ADVANCED METHODS IN AB-INITIO MOLECULAR DYNAMICS

A dissertation submitted to

ETH ZURICH

for the degree of

Doctor of Natural Sciences

presented by

FAWZI ROBERTO MOHAMED

Diplomierter in Rechnergestützten Wissenschaften, ETH Zürich

born: 09.09.1976

citizen of Sorengo, Ticino, Switzerland

accepted on the recommendation of

Prof. Michele Parrinello, examiner
Prof. J. Hütter, co-examiner
Prof. Ulrich W. Suter, co-examiner

2006
Abstract

Ab-initio Molecular dynamics is a powerful method to study the properties of molecular systems. It can be used to explore states where not much is known, check assumptions and as starting point for more empirical theories. Unfortunately ab-initio molecular dynamics is computationally costly, and so not applicable to all the systems one would like. In this thesis we present some solutions that are practically realized in the openly available, and actively developed cp2k program (CP204) to apply this method to systems that are of common interest.

Density Functional Theory (DFT), and the method we use to solve its equations efficiently (GPW) are presented in detail, along with the reasons their efficiency. The $E^{\text{XC}}$ functionals, are analyzed and some partial solutions to reduce the ripples of the energy functional, inspired by the new multi grid spline methods of section 2.7, are proposed. The parallelization strategy, and the algorithms developed for it, like the new sparse-matrix full-matrix multiplication are described, as well as more technical aspects like the input, error handling, and algorithm environments. Then the method is validated and its performance checked.

In the third chapter, after a review of a comparison between various sampling methods on ab-initio liquid water, we perform extensive tests on temperature and functional dependence of liquid water. The uncertainty on temperature along with the sensitivity of water to temperature changes can explain the spread of values found in the literature. Results seem to confirm that water described by the BLYP functional is over-structured, and that almost all functionals are sluggish diffusive. A quantum treatment of the ion motion (that is known to be important in water) could explain the difference of diffusibility. Unfortunately functional dependence turns out to be rather large which shows that DFT is not yet a black-box method.
To simulate bigger systems, to take into account the screening or periodicity effects due to the environment the hybrid QM/MM method is a valid solution for many situations. A novel method to calculate the Coulomb interaction between the MM and QM part that fits the Coulomb potential with Gaussians is explained and validated. A fully consistent periodic extension to this method is discussed and a new decoupling scheme to calculate the electrostatic of the QM system is proposed.

Finally a perturbation approach that uses the wave functions (not the density) as basic variable is studied. This approach is a very attractive and natural way to treat systems that decompose in fragments (molecules) and can be useful also for analysis purposes. The theory is developed in a fully non orthogonal setting, and some possible applications are highlighted, in particular the possibility of calculating non-converged forces. For this purpose a method to calculate the derivatives in a conjugated gradient procedure, even if some of these derivatives, for efficiency reasons need to be calculated as little as possible, is developed.
Riassunto

La simulazione molecolare ab-initio è un metodo efficace per studiare le proprietà dei sistemi molecolari. Può essere usato sia per esplorare materiali e stati termodinamici su cui ci sono poche conoscenze, sia per controllare assunzioni che come punto di partenza per teorie più empiriche. Purtroppo questo genere di simulazioni sono costose e quindi non possono essere applicate a tutti i sistemi. In questa tesi vengono presentate alcune soluzioni per applicare questi metodi ai sistemi reali. Tali soluzioni sono realizzate in pratica nel programma open source cp2k: un programma che è attivamente sviluppato e disponibile a tutti (CP204).

In questa tesi la teoria funzionale della densità (DFT) ed il metodo, che usiamo per risolverne in maniera efficiente le equazioni (GPW), vengono spiegati sottolineando le ragioni della loro efficienza. Viene analizzata la valutazione dei funzionali di scambio e correlazione e vengono proposte alcune soluzioni per ridurne le oscillazioni, ispirate ai metodi multi griglia basati su splines della sezione 2.7. La strategia di parallelizzazione e gli algoritmi sviluppati per l’esecuzione parallela, come il nuovo metodo per moltiplicare matrici sparse con matrici dense in parallelo, vengono discussi così come aspetti più tecnici come l’input, la gestione dell’errore e gli ambienti algoritmici. Infine il metodo è validato e la sua efficienza è testata.

Nel terzo capitolo, dopo aver brevemente presentato un confronto fra vari metodi per misurare ab-initio le proprietà termodinamiche dell’acqua liquida, vengono eseguiti una serie di test estensivi. I test affrontano l’effetto della temperatura e del funzionale di scambio e correlazione sulle proprietà dell’acqua simulata. L’incertezza sulla temperatura e la sua grande influenza sulle proprietà dell’acqua possono spiegare l’ampio spettro di valori presenti nella letteratura. I risultati sembrano confermare che l’acqua descritta con il funzionale BLYP è sovrastrutturata e quasi tutti i funzionali sono troppo poco diffusivi. Gli effetti quantistici sul movimento degli ioni, di cui è risaputa l’importanza, potrebbero
spiegare la differenza di diffusibilità. Purtroppo la dipendenza dal funzionale risulta essere abbastanza elevata e questo dimostra che per questa applicazione DFT non è un metodo che può essere usato a scatola chiusa.

Il metodo ibrido QM/MM è spesso una valida soluzione per simulare sistemi più grandi e tener conto degli effetti di schermatura o di periodicità dovuti all’ambiente. Nel calcolo dell’interazione tra la parte QM e MM viene applicato un metodo innovativo che espande il potenziale culombiano usando delle funzioni gaussiane, e tale metodo viene spiegato e testato. Inoltre viene discussa una sua estensione che tratta le condizioni periodiche di contorno in un modo completamente consistente, ed un nuovo metodo che calcola l’interazione elettrostatica della parte QM.

Infine viene studiato un’approccio perturbativo che usa le funzioni d’onda come base e non la densità. L’approccio adottato è un modo naturale per trattare sistemi che sono composti da frammenti (molecole) e può inoltre essere molto attrattivo per l’analisi. La teoria viene sviluppata per orbitali e basi non ortogonali ed alcune sue possibili applicazioni vengono discusse. In particolare viene sviluppata la possibilità di calcolare le forze in modo consistente senza arrivare a convergenza. A tale scopo viene sviluppata la modalità per calcolare le derivate all’interno del metodo del gradiente coniugato ed il termine $\partial C/\partial R$ con un costo comparabile a quello di una valutazione delle forze.
## Contents

