbf{r})$  is by solving the Poisson equation

$$\nabla^2 V_H(\mathbf{r}) = -4\pi\rho(\mathbf{r}). \quad (2.154)$$

In the wavelet basis this is equivalent to solving the linear equations

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad (2.155)$$

where  $\mathbf{A}$  is the Laplacian matrix in wavelet space and  $\mathbf{x}$  and  $\mathbf{b}$  are the wavelet expansions of the potential and the density respectively.

Solving,  $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$  is not the way to go. Namely,  $\mathbf{A}$  is singular so the inverse does not exist. Moreover, the inverse of a sparse matrix is not necessarily sparse itself. Thus,

even if the inverse would exist, it would probably, unlike matrix  $\mathbf{A}$  itself, not be a sparse matrix, resulting in inefficient calculation.

A better way is to use an iterative procedure like the conjugate gradient method to minimize the function

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{b}, \quad (2.156)$$

since in this minimum  $\nabla f(\mathbf{x}) = \mathbf{A} \mathbf{x} - \mathbf{b} = \mathbf{0}$  holds.

**One dimensional** Because  $\mathbf{A}$  is a singular matrix, the solution for  $\mathbf{x}$  is not completely fixed. This can be seen in the following way:

$$V_H = V_{\text{homo}} + V_{\text{inhomo}}, \quad (2.157)$$

where the homogeneous part  $V_{\text{homo}}$  is of the form  $ax + b$  in one dimensional space. In case of periodic boundary conditions the value of  $a$  is fixed to 0. But the constant  $b$  is free, causing the solution for  $V_H$  to be fixed up to a constant.

This can be solved using a penalty function. By adding the penalty function a new variational principle is introduced

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{b} + \frac{1}{2} K (x_0 - \tilde{x}_0)^2. \quad (2.158)$$

This extra penalty  $\frac{1}{2} K (x_0 - \tilde{x}_0)^2$  translates to adding an extra constant  $K$  to the matrix element  $A_{0,0}$  and the value  $K\tilde{x}_0$  to  $b_0$  and will force the right offset, making the matrix non-singular.

**Preconditioning** The convergence rate of the conjugate gradient method used to solve the Poisson equation depends on the spectral properties of the coefficient matrix for the Laplacian  $\mathbf{A}$ . A measure for these spectral properties is the condition number of the matrix. Using this condition number a relation for the upper-bound of the error as a function of the iteration number  $k$  can be derived [Gol96]

$$\|\mathbf{x}_k - \hat{\mathbf{x}}\|_A \leq 2 \|\mathbf{x}_0 - \hat{\mathbf{x}}\|_A \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k, \quad (2.159)$$

where  $\kappa$  is the spectral condition number, the norm is defined as  $\|\mathbf{y}\|_A = \sqrt{\mathbf{y}^T \mathbf{A} \mathbf{y}}$ , and  $\hat{\mathbf{x}}$  the absolute minimizer. Thus, the closer the condition number is to one, the faster the convergence

Using a preconditioner  $\mathbf{M}$  the linear system for the Poisson equation can be transformed into one that has the same solution, but that has more favorable spectral properties, i.e., a lower condition number:

$$\mathbf{M}^{-1} \mathbf{A} \mathbf{x} = \mathbf{M}^{-1} \mathbf{b}. \quad (2.160)$$

Since applying a preconditioner introduces some extra costs, both initially and per iteration, there is a trade-off between the cost of constructing and applying the preconditioner, and the gain in convergence speed. The initial cost is in generating the preconditioner matrix, whereas in every iteration a matrix multiplication has to be performed.

Since the matrix  $\mathbf{A}$  for the Laplacian is sparse, viz. band-diagonal, the preconditioner preferably preserves this sparsity pattern in order for the extra costs not to become dom-

inant. In the rest of this section various preconditioners [Bar94] will be discussed and tested for the one dimensional case. The results obtained will be used to construct a good preconditioner for the higher dimensional cases.

**Jacobi preconditioner** The simplest preconditioner consists of just the diagonal of the original matrix  $\mathbf{A}$ :

$$m_{i,j} = \begin{cases} a_{i,i} & \text{if } i = j, \\ 0 & \text{else.} \end{cases} \quad (2.161)$$

This Jacobi preconditioner has the advantage that it requires no extra storage, that it is very easy to implement and that the extra computational costs are very low. Despite its simplicity it reduces the condition number notably, although not as much as more sophisticated preconditioners to follow.

**SSOR preconditioner** Like the Jacobi preconditioner, the symmetric successive over-relaxation preconditioner (SSOR) can be derived directly from the coefficient matrix  $\mathbf{A}$ . If the original, symmetric, matrix is decomposed as

$$\mathbf{A} = \mathbf{D} + \mathbf{L}_A + \mathbf{L}_A^T \quad (2.162)$$

in its diagonal, lower, and upper triangular part, the SSOR matrix is defined as

$$\mathbf{M} = (\mathbf{D} + \mathbf{L}_A) \mathbf{D}^{-1} (\mathbf{D} + \mathbf{L}_A)^T. \quad (2.163)$$

This is often parameterized to

$$\mathbf{M} = \frac{1}{2-\omega} \left( \frac{1}{\omega} \mathbf{D} + \mathbf{L}_A \right) \left( \frac{1}{\omega} \mathbf{D} \right)^{-1} \left( \frac{1}{\omega} \mathbf{D} + \mathbf{L}_A \right)^T \quad (2.164)$$

what for the optimal  $\omega$  will reduce the condition number even further. The advantage of these preconditioners is that  $\mathbf{L}_A$  is strictly lower triangular, band limited, because of which the product of the inverse with  $\mathbf{A}$  can easily be calculated using forward substitution in linear time. In the same way, the product of  $\mathbf{A}$  with  $\mathbf{L}_A^T$  can be calculated using back substitution.

**Incomplete factorization preconditioners** A way to solve a set of linear equations  $\mathbf{Ax} = \mathbf{b}$  directly is by LU-decomposition of the coefficient matrix  $\mathbf{A}$ . The action of  $\mathbf{A}^{-1}$  on  $\mathbf{b}$  can again be computed using forward and back substitution of respectively the lower triangular ( $\mathbf{L}$ ) and the upper triangular ( $\mathbf{U}$ ) part. The problem, however, is that though  $\mathbf{A}$  is sparse,  $\mathbf{L}$  and  $\mathbf{U}$  are not sparse, thus resulting in an order  $N^2$  algorithm.

A broad class of preconditioners is based on incomplete factorizations of the coefficient matrix. A factorization is called incomplete if during the factorization process certain fill elements, zero positions that would be nonzero in an exact factorization, have been ignored. The efficacy of such a preconditioner depends on how well it approximates  $\mathbf{A}$ . Incomplete factorizations come in two types. In the first place, one can ignore all fill outside selected positions in  $\mathbf{L}$ , thus retaining a certain sparsity pattern. The second option is to discard all fill whose magnitude is below a preset drop tolerance.

Here only the former method is considered, because for the Laplacian matrix the latter results in too dense a matrix. The fixed sparsity pattern is chosen to be equal to the sparsity pattern of the original coefficient matrix  $\mathbf{A}$ , i.e., no fill-in. This factorization type is called the Incomplete Factorization of degree zero.

Two types will be considered:

- The sparse approximate inverse preconditioner
- The incomplete Cholesky factorization with no fill-in.

**Sparse Approximate Inverse preconditioner** The inverse of  $\mathbf{A}$  can be factorized as follows

$$\mathbf{A}^{-1} = \mathbf{Z}\mathbf{D}^{-1}\mathbf{Z}^T, \quad (2.165)$$

where  $\mathbf{Z}$ , with columns  $\mathbf{z}_i$ , is upper triangular using the following scheme [Ben96]:

```

 $\mathbf{z}_i^{(0)} = \mathbf{e}_i, \quad (0 \leq i < N)$ 
for  $i = 0$  to  $N - 1$  do
  for  $j = i$  to  $N - 1$  do
     $d_{j,j}^{(i)} = \mathbf{a}_i^T \mathbf{z}_j^{(i)}$ 
  end for
  for  $j = i + 1$  to  $N - 1$  do
     $\mathbf{z}_j^{(i+1)} = \mathbf{z}_j^{(i)} - \frac{d_{j,i}^{(i)}}{d_{i,i}^{(i)}} \mathbf{z}_i^{(i)}$ 
  end for
end for
 $\mathbf{z}_i = \mathbf{z}_i^{(i)} \text{ and } d_{i,i} = d_{i,i}^{(i)}, \quad (0 \leq i < N)$ 

```

However, using an incomplete factorization in the same scheme a preconditioner can be created that is an approximation for the inverse

$$\mathbf{M}^{-1} = \bar{\mathbf{Z}} \bar{\mathbf{D}}^{-1} \bar{\mathbf{Z}}^T \approx \mathbf{A}^{-1}, \quad (2.166)$$

where  $\bar{\mathbf{Z}}$  has the same sparsity pattern as the upper triangular part of  $\mathbf{A}$ .

**Incomplete Cholesky preconditioner** Again the matrix is split in its lower, upper and diagonal part

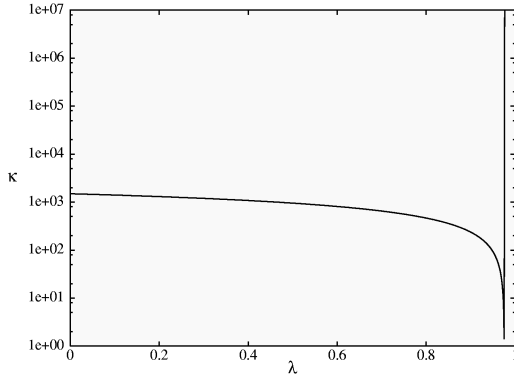
$$\mathbf{A} = \mathbf{L}_A + \text{diag}(\mathbf{A}) + \mathbf{L}_A^T. \quad (2.167)$$

The preconditioner is now

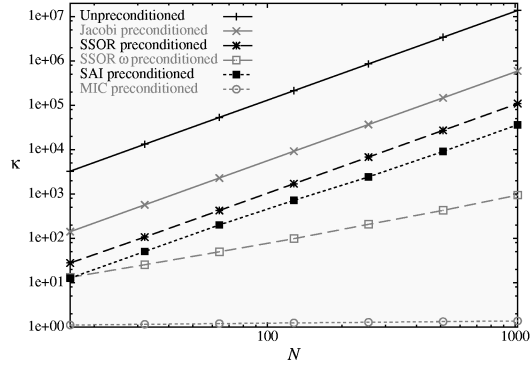
$$\mathbf{M} = (\mathbf{L}_A + \mathbf{D})\mathbf{D}^{-1}(\mathbf{L}_A^T + \mathbf{D}) \quad (2.168)$$

where  $\mathbf{D}$  contains no longer the diagonal of the original matrix, but is a diagonal matrix such that  $\text{diag}(\mathbf{M}) = \text{diag}(\mathbf{A})$ . The diagonal elements  $d_{i,i}$  of the diagonal matrix  $\mathbf{D}$  that make this true are given by:

$$d_{i,i} = \begin{cases} a_{0,0} & \text{if } i = 0 \\ a_{i,i} - a_{i,i-1} d_{i-1,i-1}^{-1} a_{i-1,i} & \text{if } 0 < i < N. \end{cases} \quad (2.169)$$



**Figure 2.11:** The condition number of the MIC preconditioned Laplacian matrix as a function of  $\lambda$ .



**Figure 2.12:** The condition number  $\kappa$  of the preconditioned Laplacian matrix as a function of the matrix size  $N$  using the various preconditioners.

However, this method breaks down for the Laplacian matrix in the sense that it yields negative pivots in case more than 16 interpolets are used. Therefore, the procedure is slightly modified introducing a parameter  $\lambda \in [0, 1]$  resulting in a modified incomplete Cholesky factorization MIC(0):

$$d_{i,i} = \begin{cases} a_{0,0} & \text{if } i = 0 \\ a_{i,i} - \lambda a_{i,i-1} d_{i-1,i-1}^{-1} a_{i-1,i} & \text{if } 0 < i < N. \end{cases} \quad (2.170)$$

The condition number as a function of  $\lambda$  has been plotted in Figure 2.11 for matrix size  $N = 128$ . In the limit  $\lambda = 0$  this preconditioner is exactly the SSOR preconditioner. The larger  $\lambda$  the smaller the condition number until the procedure breaks down because of the appearance of negative pivots, like for  $\lambda = 1$ , where we have again the standard Incomplete Cholesky factorization. Somewhere in between there is an optimal value  $\lambda_{\text{opt}}$ . This optimal value can be derived from Eq. (2.170) as follows. Let the series  $d_{i,i}$  converge to  $p$ , then by substituting  $a_{i,i} = r_0$  and  $a_{i-1,i} = a_{i,i-1} = r_1$  and rewriting Eq. (2.170) becomes

$$p^2 - pr_0 + \lambda r_1^2 = 0. \quad (2.171)$$

The minimum of the parabola is in  $p = r_0/2$ . Solving Eq. (2.171) in this minimum yields for  $\lambda$

$$\lambda_{\text{opt}} = \frac{r_0^2}{4r_1^2}. \quad (2.172)$$

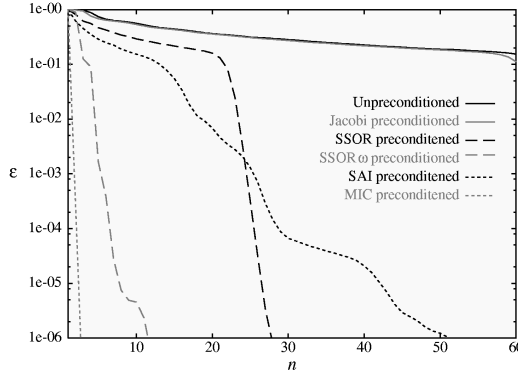
Substituting the values for  $r_0 = -20/9$  and  $r_1 = 9/8$  for the interpolet number 4 this yields

$$\lambda_{\text{opt}} = \frac{6400}{6561} \approx 0.9754610577656. \quad (2.173)$$

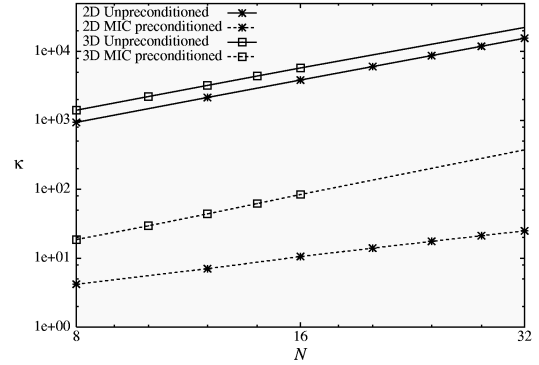
This method has the advantage that again only the original matrix elements are necessary for off-diagonal elements. Only the diagonal elements have to be calculated.

In order to compare all these different preconditioners, the condition numbers for the preconditioned Laplacian matrix using these preconditioners as a function of the size of the matrix has been plotted in Figure 2.12. As can be seen, the modified Incomplete





**Figure 2.13:** The error  $\epsilon$  as a function of the iteration number using various preconditioners.



**Figure 2.14:** The condition number  $\kappa$  of the unpreconditioned and the MIC preconditioned Laplacian matrix as a function of the size  $N$  for both two and three dimensions.

Cholesky factorization yields the lowest condition number by far.

Using these preconditioners the solution of the Poisson equation for a Gaussian density distribution has been calculated using 128 wavelets to show how these calculations converge to the analytical result. This is shown in Figure 2.13 where the error, i.e. the difference between the calculated potential  $V_H$  and the analytically expected result, has been plotted as a function of the iteration number. The figure shows that the MIC preconditioner, which also has the lowest condition number, results in the fastest convergence. This preconditioner is therefore used in further calculations.

**Higher dimensions** Also in higher dimensions the Laplacian matrix is singular. This singularity can be handled in the same way as in 1D, i.e., by adding a penalty function to force the boundary to a fixed value:

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{b} + \frac{1}{2} K \left( \sum_{i=0}^{N-1} (x_i - \tilde{x}_i)^2 + \sum_{j=0}^{N-1} (x_{jN} - \tilde{x}_{jN})^2 \right). \quad (2.174)$$

For these boundary values the analytically solvable results are taken for the potential of a point charge with the same total charge and center. When necessary, corrections to the asymptotes of higher multipoles can be added.

Equation (2.170) can be generalized for higher dimensions. In the two dimensional case it becomes

$$d_{i+jN, i+jN} = a_{i+jN, i+jN} - \lambda \delta_i \frac{a_{i+jN, i+jN-1} a_{i+jN-1, i+jN}}{d_{i+jN-1, i+jN-1}} - \lambda \delta_j \frac{a_{i+jN, i+jN-N} a_{i+jN-N, i+jN}}{d_{i+jN-N, i+jN-N}}, \quad (2.175)$$

where

$$\delta_i = \begin{cases} 0 & \text{if } i = 0, \\ 1 & \text{otherwise.} \end{cases} \quad (2.176)$$

The optimal value for  $\lambda$  can be derived in the same way as in the one dimensional case, and is for the  $n$ -dimensional case given by  $\lambda_{\text{opt}}^{nD} = n \lambda_{\text{opt}}^{1D}$ .

The condition numbers for both the 2D and 3D Laplacian are given in Figure 2.14, both for the MIC preconditioned and the unpreconditioned case. The condition numbers for the preconditioned cases are again much lower than for the unpreconditioned cases, thus again improving the convergence. In 3D, using  $128^3$  interpolets the error in the Hartree energy for a Gaussian density distribution converges from a random initial guess to  $10^{-6}$  in about 40 iterations.

### 2.4.3 Potential well

Another physically interesting problem in one dimensional space is the potential well. This problem is again described by the Schrödinger Equation (2.137), where the potential is given by

$$V(x) = \begin{cases} -V_0 & \text{if } -a < x < a, \\ 0 & \text{else.} \end{cases} \quad (2.177)$$

This problem has been studied extensively in literature and can be solved analytically as well, see e.g. Ref. [Gas74]. It is discussed here because it is especially suitable to demonstrate the power of pruning. The potential is very smooth except for two discontinuities. As a result low resolution suffices in most of the space whereas high resolution is needed only in the vicinity of the steps in the potential, making this problem an excellent example to demonstrate pruning.

**Analytical solution** Because the potential well is negative, energy states with a negative energy do exist. The energies of these states are given by

$$E = -\frac{1}{2}k^2. \quad (2.178)$$

The solutions outside the well that are bounded at infinity are

$$\Psi(x) = C_1 e^{kx} \quad x < -a, \quad (2.179)$$

$$\Psi(x) = C_2 e^{kx} \quad x > a, \quad (2.180)$$

and the solutions inside the well

$$\Psi(x) = A \cos qx + B \sin qx \quad -a \leq x \leq a, \quad (2.181)$$

where

$$q^2 = 2(V_0 - |E|). \quad (2.182)$$

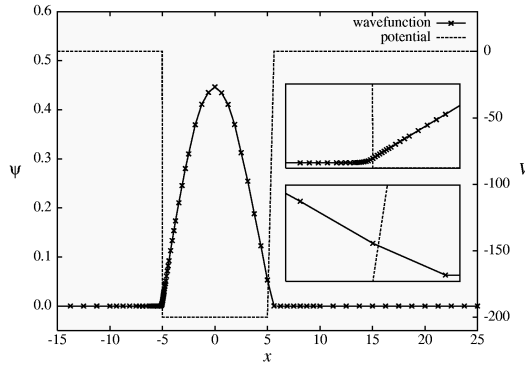
Combining these by matching solutions and derivatives at the edges yields that for the even and odd solutions, respectively,  $k$  is of the form

$$k = q \tan qa \quad \text{and} \quad k = -q \cot qa, \quad (2.183)$$

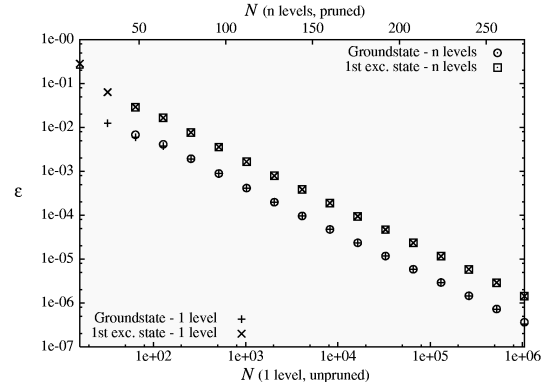
which can be rewritten using  $\lambda = 2V_0 a^2$  and  $\gamma = qa$  as

$$\frac{\lambda - \gamma^2}{\gamma} = \tan \gamma \quad \text{and} \quad \frac{\lambda - \gamma^2}{\gamma} = -\cot \gamma. \quad (2.184)$$

Solving these equations yields all energies of the even and odd solutions respectively, i.e., if  $\gamma$  is known  $k$  is known via Eq. (2.183) and thus the energy via Eq. (2.178).



**Figure 2.15:** The one dimensional potential well, with  $a = 5$ ,  $b = 0$ , and  $V_0 = 200$  and the ground-state wavefunction as calculated in the pruned wavelet basis, with wavelets centered at the position of all crosses. In the insets has been zoomed in on the vicinity of the steps in the potential.



**Figure 2.16:** The error in the calculated energy for the ground state and the first excited state both using the pruned and the unpruned basis.

**Numerical solution** As an example a one dimensional potential well of depth  $V_0 = 200$  and width  $a = 5$  is considered in a periodic box of length 40. The discontinuities in such a potential cause the need of very high resolution to perform accurate calculations. However, in most of the space the potential and the resulting wavefunctions are very smooth and require only low resolution. Using wavelets it is now possible to do exactly this, namely to use low resolution in general and higher resolution only there where necessary, by means of pruning. To demonstrate this a wavelet basis is created that starts with 32 wide wavelets which cover the whole range of the computation. This will provide a good basis for the region away from the well. An extra 16 narrower wavelets are added in the area around the well ( $-10 \leq x < 10$ ) and another 5 levels with 8 narrower and narrower wavelets each around the left step in the potential.

In Figure 2.15 the potential (right y-axis) and the resulting ground-state wavefunction as calculated (left y-axis) have been shown using this wavelet basis. In the two insets in the same figure has been zoomed in on the regions around the two steps in the potential. Because of symmetry in the problem both sides should be the same. But because of the lack of resolution at the right step, where contrary to the left step no extra levels of narrower wavelets were added, the potential could not be represented accurately enough. This causes the potential to become smoother and the resulting wavefunction to be inaccurate. The difference shows that the extra resolution used at the left side is needed.

By adding the same 5 levels with 8 wavelets each around the right step as were used around the left step a wavelet basis is created consisting of 128 wavelets. This basis corresponds to a basis of 2048 wavelets if the basis had not been pruned or only the finest level wavelets had been used. For our example both bases obtain the same precision result of the energy.

In order to obtain a higher precision, an extra level of resolution should be added. For the pruned basis this means adding another 16 wavelets, whereas for the unpruned basis this means doubling the number of wavelets. To show how the error depends on the number of wavelets the ground state and the first excited state have been calculated. The ground state is even and the first excited state odd. The theoretical energies of these states are  $-199.9516243676$  and  $-199.8064979341$  respectively. For both states the error, i.e. the difference between the calculated and the theoretical energy, has been plotted in Figure 2.16 as a function of the number of wavelets used, both for the pruned (top  $x$ -axis) and the unpruned (bottom  $x$ -axis) basis. Because the errors are equal for the pruned and the unpruned basis with corresponding finest resolution can be concluded that indeed low resolution satisfies in most of the space and that the higher resolution is only needed around the steps in the potential.

In the one dimensional case, the number of wavelets in an unpruned basis doubles when an extra level of resolution is added, whereas in a pruned basis only some wavelets have to be added where the extra resolution is needed. In  $n$ -dimensional space the number of wavelets multiplies even by  $2^n$  for every extra level of resolution, causing the reduction in number of wavelets in the pruned basis in higher dimensional cases to be even much larger.

## 2.5 Application to 3D molecular systems

So far has been shown by means of some well-known problems that wavelets can be used for electronic structure calculations. Now our approach of the electronic structure calculation for molecular (3D) systems will be described. The electronic structure is calculated by solving the Kohn-Sham equations self-consistently.

### 2.5.1 Kohn-Sham equations

As described before, density functional theory states that the many electron problem can be replaced by an equivalent set of self-consistent one-electron equations, the Kohn-Sham equations. In the local-spin-density-approximation (LSD), these equations read

$$\hat{H}\Psi_i^\sigma(\mathbf{r}) = \epsilon_i^\sigma \Psi_i^\sigma(\mathbf{r}), \quad (2.185)$$

where the Hamiltonian  $\hat{H}$  is given by

$$\hat{H} = -\frac{1}{2}\nabla^2 + \hat{V}_{pp}(\mathbf{r}) + \hat{V}_H(\mathbf{r}) + \hat{V}_{xc}^\sigma(\mathbf{r}). \quad (2.186)$$

The eigenfunctions  $\Psi_i^\sigma$  are the one-electron wavefunctions that correspond to the minimum of the Kohn-Sham energy functional. In these wavefunctions,  $i$  is the orbital index and  $\sigma$  denotes the spin, which can be either up  $\uparrow$  or down  $\downarrow$  in case of the local spin density approximation (LSD) and has only one value 0 in case of the local density approximation (LDA). The Hamiltonian  $\hat{H}$  consists of four different parts: a part related to the kinetic energy of the electrons, the pseudopotential  $\hat{V}_{pp}$ , the Hartree potential  $\hat{V}_H$  and

the exchange-correlation potential  $\hat{V}_{xc}$ . The details of these potentials are described below. The potentials depend on the wavefunction via the electron spin density  $\rho^\sigma$ , which is defined as

$$\rho_\sigma(\mathbf{r}) = \sum_i f_i^\sigma |\Psi_i^\sigma(\mathbf{r})|^2. \quad (2.187)$$

Here  $f_i^\sigma$  is the occupation number, i.e., the number of electrons in orbital  $i$ . In case of LSD every orbital can contain at most one electron. In case of LDA where there is no longer a distinction between spin up and spin down, orbitals can contain at most two electrons.

## 2.5.2 Pseudopotentials

**Why pseudopotentials** In the original Kohn-Sham equations, the interaction of the positively charged nuclei with the electrons is described using the full Coulombic potential  $\hat{V}_{ne}$ . However, it is well-known that most physical and chemical properties are dependent on the valence electrons to a much greater degree than on the tightly bound core electrons. In practice, this means that the innermost electrons can be left out of explicit calculations. Instead these chemically inert core electrons are eliminated in the frozen-core approximation, being considered together with the nuclei as rigid ion cores. In turn, all electrostatic and quantum-mechanical interactions of the valence electrons with the ion cores are accounted for by an effective potential, the so-called pseudopotential  $\hat{V}_{pp}$ .

The advantage is twofold. In the first place the number of effective electrons, and thus the number of orbitals, is smaller. Secondly, the orbitals are smoother, where the singularity in the normal electron-nucleus interaction is removed and where the nodal structure near the nuclei that keeps core and valence states orthogonal in an all-electron framework are avoided.

Pseudopotentials of course also have their disadvantage in that they introduce an approximation. Apart from efficiency a principle objective is the transferability of the pseudopotential. This transferability is the ability of one and the same pseudopotential to accurately describe the valence electrons in different atomic, molecular, and solid-state environments. Regarding transferability the quality of pseudopotentials depends on correct scattering properties, the actual choice of the core radius cut-off  $r_c$ , an adequate account of the non-linear exchange-correlation interaction between core and valence electrons, the validity of the frozen core approximation, and the implicit treatment of higher angular momentum components.

Concluding, pseudopotentials are thus required to correctly represent the long range interactions of the core and to produce pseudo-wavefunctions that approach the full wavefunction outside a core radius  $r_c$ . Inside this radius, the pseudopotential and the wavefunction should be as smooth as possible, minimizing the number of necessary basis functions in the expansion.

**Kinds of pseudopotentials** Many pseudopotentials have been reported. Usually they consist of both a local and a non-local part

$$\hat{V}_{pp}(\mathbf{r}) = V_{\text{local}}(r) + \sum_l |l\rangle \hat{V}_l(r, r') \langle l|. \quad (2.188)$$

The first pseudopotentials were empirical. We have implemented the Shaw [Sha68] pseudopotential and the Topp-Hopfield [Top73] pseudopotential, which are examples of simple, completely local pseudopotentials. These pseudopotentials are of the form

$$V_{pp}^{\text{S,TH}}(\mathbf{r}) = \begin{cases} -Z/r & r \geq r_c \\ f(r) & r \leq r_c \end{cases}, \quad (2.189)$$

where  $f(r)$  equals a constant  $C$  for the Shaw pseudopotential whereas it is of the form  $V_0 \cos(kr) + C$  for the Topp-Hopfield pseudopotential. The Shaw pseudopotential is available for some atoms and the Topp-Hopfield pseudopotential for sodium only.

Phillips and Kleinman [Phi59] showed that the effective potential which has the pseudo-wavefunctions as its eigenstates could be derived from the all-electron potential and the core state wavefunctions and energies, thus introducing a non-empirical approach to finding pseudopotentials. These methods became useful when the norm-conservation constraint was introduced. This constraint ensures that outside the core the pseudo wavefunctions behave like their all-electron counterparts. Because of this, properly constructed pseudopotentials obeying the norm-conserving property present a rather accurate approximation and allow for an adequate description of the valence electrons over the entire chemically relevant range of systems: atoms, molecules, and solids.

We have implemented the Bachelet-Hamann-Schlüter [Bac82] pseudopotential, which is especially suited for plane waves what makes the non-local part hard to implement for wavelets, and the Hartwigsen-Goedecker-Hutter [Har98] pseudopotential, which is well suited also for grid methods. These Hartwigsen-Goedecker-Hutter pseudopotentials have the following form for the local part

$$\begin{aligned} \hat{V}_{loc}^{HGH}(r) = & -\frac{Z}{r} \text{erf}\left(\frac{r}{\sqrt{2}r_{loc}}\right) + \exp\left(-\frac{1}{2}\left(\frac{r}{r_{loc}}\right)^2\right) \\ & \times [c_1 + c_2\left(\frac{r}{r_{loc}}\right)^2 + c_3\left(\frac{r}{r_{loc}}\right)^4 + c_4\left(\frac{r}{r_{loc}}\right)^6] \end{aligned} \quad (2.190)$$

and for the non-local part

$$\hat{V}_l^{HGH}(\mathbf{r}, \mathbf{r}') = \sum_{i=1}^3 \sum_{j=1}^3 p_i^l(r) h_{i,j}^l p_j^l(r') \quad (2.191)$$

where

$$p_i^l(r) = \frac{\sqrt{2} r^{l+2(i-1)} \exp\left(-\frac{r^2}{2r_l^2}\right)}{r_l^{l+(4i-1)/2} \sqrt{l + \frac{4i-1}{2}}}. \quad (2.192)$$

The local part of the pseudopotentials only depends on the positions of the nuclei. Thus for a fixed nuclear configuration this only has to be calculated once, whereas the non-local part depends on the wavefunction and thus has to be recalculated for every new approximation of the wavefunctions.

### 2.5.3 Exchange-correlation energy

The exchange correlation potential  $\hat{V}_{xc}$  describes the non classical interaction between the electrons and is given by the functional derivative of an exchange correlation energy functional

$$V_{xc}^\sigma(\mathbf{r}) = \frac{\delta E_{xc}(\rho_\uparrow, \rho_\downarrow)}{\delta \rho_\sigma}. \quad (2.193)$$

The approximation of this unknown functional  $E_{xc}(\rho)$  is of crucial importance to any application of density functional theory. The exchange correlation functionals considered here belong to the class of generalized gradient approximations

$$E_{xc}^{GGA} = \int d\mathbf{r} \rho(\mathbf{r}) \epsilon_{xc}^{GGA}(\rho(\mathbf{r}), \nabla \rho(\mathbf{r})), \quad (2.194)$$

where the unknown functional is approximated by a functional that only depends on the density and the gradient of this density at a given point in space. In the simplest case  $\epsilon_{xc}^{GGA}(\rho)$  is the exchange correlation energy density of an interacting but homogeneous electron gas with the density given by the local density  $\rho(\mathbf{r})$  at space-point  $\mathbf{r}$  in the inhomogeneous system. This simple but powerful approximation is called the local density approximation (LDA) for spin-unpolarized systems, and it is called local-spin-density approximation (LSD) for spin-polarized systems. It is of the form

$$E_{xc}^{LSD} = - \int d\mathbf{r} c_x \left( \rho_\uparrow(\mathbf{r})^{4/3} + \rho_\downarrow(\mathbf{r})^{4/3} \right). \quad (2.195)$$

The corresponding exchange potentials are

$$V_{xc}^\sigma(\mathbf{r}) = -\frac{4}{3} c_x \rho_\sigma(\mathbf{r})^{1/3}. \quad (2.196)$$

where  $c_x = \frac{3}{4} \left( \frac{6}{\pi} \right)^{1/3}$ . In case of LDA this reduces to  $V_{xc}^{LDA}(\mathbf{r}) = -\frac{4}{3} c_x \rho_0(\mathbf{r})^{1/3}$  where the constant  $c_x$  is equal to  $\frac{3}{4} \left( \frac{3}{\pi} \right)^{1/3}$ .

More precise approximations for the exchange correlation potential do not only use the density, but also the gradient of the density  $\nabla \rho_\sigma$ . In these approximations the combined exchange-correlation functional is typically split in two additive terms,  $\epsilon_x$  and  $\epsilon_c$  for exchange and correlation respectively.

We have implemented one such generalized gradient approximation (GGA) by Becke [Bec88] and one by Perdew, Burke and Ernzerhof [Per96]. The formula for the Becke exchange energy functional is

$$E_x^{\text{Becke}} = E_{xc}^{LSD} - \beta \int d\mathbf{r} \rho^{4/3} \frac{2^{1/3} \eta^2}{1 + 6\beta 2^{1/3} \eta \sinh^{-1}(2^{1/3} \eta)}, \quad (2.197)$$

where  $\beta$  is a parameter and  $\eta$  the dimensionless quantity for the gradient of the density

$$\eta = \frac{|\nabla \rho|}{\rho^{4/3}}. \quad (2.198)$$

Out of this functional the potential can be derived as the derivative to the density. This gives a complicated expression, depending on  $\eta$  and  $\tau$ , a dimensionless quantity for the Laplacian of the density,

$$\tau = \eta^2 - \frac{\nabla^2 \rho}{\rho^{5/3}}. \quad (2.199)$$

The formula for the Perdew, Burke and Ernzerhof exchange energy functional is

$$E_x^{\text{PBE}} = \int d\mathbf{r} \rho \epsilon_x^{\text{unif}}(\rho) \left( 1 + \kappa - \frac{\kappa}{1 + \frac{\mu \eta^2}{\kappa}} \right), \quad (2.200)$$

and for the correlation energy functional

$$E_c^{\text{PBE}} = \int d\mathbf{r} \rho \left( \epsilon_c^{\text{unif}}(\rho) + h(\rho, \eta) \right). \quad (2.201)$$

Also for the correlation energy many functionals exist. However, this part of the Hamiltonian will have an even smaller contribution to the final result than the exchange one, so we choose for a local version: the correlation energy functional of Perdew and Wang [Per92]. This functional is written in terms of the so-called density parameter

$$r_s = \left( \frac{3}{4\pi\rho} \right)^{1/3}. \quad (2.202)$$

The functional is

$$E_c^{\text{PW}} = -2 \int d\mathbf{r} \rho A (1 + \alpha_1 r_s) \ln \left( 1 + \frac{1}{2A(\beta_1 r_s^{1/2} + \beta_2 r_s + \beta_3 r_s^{3/2} + \beta_4 r_s^{p+1})} \right), \quad (2.203)$$

where  $A, \alpha_1, \beta_1, \beta_2, \beta_3$  and  $\beta_4$  are parameters.

A promising next step are functionals that include higher-order powers of the gradient in the sense of a generalized gradient expansion beyond the first term, e.g. including the Laplacian. However, since the price performance ratio in other methods is still not clear such functionals have not been considered here.

## 2.5.4 Hartree potential

The Hartree potential  $\hat{V}_H$  describes the interaction between electrons and is given by

$$\hat{V}_H(\mathbf{r}) = \int d\mathbf{r}' \frac{\rho_\uparrow(\mathbf{r}') + \rho_\downarrow(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}. \quad (2.204)$$

An important part of the electronic structure calculations is to solve this type of integrals. However, such integrations are very costly, and problematic because of the singularity. Instead of calculating the Hartree potential in this way directly, it can also be calculated by solving the Poisson equation

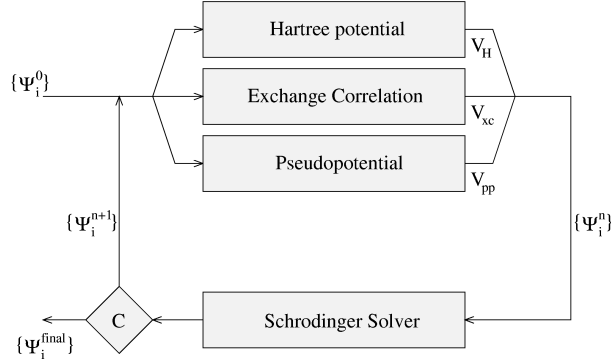
$$\nabla^2 V_H(\mathbf{r}) = -4\pi (\rho_\uparrow(\mathbf{r}) + \rho_\downarrow(\mathbf{r})). \quad (2.205)$$

The way this Poisson equation can be solved has been discussed in Section 2.4.2.

## 2.5.5 Iterative scheme

As can be seen from Eqs. (2.188), (2.193) and, (2.204) the Hamiltonian  $\hat{H}$  depends on the density and via Eq. (2.187) thus on the wavefunctions. This system of non-linear coupled differential equations can be solved self-consistently. We start with an initial guess for the orbital wavefunctions  $\{\Psi_i^{\sigma,(0)}\}$ . The corresponding electron density is then calculated using Eq. (2.187). Given this density  $\rho^\sigma$  the Hartree potential, the exchange correlation potential and the non-local part of the pseudopotential are calculated. Once





**Figure 2.17:** The iterative scheme used for solving the Kohn-Sham equation.

the potentials have been calculated, a steepest descent or conjugate gradient method is employed to obtain a better approximation for the wavefunction. This is done in the same way as discussed in Section 2.4.1, i.e., by minimizing the orbital energy  $E_i^\sigma$ , which is related to the individual orbital  $\Psi_i^\sigma$  by

$$E_i^\sigma(\Psi_i^\sigma) = \frac{\langle \Psi_i^\sigma | \hat{H} | \Psi_i^\sigma \rangle}{\langle \Psi_i^\sigma | \Psi_i^\sigma \rangle}, \quad (2.206)$$

where the orbital wavefunctions  $\Psi_i^\sigma$  satisfy the appropriate orthonormalization conditions.

This is done in updating the wavefunction in a certain direction  $h_i^\sigma$

$$\Psi_i^{\sigma,(n+1)} = \Psi_i^{\sigma,(n)} + \lambda_i^{\sigma,(n)} h_i^{\sigma,(n)}. \quad (2.207)$$

where for the steepest descent method  $h_i^\sigma$  equals the gradient

$$g_i^{\sigma,(n)} = \frac{\delta E_i^\sigma(\Psi_i^\sigma)}{\delta \Psi_i^{\sigma*}} = \hat{H} \Psi_i^\sigma. \quad (2.208)$$

whereas for the conjugate gradient method  $h_i^\sigma$  is given by

$$h_i^{(n)} = \begin{cases} g_i^{(n)} & n = 0 \\ g_i^{(n)} + \gamma^{(n-1)} h_i^{(n-1)} & n > 0 \end{cases} \quad (2.209)$$

where

$$\gamma^{(n-1)} = \frac{\langle g_i^{(n)} | g_i^{(n)} \rangle}{\langle g_i^{(n-1)} | g_i^{(n-1)} \rangle}. \quad (2.210)$$

The optimal step size  $\lambda$  of the update is again given by Eq. (2.149).

The conjugate gradient method works as long as the Hamiltonian is constant. The non-local part of the pseudopotential, the Hartree potential and the exchange and correlation potential, however, depend on the wavefunction and thus change together with this wavefunction in every step of the iteration. For a number of iterations however the potentials can be kept constant. This number of iterations before the potentials are recalculated may be chosen fixed or it may depend on a convergence criterion. This procedure is repeated till self-consistency is reached.