1 Introduction .............................................. 1

2 Ab-initio MD .............................................. 5

   2.1 Born-Oppenheimer ...................................... 6

   2.1.1 Density Functional Theory ............................. 8

   2.2 The CP2K program ....................................... 10

   2.3 Introduction ............................................. 11

   2.4 Gaussian and plane waves method .................. 12

      2.4.1 Energy functional .................................. 12

      2.4.2 Pseudo potentials .................................. 14

      2.4.3 Electrostatic energy ................................ 16

      2.4.4 Exchange-Correlation Potential ..................... 18

      2.4.5 Implementation .................................... 23

      2.4.6 Symbolic Calculations .............................. 25

      2.4.7 Linear scaling Kohn-Sham matrix construction .... 27

   2.5 GPW forces .............................................. 29

      2.5.1 Deriving the Kohn-Sham matrix from the GPW energy 29
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5.2</td>
<td>Forces on the ions</td>
<td>30</td>
</tr>
<tr>
<td>2.6</td>
<td>Program structure and implementation</td>
<td>33</td>
</tr>
<tr>
<td>2.6.1</td>
<td>Algorithm environments</td>
<td>33</td>
</tr>
<tr>
<td>2.6.2</td>
<td>F77 Interface</td>
<td>34</td>
</tr>
<tr>
<td>2.6.3</td>
<td>Input</td>
<td>35</td>
</tr>
<tr>
<td>2.6.4</td>
<td>Memory management</td>
<td>36</td>
</tr>
<tr>
<td>2.6.5</td>
<td>Parallelization strategy</td>
<td>36</td>
</tr>
<tr>
<td>2.7</td>
<td>Splines</td>
<td>38</td>
</tr>
<tr>
<td>2.7.1</td>
<td>Multi grid</td>
<td>38</td>
</tr>
<tr>
<td>2.7.2</td>
<td>Periodic uniform splines</td>
<td>40</td>
</tr>
<tr>
<td>2.7.3</td>
<td>Periodic prolongation/restriction</td>
<td>40</td>
</tr>
<tr>
<td>2.7.4</td>
<td>Non-periodic uniform splines</td>
<td>43</td>
</tr>
<tr>
<td>2.7.5</td>
<td>Non-periodic prolongation/restriction</td>
<td>47</td>
</tr>
<tr>
<td>2.8</td>
<td>Basis sets</td>
<td>49</td>
</tr>
<tr>
<td>2.9</td>
<td>Wave function optimization</td>
<td>51</td>
</tr>
<tr>
<td>2.9.1</td>
<td>Traditional diagonalization</td>
<td>52</td>
</tr>
<tr>
<td>2.9.2</td>
<td>Orbital transformation method</td>
<td>54</td>
</tr>
<tr>
<td>2.10</td>
<td>Molecular dynamics</td>
<td>55</td>
</tr>
<tr>
<td>2.10.1</td>
<td>Algorithms</td>
<td>56</td>
</tr>
<tr>
<td>2.10.2</td>
<td>Wave function extrapolation</td>
<td>57</td>
</tr>
<tr>
<td>2.10.3</td>
<td>BO-MD</td>
<td>58</td>
</tr>
<tr>
<td>2.11</td>
<td>Accuracy</td>
<td>62</td>
</tr>
<tr>
<td>2.11.1</td>
<td>BSSE in liquid water</td>
<td>64</td>
</tr>
</tbody>
</table>
2.12 Benchmarks ....................................................... 66
2.12.1 Liquid water ............................................. 66

3 Ab-initio Liquid Water ........................................... 71
3.1 Investigation of Different Sampling Approaches .............. 72
3.1.1 Car-Parrinello Method ..................................... 72
3.1.2 Monte Carlo ............................................. 74
3.1.3 Article .................................................. 74
3.2 Temperature and functional dependence in water ............ 75
3.3 Introduction ................................................ 76
3.4 Validation of methods and Gaussian basis sets in the condensed phase ... 77
3.4.1 Electronic structure computation and molecular dynamics ... 77
3.4.2 Basis set convergence ................................... 79
3.5 Temperature dependence of properties of BLYP water .......... 80
3.6 Performance of six different density functionals .............. 85
3.7 Conclusion ................................................ 89

4 The Quantum Mechanics / Molecular Mechanics (QM/MM) Method 93
4.1 Classical potentials ......................................... 94
4.1.1 QM/MM interaction ..................................... 96
4.1.2 Efficiency ........................................... 98
4.2 GEEP: Gaussian Expansion of the Electrostatic Potential .... 99
4.2.1 QM/MM Forces ....................................... 103
4.3 Algorithm ................................................. 103
5 **EP: Energy through DFT-Perturbation Theory**

5.1 Derivative Notation ............................................. 116

5.2 Energy expression .............................................. 118

5.3 Second order expansion ....................................... 119
  5.3.1 Derivatives .................................................. 119
  5.3.2 Equation to solve .......................................... 122

5.4 Partial convergence and MD ................................. 122
  5.4.1 Non-convergence forces ................................... 124
  5.4.2 Perturbation starting point: fragment calculations .... 125

5.5 Minimizer Derivatives ........................................... 127
  5.5.1 Conjugated gradient ....................................... 127
  5.5.2 CG step function .......................................... 129
  5.5.3 Direct derivative accumulation .......................... 130
  5.5.4 Inverse CG step ........................................... 131
  5.5.5 Inverse derivative accumulation ......................... 132
  5.5.6 $F$ derivatives ........................................... 133
  5.5.7 Numerical stability ....................................... 134
CONTENTS

5.6 The big picture ............................................. 134

5.7 Conclusion and Outlook ............................... 135

6 Conclusion .................................................. 137

7 Acknowledgments ....................................... 141

List of figures ............................................... 143

List of tables ................................................ 145

Bibliography .................................................. 147

A Appendix .................................................... 163

A.1 Differential Geometry .................................. 163

A.2 bra/ket notation ........................................ 165

A.3 bra/ket derivatives ...................................... 166

A.3.1 Energy Derivatives .................................... 167

A.4 XC derivatives ........................................... 170

A.5 Derivatives of a CG step .............................. 173

A.5.1 db derivative .......................................... 174

A.5.2 dA derivative .......................................... 174

A.5.3 F derivative ............................................ 176

Curriculum Vitae ............................................. 179

Publications ................................................ 181
Chapter 1

Introduction

Computer simulation is increasingly used to study chemical and physical reactions as complement to the experiment. It can simulate states that are very difficult or impossible to probe and measure properties that are difficult to obtain experimentally. It can give insight in the mechanisms of chemical reactions, and about the structure of materials. The drawbacks of computer simulations can be summarized as follows:

1. they can be only as good as the underlying theory
2. they must be validated to be sure that no programming error affects the results
3. even with the newest computers the systems that one can simulate are smaller, and can be simulated for much less time that one would like, thus the approximations due to finite system size or time of simulation should be checked

A fruitful way to look at simulations is to see them as an experiment of a theory: they look at some consequences of a theory by performing a virtual experiment. Like any experiment the result of a simulation can be affected by experimental error both systematic and random.

In the present work we focus on atomistic simulations, i.e. simulations which atoms are explicitly present, and furthermore where the nuclei are treated classically, which is an approximation that is valid in the vast majority of cases. Then to specify the state of the system one has to give the positions of all the nuclei, volume, temperature, or the other thermodynamic quantities of interest. Each set of positions is a configuration $\mathbf{R}_f$. Given
the positions, a definition of the system (atom types,...) plus the parameters connected to
the thermodynamic variables (volume/cell size,...), that for the moment we will assume
as fixed, each (microscopic) configuration has a potential energy $V(R)$.

A very efficient way to define the potential energy is to use classical empirical force
fields. These potentials do not describe the electrons explicitly, they just use an analytic
form for $V(R_i)$. Normally they have a fixed topology, they cannot describe a chemical
reaction, and they describe a system only in a often not fully known range of situations.

To describe chemical reactions, or for systems that are difficult to parameterize one
can try to describe the electrons quantum mechanically. This solves all the problems,
bonds are automatically taken into account, and there are no empiric parameters. For
this reasons these calculations are also called ab-initio, and these are the kind of sim¬
ulations this work focuses on. It should be pointed out that also the transferability of
ab-initio simulations is not perfect, because to be practical, they have to include some
approximations, but normally it is very good, and the approximation that one performs
are known.

To define potentials for the ions we use the Born-Oppenheimer approximation which
assumes that the electrons, being much lighter than the nuclei, are always at equilibrium
with respect to the ion positions. This reduces the problem to finding the ground state
of the electrons for each configuration,

$$V(R_I) = \min_{\psi_i} E_{QM}(R_I, \psi_i), \quad (1.1)$$

where $E_{QM}$ is the potential energy of the system of quantum electrons with the classical
ions.

To find the ground state of this system, instead of solving the time independent
Schrödinger equation it is possible to use Density Functional Theory (DFT). DFT is
particularly attractive because it uses the electronic density and is computationally much
less demanding than finding the complete many particle wave function. Nonetheless
theoretically it can give the exact ground-state energy.

The evaluation of the DFT energy is still expensive, and different schemas have
been devised to perform it. Here we will present the GPW schema (LHP97) which tries
to combine the advantages of a localized basis-set (Gaussian) and an auxiliary basis set
(Plane waves) in the new and highly efficient implementation CP2K/QUICKSTEP to which
I significantly contributed.

The optimization and parallelization of the computation of the kernel methods is a crucial aspect for practical applications, thus it is described in chapter 2, along with some results confirming the performance and validating the method.

Once one has the potential energy it is possible, for example, to calculate thermodynamic properties by sampling over microscopic configurations, it is also possible to study microscopic evolution of the systems to better understand mechanisms: how a reaction, a conformational change or other chemical processes take place. To sample the microscopic properties there are mainly two common approaches, the molecular dynamics (MD) approach and the Monte Carlo/Metropolis approach. In chapter 3 we look at ab-initio MD simulations of water, and at the temperature and functional dependence of the simulation results.