Figure 2.17 gives a schematic overview of the approach used. We start with an initial guess for the orbital wavefunctions  $\{\Psi_i^0\}$ . The corresponding electron density is then

calculated using Eq. (2.187). Given this density  $\rho$  the Hartree potential, the exchange correlation potential and the non-local part of the pseudopotential are calculated.

## 2.5.6 Results for atoms and molecules

The main output of the program are energies and single electron orbitals. However, in computational chemistry these energies are of main interest, because from these energies many other properties can be derived, such as

- ionization potentials,
- electron affinities,
- excitation energies,
- binding energies,
- bond frequencies,
- force constants, and
- bond lengths.

Because of the use of pseudopotentials there can be an offset in the energies calculated. However, most interesting properties depend on differences of energies, where these offsets cancel each other.

**Atoms** For instance, for single atoms, using differences of energies, ionization potentials, electron affinities, and excitation energies can be calculated. The calculated ionization potentials and electron affinities for the first 11 elements of the periodic table are given in Table 2.7. Most of the ionization potentials can be calculated as the difference of two energies. Some fields for the second ionization potential are empty because

Atom	First ionization potential (eV)		Second ionization potential (eV)		Electron affinity (eV)	
	calc.	exp.	calc.	exp.	calc.	exp.
H	13.0	13.6			0.87	0.75
He	24.3	24.6	52.8	54.4	<sup>a</sup>	
Li	5.4	5.4		<sup>b</sup>	0.51	0.62
Be	8.9	9.3	18.0	18.2	<sup>a</sup>	
B	9.0	8.3	24.4	25.2	0.08	0.28
C	12.3	11.3	25.3	24.4	1.14	1.26
N	15.6	14.5	30.7	29.6	<sup>a</sup>	
O	14.2	13.6	36.4	35.1	0.89	1.46
F	17.1	17.4	34.2	35.0	2.85	3.40
Ne	21.6	21.6	40.6	41.0	<sup>a</sup>	
Na	5.2	5.1		<sup>b</sup>	0.63	0.55

**Table 2.7:** First and second ionization potentials and electron affinity for various atoms. Experimental data from [Lid93].

<sup>a</sup>Negative ion is not stable.

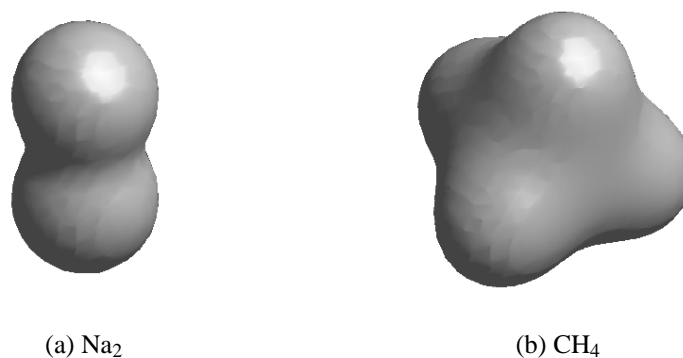
<sup>b</sup>Electron can not be removed since it is part of the core that is described by the pseudopotential.

they can not be calculated since the electrons that should be removed are part of the core that is described by the pseudopotentials. Besides, if the last valence electron is removed in the ionization one of the energies that is used to calculate the ionization potential is zero by definition. So here the offset in the energy may influence the result. About the electron affinities should be noted that not all atoms can accept an electron, e.g. for He, Be, N, and Ne this leads to unstable configurations, i.e., a negative electron affinity. All calculations have been performed on a cubic grid of  $64^3$  points using Hartwigsen-Goedecker-Hutter pseudopotentials, LSD exchange and correlation energy. Comparison with experimental data shows that our code yields accurate results.

**Molecules** The electronic structure of some small molecules has been calculated. The electron density of both di-sodium  $\text{Na}_2$  and methane  $\text{CH}_4$  has been plotted in Figure 2.18 using an iso-density surface.

The energies of a number of diatomic molecules have been calculated for several inter-nuclei distances. From these data several bond properties can be derived. The bond length can be found as the inter-nuclei distance ( $R$ ) for which the energy of the molecule  $E_{AB}$  is minimal. The binding energy  $B$  can be determined by the difference of the energy of the molecule minus the energies of the two separate atoms  $B = E_{AB} - E_A - E_B$ . Finally, the bond strength, which can be described both with the bond frequency  $\omega$  or with the force constant  $f$ , is related to the second derivative of the energy versus the inter-nuclei distance at the minimum.

These properties have been calculated for di-sodium using all four different pseudopotentials and LSD exchange and correlation energy. The results are given in Table 2.8a. The simple Shaw and Topp-Hopfield (T-H) pseudopotentials give slightly too weak binding, resulting in a too low binding energy, a larger than expected bond length, and too small a force constant. The Bachelet-Haman-Schlüter (BHS) and the Hartwigsen-Goedecker-Hutter (HGH) pseudopotentials on the other hand give slightly too strong a binding, resulting in slightly too high binding energy, a smaller than expected bond



**Figure 2.18:** Isodensity plots of the electron density for (a) di-sodium ( $\text{Na}_2$ ) and (b) methane ( $\text{CH}_4$ ).

	Shaw	T-H	BHS	HGH	Exp.
$B$ (eV)	0.60	0.69	0.84	0.83	0.76 <sup>(*)</sup>
$R$ (a.u.)	6.01	5.90	5.54	5.56	5.82 <sup>(*)</sup>
$f$ (N/cm)	0.13	0.15	0.19	0.18	0.17 <sup>(*)</sup>
$\omega$ (cm <sup>-1</sup> )	136	151	166	165	158 <sup>(*)</sup>

 (a) Di-sodium (Na<sub>2</sub>).

	LDA	LSD	GGA	Exp.
$B$ (eV)	6.30	4.90	5.11	4.52 <sup>(*)</sup>
$R$ (a.u.)	1.48	1.44	1.40	1.40 <sup>(*)</sup>
$f$ (N/cm)	4.48	5.41	5.77	5.75 <sup>(*)</sup>
$\omega$ (cm <sup>-1</sup> )	3882	4269	4410	4401 <sup>(*)</sup>

 (b) The hydrogen molecule (H<sub>2</sub>).

	LDA	LSD	GGA	Exp.
$B$ (eV)	16.80	10.76	12.35	11.16 <sup>(*)</sup>
$R$ (a.u.)	2.14	2.12	2.10	2.13 <sup>(*)</sup>
$f$ (N/cm)	15.57	17.98	19.46	19.02 <sup>(*)</sup>
$\omega$ (cm <sup>-1</sup> )	1963	2109	2194	2169 <sup>(*)</sup>

(c) Carbon monoxide (CO).

	LSD	GGA	Exp.
$B$ (eV)	20.3	17.9	18.2 <sup>(†)</sup>
$R$ (a.u.)	2.06	2.04	2.05 <sup>(*)</sup>
$f$ (N/cm)	5.31	5.45	5.44 <sup>(*)</sup>
$\omega$ (cm <sup>-1</sup> )	3113	3153	3151 <sup>(*)</sup>

 (d) Methane (CH<sub>4</sub>).

**Table 2.8:** Comparison of bond properties as calculated using various pseudopotentials and various functionals with experimental results from <sup>(\*)</sup>[Lid93] and <sup>(†)</sup>[Per96] for various molecules.

length, and a too large force constant.

For the hydrogen molecule H<sub>2</sub> and for carbon monoxide CO the resulting properties are given in Tables 2.8b and 2.8c, where the results are shown for various exchange and correlation functionals in combination with HGH pseudopotentials. Here should be noted the substantial difference between LDA and LSD, showing that spin can not be neglected. In the third column (GGA) the Becke exchange energy functional is used instead of pure LSD. For molecules consisting of more than two atoms, like methane CH<sub>4</sub>,  $B$  is the atomization energy, i.e., the difference between the energy of the whole molecule and the sum of the energies of all separate atoms. Results for methane are shown in Table 2.8d. The frequency  $\omega$  calculated for methane is the one for the hydrogen stretching mode.

The results in the tables show again good agreement with results reported in literature obtained using Kohn-Sham calculations with different basis sets and experiments, see e.g. Ref. [Per96]. This shows that the program is useful for the calculation of diverse properties of atoms and molecular systems.

## 2.6 Application to 2D quantum dots

**Quantum dots** Quantum dots have recently attracted much interest both experimentally and theoretically [Joh95]. Quantum dots are man made nanoscale structures in which electrons are confined in all three spatial directions. As they show typical atomic properties like discrete energy levels and shell structures, they are often referred to as artificial atoms. However, in contrast to natural atoms, in quantum dots the number of electrons is tunable.

Starting from quantum dots as a structure more complex systems are conceivable and likely to have perspective in future applications. An example is the analogy of a two-atom molecule consisting of two coupled quantum dots, where the coupling can be both vertically and laterally.

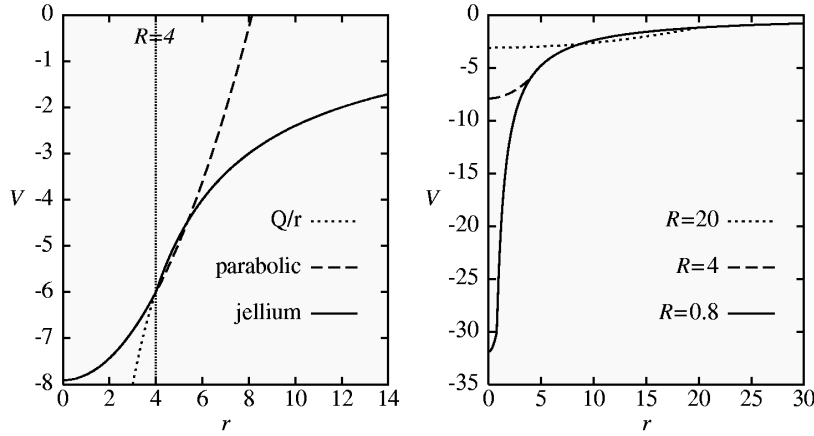
Many studies have been performed to describe quantum dots using a parabolic confining potential, see for example [Hir99, Lee98, Pfa93, Par00]. However, such infinite potentials are unphysical. Furthermore, they give practical problems when describing laterally coupled quantum dots. Various other potentials have been suggested that do not have this problem. A first example is a parabolic potential only from the nearest center [Wen00]. Such a potential enables lateral coupling but still has the disadvantage that it goes to infinity. In order to get rid of the infinity others have suggested potentials that are close to the harmonic potential at their bottom but that have a smooth lateral boundary with a finite asymptotic value. Examples of such potentials are a Gaussian potential [Ada00], and a smooth boundary potential [dF00].

Here we propose a new potential that is also harmonic at its center and has a constant asymptote, but that is physically motivated as derived from the description of a quantum dot as a 2D jellium. The potential we propose is namely the parabolic potential of a homogeneous density distribution within a 2D jellium, with the Coulombic potential outside that jellium. This potential is described first. A description follows of how the Kohn-Sham equations are used to calculate the electronic structure of such quantum dots using our wavelet based electronic structure code. This is used subsequently to study our new potential for various radii of the jellium. These results are compared with its limit for infinitely small and infinitely large  $R$ , i.e., 2D atoms and a pure parabolic potential respectively. Finally, a study is described of a quantum dot molecule consisting of two laterally coupled quantum dots.

**Potential** The potential that we propose to study laterally coupled quantum dots is derived from the description of a quantum dot as a 2D jellium. This stems from the way quantum dots can be created experimentally. Because of the application of a gate potential, an area of electron-depletion will appear near the gate which can be modeled most simply by assuming a constant positive background (i.e., jellium). We propose the parabolic potential of a homogeneous density distribution within the 2D jellium, with the Coulombic potential outside that jellium, i.e.,

$$V_e(r) = \begin{cases} -\frac{Q}{R} - \frac{Q}{4\pi R^2}(R^2 - r^2) & r \leq R \\ -\frac{Q}{r} & r > R, \end{cases} \quad (2.211)$$

where  $R$  is the radius of the circular jellium and  $Q$  is its total charge. The offset  $-\frac{Q}{R} - \frac{Q}{4\pi}$  in the harmonic part is required to make the potential continuous and asymptotically going to zero at large distances. This potential has been plotted in Figure 2.19. The left part of the figure shows how the potential is constructed from its two parts. The right hand side shows the potential for three different radii of the jellium.



**Figure 2.19:** Left: The potential consists of two parts: within a radius  $R$  quadratic and outside this region Coulombic. Right: The potential for three different radii of the jellium, with constant charge  $Q$ .

**Approach** The electronic structure for the two dimensional quantum dots is calculated by solving the Kohn-Sham equations numerically within the effective mass approximation

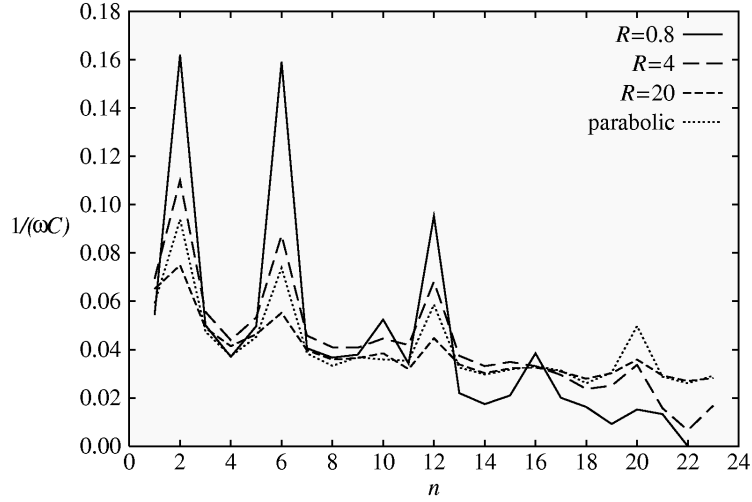
$$\left(-\frac{1}{2}\nabla^2 + V_e(\mathbf{r}) + V_H(\mathbf{r}) + V_{xc}(\mathbf{r})\right) \Psi_i(\mathbf{r}) = \epsilon_i \Psi_i(\mathbf{r}). \quad (2.212)$$

In case of GaAs the unit for length is  $101.89 \text{ \AA}$  and the unit for energy  $10.96 \text{ meV}$ . For solving this two dimensional problem we have developed a wavelet based electronic structure calculations program, analogous to our three dimensional electronic structure code for molecules. In short, the program calculates the electronic structure self-consistently. Starting from an initial guess of the wave functions, the parts of the Hamiltonian are calculated. Using this ‘fixed’ Hamiltonian, the wavefunction is improved on using a minimization scheme. Then the Hamiltonian is recalculated, the wavefunction updated, etc. This is repeated till a self-consistent solution has been reached. The choice for wavelets as basis set allows for an accurate description over a range of length scales. This is because wavelets constitute a multiresolution approach, allowing one to use low resolution and to add extra resolution only in those regions where necessary, which is a nice property because of the well-known fact that electronic wave functions vary much more rapidly near the ‘atomic’ centers than in inter-atomic regions.

The parts constituting the Hamiltonian are however different in two dimensional space from the ones in three dimensional space. The external potential describes the confinement of the quantum dot. We use the one proposed in Eq. (2.211) and the simple harmonic one as for comparison. A two dimensional version of the exchange and correlation energy functionals is given by Tanatar and Ceperley [Tan89]. They suggest the use of the local density approximation (LDA) for the exchange potential. In 2D this potential is of the form

$$V_x(\mathbf{r}) = -\sqrt{\frac{8\rho_e}{\pi}}. \quad (2.213)$$

The correlation potential is the functional derivative of this correlation energy func-



**Figure 2.20:** Scaled inverse of the capacitance for different confining potentials as a function of the number of electrons.

tional, yielding

$$V_c(\mathbf{r}) = \frac{a_0 (6a_1 t^2 + 7t^3)}{4(a_3 + a_2 t + a_1 t^2 + t^3)} - \frac{a_0 t^2 (a_2 t + 2a_1 t^2 + 3t^3)(a_1 + t)}{4(a_3 + a_2 t + a_1 t^2 + t^3)^2} \quad (2.214)$$

where  $t = (\pi \rho_e)^{1/4}$ . The Hartree potential  $V_H$  is calculated using a two dimensional version of Chelikowsky's direct integration method [Che94] that we have developed. Using this method, the Hartree potential  $V_H$  is evaluated on a grid by assuming the integrand does not change appreciably within a square of area  $h^2$  around each grid point  $i, j$ .  $V_H$  is given by

$$V_H(x_i, y_j) = \sum_{i', j'} \rho_e(x_{i'}, y_{j'}) g(x_i - x_{i'}, y_j - y_{j'}), \quad (2.215)$$

where for  $i, j \neq i', j'$

$$g(x_i - x_{i'}, y_j - y_{j'}) = \frac{h^2}{\sqrt{(x_i - x_{i'})^2 + (y_j - y_{j'})^2}}. \quad (2.216)$$

For  $i, j = i', j'$ , i.e., near the square root singularity, an explicit integration over the square yields

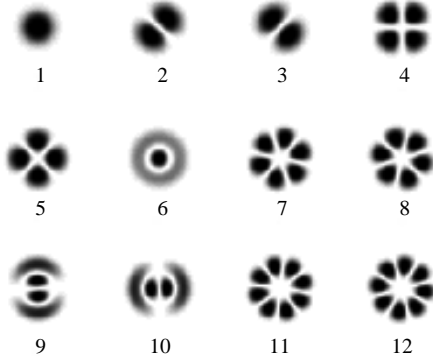
$$g(0, 0) = 2h \ln \left( \frac{\sqrt{2} + 1}{\sqrt{2} - 1} \right). \quad (2.217)$$

**Single quantum dot** The confining potential we proposed in Eq. (2.211) has two interesting limits. First, for  $R \rightarrow \infty$  we obtain the harmonic potential

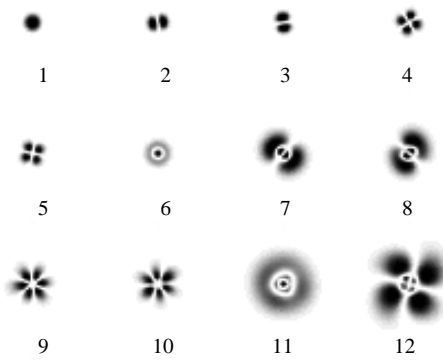
$$V_e(r) = -\frac{Q}{4\pi} + \frac{1}{2} \omega^2 r^2, \quad (2.218)$$

where  $\omega$  is related to the background charge density  $\rho = Q/\pi R^2$  via

$$\omega = \sqrt{\frac{\rho}{2}}. \quad (2.219)$$



**Figure 2.21:** The first 12 orbitals for a parabolic confining potential.



**Figure 2.22:** The first 12 orbitals for our potential with  $R = 0.8$  and  $Q = 24$ .

In this limit we return to the parabolic confining potential that has been studied extensively. Second, for  $R \rightarrow 0$  the potential reduces to

$$V_e(r) = -\frac{Q}{r}, \quad (2.220)$$

which is the potential for two dimensional atoms [Pyy91, Neg96].

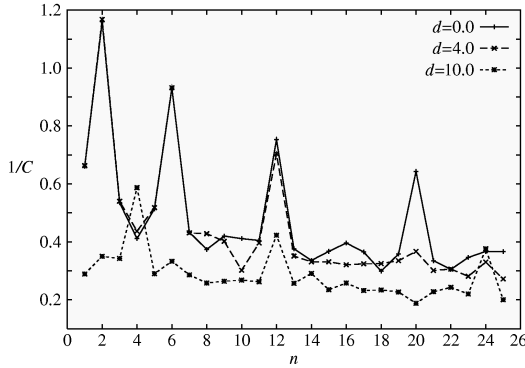
These two limits have a different shell structure. For the parabolic potential the shell will be filled with 2, 6, 12, 20, ... electrons, whereas for the two dimensional atoms these numbers are 2, 8, 14, 24, .... This shell structure can be visualized using the capacitance

$$C(n) = \frac{1}{\mu(n+1) - \mu(n)}, \quad (2.221)$$

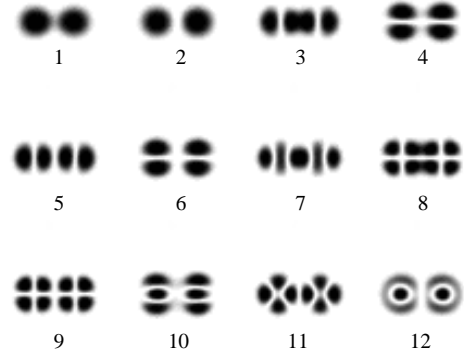
where  $n$  is the number of electrons in the quantum dot and  $\mu$  is the chemical potential  $\mu(n) = E(n) - E(n-1)$ . Filling of a shell will result in a peak in the inverse of this capacitance. This calculated inverse capacitance has been plotted in Figure 2.20 the parabolic potential ( $R \rightarrow \infty$ ) and for three different values of  $R$ , each a factor 5 apart, namely 20, 4 and 0.8. For the choice  $Q = 24$ , these radii correspond, according to Eq. (2.219), with strength  $\omega$  of the parabolic potential near the center of  $1.07 \text{ meV}$ ,  $5.36 \text{ meV}$ , and  $26.78 \text{ meV}$  respectively, which are numbers consistent with experimental values. For the pure parabolic potential the value  $1.07 \text{ meV}$  was used. As can be seen in the figure, for  $R = 20$  the shell filling goes, as expected, similar to the case of the pure parabolic potential. For  $R = 4$  the difference between the higher orbitals is already larger and for  $R = 0.8$  the distinction can be seen by a shift of the peak in the inverse capacitance from  $n = 20$  to  $n = 16$ . For even smaller dot radii the shell filling is expected to shift further to the one for two dimensional atoms.

The orbitals corresponding to the parabolic potential and to the potential with  $R = 0.8$  are plotted in Figures 2.21 and 2.22 respectively, where the scaling in the two figures differs a factor 2. As expected, the innermost orbitals are very similar. The higher orbitals, however, are much wider for the potential with  $R = 0.8$ , since these electrons are much less confined.





**Figure 2.23:** Inverse of the capacitance for different inter-center distances as a function of the number of electrons.



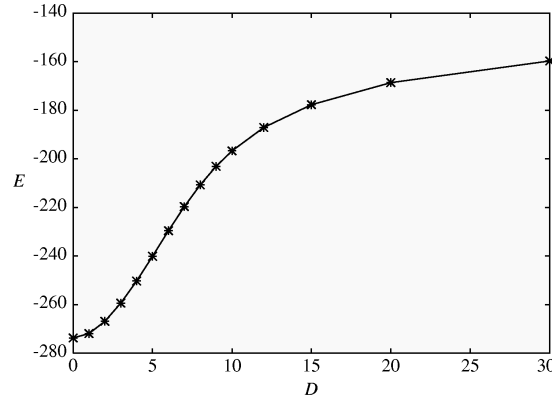
**Figure 2.24:** The first 12 orbitals for two laterally coupled quantum dots at a distance  $d=10.0$  apart.

**Laterally coupled quantum dots** Because of the asymptotic behavior of the potential, where the potential goes to zero for large distances, arrays of quantum dots can be modeled straightforwardly. For example, here we calculate the electronic structure of two laterally coupled quantum dots which form a quantum dot molecule.

We will consider two quantum dots with charge  $Q = 24$  and radius  $R = 4$ , corresponding to a parabolic potential of  $5.36 \text{ meV}$ . These quantum dots are studied as a function of the distance  $d$  between their centers, where the two quantum dots contain 12 electrons each.

For  $d = 0$  the two quantum dots are centered at the same position and will therefore form just one quantum dot with the same radius but twice as high a charge density and thus twice as strong a confinement. For such a quantum dot, the first twelve orbitals will have the form as with the pure parabolic potential, thus orbitals filled at 2, 6, 12 and 20 electrons. This can be seen again using the inverse of the capacitance, which is shown in Figure 2.23. In this same figure can be seen that if the inter-center distance  $d$  increases, quantum dot molecular orbitals come into existence with different shell structure. For  $d = 10.0$  the first three peaks in the inverse of the capacitance are at 4, 12 and 24 respectively. The orbitals for this case are shown in Figure 2.24. In between, for  $d = 4.0$ , both peaks at 20 and 24 can just be discerned. If the distance between the quantum dots increases further and further, the two quantum dots will be separated, resulting in two single quantum dots, which were described in the previous section.

In Figure 2.25 the energy of this system of two quantum dots, with totally 24 electrons, has been plotted as a function of the distance  $d$  between the two centers of the dots. The two limits of the energy as a function of the inter-center distance of the quantum dots, i.e., for  $d = 0$  and  $d \rightarrow \infty$ , can be understood using the 2D pure harmonic oscillator, i.e., without electron-electron interaction. Namely, in both limits only the lower orbitals are filled. These orbitals are very similar to the solution of an harmonic oscillator up to an offset  $-\frac{Q}{R} - \frac{Q}{4\pi}$ , see Eq. (2.211). Such an offset does not change the shape of the eigenfunctions, it just causes an offset with this same value in the energy eigenvalues,



**Figure 2.25:** The energy of the two quantum dot system versus the inter center distance.

which are well-known for the 2D pure harmonic oscillator to be  $m\omega$  with degeneracy  $m$ , for  $m = 1, 2, 3, \dots$ , where each orbital can occupy 2 electrons. Thus, in terms of  $\omega$  the lowest energy eigenvalues of one quantum dot are  $m\omega - \frac{1}{2}(4\pi R + R^2)\omega^2$  with degeneracy  $m$ .

One single quantum dot thus has two contributions to the total energy, in the first place a contribution of the harmonic oscillator  $2 \cdot (1 \cdot 1 + 2 \cdot 2 + 3 \cdot 3)\omega = 28\omega$ , and in the second place the offset, which is for our  $Q$  and  $R$  equal to  $12 \cdot (-8(\pi + 1)\omega^2)$ . Because  $Q = 24$  and  $R = 4$ ,  $\omega = \sqrt{3/(4\pi)}$ . And thus the total energy is equal to  $28\sqrt{3/(4\pi)} - 72(\pi + 1)/\pi$ , which is approximately equal to  $-81$ . Thus two of such quantum dots infinitely far apart yields  $E \approx -162$ , when the electron-electron interaction is neglected.

The other limit is where the two quantum dots are centered at the same place. This yields one quantum dot with the same radius but a twice as high charge, and twice as many, i.e. 24, electrons. Thus, again two contributions to the total energy, namely  $2 \cdot (1 \cdot 1 + 2 \cdot 2 + 3 \cdot 3 + 4 \cdot 4 + 2 \cdot 5)\omega = 80\omega$ , and an offset  $24 \cdot (-8(\pi + 1)\omega^2)$ , where  $\omega = \sqrt{3/(2\pi)}$ . And thus the total energy is equal to  $80\sqrt{3/(2\pi)} - 288(\pi + 1)/\pi$ , which is approximately equal to  $-324$ .

**Conclusion** With the addition of specific 2D correlation and exchange potentials and a newly developed two-dimensional version of Chelikowsky's direct integration method, our wavelet based electronic structure calculation code is also able to deal with two dimensional problems. This has been demonstrated for 2D quantum dots as the expected electronic structure was calculated for quantum dots for a parabolic confining potential. However, as such an infinite potential is unphysical also a new potential has been proposed, which consists of the combination of the parabolic potential of a homogeneous density distribution within a 2D jellium with the Coulombic potential outside that jellium. Using this description of quantum dots, not only the shell filling of a single quantum dot was studied, but also that of a quantum dot molecule consisting of two laterally coupled quantum dots. Varying the inter center distance within such a quantum dot molecule yielded energies in agreement with analytical results in the limits of zero and infinitely large inter center separation. Thus, in the same way larger arrays of

quantum dots could be simulated as well.

## 2.7 Conclusion

We have proved that our electronic structure calculation framework, based on various wavelet types, preconditioners, and pruning by introducing extra levels of fine wavelets only in regions where necessary, works and is accurate. This electronic structure framework is applicable for 1D, 2D and 3D problems.

First this has been applied on analytically solvable case studies: the harmonic oscillator, the Poisson equation, and a potential well. With these case studies the solution approach, including the minimization approach, the effects of different types and orders of wavelets, different preconditioners and the advantage of pruning was shown in relation with accuracy and computational cost. By comparison of the numerical and analytical solutions it is shown that the method yields accurate results and that the fastest convergence to the analytical results is obtained using the interpolant 6 in combination with the MIC preconditioner, which are thus used for the 2D and 3D problems. It has also been shown that problems where high resolution is needed locally, e.g. in case of discontinuities, can be solved efficiently and accurately by means of pruning, because extra resolution can be added locally where necessary instead of over the whole domain. After that the code was successfully used to obtain interesting results in 2D quantum dots and small 3D molecular systems. The effect of different choices for pseudopotentials and exchange-correlation potentials were studied on properties such as ionization potentials, electron affinities, excitation energies, binding energies, bond frequencies, force constants, and bond lengths of atoms and small molecules. These results compare well with experimental data and electronic structure calculations with other basis sets, proving the applicability of the code for this class of problems.

Also, we have studied the electronic states of quantum dots using a new potential derived from the description of a quantum dot as a 2D jellium. Specific to solve such 2D problems is a two-dimensional version of Chelikowsky's direct integration method which has been developed to evaluate the Hartree potential. For single quantum dots the orbitals have been calculated and the filling of the shells has been compared with the case of a parabolic model potential. By laterally coupling quantum dots, what can be done using our potential straightforwardly, quantum dot molecules have been modeled. The shell fillings for two laterally coupled quantum dots have been calculated as a function of the inter center distance. The corresponding energies agree with analytical results in the limits of zero and infinitely large inter center separation. This thus enables to study larger arrays of quantum dots as well.

With this framework we are able to include high resolution for local detail in a very cost effective way. However, for the larger system sizes required in chemical and biological systems too many basis functions are needed, resulting in too high computational cost. We estimate that parallelizing the code will not bring enough improvement to simulate

the larger systems we are interested in. That is the reason that instead of parallelizing the code we extended the framework with molecular dynamics with empirical potentials, notwithstanding the applicability of our framework for small systems where locally very high resolution is needed.



# 3

## Coarse grained molecular dynamics

---

**I**n coarse grained molecular dynamics simulations the average behavior of several atoms is combined into one single particle. Because this reduces the number of particles and increases the time step size that can be made compared to fully atomistic molecular dynamics, as the fastest motion of light atoms like hydrogens is removed, the use of coarse grained molecular dynamics simulations allows to study system sizes and time intervals that are still out of reach for fully atomistic simulations. Here, a coarse grained model is presented for phospholipids in water. Using this model, the dynamics of membrane and vesicle formation are studied as well as the process of vesicle fusion.

Part of this chapter is described in:

A.J. Markvoort, K. Pieterse, M.N. Steijaert, P. Spijker and P.A.J. Hilbers, **The bilayer-vesicle transition is entropy driven**, J. Phys. Chem. B, **109**, 22649-22654 (2005).

---

## 3.1 Introduction

In the previous chapter we described a method to speedup quantum mechanical calculations. However, such ab-initio methods can only be applied to systems with a few atoms, especially if we are interested in dynamics. This dynamics can be studied using molecular dynamics simulations, where the time evolution of a set of interacting atoms is followed by integrating their equations of motion. The force the atoms exert on each other can be calculated from the electronic structure, but instead of calculating the forces on the particles from first principles, empirical potentials can be used as well. In this field of empirical potentials there is another distinction between fully atomistic models and coarse grained models. In fully atomistic simulations, the focus for biological systems is mainly on fine tuning the parameters and employing larger and larger computers to simulate larger systems. However, system sizes and time intervals needed to study most biologically interesting phenomena are still out of reach. In this chapter, we therefore propose the use of coarse grained molecular dynamics simulations for biological systems. In such a coarse grained method, not all atoms are modeled in atomistic detail, but some atoms are considered together in one particle. This obviously results in a reduction of the number of particles, but it also has the advantage that larger time steps can be made, as the fastest motion of light atoms like hydrogen is removed, such that larger systems can be followed during larger time intervals. To demonstrate the applicability we focus on membranes. More specifically, we study the aggregation of lipids into bilayers and the spontaneous formation of vesicles.

**Membranes** Membranes [Alb89, Lip95, Nag00] are important in all organisms as the plasma membrane of cells forms the protective barrier between the cytosol and the cell exterior. This allows for the maintenance of the essential differences between the cytosol and the exterior. In eucaryotic cells, membranes also separate different regions within the cell such that these regions can perform specific tasks. Besides this function as a partitioner, membranes are important in a number of processes including transport, cell-cell signaling and reproduction. The contemporary view on the structure of membranes is based on the Fluid Mosaic Model, as suggested by Singer and Nicolson in 1972 [Sin72]. In this model a membrane consists of a fluid bilayer of a mixture of lipids in which proteins float and which is about 5 nm thick. The membrane is called fluid as the lipids and proteins can move in the plane of the membrane by means of lateral and rotational diffusion. Lipids can also ‘flip-flop’ from one monolayer to the other, although much more rarely, allowing the two halves of the membrane to have different compositions.

Membrane bound proteins can have many functions among which are transportation, catalysis, and signaling. Depending on their function these proteins can be either fully embedded in the bilayer, partly embedded or only associated with the bilayer surface. The lipid bilayer however provides the basic structure of the membrane and serves as a relatively impermeable barrier to most water-soluble molecules. Because the body of

the membrane is formed by the lipid bilayer we concentrate on these lipids.

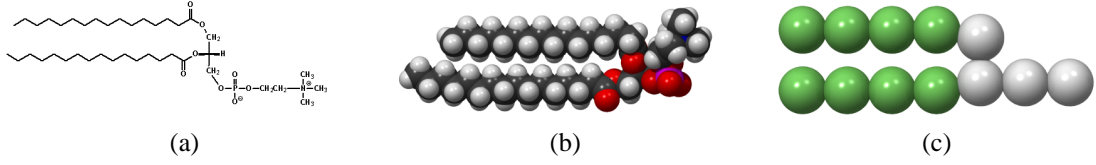
**Membrane lipids** All lipid molecules in cell membranes are amphiphilic, i.e., they have a hydrophilic (water-loving, polar) head group and a hydrophobic (water-fearing, non-polar) tail. The most abundant membrane lipids are phospholipids, having a polar head group and two hydrophobic hydrocarbon tails. The head groups can differ and the tails can differ in length and degree of saturation. These differences can be of importance as they influence the ability of phospholipid molecules to pack, and as such the fluidity of the membrane, but the general structure of all lipids is the same.

**Aggregation of amphiphiles** Lipids form membranes because, due to their amphiphilic nature, lipids have a tendency to aggregate in a watery environment. The repulsive interactions between the apolar parts of the molecules with the polar parts of the molecules as well as with the polar water molecules, force the system in a configuration where the apolar tails are isolated from the polar environment as much as possible. Depending on the type of amphiphile (e.g., geometry, distribution of polar and apolar regions, and electrical charges) and system constants (e.g., amphiphile concentration and temperature), aggregates of different size and geometry are formed [Isr91]. At low concentrations most amphiphiles are present as isolated monomers. The concentration of monomers rises with the total amphiphile concentration, until the critical micelle concentration (CMC) [Isr91] is reached. Above this CMC the number of monomers is constant whereas the number of aggregates increases. These aggregates can for example be spherical, rod-like or disc-like micelles, or for very high concentrations ordered continuous structures. Cylindrical amphiphiles like lipids can also form bilayers and vesicles. Although the steady state aggregation behavior of common lipids is known, the complete dynamics of the formation of aggregates is not yet completely understood. Several mechanisms have been suggested [Shi02b, Len02], but because of the combination of small length scale and short time scale of the dynamics, it is hard to find an experimental setup to study this. However, coarse grained molecular dynamics provides a good way to simulate this dynamics of amphiphiles.

## 3.2 Coarse grained lipid model

Biological membranes can consist of a variety of lipids. All these lipids are amphiphilic in nature, having a hydrophilic head group and one or two hydrophobic tails. The most important type of membrane lipids are the phospholipids, among which the group of glycerophospholipids is the most common. To describe the interactions between phospholipids in water, we use a coarse grained (CG) model in combination with a simplified force field. The CG model differs from the more common atomistic models in the fact that the molecules consist of so-called coarse grained particles, which combine the average behavior of several atoms in one single particle. The advantage of coarse graining





**Figure 3.1:** Comparison of the DPPC with our coarse grained model. **(a)** chemical structure of DPPC. **(b)** and **(c)** van der Waals representations of atomistic DPPC and the coarse grained lipid molecule respectively.

is that the resulting system consists of fewer particles and that larger time steps can be made. This enables the study of larger systems over larger time intervals, i.e., system sizes and time intervals that are out of reach of fully atomistic simulations. Our coarse grained lipids are based on lipids of the glycerophospholipids class, dipalmitoylphosphatidylcholine (DPPC) being a typical example. In the simulation sections, we compare our model with DPPC because of the wide availability of experimental data for this type of lipid and because this lipid captures all essentials from the neutral lipids. The chemical structure and the van der Waals representation of this lipid are depicted in Figures 3.1a and 3.1b respectively.

In our CG model we use the coarse grained molecule as shown in part c of the same figure to model the lipid. In this model two types of coarse grained particles are used to describe the chemically relevant groups in the phospholipid. The apolar tails of the phospholipids are represented by T particles, whereas the polar head groups are represented by H particles. The molecule consists of two chains of four T particles representing the two saturated fatty acid tails of DPPC and four H particles representing the choline, phosphate and glycerol backbone of the DPPC head group. This coarse grained molecule is referred to in the remainder of the text as the lipid. To simulate phospholipid–water mixtures, a third particle type W is present in our CG model which describes the solvent (water).

### 3.2.1 CG model potentials

The force between two particles is governed by the gradient of the potentials between these particles. In our force field three potentials are defined, namely a bonded, a non-bonded, and a bending potential. The first one is used for particles that share a bond, while the second is used for the interactions between all other pairs of particles. In first instance only these two potentials are used. Later on, the bending potential is used to study the effect of the rigidity of the lipid on the bilayer and vesicle formation. All three potentials are shown in Figure 3.2.

**Bonded potential** Bonded interactions are described by vibrational movements, as in a mass–spring system. These movements are harmonic and, therefore, are described by the harmonic potential

$$V_{\text{Bond}}(r_{ij}) = k_{ij} (r_{ij} - r_{0,ij})^2, \quad (3.1)$$

where  $k_{ij}$  is the binding constant between bonded particle  $i$  and  $j$ ,  $r_{0,ij}$  is the equilibrium bond length, and  $r_{ij} = |\vec{x}_j - \vec{x}_i|$  is the scalar distance between particle  $i$  and particle  $j$ , where  $\vec{x}_i$  is the position vector of atom  $i$  and  $\vec{x}_j$  similarly for particle  $j$ .

**Bending potential** The tails of lipid molecules are rather flexible, however, the *anti* conformation is energetically favorable for hydrocarbon chains. To mimic this behavior, some rigidity can be incorporated into our lipid tails by means of a bending potential between T particles. These interactions depend on the angle  $\theta_{ijk}$  between two bonds that share a common particle:

$$V_{\text{Bend}}(\theta_{ijk}) = k_{ijk} (\cos(\theta_{ijk}) - \cos(\theta_{0,ijk}))^2, \quad (3.2)$$

where both particles  $i$  and  $j$ , and  $j$  and  $k$  share a bond. In this potential,  $k_{ijk}$  is the bending force constant and  $\theta_{0,ijk}$  the equilibrium angle.

Dihedral potentials, which are often used in fully atomistic simulations, are omitted from our model because the bending potential alone suffices to introduce the desired rigidity in our lipid tails.