One would always like to simulate bigger systems for longer times. All the optimizations described in chapter 2 help, but one would always like to have more. If one expects only part of the system to be difficult to describe with classical potentials, a good solution is to use a hybrid method where part of the system is treated classically and part ab-initio. This class of methods is called QM/MM and a novel implementation of this method that uses an innovative approach to calculate the electrostatic interaction between QM and MM subsystems is presented in chapter 4, where some possible developments like a better periodic solver, and embedding are also discussed.

Finally an approach based on DFT perturbation theory, which gives an intuitive approach very useful for analysis, and some theoretical developments connected to it are presented in chapter 5. In particular a way to keep into account weak interactions is explored.
Chapter 2

Ab-initio MD

For an ab-initio potential there are just atoms, the molecules just arise implicitly as groups of atoms that are strongly bound together, but they are not explicitly programmed in the potential. This is the reason ab-initio potentials can describe chemical reactions, and as the approximations of ab-initio theories are not dependent on the system or its state. Thus one can quite confidently explore states where not much is known, and still expect reasonable results. This is the reason ab-initio simulations are interesting.

As drawback they are costly to simulate (further restricting the size of the system and the time that can be simulated). Their big advantage, having basically no free parameters that have to be empirically adjusted, also means that it is difficult to take advantage from the knowledge that some molecules are stable in a given situation. Furthermore if in a simulation you do encounter an effect due to the approximation of the theory, normally it is quite difficult to remove it without going to higher level methods that have a much higher cost: you cannot easily re-parameterize your model for your specific system.

In this chapter after shortly reviewing the Born-Oppenheimer approximation in section 2.1 and DFT (section 2.1.1) we will describe the theory of the GPW method (LHP97), and its implementation in QUICKSTEP, which is part of the freely available program package CP2K(CP204). We review the GPW energy functional and illustrate its linear scaling nature in section 2.4, whereas the derivatives of the functional are described in section 2.5. In section 2.6 details on the program structure and implementation are provided. Basis sets are shortly discussed in section 2.8 and in section 2.9 two methods to perform wave function optimization are presented. Molecular dynamics simulations and improved wave
function extrapolation method are the subject of section 2.10. The accuracy of the method is illustrated for gas phase and condensed phase systems in section 2.11, and the efficiency of the code for serial and parallel calculations is shown in section 2.12.

### 2.1 Born-Oppenheimer

To be able to associate to each configuration a unique potential energy an approximation that works well in most cases is the non-relativistic Born-Oppenheimer approximation: i.e to assume that the electrons (being much lighter than the nucleus) are always equilibrated, and that the core, being slow, can be treated classically. This means that the forces on the nuclei will depend only on the atom positions, as the electrons will be constrained to be in equilibrium with respect to the atom positions. Unless one wants to simulate really extreme temperatures, processes where level crossing plays an important role, photo chemistry,... the electrons can be assumed to be in the ground-state.

Thus the equations of motion are

\[
M_I \ddot{R}_I = -\nabla_I \min_{\Psi_0} \langle \Psi_0 | \mathcal{H} | \Psi_0 \rangle - \nabla_I E^{II} \\
E_0 \Psi_0 = \mathcal{H} \Psi_0, \tag{2.1}
\]

where \( M_I \) are the masses of the ions, \( R_I \) are the space positions of the nuclei (ions), \( \nabla_I \) represent the derivative with respect to the \( R_I \), \( \Psi_0 = \Psi_0(x_i) \) is the wave function of the electrons, \( \mathcal{H} \) the electronic Hamiltonian, \( E_0 \) the ground-state energy and

\[
E^{II} = \sum_{i<j} \frac{Z_i Z_j}{|R_i - R_j|} \tag{2.2}
\]

with \( Z_I \) the charge of nuclei and \( E^{II} \) the ion-ion interaction energy. \( E^{II} \) could be added to the definition of \( \mathcal{H} \), but for later convenience we don’t do it.

The electronic wave function \( \Psi_0 \) has to be normalized (\( \langle \Psi_0 | \Psi_0 \rangle = 1 \)) and furthermore, as electrons are fermions, it should be antisymmetric with respect to the exchange of the variables \( x_i = (r_i, s_i) \), where \( r_i \) are space coordinate of the electron \( i \) and \( s_i \) its spin coordinate.

For an isolated N-electron atomic or molecular system the non-relativistic electronic
Hamiltonian can be written (in atomic units) as

$$\mathcal{H} = T + V_{ne} + V_{ee} = \sum_{i=1}^{N} -\frac{1}{2} \nabla^2_i + \sum_{i} v(r_i) + \sum_{i<j}^{N} \frac{1}{|r_i - r_j|},$$

(2.3)

where $T$ is the kinetic operator, $V_{ee}$ the electron electron interaction, and $V_{ne}$ the nuclei-electron interaction in which

$$v(r_i) = \sum_{I,i} \frac{Z_I}{|R_I - r_i|}$$

(2.4)

It can be observed that in (2.1) the nuclei are treated classically, and they move in a potential

$$V(R_I) = \min_{\Psi_0} \langle \Psi_0 | \mathcal{H} | \Psi_0 \rangle + E^{II}$$

(2.5)

which depend only implicitly on the electrons. Indeed the BO (Born Oppenheimer) approximation by assuming that the electrons are so much faster than the nuclei that the nuclei feel only the average force due to the electrons excludes any direct coupling between the movement of the atoms and the movement of the electrons. Actually the electrons don’t have to be necessarily in the ground state, one could also give a finite temperature to the electrons, or do a simulation in an excited state, but normally with BO MD (Born Oppenheimer Molecular Dynamics) one assumes that the electrons are in the ground state, and this is the case that we want to explore.

It turns out that these approximations work well for a very large and interesting class of problems.

As we just said in the BO framework the atoms can be seen as if they where moving in a classical potential $V(R_I)$, and one can perform the simulation just as in the classical MD (Molecular Dynamics), the only difficulty being to calculate the potential $V(R_I)$ and its force $(-\nabla V(R_I))$. Unfortunately calculating these is much more expensive than with a classical potential and the scaling with the number of atoms is also much worse ($\leq O(N^4)$ against $O(N)$). So to be able to apply this method to systems of hundreds of atoms for tens of picoseconds one has to simplify the calculation of the ground state energy.
2.1.1 Density Functional Theory

To describe the ground-state of a system of electrons Density Functional Theory (HK64; KS65) (DFT) is very attractive because the use of the electronic density $n(r)$ (i.e. the probability of finding an electron at the position $r$) instead of the wave-functions $\psi(r)$ is computationally advantageous.

Hohenberg (HK64) showed how it was in principle possible to use just the electronic density

$$n(r) = \langle r | \Psi_0 \rangle \langle \Psi_0 | r \rangle$$

(2.6)

to calculate the ground-state energy $E_0$ (or actually also a finite temperature state). The wave function $\Psi_0$ and the energy $E_0$ are clearly determined by the number of electron $N_e$ and the potential $v$ (which in turn is determined by the position of the ions). Hohenberg and Kohn showed that the density $n(r)$ defines $N_e$ and $v$ in an unique way. This means that

$$E_v[n(r)] = \int n(r)v(r)dr + F_{HK}[n(r)]$$

(2.7)

with an universal functional $F_{HK}$. Following Levy (Lev89; Lev82; LP85) one can formally define this functional as

$$F_{HK}[n(r)] = \min_{\Psi \rightarrow n(r)} \langle \Psi | (T + V_{ee}) | \Psi \rangle$$

(2.8)

where with $\Psi \rightarrow n(r)$ we mean that the minimization is to be performed on all the $\Psi$ such that $n(r) = \langle r | \Psi \rangle \langle \Psi | r \rangle$. Thus one should be able to find the ground state energy as

$$E_0 = \min_{n(r)} \left( \int n(r)v(r)dr + F_{HK}[n(r)] \right)$$

(2.9)

where the minimization is to be performed on the $n(r)$ that are N-representable, i.e. that satisfy

$$n(r) \geq 0$$

(2.10)

$$\int |\nabla \sqrt{n(r)}|^2 dr < \infty$$

(2.11)

$$\int n(r)dr = N_e,$$

(2.12)

something that all the reasonable ground-state densities do.

Unfortunately no efficient procedure to calculate $F_{HK}$ is known, and naive approach along the lines of (2.8) is more costly than just finding the ground state. Anyway the
knowledge that such a functional exists spurred much research to find an approximate expression for it that would be efficient to calculate. It seems that it is extremely difficult to find a direct expression for $F_{HK}$ that has the accuracy in the energy needed for chemical applications ($\sim 1 \text{mH}$).

To achieve it Kohn and Sham (KS65) reintroduced the orbitals to evaluate the kinetic part of $F_{HK}$. The Kohn Sham kinetic energy is

$$E^T[n(r)] = \min_{\psi_i \rightarrow n(r)} \sum_i -\langle \psi_i | \frac{1}{2} \nabla | \psi_i \rangle.$$  \hfill (2.13)

In this $\psi_i$ represent an electronic orbital and they are orthonormal, i.e.

$$\langle \psi_i | \psi_j \rangle = \delta_{ij}.$$ \hfill (2.14)

It is to be noted not all the densities that satisfy (2.10-2.12) can be represented with just $N_e$ orthogonal orbitals (and a single determinant), densities that satisfy this stronger request are called noninteracting v-representable. This is a requirement that some densities don’t fulfill, but it seems to work well in practice. $E^T \leq T$, but the difference between the two seems to be quite small (of the order of the correlation energy).

The Kohn Sham energy then is

$$T_{KS}[n] + \int n(r)v(r)dr + \frac{1}{2} \int \frac{n(r)n(r')}{|r-r'|}drdr' + E^{XC}$$ \hfill (2.15)

where $E^{XC}$ (the exchange and correlation energy) is defined as the part of $F_{HK}$ that is still missing after removing $E^T$ and

$$E^H = \frac{1}{2} \int \frac{n(r)n(r')}{|r-r'|}drdr'.$$ \hfill (2.16)

from it. This formulation is much more accurate than (2.7) because $E^{XC}$ is much smaller than $F_{HK}$, and better approximations for it exist, some of which will be presented in section 2.4.4.

To minimize (2.15) normally one doesn’t use the density $n$ as basis variable, but the orbitals $\psi_i$, because from them it is easy to find the corresponding $n$:

$$n(r) = \sum_i \langle r | \psi_i \rangle \langle \psi_i | r \rangle$$ \hfill (2.17)

which means that the minimization is performed only on the noninteracting v-representable $n$. 
2.2 The CP2K program

The Theory can be fascinating and very useful, but we are interested in practical realization of it in the form of computer programs that can be run. Computer simulation can advantageously take a place between theory and experiment. With it the consequences of theory can be explored in a novel way and one can look at details that can be extremely difficult or impossible to look at in experiment. This can help at getting a better insight at the mechanisms that govern a property or a chemical reaction.

Simulation results (like experimental ones) have to be carefully checked to exclude stupid technical problems (programming errors, measurement problems). Only once one is convinced that the technical apparatus works, more interesting things can be looked at. It should also be clear that a Computer Simulation cannot describe phenomena that are not accounted for in its underlying theory.

Here all the implementation has been done building or extending the simulation package CP2K (CP204; VKM+05). CP2K is an open-source project hosted at http://cp2k.berlios.de/ and is written in Fortran 90. I firmly believe that the development of open-source software is very useful for scientific progress: this way others can easily reproduce your data, and control how the simulation was performed. Also new development can take advantage of what has been done before without having to reinvent the wheel. This is very compatible with the scientific method, and I think that it should be favored, especially when the research is founded with public money.

The goal of CP2K is to create an infrastructure to perform various kinds of atomic simulations. Various methods are available, classical, ab-Initio, polarizable,... and having them in the same framework enables one to combine them in novel ways. Quickstep, the DFT part of the CP2K package, and the part on which I worked most, aims to be a fast DFT program able to tackle big systems. First I will describe in detail the methods used in Quickstep, and its abilities.

Part of this chapter is reproduced with permission from the review paper about CP2K/QUICKSTEP from Joost VandeVondele, Matthias Krack, Fawzi Mohamed, Michele Parrinello, Thomas Chassaing, and Jürg Hutter, QUICKSTEP: fast and accurate density functional calculations using a mixed Gaussian and plane waves approach, Computer Physics Communications 167 (2005) 103128. Copyright 2005 Elsevier B.V.
2.3 Introduction

I was the main contributor of the sections on exchange and correlation, basis set superposition error, program structure, parallel matrix multiplication (with Joost VandeVondele), and I actively participated in writing down the total energy expression and in the general review of the whole article.

2.3 Quickstep

Density functional theory (HK64; KS65) (DFT) is a well-established method to perform electronic structure calculations. The accuracy of the method is such that many properties of systems of interest to chemistry, physics, material science, and biology can be predicted in a parameter-free way. The standard computational approach to DFT is already efficient and thus appropriate for fairly large systems, currently about 100 atoms. Nevertheless, the computation of the Hartree (Coulomb) energy and the orthogonalization of the wave functions are not scaling linearly with system size, and these terms therefore dominate the computational cost for larger systems (Goe99). The hybrid Gaussian and plane waves (GPW) method (LHP97) provides an efficient way to treat these terms accurately at a significantly reduced cost. We present here the implementation of this method in QUICKSTEP, which is part of the freely available program package CP2K (CP204).

The method uses an atom-centered Gaussian-type basis to describe the wave functions, but uses an auxiliary plane wave basis to describe the density. With a density represented as plane waves or on a regular grid, the efficiency of Fast Fourier Transforms (FFT) can be exploited to solve the Poisson equation and to obtain the Hartree energy in a time that scales linearly with the system size. Fast Fourier Transforms and regular grids are well established in plane wave codes (MH00) and their efficiency has recently been exploited in a similar method (FMP02a; FMP02b; FMP03; FM03; BFMP04).

The use of an auxiliary basis set to represent the density goes back to the seventies (Whi73; DCS79) and has become increasingly popular as resolution of the identity (RI) method or density fitting method. Contrary to the GPW method, most RI methods expand the density in an auxiliary basis of the same nature as the primary basis, but optimized specifically for this purpose (VAF93; ETÖ+95; EWTA97). Since a density expanded in plane waves can be represented on a real-space grid, there is a direct connection to methods that use numerical calculation of matrix elements (Del89; TH94) or grid discretization and finite element methods (for a recent review see Ref. (Bec00)).
GPW method is most similar to methods that employ auxiliary real-space grids but differ by the choice of localized primary basis functions used to represent the wave functions (CLI95; OAS96; GHGB97; KHF00; MSHP02; LYT03).

Periodic boundary conditions follow naturally from the FFT based treatment of the Poisson equation, and the GPW method scales linearly for three dimensional systems with a small prefactor and an early onset. The GPW method seems therefore best suited for the simulation of large and dense systems, such as liquids and solids, and all recent applications of the method fall in this category (KP00; HRG+03; KM04; KMM+04; VMK+04). For these systems, it is important to be able to efficiently perform stable molecular dynamics simulations, in order to address finite temperature effects. Plane wave codes and the basic GPW implementation presented here require that the nuclei are described using pseudo potentials. This approximation is highly accurate if e.g. Goedecker-Teter-Hutter (GTH) pseudo potentials are employed (GTH96; HGH98). An extension of the GPW method, the Gaussian and augmented-plane-wave (GAPW) method (LHP99) allows for all electron calculations.

The extensive experience with Gaussian-type basis sets shows that basis set sequences that increase rapidly in accuracy can be constructed in a systematic way (Dun89). At the same time, a compact description of the wave functions is maintained, and this opens the way for efficient methods to solve for the self consistent field (SCF) equations. Furthermore, as Gaussian functions are localized, the representations of the Kohn-Sham, overlap and density matrix in this basis become sparse with increasing system size (Goe99). This eventually allows for solving the Kohn-Sham (KS) equations using computational resources that scale linearly with system size. We have currently only implemented methods that are scaling cubically with system size, but these have been designed to reach high efficiency for Gaussian basis sets (VH03).