**Non-bonded potential** The non-bonded interactions are derived from the Lennard-Jones (LJ) potential

$$V_{\text{LJ}}(r_{ij}) = 4\epsilon_{ij} \left( \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right), \quad (3.3)$$

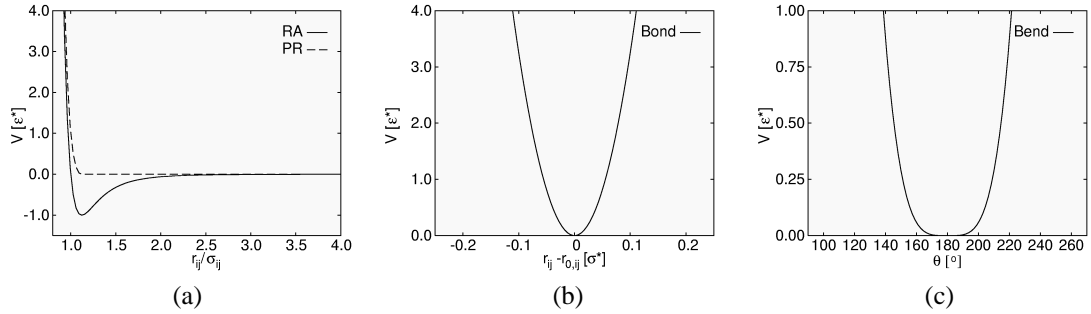
where  $\epsilon_{ij}$  is the characteristic energy in the pair potential,  $\sigma_{ij}$  is the collision diameter of the pair, and the scalar distance between the particles,  $r_{ij}$ , is defined similarly as for the harmonic potential. Formally, in the Lennard-Jones potential the interactions between all non-bonded particle pairs have to be calculated, but since this potential vanishes at larger  $r_{ij}$ , only the interactions with particles within a certain cut-off radius need to be calculated. Therefore, in our force field a so-called truncated shifted Lennard-Jones (tsLJ) potential is defined, which is derived from the standard Lennard-Jones potential as

$$V_{\text{tsLJ}}(r_{ij}) = \begin{cases} V_{\text{LJ}}(r_{ij}) - V_{\text{LJ}}(r_{c,ij}) & \text{if } r_{ij} \leq r_{c,ij} \\ 0 & \text{if } r_{ij} > r_{c,ij} \end{cases} \quad (3.4)$$

by cutting it at the cut-off radius  $r_{c,ij}$  and shifting it upward to avoid discontinuities in the potential.

Two different cut-off radii are used in our model. The first is  $r_{c,ij} = 2.5\sigma_{ij}$ . The resulting potential resembles the full Lennard-Jones potential closely since the Lennard-Jones potential reduces fast to zero at larger  $r_{ij}$ , but it has the advantage that with the truncated potential only the interactions with particles within the cut-off radius have to be taken into account instead of all pair interactions. Because of this, the computational cost reduces from quadratic to linear in the number of particles. This potential will be denoted further as the RA potential.

In phospholipid–water mixtures the water molecules and phospholipid head groups are polar (hydrophilic), while the phospholipid tails are apolar (hydrophobic). Polar and



**Figure 3.2:** The three potentials in our coarse grained model. **(a)** The truncated shifted Lennard-Jones potential for two different cut-off radii. **(b)** The bonded potential (for  $k_{ij} = 325 \epsilon^* \sigma^{*-2}$ ). **(c)** The bending potential (for  $\theta_0 = 180^\circ$  and  $k_{ijk} = 16 \epsilon^*$ ).

apolar particles will normally phase separate (for example water and oil) and our tsLJ potential has to be modified for the polar–apolar interactions to describe this behavior correctly. The latter is achieved by setting the cut-off radius for polar–apolar interaction to  $2^{1/6} \sigma_{ij}$ , thereby effectively removing the attractive part of the tsLJ potential (Figure 3.2a). The resulting potential is purely repulsive and will thus be denoted as the PR potential.

Our CG model only uses neutral particles and, therefore, a potential dealing with electrostatics is omitted from the force field. Although this confines our CG model to the description of neutral phospholipids, the gain in computational speed is considerable due to the long range nature of electrostatic interactions. The shorter ranged electrostatic effect of charge distributions within our CG particles (e.g. dipoles) is incorporated in our non-bonded truncated shifted Lennard-Jones potentials.

### 3.2.2 CG model parameters

The parameters used in our CG model are expressed in reduced units. These reduced units, with values typically around one, improve the numerical stability of the simulations and facilitate error estimation. Our system consists of the following reduced units: the unit for length  $\sigma^*$ ; the unit for mass  $m^*$ ; and the unit for energy  $\epsilon^*$ . All other reduced units (like the units for temperature  $T^*$ , pressure  $P^*$ , and time  $\tau$ ) can be derived out of these choices [Fre02, All87].

**Lennard-Jones parameters** As a basis for our parameters the T particle was taken, which represents four methylene groups (e.g.  $-(CH_2)_4-$ ). The choice for four methylene groups in one coarse grained particle has been made in a trade-off between speed-up and accuracy. A smaller unit could have been taken, but then the gain of coarse graining would have been smaller. On the other hand, a larger unit could have been taken, but then the representation with a spherical coarse grained particle becomes less good. With our choice, the speed-up compared with fully atomistic simulations is over 100, enabling the simulation of processes at a completely different time scale, while lipids can still be

represented naturally.

By definition all parameters describing the T particles are set to one, hence  $m_T = 1 m^*$ ,  $\sigma_{TT} = 1 \sigma^*$  and  $\epsilon_{TT} = 1 \epsilon^*$ . In order to investigate the phospholipids under realistic conditions and allow comparison of our simulations with experimental data, conversion of the reduced units to common units is a prerequisite. Vice versa, conversion of common units to our reduced units enables the determination of our CG parameters from experimental and atomistic simulation data.

The unified atomic mass of  $1 m^*$  can be deduced from the mass of a T particle. Since one T particle resembles four methylene groups  $m_T = 1 m^* \approx 56.1 \text{ amu}$ . The energy associated with  $\epsilon^*$  can be derived from the reduced temperature  $T^*$  via  $1 \epsilon^* = 1 \frac{T^*}{k_B}$ , where  $k_B$  is the Boltzmann constant. This reduced temperature was determined by comparing the melting temperature ( $T_m$ ) found in a simulation containing 1372 T particles ( $T_{m,T} = 0.57 T^*$ ) with the melting temperature of butane ( $T_{m,\text{butane}} = 135 \text{ K}$ ). Considering the close chemical resemblance of  $-(CH_2)_4-$  with butane, one can state that  $T_{m,T} \approx T_{m,\text{butane}}$  and, therefore,  $1 T^* \approx 236 \text{ K}$ , subsequently leading to  $1 \epsilon^* \approx 0.47 \text{ kcal mol}^{-1}$ . To get an estimation of the reduced length, a simulation containing 1000  $T_3$  (i.e., a coarse grained T-T-T molecule) and another with 600  $T_4$  oligomers was performed at  $1.24 T^*$ . The particle densities ( $d$ ) obtained from these two simulations ( $d_{T_3} = 0.70 \sigma^{*-3}$  and  $d_{T_4} = 0.75 \sigma^{*-3}$ ) were compared with the density of dodecane ( $C_{12}H_{26}$ ,  $\rho_{293 \text{ K}} = 0.749 \text{ g cm}^{-3}$ ) and hexadecane ( $C_{16}H_{34}$ ,  $\rho_{293 \text{ K}} = 0.773 \text{ g cm}^{-3}$ ), respectively, resulting in  $1 \sigma^* \approx 0.45 \text{ nm}$ .

A W particle represents four water molecules and, hence, its mass is set at  $m_W = 72.1 \text{ amu} \approx 1.284 m^*$ . Since the melting temperature of our CG particles scales linearly with their characteristic energy  $\epsilon_{ij}$  and  $T_{m,\text{water}} = 273 \text{ K}$ , one can deduce that  $\epsilon_{WW} \approx 2.0 \epsilon^*$ . Using these parameters, a simulation of W particles at  $1.3 T^*$  yielded a particle density of  $d_W = 0.813 \sigma^{*-3}$ . Taking into account that  $\rho_{\text{water},307 \text{ K}} = 0.993 \text{ g cm}^{-3}$ , this results in  $\sigma_{WW} \approx 1.024 \sigma^*$ .

To keep the model as general as possible, the polar head group of the lipids is represented by four identical H particles. The mass and collision diameter of such a H particle are given the same values as those of the T particles, while the characteristic energy is set at  $\epsilon_{HH} = 2.0 \epsilon^*$ , equal to that of the W particles.

**Cross-type Lennard-Jones parameters** Besides the interactions between non-bonded particles of the same type, also cross-type interactions are present. To prevent preferential associations from inducing artifacts in our simulations, the characteristic energies of all polar–polar cross-type interactions have been taken equal and, therefore,  $\epsilon_{HW} = 2.0 \epsilon^*$ . For the interaction between polar and apolar particles the PR potential is used. The characteristic energy of these cross-type interactions is set to one ( $\epsilon_{HT} = \epsilon_{WT} = 1.0 \epsilon^*$ ).

The collision diameter for all cross-type interactions can simply be calculated using

$$\sigma_{ij} = \frac{\sigma_{ii} + \sigma_{jj}}{2}. \quad (3.5)$$

**Harmonic bond parameters** To determine the parameters involved in the harmonic bond potential, atomistic molecular dynamics simulations of alkanes were performed using MMTK with the Amber force field [Hin00]. An estimate for the equilibrium bond length was obtained by simulating eicosane ( $C_{20}H_{42}$ ) and representing the eicosane molecules as  $T_5$  oligomers, i.e., dividing the eicosane molecules in five equal parts and using their centers of mass as the coarse grained particle position vectors. By averaging the scalar distance between adjacent centers of mass over several configurations, an equilibrium bond length of  $r_{0,TT} = 0.4728 \text{ nm} \approx 1.051 \sigma^*$  was found for the coarse grained particles.

The binding constant can be calculated from the harmonic frequency  $\omega_{ij}$  using  $k_{ij} = \mu_{ij} \omega_{ij}^2$ , with reduced mass  $\mu_{ij} = \frac{m_i m_j}{m_i + m_j}$ . To obtain the harmonic frequency of the coarse grained particles, octane ( $C_8H_{18}$ ) was simulated and represented as a  $T_2$  oligomer. Fourier analysis of the evolution of the scalar distance between the centers of mass in time, yielded a harmonic frequency of  $7.5 \text{ ps}^{-1}$ . The latter results in a binding constant of  $k_{TT} \approx 325 \epsilon^* \sigma^{*-2}$ .

Since the size and mass of the H particles are equal to those of the T particles, their corresponding parameters as well as the cross-type parameters are assigned equal values ( $r_{0,HH} = r_{0,HT} = r_{0,TT}$  and  $k_{HH} = k_{HT} = k_{TT}$ ).

**Bending parameters** The parameters involved in the bending potential are obtained from the same atomistic molecular dynamics simulations of eicosane molecules. Using the same five centers of mass per eicosane molecule as used to calculate the binding force constant, the distribution of angles between adjacent centers of mass can be determined. This resulted in a wide distribution of angles with an average angle of  $148^\circ$ . Since the lowest energy conformation of our lipid tails should be a straight chain, an equilibrium angle of  $\theta_0 = 180^\circ$  is used for our coarse grained tails. Implementing this choice in  $T_5$  oligomers, a similar angle distribution as for the fully atomistic eicosane molecules is obtained using  $k_{ijk} = 16 \epsilon^*$  for the bending constant.

The atomistic eicosane simulation also showed that for the dihedral angles between four adjacent centers of mass indeed hardly any preferential angle is present, thus legitimizing our choice to omit dihedrals from our coarse grained model.

### 3.3 Simulations of bilayer and vesicle formation

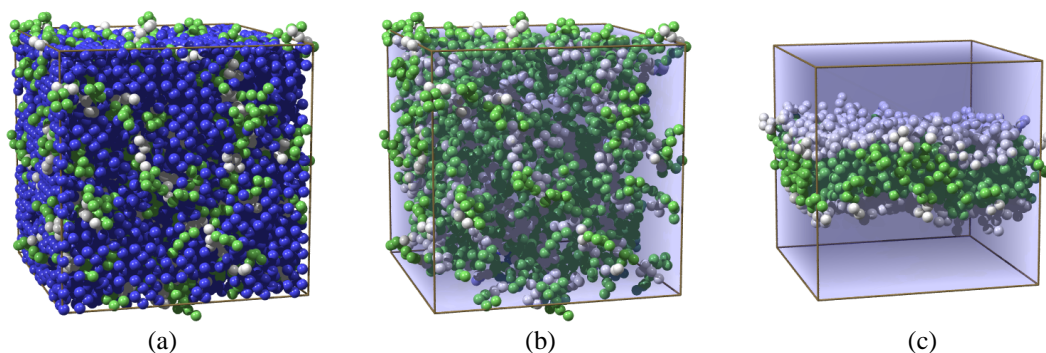
The molecular dynamics code that we use for our simulations is our in-house developed code *PumMa* which is partly based on the code presented in Ref. [Ess93]. *PumMa* is a relatively small and very efficient parallel molecular dynamics code that we could extend rather easily to our needs and that scales well on our Beowulf cluster. The use of other existing publicly available codes has been considered as well, but this has been refrained from because it was either too elaborate to implement our force field or the

performance was too low [vdH04]. A toolkit (*PumMaTK*) has been developed to easily create new start configurations and to visualize and analyze simulation results.

### 3.3.1 Spontaneous formation of a bilayer

The first test of our coarse grained model is whether our lipid molecules aggregate, ultimately forming bilayers. In first instance, lipids without the bending potential are used. To investigate the aggregation, 256 lipid molecules have been placed randomly in a cubic simulation box with length  $23.3\sigma^*$ . The remaining free space was subsequently filled with 6471 water particles. This configuration is shown in Figure 3.3a. The same configuration is shown in part b of the same figure, but now the water particles have been drawn transparently to show the lipids more clearly. Using this initial configuration, a constant temperature and constant pressure (NPT) simulation is performed existing of 500000 iterations of size  $0.005\tau$ . During this simulation the temperature and pressure were kept at  $T = 1.3T^*$  and  $P = 0.00283P^*$ , respectively (atmospheric conditions). As can be seen from the final configuration, which is shown in Figure 3.3c, the lipids aggregate and organize into a bilayer. For such a bilayer, the repulsive interactions between the apolar T particles and the polar H and W, are minimized as much as possible.

It is important to note that the periodic boundary conditions can influence the structure that is formed. Bilayers generally form perpendicular to one of the axes of the orthorhombic simulation box such that the bilayer is periodic. If the area of the bilayer is larger than the available area in the simulation box, the bilayer becomes curved. On the other hand, if the bilayer area is smaller than the available area, a bilayer with a hole or a stretched and thinner bilayer can be formed. To prevent this, pressure scaling is applied independently in all three spatial directions such that no unnatural stress is introduced in the bilayer.

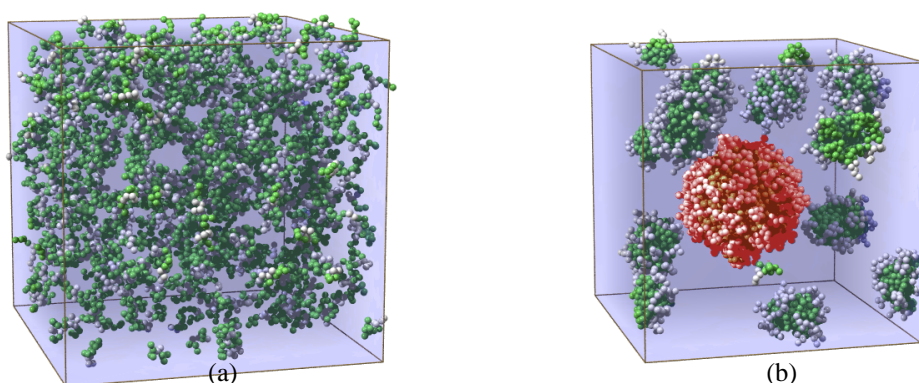


**Figure 3.3:** Spontaneous formation of a bilayer. System consists of 256 lipid and 6471 water molecules. **(a)** Initial random configuration. **(b)** The same initial configuration with the water particles drawn transparently. **(c)** Final configuration after 500000 iterations.

### 3.3.2 Spontaneous formation of vesicles

As has been stated above, the periodic boundary conditions influence the structures that are formed. In the previous simulation this effect was minimized as much as possible by allowing independent pressure coupling. A better way to circumvent the influence of the boundary conditions is by simulating a larger box, such that no periodic structures are formed.

In the same way as for the previous simulation, a larger initial configuration has been made by randomly placing the lipid molecules and filling the remaining space with water particles. This larger simulation box contains 512 lipid molecules and 66026 water particles and is shown in Figure 3.4a. During a 2.5 million iterations simulation, the lipids firstly aggregate into micelles. Subsequently some of the micelles merge into a larger aggregate that is disc-like, like a circular bilayer. Finally this aggregate transforms into a vesicle. This final configuration containing the vesicle and some micelles is shown in part b of the same figure.

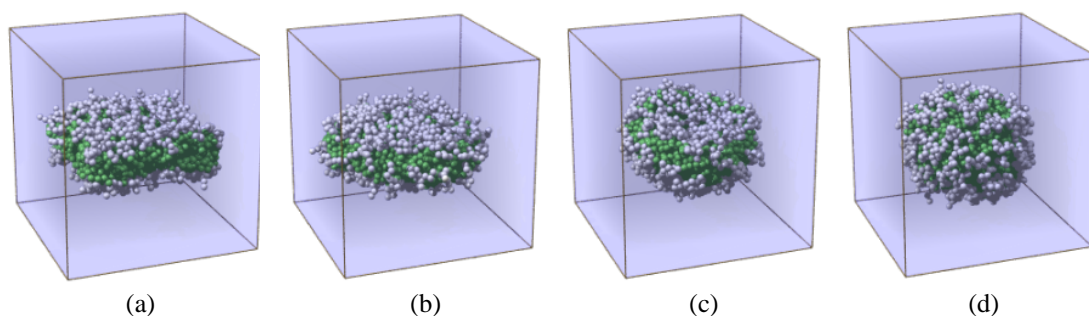


**Figure 3.4:** Starting from a larger simulation box randomly filled with lipids (a), a mixture of micelles and a vesicle is formed (b).

### 3.3.3 Vesicle formation from bilayers

To study the process of vesicle formation more closely, the bilayer that was formed in Section 3.3.1 was placed in a larger box full of water particles as shown in Figure 3.5a. This new system is now taken as the starting configuration for a new simulation. Because the bilayer is now no longer periodic, water borders the hydrophobic tails. In the initial phase of the MD simulation the bilayer tries to minimize this contact. This is done by shielding the tails with some head groups and by minimizing the edge by making the bilayer circular as shown in part b of the figure. Subsequently, the bilayer starts to curl as can be seen in part c of the figure. In this way, the edge is minimized further at the expense of the bilayer planarity. This curling continuous, minimizing the





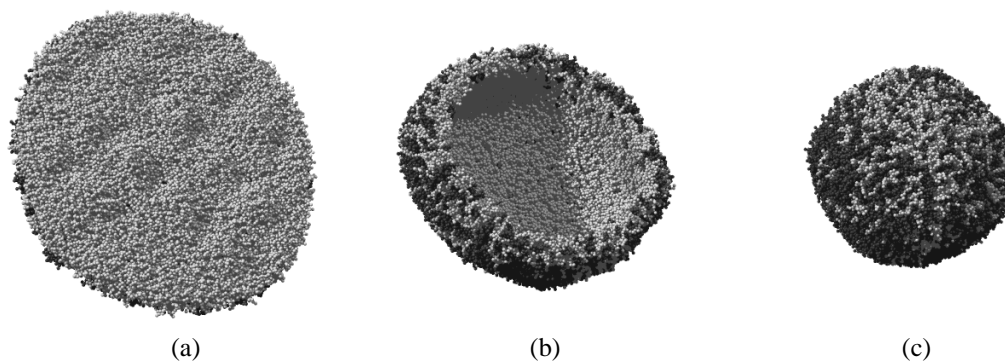
**Figure 3.5:** The bilayer to vesicle transition. The edge of the initially square bilayer (a) is minimized first by making the bilayer circular (b) and subsequently by curling (c) the bilayer until it closes, resulting in a vesicle (d).

edge further and further, until the structure closes. At this stage which is shown in Figure 3.5d, the edge has been removed completely and a vesicle has been formed. In this case the bilayer has enclosed 121 of the water particles which now form the vesicle interior.

### 3.3.4 Formation of a large vesicle

In our simulations so far only small vesicles have been formed. As soon as the bilayer has reached a critical size, it starts to transform into a vesicle. The idea is that larger vesicles could be made by either fusing small vesicles or by starting from a large bilayer. Such a large bilayer can be constructed by placing multiple images of a small bilayer next to each other. By using  $6 \times 6$  images of a small bilayer of width  $23.3\sigma^*$ , a large square bilayer is formed with a width of  $139.8\sigma^*$ . From this large square bilayer, a circular bilayer has been cut which has a diameter of  $120.0\sigma^*$  in order to have a bilayer large enough to form a vesicle with a radius of approximately  $30.0\sigma^*$ . This bilayer consists of 4852 lipid molecules and has been placed in a box containing an additional 522256 water particles.

A MD simulation of in total 2000000 iterations with a time step size of  $0.01\tau$  has been



**Figure 3.6:** Similarly to a small bilayer, a large bilayer curls to a vesicle. The particles are colored depending on the side of the bilayer they were initially situated.



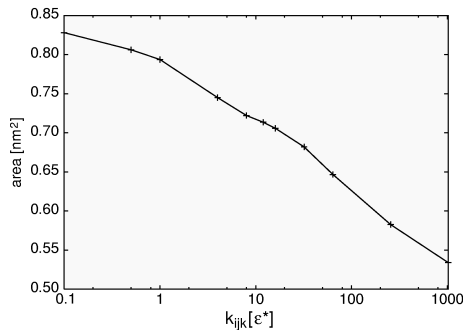
performed in approximately 440 hours on 48 AMD Athlon 1800+ CPU's. In Figure 3.6 some snapshots of the simulation are shown. During the first 400 000 iterations the membrane undulates a little. Then the membrane starts curling. This curling continues gradually until a sphere has come to existence with a small gap. Around iteration 1 700 000 the gap closes, forming the vesicle. During the last 300 000 iterations this vesicle remains stable.

The process of vesicle formation for this large bilayer seems to be equal to that of the small bilayer, but the larger bilayer allows for a better possibility to study this process [Spi03].

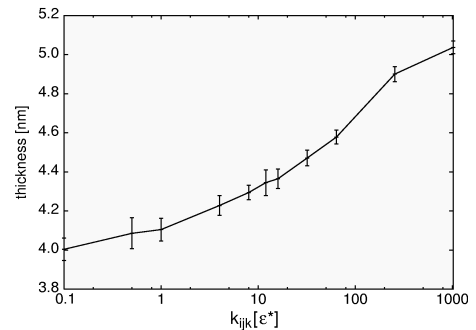
### 3.3.5 Lipid rigidity dependence on bilayers

In all simulations in the previous sections no bending restrictions were put on the lipids. Thus even with such a simple model the formation of a bilayer and vesicles can be described. However, because of the lack of bending restrictions, the lipid molecules are very flexible and the resulting bilayer is too thin compared to real bilayers and the area per lipid is too large. To study the dependence of the bilayer thickness and the area per lipid on the lipid stiffness, bilayers have been formed in simulations for lipid molecules with various bending force constants. The average area per head group as a function of the bending force constant has been shown in Figure 3.7. Furthermore, the bilayer thickness has been shown in Figure 3.8 as a function of the bending force constant as well. This thickness has been defined as the width of the slice, centered around the bilayer center of mass, containing 95% of the bilayer mass. In Figure 3.9 the resulting bilayers are shown and compared with a fully atomistic DPPC bilayer.

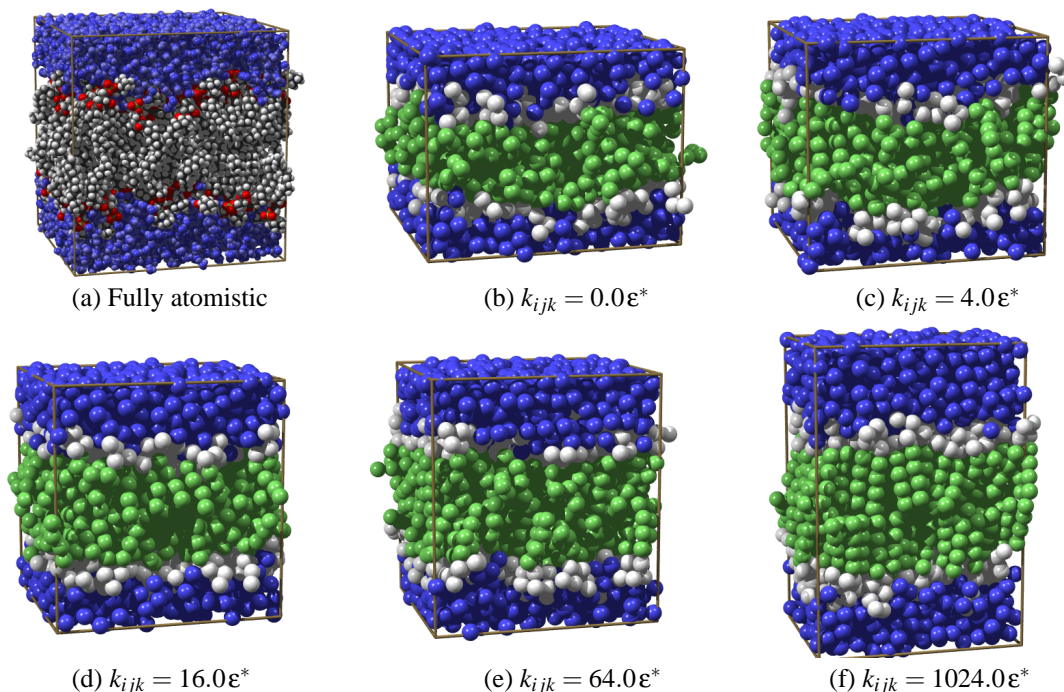
These results can be compared with experimental and fully atomistic simulation results. According to different experimental studies the average area per lipid for a DPPC bilayer is  $0.62 \pm 0.02 \text{ nm}^2$  [Tie97]. The thickness has also been measured experimentally with scattering experiments, but a comparison is hard to make because it is not precisely defined where the bilayer starts and ends. However, our 95% definition can be



**Figure 3.7:** The average area per head group [ $\text{nm}^2$ ] as a function of the bending force constant  $k_{ijk}$ .



**Figure 3.8:** The thickness [nm] of the slice containing 95% of the total bilayer mass as a function of the bending force constant  $k_{ijk}$ .



**Figure 3.9:** Comparison of the coarse grained bilayers for different bending force constants with the fully atomistic bilayer. Each coarse grained simulation consists of 128 lipid and 1165 water molecules, whereas the fully atomistic simulations contains the corresponding 128 DPPC and 4659 water molecules.

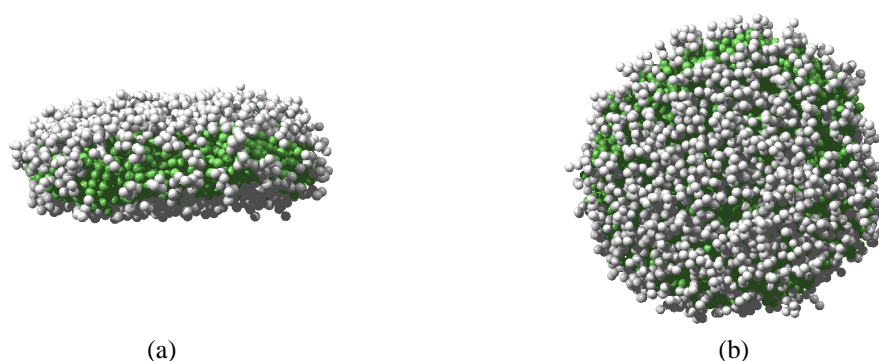
compared with the same definition in fully atomistic simulations, which yield a thickness of 4.23 nm.

From the comparison of the area per lipid, the comparison of the bilayer thickness as well as from the visualizations, it is clear that a bending constant of approximately 16 yields the results most comparable to fully atomistic simulations and experiments. This value also matches with the bending force constant that we estimated from the eicosane simulations.

Thus, with the addition of bending to our lipid model, a still very simple but good model is obtained. In this respect it is important to note that all parameters were not fitted for the bilayer. Even though all our parameters were not especially fitted for the lipids in bilayers, but for pure water and pure alkanes instead, our simple coarse grained model yields good results for lipid bilayers.

### 3.3.6 Lipid rigidity dependence on vesicle formation

In all our simulations so far small bilayers already form vesicles. Even vesicles with a diameter of 10 nm are formed. Such small lipid vesicles are not known to exist in nature. The flexibility of our coarse grained lipid molecules may contribute to this. To test in which sense the rigidity of the lipids influences the vesicle sizes, simulations with the

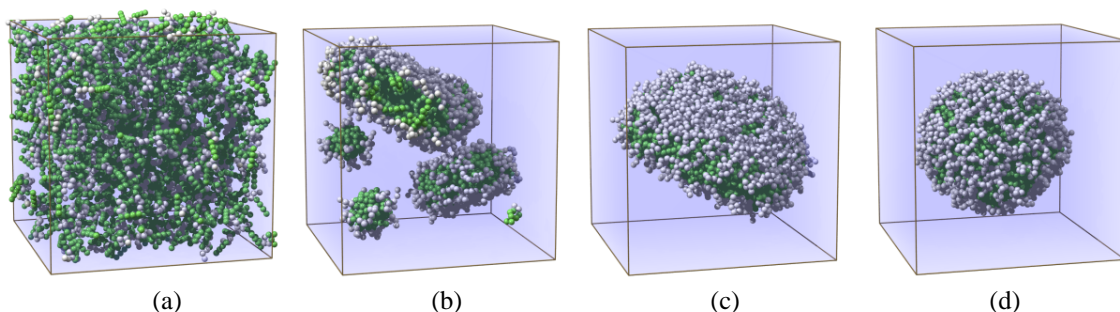


**Figure 3.10:** For rigid lipids ( $k_{ijk} = 64.0\epsilon^*$ ) even a bilayer that is twice as large as the one that forms a bilayer for fully flexible lipids does not bend at all. **(a)** Side view. **(b)** Top view.

bending potential on have been performed as well.

As we have seen before, a bilayer of only 256 unrestricted lipids already formed a vesicle. But for 2 times as many lipids with  $k_{ijk} = 64.0\epsilon^*$  the bilayer remains very flat, not showing the least tendency to curl to a vesicle as can be seen in Figure 3.10 where the bilayer is shown after as many as ten million iterations. The more rigid the lipids, the larger the bilayers need be before they start to bend to form into vesicles. Thus, our simplified model with no bending potential enabled us to study the vesicle formation already on smaller system sizes.

A way to study the dependence of the minimum size with the rigidity is to start from a random initial configuration. This has e.g. been done for  $k_{ijk} = 16.0\epsilon^*$ . A cubic simulation box with length  $40.5\sigma^*$  has been filled randomly with 512 lipid molecules and 38332 water particles (Figure 3.11a). During a simulation of 7500000 iterations, the lipids first rapidly aggregate into micelles and small bilayers, also called bicelles. These micelles and bicelles are shown after 400000 iterations in Figure 3.11b. Subsequently, the micelles and bicelles slowly merge into a larger aggregate, being a disc-like



**Figure 3.11:** Spontaneous vesicle formation. **(a)** Initially 512 lipids are dispersed throughout 38332 water particles. **(b)** After 400000 iterations these lipids have formed micelles and small bilayers. **(c)** After 6500000 iterations the micelles have merged into a large bilayer. **(d)** After 7500000 iterations this bilayer has transformed into a vesicle.

## Simulations of bilayer and vesicle formation

		N		
		256	512	1024
k	0	v	v	v
	4	v	v	v
	16	b	v	v
	64	b	b	v
	256	b	b	b

**Table 3.1:** The stiffer the lipids the larger the minimal bilayer size to form vesicles.  $N$  is the number of lipids in the bilayer,  $k$  the bending parameter and  $b$  and  $v$  indicate whether the end configuration of the simulations is a bilayer or a vesicle respectively.

$N_{\text{lipids}}$	$R_B$ [nm]	$R_S$ [nm]	$h_S$ [nm]	$N_{\text{in}}$	$N_{\text{out}}$	$N_W$
256	5.2			no vesicle		
320	5.9			no vesicle		
384	6.6			no vesicle		
448	7.1	6.0	3.9	98	350	245
512	7.6	6.3	3.9	130	382	383
640	8.5	6.8	4.0	171	469	654
768	9.4	7.2	4.0	224	544	1003
1008	10.8	7.9	4.1	310	698	1943
1536	13.1	9.2	4.2	518	1018	4641
2048	15.2	10.3	4.2	747	1301	7936
3072	18.7	12.0	4.3	1185	1887	17128
4096	21.6	13.5	4.3	1642	2454	28361

**Table 3.2:** Overview of the vesicles formed in the simulations. For all these simulations, the radii of the initial bilayers ( $R_B$ ) and the resulting outer vesicle radii ( $R_S$ ) as well as the vesicle bilayer thickness ( $h_S$ ) and the number of lipids in the inner ( $N_{\text{in}}$ ) and outer ( $N_{\text{out}}$ ) membrane layer and the number of encapsulated water particles ( $N_W$ ) are given.

bilayer. Figure 3.11c shows this circular bilayer that is formed after 6 500 000 iterations. Finally this bilayer transforms into a vesicle, encapsulating 385 water particles. This final configuration, after 7 500 000 iterations, is shown in part d of Figure 3.11. Note that, contrary to Section 3.3.2 where no bending potential was used, now all 512 lipids have to aggregate together into a bilayer before the vesicle formation starts. Because this fusion of micelles and bilayers is the slowest part of the process, the simulation time needed to obtain a vesicle is also much longer.

The process described above follows the pathway as suggested by Leng *et al.* [Len03] from experimental studies. Formation of micelles and small bilayers is fast. The fusion of these small aggregates into larger bilayers is slower but once a sufficiently large bilayer is formed, the transition to a vesicle is again fast.

The sizes of the vesicles that are formed in this way are a measure for the minimal number of lipids with the applied rigidity that are necessary for vesicle formation. However, such simulations, starting from randomly dispersed lipids, have two drawbacks. In the first place they are computationally very expensive as the rate at which bilayers grow is very low, because the time between two subsequent fusions with micelles can be quite large. Furthermore, two small bilayers could fuse to a larger bilayer which then forms a vesicle, but since the increase in number of lipids in such a fusion is so large, not the real minimal size is found.

Another way is to start with bilayers of different sizes and to check which remain disc-like bilayers and which form vesicles. In Table 3.1 the results are shown for three different bilayer sizes and five different bending constants.

In order to study the process of the disc to vesicle transition more closely, we have per-

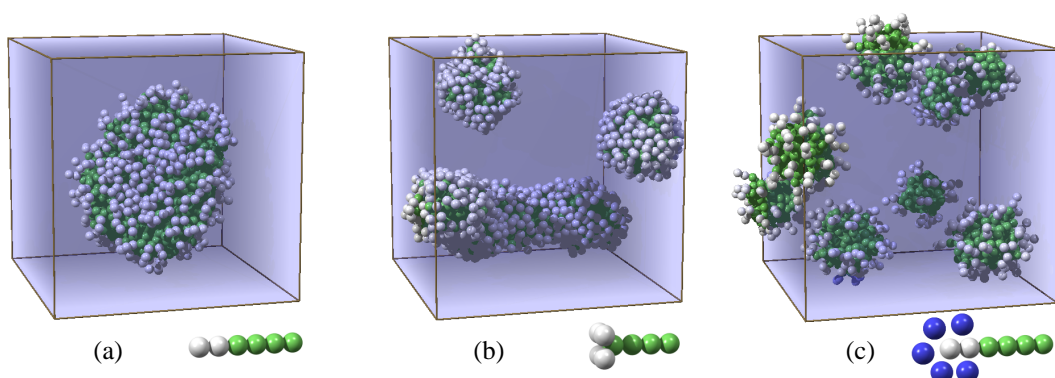
formed additional simulations starting from bilayers of various sizes with the bending constant at  $16.0\epsilon^*$ . From a periodic flat bilayer circular bilayers have been cut consisting of 256, 320, 384, 448, 512, 640, 768, 1008, 1536, 2048, 3072 and 4096 lipids respectively. All bilayers which contained less than 448 lipids remained disc-like flat bilayers after five million MD iterations, whereas the larger bilayers all formed vesicles. For all these simulations, the radii of the initial bilayers ( $R_B$ ), the resulting outer vesicle radii ( $R_S$ ) and the vesicle bilayer thickness ( $h_S$ ) as well as the number of lipids in the inner ( $N_{in}$ ) and outer ( $N_{out}$ ) membrane layer and the number of encapsulated water particles ( $N_W$ ) are given in Table 3.2.

A first observation that can be made from this table is the decreasing bilayer thickness for smaller vesicles. A second observation is that the vesicles formed from bilayers yield almost the same vesicles as from randomly distributed lipids, namely, for the vesicle consisting of 512 lipids now 383 water particles are encapsulated compared to 385 for the randomly formed vesicle described above.

### 3.3.7 Simulations with single tail lipids

All simulations shown so far were performed with our lipid model based of DPPC. With the same coarse grained particles also a single tail lipid could be made. Examples of simulations with several such single tail lipids are shown in Figure 3.12.

Figure 3.12a shows that linear single tail lipids, with a head group that has the same area as the tail, also form bilayers and vesicles. However, not all molecules form bilayers. For example, Figure 3.12b shows that single tail molecules with a head group that is wider than the tail form spherical and cylindrical micelles. Not only the head group itself can be large, but it can also be enlarged by a water shell surrounding it. By decreasing the head group head group interaction  $\epsilon_{H-H}$  to  $1.0\epsilon^*$  and increasing the head group



**Figure 3.12:** (a) Bilayers and vesicles can also be formed by single tail lipids with a small head group. (b) However, single tail lipids with a large head group do not form bilayers and vesicles but only micelles. (c) Also single tail lipids with a small head group that is enlarged by a water shell around their head group only form micelles.

water interaction  $\epsilon_{H-W}$  to  $2.5\epsilon^*$ , the lipids that otherwise form vesicles, now form micelles as shown in Figure 3.12c. This latter simulation conforms best with experimental studies of single tail lipids which usually have a charged head group inducing a water shell around it.

In accordance with the theory of Ref. [Isr91], the shape of the particles, i.e., being cylindrical or cone shaped and having a hydrophobic tail and a hydrophilic head group, seems the relevant factor for the type of aggregates formed.

## 3.4 Discussion

### 3.4.1 Theory and experimental validation

From comparison with experimental results and fully atomistic simulations we have already seen that our lipid model shows the expected behavior for bilayers. The area per lipid and the width of spontaneously formed bilayers are in line with experimental results and fully atomistic simulations.

The bilayer-vesicle transition has been studied previously experimentally, theoretically and using simulations. Leng *et al.* [Len03] showed in a recent experimental study of phospholipid vesicle formation that the vesicle size distribution is determined kinetically by the minimum size at which open membrane fragments, growing as they fuse together, can rapidly make the transition to the vesicle state. Experimental studies on vesicle formation have also been performed using other surfactants such as the synthetic surfactant sodium 6-phenyltridecanesulphonate [Far96] and anionic/cationic surfactant mixtures [Shi02b]. In these studies bilayers and vesicles are observed, but the transition is hard to observe as it is fast [Len02].

The free energy landscape on which the bilayer-vesicle transition occurs has also been studied theoretically. Fromherz [Fro83] introduced the ‘spherical cap’ model in which two contributions, an edge energy and an elastic energy, yield an energy barrier for the transition dependent on the size of the bilayer. By curling the bilayer, the length of the edge and thus the edge energy decreases, while the curvature of the bilayer and thus the elastic energy increases.

### 3.4.2 Comparison with other simulation techniques

Except from with experiments and theoretical studies, our results can also be compared with other simulation methods as the process of vesicle formation has been studied by others by means of other computer simulation techniques and models as well. These computer simulation techniques include Brownian dynamics, dissipative particle dynamics, and Monte Carlo.