2.4 Gaussian and plane waves method

2.4.1 Energy functional

Central in the Gaussian and plane wave (GPW) method (LHP97) is the use of two representations of the electron density. Such a dual representation allows for an efficient
2.4 Gaussian and plane waves method

treatment of the electrostatic interactions, and leads to a scheme that has a linear scaling

cost for the computation of the total energy and Kohn-Sham matrix with respect to the

system size. The first representation of the electron density \( n(\mathbf{r}) \) is based on an expansion

in atom centered, contracted Gaussian functions

\[
n(\mathbf{r}) = \sum_{\mu \nu} P^{\mu \nu} \varphi_{\mu}(\mathbf{r}) \varphi_{\nu}(\mathbf{r})
\]  

(2.18)

where \( P^{\mu \nu} \) is a density matrix element, and \( \varphi_{\mu}(\mathbf{r}) = \sum_{\alpha} d_{\mu \alpha} g_{\alpha}(\mathbf{r}) \) with primitive Gaussian

functions \( g_{\alpha}(\mathbf{r}) \) and corresponding contraction coefficients \( d_{\mu \alpha} \). The second representation

employs an auxiliary basis of plane waves, and is given by

\[
\tilde{n}(\mathbf{r}) = \frac{1}{\Omega} \sum_{\mathbf{G}} \tilde{n}(\mathbf{G}) \exp(i \mathbf{G} \cdot \mathbf{r})
\]  

(2.19)

where \( \Omega \) is the volume of the unit cell, and \( \mathbf{G} \) are the reciprocal lattice vectors. The expansion coefficients \( \tilde{n}(\mathbf{G}) \) are such that \( \tilde{n}(\mathbf{r}) \) is equal to \( n(\mathbf{r}) \) on a regular grid in the unit cell. The difference \(|n(\mathbf{r}) - \tilde{n}(\mathbf{r})|\) goes to zero as the cutoff energy \( E_{\text{cut}} \) goes to infinity.

This choice allows for a rapid conversion between \( n(\mathbf{r}) \), \( \tilde{n}(\mathbf{r}) \) and \( \tilde{n}(\mathbf{G}) \) using an efficient

mapping procedure (section 2.7.5.1) and fast Fourier transforms (FFT).

Using this dual representation, the Kohn-Sham DFT energy expression (HK64; KS65)
as employed within the GPW framework is defined as

\[
\]  

\[
= \sum_{\mu \nu} P^{\mu \nu} \langle \varphi_{\mu}(\mathbf{r}) | -\frac{1}{2} \nabla^2 | \varphi_{\nu}(\mathbf{r}) \rangle
+ \sum_{\mu \nu} P^{\mu \nu} \langle \varphi_{\mu}(\mathbf{r}) | V_{\text{PP}}^{\lambda}(\mathbf{r}) | \varphi_{\nu}(\mathbf{r}) \rangle
+ \sum_{\mu \nu} P^{\mu \nu} \langle \varphi_{\mu}(\mathbf{r}) | V_{\text{nl}}^{PP}(\mathbf{r}, \mathbf{r'}) | \varphi_{\nu}(\mathbf{r'}) \rangle
+ 2\pi \Omega \sum_{\mathbf{G}} \frac{\tilde{n}^*(\mathbf{G}) \tilde{n}(\mathbf{G})}{\mathbf{G}^2}
+ \int e^{XC}(\mathbf{r}) \, d\mathbf{r}
+ \frac{1}{2} \sum_{i \neq j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|}
\]  

(2.20)

where \( E^T[n] \) is the electronic kinetic energy, \( E^V[n] \) is the electronic interaction with the

ionic cores, \( E^H[n] \) is the electronic Hartree energy and \( E^{XC}[n] \) is the exchange–correlation
energy. The interaction energies of the ionic cores with charges \( Z_A \) and positions \( \mathbf{R}_A \) is denoted by \( E^{\text{II}} \). \( E^{\text{V}[n]} \) is described by norm-conserving pseudo potentials with a potential split in a local part \( V_{\text{loc}}^{\text{pp}}(r) \) and a fully non-local part \( V_{\text{nl}}^{\text{pp}}(r, r') \).

The pseudo potential terms are described in more detail in section 2.4.2, the electrostatic contributions to the total energy in section 2.4.3, and the exchange and correlation term in section 2.4.4. In section 2.4.7 we illustrate that also for three dimensional systems of moderate size linear scaling computational cost is observed with the GPW method.

### 2.4.2 Pseudo potentials

Normally the core electrons don’t have a big influence on the electronic properties, and most physical and chemical properties are mainly given by the valence electrons. This means that an accurate description of the core electrons is not necessary for many applications (including bond breaking and formation). One can take advantage of this fact to simplify the description of the atom. In fact this is needed when using an uniform grid to represent the density. The core electron are packed tightly and an expansion of an atomic all-electron density or wave function in plane waves is computationally inefficient, as the grid-spacing needed to describe the core electrons is very small.

An accurate description of the valence electrons can be obtained using a pseudo potential description of the nuclei. A technique that is well established in the plane wave community.

The idea of the pseudo potential is to describe explicitly only the valence electrons, and replace the effect of the core electrons with a suitably modified potential. The valence electrons should have the same behavior (eigenvalues) as in the all-electron case, pseudo potential should be as smooth as possible, and it should be transferable (i.e. applicable to many different situations).

To achieve this normally one optimizes the pseudo wave function in the atomic case, where the all-electron wave functions can be calculated, even with relativistic corrections. In particular the pseudo potential should \((\text{HSC}79; \text{BHS}82)\) reproduce the atomic eigenvalues for the valence electrons, and the normalized pseudo and all-electron functions should agree beyond a given radius \( r_{\text{core}} \). Furthermore the potential should be norm conserving.
2.4 Gaussian and plane waves method

\[ \int_{|r|<R} |\psi_{\text{pseudo}}(r)|^2 dr = \int_{|r|<R} |\psi_{\text{all-electron}}(r)|^2 dr \]  (2.21)

for all the \( R \) outside the core region (\( R > r_{\text{core}} \)), which guarantees that the electrostatic potential is the correct one and that the phase shift of a scattered particle is the same.

Likewise the logarithmic derivative of the wave functions, that as can be seen in models based on the muffin tin potential, controls the band structure, and the first energy derivative should agree outside the core region. To actually guarantee the correct scattering properties a non local radial part, one for each angular momentum, should be used. The separable form introduced by Allan and Teter (AT87a) was a big step forward as it particularly efficient and attractive, reducing the number of integrals to calculate.

For the GPW method we take advantage of the experience with the pseudo potential scheme, and implement it using the pseudo potential of Goedecker, Teter, and Hutter (GTH) (GTH96; HGH98).

This is an accurate and transferable pseudo potential (see also section 2.11) that has an analytic form that allows for an efficient treatment of all terms within the GPW method.

The norm-conserving, separable, dual-space GTH pseudo potentials consist of a local part including a long-ranged (LR) and a short-ranged (SR) term

\[ V_{\text{loc}}^{\text{PP}}(r) = V_{\text{loc}}^{\text{LR}}(r) + V_{\text{loc}}^{\text{SR}}(r) \]  (2.22)

\[ = - \frac{Z_{\text{ion}}}{r} \text{erf}(\alpha_{\text{PP}}^1) + \sum_{i=1}^{4} C_i^{\text{PP}} \left( \sqrt{2} \alpha_{\text{PP}}^i r \right)^{2i-2} \exp \left[ - \left( \alpha_{\text{PP}}^i r \right)^2 \right] \]  (2.23)

with

\[ \alpha_{\text{PP}}^i = \frac{1}{\sqrt{2} r_{\text{loc}}^{PP}} \]

and a non-local part

\[ V_{\text{nl}}^{\text{PP}}(r, r') = \sum_{lm} \sum_{ij} \langle r | p_l^{jm} \rangle h_i^j \langle p_j^{jm} | r' \rangle \]  (2.24)

with the Gaussian-type projectors

\[ \langle r | p_l^{jm} \rangle = N_l^i Y^{lm}(\hat{r}) r^{l+2i-2} \exp \left[ - \frac{1}{2} \left( \frac{r}{r_t} \right)^2 \right] \]

where \( N_l^i \) are normalization constants and \( Y^{lm}(\hat{r}) \) spherical harmonics. The small set of GTH pseudo potential parameters (\( r_{\text{loc}}^{PP}, C_i^{PP}, r_t, \) and \( h_i^j \)) have been optimized with respect
to atomic all-electron wave functions as obtained from fully relativistic density functional
calculations using a numerical atomic program. The optimized pseudo potentials include
all scalar relativistic corrections via an averaged potential (HGH98), and improve therefore
the accuracy for applications involving heavier elements. The emphasis in the construction
of these pseudo potentials has been on accuracy, and hence these pseudo potentials are
computationally more demanding for plane wave methods, as a large plane wave basis
typically is required.

The GPW method is less sensitive to the hardness of the pseudo potential since the
kinetic energy (see Eq. 2.20) and the short range pseudo potential terms are computed
analytically in the Gaussian basis. The long range term can be efficiently treated as part
of the electrostatic energy (see section 2.4.3), whereas the short range terms can be easily
computed as two and three center overlap integrals.

An extended database (H–Rn) with GTH pseudo potential parameters based on the
local density approximation is available (CP204) for use with QUICKSTEP. In addition,
parameters for the common elements have been optimized for the gradient-corrected
exchange-correlation potentials of Becke, Lee, Yang, and Parr (BLYP) (Bec88a; LYP88;
MSSP89), Becke and Perdew (BP) (Bec88a; Per86), Hamprecht, Cohen, Tozer and Handy
(HCTH/120, HCTH/407) (HCTH98) and Perdew, Burke and Ernzerhof (PBE) (PBE96).

2.4.3 Electrostatic energy

The electrostatic energy in a periodic system is defined by a conditionally converging sum
in which the separate contributions of ions and electrons are infinite. All terms of the
electrostatic energy are therefore treated simultaneously

\[ E^{\text{ES}} = \int V_{\text{loc}}^{\text{PP}}(r)n(r)dr + 2\pi \Omega \sum_G \frac{\tilde{n}(G)\tilde{n}(G)}{G^2} + \frac{1}{2} \sum_{i\neq j} \frac{Z_iZ_j}{|R_i - R_j|} \]  

(2.25)

using the Ewald sum method (AT87b) as it is commonly implemented in plane wave
electronic structure codes (MH00). The long range part of all electrostatic interactions
is treated in Fourier space, whereas the short range part is treated in real space. This
separation is conveniently achieved for the ionic cores if a Gaussian charge distribution
\(n^I_{c}(r)\) for each nucleus is introduced and defined as

\[ n^I_{c}(r) = -\frac{Z_I}{(R^I_J)^3} \pi^{-3/2} \exp \left[ -\left( \frac{r - R^I_I}{R^I_J} \right)^2 \right] \]  

(2.26)
Figure 2.1: Shown is the rapid convergence of the absolute error in the electrostatic energy Eq. 2.29 with respect to plane wave cutoff at fixed density matrix. The system is a single water molecule described with fairly hard GTH pseudo potentials and a TZV2P basis in a 10Å cubic cell. The relation $E_{\text{cutoff}} = \frac{\pi^2}{2h^2}$ is used throughout this work to convert the grid spacing $h$ to the corresponding plane wave cutoff.
in which the parameter $R_j^c$ can be chosen for optimal performance. In QUICKSTEP, this parameter is set to

$$R_j^c = \sqrt{2} r_{pp}^{loc}$$

so that the corresponding potential of the Gaussian charge distribution

$$V_{core}^{\prime}(r) = \int dr' \frac{n_i^i(r')}{|r-r'|} = -\frac{Z_i}{|r-R_j|} \text{erf} \left[ \frac{|r-R_j|}{R_j^c} \right],$$

cancels exactly the long-ranged term $V_{loc}^{LR}(r)$ of the local pseudo potential.

We rewrite the expression for the energy Eq. 2.25 using Eq. 2.26 as

$$E_{ES} = \int V_{loc}^{SR}(r)n(r)dr$$

$$+ \frac{\Omega}{2} \sum_G \tilde{n}_{tot}(G) v^H(G)$$

$$+ \frac{1}{2} \sum_{i \neq j} \frac{Z_iZ_j}{|R_i-R_j|} \text{erfc} \left[ \frac{|R_i-R_j|}{\sqrt{R_i^c + R_j^c}} \right] - \sum_i \frac{1}{2\pi R_j^c},$$

where erfc is the complementary error function, the Hartree potential $v^H(G) = 4\pi \tilde{n}_{tot}(G)/G^2$, and a total charge distribution $\tilde{n}_{tot}(G) = \tilde{n}(G) + \tilde{n}_c(G)$ has been introduced. The last three terms of Eq. 2.29 define the total Hartree energy ($E_{H[n_{tot}]}$), the overlap energy ($E_{ovl}$) and self energy ($E_{self}$) respectively.

The two representations of the electrostatic energy Eq. 2.25 and Eq. 2.29 are strictly equivalent if an infinite sum over $G$ vectors is employed. In practice, a difference between the two energy expressions, due to the use of finite density grids, can be observed, but this difference is rapidly convergent with respect to $G$ i.e. the grid spacing used. The rapid convergence of the electrostatic energy Eq. 2.29 with respect to the plane wave cutoff, and thus the size of the auxiliary basis is shown in Fig. 2.1.

### 2.4.4 Exchange–Correlation Potential

A necessary ingredient in practical application of DFT is the introduction of an approximate exchange and correlation functional $E_{XC}$. In the present implementation of QUICKSTEP typical generalized gradient approximations (GGA) and meta-GGAs based on the kinetic energy density $\tau$ such as BLYP (Bec88a; LYP88; MSSP89), PBE (PBE96),
2.4 Gaussian and plane waves method

HCTH (HCTH98; BDHS00; BH01), OLYP (HC02), TPSS (TPSS03) can be computed efficiently. These functionals have the common general form

\[ E_{\text{XC}}[n] = \int e^{\text{xc}}(n_\uparrow(r), n_\downarrow(r), \nabla n_\uparrow(r), \nabla n_\downarrow(r), \tau_1, \tau_1) \, dr, \] (2.30)

where

\[ \tau_1(r) = \frac{1}{2} \sum_{i} \nabla \Psi_i^\uparrow(r) \cdot \nabla \Psi_i^\downarrow(r) = \frac{1}{2} \sum_{\mu\nu} P_{\downarrow}^{\mu\nu} \nabla \phi_\mu(r) \cdot \nabla \phi_\nu(r) \] (2.31)

is the local kinetic energy density. This form does not cover functionals where the Hartree-Fock exchange term is explicitly introduced (Bec93b; Bec93a) such as the popular B3LYP functional(SDCF94). Currently, no implementations of Hartree-Fock exchange can approach the efficiency with which Eq. 2.30 can be evaluated.

2.4.4.1 Numerical Issues

To compute Eq. 2.30 and its derivatives with respect to the density matrix we use a discrete representation of \( E_{\text{XC}} \) on the same uniform density grid that has been used for the Hartree energy as is common in plane wave based calculations (WB94) and other grid based methods (BMS01). This avoids the use of the more accurate techniques employed within the quantum chemistry community (Bec88b; MHL93; TA95) as these methods would, within the GPW scheme, dominate the total cost of the calculation by a relatively large factor.

Actually to compute KS matrix one needs

\[ v_{\text{xc}} = \frac{\delta E}{\delta n} \] (2.32)

The details of its derivation are discussed in the appendix A.4, here we will present how to calculate it in practice.