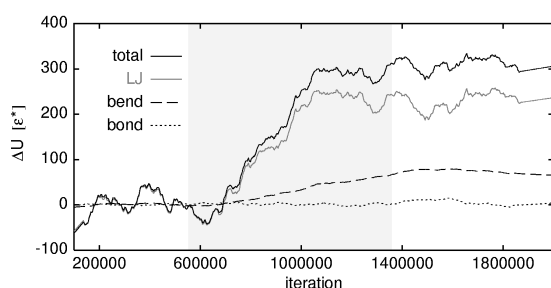
Others have also performed molecular dynamics simulations on lipids and bilayers. Fully atomistic simulations on small bilayers are common practice [Tie97], but the sim-

ulation of large bilayers or vesicle formation is difficult due to the time scale and the number of particles involved. For instance, only one study has shown the formation of a small vesicle on an atomistic level [dV04]. To investigate biologically relevant processes using contemporary computational power also other groups have developed coarse grained models. Goetz *et al.* [Goe98, Goe99] studied the self-assembly of small bilayers from simple one and two tailed amphiphiles. The lipid model which Shelley *et al.* [She01] used to simulate small bilayers is more detailed, consisting of many particle types. Marrink *et al.* [Mar04, Mar03c] showed not only bilayer but also vesicle formation using their own model. This model does include electrostatic potentials, but they are cut-off at small distance thus neglecting their long range nature. Here, we not only study spontaneous bilayer and vesicle formation, but also the formation of vesicles with different sizes and the changes in energy during the vesicle formation. In the lipid model used, we refrained from as much detail as possible to obtain a simple model that still yields realistic behavior.

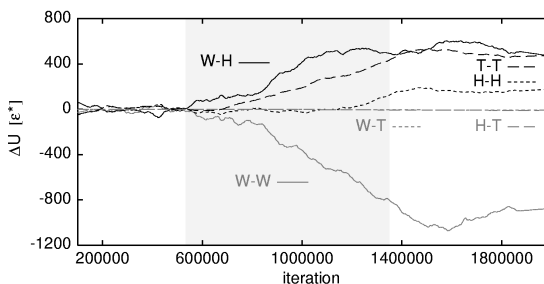
Molecular dynamics is more suited to study the vesicle formation process than other simulation techniques. For instance, in the Brownian dynamics studies the solvent is not taken into account and the amphiphiles are modeled as rigid rods. Using this technique, Noguchi *et al.* were able to show the self-assembly of amphiphiles into vesicles [Nog01]. The lack of a solvent is of course advantageous because it saves a lot of computational cost. However, the absence of the solvent is a drawback of the method as well, because the interaction of the solvent with the amphiphiles appears to be crucial for the aggregation behavior. The computational gain of omitting the solvent is also why Drouffe *et al.* [Dro91] could simulate vesicle formation already in 1991 in a completely different computer simulation. In this simulation no single lipids are present. Instead, a single particle represents part of a bilayer, containing two hydrophilic regions for the head groups of the amphiphiles on both side of the bilayer and a hydrophobic region for the hydrophobic bilayer core, such that a vesicle could be formed in a simulation containing only 252 particles. A bit more detailed solvent free computer model has been published more recently [Far03].

More realistic simulations, i.e., containing a solution, are performed using dissipative particle dynamics (DPD) [Hoo92, Gro97]. In this approach, the property that the Brownian motion of uncharged colloids does not depend on the atomistic details of the solvent, but only on the density, temperature, and viscosity of that solvent, is used. The forces due to individual solvent molecules are lumped together to yield effective friction and a fluctuating force between moving fluid elements. Using this technique, self-assembly of membranes has been shown [Ven99, Shi02a] as well as spontaneous vesicle formation [Yam02] in a way very similar to our simulation results. However, this approach does not yield the correct atomistic description of the molecular motion, which could be crucial for the vesicle formation process, and which is provided by molecular dynamics. Another advantage of MD is that the potential energy of the system can be followed during the process.





**Figure 3.13:** The change in the potential energy during the disc-vesicle transition. The course of the total potential energy as well as its contributions from the LJ, bond and bending potential are given.



**Figure 3.14:** The six different contributions to the change in Lennard-Jones energy during the disc-vesicle transition.

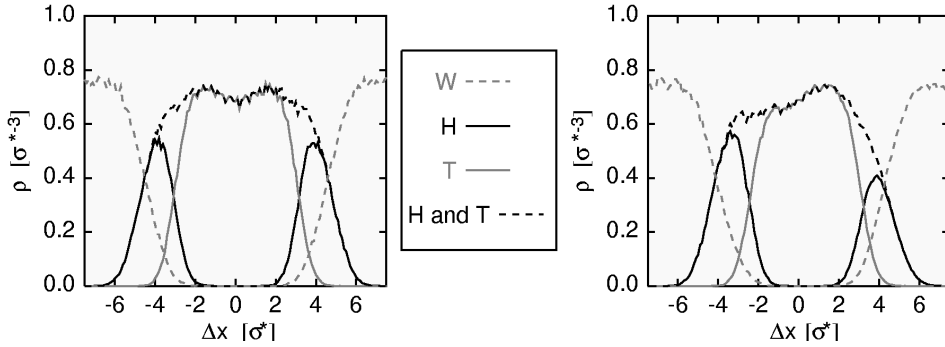
### 3.4.3 Vesicle formation analysis

**Potential energy** Our molecular dynamics simulations allow for the analysis of the potential energy landscape of the disc-vesicle transition. Figure 3.13 shows the change in the total potential energy of the system during the disc-vesicle transition for a bilayer consisting of 1008 (two tailed) lipids with bending force constant  $k_{ijk} = 16.0\epsilon^*$  (from Section 3.3.6). The iteration range in which the transition actually takes place is shaded gray. In the period before the actual transition the bilayer undulates and in the period thereafter the vesicle stabilizes.

Apart from the total potential energy, Figure 3.13 also shows the contributions to the total potential energy of the Lennard-Jones potential, the bond potential, and the bending potential. The bond potential hardly changes during the transition, whereas the bending potential increases. But the main increase is in the Lennard-Jones potential energy. This Lennard-Jones energy can be split further into six different contributions for the six different types of pair interactions between our three particle types. These contributions are shown in Figure 3.14. Noticeable are the decrease of the potential energy due to the W-W interaction and an increase for the T-T, H-H, and W-H interactions, pointing to a less favorable packing of the lipids in the curved bilayer. This is also visible from the density profiles in Figure 3.15. For the flat bilayer (left) the density distribution is rather homogeneous with a small dip at the bilayer center. For the vesicle bilayer the density distribution is far from homogeneous. The head group density for the inner layer is larger than for the outer layer. For the tails it is exactly the other way around, something that can be explained from geometrical arguments.

The important point is that the total potential energy  $U$  of the system increases during the bilayer-vesicle transition. Since the simulations have been performed in the NPT ensemble, thus at constant pressure  $P$  and temperature  $T$ , the corresponding thermodynamic potential is the Gibbs free energy  $G$ . The latter can be written as  $G = U + K - TS + PV$ ,



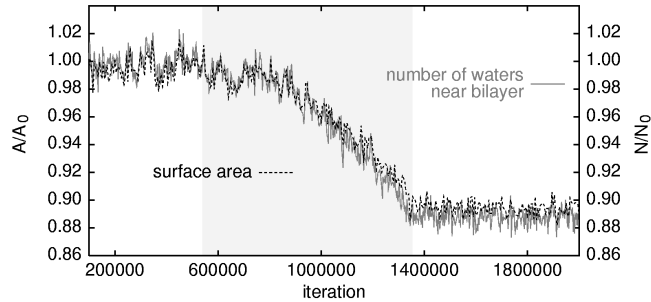


**Figure 3.15:** Density profiles of both the flat bilayer (left) and of the curved bilayer in a vesicle (right). The density  $\rho$  is defined as the number of particles of a specific type per unit of volume and  $\Delta x$  denotes the shortest distance from the particle to the middle of the bilayer.

where  $K$  is the kinetic energy,  $S$  the entropy, and  $V$  the volume of the simulation box. Because the temperature  $T$  is constant, the kinetic energy  $K$  is also constant. The volume of the simulation box  $V$  increases slightly, resulting in an increase of  $PV$  with  $0.5\epsilon^*$ . Combined,  $U$ ,  $K$  and  $PV$  result in an increase of the free energy  $G$ . The system will, however, always evolve towards a state with the lowest free energy, implying a decrease of  $G$ . Since  $T$  is constant, this means that the entropy has to increase during the transition. Hence, these simulations show that the transition is entropy-driven.

An explanation for the entropy increase could be that the number of water particles in the bulk increases during the transition. An indication for this is the decrease in W-W LJ energy in Figure 3.14. An increase in water particles in the bulk coheres with a decrease of the contact area between the bilayer and the solvent during vesiculation.

**Bilayer surface** A measure for the contact area of a bilayer with a solvent is given by its solvent accessible surface. We have calculated this surface area for the bilayer containing 1008 lipids during its vesicle formation using MSMS [San96] with a probe size of  $1.5\sigma^*$ . Figure 3.16 shows the course of the surface area relative to the surface



**Figure 3.16:** The solvent accessible surface of the bilayer and the number of water particles within a distance of  $2.5\sigma^*$  of the bilayer, both given relative to their equilibrium value before vesiculation.

area before vesiculation. The range in which the actual transition takes place is again shaded grey. Within this range, the surface area indeed decreases. This is confirmed by determining the number of water particles in the vicinity (within  $2.5\sigma^*$ ) of the bilayer. The course of this number of particles relative to the number before vesiculation closely matches the relative surface area (Figure 3.16). The resulting increase of the number of water particles in the bulk yields an increased entropy. Hence, instead of the often assumed energy minimization of the edge of the bilayer as the driving force, it is the water that drives the transition.

In order to proof this in more detail, entropy calculations could be performed and similar simulations could be performed with a more detailed water model.

### 3.4.4 Comparison with Fromherz' model

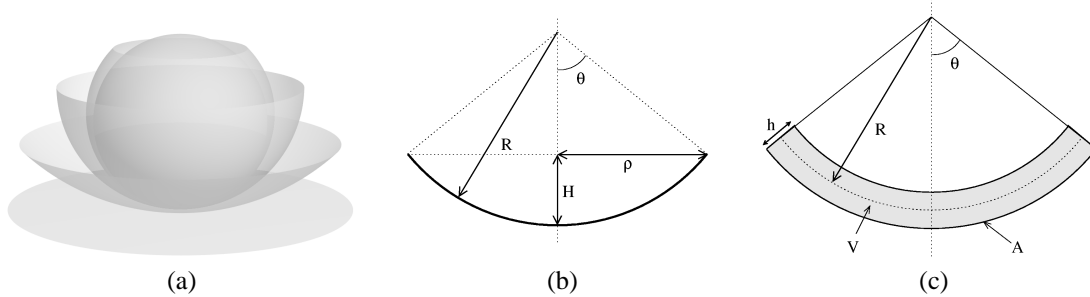
**Fromherz' model** The geometrical transition of the vesicle formation process as observed in our simulations indeed seems to follow Fromherz' [Fro83] spherical cap model, which is shown in Figure 3.17a and 3.17b. In this model the bilayer is described as a 2D structure with constant area  $A_0$  that has a radius of curvature  $R$ , shell height  $H$ , edge radius  $\rho$  and edge length  $L = 2\pi\rho$ . The ratio  $\Omega = H/H_S = R_S/R$  is used as the shape parameter, where  $H_S$  and  $R_S$  are the shell height and the radius of curvature for the closed sphere, respectively.

This model is used to derive a vesiculation index, which predicts the minimal vesicle size from the bilayers edge tension and elasticity. Starting point for this derivation is the Gibbs free energy relation

$$dg_F = -\Pi dA_0 + \gamma dL + \kappa d(1/R) + \mu_L dN_L + \mu_A dN_A, \quad (3.6)$$

with surface pressure  $\Pi$ , edge tension  $\gamma$ , elastic bending stress  $\kappa$ , the number of lipids  $N_L$  and solute molecules  $N_A$ , and chemical potentials  $\mu_A$  and  $\mu_L$ .

Under the assumptions that the area  $A_0$  and the number of lipids  $N_L$  and solute molecules  $N_A$  are constant, two terms to the free energy remain, which are the edge tension and the elastic energy of curving, respectively. In the disc-vesicle transition, the length of the edge and thus the edge tension decreases whereas the curvature of the bilayer and thus



**Figure 3.17:** (a) Schematic figure of vesicle formation according to Fromherz' spherical cap model. (b) 2D representation of this model including definitions. (c) Extension of the model to a 3D bilayer of thickness  $h$ .

the elastic energy increases. Whether or not the transition takes place depends on the ratio between the changes in these two terms, which is represented by the vesiculation index.

By using that the elastic component of  $g_F$  is given by

$$g_{el} = \frac{1}{2}(k_c + \frac{1}{2}\bar{k}_c)(2/R)^2 A_0, \quad (3.7)$$

and that the edge component of  $g_F$  is given by

$$g_{ed} = \gamma_M L, \quad (3.8)$$

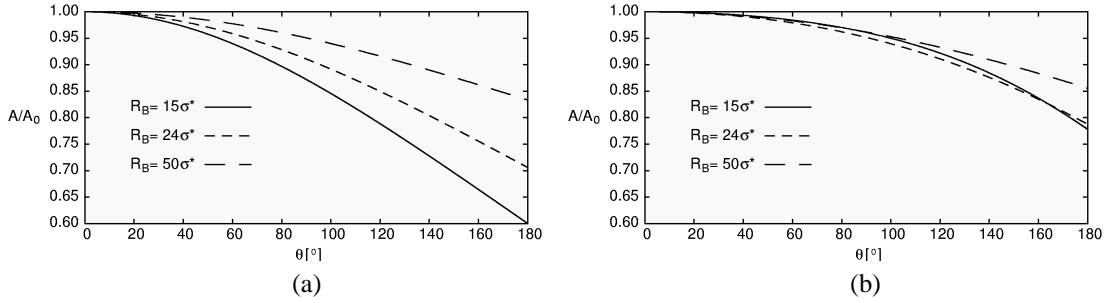
the vesiculation index is given by  $V_F = R_S \gamma_M / 2(k_c + \frac{1}{2}\bar{k}_c)$ , where  $\gamma_M$  is a constant edge tension and  $k_c$  and  $\bar{k}_c$  are the splay and saddle splay modulus of bending.

Our simulations provide a good tool to test these assumptions of constant surface area, constant edge tension, and the formula for the elastic free energy. However, it is more convenient to use a different parameter for the closure: the angle  $\theta$  as defined in Figure 3.17b. The radius  $R$  as a function of this angle is  $R(\theta) = \sqrt{A_0 / (2\pi(1 - \cos\theta))}$  and the edge length is then given by  $L = 2\pi R(\theta) \sin\theta$ .

**Bilayer surface** The first assumption in Fromherz' model was that the area  $A_0$  is constant, whereas in Section 3.4.3 has been shown that the solvent accessible surface area decreases during vesiculation. However, the solvent accessible surface area in Fromherz' model is not  $A_0$  but  $A = 2A_0 + hL$ , where  $h$  is the bilayer thickness which is assumed to be constant. This surface area  $A$  as a function of the closure angle  $\theta$ , relative to the surface area for the flat bilayer ( $\theta = 0$ ), is shown in Figure 3.18a for three different radii ( $R_B$ ) of the initial circular bilayer, each with bilayer thickness  $h = 10\sigma^*$ . According to this model, a bilayer with radius  $R_B = 24\sigma^*$  thus reduces its surface with approximately 30% during the vesiculation. This is much more than the approximately 12% calculated from our MD simulation with almost the same bilayer size as shown in Figure 3.16. Thus, especially for small bilayers, Fromherz' model overestimates the entropic gain of the area reduction.

**Explicit thickness** The bilayer thickness can also be incorporated in the model explicitly. To obtain an analytical expression for the surface area of the bilayer, we extended Fromherz' model with the thickness of the bilayer. The new definitions are shown in Figure 3.17c. This figure shows a 2D intersection of the axial symmetric 3D situation. The assumption of constant area  $A_0$  is now replaced by the assumption of constant bilayer volume  $V_0$ , where  $V_0 = A_0 h$ . Furthermore, just like in Fromherz' model, we keep the bilayer thickness  $h$  constant. For a curved bilayer,  $R$  still denotes the radius of curvature and  $\theta$  the angle for which the total volume of the curved bilayer equals the constant volume. Assuming the volume  $V$  and the bilayer thickness  $h$  to be constant during the disc-vesicle transition, the radius  $R$  as a function of  $\theta$  is now given by

$$R(\theta) = \sqrt{\frac{V_0}{1 - \cos\theta} - \frac{\pi h^3}{6}}. \quad (3.9)$$



**Figure 3.18:** (a) Surface area as function of closure according to 2D bilayer model. (b) The same surface area according to our 3D bilayer model. Thus, the smaller the original bilayer the more the 2D bilayer model overestimates the relative surface reduction of bilayer closing. Furthermore, for small  $\theta$  the surface reduction is minimal.

The surface of the curved bilayer is then

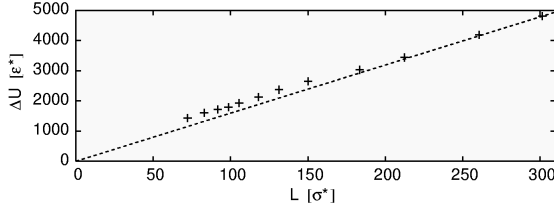
$$A(\theta) = \frac{2V_0}{h} + \frac{2\pi}{3}h^2(1 - \cos\theta) + 2\pi R(\theta)h \sin\theta. \quad (3.10)$$

Compared to the 2D model, the first term is identical as  $A_0 = V_0/h$ . The second term, which is a result of the bilayer thickness, is new, and in the last term only the definition of  $R(\theta)$  changed. Figure 3.18b shows this surface area as a function of  $\theta$  again for the same three bilayers with different radii ( $R_B$ ) and thickness  $h = 10\sigma^*$ . From comparison between Figures 3.18a and 3.18b it is clear that, especially for the smaller bilayers, the difference in surface area calculated is quite large. For bilayers that are very large compared to their thickness, the two models converge towards each other. However, because of the omission of explicitly considering the bilayer thickness in Fromherz' model, the reduction in surface area during vesiculation is overestimated for bilayer sizes that spontaneously form vesicles.

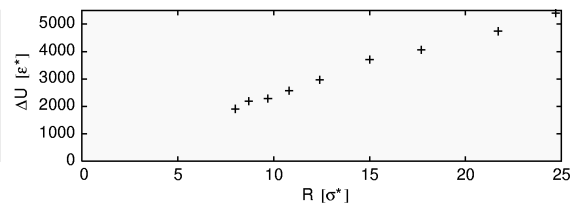
Furthermore, Figure 3.18b shows that, explicitly taking into account the bilayer thickness, the surface area hardly reduces in the initial phase of curving. For very small bilayers the area would, in first instance, even increase slightly. The further the closure, the faster the area reduces, thus the faster the bilayer closes further. This explains why no intermediates are found in experimental studies ([Len02]).

In this new model the surface reduction during vesiculation is approximately 20%. This is still more than the decrease in area measured from our simulations, for which two reasons exist. In the first place, the reduction in surface area will be smaller as the molecules pack less well in a curved bilayer and the surface will be rougher than for the analytical smooth surface. And in the second place, the bilayer thickness appears not to be constant during the simulations. As the bilayer thickness decreases during the vesicle formation, the vesicle radii, and thus the surfaces, will be slightly larger.

**Edge energy versus elastic energy** The free energy can be divided into two parts: the internal energy and the entropy. Molecular dynamics simulations are a perfect tool to study the internal energy. In Section 3.4.3 we have seen the potential energy as a



**Figure 3.19:** Potential energy contribution of the edge is almost linear in the length of the edge.



**Figure 3.20:** Potential energy contribution due to bilayer curvature for closed vesicles is not independent of the vesicle size.

function of the closure of the vesicle. This potential energy is the combined effect of the two terms, edge energy and elastic energy, in Fromherz' model. However, from the differences between the potential energies of different configurations the internal energy contribution to both Fromherz terms can be obtained. These configurations are a periodic bilayer, a free disc-like flat bilayer, and a closed vesicle. The edge tension is obtained from the difference in energy between a periodic bilayer and a free bilayer, consisting of the same number of lipids, as they are both flat and the former does not have an edge and the latter does. Similarly, the elastic energy is obtained from the difference between a periodic bilayer and a vesicle as these are both continuous bilayers, where one is flat and the other curved.

The results for the free bilayer and the vesicle are obtained from the same simulations, thus both the number of lipid molecules and the number of water particles in all these configurations are the same. However, the periodic bilayer simulations are performed in a smaller box containing fewer water particles. Because the number of water molecules differs, the potential energy is calculated as the potential energy of the system minus the potential energy of a system with purely water consisting of the same number of water molecules.

Both the edge energy  $U_{\text{edge}}$  and the elastic energy  $U_{\text{elastic}}$  are calculated for the various bilayer sizes. The edge energy is shown in Figure 3.19 as a function of the edge length  $L$ . As this edge energy is, according to Equation 3.8, expected to be linear in  $L$ , also the line  $V = 15.9L$  is drawn in this figure. From the figure can be seen that the edge energy is indeed almost linear in the edge length. But for small bilayers the energy is slightly higher. This is caused by the high curvature of the edge for these small bilayers, such that the edge can be shielded less efficiently.

The elastic energy of the closed vesicles as a function of the vesicle radius  $R$  is shown in Figure 3.20. Theoretically, from Equation 3.7, this is expected to be independent on the vesicle size, since the vesicle area scales as  $R^2$  and the contribution per unit of area, via the curvature, as  $R^{-2}$ . However, from the figure it is clear that, at least for the vesicle sizes considered, the elastic energy increases with the vesicle size, thus implicating that  $\kappa$  is not curvature independent. In order to find a formula for the dependency, the formation of still larger vesicles should be simulated as well and all simulations should be repeated to obtain better statistics as they have so far only been performed once.

Thus, for the smaller vesicles, i.e., vesicles of the size that are formed spontaneously

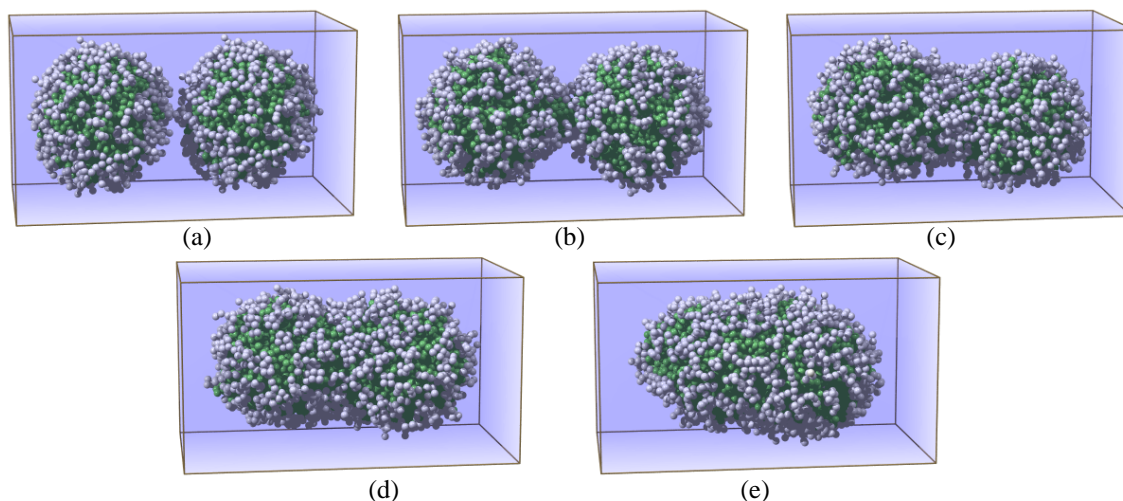
when micelles merge to large enough bilayers, Fromherz' model overestimates the elastic energy of a vesicle and underestimates the edge energy resulting in an overestimation of the energy barrier.

### 3.5 Vesicle fusion

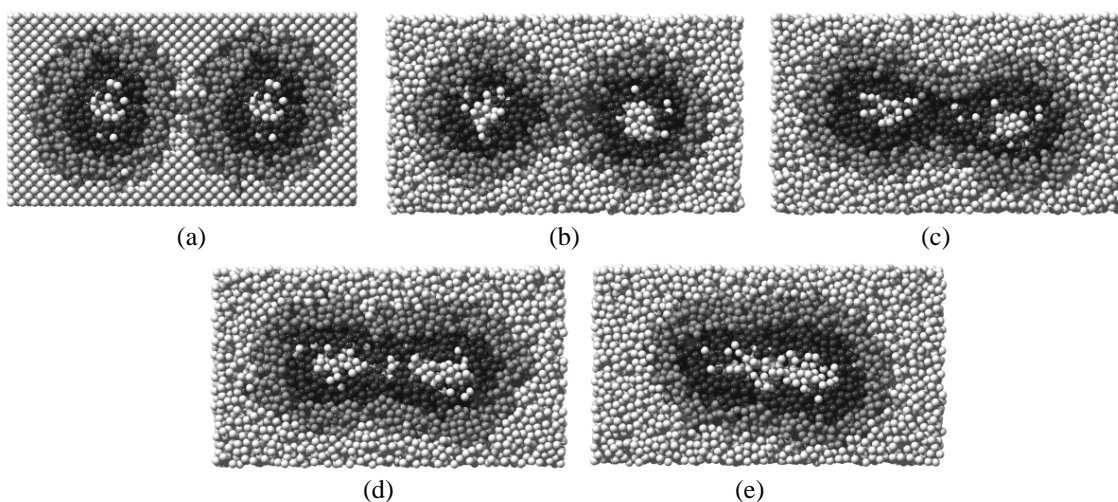
DPPC vesicles that are found in nature are larger than the ones that form spontaneously in our and other atomistic [dV04] simulations. A reason for this could be that small vesicles rapidly fuse to larger vesicles, which subsequently remain stable and are measured in experiments. The fusion process of two vesicles can be studied with our model as well by placing two spontaneously formed vesicles next to each other.

**Small vesicles** In Figure 3.21 vesicle fusion is shown for two small vesicles, consisting of 256 lipids each, for the case with no bending potential. This figure shows five stages of the fusion process. First the vesicles have to come close together (a). Once they meet, the outer membrane layers merge (b). The newly formed combined outer layer has a high curvature near the point of fusion. When the contact area between the two vesicles increases, this curvature is decreased (c and d), and finally an ellipsoid vesicle is formed (e).

In Figure 3.21 the fusion process has been followed from the outside. However, an advantage of molecular dynamics simulations is that it also allows for looking inside the vesicles. Figure 3.22, e.g., shows a slice through the hearts of the two fusing vesicles. In



**Figure 3.21:** Two small vesicles fuse rapidly. When two vesicles come close together (a), the outer membrane layers merge (b). The high curvature of the new outer layer near the point of fusion decreases when the contact area between the two vesicles increases (c). The inner layers then also merge (d) and finally an ellipsoid vesicle is formed (e) where also the interiors have merged.

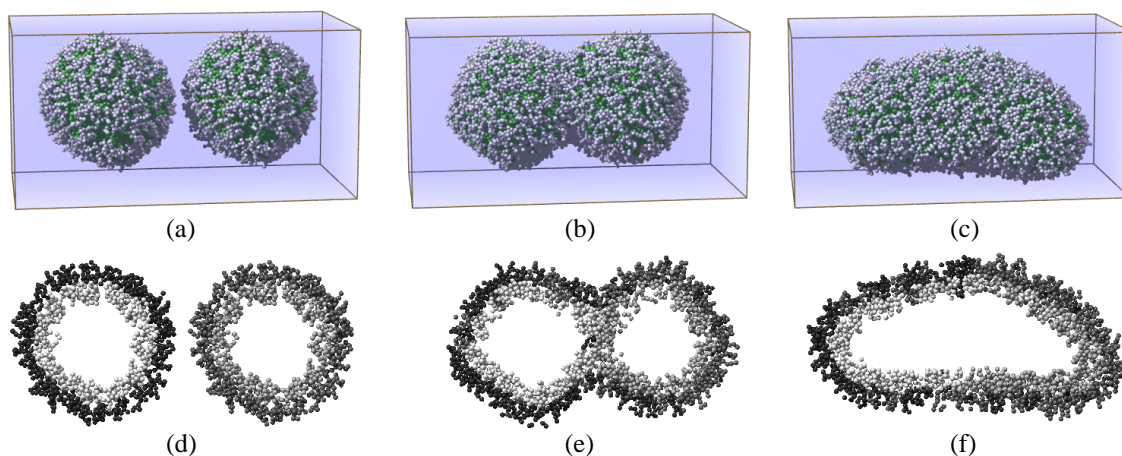


**Figure 3.22:** Slices through the vesicle hearts for the same stages of the vesicle fusion process as in Figure 3.21.

this figure the water is colored white, the lipids that are in the outer layer of the original vesicles light gray and the lipids originally in the inner membranes dark gray. From this figure can be seen that the outer membrane layers fuse first, a stage called hemifusion, followed by fusion of the inner layers and the vesicle interiors. From this figure it is clear that the vesicles indeed fused completely. Because of the coloring method used, it is also clear that the outer layer of the fused vesicle is formed from the outer layers of the original vesicles and that the inner layer of the fused vesicle is formed from the inner layers of the original vesicles. No exchange of lipids between inner and outer layer has occurred.

**Larger vesicles** To study the fusion process in more detail, the process has been repeated for two larger vesicles. Now both vesicles consist of 1024 lipids. The fusion process, which is shown in Figure 3.23, is analogous to that of the small vesicles: the vesicles come together, the outer membranes fuse, followed by the inner membranes and finally followed by the interiors. An advantage of these larger vesicles is that the transition configurations can be seen more clearly. E.g., Figure 3.23e shows that when the two outer layers have fused and straighten, the two inner layers of the original vesicles are still intact and together form a bilayer at the place of fusion. When the straightening of the outer layer continues, the membrane becomes thinner and thinner and the curvature at the corners higher and higher. This results in breakage of this bilayer at one of the sides. At that time, the inner layers and the two vesicle interiors fuse, resulting in full fusion.

**Discussion** In the literature, various theories for vesicle fusion are present. The basis of vesicle fusion seems simple enough; two vesicles touch and merge to form a single vesicle. This process is thought to follow various discrete stages, i.e., from two adjacent



**Figure 3.23:** *Two larger vesicles still fuse, but already less easily because there are fewer hydrophobic patches at the interface.*

vesicles via hemifusion to full fusion. These stages are macroscopic phenomena, relatively easy to discern and these have for example been shown for lipid bilayer vesicle fusion captured by high-speed microfluorescence spectroscopy [Lei03]. However, the structures of these stages and their transitions on a microscopic scale are hard to verify experimentally and are thus cause of debate.

The first question is about initial contact. It is often assumed that proteins are necessary to bring two vesicles in close proximity. Two classes of protein models can be discerned [Jah02]. In the first place, proximity models, such as the fence model, the scaffold model, and local perturbation model, propose that very close apposition of membranes is enough to start fusion. In the second place, pore models argue that continuous proteinaceous pores between the membranes are the basis of fusion.

The second question is about the lipid reorganization of the outer monolayers at the initial phase of hemifusion. Whether or not proteins are necessary to bring the vesicles into close contact, this actual lipid contact is widely accepted to follow the stalk mechanism originally conceived of by Markin and Kozlov [Koz83, Mar84]. Alternative models that receive much less support include the inverted micellar intermediate [Ver79] and the extended lipid hemifusion [Kin96]. However, there are several variants of the stalk model; the original stalk [Koz83, Mar84], the stress-free stalk [Mar02], the modified stalk [Kuz01], and the transmembrane contact stalk [Koz02b].

The third question is about the transition from hemifusion to complete fusion. The stalk-pore hypothesis postulates that the stalk widens radially into a bilayer, called a hemifusion diaphragm, which is subsequently opened by a fusion pore [Koz02a]. Other hypotheses are the direct stalk-pore hypothesis [Kuz01], the anisotropic stalk-pore hypothesis [Koz02a], and the condensed stalk hypothesis [Luk04].

More detailed analysis of our simulations could elucidate which of the proposed mechanisms is most plausible [Sme05].



## 3.6 Conclusion

A coarse grained model for lipids has been developed. Two types of coarse grained particles were used to describe the chemically relevant groups in the lipids: one for the fatty acid tails and the other for the head group. In this lipid model, we refrained from as much detail as possible to obtain a simple model that still yields realistic behavior. The results from molecular dynamics simulations with this coarse grained model show the correct aggregation behavior. Single tailed lipids with a water shell around their head group only form micelles, whereas micelles of two tailed phospholipids spontaneously aggregate further into bilayers. Moreover, properties of the resulting bilayers, such as bilayer thickness, area per head group, density, and water permeability compare well with experimental as well as other theoretical studies.

By the simulation of larger systems, not only spontaneous bilayer formation was obtained, but spontaneous vesicle formation as well. These simulations confirmed the pathway of vesicle formation as suggested in literature from experimental studies. The lipids first rapidly aggregate into micelles and small bilayers, also called bicelles. Subsequently, the micelles and bicelles, limited by the diffusion of these aggregates, slowly merge into a larger aggregate, being a disc-like bilayer, which finally bends to form a vesicle. The advantage of such simulations is that the process can be studied in detail. As bilayers are an intermediate in this vesicle formation process, additional simulations were performed on bilayers of various sizes, which were constructed from a spontaneously formed periodic bilayer, to study the disc to vesicle transition in more detail. As the use of molecular dynamics allows for the study of the potential energy landscape, analysis of these simulations showed that for membranes of phospholipid molecules the potential energy of the membrane as well as the potential energy of the whole system (membrane plus water) increases during the bilayer-vesicle transition. The reason that the transition does take place is that for sufficiently large bilayers, the solvent accessible surface of the bilayer decreases during the transition. This results in fewer water molecules near the membrane and more in the bulk, yielding a higher entropy of the water. The bilayer-vesicle transition is thus an entropy driven mechanism. Further analysis of these simulations also showed deviations from the assumptions on which Fromherz' model, which predicts the minimal vesicle size from the bilayer bending modulus and the edge tension, is based.

The same lipid model can be used further to analyze more complex processes such as vesicle fusion. Another step could be the development of a coarse grained model for proteins.

# 4

## Artificial Chemistry Molecular Dynamics

---

***B***y adding stochastic reactions to molecular dynamics an Artificial Chemistry Molecular Dynamics (ACMD) approach is developed. In this approach, the realistic three-dimensional dynamics from molecular dynamics is coupled with the ability to deal with a varying molecular mixture. The method could be used in all small closed compartments where both diffusion and reaction occur as active processes. The applicability of this new framework is demonstrated on some biologically inspired case studies. In particular, both unilamellar and multilamellar vesicle formation, and vesicle growth, bursting and healing.

Part of this chapter is described in:

A.J. Markvoort, D. Bosnacki and P.A.J. Hilbers, **Artificial chemistry in coarse grained molecular dynamics and its application to vesicle dynamics**, to be published.

---

## 4.1 Introduction

In the previous chapter coarse grained molecular dynamics has been used to study the dynamics of membrane and vesicle formation. In such simulations the set of molecules present is fixed, i.e., both the number of particles and their types are fixed. However, in nature, molecular mixtures change because of chemical reactions. In this chapter phenomena similar to the one studied in the previous chapter are studied, but now in the presence of chemical reactions.

One way to introduce chemical reactions in molecular dynamics (MD) is to combine the bonded potential and the Lennard-Jones potential into one single potential. In such a potential the parabolic part for the bonded state and the Lennard-Jones potential for larger particle separations are separated by an energy barrier. However, not all reactions depend on just two particles. Instead, many reactions can also be catalyzed by additional particles. To incorporate this into the model, three-particle potentials should be introduced. This then requires the calculation of all triplet interactions in the system, tremendously increasing the computational cost of the simulation.

A second way is to use the two original potentials and to switch between these potentials at the moment of a reaction. This moment of the reaction can then be determined stochastically taking into account the surroundings. This latter option results in hardly any additional computational cost. Furthermore, it has the advantage that this method enables, apart from bond forming and bond breaking reactions, also transitions of particle types. This is especially useful for our coarse grained models where every particle represents several atoms and thus also changes within a single coarse grained particle are possible. We call this method, obtained by adding stochastic reactions to our molecular dynamics algorithm, Artificial Chemistry Molecular Dynamics (ACMD).

## 4.2 Combining Molecular Dynamics and Artificial Chemistry

**Artificial chemistry** An artificial chemistry (AC) is a man-made chemical system. More precisely, an AC can be defined by a set of objects and a set of reaction rules which specify how the objects interact. An overview of artificial chemistries is given by Dittrich et al. [Dit01].

In our family of artificial chemistries, the objects are the coarse grained particles. The reaction rules are chemical reactions represented by stochastic transitions of particle types and/or formation or cancellation of bonds between particles. Possible reaction

schemes are



where every  $X$  can be replaced by an arbitrary particle type or molecule. These three reaction schemes represent bonding (b), decomposition (d), and type conversion (t), respectively. In these reaction schemes,  $k_x$  is the ‘rate’ of a reaction of type  $x$ , where this rate is defined as the probability of an enabled reaction to take place in a unit of time as explained below.

A reaction is said to be enabled when this reaction is possible. The decomposition and type conversion reactions are always enabled, but for a bond forming reaction an important prerequisite is that the particles that react should be at a proper distance from one another, i.e., for two particles to bind their distance should be small (around the bond length), and no other particles should be in between.

As stated above, the reactions are represented by stochastic transitions, i.e., reactions take place with a certain probability. The probability of a particle  $i$  to be involved in a certain reaction of type  $x$  is given by the probability  $p_x^i$ . In order to model the role of catalysis as well, this probability has been split in two parts:

$$p_x^i = a_x + b_x \sum_{j \in \text{cat.}} f(r_{ij}), \quad (4.2)$$

where

$$f(r) = \begin{cases} 1 - \frac{r}{d} & \text{if } r \leq d \\ 0 & \text{if } r > d. \end{cases} \quad (4.3)$$

In the first place, this probability consists of a constant contribution  $a_x$ , representing the probability of the particle to be involved in an autonomous reaction. The second contribution, i.e., the second term in the right hand side term, increases the probability in case of the presence of catalysts in the particles vicinity. Every catalyst particle  $j$  within a radius  $d$  of particle  $i$  increases the reaction probability of particle  $i$  by an amount depending on their mutual distance. A prerequisite for  $p_x^i$  to be a probability is of course that the upper limit equals 1. However, usually  $p_x^i$  will be much smaller than one, because the chance that a reaction takes place within the short time interval of one iteration is small. Also, because the interaction radius of a catalyst is usually short, the number of contributions within the summation is limited. Thus if  $a_x$  and  $b_x$  are both small,  $p_x^i$  is small as well. Otherwise, Eq. (4.2) should be maximized to one.

The rate  $k_x$  for a reaction of type  $x$  is defined as the average of the reaction probabilities of all particles participating in that reaction. The rate  $k_x$  of particle  $i$  to convert its type (as in Eq. (4.1c)) according to reaction  $x$  thus equals  $p_x^i$ . The rate for a bonding or decomposition reaction is thus defined as the mean of the probabilities  $p_x^i$  and  $p_x^j$  of the two particles  $i$  and  $j$  between which the bond is formed or removed.

**Combining** The idea of combining molecular dynamics and artificial chemistry is to overcome the fixed molecular composition in traditional MD, while maintaining the realistic dynamics (diffusion). Molecular dynamics and artificial chemistry can be combined by identifying the objects in the AC with the particles in the MD. The coupling can then be obtained by alternating MD and AC steps, resulting in a molecular dynamics simulation of a varying molecular composition.

By coupling MD and AC a deterministic method is combined with a stochastic method. In this combined method three time scales can be discerned. The first time scale is the time step size of the MD algorithm which is typically in the order of femtoseconds. The second time scale is the duration of a single reaction. And the last time scale is the average time between two successive reactions. The duration of a single reaction is assumed to be infinitely small as a bond is only something abstract and a reaction in our method is just replacing one potential by the other.

The frequency at which chemical reactions occur depends on the height of the energy barrier that has to be overcome. The reaction probabilities in our AC have to be chosen in order to obtain this rate. In the choice of the reaction probabilities, a distinction should be made between reactions with a high energy barrier and reactions with a low energy barrier.

Reactions with a low energy barrier occur easily. Reaction partners need time to diffuse toward each other, but when they meet they react sooner or later. Because these reactions are diffusion limited and this diffusion is modeled well by MD, our method will be especially appropriate for such systems.

On the other hand, reactions with a high energy barrier occur only seldom. Possibly, these reaction occur so rarely that in the time intervals and system sizes reachable by molecular dynamics simulations hardly any reactions can be observed. For such a system, the time between two consecutive reactions is much larger than the time needed to obtain a new equilibrium situation by diffusion. To enable the study of these systems as well, the reaction probabilities can be increased above their natural value, increasing the number of reactions in our simulation. We hereby require that still enough time remains between successive reactions to allow the system to obtain a new equilibrium by diffusion.

In the implementation, both the inspection of which reactions are enabled and the calculation of the corresponding reaction rates can be interwoven with the force calculation in an MD iteration. Namely, the forces on a particle as well as the reaction pairs it can belong to (enabled reactions) as well as its reaction probabilities all depend on the distances to the neighboring particles. This leaves for the AC step a loop over all enabled reactions, where for each of these possible reactions has to be calculated stochastically whether the reaction takes place or not. If so, the molecules are changed according to the reaction. Interweaving the reaction rate calculation with the MD force calculation results only in a minimal extra computational cost. Since the actual reaction step scales linearly and is negligible compared to an MD iteration, the performance of the MD

program is not notably changed by adding the artificial chemistry.

The total energy can be preserved under reactions by calculating the potential energy of the reacting particles before and after the reaction and subtracting the difference from the kinetic energy of these particles. This difference in potential energy can be both in the potential energy between the two particles themselves (difference between bonded and non-bonded potential) and with other particles if the type of one or both of the particles changes (different non-bonded potentials).

Here we coupled artificial chemistry with coarse grained molecular dynamics, but it could be coupled with fully atomistic molecular dynamics as well in the same manner.

### 4.3 Biologically inspired case studies

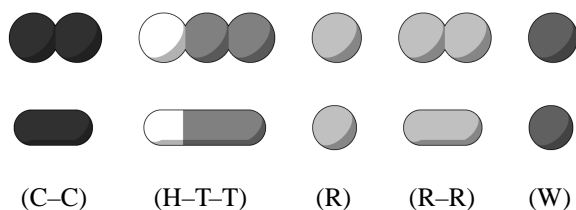
We have implemented the artificial chemistry as described above in our in-house developed molecular dynamics code *PumMa*. In order to demonstrate the applicability of the method, four case studies are shown, dealing with vesicle formation, vesicle bursting, and vesicle growth.

Lipid vesicles (liposomes) are closed structures in which (at least) one lipid bilayer separates an aqueous inner compartment from the bulk external aqueous medium. Because of their analogy to biological cells and the self-assembly and self-organization, such lipid vesicles are considered as possible cell precursors [Lui99].

However, the exact structure of the earliest membrane molecules is unknown and our purpose here is not to simulate the membrane of an existing cell in detail, but to examine the possible minimal requirements for the first cell membrane. In our bottom-up approach, and because we have seen in Chapter 3 that bilayers and vesicles can be formed from single tail linear molecules, we therefore abstract from the real chemical structure and represent a lipid molecule by a coarse grained H–T–T molecule, with its hydrophilic head built of one H particle and its hydrophobic tail consisting of two T particles.

Apart from these lipid molecules the simulations contain water, catalysts and resource molecules. Depending on the case study, these latter are able to react into catalyst or lipid molecules or both. Water is represented by a W molecule consisting of only one particle. The resource particles come in two types, R and R–R, consisting of one and two particles respectively. The catalysts, finally, are represented by C–C molecules consisting of two catalytic particles. Thus, five kinds of molecules are used in total: the solvent W (water), resource molecules R and R–R, catalysts C–C and amphiphilic molecules H–T–T. These molecules, which are shown in Figure 4.1, contain five types of particles: C, H, R, T, and W. Three of these particle types, C, H, and W are hydrophilic; T is hydrophobic; and R is amphiphilic.

The C particles have been chosen hydrophilic to be soluble in the solvent W, but not to pass the hydrophobic core of a membrane too easily. By choosing the R particles amphiphilic, they are able to move through the entire universe: through water as well as



**Figure 4.1:** Van der Waals representation (top) and licorice representation (bottom) of the molecules in our case studies.

through the hydrophobic core of membranes.

To model these hydrophobic, hydrophilic and amphiphilic behaviors, the RA potential (as defined in Chapter 3) has been used for all non-bonded interactions, except for the interaction of a particle of type T with particles of type C, H or W. For these interactions between hydrophobic (T) and hydrophilic particles (C, H and W) the PR potential is used, which is purely repulsive.

All simulations are performed again in reduced units  $\epsilon^*$ ,  $\sigma^*$ , and  $m^*$ . The simulation parameters regarding the MD part are the same in all simulations. Every particle has a mass of  $1 m^*$  and Lennard–Jones parameters  $\epsilon = 1 \epsilon^*$  and  $\sigma = 1 \sigma^*$ . Furthermore, all bonds have an equilibrium length of  $1 \sigma^*$  and a force constant  $c = 100 \epsilon^*/\sigma^{*2}$ . The simulations have been performed at a temperature  $0.6 T^*$  and pressure  $0.0242 P^*$  using time steps of  $0.005 \tau$ . This temperature and pressure have been chosen to resemble atmospheric conditions where the water is fluid and the lipids form membranes. The artificial chemistries, i.e., the reaction sets with their respective probabilities, differ per case.

### 4.3.1 Vesicle Formation

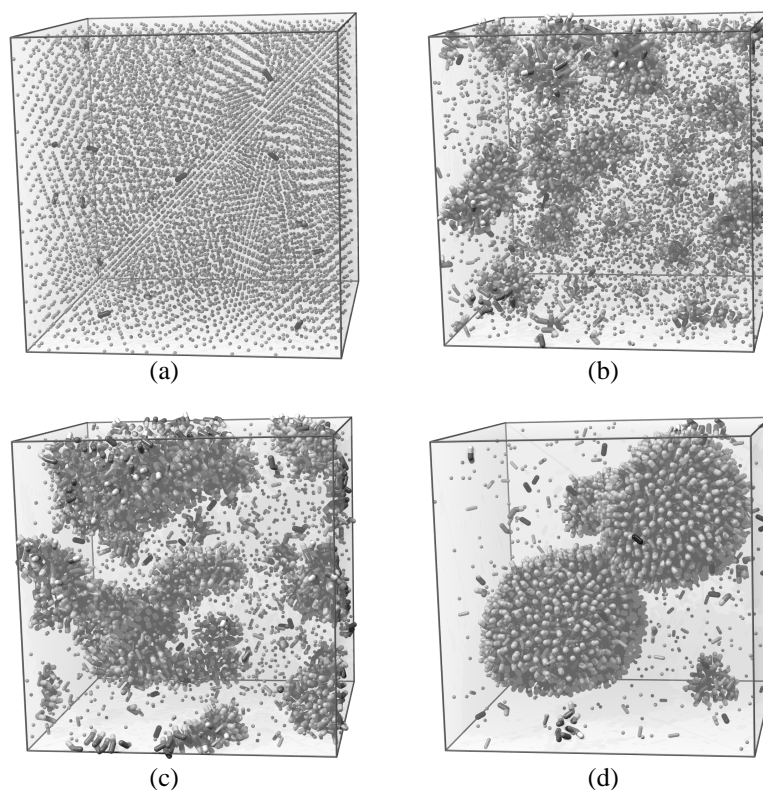
The artificial chemistry that we use in our first case study features all of the five above mentioned kinds of molecules: W, R, R–R, C–C, and H–T–T.

During the simulation, membrane particles H–T–T can be formed in two successive steps. First, two resource particles R react to an intermediate, R–R. This intermediate can subsequently react with another resource particle R to form a membrane molecule H–T–T. Thus, we have the following reaction set:



Both reaction probability constants  $a_{RR}$  and  $a_{HTT}$  have been set to  $10^{-7}$  and both  $b_{RR}$  and  $b_{HTT}$  have been set to 1, such that reactions almost only take place in the vicinity of the few catalyst molecules C–C. In principle reactions can take place after every MD iteration step. However, with these reaction rates most of the time no reactions are performed. On average only once in about 250 time steps a reaction takes place, giving the simulation some time to equilibrate after a reaction.

The initial configuration has been obtained as follows. A simulation box has been

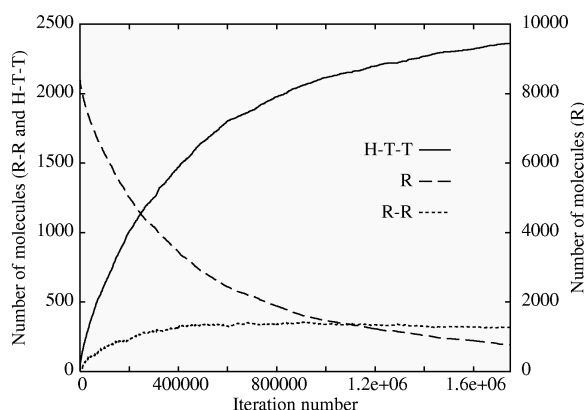


**Figure 4.2:** *Spontaneous vesicle formation. (a) The initial configuration consists of 20 randomly placed catalyst molecules C–C completed by 80% W (not drawn) and 20% R molecules on an fcc grid. (b) After 150 000 iterations the produced membrane molecules have formed micelles, shielding their hydrophobic tail particles T with their hydrophobic head particles H. (c) After 650 000 iterations some micelles fused, forming bilayers. (d) After 1 750 000 iterations the bilayers fused further and curled forming two vesicles.*

created in which 20 catalyst molecules C–C have been placed randomly. This cubic simulation box, with length  $37.6\sigma^*$ , is further filled with approximately 80 percent of solvent particles W and 20 percent of resource particles R, by placing these single particle molecules on a so-called face centered cubic (fcc) lattice. This initial configuration, consisting of 20 C–C molecules, 33968 W particles, and 8491 R particles, is shown in Figure 4.2a. In this and in all subsequent case studies all particles were initially assigned random velocities drawn from a Gaussian distribution, such to obtain the correct velocity distribution and temperature.

The number of resource, intermediate and membrane molecules is shown as a function of time in Figure 4.3. The numbers of solvent and catalyst molecules have not been plotted since those are constant. After 150 000 iterations 2907 resource particles have reacted into 213 R–R and 827 H–T–T molecules. Since C, H, and W particles are all hydrophilic these remain mixed with the amphiphilic R particles, whereas the hydrophobic T particles tend to aggregate, resulting in the formation of small micelles (Figure 4.2b).





**Figure 4.3:** The number of molecules per type versus the iteration number.

Part c of the same figure shows that some of these micelles group together resulting in bilayers. Because the edge of such a bilayer is energetically unfavorable (see Chapter 3), the bilayers finally curl into vesicles (part d). At the end, the simulation box contains 2 such vesicles and some smaller aggregates.

In the following case studies one such a vesicle containing several catalytic particles in its interior is used as a starting point.

### 4.3.2 Vesicle bursting and healing

In the artificial chemistry of this second case study we do not use R–R molecules. Instead of the membrane molecule forming reactions, the reaction set consists only of an auto-catalytic reaction:

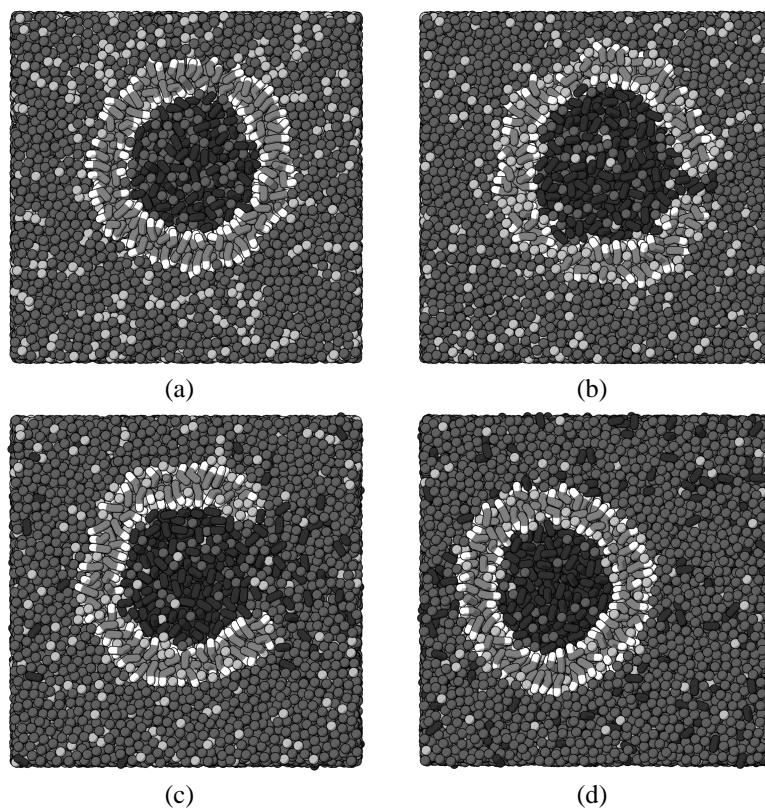


This reaction is auto-catalytic because the reaction is catalyzed by its own reaction product C–C. By definition, auto-catalysis is a self-reproducing process, increasing the concentration of the products until the resources are exhausted.

In order to make the reaction fully autocatalytic,  $a_{CC}$  has been set to zero. Furthermore, because there are many more catalyst molecules than in the former case study, the reaction rate has been decreased,  $b_{CC} = 10^{-5}$ , in order to prevent the reactions from occurring too soon after each other.

The initial configuration is shown in Figure 4.4a. This initial vesicle consists of 1152 membrane molecules with 486 water and 433 catalyst molecules inside and no catalysts outside the vesicle.

Because all catalysts are inside the vesicle, new molecules are only formed inside the vesicle as well. These molecules remain inside the vesicle as they can hardly pass the membrane due to the hydrophilic nature of their C particles. At the same time, resource particles can keep entering the vesicle due to their amphiphilic nature, causing the vesicle to inflate (Figure 4.4b) until the membrane is stretched out so much that it breaks. At that point, the surplus of particles inside the vesicle flows out (Figure 4.4c)

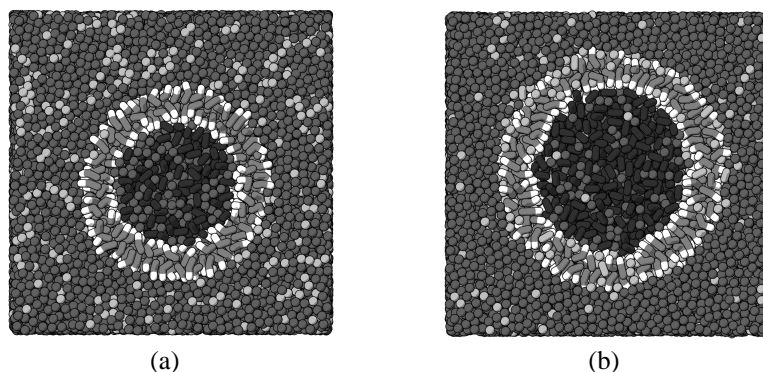


**Figure 4.4:** *Intersections of the simulation box at different stages of the bursting vesicle. A vesicle (a) filled with auto-catalytic particles grows and then bursts (b) resulting in outflow of excess interior particles (c) after which the vesicle closes again (d).*

and finally the vesicle closes itself again (Figure 4.4d). In principle this process could repeat itself, but because of the bursting, the auto-catalytic reactions now also take place outside the vesicle and thereby clear away the resource particles.

### 4.3.3 Vesicle growth without bursting

In some sense the third case is a combination of the previous two cases. Again all five kinds of molecules are used, while the reaction set consists of reactions that produce both membrane particles (Equations (4.4a) and (4.4b)) and catalysts (Equation (4.5)). The rate constants used are:  $a_{RR} = a_{HTT} = a_{CC} = 0$  and  $b_{RR} = b_{HTT} = b_{CC} = 10^{-5}$ , such that the reactions can take place in the vicinity of catalyst molecules exclusively. Because the initial configuration consists again of a vesicle built of 1152 membrane molecules filled with 486 water and 433 catalyst molecules in an environment randomly filled with 20% resource (6770 R) and 80% water (30376 W) molecules, the reactions occur once again only inside the vesicle. The catalyst molecules formed remain again inside the vesicle. On the other hand, the membrane molecules, because of their hydrophobic tails, fuse with the bilayer directly or form micelles first which fuse with the bilayer subsequently. These molecules will fuse in first instance with the bilayers inner



**Figure 4.5:** Intersections of (a) the initial vesicle and (b) the grown vesicle after 3 000 000 iterations.

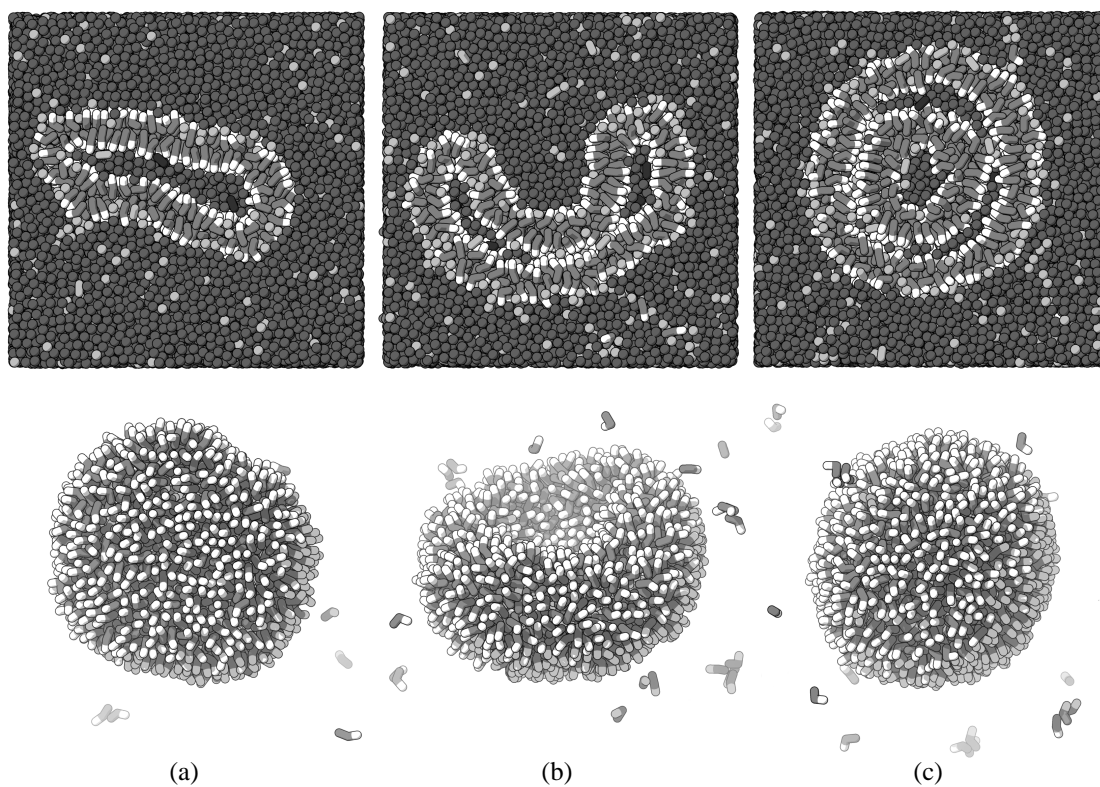
leaflet. And depending on the mobility of the bilayer, these new membrane molecules will stay in that inner layer or also flip to the outer layer. In the latter case, as is the situation for our small H–T–T molecules, the bilayer as a whole grows. As a result, both the membrane size and the interior volume increase, resulting in a vesicle growth as shown in Figure 4.5. After 3 000 000 iterations, the membrane consists of 1726 H–T–T molecules, enclosing 3084 C, W and R particles.

#### 4.3.4 Multilamellar vesicle creation

In the fourth case study we use the same artificial chemistry as in Case 4.3.1, i.e., the same set of molecule types, reactions and reaction rates are used. The initial configuration is the same as in Case 4.3.2, except for the number of catalytic molecules (now only 20) inside the vesicle.

Starting from this initial configuration, extra membrane molecules are created once again in the interior of the vesicle. These membrane molecules fuse again with the bilayer directly or indirectly via intermediate micelles, because of which the membrane grows. However, the number of particles in the interior remains almost constant because hardly any water particles diffuse through the membrane. As a result, the vesicle starts to distort. First an oblate ellipsoid-like shape (Figure 4.6a, iteration 500 000) is formed with a higher surface to volume ratio than the initial spherical vesicle. Because of the surplus of membrane particles, the vesicle does not only reshape, but some of the membrane particles dissolve in the water as well. This results in the monomers visible in the figure. The high curvature at the edge of the oblate ellipsoid is energetically unfavorable, like the edge of a bilayer, resulting in the oblate ellipsoid to curl (Figure 4.6c, iteration 1 000 000), minimizing this edge, and finally forming a multilamellar vesicle (Figure 4.6d, iteration 1 400 000), i.e., a vesicle with another vesicle in its interior.

That a true multilamellar vesicle has been formed with two separate membranes can be shown by clustering the tail particles (T) of the membrane molecules. This is done as follows. Two particles are said to be directly connected if they are closer than a given threshold distance apart. Two particles are said to be belonging to the same cluster,



**Figure 4.6:** Different stages of the formation of a multilamellar vesicle. For every stage an intersection (top) is shown as well as all membrane molecules (bottom). The initial vesicle first transforms into an oblate ellipsoid (a), subsequently starts curling (b), and finally closes to form a multilamellar vesicle (c).

if they are either directly connected or indirectly connected through a path of directly connected particles. For the final configuration of this simulation, this results in two clusters; one formed by the 587 H–T–T molecules of the inner membrane and one containing the 1874 H–T–T molecules of the outer membrane.

The interior of the original vesicle, and consequently all catalyst particles C–C, ends up in the thin layer between the inner and the outer vesicle. Conversely, the interior of the inner vesicle stems from the exterior of the original vesicle.

## 4.4 Discussion

The results of the case studies show the feasibility and usefulness of extending the molecular dynamics framework with chemical reactions. Of course, no vesicle fission is shown yet and the actual mechanisms of cell division and metabolism are incomparably more complex than the analogous phenomena in our model. Nevertheless, as it has also been argued about Ono’s model [Ono01a], the results obtained with these models could provide a better insight into the origin and the mechanisms of self-maintenance and self-reproduction of the first cells. The fact that we work in three dimensions can

only strengthen this kind of arguments.

In the sequel we discuss some general observations and facts that hold for all experiments.

**Steady supply of resource particles** The production rate of new molecules decreases in time, as can be deduced from Figure 4.3. This can be explained by the decrease of the number of resource particles with every reaction. A lower concentration of resource particles results in a decreasing probability of two of those particles to meet in the vicinity of a catalyst molecule.

This decrease in concentration can be prevented by introducing equilibrium reactions (i.e., reactions of transition type given by Equation (4.1c)) between W and R. For instance, when every W particle has a probability of  $10^{-7}$  to transform into a resource particle R and vice versa every R particle has a probability of  $10^{-6}$  to transform back into a W particle a constant concentration of 10% R particles (versus 90% W particles) will be obtained during the simulation.

Ono's model from Ref. [Ono01a] ensures a steady supply of resource particles via recycling. For this purpose this model contains also decomposition reactions of Equation (4.1b). Both the auto-catalytic and membrane molecules decompose into particles of a special (waste) type. The waste particles are recycled into resource particles. This has not yet been used in our model, because we wanted to keep, in first instance, the complexity of the chemistry minimal, but it may be a requirement to obtain fission.

**Other types of membrane molecules.** Instead of the H-T-T membrane molecules also even simpler or more complex membrane molecules could be used.

We experimented with a simpler model in which the membrane particles are of the form H-T, i.e., they contain only one T particle. As a consequence this model does not feature the molecule R-R which is just an intermediate step toward the production of H-T-T molecules and reactions (4.4a) and (4.4b) are replaced with a single reaction



With this simpler model we successfully repeated all case studies observing the same phenomena as with the model from the text.

On the other hand, more complex membrane molecules could prevent the flip-flop of membrane molecules between the inner and outer leaflet. However, the use of such more complex membrane molecules has the disadvantage that a much more complex chemistry arises if one wants to build everything from single resource particles.

**Comparison with 2D work by Ono** In the model of Ono [Ono01a] only membrane particles of the form H-T are used and, consequently, no intermediate R-R particles. Furthermore, Ono's model is dissipative particle dynamics like. The force on a particle can be split in three parts. Apart from the conservative forces acting between particles, that are also present in MD, Ono's model also uses a dissipative force and a random force

for every particle. The conservative forces, which can be both repulsive and attractive in MD, are purely repulsive in Ono's model. The size of this force depends on the extent of hydrophobicity. The repulsion between hydrophilic and hydrophobic particles is the strongest, followed by the repulsion between particles of the same type, and finally the weakest force is between the neutral (the resource R and the waste Y) and the other particles. So, attraction is actually implemented as a weaker repulsion. Further, in Ono's model the mass of all particles is one unit, except the water (W) particle which has a mass of 0.5 units.

Furthermore, there are some essential differences between membranes in 2D and 3D. As shown in Chapter 3, bilayers in 3D curl forming vesicles by minimizing their total surface. Conversely, in 2D the surface is only minimized by the actual closing itself, not yet by curling. As a result, in 2D there is no driving force to curl. Instead, vesicle formation is just coincidence. Using similar arguments, one can conclude that the formation of multilamellar vesicles in 2D would be even less likely. Indeed, this is not reported in Ono's experiments.

## 4.5 Conclusion

We have presented a novel framework for molecular simulations in three dimensions which combines coarse grained molecular dynamics with chemical reactions. In this method, a reaction is a switch from one potential describing the interaction between two particles to another potential or a change of the parameters of the potential. The occurrence of reactions is determined stochastically taking into account the surrounding of every particle. For example, two particles that are within a certain distance of each other form a bond with a certain chance and this chance can be increased by the presence of catalysts in the neighborhood. As the surroundings of each particle are already inspected in molecular dynamics to obtain the forces on the particles, the additional computational cost is minimal. Furthermore, apart from bond forming and breaking reactions, this method also enables transitions of particle types.

The feasibility and usefulness of the approach have been demonstrated on simulations of biological phenomena related to the origin of life. In particular, we have presented a three-dimensional model that features formation of membranes, micelles, and vesicles, as well as growth, and bursting of the latter.

One natural avenue for future work is to cover a complete 'life cycle' of a vesicle in one continuous simulation. This will include formation, self-maintenance (growth) and self-reproduction (fission). Also, in order to make our models more realistic, recycling of the particles through degradation reactions could be introduced. Another direction is to experiment with less abstract, i.e., more realistic molecules and chemical reactions.



# 5

## Hybrid Molecular Dynamics - Direct Simulation Monte Carlo

---

***F***or many practical cases, such as gas-wall interactions, molecular dynamics simulations are a perfect tool. Here we use such molecular dynamics simulations to study the wettability influence on heat and particle flow in nanochannels. However, to study larger microchannels molecular dynamics is computationally too expensive. Consequently, one has to resort to other faster simulation methods which are, however, usually less accurate. Other particle based methods, like the direct simulation Monte Carlo method, are much faster than molecular dynamics, but lack especially the precision to describe the interactions at the walls accurately. To combine the advantages of both methods, i.e., the accuracy of molecular dynamics and the speed of the direct simulation Monte Carlo method, a hybrid method has been devised in which molecular dynamics is used where necessary for accuracy and the direct simulation Monte Carlo method where possible.

Parts of this chapter are described in:

A.J. Markvoort, P.A.J. Hilbers and S.V. Nedeia, **Molecular dynamics study of the influence of wall-gas interactions on heat flow in nanochannels**, Phys. Rev. E, **71**, 066702 (2005).

S.V. Nedeia, A.J.H. Frijns, A.A. van Steenhoven, A.J. Markvoort and P.A.J. Hilbers, **Hybrid method coupling molecular dynamics and Monte Carlo simulations to study the properties of gases in microchannels and nanochannels**, Phys. Rev. E, **72**, 016705 (2005).

---



## 5.1 Introduction

A host of novel techniques, such as thin film manufacturing, nanotube manufacturing and characterization, the development of novel materials, and microchannel cooling, demand the prediction of heat transfer characteristics at the nanometer scale [Maj00]. In this respect, the transport properties of gases at the gas-solid interface play a very important role, and are studied with various different experimental and theoretical techniques [Som95]. A good example is formed by micro- and nanochannels. These channels can be used to cool mechanical and electrical components in a compact and efficient way. Cooling these devices is essential since most components produce heat when operating. Using a gas or fluid flow through these channels, the devices can be cooled locally where the power is produced. This becomes more and more important as these components become smaller and smaller and produce relatively more power [Sch02]. Theoretical techniques used to study micro- and nanochannels include continuum approaches and particle simulation methods. Large systems can be described well using a continuum approach. However, when the system size decreases or when one focuses on the interface behavior, the continuum approach starts to fail. Much effort has been put into extending macroscopic analyses to microscopic conditions in time and space. For example, the validity of the continuum approach has been identified with the validity of the Navier-Stokes equations [Bir94]. This requires that the Knudsen number ( $\text{Kn} = \lambda/L$ , where  $\lambda$  is the mean free path of the molecules and  $L$  the physical length of the system) is small compared to unity, with the limit  $\text{Kn} = 0.1$ . When the characteristic size of the device decreases or when the gas is more rarefied, such that  $\text{Kn} > 0.1$ , the continuum flow model is no longer valid and must be replaced by another model. A possibility is to change the governing equations of the flow model from the Navier-Stokes equations to the Boltzmann equation, which involves the molecular velocities instead of the macroscopic quantities. This integro-differential equation can be solved using a finite element or finite difference method or alternatively using a particle simulation method as the direct simulation Monte Carlo (DSMC) method.

But there are clear limitations to these methods. These are often simplified models, like the DSMC method where particles are represented as hard spheres and boundary conditions are used to represent the gas-solid interface [Fre97, Fre99]. These boundary conditions are a crucial ingredient in continuum fluid mechanical calculations. However, they cannot be derived from the continuum differential equations themselves, and it is often not easy to determine them experimentally, whereas the transport properties of gases at the gas-solid interface can play a very important role in the overall behavior. An alternative is to use molecular dynamics (MD), allowing for the simulation of both the wall and the gas explicitly. MD has long been used in statistical mechanics and chemistry, but the molecular dynamics technique can also be adapted to study systems with thermodynamic fluxes and thus can also be used to study microscopic heat transfer phenomena [Pou03]. Using such non-equilibrium molecular dynamics techniques the steady state response to flows of momentum or energy can be studied. The first at-

tempts were made around 1975 by Hoover [Hoo75, Hoo83]. The technique has grown in popularity, especially as with the increasing computer power allowing the simulation of larger and larger systems on one side and the miniaturization of many electronic and mechanical devices on the other side, practical applications come within reach. Various molecular dynamics studies have been reported for specific gas-solid and fluid-solid interfaces, like for example the argon-nickel [Chi93] or the water-platinum [Kim02] interface.

Here we perform a more general systematic molecular dynamics study in order to investigate the influence of the gas-gas and gas-surface interaction parameters on the heat transport over a gas-surface interface, both in the case of a stationary gas and in the case of a gas flow. For this purpose, we studied the behavior of a gas confined between two parallel plates, for gas densities ranging from rarefied gases to very dense gases and for various interaction strengths using an adapted version of our parallel molecular dynamics code *PumMa*.

Such MD simulations are able to simulate the effects near the solid wall and near the boundaries of areas of phase transitions accurately. However, because all the pair interactions between all particles have to be calculated, the MD method is too time-consuming for the number of particles needed to simulate a dense gas flow in a microchannel. Therefore, subsequently a hybrid method has been developed coupling molecular dynamics and the direct simulation Monte Carlo method to combine the advantages of the MD and MC simulations, performing MD near the boundaries for the accuracy of the interactions with the wall, and MC in the bulk because of the low computational costs.

## 5.2 MD study of wettability effect on heat and particle flow in nanochannels

### 5.2.1 Model

**Molecular dynamics** As has been explained before, molecular dynamics is a computer simulation technique where the time evolution of a set of interacting particles is followed by numerically solving the equations of motion (Newton's law) of classical multi-body systems. Given the positions, masses and velocities of all particles in the system and the forces on the particles, the motion of all (individual) particles can be followed in time by calculating the (deterministic) particle trajectories.

The forces between particles are governed by the gradient of the potentials between these particles. A commonly used potential is the Lennard-Jones (LJ) potential (see Equation (3.3)). This potential describes the van der Waals interactions and it consists of two parts; a repulsive and an attractive part.

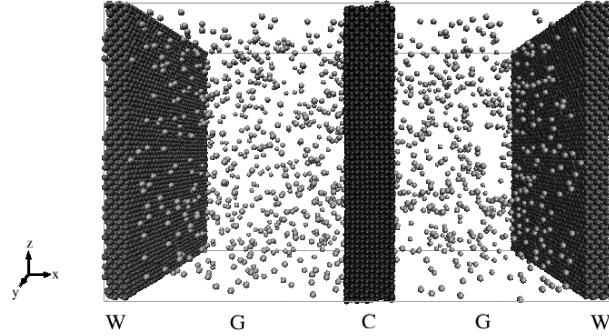
This Lennard-Jones potential is especially appropriate for noble gases, but following [Hal75, Gua96] it can also be used (as a pseudopotential) for metals. Of course, more

realistic potentials for metals are available which take into account many-atom interactions, but because LJ interactions capture the essence of all systems and we are not directly interested in one particular metal this potential suffices for our needs.

Again two different cut-off radii are used. A cut-off radius  $r_{c,ij} = 2.5\sigma_{ij}$  is taken into account again to closely resemble the original Lennard-Jones potential. This potential is denoted once more in the rest of the text as ‘repulsive and attractive’ (RA) or simply Lennard-Jones. The second cut-off radius used is again  $r_{c,ij} = 2^{1/6}\sigma_{ij}$ , which only leaves the repulsive part of the potential, closely resembling hard spheres that are used in other simulation techniques. This potential is denoted further once more as purely repulsive (PR) or hard spheres. As we have seen before, these potentials can also be thought of in respect to hydrophilic and hydrophobic interactions. Hydrophilic-hydrophilic interactions are described by the RA potential, where the parameter  $\epsilon_{ij}$  provides the degree of attraction between the particles. The smaller  $\epsilon_{ij}$  is, the smaller the attraction. And the PR potential, which lacks the attractive part altogether, describes purely hydrophilic-hydrophobic interactions.

**Model parameters** Because we are not directly interested in one specific system but in the dependency on the gas-wall interaction, the parameters used in our model are expressed again in reduced units. Our system consists of the following reduced units: the unit for length  $\sigma^*$ , the unit for mass  $m^*$ , and the unit for energy  $\epsilon^*$ . All other reduced units can be derived out of these choices [Fre02, All87]. The units for length and mass are chosen such that our particles have size  $1\sigma^*$  and mass  $1m^*$ . The unit of energy is chosen such that the parameter  $\epsilon_{ij}$  in the potentials, which varies in the different simulations and between wall and gas particles, is around unity.

**System** All MD simulations that we present to study the gas-solid interface in thermal equilibrium, in the presence of a heat flux, as well as in the presence of a Poiseuille flow are performed on the same system. The system that we have used for all these studies is shown in Figure 5.1. It consists of two walls that are placed in a box of size  $80.00\sigma^* \times 46.89\sigma^* \times 46.89\sigma^*$ , separated from each other in the  $x$  direction. These walls consist of 18000 particles each, where these particles form a face centered cubic (fcc) lattice. We name one wall W and the other C. Because of the use of periodic boundary conditions this represents two infinitely large parallel plates. The space in between the two plates is filled with gas particles (G). Simulations are performed for different gas densities  $n_0$ . This density is defined as the number of gas particles divided by the volume available to the gas, i.e., the volume of the simulation box minus the volume of the two walls. The total number of particles in the box ranges from 37300 for the lowest gas density ( $n_0 = 0.01\sigma^{*-3}$ ) to 91998 for the highest density gas ( $n_0 = 0.4\sigma^{*-3}$ ) simulated. The temperature of the two plates can be controlled independently by coupling them to a heat bath, whereas the gas can only heat up or cool down by collisions with these walls. The walls are in no way restricted, i.e., except for the Lennard-Jones potential



**Figure 5.1:** Snapshot of the system simulated: Two fcc walls (W and C) of which the temperature can be controlled separately and gas in between. The gas density ( $n_0$ ) in this case equals  $0.01 \sigma^{*-3}$ . Both the walls and the gas are simulated using molecular dynamics.

there are no other forces on any of the particles in the system.

The walls were formed in a prior simulation. In this simulation 18000 particles were placed randomly in a simulation box. This system was initially given a high temperature. By cooling this system down the system crystallized. This crystal was placed in a wider box forming one wall. None of the atoms was fixed or restricted in any way such that the walls can in principle move through the simulation box. However, as can be seen for example from the clear peaks in the inset of Figure 5.2 where the density profile of a wall is given, the walls keep their position. The mass of a wall is so large compared to the mass of one gas particle that a single collision hardly affects the wall. Multiple collisions are needed, but simultaneously also collisions from the other side of the wall take place. When the system is in equilibrium, the forces on the wall from both sides cancel each other. The walls thus do not need to be restricted in any way. The walls are kept together by the Lennard Jones interaction between the particles that formed the crystal in the beginning.

The system consists of two types of particles: gas particles (G) and solid (or wall) particles (S). The mass and the size of both particle types are taken equal, namely, the mass of each particle is  $m = 1 m^*$  and the size  $\sigma = 1 \sigma^*$ . That the gas particles are in the gas phase and the wall particles solid is purely controlled by the Lennard-Jones parameter  $\epsilon$ . For the solid-solid interaction, the RA potential with strength  $\epsilon_{S-S} = 6\epsilon^*$  is used whereas for the gas-gas interaction ( $\epsilon_{G-G}$ ) and the gas-solid interaction ( $\epsilon_{G-S}$ ) the RA potential with values between  $0.05\epsilon^*$  and  $0.5\epsilon^*$  or the PR potential are used.

The mass and the size of the solid particles could have been chosen differently, but this choice was made to keep the system as simple as possible, though realistic. To show that the values chosen are in realistic ranges, we consider the example of an Argon gas and a Calcium crystal. In SI units the corresponding LJ parameters are  $\epsilon = 0.0104 \text{ eV}$  and  $\sigma = 3.40 \text{ \AA}$  for Ar and  $\epsilon = 0.2152 \text{ eV}$  and  $\sigma = 3.60 \text{ \AA}$  for Ca. Converting these parameters to our reduced units yields  $\epsilon = 6.0\epsilon^*$ ,  $\sigma = 1.0\sigma^*$ , and  $m = 1.0m^*$  for the crystal and  $\epsilon = 0.048\epsilon^*$ ,  $\sigma = 0.944\sigma^*$ , and  $m = 0.997m^*$  for the gas.

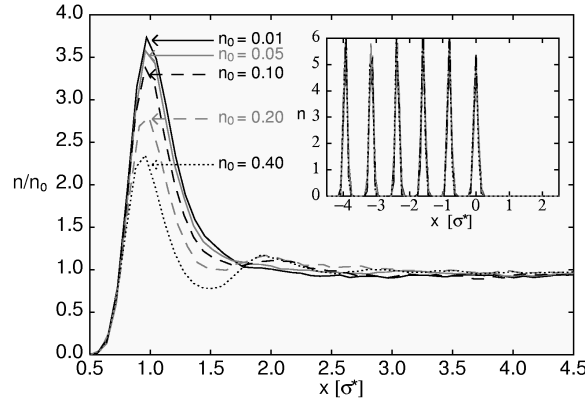
Every simulation consists of two parts. In the first part the system is run until equilibrium is reached. From the second part, the macroscopic quantities like density, temperature, flow velocity, and heat flux are obtained. The number of iterations differs per simulation as a lower density gas needs more time to come to equilibrium. In order to keep the number of iterations needed for the lowest gas densities tractable we used the following procedure. We start with the highest concentration simulations, where the average gas density  $n_0 = 0.40 \sigma^{*-3}$ . A configuration from this simulations is taken when this simulation has come to equilibrium and half of the gas particles are removed to obtain an initial configuration for gas density  $0.20 \sigma^{*-3}$ . This is repeated till the lowest density ( $n_0 = 0.01 \sigma^{*-3}$ ) is reached. The initial configurations obtained in this way are already closer to equilibrium than randomly generated configurations, but still sufficient iterations were used to let the system come to equilibrium for every concentration. The simulations at the lowest densities for example consist of 5 000 000 iterations, taking approximately 200 hours on 8 cpu's of our AMD Athlon 1800+ Beowulf cluster.