For the evaluation of the exchange and correlation contribution, in the spin unpolarized case, the following operations are performed for each grid point:

1. collocation of \( n(r) = \sum_{\mu\nu} P_{\mu\nu} \phi_\mu(r) \phi_\nu(r) \)
2. collocation of \( \tau(r) = \sum_{\mu\nu} (1/2) P_{\mu\nu} \nabla \phi_\mu(r) \cdot \nabla \phi_\nu(r) \)
3. numerical approximation of \( \nabla n \) based on the values of \( n(r) \) on the grid
4. evaluation of $e^{xc}$ and its derivatives $\partial e^{xc}/\partial n$, $\partial e^{xc}/\partial \nabla n$, $\partial e^{xc}/\partial \tau$ on each point of the grid

5. computation of $v_n^{xc}$ and $v_{\tau}^{xc}$ on the grid

\[
v_n^{xc} = \frac{\partial e^{xc}}{\partial n} - \nabla \cdot \left( \frac{\partial e^{xc}}{\partial |\nabla n|} \frac{\nabla n}{|\nabla n|} \right)
\]

\[
v_{\tau}^{xc} = \frac{\partial e^{xc}}{\partial \tau}
\]  

6. calculation of the matrix element of the sum of $v_n^{xc}$ and the Hartree potential $v^H(r)$ (see section 2.4.3) between the Gaussian

\[
\int (v_n^{xc}(r) + v^H(r)) \phi_\mu(r) \phi_\nu(r) \, dr
\]  

7. calculation of the matrix element of $v_{\tau}^{xc}$ between the Gaussian

\[
\frac{1}{2} \int v_{\tau}^{xc}(r) \nabla \phi_\mu(r) \cdot \nabla \phi_\nu(r) \, dr
\]

where the grid based collocation, integration and consistent differentiation are discussed in more detail in sections 2.5.1 and 2.7.5.1.

For the numerical evaluation of the exchange-correlation functional care should be taken to fulfill as much as possible the exact relationships of the density. A very important one for functionals that depend on the kinetic energy density is $|\nabla n| < 8n\tau$ that can be derived from the definition of $\tau$ and $n$ using Schwartz’s inequality $(a|b|^2 \leq (a|a)(b|b)$ with $a = \nabla \psi_i$ and $b = \psi_i$. As in our method the derivatives to calculate $\tau$ are calculated directly from the gaussian representation whereas the derivatives of $\nabla n$ are calculated from the grid representation, this relationship can be violated and has has to be enforced.

The presence of terms such as

\[
t = -\frac{|\nabla n|^2}{n^\alpha}
\]

\[
\frac{\partial t}{\partial |\nabla n|} = -2 \frac{|\nabla n|}{n^\alpha}.
\]  

in GGAs and meta-GGAs leads to very sensitive behavior in regions of vanishing density such as the tails of the atomic densities. The near singularities encountered in Eq. 2.37 are in that case customarily resolved by removing the contributions to $e^{xc}$ and $v^{xc}$ of the regions where the density $n$ is lower than a given cutoff $\epsilon$. However, using pseudo
Figure 2.2: Behavior of $n$ and $v^{xc}$ with the BLYP functional close to the core of an O atom in a water molecule along the bisector of the HOH angle with an unusually large cutoff of 5000 Ry. The left panel shows the electron density, whereas the three right panels show $v^{xc}$ as calculated using a derivative in $G$ space, using a quadratic spline $(D_6(S_6)^{-1})$ and using the operators $S_{10} - D_6$ as defined in the text. It can be observed that the latter methods lead to a more physical exchange and correlation potential surface.
potentials, the density can also be small in the core region, where gradients are typically larger.

This is especially true for the GTH pseudo potentials that by construction have a zero pseudo charge density at the core for all elements apart from H. We illustrate in Fig. 2.2 that for these pseudo potentials the core region is by far the most problematic part of the exchange and correlation potential. The pronounced spike of $v^{xc}$ in the core gives rise to small variations in the total energy as atoms move relative to the grid. These oscillations are mostly periodic with respect to the grid spacing, and thus removing all those high frequencies, solves the problem. This can be done by going to a grid with effectively half as much grid points in each direction, which is equivalent to using four times the cutoff for the XC calculation, a very expensive solution.

The $G$ space differentiation

$$\nabla_x \tilde{n}(G) = iG \tilde{n}(G)$$

is commonly used in plane waves codes but is not the best choice with the GPW method. Whereas $G$ space differentiation of the density on the grid yields the exact derivative $\nabla n(r)$ in the former case, the approximate $\nabla \tilde{n}(r)$ is obtained in the later case. When used, the differentiation of a small spike of $\partial e/\partial |\nabla n|$ in (2.33) gives rise to the strong 'ringing' effects illustrated in Fig. 2.2. Even though integration effectively filters out the highest frequencies, the energy oscillates significantly when the system is translated (see panel (b) of Fig. 2.3).

We have explored different schemes to compute the exchange and correlation energy more accurately, and to describe them we introduce a nearest neighbor smoothing operator $S_q$ defined as

$$(S_q f)_{i,j,k} = \frac{q^3}{q^3 + 6q^2 + 12q + 8} \sum_{l=-1}^{1} \sum_{m=-1}^{1} \sum_{n=-1}^{1} q^{-|l|-|m|-|n|} f_{i+l,j+m,k+n},$$

and a smoothed finite differences operator $D_q$ that for the $x$ derivative is

$$(D_q f)_{i,j,k} = \frac{q^2}{2(q^2 + 4q + 4)} \sum_{l=-1}^{1} \sum_{m=-1}^{1} q^{-|l|-|m|} (f_{i-l,j+l,k+m} - f_{i+l,j+l,k+m})$$

and likewise for the other directions.

To avoid the 'ringing' a numerical derivative that assumes less continuity can be used. $D_6(S_6)^{-1}$ calculates the derivative of the quadratic spline interpolating $n$ on the
grid. It behaves better than the \( G \) derivative, but the energy oscillations are not sufficiently reduced. \( D_6 \) alone, i.e. without sharpening step \((S_6)^{-1}\), gives information on the neighborhood rather than on the grid point itself, and damps the oscillations more, at a cost in the accuracy of the energies (see panels (a) and (b) of Fig. 2.3).

For a translation invariant evaluation of the integral of a function \( f \) over the grid points \((i,j,k)\) it is appropriate to associate to each mesh point not the value of the function itself, but rather an estimate of its average value in a neighborhood of \((i,j,k)\). For a highly non-linear term such as the exchange correlation energy this average can not easily be estimated. We therefore evaluate the XC functionals using a locally averaged density \( \hat{n}(r) \) employing the smoothing operator \( S_q \). Typical values for \( q \) are 10 or 50 depending on the required amount of smoothing. Such a smoothing is equivalent with a redefinition of \( E^{XC} \) that reduces to the identity as the cutoff is increased. \( \hat{v}^{xc}_n \) can be calculated as function of \( \hat{n} = S_q n \) as

\[
\hat{v}^{xc}_n = \hat{v}^{xc}_n \frac{\delta \hat{n}}{\delta n} = S_q \hat{v}^{xc}_n. \tag{2.41}
\]

Combining the quadratic spline and \( D_6 \) derivatives with the smoothing on \( n \) brings the oscillations of the energy to an acceptable level. \( S_{50} - D_6 (S_6)^{-1} \) has good convergence characteristics, and implies only a small grid spacing dependent re-normalization of \( E^{XC} \). The operator \( S_{10} - D_6 \) implies a significant amount of smoothing, resulting in even less grid dependence in the forces, and is fast to calculate since an inversion step is not necessary, but might be less appropriate to study systems where significant charge reorganization takes place. The exchange and correlation potentials obtained with these methods are well behaved, which also helps the convergence of the SCF procedure.

Nevertheless, none of the methods presented here is fully satisfactory, as a balance between the different accuracy goals is difficult to achieve. Non linear core corrected pseudo potentials (LFC82) could provide a more elegant solution as the problematic region of small density would be removed. It is likely that these pseudo potentials can be treated efficiently, and they would bring additional benefits for strongly spin polarized systems. The Gaussian and augmented-plane-wave (GAPW) method (LHP99; KP00) also resolves the issues described here in a more fundamental way.

### 2.4.5 Implementation
Figure 2.3: Panel (a) shows the average systematic error in the interaction energy with the BLYP functional for water dimer configurations (~ 5 kcal from the minimum), panel (b) the oscillations of the interaction energy due to imperfect translation invariance, and panel (c) the forces on the center of mass. The $G$ space derivatives, the $(D_6(S_6)^{-1})$ and $D_6$ derivatives, and the combined $n$ smoothing-derivative operators $S_{50} - D_6(S_6)^{-1}$ and $S_{10} - D_6$ are compared with respect to the cutoff (triangles, +, ×, squares and circles). The reference interaction energies were calculated with a cutoff of 2000 Ry using the usual $G$ space derivatives. The oscillations of the energy and the magnitude of the forces on the center of mass of the last two methods are at an acceptable level for cutoffs of about 300 Ry.
To perform perturbation theory and other operations that need higher derivatives of the XC functional a good way to handle all the possible combinations of partial derivatives of $e^{\text{xc}}$ that can come up becomes mandatory. One needs to be able to know which derivatives are present, and respect symmetries (commutativity of derivatives). This means that one has to be able to uniformly label the derivatives of the XC functional in such a way that the other routines have access to the meta information about which derivatives are present and where they are stored.