### 5.2.2 Simulation results for density oscillations near interface

We start by considering the behavior of the gas particles near the wall. The wall influences the nearby gas particles. As a result the gas density near the wall can deviate from the density in the middle of the channel (the bulk density). These density oscillations near the interface are studied both as a function of the bulk density and as a function of the gas-wall and gas-gas interaction since this behavior at the gas wall interface is the basis for understanding heat conduction and flow.

In order to study the gas particles at the interface most purely, the system is studied in thermodynamic equilibrium. Both walls and the gas have temperature  $T = 1 T^*$  and the total momentum of the system is zero. In this case, all four gas-surface interfaces in our system are all identical. Therefore we concentrate our attention on one of them, namely the interface at the left in Figure 5.1. First the influence of the gas density is studied and subsequently the influence of the gas-gas and the gas-wall interaction strength.

**Density dependence** The influence of the bulk density on the density near the wall is shown in Figure 5.2. In this figure density profiles are shown for gas densities ranging from  $n_0 = 0.01 \sigma^{*-3}$  to  $n_0 = 0.4 \sigma^{*-3}$ , where these profiles are all normalized with  $n_0$  to make them comparable. In the inset also the density profile of the wall is shown. From this inset can be seen that the origin of the coordinate system has been chosen such that the last lattice plane of the wall is centered around  $x = 0$ . The Lennard-Jones parameters used for these simulations, both for the gas-gas interaction  $\epsilon_{G-G}$  and for the gas-wall interaction  $\epsilon_{G-S}$ , equal  $0.5 \epsilon^*$ . For all gas densities, the normalized density is slightly lower than unity in the bulk as a result of an increase in the density near the walls. This effect of an increased density near the wall is referred to as wetting of the surface. Particles sticking to the wall are entropically unfavorable, but energetically it is

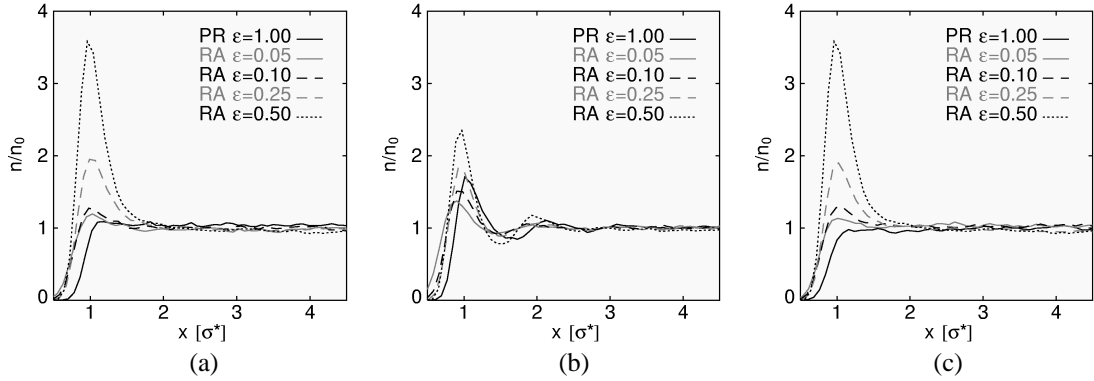


**Figure 5.2:** Profile of the relative density near the wall as a function of the distance to the wall for various gas densities. Both for the gas-wall and the gas-gas interaction the RA potential is used with  $\epsilon_{G-G} = \epsilon_{G-S} = 0.5\epsilon^*$ . The inset shows the density profile of the wall. The last lattice plane of the wall is centered around  $x = 0$ .

much more favorable for a particle to be nearby the wall, because there it has more near neighbors and thus more negative energy contributions than in the gas phase. Because this effect is the highest for a low density gas a higher peak in the relative density near the wall is visible for a low density gas than for a high density gas. For a high density gas the interface can also be saturated, resulting in a second layer which is visible as a second peak for the dense gases around  $x = 2\sigma^*$ .

**Interaction dependence** Apart from the bulk density, the density deviations also depend on the gas-gas and gas-wall interactions. In Figure 5.3a the density profiles are shown first for the case of a relatively low gas density ( $n_0 = 0.05\sigma^{*-3}$ ) when  $\epsilon_{G-G}$  and  $\epsilon_{G-S}$  are varied simultaneously. Important aspects in this figure are the differences in the heights of the peaks for different interaction parameters and the area in vicinity to the wall. For the PR potential the first particles are at a larger distance from the wall compared to the RA potential, what we refer to as the depletion layer for the PR potential. The differences in peak height can be explained because the larger the interaction parameter of the interaction of gas particles with the wall particles the larger the energetic gain for particles to be close to the wall.

In part b) of the same figure, the density profiles are shown for the case of a relatively high gas density ( $n_0 = 0.4\sigma^{*-3}$ ). Also here the depletion layer for the PR interaction potential is visible, although being somewhat smaller. A larger difference is formed by the heights of the density peaks. For the strongest attractive interactions the peaks for  $n_0 = 0.4\sigma^{*-3}$  are smaller than for  $n_0 = 0.05\sigma^{*-3}$ , for which two reasons exist. In the first place, the surface is already much more covered in case of a high density and in the second place, the bulk particles in a high density gas have already many more close neighbors such that the energetic advantage of being near the wall is relatively smaller than for a rarefied gas. For the weakest attractive interactions it is the other way around. Here the peaks for  $n_0 = 0.4\sigma^{*-3}$  are higher than for  $n_0 = 0.05\sigma^{*-3}$ . A remarkable

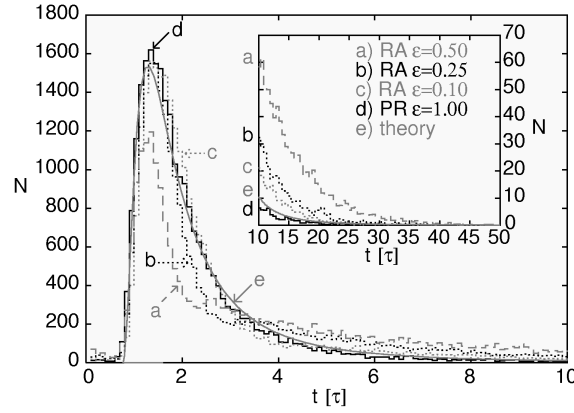


**Figure 5.3:** Density profiles. **(a)** Density profiles for a low density gas ( $n_0 = 0.05 \sigma^{*-3}$ ) for different interaction parameters  $\epsilon_{G-G} = \epsilon_{G-S}$ . **(b)** The same, but now for a dense gas ( $n_0 = 0.4 \sigma^{*-3}$ ). **(c)** Density profiles for a low density gas ( $n_0 = 0.05 \sigma^{*-3}$ ) for different gas-surface interaction parameters  $\epsilon_{G-S}$ , but with constant gas-gas interaction ( $RA \epsilon_{G-G} = 0.5 \epsilon^*$ ).

difference is that for the high density gas even the PR has a peak in the density near the wall. This can be explained as that the peak in the density is not only the result from the attractive force from the wall, but also from ‘pushing’ by the bulk gas atoms.

Finally, Figure 5.3c shows the effect of only varying the gas-wall interaction while keeping the gas-gas interaction at  $0.5 \epsilon^*$  in case of the low density gas. Comparison with Figure 5.3a shows hardly any differences, indicating that, for low density gases, the influence of the gas-gas interaction parameter is negligible compared to the gas-surface interaction parameter.

For low gas densities the density peak at the interface thus depends on the attractive part of the gas-wall interaction potential. The higher the interaction of the gas with the wall is, the higher the gas density near the wall. An explanation for this is that in the presence of an attraction between the gas and the wall, some gas particles stick to the wall for some time. To study this, the time ( $\Delta t$ ) spent by a particle per collision with a wall is measured. A way to measure this time is to measure the time that a particle in a low density gas spends within an interface region which is defined as a slice of width  $2\sigma^*$  from the center of the last surface lattice plane. The particles in a low density gas with a density of  $n_0 = 0.01 \sigma^{*-3}$  have a mean free path ( $\lambda = 1/(\sqrt{2}\pi\sigma^2 n_0)$ ) of about  $22\sigma^*$ , which is much larger than the width of the interface region. As a result, the chance that a particle collides with another gas particle in this interface region is small, resulting in the fact that the time spent in the interface region by a particle equals the distance to be traveled in the interface region in the  $x$  direction divided by its velocity in the  $x$  direction plus the time spent at the interface. In Figure 5.4 histograms of this time ( $\Delta t$ ) spent by a particle per collision with a wall within the interface region are given for different gas-wall interaction parameters for a gas density  $n_0 = 0.01 \sigma^{*-3}$  and gas and wall temperature  $T = 1 T^*$ . In the same figure also the theoretically expected time distribution is given for the case of reflective walls. In this theoretically expected



**Figure 5.4:** Histograms of the time spent per particle per collision with a wall within the interface region for different gas-wall interaction parameters. The inset zooms in at the tails.

time distribution an additional collision time of  $0.15\tau$  is added to account for the time needed to flip the velocity at the interface. The velocity cannot just flip from negative to positive. Instead, the particle should decelerate and again accelerate in the opposite direction resulting in the extra time. For the PR interaction, thus in case there is no attracting force of the wall, the distribution is close to this theoretical result. In case of an attractive gas-surface interaction, high velocity particles are hardly affected, but slower particles are caught by the wall. As a result, the peak decreases for average collision times as these particles are trapped for a while and are thus visible for even higher residence times as can be seen clearly in the inset in the same figure.

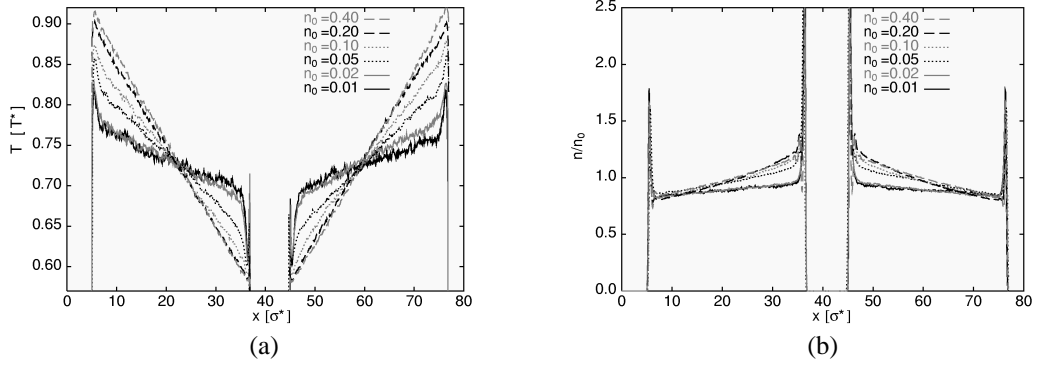
Summarizing, density peaks at the interface can have two distinct roots. In the first place they are the result of gas particles sticking to the wall because of the energetic gain. For high gas densities they are also the result of other gas particles pushing from only one side, the bulk side. Next we will study how this wetting influences heat conduction.

### 5.2.3 Heat flux dependence on wettability

To study the heat conduction a temperature difference between the two plates is implied. One wall, the warm wall W, is kept at temperature  $T = 1.0T^*$  whereas the other wall, the cold wall C, is kept at temperature  $T = 0.5T^*$ . As a result, the gas in between the two plates shows a temperature gradient. The influence of the gas density as well as the influence of the gas-gas and gas-wall interactions on this temperature gradient are studied.

**Density dependence** For  $\epsilon_{G-G} = \epsilon_{G-S} = 0.25\epsilon^*$ , the temperature and normalized density profiles for different gas densities are shown in Figure 5.5. From the figure it is clear that when the average gas density is higher, a higher temperature gradient is present in the gas. Furthermore, for high densities the temperature profile is linear, whereas for low densities it is linear in the bulk and different near the interfaces. These





**Figure 5.5:** Temperature (a) and density (b) profiles of the gas for different gas densities when one wall has a temperature  $1.0T^*$  whereas the other wall is kept at  $0.5T^*$ .

increased gradients in the temperature near the interfaces coincide with the increased density near these interfaces.

Xue *et al.* [Xue04, Eas04] studied the effect of ordering near the walls for the solid-liquid interface, also using MD. They concluded from their simulations that the layering of the liquid near the interface does not enhance the thermal transport. Our results for high density gases which limit to the liquid phase match with this conclusion. However, for lower density gases, an enhanced thermal transport is clearly visible from the increased temperature gradients in Figure 5.5, an effect that is outlined in the next section. Another property that can be derived from our simulations is the heat current. The heat current vector is given by [McG04]

$$\mathbf{q} = \frac{d}{dt} \sum_i \mathbf{x}_i E_i, \quad (5.1)$$

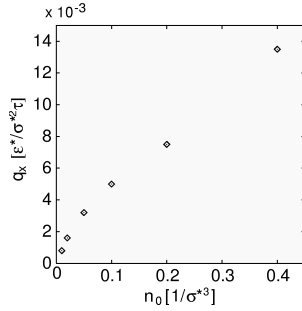
where the summation is over all the particles in the system, and  $\mathbf{x}_i$  and  $E_i$  are the position vector and energy of particle  $i$  respectively. For a pair potential, such as the potentials that we use, Equation (5.1) can be recast as

$$\mathbf{q} = \sum_i \mathbf{v}_i E_i + \frac{1}{2} \sum_{i,j} (\mathbf{F}_{ij} \cdot \mathbf{v}_i) \mathbf{r}_{ij}, \quad (5.2)$$

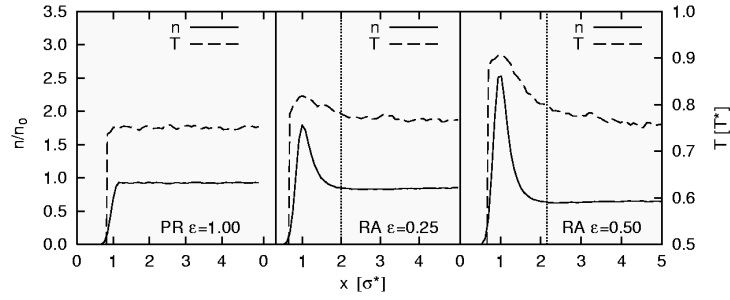
where  $\mathbf{v}$  is the velocity vector of a particle, and  $\mathbf{r}_{ij}$  and  $\mathbf{F}_{ij}$  are the inter particle separation and force between particles  $i$  and  $j$ , respectively. Because of the geometry of our system we are here interested in the  $x$  component of this vector, i.e.  $q_x$ .

This heat flux in the  $x$  direction as a function of the gas density is shown in Figure 5.6. This heat flux can also be calculated by measuring the energy that is added and removed by the heat bath that is used to keep the walls at their constant temperature. This energy divided by the simulation time and twice the area in  $yz$  direction of the simulation box yields the same numbers. An increase in the density clearly results in an increase in the heat flux.

**Interaction dependence** As shown in the previous section, for a gas density  $n_0 = 0.01 \sigma^{*-3}$  large temperature jumps occur near the walls. From Figure 5.7 it can be seen that the exact shape of these jumps depends on the gas-gas interaction and the gas-



**Figure 5.6:** The heat flux  $q_x$  for different gas densities for  $\epsilon_{G-G} = \epsilon_{G-S} = 0.25\epsilon^*$ .



**Figure 5.7:** Density and temperature profiles of the gas for different gas interaction parameters, zoomed in at the interface region for an average gas density  $n_0 = 0.01\sigma^{*-3}$ .

wall interaction. In this figure the density and temperature profiles are given for three different values for  $\epsilon_{G-G}$  and  $\epsilon_{G-S}$ , where has been zoomed in at the interface region. This figure shows that the width of the temperature jump overlaps with the width of the density peak near the wall. This increased density near the wall comes from wetting, as we have seen in Section 5.2.2.

As a result of particles sticking to the wall, their velocity is adapted much more to the wall temperature than for particles that only make a single collision with the wall. This can be seen in Table 5.1, where the number of particles  $N$  of the bulk gas (thus excluding the interface region) of the left compartment and their average temperature  $T_{avg}$  are given both for particles moving from the warm to the cold wall and for particles moving from the cold wall to the warm wall. The higher the attraction with the wall the fewer particles in the bulk and at the same time the higher the difference in temperature between gas particles moving to the left and particles moving to the right.

As can be seen from Table 5.2 this also influences the heat flux. In this table the heat flux is given for different combinations of interaction parameters for the same gas density ( $n_0 = 0.01\sigma^{*-3}$ ). As the deviations within the rows of the table are much larger than

potential	$C \rightarrow W$		$W \rightarrow C$	
	$N$	$T_{avg}$	$N$	$T_{avg}$
PR	294	0.71	292	0.75
RA 0.10	287	0.71	274	0.75
RA 0.25	281	0.68	261	0.77
RA 0.50	225	0.62	204	0.80

**Table 5.1:** The difference in average temperature of particles moving in the direction from the cold to the warm wall ( $C \rightarrow W$ ) versus particles moving in the opposite direction ( $W \rightarrow C$ ) depends on the gas interaction parameters.

		gas-wall			
		PR	RA 0.10	RA 0.25	RA 0.50
gas-gas	PR	0.00046	0.00047	0.00087	0.00124
	RA 0.10	0.00046	0.00046	0.00086	0.00122
	RA 0.25	0.00046	0.00046	0.00081	0.00117
	RA 0.50	0.00045	0.00046	0.00080	0.00113

**Table 5.2:** Heat flux  $q_x$  [ $\epsilon^*/(\sigma^2\tau)$ ] for different parameters for the gas-gas as well as the gas-wall interaction for a low gas density  $n_0 = 0.01\sigma^{*-3}$ .

within the columns, we notice again that the relevant parameter is the gas-wall interaction strength, whereas the gas-gas interaction is of much less influence on the resulting heat flux. An increased gas-wall attraction thus results in an increased ordering at the interface and an increased heat flux, grounding our conclusion from the previous section that for gases, contrary to liquids, the layering near the interface enhances thermal transport.

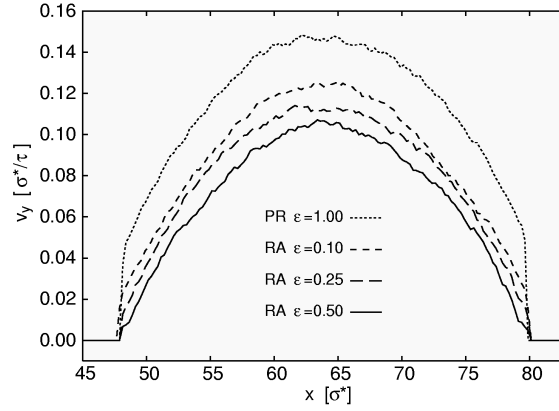
### 5.2.4 Gas flow dependence on wettability

The wettability does not only influence heat transport. As shown by Cieplak *et al.* [Cie00, Cie01] and Nagayama *et al.* [Nag04], the flow profile of a liquid near a solid-liquid interface also depends on the interface wettability.

**Thermal equilibrium** A Poiseuille flow can be induced in the model in different ways. The first method is a gravitational flow. This is achieved by applying an additional force to all gas particles. A second method, the one that we use here, is a pressure driven flow. This is created by applying the additional force only to the gas particles at the inlet of the channel. In order to generate a flow in the positive  $y$  direction, i.e. from the back to the front in Figure 5.1, an additional force in the  $y$  direction is applied to all gas particles with a  $y$  coordinate between zero and three, i.e. in a small slice at the back of the figure.

Because of this additional force the gas starts to flow. Because the wall particles are not restricted in position and there is friction between the gas and the wall, the walls start to move also. There are again several ways to prevent this. One solution that is often applied is to add additional harmonic forces to all wall particles to keep them close to their original position. However, since we do not want to add any additional forces to restrict the walls we apply a different method. We repeatedly remove the linear momentum that is transferred from the gas to the wall because of the friction. By resetting the total linear momentum of the wall particles to zero every 10 iterations, when this total linear momentum is still negligible, the walls remain at their place without having to constrain the particles within the walls.

The friction between the wall and the gas increases with the flow velocity and when the total frictional force equals the additional force on the gas particles an equilibrium flow is reached, resulting in a velocity profile that is quite Poiseuille in appearance. The resulting flow profiles are shown in Figure 5.8 for different gas-wall interaction parameters for a gas density  $n_0 = 0.4 \sigma^{*-3}$  and  $\epsilon_{G-G} = 0.25 \epsilon^*$ . All four velocity profiles have the same shape as expected since the gas-gas interaction is the same. The difference between the four profiles is the velocity of the gas at the interface. Since the walls are stationary this velocity at the interface is the slip. As we have seen, in case of a strongly attractive force between the gas and the wall particles, gas particles stick to the wall resulting in such a large friction that the flow velocity at the interface is zero, i.e. no



**Figure 5.8:** Flow-velocity profiles in the channels for different gas-wall interaction parameters for an average gas density  $n_0 = 0.4 \sigma^{*-3}$  and a RA  $\epsilon_{G-G} = 0.25 \epsilon^*$  gas-gas interaction.

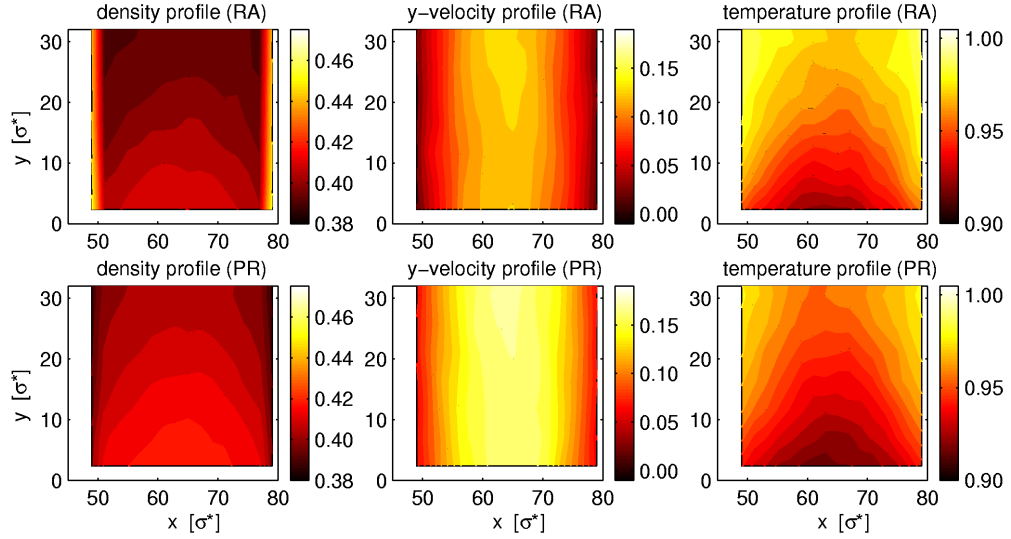
slip. However, the lower the gas-wall attractive interaction is, the smaller the friction and thus the larger the slip. For the case of the purely repulsive gas-wall interaction the extra depletion layer between the wall and the gas results in an even lower friction and thus an even larger slip.

**Cold gas cooling warm walls** As we have shown, the wettability influences the heat transfer over the interface as well as the flow velocity at the interface. And, both effects are influenced in an opposite fashion: an increased wettability results in an increased heat transfer and at the same time in a decreased flow velocity. When cooling warm walls with a cold gas flow, both these effects play a role.

In order to study this heat transfer from warm walls to a cold fluid, the temperature of the fluid has to be reset when it crosses the periodic boundary. This can be achieved by rescaling the velocity of every particle that crosses the periodic boundary in the flow direction. The temperature is defined as the deviations in the velocities from the local mean flow velocity. The rescaling is thus performed by first subtracting the local mean velocity corresponding to the  $x$  position of the particle, subsequently rescaling the velocity to the desired temperature and finally again adding the local mean flow velocity that was removed in the beginning.

The results for a gas inflow temperature  $0.9 T^*$  and walls at temperature  $1.0 T^*$  are shown in Figure 5.9 for two different gas-wall interaction parameters. On the top the profiles are shown for the RA potential with  $\epsilon_{G-S} = 0.25 \epsilon^*$ , whereas the bottom part is for the PR potential with  $\epsilon_{G-S} = 1.0 \epsilon^*$ .

The profiles at the left hand side show the density distribution. Because of the attractive interaction between the gas and the wall particles density peaks appear again in case of the RA interaction, whereas there is again a dip in the density near the walls in case of the repulsive PR interaction. The profiles in the middle show the flow velocities in the channels. It is clearly visible that the flow for the PR potential is again higher than for the RA potential. The attraction with the wall causes friction resulting in hardly any slip



**Figure 5.9:** Density, flow-velocity and temperature distributions in the channels for a gas of average density  $n_0 = 0.4\sigma^{*-3}$  at temperature  $T = 0.9T^*$  flowing along wall at temperature  $T = 1.0T^*$  for two different sets of interaction parameters: (top) RA  $\epsilon_{G-S} = 0.25\epsilon^*$  interaction between gas and wall particles, (bottom) PR  $\epsilon_{G-S} = 1.00\epsilon^*$  interaction between gas and wall particles, whereas the gas-gas interaction is the same in both cases (RA  $\epsilon_{G-G} = 0.25\epsilon^*$ ).

near the walls. However, in case of the repulsive interaction, a large slip at the walls is present. But the attractive interaction also enables more heat transfer between the wall and the gas, thus resulting in a higher temperature of the gas in case of an attractive interaction than in case of repulsive interaction as can be seen from the profiles at the right of the figure.

In order to keep the walls at their constant temperature, as much energy has to be added from the heat bath to the walls as is removed from the walls by the gas. The net effect of the wettability on the cooling can thus be studied by measuring this energy that is added to the walls by the heat bath. For the simulations described above, the heat that is carried away equals 50 units of energy per unit of time for the PR gas-wall interaction versus 52 units of energy per unit of time for the RA gas-wall interaction. The cooling is thus slightly better for the RA gas-wall interaction. However, when the forces to generate the flow for the case of the RA gas-wall interaction are increased with 45% such that the mean flow velocity equals that of the PR case, the amount of heat that is carried away increases to 135 units of energy per unit of time. Thus, at the expense of an increasing pressure drop much more heat can be carried away.

Thus, although the flow is smaller, the amount of heat that can be removed is larger in case of attractive walls because of the better heat transfer over the interface. This explains the results of experiments of convective heat transfer in silicon micro channels with different surface conditions, i.e., a micro channel with hydrophilic walls versus a micro channel with hydrophobic walls [Wu03].

### 5.2.5 Conclusion of molecular dynamics simulations

Two potentials were used to describe the gas-wall interaction; the Lennard-Jones potential with different interaction strengths and the PR potential for a purely repulsive interaction. The most remarkable difference in the resulting density profiles at the interfaces is that hard sphere gases result in a depletion layer compared to RA gases.

In case of an attractive gas-surface interaction an increased density near the wall is visible for all gas densities. For a purely repulsive gas-surface interaction this increase is only visible for high gas densities. Thus, for high densities the gas is not only attracted by the wall but also pushed against the wall by the other gas particles in the bulk.

From varying the gas-surface interaction only on one side and both the gas-surface and the gas-gas interactions on the other, it is clear that the gas-gas interaction is not as important as the gas-surface interaction for the behavior at the interface.

Whereas the increased layering at the solid-liquid interface for higher solid-liquid binding strength seems to have no effect on the thermal conductivity [Xue04], the solid-gas binding strength seems to have an effect on the solid-gas thermal conductivity.

Hard sphere (PR) interaction results in specular walls whereas strong attractive interaction results in thermal walls (RA). The amount in which a wall behaves like a thermal wall depends on the gas-wall interaction strength.

In case of flow, also the flow profile depends on the gas-wall interaction strength. The weaker the interaction is, the larger the slip at the interface. Thus in case of a cold flow along a warm wall there are two opposite effects. The heat transfer from the wall to the gas is enhanced for high attractive interaction whereas this causes the gas to flow slower. But the net result is that more heat can be transferred in case of an attractive interaction than in case of a purely repulsive interaction. This explains the results from Wu *et al.* [Wu03] who concluded that the Nusselt number and apparent friction constant of the trapezoidal microchannels having strong hydrophilic surfaces (thermal oxide surfaces) are larger than those having weak hydrophilic surfaces (silicon surface). This suggests that convective heat transfer can be enhanced by increasing the surface hydrophilic capability at the expense of increasing pressure drop.

A disadvantage of using molecular dynamics is that the simulations are quite time consuming. In order to obtain proper statistics for the profiles of macroscopic quantities simulations up to millions of iterations can be necessary, requiring weeks of simulation even when using 10 CPU's in parallel on our AMD Athlon 1800+ Linux cluster. The simulation of larger systems thus requires much more computer power or more efficient and thus simplified methods, like the direct simulation Monte Carlo method.

## 5.3 Direct simulation Monte Carlo method

### 5.3.1 The method

Another particle simulation technique that can be used to study heat flow in micro- and nanochannels is the direct simulation Monte Carlo method [Bir94]. In DSMC, movements and collisions of particles occur where the collisions are generated stochastically with scattering rates and post-collision velocity distributions determined from the kinetic theory of gases. However, this method, based on the Boltzmann equation, is only valid for low density gases.

For a dense gas the governing equation of the flow model is an extension of the Boltzmann equation, called the Enskog equation [Ens22, vB73]

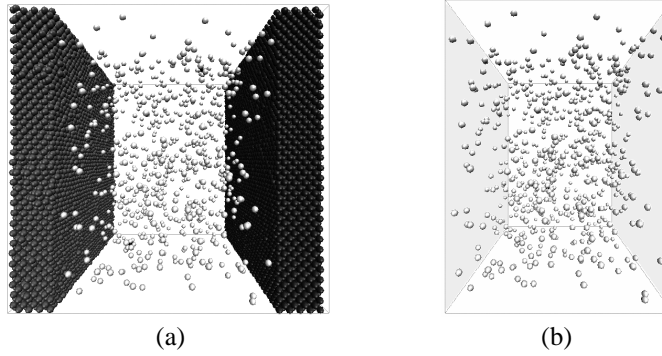
$$\frac{\partial F}{\partial t} + \xi \cdot \nabla F = J_E(F, F), \quad (5.3)$$

where  $F(x, \xi, t)$  is the one-particle distribution function of the molecular velocity  $\xi$ . The collision integral  $J_E(F, F)$  keeps the same binary structure of the corresponding Boltzmann term, but the colliding molecules occupy different positions in space and the collision frequency is modified by the factor  $Y$  which plays the role of an approximate pair correlation function. The  $Y$  function has the form:  $Y(\eta) = \frac{1}{2} \frac{2-\eta}{(1-\eta)^3}$ , where  $\eta$  is the reduced density.

Different MC particle simulation methods have been proposed to solve this equation. The first method described by Alexander et al. [Ale95, Ale97] is an attempt to bring into DSMC the spatial correlations which are absent in an ideal gas. This method encounters problems with boundary conditions when walls are introduced. A particle method for the numerical solution of the Enskog equation has been presented by Montanero and Santos [Mon96, Mon97] extending the scheme originally proposed by Nanbu [Nan86] for the Boltzmann equation. The method correctly reproduced the transport properties of the Enskog gas, but inherited from the original Nanbu scheme the feature of conserving momentum and energy only in a statistical way and not in a single collision. The particle method proposed by Frezzotti [Fre99] constructed for the Enskog equation in the spirit of the DSMC method exactly conserves momentum and energy. We use this latter method for our MC simulations of nano- and microchannels.

In this particle simulation method the molecules of the gas are replaced by a number of mathematical particles. Each particle is characterized by its position  $\mathbf{x}$  and its velocity  $\mathbf{v}$ . The particles are allowed to move in the flow field region and to collide, where the collision partners are selected from prescribed collision probabilities. First the particles are advected with their velocities and then collisions take place at each spatial location by dividing the domain into cells. In the case of the Boltzmann equation collisions are computed locally between particles belonging to the same cell. In case of the Enskog equation, because of the finite extent of a molecule, the particles in a given cell can also collide with particles located in neighboring cells.

Important macroscopic quantities that can be computed from such DSMC simulations



**Figure 5.10:** The model of a microchannel consists of two parallel infinite plates at distance  $L$  apart from each other, and gas molecules confined between these plates. Depending on the simulation method, the walls can be modeled **(a)** explicitly, or **(b)** using boundary conditions.

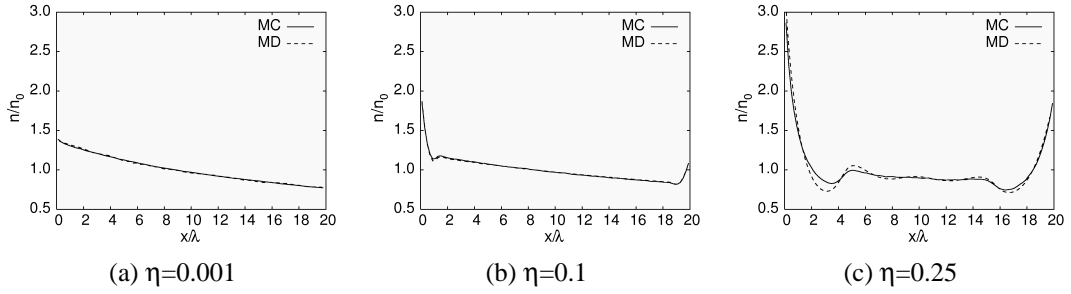
are the number density  $n$ , the mean velocity  $\mathbf{u}$ , the temperature  $T$ , the heat flux  $\mathbf{q}$  and the stress  $P$ . How these macroscopic quantities can be obtained from the particle distribution and pre-collision and post-collision velocities of the interacting molecules is described by Frezzotti [Fre97, Fre99].

### 5.3.2 The physical model

Our model to study the heat flow in nano- and microchannels using DSMC is similar to the model used before for MD. It consists once more of two parallel plates at a distance  $L$  apart from each other and of gas molecules confined between these two walls. In the MD simulations described before, the walls were simulated explicitly like in Figure 5.10a. However, since the direct simulation Monte Carlo method can only simulate gases, in this method the walls have to be described by the boundary conditions (e.g. thermal walls, specular walls, etc.). This is shown in Figure 5.10b by the two gray shaded walls. Periodic boundary conditions are used again in the directions parallel to the walls such that these walls form infinite plates. An advantage of using boundary conditions to model the walls is that the simulation contains much less particles, because of the absence of the wall particles. Furthermore, it now suffices to use a single channel, because both walls do not influence each other anymore as was the case for periodic boundaries, resulting in a further reduction of the number of particles in the simulation. However, for accurate simulations, all the boundary effects that we have seen before in the MD simulations with explicit walls have to be incorporated in these boundary conditions modeling the walls. As we will see in the next section this is not the case for standard thermal wall boundary conditions. But, because the development of new boundary conditions is not our goal here, we will use these thermal wall boundary conditions both for MD and DSMC to compare the results of MD and DSMC when using the same boundary conditions.

Both plates have again their own temperature,  $T_1$  and  $T_2$  respectively, where this temper-





**Figure 5.11:** Comparisons between MC and MD density profiles for different reduced gas densities  $\eta$  and wall temperatures  $T_1 = 1.0$  and  $T_2 = 2.0$ .

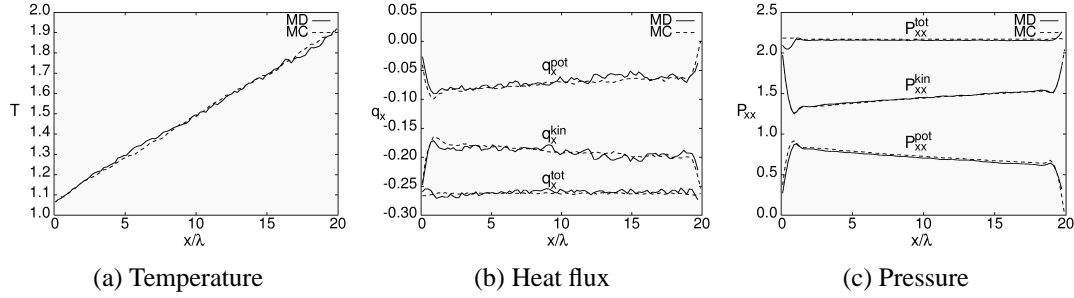
ature is uniform on the plate surface and constant in time. The gas consists of spherical particles of diameter  $a$ . The density of the gas can be expressed as  $n$ , being the number of particles per unit of volume, or using a reduced density  $\eta$ , which also takes the particle sizes into account and is related to the number density as  $\eta = \pi n a^3 / 6$  [Fre99]. The mean free path  $\lambda$  of the gas particles is related to this reduced density as  $\lambda = 1 / (\sqrt{2} \pi a^2 n Y(\eta))$ , where the  $Y$  factor is the pair correlation function as defined above. For a relatively dense gas with  $\eta \approx 0.1$ , the mean free path and the molecular diameter  $a$  have the same order of magnitude.

### 5.3.3 Comparison of MD and DSMC results

To compare MD and DSMC results, nanochannels with different gas densities are simulated both using MD and using DSMC, both using thermal wall boundary conditions to model the walls. The left wall is kept at a temperature  $T = 1.0 T^*$ , whereas the right wall is kept at a temperature  $T = 2.0 T^*$ . For the DSMC simulations we started from the 1D code by Bird [Bir94], which we adapted for dense gases according to Frezzotti's method [Fre97]. For the MD simulations, thermal wall boundary conditions have been added to our molecular dynamics code *PumMa*.

In Figure 5.11 the resulting density profiles of the MD and the DSMC simulations are compared for three different gas densities, ranging from a rarefied gas ( $\eta = 0.001$ ) to a dense gas ( $\eta = 0.25$ ). Comparison of the MD and MC results shows that the results are almost identical for low densities, but that for higher gas densities a larger and larger deviation is visible near the walls. Although the density profiles for both MC and MD have the same shape, differences appear in the peak region. These differences are an effect of the different collision mechanisms between the gas particles in both simulation methods as the boundary conditions are the same.

In Figure 5.12, also the MC and MD results for the temperature, heat flux and pressure profiles are compared for a gas density  $\eta = 0.1$ . The figure shows that the temperature, heat flux and the pressure are in good agreement as well. The total pressure and total heat flux should be independent of the position. The deviations for the MD results visible in the figure for the total pressure and the total heat flux near the walls are caused



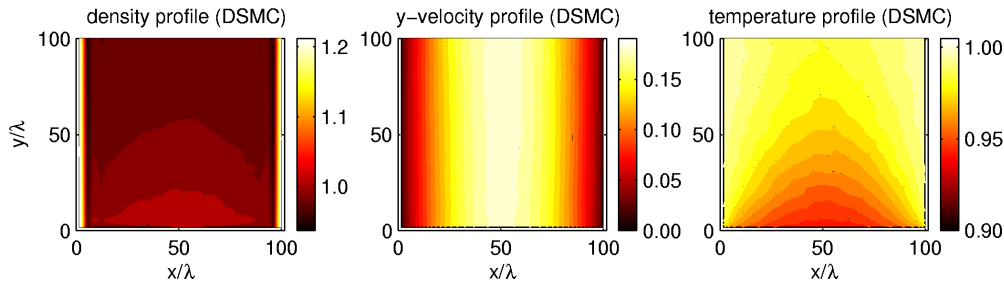
**Figure 5.12:** Comparisons between MC and MD simulation results for temperature, heat flux and pressure profiles for a reduced gas density  $\eta = 0.1$  and wall temperatures  $T_1 = 1.0$  and  $T_2 = 2.0$ .

because the interaction with the walls is not taken into account in the calculation of the pressure and heat flux.

Comparison of the MD and MC results shows that the results in the bulk are very similar. However, even though the same boundary conditions have been used, at the interface the results differ, especially for higher gas densities. From comparison with the first part of this chapter it is clear that when explicit walls are used in MD, the profiles differ even much more.

These differences are also evident when we study particle flow. For this purpose the MC code has been extended to a 2D version as well to which flow has been added in a way equivalent to the MD case. Results of such 2D simulations, with again a cold gas flow ( $T = 0.9T^*$ ) in the  $y$  direction between warm walls ( $T = 1.0T^*$ ), are shown in Figure 5.13. Comparison with Figure 5.9 where the MD results for an equivalent system (but with the walls simulated explicitly) were shown, shows large differences, which are especially visible in the resulting temperature profiles.

These comparisons show that the accuracy of the simulation results depends on how accurate we model the interactions between the gas molecules and between the gas and the wall and that one should be very careful with using simplified methods to study phenomena where interfaces play a key role, such as heat flow and velocity profiles that we have studied, since the behavior at these interfaces determines ultimately the



**Figure 5.13:** Two dimensional normalized density, flow-velocity and temperature distributions in the channel for a DSMC simulation of a gas with reduced density  $\eta = 0.213$ .

behavior of the whole system. Pure MC is thus not accurate enough to describe micro- and nanochannels. On the other hand MD simulations can be very accurate but, to model a complete microchannel, MD simulations are too slow. If the system to be studied is too large to be studied with molecular dynamics completely, hybrid methods could be used that couple molecular dynamics to another more efficient simulation technique, and use molecular dynamics there where necessary for accuracy and the other technique where possible.

## 5.4 Hybrid molecular dynamics-Monte Carlo

### 5.4.1 Coupling

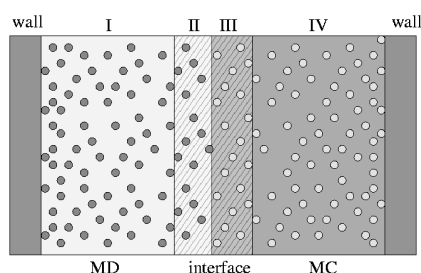
In order to perform more efficient, though accurate, simulations, we propose a simulation method that combines the advantages of the molecular dynamics and the Monte Carlo simulations, by simulating particles near the wall over the scale of the oscillation region using the molecular dynamics technique to obtain more accurate results near the wall, and the Monte Carlo technique for the particles in the bulk where it yields, as we have seen above, the same results as MD but at a much lower computational cost. This is achieved by dividing our simulation domain into subdomains, where we use either MD or MC, and a coupling between these subdomains.

In order to explain the method and to test it, we use in first instance two subdomains: one with MD and the other one with MC. Because the MD method needs information from the neighboring MC particles and vice versa, an interface coupling the two subdomains is built as shown in Figure 5.14. Our hybrid simulation method couples the MD and MC simulations through a buffer layer at the interface between the two domains. In Figure 5.14 we have labeled with (II) the buffer layer of MC subdomain (III+IV), and with (III) the buffer layer of the MD subdomain (I+II).

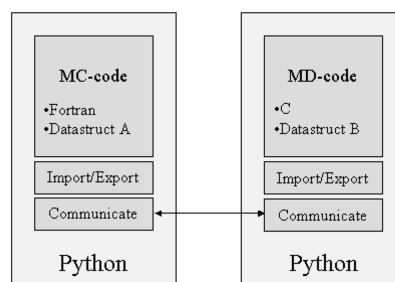
Our hybrid simulation algorithm consists of the following steps:

1. Generate an initial configuration, i.e. the initial positions and velocities for all particles in the whole domain.
2. Assign particles in region I and II to MD code, and in III and IV to MC code.
3. Send particle information of particles in region II to MC code, and of particles in region III to MD code.
4. Perform MD simulation of particles in I, II, and III, and MC simulation in II, III and IV, computing their new positions and velocities.
5. Restart from step 3.

Initially, the particles are randomly placed in the simulation domain, and the velocities of the particles are generated from a Maxwell-Boltzmann distribution. MC and MD simulations are updating the velocities and positions of the particles assigned to their subdomain in parallel using also the information from the buffer layer in the other subdomain. The time step size in MC depends on the mean free path, whereas in MD it is



**Figure 5.14:** The coupling of the MD and MC simulations is obtained via an interface layer.

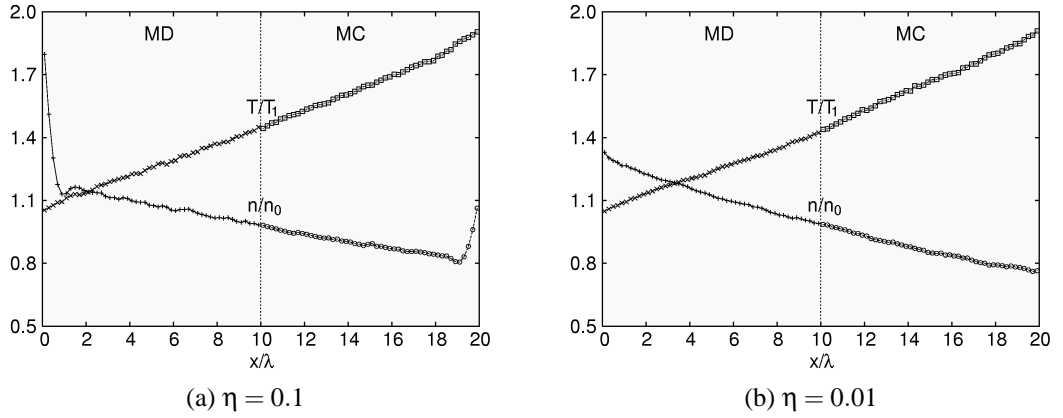


**Figure 5.15:** The MD and the MC code are coupled via a python interface.

independent of this mean free path but depends on the shape of the interaction potential. Because the time step in MD is usually small compared to the time step in MC, a number of MD time steps has to be performed for every single MC simulation step. For the case  $\eta = 0.1$ , ( $\lambda \approx a$ ), five MD steps and one MC step are used per iteration of the hybrid method to synchronize the physical time in both subdomains.

A straightforward approach to update particles in the interface layer (II+III) after each iteration is by allowing each subdomain to communicate the new positions and velocities of its own particles in the interface layer to the other subdomain. First we investigated the coupling between the two methods realized by importing and exporting particles from one subdomain to another. As we couple two simulation methods based on a different mechanism of computing the interactions between particles, problems were encountered as expected when trying to couple the less detailed method with the more accurate method. This is the case for coupling the MC and MD particle domains. Whereas for MD to MC particle coupling, particles from the MD domain could be directly imported into the MC domain using the exact positions and velocities (of course taking the proper unit conversions into account), this cannot be done for MC to MD particle coupling as in MC simulations particles can overlap each other. Imported into the MD domain, this results in very large energies and forces, leading to energy conservation problems and a temperature peak at the interface. Attempts to circumvent this problem by repositioning particles that overlap and by rescaling velocities to conserve the total energy did not solve the problems completely. Therefore, another approach has been used that uses macroscopic averages instead of single particles. Thus, for the MC to MD coupling, the MD boundary layer (III) is divided into subcells, and the average properties of the particles in the subcells are imported from the MC domain rather than updating the exact positions and molecular velocities.

To update the temperature in the MD buffer layer (III), particle velocities are rescaled according to the corresponding imported MC average temperature per subcell. Updating the density in the buffer layer (III) is more difficult because of the problems encountered with energy conservation when generating or removing particles in the MD subdomain. Therefore, we introduce a soft border for the MD buffer layer (III), such that this border

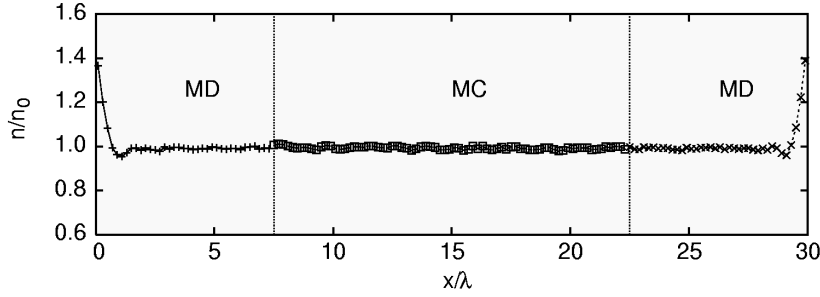


**Figure 5.16:** The density and temperature profiles in the channel as obtained from the hybrid MD-MC simulations for two different densities when  $T_2/T_1 = 2$ , and  $L = 20\lambda$ . The domain is split in two subdomains, the left one being MC (50%), and the right one MD (50%).

is shifted to the right if the density is decreasing and to the left if the density is increasing. For the MD to MC coupling, MD particle velocities and positions can be directly imported from the MD and updated in the boundary layer (II). The size of the boundary layer is usually around a few mean free paths.

Both for the MD and the MC method we already had optimized codes available; an MC simulator written in the programming language Fortran77 and an MD simulator written in C. In order to implement the hybrid method described above a new code could be written. However, because both simulators are implemented in different programming languages and use completely different data structures this implies rewriting at least one code completely. Another possibility is to couple the existing codes using an interface. For this purpose a higher order programming language can be used that provides easy communication and input/output routines, like the programming language Python. The idea behind this is separation of concerns: using a low level language for the computationally demanding parts and a more manageable language for the rest. To prevent performance loss Van Leeuwen [vL02] developed some additional Python communication routines.

Following this latter option of coupling the existing codes via a Python interface, our simulator consists of three parts; the MD code for the MD simulations, the MC code for the MC simulations, and an interface between the first two codes written in Python, coupling the MD and MC simulations. The way this is done has been depicted schematically in Figure 5.15. Both the MD code and the MC code have been wrapped in a Python layer. The MD and the MC code have been extended both with two subroutines to import and export data from and to that Python layer. This resulted in two separate components that can communicate with each other via their Python layers. The MD and MC components are independent, the MD and MC steps being computed by two different independent codes. Another advantage of this approach is that the MD and MC components, being independent, can reside on different computers, and can run in



**Figure 5.17:** The density profile in the channel for  $\eta = 0.1$ ,  $T_2 = T_1 = 1 T^*$ , and  $L = 30\lambda$  as obtained from a hybrid MD-MC simulation when the domain is split into three parts: two MD domains (25%) near the walls and one MC domain (50%) in the middle.

parallel being synchronized and coupled by the interface written in Python, such that this approach could also be used for grid computing.

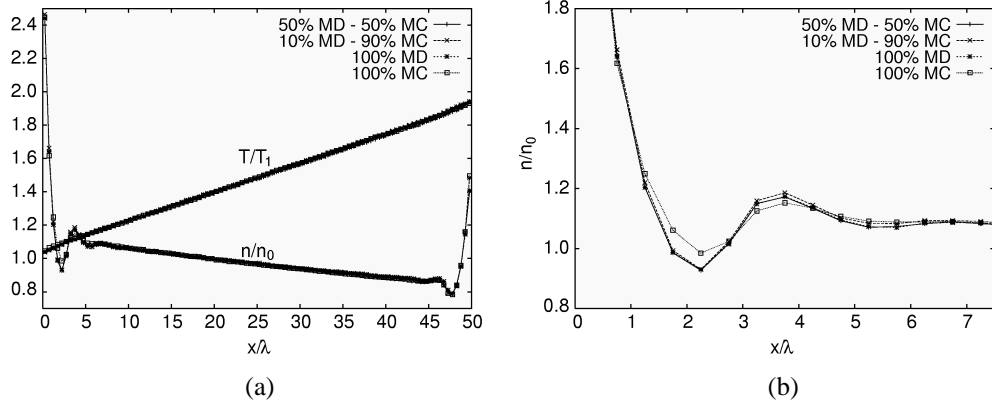
### 5.4.2 Hybrid MD-MC results

The applicability of our hybrid MD-MC simulation method is demonstrated by some simulations of the same channels as were used for the comparison between MD and MC. To start with, we split the channel into two equal parts, one half being the MD subdomain, and the other half the MC subdomain like in Figure 5.14. Thus in this first case there is one wall in the MD subdomain and one in the MC subdomain. Both these walls are again provided by the thermal wall boundary conditions such that the results can be compared with pure MC and pure MD simulations with the same boundary conditions. The temperature of the warm wall  $T_2$  is also again twice the temperature of the cold wall  $T_1 = 1.0 T^*$ . Figure 5.16 shows the hybrid MD-MC simulation results for the density and temperature profile in the channel when the width of the channel  $L = 20\lambda$  both for a dense ( $\eta = 0.1$ ) and for a more dilute gas ( $\eta = 0.01$ ).

The next step is to use MD near both channel walls and MC in the middle for the bulk. In our implementation this means adding one extra MD component, which can again run in parallel with the other two components. Figure 5.17 shows the density profile in the channel for such a simulation, for the case when both walls and the gas are at the same temperature  $T = 1 T^*$ .

### 5.4.3 Accuracy of hybrid simulations

The profiles resulting from the hybrid method look similar to the profiles from the pure MD and pure MC simulation. To prove that the hybrid method coupling MD and MC simulations yields accurate results a comparison between the results of pure MD and pure MC and two hybrid simulations with different subdomain divisions is made. In the first hybrid simulation MD is used for the left half and MC for the right half, whereas in the second simulation only in a small layer of 10% of the box near the left wall MD



**Figure 5.18:** Comparison of the density and temperature profiles for different methods (a) over the whole box and (b) zoomed in at the region near the interface.

is performed and MC in the rest. This is done for a dense gas with  $\eta = 0.2$  because for such a high density gas the pure MD and MC results differ even when, like here, the same boundary conditions are used. In Figure 5.18a the density and temperature profiles of the four simulations are compared. In part b of the same figure has been zoomed in onto the density profile near the wall with the MD subdomain. The profiles show the first 15% of the box. Thus, for the hybrid simulation with only 10% MD this includes the MD domain as well as part of the MC domain.

We quantify the accuracy of the simulation results for the number density  $n$ , by computing the deviations of the simulation results using the method  $mx$  from the pure MD simulation results that are considered to be the exact solution. The deviations are given by the relation  $\sqrt{\int dx (n_{mx}(x) - n_{MD}(x))^2} / \int dx n_{MD}(x)$ , where  $n_{mx}$  denotes the density results when using the simulation method  $mx$ . The  $mx$  simulation method can be the MC, MD, or hybrid MD-MC with different sizes of the MD and MC domains. When looking at the whole simulation box, the deviations of the MC simulation results are found to be around 0.8%, whereas they are around 0.3% for the hybrid method. When zooming in again at the first 15% of the box the difference is even more clear. The deviations of the MC simulation results are found to be around 1.9% and for the hybrid between 0.2% and 0.3%.

#### 5.4.4 Timings of hybrid simulations

Besides the accuracy of the methods also the required computational time is important. Therefore, we measured the simulation times for the system as described above consisting of 20000 particles. Comparisons between pure MD, pure MC and hybrid MD-MC simulation times are performed for simulations consisting of 500 combined MD-MC iterations. This is done for two different reduced densities:  $\eta = 0.1$  and  $\eta = 0.01$ . For a dense gas ( $\eta = 0.1$ ), the 500 combined iterations consisted of 500 MC steps and 2500 MD steps, while for a dilute gas ( $\eta = 0.01$ ) these iterations consisted of 500 MC steps and 30000 MD steps, because in this case 60 MD steps are needed per MC iteration.

timings	$\eta = 0.1$	$\eta = 0.01$
pure MC	2.4	2.3
pure MD	203.3	1169.6
MD (50%) - MC (50%)	246.5	1162.9
MD (10%) - MC (90%)	43.4	206.9

**Table 5.3:** Simulation times for 500 combined iterations and 20000 particles. For  $\eta = 0.1$ , simulation times are computed for 500 MC steps and 2500 MD steps. For ( $\eta = 0.01$ ), timings are done for 500 MC steps and 30000 MD steps.

For the hybrid simulations the same two situations were considered as before, i.e., the case where the simulation domain is equally divided between MD and MC and the case where the MD domain covers only 10% of the whole domain and 90% forms the MC domain. The timing results are presented in Table 5.3. We notice that the speedup when using the hybrid MD-MC method for 50% MD and 50% MC is very small when compared to pure MD simulations times, but this speedup increases drastically when the bulk is larger than the region near the wall. For example, when the MC domain is extended to 90% of the simulation domain and the MD domain reduced to 10% of the simulation domain, the speedup of the simulations increases roughly with a factor five.

### 5.4.5 Discussion of hybrid simulations

We coupled two particle simulation methods, namely molecular dynamics and the direct simulation Monte Carlo method and combined in this way in one hybrid method the advantages of both simulation methods. This method coupled MD for the accuracy near the boundary and MC for lower computational cost in the bulk. Resulting profiles and timing results demonstrated the methods feasibility.

The main argument for using the hybrid method so far has been the accuracy of the gas-surface interactions. This is especially the case since the walls can be simulated explicitly with MD instead of using boundary conditions, whereas the bulk gas is simulated using DSMC.

Another good argument in favor of the hybrid method is the possible extension to two-phased microchannel cooling. Namely, microchannel cooling performs extra good in case of a phase transition inside the channel, i.e., in case of evaporation of a liquid. Because such a phase transition can also be modeled with MD and not with MC, this could also be done with the hybrid method.

A next step in this approach of hybrid methods can be the coupling of MC with another MC simulation with larger particles, or the coupling of MC with a continuum approach with partial differential equations. In this way a stepwise transition of length and time scales can be made. In the presented approach such an extra coupling is realized by adding a new component.



## **5.5 Conclusion**

First, the dependence of the wettability on the heat flow in a nanochannel was studied using MD. This was done for both rarefied and dense gases and both for a static and a flowing gas. These simulations show that the type and strength of the gas-surface interaction is determining the behavior of the whole system for heat transfer as well as for flow profiles. However, MD is computationally expensive such that only very small nanochannels can be studied. In order to study larger systems we have to resort to other methods. However, because the interfaces are so important, it is vulnerable to use a method where these gas-surface interfaces are described using boundary conditions. This is shown by comparisons of MC and MD simulation results. When using the same boundary conditions, MD and MC results are alike, especially in the bulk, although they differ near the boundaries for high gas densities. But, these results differ from the prior more accurate MD simulations where the walls were modeled explicitly and where the behavior near the wall is strongly influenced by the strength of the gas-wall interactions such that density peaks can appear even in the case of a dilute gas.

To circumvent this problem we developed a hybrid method by coupling two particle simulation methods, namely molecular dynamics and direct simulations Monte Carlo. This method couples MD simulations near the walls for the accuracy of the interaction with the wall, and MC in the bulk where it yields the same results at much lower computational costs. The accuracy of the hybrid simulation results were tested by comparing them with the pure MD and pure MC simulation results. Also comparisons of the computing times for these different techniques were given. These results showed that the hybrid MD-MC simulation results are very accurate compared to the pure MC simulation results and the hybrid simulations are fast compared to the pure MD simulation times. Now the coupling of MD and MC has been shown, the next step would be to model the walls again explicitly in MD and the bulk gas using DSMC, or to couple MD with different techniques.

# 6

## Concluding remarks

---

***I*n this thesis, the development and application of various molecular simulation methods have been described. Here we shortly summarize results of the separate parts, emphasize the relation between the different subjects and give some directions how the connection between these subjects can be exploited for further application of molecular simulations.**

## Concluding remarks

---

In this thesis the development of a framework consisting of various molecular simulation approaches and its application to a broad range of physical, chemical and biological inspired cases has been described, where every approach is used in a trade-off between the size of the system, the length and time scale of the internal processes, and the required level of detail and accuracy for reasonable computation time.

In Chapter 2 we developed an electronic structure calculation framework based on wavelets and proved with the aid of analytically solvable cases that it works and yields accurate results. From comparison between the use of different types and orders of wavelets and different preconditioners, the interpolant 6 wavelet in combination with a Modified Inverse Cholesky preconditioner was found to yield accurate results at the lowest computational expense. These cases also perfectly demonstrated the power of the wavelet basis. By introducing extra levels of fine wavelet basis functions only in the regions where needed, a very cost effective way is provided to enable the inclusion of extra high resolution for local detail where necessary instead of the need to add this resolution over the whole domain. A good example of this was provided by the case of the potential well where extra resolution around the discontinuities in the potential tremendously increased the accuracy with hardly any additional computational cost. Thereafter, the code was successfully used to study real cases, namely 2D quantum dots and small 3D molecular systems. In the 3D molecular systems, the effects of different choices for pseudopotentials and exchange-correlation potentials were studied on properties such as ionization potentials, electron affinities, excitation energies, binding energies, bond frequencies, force constants, and bond lengths of atoms and small molecules. With increasing complexity of the potentials, i.e., going from the local density approximation via the local spin density approximation to generalized gradient approximations, the results compare increasingly well with experimental data and electronic structure calculations with other basis sets, proving the suitability of the code for this class of problems. In order to study the electronic states of 2D quantum dots as well, a two-dimensional version of Chelikowsky's direct integration method has been developed to evaluate the Hartree potential. The correctness of this 2D version was proved as for single quantum dots the orbitals and the filling of the shells agree with results in literature for the case of a parabolic model potential for the quantum dots. However, because such an infinite potential is unphysical, we introduced a new potential which we derived from the description of a quantum dot as a 2D jellium. Calculation of the orbitals of such quantum dots resulted in filling of the lowest energy orbitals as for the parabolic potential, whereas the highest energy orbitals are closer to those expected for 2D atoms. By laterally coupling quantum dots, what can be done using the newly developed potential straightforwardly, quantum dot molecules have been modeled. The shell filling for two laterally coupled quantum dots has been calculated as a function of the inter center distance. This possibility to couple quantum dots also enables the study of larger quantum dot arrays.

From the electronic structure also forces between atoms can be calculated that can be

used to study dynamics. However, for many applications the electronic structure does not need to be calculated to study the dynamics. Instead, it suffices to use empirical potentials to describe the interactions between these atoms, allowing for the simulation of much larger systems. In this way the influence of surface wettability on static and flowing gases in nanochannels has been successfully studied in Chapter 5. By systematically changing the interaction potential between the gas and the wall atoms, the influence of the wettability on particle and heat flow were studied in a molecular dynamics simulation where the walls were modeled explicitly. These simulations showed that the strength of the gas-wall interaction determines the extent of layering of gas particles against the wall, and that, whereas such an increased layering seems to have no effect on the thermal conductivity at a solid-liquid interface, it has an effect on the thermal conductivity of the solid-gas interface. Furthermore, these simulations showed that the type and strength of the gas-surface interaction determines the flow profiles and that the effects at the interface are crucial for the behavior of the whole nanochannel.

In other cases even coarse grained models can be used where the average behavior of several atoms is described by a single coarse grained particle instead of every atom separately. In Chapter 3 this was applied for lipids, for which a coarse grained lipid model was developed. In this model we refrained from as much detail as possible while maintaining realistic behavior. The model is as simple as possible while maintaining all essential features. Instead of all approximately 130 atoms, it consists of only 12 particles of two different types, four hydrophilic head group particles and two tails of four hydrophobic tail particles each. A third particle type is used for water molecules that form the solvent. Three potentials are used to describe the interactions between the particles; one for atoms that are bonded to each other, one for between all other pairs, and a bending potential to add additional rigidity to the lipids. The suitability of the model was shown as the expected aggregation behavior was obtained as well as bilayer properties comparable with experimental results. Using this coarse grained model, molecular dynamics simulations showed spontaneous bilayer and vesicle formation. These simulations confirmed the pathway of vesicle formation as suggested in literature from experimental studies. The lipids first rapidly aggregate into micelles and small bilayers, also called bicelles. Subsequently, the micelles and bicelles slowly merge into a larger aggregate, being a disc-like bilayer, which finally bends to form a vesicle. Further analysis of these simulations showed that for such lipid membranes the potential energy of the membrane as well as the potential energy of the whole system increases during the bilayer-vesicle transition. The reason that the transition from flat bilayers to vesicles does take place is that for sufficiently large bilayers, the solvent accessible surface of the bilayer decreases during the transition, resulting in fewer water molecules near the membrane and more in the bulk, and a higher entropy of the water. The bilayer-vesicle transition is thus an entropy driven mechanism. Furthermore, it was shown that the same modeling technique can be used to analyze more complex processes such as vesicle fusion. Another step could be the development of a coarse

## Concluding remarks

---

grained model for proteins, such that biomembranes including proteins can be studied. In Chapter 4 we showed that the limitation of molecular dynamics of having a fixed molecular composition can be overcome by extending the above described framework with stochastic chemical reactions. A novel framework for molecular simulations was developed which combines coarse grained molecular dynamics with (artificial) chemical reactions. In this method, such a reaction is a switch from one potential describing the interaction between two particles to another potential or a change of the parameters of the potential. The occurrence of reactions is determined stochastically taking into account the surrounding of every particle. For example, two particles that are within a certain distance of each other form a bond with a certain chance and this chance can be increased by the presence of catalysts in the neighborhood. As the surroundings of each particle is already inspected in molecular dynamics to obtain the forces on the particles, this approach hardly results in any additional computational costs and has the advantage that this method enables, apart from bond forming and breaking reactions, also transitions of particle types. This technique was applied on some biologically inspired cases related to the origin of life using an even simpler lipid model where the two tails are no longer present separately but which results in simple chemical reactions. With lipid forming and auto-catalytic reactions the formation of membranes, micelles, and unilamellar and multilamellar vesicles, as well as healing, growth and bursting of the latter were shown. So proving the feasibility and usefulness of adding such stochastic reactions to coarse grained molecular dynamics method, but the method could also be coupled to atomistic molecular dynamics in the same way.

In Chapter 5 it was also shown that the nanochannels that were studied with molecular dynamics could also be simulated using other particle methods, like the direct simulation Monte Carlo method, that are less time consuming than molecular dynamics, but that these methods are not always accurate enough. For example for our nanochannels, these methods can not explicitly model the walls but describe these walls using boundary conditions. As was shown, the influence of the gas-surface interface is so important that it is inaccurate to use a method where these interfaces are described using thermal wall boundary conditions instead of an explicitly simulated wall. However, for the bulk, i.e., away from the walls, the method turned out to be accurate enough. Subsequently it was shown that this latter could be exploited by coupling molecular dynamics to this direct simulation Monte Carlo method as the results obtained with this hybrid method, coupling molecular dynamics for accuracy near the walls with the direct simulation Monte Carlo method for efficiency in the bulk gas, compare well with results from pure molecular dynamics simulations but at lower computational cost.

As we have seen, each of the simulation techniques has its own range of applicability. Especially for many biologically interesting problems, the system size of the problem is larger than the level of detail needed allows for in reasonable computational time. One possibility is to speed up the method. In Chapter 2 a good example of such an exertion was described. But most currently available simulation techniques have already been

developed so far that further optimization is hardly possible. However, the same feature that was exploited for electronic structure calculations using wavelets, namely the fact that for many physical, chemical and biological problems the finest level of detail is often only needed in part of the domain, can also be exploited at other length scales. Namely, for many problems different techniques could be combined in a hybrid method so optimizing for a second time on the trade-off between the size of the system, the length and time scale of the internal processes, the level of detail and accuracy required for a reasonable computation time. In such hybrid methods the more detailed method will only be used in the area where fine details are needed and the less detailed method elsewhere, so enabling again the simulation of larger systems. We already demonstrated such a coupling of molecular dynamics with the direct simulation Monte Carlo method, but there are possibilities at all length scales. Starting at the highest length scale, coarse grained molecular dynamics could be coupled with continuum theory. A nice application of this would be our lipid simulations, where coarse grained molecular dynamics simulation of the lipids could be combined with a continuum treatment of the solvent further away from the bilayer, which for our simulations exhaust a large part of the computation time. Going to smaller length scales, another option is to couple coarse grained molecular dynamics with fully atomistic molecular dynamics. An application of this could be a coarse grained molecular dynamics simulation of a lipid bilayer with fully atomistic treatment of a protein in that bilayer. And again one step further to more detail, fully atomistic molecular dynamics simulations where part of the forces is calculated quantum mechanically. For example for the molecular dynamics simulation of a large protein, where the docking site of the protein is evaluated quantum mechanically. Chapters 4 and 5 demonstrated that it is possible to combine different techniques. Further development of such hybrid simulations will further increase the possibilities of computer simulations. Especially as in the mean time experimental techniques also develop further, reaching smaller and smaller systems with better and better time resolution, molecular simulations and experiments start to approach each other such that, where validation of molecular simulations is now often still difficult, molecular simulations and experiments will go hand in hand increasingly.



# Bibliography

- Ada00** J. Adamowski, M. Sobkowicz, B. Szafran, and S. Bednarek. *Electron pair in a Gaussian confining potential*. Phys. Rev. B, 62:4234–4237, 2000.
- Alb89** B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J.D. Watson. *Molecular Biology of the Cell*. Garland, New York, 1989.
- Ald57** B.J. Alder and T.E. Wainwright. *Phase transition for a hard sphere system*. J. Chem. Phys., 27:1208–1209, 1957.
- Ald59** B.J. Alder and T.E. Wainwright. *Studies in molecular dynamics. I. General method*. J. Chem. Phys., 31:459–466, 1959.
- Ale95** F.J. Alexander, A.L. Garcia, and B.J. Alder. *A consistent Boltzmann algorithm*. Phys. Rev. Lett., 74(26):5212–5215, 1995.
- Ale97** F.J. Alexander, A.L. Garcia, and B.J. Alder. *The consistent Boltzmann algorithm for the van der Waals equation of state*. Physica A, 240:196–201, 1997.
- All87** M.P. Allen and D.J. Tildesley, editors. *Computer Simulation of Liquids*. Oxford University Press, 1987.
- Ari99** T.A. Arias. *Multiresolution Analysis of Electronic Structure - Semicardinal and Wavelet Bases*. Rev. Mod. Phys., 71:267–311, 1999.
- Aus98** D.G. Austing, T. Honda, K. Muraki, Y. Tokura, and S. Tarucha. *Quantum dot molecules*. Physica B, 249-251:206–209, 1998.
- Bac82** G.B. Bachelet, D.R. Hamann, and M. Schlüter. *Pseudopotentials that work*. Phys. Rev. B, 26:4199–4228, 1982.
- Bar73** J. Barojas, D. Levesque, and B. Quentrec. *Simulation of Diatomic Homonuclear Liquids*. Phys. Rev. A, 7:1092–1105, 1973.
- Bar94** R. Barrett, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the solution of linear systems: Building blocks for iterative methods*. SIAM, 1994.
- Bar03** J. Barrat and F. Chiaruttini. *Kapitza resistance at the liquid-solid interface*. Mol. Phys., 101:1605–1610, 2003.
- Bec88** A.D. Becke. *Density-functional exchange-energy approximation with correct asymptotic behavior*. Phys. Rev. A, 38:3098–3100, 1988.
- Ben95** S.C. Benjamin and N.F. Johnson. *Entangled electronic states in multiple-quantum-dot systems*. Phys. Rev. B, 51:14733–14736, 1995.
- Ben96** M. Benzi, C.D. Meyer, and M. Tuma. *A sparse approximate inverse preconditioner for the conjugate gradient method*. SIAM J. Sci. Comput., 17:1135–1149, 1996.
- Ber97** O. Berger, O. Edholm, and F. Jahnig. *Molecular dynamics simulations of a fluid bilayer of dipalmitoylphosphatidylcholine at full hydration, constant pressure, and constant temperature*. Biophys. J., 72:2002–2013, 1997.
- Bey92** G. Beylkin. *On the representation of operators in bases of compactly supported wavelets*. SIAM J. Numer. Anal., 6:1716–1740, 1992.
- Bin79** K. Binder. In *Monte Carlo Methods in Statistical Physics*, volume 7 of *Topics in Current Physics*. Springer, Berlin, 1979.
- Bir94** G.A. Bird. *Molecular Gas Dynamics and the Direct Simulations of Gas Flows*. Clarendon Press, Oxford, 1994.
- Bre98** J. Breyer, J. Ackermann, and J. McCaskill. *Artificial Chemistries - A Review*. Artif. Life, 4:25–40, 1998.
- Bro83** B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, and M. Karplus.



## Bibliography

---

- CHARMM: a program for macromolecular energy, minimization and dynamics calculations.* J. Comp. Chem., 4:187–217, 1983.
- Bro88** C.L. Brooks, M. Karplus, and B.M. Pettitt. *Proteins: A theoretical perspective on Structure, Dynamics and Thermodynamics.* Adv. Chem. Phys., 71:1–259, 1988.
- Bur98** C.S. Burrus, R.A. Gopinath, and H. Guo. *Introduction to Wavelets and Wavelet Transforms.* Prentice Hall, 1998.
- Car87** R. Car and M. Parrinello. *The unified approach to density functional and molecular dynamics in real space.* Solid State Commun., 62:403–405, 1987.
- Che94** J.R. Chelikowsky, N. Troullier, K. Wu, and Y. Saad. *Higher-order finite-difference pseudopotential method: An application to diatomic molecules.* Phys. Rev. B, 50:11355–11364, 1994.
- Chi93** V. Chirita, B.A. Pailthorpe, and R.E. Collins. *Molecular dynamics study of low-energy Ar scattering by the Ni(001) surface.* J. Phys. D: Appl. Phys., 26:133–142, 1993.
- Cho93** K. Cho, T.A. Arias, J.D. Joannopoulos, and P.K. Lam. *Wavelets in Electronic Structure Calculations.* Phys. Rev. Lett., 71:1808, 1993.
- Chu92** C.K. Chui, editor. *Wavelets: A Tutorial in Theory and Applications.* Academic Press, 1992.
- Cie00** M. Cieplak, J. Koplik, and J.R. Banavar. *Molecular dynamics of flows in the Knudsen regime.* Physica A, 287:153–160, 2000.
- Cie01** M. Cieplak, J. Koplik, and J.R. Banavar. *Boundary Conditions at a Fluid-Solid Interface.* Phys. Rev. Lett., 86:803–806, 2001.
- Coh92** A. Cohen, I. Daubechies, and J.C. Feauveau. *Biorthogonal bases of compactly supported wavelets.* Commun. Pur. Appl. Math., 45:485–560, 1992.
- Coh93** A. Cohen, I. Daubechies, and P. Vial. *Wavelets on the Interval and Fast Wavelet Transforms.* Appl. Comput. Harmon. A., 1:54–81, 1993.
- Dau88** I. Daubechies. *Orthonormal basis of compactly supported wavelets.* Commun. Pur. Appl. Math., 41:909–996, 1988.
- Dau92a** I. Daubechies. *Ten Lectures on Wavelets.* SIAM, 1992.
- Dau92b** I. Daubechies. *Ten Lectures on Wavelets*, page 166. SIAM, 1992.
- Dau93a** I. Daubechies. *Two recent results on wavelets: wavelet bases for the interval, and biorthogonal wavelets diagonalizing the derivative operator.* In L. Schumaker and G. Webb, editors, *Topics in Theory and Applications of Wavelets*, pages 237–257. Academic Press, 1993.
- Dau93b** I. Daubechies. *Wavelets on the interval.* In Y. Meyer and S. Roques, editors, *Progress in Wavelet Analysis and Applications*, pages 95–107. Editions Frontieres, 1993.
- Des89** G. Deslaurier and S. Dubuc. *Symmetric Iterative Interpolation Process.* Constr. Approx., 5:49–68, 1989.
- dF00** S. de Filippo and M. Salerno. *Spectral properties of a model potential for quantum dots with smooth boundaries.* Phys. Rev. B, 62:4230–4233, 2000.
- Dit01** P. Dittrich, J. Ziegler, and J. Banzhaf. *Artificial Chemistries - A Review.* Artif. Life, 7(3):225–275, 2001.
- Don92** D.L. Donoho. *Interpolating Wavelet Transforms.* Preprint, Department of Statistics, Stanford university, 1992.
- Dro91** J.M. Drouffe, A.C. Maggs, and S. Leibler. *Computer Simulations of Self-Assembled Membranes.* Science, 254:1353–1356, 1991.
- Dua98** Y. Duan and P.A. Kollman. *Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution.* Science, 282:740–744, 1998.
- dV04** A.H. de Vries, A.E. Mark, and S.J. Marrink. *Molecular Dynamics simulation of the spontaneous formation of a small DPPC vesicle in atomistic detail.* J. Am. Chem. Soc., 126:4488–4489, 2004.
- Eas04** J.A. Eastman, S.R. Phillpot, S.U.S. Choi, and P. Keblinski. *Thermal Transport in Nanofluids.* Annu. Rev. Mater. Res., 34:219–246, 2004.
- Egb88** E. Egberts and H.J.C. Berendsen. *Molecular dynamics simulation of a smectic liquid crystal with atomic detail.* J. Chem. Phys., 89:3718–3732, 1988.
- Ens22** D. Enskog. *Kinetische Theorie der Waerme Leitung, Reibung and Selbstdiffusion in Gewissen*

- Verdichten Gasen und Fluessigkeiten.* Kungl. Svenska Veteenskapakad. Handl., 63:3–44, 1922.
- Ess93** K. Esselink, B. Smit, and P.A.J. Hilbers. *Efficient Parallel Implementation of Molecular Dynamics on a Toroidal Network. Part I. Parallelizing Strategy.* J. Comp. Phys., 106:101–107, 1993.
- Far96** K.D. Farquhar, M. Misran, B.H. Robinson, D.C. Steytler, P. Morini, P.R. Garrett, and J.F. Holzwarth. *The kinetics and mechanism of micelle-vesicle transitions in aqueous solution.* J. Phys.-Condens Mat., 8:9397–9404, 1996.
- Far03** O. Farago. *“Water free” computer model for fluid bilayer membranes.* J. Chem. Phys., 119:596–605, 2003.
- Fer27** E. Fermi. *Un metodo statistico per la determinazione di alcune proprieta dell atomo.* Rend. Accad. Naz. Lincei, 6:602–607, 1927.
- Fis96** G.S. Fishman. *Monte Carlo Concepts, Algorithms and Applications.* Springer Verlag, 1996.
- Fre97** A. Frezzotti. *A particle scheme for the numerical solution of the Enskog equation.* Phys. Fluids, 9:1329–1335, 1997.
- Fre99** A. Frezzotti. *Monte Carlo simulation of the heat flow in a dense hard sphere gas.* Eur. J. Mech. B-Fluid., 18:103–119, 1999.
- Fre02** D. Frenkel and B. Smit, editors. *Understanding Molecular Dynamics.* Academic Press, 2002.
- Fri04** A.J.H. Frijns, S.V. Nedea, A.J. Markvoort, A.A. van Steenhoven, and P.A.J. Hilbers. *Molecular Dynamics and Monte Carlo Simulations for Heat Transfer in Micro and Nano-channels.* Lect. Notes Comput. Sc., 3039:661–666, 2004.
- Fro83** P. Fromherz. *Lipid-Vesicle Structure: Size Control by Edge-Active Agents.* Chem. Phys. Lett., 94:259–266, 1983.
- Fuj96** M. Fujito, A. Natori, and H. Yasunaga. *Many-electron ground states in anisotropic parabolic quantum dots.* Phys. Rev. B, 53:9952–9958, 1996.
- Gar94** A.L. Garcia. *Numerical Methods for Physics.* Prentice Hall, Englewood Cliffs, NJ, chap. 10, 1994.
- Gas74** S. Gasiorowicz. *Quantum physics.* John Wiley & Sons, 1974.
- Gib60** J.B. Gibson, A.N. Goland, M. Milgram, and G.H. Vineyard. *Dynamics of Radiation Damage.* Phys. Rev., 120:1229–1253, 1960.
- Goe98** R. Goetz and R. Lipowsky. *Computer simulations of bilayer membranes: Self-assembly and interfacial tension.* J. Chem. Phys., 108:7397–7409, 1998.
- Goe99** R. Goetz, G. Gompper, and R. Lipowsky. *Mobility and Elasticity of Self-Assembled Membranes.* Phys. Rev. Lett., 82:221–224, 1999.
- Gol96** G.H. Golub and C.F. van Loan. *Matrix computations.* Johns Hopkins University Press, Baltimore, 1996.
- Gra03** M. Gradzielski. *Vesicles and vesicle gels – structure and dynamics of formation.* J. Phys.-Condens Mat., 15:R655–R697, 2003.
- Gro97** R.D. Groot and P.B. Warren. *Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation.* J. Chem. Phys., 107:4423–4435, 1997.
- Gua96** P. Guan, D.R. McKenzie, and B.A. Pailthorpe. *MD simulations of Ag film growth using the Lennard-Jones potential.* J. Phys.-Condens Mat., 8:8753–8762, 1996.
- Hal75** T. Halicioglu and G.M. Pound. *Calculation of potential energy parameters from crystalline state properties.* Phys. Stat. Sol. A, 30:619–623, 1975.
- Har68** G.D. Harp and B.J. Berne. *Linear and angular momentum autocorrelation functions in diatomic liquids.* J. Chem. Phys., 49:1249–1254, 1968.
- Har98** C. Hartwigsen, S. Goedecker, and J. Hutter. *Relativistic separable dual-space Gaussian pseudopotentials from H to Rn.* Phys. Rev. B, 58:3641–3662, 1998.
- Heh86** W.J. Hehre, L. Radom, P.v.R. Schleyer, and J.A. Pople. *Ab Initio Molecular Orbital Theory.* John Wiley & Sons, 1986.
- Her84** J. Hermans, H.J.C. Berendsen, W.F. van Gunsteren, and J.P.M. Postma. *A Consistent Empirical Potential for Water-Protein Interactions.* Biopolymers, 23:1513–1518, 1984.
- Hin00** K. Hinsen. *The molecular modeling toolkit: A new approach to molecular simulations.* J.

## Bibliography

---

- Comp. Chem., 21:79–85, 2000.
- Hir99** K. Hirose and N.S. Wingreen. *Spin-density-functional theory of circular and elliptical quantum dots*. Phys. Rev. B, 59:4604–4607, 1999.
- Hoh64** P. Hohenberg and W. Kohn. *Inhomogeneous electron gas*. Phys. Rev., 136:B864–B871, 1964.
- Hoo75** W.G. Hoover and W.T. Ashurst. *Theoretical chemistry: advances and perspectives*, volume 1, chapter Nonequilibrium molecular dynamics, pages 1–51. Academic Press, New York, 1975.
- Hoo83** W.G. Hoover. *Nonequilibrium molecular dynamics*. Ann. Rev. Phys. Chem., 34:103–127, 1983.
- Hoo92** P.J. Hoogerbrugge and J.M.V.A. Koelman. *Simulating microscopic hydrodynamics phenomena with dissipative particle dynamics*. Europhys. Lett., 19:155–160, 1992.
- Hub98** B.B. Hubbard. *The World according to Wavelets*. A K Peters, 1998.
- Isr91** J.N. Israelachvili. *Intermolecular and Surface Forces*. Academic Press, London, 1991.
- Jac98** L. Jacak, P. Hawrylak, and A. Wójs. *Quantum Dots*. Springer, 1998.
- Jah02** R. Jahn and H. Grubmüller. *Membrane fusion*. Curr. Opin. Cell Biol., 14:488–495, 2002.
- Jam93** L. Jameson. *On the Daubechies-Based Wavelet Differentiation Matrix*. Technical Report TR-93-95, ICASE, 1993.
- Jam96** L. Jameson. *The Differentiation Matrix for Daubechies-Based Wavelets on the Interval*. SIAM J. Sci. Comput., 17:498–516, 1996.
- Jem97** P. Jemmer and P.J. Knowles. *Generation of functional derivatives in Kohn-Sham density-functional theory*. Comput. Phys. Commun., 100:93–98, 1997.
- Jem98** P. Jemmer and P.J. Knowles. *Symbolic Algebra in Functional Derivative Potential Calculations*. J. Comput. Chem., 19:300–307, 1998.
- Joh95** N.F. Johnson. *Quantum dots: few-body, low dimensional system*. J. Phys.-Condens Mat., 7:965–989, 1995.
- Kal99** L. Kal, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, N. Krawetz, J. Phillips, A. Shinozaki, K. Varadarajan, and K. Schulten. *NAMD2: Greater scalability for parallel molecular dynamics*. J. Comput. Phys., 151:283–312, 1999.
- Kar90** M. Karplus and G.A. Petsko. *Molecular dynamics simulations in biology*. Nature, 347:631–639, 1990.
- Kim02** T. Kimura and S. Maruyama. *Molecular dynamics simulation of water droplet in contact with platinum surface*. 12th Int. Heat Transf. Conf., pages 537–542, 2002.
- Kin96** P.K.J. Kinnunen. *On the molecular-level mechanisms of peripheral protein-membrane interactions induced by lipids forming inverted non-lamellar phases*. Chem. Phys. Lipids, 81:151–166, 1996.
- Kle82** L. Kleinman and D.M. Bylander. *Efficacious form of model pseudopotentials*. Phys. Rev. Lett., 48:1425–1428, 1982.
- Koh65** W. Kohn and L.J. Sham. *Self-consistent equations including exchange and correlation effects*. Phys. Rev., 140:A1133–A1138, 1965.
- Kox80** A.J. Kox, J.P.J. Michels, and F.W. Wiegeler. *Simulation of a lipid monolayer using molecular dynamics*. Nature, 287:317–319, 1980.
- Koz83** M.M. Kozlov and V.S. Markin. *Possible mechanism of membrane fusion*. Biofizika, 28:242–247, 1983.
- Koz02a** Y. Kozlovsky and L.V. Chernomordik M.M. Kozlov. *Lipid intermediates in membrane fusion: formation, structure, and decay of hemifusion diaphragm*. Biophys. J., 83:2634–2651, 2002.
- Koz02b** Y. Kozlovsky and M.M. Kozlov. *Stalk Model of Membrane Fusion: Solution of Energy Crisis*. Biophys. J., 82:882–895, 2002.
- Koz03** Y. Kozlovsky and M.M. Kozlov. *Membrane Fission: Model for Intermediate Structures*. Biophys. J., 85:85–96, 2003.
- KS91** R.D. King-Smith, M.C. Payne, and J.S. Lin. *Real-space implementation of non-local pseudopotentials for first principles total-energy calculations*. Phys. Rev. B, 44:13063–13066, 1991.
- Kum90** A. Kumar, S.E. Laux, and F. Stern. *Electron states in a GaAs quantum dot in a magnetic field*. Phys. Rev. B, 42:5166–5175, 1990.

- Kuz01** P.I. Kuzmin, J. Zimmerberg, Y.A. Chizmadzhev, and F.S. Cohen. *A quantitative model for membrane fusion based on low-energy intermediates*. P. Natl. Acad. Sci. USA, 98:7235–7240, 2001.
- Lad86** A.J.C. Ladd, B. Moran, and W.G. Hoover. *Lattice thermal conductivity: A comparison of molecular dynamics and anharmonic lattice dynamics*. Phys. Rev. B, 34:5058–5064, 1986.
- Lee98** I. Lee, V. Rao, R.M. Martin, and J. Leburton. *Shell filling of artificial atoms within density-functional theory*. Phys. Rev. B, 57:9035–9042, 1998.
- Lei03** G. Lei and R.C. MacDonald. *Lipid Bilayer Vesicle Fusion: Intermediates Captured by High-Speed Microfluorescence Spectroscopy*. Biophys. J., 85:1585–1599, 2003.
- Len02** J. Leng, S.U. Egelhaaf, and M.E. Cates. *Kinetic pathway of spontaneous vesicle formation*. Europhys. Lett., 59:311–317, 2002.
- Len03** J. Leng, S.U. Egelhaaf, and M.E. Cates. *Kinetics of the Micelle-to-Vesicle Transition: Aqueous Lecithin-Bile Salt Mixtures*. Biophys. J., 85:1624–1646, 2003.
- Lid93** D.R. Lide and H.P.R. Frederikse, editors. *CRC Handbook of Chemistry and Physics*. CRC Press, 1993.
- Lip95** R. Lipowsky and E. Sackmann, editors. *Structure and dynamics of membranes*. Elsevier, Amsterdam, 1995.
- Lip98** R.A. Lippert, T.A. Arias, and A. Edelman. *Multiscale computations with interpolating scaling functions*. J. Comput. Phys., 140:278–310, 1998.
- Lui99** P.L. Luisi, P. Walde, and T. Oberholzer. *Lipid vesicles as possible intermediates in the origin of life*. Curr. Opin. Colloid In., 4:33–39, 1999.
- Luk98** J.J. Lukkien, J.P.L. Segers, P.A.J. Hilbers, R.J. Gelten, and A.P.J. Jansen. *Efficient Monte Carlos Methods for the Simulation of Catalytic Surface Reactions*. Phys. Rev. E, 58:2598–2610, 1998.
- Luk04** D.B. Lukatsky and D. Frenkel. *Multiple stalk formation as a pathway of defect-induced membrane fusion*. Eur. Phys. J. E, 14:3–6, 2004.
- Mad03** D. Madina, N. Ono, and T. Ikegami. *Cellular Evolution in a 3D Lattice Artificial Chemistry*. In *Proceedings of 7th European Conference on Artificial Life (ECAL '03)*, pages 59–68. Springer, 2003.
- Maj00** A. Majumdar. *Nanoengineering: Promises and Challenges*. Micro. Thermophys. Eng., 4:77–82, 2000.
- Mak90** P.A. Maksym and T. Chakraborty. *Quantum Dots in a Magnetic Field: Role of Electron-Electron Interactions*. Phys. Rev. Lett., 65:108–111, 1990.
- Mal89** S.G. Mallat. *A theory for multiresolution signal decomposition: The wavelet representation*. IEEE T. Pattern Anal., 11:674–693, 1989.
- Mar84** V.S. Markin, M.M. Kozlov, and V.L. Borovjagi. *On the theory of membrane fusion: the stalk mechanism*. Gen. Physiol. Biophys., 3:361–377, 1984.
- Mar85** J.L. Martins, J. Buttet, and R. Car. *Electronic and structural properties of sodium clusters*. Phys. Rev. B, 31:1804–1816, 1985.
- Mar01a** A.J. Markvoort, R. Pino, and P.A.J. Hilbers. *Interpolating Wavelets in Kohn-Sham Electronic Structure Calculations*. Lect. Notes Comput. Sc., 2073:541–550, 2001.
- Mar01b** S.J. Marrink, E. Lindahl, O. Edholm, and A.E. Mark. *Simulation of the Spontaneous Aggregation of Phospholipids into Bilayers*. J. Am. Chem. Soc., 123:8638–8639, 2001.
- Mar02** V.S. Markin and J.P. Albanesi. *Membrane Fusion: Stalk Model Revisited*. Biophys. J., 82:693–712, 2002.
- Mar03a** A.J. Markvoort, R. Pino, and P.A.J. Hilbers. *Laterally coupled jellium-like 2D quantum dots*. J. Phys.-Condens Mat., 15:6977–6984, 2003.
- Mar03b** S.J. Marrink and A.E. Mark. *The Mechanism of Vesicle Fusion as Revealed by Molecular Dynamics Simulations*. J. Am. Chem. Soc., 125:11144–11145, 2003.
- Mar03c** S.J. Marrink and A.E. Mark. *Molecular Dynamics Simulation of the Formation, Structure, and Dynamics of Small Phospholipid Vesicles*. J. Am. Chem. Soc., 125:15233–15242, 2003.
- Mar04** S.J. Marrink, A.H. de Vries, and A.E. Mark. *Coarse Grained Model for Semiquantitative Lipid Simulations*. J. Phys. Chem. B, 108:750–760, 2004.

## Bibliography

---

- Mar05a** A.J. Markvoort, P.A.J. Hilbers, and S.V. Nedeia. *Molecular dynamics study of the influence of wall-gas interactions on heat flow in nanochannels*. Phys. Rev. E, 71:066702, 2005.
- Mar05b** A.J. Markvoort, K. Pieterse, M.N. Steijaert, P. Spijker, and P.A.J. Hilbers. *The bilayer-vesicle transition is entropy driven*. J. Phys. Chem. B, 109:22649–22654, 2005.
- McC77** J.A. McCammon, B.R. Gelin, and M. Karplus. *Dynamics of folded proteins*. Nature, 267:585–590, 1977.
- McG04** A.J.H. McGaughey and M. Kaviani. *Thermal conductivity decomposition and analysis using molecular dynamics simulations. Part I. Lennard-Jones argon*. Int. J. Heat Mass Transfer, 47:1783–1798, 2004.
- Met53** N. Metropolis, A.W. Rosenbluth, M.N. Metropolis, A.H. Teller, and E. Teller. *Equation of state calculations by fast computing machines*. J. Chem. Phys, 21:1087–1092, 1953.
- Mey92** Y. Meyer. *Ondelettes due l'intervalle*. Rev. Mat. Iberoam., 7:115–133, 1992.
- Mon96** J.M. Montanero and A. Santos. *Monte Carlo simulation method for the Enskog equation*. Phys. Rev. E, 54:438–444, 1996.
- Mon97** J.M. Montanero and A. Santos. *Simulation of the Enskog equation a la Bird*. Phys. Fluids, 9:2057–2060, 1997.
- Mon02** P.A. Monnard and D.W. Deamer. *Membrane self-assembly processes: Steps toward the first cellular life*. Anat. Rec., 268:196–207, 2002.
- Mül03** M. Müller, K. Katsov, and M. Schick. *A New Mechanism of Model Membrane Fusion Determined from Monte Carlo Simulation*. Biophys. J., 85:1611–1623, 2003.
- Nag00** J.F. Nagle and S. Tristram-Nagle. *Structure of lipid bilayers*. Biochim. Biophys. Acta, 1469:159–195, 2000.
- Nag04** G. Nagayama and P. Cheng. *Effects of interface wettability on microscale flow by molecular dynamics simulation*. Int. J. Heat Mass Transfer, 47:501–513, 2004.
- Nan86** K. Nanbu. *Theoretical basis of the direct simulation Monte Carlo method*. Proceedings of the 15th International Symposium on Rarefied Gas Dynamics, edited by V. Boffi and C. Cercignani:Teubner, Stuttgart, 1986.
- Ned05a** S.V. Nedeia, A.J.H. Frijns, A.A. van Steenhoven, A.J. Markvoort, and P.A.J. Hilbers. *Hybrid method coupling Molecular Dynamics and Monte Carlo simulations to study the properties of gases in microchannels and nanochannels*. Phys. Rev. E, 72:016705, 2005.
- Ned05b** S.V. Nedeia, A.J.H. Frijns, A.A. van Steenhoven, A.J. Markvoort, and P.A.J. Hilbers. *Hybrid Molecular Dynamics-Monte Carlo Simulations for the properties of a dense and dilute hard-sphere gas in a microchannel*. AIP Conf. Proc., 762:767, 2005.
- Neg96** T. Negadi and M. Kibler. *The Periodic Table in Flatland*. Int. J. Quantum Chem., 57:53–61, 1996.
- Nie98a** O.M. Nielsen. *Wavelets in Scientific Computing*. PhD thesis, Technical University of Denmark, 1998.
- Nie98b** Y. Nievergelt. *Wavelets Made Easy*. Birkhäuser, 1998.
- Nil98** L. Nilsson. *Protein Nucleic Acid Interactions*. In P. Schleyer et. al., editor, *Encyclopedia of Computational Chemistry*, volume 3, pages 2220–2229. John Wiley & Sons, 1998.
- Nog01** H. Noguchi and M. Takasu. *Self-assembly of amphiphiles into vesicles: A Brownian dynamics simulation*. Phys. Rev. E, 64:41913, 2001.
- Nog02a** H. Noguchi. *Fusion and toroidal formation of vesicles by mechanical forces: A Brownian dynamics simulation*. J. Chem. Phys, 117:8130–8137, 2002.
- Nog02b** H. Noguchi and M. Takasu. *Structural changes of pulled vesicles: A Brownian dynamics simulation*. Phys. Rev. E, 65:51907, 2002.
- Ono99** N. Ono and T. Ikegami. *Model of Self-Replicating Cell Capable of Self-Maintenance*. In *Proceedings of 5th European Conference on Artificial Life (ECAL '99)*, pages 399–406. Springer, 1999.
- Ono01a** N. Ono. *Artificial Chemistry: Computational Studies on the Emergence of Self-Reproducing Units*. PhD thesis, The University of Tokyo, 2001.
- Ono01b** N. Ono and T. Ikegami. *Artificial Chemistry: Computational Studies on the Emergence of Self-Reproducing Units*. In *Proceedings of 6th European Conference on Artificial Life (ECAL*

- '01), pages 186–195. Springer, 2001.
- Pan99** L.S. Pan, G.R. Liu, and K.Y. Lam. *Determination of slip coefficient for rarefied gas flows using direct simulation Monte Carlo*. J. Micromech. Microeng., 9:89–96, 1999.
- Pao86** G.V. Paolini, G. Ciccotti, and C. Massobrio. *Nonlinear thermal response of a Lennard-Jones fluid near the triple point*. Phys. Rev. A, 34:1355–1362, 1986.
- Par89** R.G. Parr and W. Yang. *Density-functional theory of atoms and molecules*. Oxford University Press, 1989.
- Par00** B. Partoens and F.M. Peeters. *Molecule-Type Phases and Hund's Rule in Vertically Coupled Quantum Dots*. Phys. Rev. Lett., 84:4433–4436, 2000.
- Pay92** M.C. Payne, M.P. Teter, D.C. Allan, T.A. Arias, and J.D. Joannopoulos. *Iterative minimization techniques for ab initio total energy calculations: molecular dynamics and conjugate gradients*. Rev. Mod. Phys., 64:1045–1097, 1992.
- Per92** J.P. Perdew and Y. Wang. *Accurate and simple analytic representation of the electron-gas correlation energy*. Phys. Rev. B, 45:13244–13249, 1992.
- Per96** J.P. Perdew, K. Burke, and M. Ernzerhof. *Generalized gradient approximation made simple*. Phys. Rev. Lett., 77:3865–3868, 1996.
- Pfa93** D. Pfannkuche, V. Gudmundsson, and P.A. Maksym. *Comparison of a Hartree, a Hartree-Fock, and an exact treatment of quantum-dot Helium*. Phys. Rev. B, 47:2244–2250, 1993.
- Pha86** S. Phatisena, R.E. Amritkar, and P.V. Panat. *Exchange and correlation potential for a two-dimensional electron gas at finite temperatures*. Phys. Rev. A, 34:5070–5079, 1986.
- Phi59** J.C. Phillips and L. Kleinman. *New Method for Calculating Wave Functions in Crystals and molecules*. Phys. Rev., 116:287–294, 1959.
- Pic89** W.E. Pickett. *Pseudopotential methods in condensed matter applications*. Comput. Phys. Rep., 9:115–198, 1989.
- Pin03a** R. Pino, A.J. Markvoort, and P.A.J. Hilbers. *Estimating ground-state properties of quantum dot arrays using a local Thomas-Fermi-Dirac approximation*. Eur. Phys. J. B, 23:103–106, 2003.
- Pin03b** R. Pino, A.J. Markvoort, and P.A.J. Hilbers. *Local Thomas-Fermi Approximation for Modeling the Electronic Structure of Planar Devices*. Physica B, 325:149–156, 2003.
- Pin03c** R. Pino, A.J. Markvoort, P.A.J. Hilbers, and R.A. van Santen. *Fully Local Orbital-Free Calculation of Electronic Structure Using Pseudopotentials*. Physica B, 339:119–129, 2003.
- Pin03d** R. Pino, A.J. Markvoort, R.A. van Santen, and P.A.J. Hilbers. *Local Functional Derivative of the Total Energy and the Shell Structure*. J. Mol. Struct., 638:91–98, 2003.
- Pol71** E. Polak. *Computational methods in optimization*, chapter 2.3. Academic press, New York, 1971.
- Pou03** D. Poulidakos, S. Arcidiacono, and S. Maruyama. *Molecular dynamics simulation in nanoscale heat transfer: a review*. Micro. Thermophys. Eng., 7:181–206, 2003.
- Pul69** P. Pulay. *Ab initio calculation of force constants and equilibrium geometries in polyatomic molecules. I. Theory*. Mol. Phys., 17:197, 1969.
- Pyy91** P. Pyykkö and Y. Zhao. *The Elements of Flatland: Hartree Fock Atomic Ground States in Two Dimension for  $Z=1-24$* . Int. J. Quantum Chem., XL:527–544, 1991.
- Rah64** A. Rahman. *Correlations in the motion of atoms in liquid argon*. Phys. Rev., 136:A405–A411, 1964.
- Rah71** A. Rahman and F.H. Stillinger. *Molecular Dynamics Study of Liquid Water*. J. Chem. Phys., 55:3336–3359, 1971.
- Rou04** B. Roux and K. Schulten. *Computational Studies of Membrane Channels*. Structure, 12:1343–1351, 2004.
- Ryc75** J.P. Ryckaert and A. Bellemans. *Molecular dynamics of liquid n-butane near its boiling point*. Chem. Phys. Lett., 30:123–125, 1975.
- San96** M.F. Sanner, J.-C. Spohner, and A.J. Olson. *Reduced surface: an efficient way to compute molecular surfaces*. Biopolymers, 38:305–320, 1996.
- Sch99** T. Schlick, R.D. Skeel, A.T. Brunger, L.V. Kale, J.A. Board, J. Hermans, and K. Schulten. *Algorithmic challenges in computational molecular biophysics*. J. Comput. Phys., 151:9–48,

## Bibliography

---

- 1999.
- Sch02** R.R. Schmidt and B.D. Notohardjono. *High end server low temperature cooling*. IBM J. Res. Dev., 46:739–751, 2002.
- Sco02** H.L. Scott. *Modeling the lipid component of membranes*. Curr. Opin. Struc. Biol., 12:495–502, 2002.
- Sha68** R.W. Shaw. *Optimum form of a modified Heine-Abarenkov model potential for the theory of simple metals*. Phys. Rev., 174:769–781, 1968.
- She01** J.C. Shelley, M.Y. Shelley, R.C. Reeder, S. Bandyopadhyay, and M.L. Klein. *A Coarse Grain Model for Phospholipid Simulations*. J. Phys. Chem. B, 105:4464–4470, 2001.
- Shi02a** J.C. Shillcock and R. Lipowsky. *Equilibrium structure and lateral stress distribution of amphiphilic bilayers from dissipative particle dynamics simulations*. J. Chem. Phys., 117:5048–5061, 2002.
- Shi02b** A. Shioi and T.A. Hatton. *Model for Formation and Growth of Vesicles in Mixed Anionic/Cationic (SOB/CTAB) Surfactant Systems*. Langmuir, 18:7341–7348, 2002.
- Sin72** S.J. Singer and G.L. Nicolson. *The fluid mosaic model of the structure of cell membranes*. Science, 175:720–731, 1972.
- Sme03** A.F. Smeijers. *Molecular Dynamics Simulation of the Spontaneous Formation of a Bilayer Containing Two Types of Phospholipids*. Technical report, Technische Universiteit Eindhoven, 2003.
- Sme05** A.F. Smeijers. *Vesicle fusion*. Technical Report BMI 05/09, Technische Universiteit Eindhoven, 2005.
- Smi90** B. Smit, P.A.J. Hilbers, K. Esselink, L.A.M. Rupert, N.M. van Os, and A.G. Schlijper. *Computer simulations of a water/oil interface in the presence of micelles*. Nature, 348:624–625, 1990.
- Som92** S.A. Somers and H.T. Davis. *Microscopic dynamics of fluids confined between smooth and atomically structured solid surfaces*. J. Chem. Phys., 96:5389–5407, 1992.
- Som95** G.A. Somorjai. *Molecular level studies of solid-gas and solid-liquid interfaces*. Surf. Sci., 335:10–22, 1995.
- Spi03** P. Spijker. *Vesicle Formation and Exocytosis*. Technical report, Technische Universiteit Eindhoven, 2003.
- Ste04** M. Steijaert. *Amphiphile aggregation and vesicle formation: a molecular dynamics study*. Technical Report BMI 04/02, Technische Universiteit Eindhoven, 2004.
- Sti89** I. Stich, R. Car, M. Parrinello, and S. Baroni. *Conjugate gradient minimization of the energy functional: A new method for electronic structure calculation*. Phys. Rev. B, 39:4997–5004, 1989.
- Sto96** M. Stopa. *Quantum dot self-consistent electronic structure and the Coulomb blockade*. Phys. Rev. B, 54:13767–13783, 1996.
- Tan89** B. Tanatar and D.M. Ceperley. *Ground state of the two-dimensional electron gas*. Phys. Rev. B, 39:5005–5016, 1989.
- Tho27** L.H. Thomas. *The calculation of atomic fields*. PCPS-P Camb Philol S., 23:542–548, 1927.
- Tie97** D.P. Tieleman, S.J. Marrink, and H.J.C. Berendsen. *A computer perspective of membranes: molecular dynamics studies of lipid bilayer systems*. Biochim. Biophys. Acta, 1331:235–270, 1997.
- Top73** W.C. Topp and J.J. Hopfield. *Chemically motivated pseudopotential for sodium*. Phys. Rev. B, 7:1295–1303, 1973.
- Tro91a** N. Troullier and J.L. Martins. *Efficient pseudopotentials for plane-wave calculations*. Phys. Rev. B, 43:1993–2006, 1991.
- Tro91b** N. Troullier and J.L. Martins. *Efficient pseudopotentials for plane-wave calculations II*. Phys. Rev. B, 43:8861–8869, 1991.
- Tsu95** E. Tsuchida and M. Tsukada. *Electronic-structure calculations based on the finite element method*. Phys. Rev. B, 94:5573–5578, 1995.
- Tym97** C.J. Tymczak and X. Wang. *Orthonormal Wavelet Bases for Quantum Molecular Dynamics*. Phys. Rev. Lett., 78:3654–3657, 1997.

- 
- Var74** F.J. Varela, H.R. Maturana, and R. Uribe. *Autopoiesis: The organization of living systems, its characterization and a model*. Biosystems, 5:187–196, 1974.
- vB73** H. van Beijeren and M.H. Ernst. *The modified Enskog equation*. Physica, 68:437–456, 1973.
- vdH04** F.G.C.M. van den Heuvel. *Scalability & Surface Tension Calculation*. Technical Report BMI 04/11, Technische Universiteit Eindhoven, 2004.
- vdP83** P. van der Ploeg and H.J.C. Berendsen. *Molecular dynamics of a bilayer membrane*. Mol. Phys., 49:233–248, 1983.
- Ven99** M. Venturoli and B. Smit. *Simulating the self-assembly of model membranes*. Phys. Chem. Comm., 10, 1999.
- Ver67** L. Verlet. *Computer experiments on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules*. Phys. Rev., 159:98–103, 1967.
- Ver68** L. Verlet. *Computer experiments on classical fluids. II. Equilibrium correlation functions*. Phys. Rev., 165:201–214, 1968.
- Ver79** A.J. Verkleij, C. Mombers, W.J. Gerritsen, L. Leunissen-Bijvelt, and P.R. Cullis. *Fusion of phospholipid vesicles in association with the appearance of lipidic particles as visualized by freeze fracturing*. BBA-Biomembranes, 555:358–361, 1979.
- vL02** J.J.A. van Leeuwen. *A component based approach to parallel programming and its application to molecular dynamics*. Master’s thesis, Technische Universiteit Eindhoven, 2002.
- Vog87** R. Vogelsang, C. Hoheisel, and G. Ciccotti. *Thermal conductivity of the Lennard-Jones liquid by molecular dynamics calculations*. J. Chem. Phys., 86:6371–6375, 1987.
- Wei84** S.J. Weiner and P.A. Kollman. *A new force field for molecular mechanical simulation of nucleic acids and proteins*. J. Am. Chem. Soc., 106:765–784, 1984;.
- Wei96** S. Wei and M.Y. Chou. *Wavelets in Self-Consistent Electronic Structure Calculations*. Phys. Rev. Lett., 76:2650–2653, 1996.
- Wen00** A. Wensauer, O. Steffens, M. Suhrke, and U. Rössler. *Laterally coupled few-electron quantum dots*. Phys. Rev. B, 62:2605–2613, 2000.
- Wes92** P. Wesseling. *An Introduction to Multigrid Methods*. John Wiley & Sons, 1992.
- Whi89** S.R. White, J.W. Wilkins, and M.P. Teter. *Finite-element method for electronic structure*. Phys. Rev. B, 39:5819–5833, 1989.
- Wu03** H.Y. Wu and P. Cheng. *An experimental study of convective heat transfer in silicon microchannels with different surface conditions*. Int. J. Heat Mass Transfer, 46:2547–2556, 2003.
- Xue03** L. Xue, P. Keblinski, S.R. Phillpot, S.U.S. Choi, and J.A. Eastman. *Two regimes of thermal resistance at a liquid-solid interface*. J. Chem. Phys., 118:337–339, 2003.
- Xue04** L. Xue, P. Keblinski, S.R. Phillpot, S.U.S. Choi, and J.A. Eastman. *Effect of liquid layering at the liquid-solid interface on thermal transport*. Int. J. Heat Mass Transfer, 47:4277–4284, 2004.
- Yam02** S. Yamamoto, Y. Maruyama, and S. Hyodo. *Dissipative particle dynamics study of spontaneous vesicle formation of amphiphilic molecules*. J. Chem. Phys., 116:5842–5849, 2002.
- Yam03** S. Yamamoto and S. Hyodo. *Budding and fission dynamics of two-component vesicles*. J. Chem. Phys., 118:7937–7943, 2003.





## Summary

In many biology, chemistry and physics applications molecular simulations can be used to study material and process properties. The level of detail needed in such simulations depends on the application. In some cases quantum mechanical simulations are indispensable. However, traditional ab-initio methods, usually employing plane waves or a linear combination of atomic orbitals as a basis, are extremely expensive in terms of computational as well as memory requirements. The well-known fact that electronic wave functions vary much more rapidly near the atomic nuclei than in inter-atomic regions calls for a multi-resolution approach, allowing one to use low resolution and to add extra resolution only in those regions where necessary, so limiting the costs. This is provided by an alternative basis formed of wavelets. Using such a wavelet basis, a method has been developed for solving electronic structure problems that has been applied successfully to 2D quantum dots and 3D molecular systems. In other cases, it suffices to use effective potentials to describe the atomic interaction instead of the use of the electronic structure, enabling the simulation of larger systems. Molecular dynamics simulations with such effective potentials have been used for a systematic study of surface wettability influence on particle and heat flow in nanochannels, showing that the effects at the solid-gas interface are crucial for the behavior of the whole nanochannel. Again in other cases even coarse grained models can be used where the average behavior of several atoms is combined into a single particle. Such a model, refraining from as much detail as possible while maintaining realistic behavior, has been developed for lipids and with this model the dynamics of membranes and vesicle formation have been studied in detail. A disadvantage of molecular dynamics simulations with effective potentials is that no reactions are possible. Therefore a new method has been developed, where molecular dynamics is coupled with stochastic reactions. Using this method, both unilamellar and multilamellar vesicle formation, and vesicle growth, bursting, and healing are shown. Still larger systems can be simulated using other methods, like the direct simulation Monte Carlo method. However, as shown for nanochannels, these methods are not always accurate enough. But, exploiting again that the finest level of detail is often only needed in part of the domain, a hybrid method has been developed coupling molecular dynamics, where needed for accuracy, and direct simulation Monte Carlo, where possible in order to speed up the calculation. Further development of such hybrid simulations will further increase molecular simulation's scientific role.



## Samenvatting

Dit proefschrift gaat over de ontwikkeling en de toepassing van een aantal moleculaire simulatie technieken. Zulke moleculaire simulaties kunnen binnen de natuurkunde, chemie en biologie gebruikt worden om materiaal en proces eigenschappen te bestuderen. De benodigde mate van detail wordt bepaald door de toepassing. Voor sommige toepassingen zijn kwantummechanische berekeningen noodzakelijk. Traditionele ab-initio methoden maken vaak gebruik van basissets bestaande uit sinussen of atomaire orbitalen en zijn zowel qua geheugengebruik als qua rekenkracht duur. Het feit dat elektrongolffuncties veel meer variëren rond de atoomkernen dan in gebieden tussen de kernen in, vraagt om een methode waarbij op de ene plaats een veel hogere resolutie gebruikt kan worden dan op de andere. Hierin wordt precies voorzien met de wavelet basis. Een methode gebaseerd op zulke wavelets is ontwikkeld en wordt toegepast op moleculen en ‘quantum dots’. Voor andere toepassingen hoeven de interacties tussen de atomen niet kwantummechanisch uitgerekend te worden, maar volstaat het gebruik van effectieve potentialen, zodat grotere systemen berekend kunnen worden. Met behulp van Moleculaire Dynamica simulaties met zulke effectieve potentialen is de invloed van de gas-wand interactie op warmte transport en stroming in nanokanalen bestudeerd. In weer andere gevallen kan het gemiddelde gedrag van enkele atomen beschreven worden door een enkel deeltje. Hiervan gebruik makend is een model gemaakt voor lipiden, waarbij van zoveel mogelijk detail geabstraheerd is, maar dat toch realistisch gedrag vertoont. Met behulp van dit model is de dynamica van membranen en de formatie van vesicels bestudeerd. Een nadeel van simulaties met effectieve potentialen is dat er geen chemische reacties mogelijk zijn. Om toch reacties mogelijk te maken hebben we Moleculaire Dynamica gekoppeld met stochastische reacties. Met deze techniek laten we de formatie van enkel- en dubbelwandige vesicels zien evenals groei, barsten en genezen van zulke vesicels. Met behulp van andere minder nauwkeurige maar snellere methoden, zoals de Direct Simulation Monte Carlo methode, kunnen weer grotere systemen bestudeerd worden. Maar zoals we laten zien voor nanokanalen, zijn deze methoden niet altijd nauwkeurig genoeg. Echter, wederom gebruik makend van het feit dat vaak niet overal in de ruimte dezelfde mate van detail vereist is, hebben we een hybride methode ontwikkeld bestaande uit een koppeling van Moleculaire Dynamica en Direct Simulation Monte Carlo. Door alleen daar waar voor de nauwkeurigheid vereist Moleculaire Dynamica te gebruiken en daar waar mogelijk de efficiëntere methode kan de simulatie versneld worden. Verdere ontwikkeling van zulk soort hybride methoden zal de rol van moleculaire simulaties verder versterken.



# Dankwoord

Graag wil ik hier aan het eind van mijn proefschrift iedereen bedanken die direct dan wel indirect een bijdrage heeft geleverd aan de totstandkoming van dit proefschrift. Daarbij wil ik een aantal personen in het bijzonder noemen.

In de eerste plaats wil ik mijn promotor Peter Hilbers bedanken die mij alle vrijheid heeft gegeven het onderzoek van de grond af op te bouwen en in te richten. Hij heeft mij hierbij met behulp van opbouwende kritische vragen en suggesties telkens gestimuleerd en heeft mij op velerlei wijzen steeds gesteund.

Ook wil ik de overige leden van mijn promotie commissie bedanken voor hun belangstelling voor het onderzoek, opbouwende opmerkingen en discussies. Van hen wil ik mijn tweede promotor Thijs Michels nog speciaal bedanken voor zijn begeleiding bij mijn eerste wetenschappelijke stappen en Rutger van Santen voor de huidige voortgezette samenwerking.

Een aantal mensen waarmee ik intensief heb samengewerkt aan onderwerpen uit dit proefschrift verdienen ook een bijzondere vernoeming. Hierbij wil ik allereerst Ramiro Pino bedanken voor zijn steun tijdens het begin van mijn promotieonderzoek. Zowel inhoudelijk op het gebied van kwantum mechanische berekeningen alsook vriendschappelijk heeft hij mij enorm gesteund. Ook ben ik zeer blij met zijn komst uit Amerika om tijdens mijn promotieplechtigheid de rol als copromotor te vervullen. De volgende in deze rij die ik wil bedanken is Koen Pieterse voor de prettige samenwerking bij het opzetten van onze moleculaire simulatie groep. Ook zijn kunst van het visualiseren van simulaties zal ik niet snel vergeten aangezien een voorbeeld hiervan de omslag van dit proefschrift siert. Als derde in deze rij wil ik Silvia Nedeia bedanken voor de prettige samenwerking op het gebied van de hybride simulaties. Tot slot in deze rij gaat mijn dank uit naar de overige medeauteurs van een aantal van mijn artikelen, Dragan Bosnacki, Arjan Frijns en Anton van Steenhoven, voor hun ideeën, opmerkingen en steun bij het tot stand komen van deze artikelen.

Een deel van het werk is ook verricht in samenwerking met stagiair(e)s en afstudeerders van zowel Informatica als Biomedische Technologie die ik in de loop van de tijd heb mogen begeleiden: Jasper van Leeuwen, Frank van den Heuvel, Sander Smeijers, Marvin Steijaert, Pieter van Grootel, Peter Spijker, Inge van den Berg en Freek van Dooren. Behalve directe bijdragen aan het proefschrift zijn er natuurlijk ook velen indirect bij betrokken geweest. Hierbij wil ik allereerst al mijn overige collega's en oud collega's bedanken. Vele leerzame bijeenkomsten van de VLSI-club en de Bio-club, evenals de discussies op de gang, maar ook de gezelligheid tijdens menig koffiepauze en borrel hebben zowel mijn verblijf bij Informatica als bij Biomedische Technologie tot een

## Dankwoord

---

zowel leerzame als plezierige tijd gemaakt. Hierbij wil ik in het bijzonder mijn oud kamergenoten Roy Willemsen en John Segers noemen, als ook de twee andere AIO's die indertijd de stap van Informatica naar Biomedische Technologie maakten, namelijk Ronald Ruimerman en Anthony Liekens, en tot slot de vaste staf leden bij Informatica en Biomedische Technologie: Johan Lukkien, Rudolf Mak, Tom Verhoeff, Huub ten Eikelder en Nico Kuijpers. En natuurlijk niet te vergeten Margret Philips voor de secretariële ondersteuning.

Tot slot wil ik graag nog mijn familie bedanken, die ondanks dat ze veelal geen flauw idee hadden waar ik mee bezig was of waar ik regelmatig met mijn gedachten zat, mij altijd onvoorwaardelijk hebben gesteund.

Bart Markvoort

## Curriculum Vitae

Bart Markvoort was born on July 29, 1973, in Utrecht, The Netherlands. He attended the Scholengemeenschap Augustinianum, in Eindhoven, where he obtained his Gymnasium diploma in 1991. From 1991 he studied applied physics at the Technische Universiteit Eindhoven. In April 1997 he graduated on the subject of small angle x-ray scattering on asphaltenes in the Theoretical Physics group.

After his graduation, he started his Ph.D. work in the Parallel Systems group of the Computer Science department at the Technische Universiteit Eindhoven. Since September 1st, 2001, Bart continued his research as a research fellow and teacher within the BioModeling and bioInformatics group of the Biomedical Engineering department at the same university.

This research resulted in this thesis, on which he expects to receive his Ph.D. at March 14, 2006.





## Publications

A.J. Markvoort, D. Bosnacki and P.A.J. Hilbers, **Artificial chemistry in coarse grained molecular dynamics and its application to vesicle dynamics**, to be published.

A.J. Markvoort, K. Pieterse, M.N. Steijaert, P. Spijker and P.A.J. Hilbers, **The bilayer-vesicle transition is entropy driven**, J. Phys. Chem. B, **109**, 22649-22654 (2005)

S.V. Nedeia, A.J.H. Frijns, A.A. van Steenhoven, A.J. Markvoort and P.A.J. Hilbers, **Hybrid method coupling molecular dynamics and Monte Carlo simulations to study the properties of gases in microchannels and nanochannels**, Physical Review E, **72**, 016705 (2005)

A.J. Markvoort, P.A.J. Hilbers and S.V. Nedeia, **Molecular dynamics study of the influence of wall-gas interactions on heat flow in nanochannels**, Physical Review E, **71**, 066702 (2005)

A.J.H. Frijns, S.V. Nedeia, A.J. Markvoort, A.A. van Steenhoven and P.A.J. Hilbers, **Molecular Dynamics and Monte Carlo Simulations for Heat Transfer in Micro and Nano-channels**, Lecture Notes in Computer Science, **3039**, 666-671 (2004)

A.J. Markvoort, Ramiro Pino and P.A.J. Hilbers, **Laterally coupled jellium-like 2D quantum dots**, Journal of Physics: Condensed Matter, **15**, 6977-6984 (2003)

R. Pino, A.J. Markvoort, P.A.J. Hilbers and R.A. van Santen, **Fully Local Orbital-Free Calculation of Electronic Structure Using Pseudopotentials**, Physica B, **339**, 119-129 (2003)

R. Pino, A.J. Markvoort, R.A. van Santen and P.A.J. Hilbers, **Local Functional Derivative of the Total Energy and the Shell Structure in Atoms and Molecules**, Journal of Molecular Structure, **638**, 91-98 (2003)

R. Pino, A.J. Markvoort and P.A.J. Hilbers, **Local Thomas-Fermi Approximation for Modeling the Electronic Structure of Planar Devices**, Physica B, **325**, 149-156 (2003)

R. Pino, A.J. Markvoort and P.A.J. Hilbers, **Estimating ground-state properties of quantum dot arrays using a local Thomas-Fermi-Dirac approximation**, European Physical Journal B, **23**, 103-106 (2003)

## **Publications**

---

A.J. Markvoort, P.A.J. Hilbers and Ramiro Pino, **Interpolating Wavelets in Kohn-Sham Electronic Structure Calculations**, Lecture Notes in Computer Science, **2073**, 541-550 (2001)

R.J.F. van Haren, A.J. Markvoort, A.J.C. van Rijswijk, F.A.P. Blom and J.H. Wolter, **Reflection of finite-width edge channels**, Physical Review B, **54**, 2806-2812 (1996)