The derivatives of $e^{\text{xc}}$ are needed only for a very short time, to calculate the functional derivative with respect to $n$ and $\tau$, and then from it the one with respect to the Gaussian basis set.

The ideal method would be to calculate the functional on the fly when needed and then discard the result as soon as it is not needed anymore (a streaming calculation of the functional). This would reduce the memory footprint of the application, but unfortunately Fortran is not well suited for this, as it is not possible to inline these function calls in a portable way. Thus the routine work on arrays of points, and returns a list of arrays, each with a string describing the derivative. There are functions to handle the extraction of meta information, search for a given derivative, and the handling of the symmetries. A batched approach, not calculating the functional on the whole grid in one shot would be doable, and useful especially if you would like to use a finer grid for the XC, but is not implemented.

The presence of this meta information lets the perturbation kernel (for example) work with any functional.

### 2.4.6 Symbolic Calculations

Modern XC functionals have a complicated algebraic form, whose complete definition is often spread on several papers and calculating the higher derivatives rapidly leads to very long and tedious formulas. The code of the TPSS (TPSS03) functional has been given by the authors to Gaussian, and because of copyright agreements cannot be given to others. For other functionals code for the higher derivatives is not implemented by others. This leads to the unsatisfactory situation that a lot of tedious and uninteresting work has to be spent to implement and test functionals.
A solution to this, that I developed especially for the TPSS functional (which is the most complex functional we have in cp2k), was to use Maple(Wat04) to define the functional. The formulas of the functionals are too complex to be calculated in just one step (the result is extremely long and very inefficient to calculate). So the basic idea is to define collections of equations of the form $\text{symbol} = \text{formula}$ that when executed in the correct order lead to the desired result. Using it one often can just copy the formulas from the paper at verbatim, making it easy to check for errors. In maple then one can write functions to check for inconsistencies between various definitions (and check that the same symbol in two different papers really means the same thing), to order them respecting the dependencies and find out which symbols are still undefined.

Once ordered the sequence of equations can be converted in a procedure that calculate the functional. This procedure can be automatically derived with respect to its arguments leading to a procedure that calculates the derivative. The procedure can be then transformed back into a sequence of equations. It is mandatory to never convert everything in one huge formula, because all the repetitions of a symbol would be replaced by the corresponding formula making the result too big.

All the equations to calculate the functional, and all it derivatives should then be merged back into a unique list removing duplicates and checking for inconsistencies. This list then is converted to a routine and finally converted to Fortran declaring the local variables global, and activating optimizations (that substitutes common expressions with temporary variables). This way optimized code is generated, and the overhead to calculate the derivatives is not too big; and by keeping the name of the local variables in the optimized code it is still possible to understand what is being calculated where.

The TPSS calculation has the extra difficulty that it contains a maximum between two functions (the PBE correlation of single spin and double spin), and the automatic derivation of maple cannot cope with it, so calculating the derivatives in the two cases and adding an if statement by hand was needed.

Maple notebooks to perform these calculations are part of the cp2k distribution (CP204) in cp2k/doc. These notebooks can be used to generate code also for C and other languages.
2.4 Gaussian and plane waves method

2.4.7 Linear scaling Kohn-Sham matrix construction

In order to obtain linear scaling Kohn-Sham matrix construction with the GPW energy functional, only two steps are needed. The first step is a screening to eliminate all negligible terms that involve a product of two Gaussian basis functions, and exploits the fact that such products are negligible once the Gaussian centers are sufficiently far apart (see also section 2.7.5.1). In this way, the number of remaining non-zero terms scales linearly with system size, and this strategy is employed for most terms in Eq. 2.20 and in the definition of \( n(r) \). To this end a list of neighboring atoms is created for each atom.

The neighbor lists are constructed using a linked-list method in combination with a partitioning of the simulation cell into sub-cells. This is a standard technique employed by Monte-Carlo and classical molecular dynamics codes which allows to build the neighbor lists in a time scaling linearly with the number of particles in the simulation cell (FS02). In QUICKSTEP, an individual sub-cell size is defined for each pair of atomic kinds in the system depending on the sum of their interaction radii, which leads to an adaptive cell partition scheme. Moreover, the work of Bekker et al. (BDRB95), in which instead of calculating the interactions of a particle in central unit cell with its periodic images one calculates the interactions of the periodic images with the central unit cell, was generalized for arbitrary interaction ranges. Using it the most inner loop for the neighbor search runs over the atoms in the central unit cell and not over the periodic images. This way the computation of the periodic boundary conditions in the inner loop can be avoided.

A second step is the transformation of the density \( \tilde{n}(r) \) to \( \tilde{n}(G) \) using Fast Fourier Transforms (FFT). This efficient step is formally \( O(N \ln N) \), but often still considered linear scaling since \( O(N \ln N) < O(N^{1+\epsilon}) \). We have illustrated this behavior in Fig. 2.4 for samples of liquid water in a periodically repeated cubic unit cell using a DZVP/DZV basis, a 200 Ry cutoff, a LDA functional, a screening threshold of \( 10^{-10} \) and converging the energy to better than \( 10^{-12} \) a.u. / atom using a traditional diagonalization scheme. From this data we can conclude that the GPW method exhibits linear scaling behavior with a small prefactor for the Kohn Sham matrix build up even for three dimensional and relatively small systems.
Figure 2.4: Shown are the timings of a full wave function optimization for liquid water samples of increasing size using QUICKSTEP on a single Power4+ CPU. A relatively small DZVP/DZV basis (17 basis functions per molecule) has been employed. The full execution time is, to very good accuracy, the sum of the linear scaling Kohn-Sham matrix construction (diamonds) and the cubic diagonalization/DIIS based density matrix update (circles). The solid lines are single parameter fits using the ideal forms $a_1N$ and $a_3N^3$ respectively, and the quality of this fit confirms convincingly the linear scaling nature of the GPW method and its implementation. Triangles indicate the fraction of time spent in the FFT routines as part of the Kohn-Sham matrix constructions, whereas the dashed line is a fit using $a_{FFT}N \ln N$ as a functional form. The insert shows the same Kohn-Sham matrix construction data using linear axes.
2.5 GPW forces

2.5.1 Deriving the Kohn-Sham matrix from the GPW energy

In this section we present how the exact derivative $H_{\mu\nu} = \partial E / \partial P^{\mu\nu}$ of the total energy is computed taking into account all approximations that lead from the Gaussian based density $n(P^{\mu\nu})$ to the density represented on the grid $\tilde{n}(P^{\mu\nu})$. This includes the mapping from the Gaussian basis to the grid using finite radii, the use of multi-grids, and of grid based methods to compute $\nabla \tilde{n}(r)$. We use the notation $\tilde{n}_i$ to denote a value of $\tilde{n}(r)$ on a particular grid point with coordinates $r_i$ using a single index $i$ for notational convenience. Truncation to a finite radius of the products $\phi_\mu(r)\phi_\nu(r)$ is equivalent to summing over a subset $\{\mu', \nu'\}$ of $\{\mu, \nu\}$ in the definition

$$\tilde{n}_i = \sum_{\{\mu', \nu'\}} P^{\mu'\nu'} \phi_{\mu'}(r_i)\phi_{\nu'}(r_i). \quad (2.42)$$

The derivative of $E(P^{\mu\nu}, \tilde{n}(P^{\mu\nu}))$ will be obtained explicitly using the chain rule as

$$H_{\mu\nu} = \frac{\partial E}{\partial P^{\mu\nu}} + \frac{\partial E}{\partial \tilde{n}_i} \frac{\partial \tilde{n}_i}{\partial P^{\mu\nu}}, \quad (2.43)$$

where summation over repeated indices, such as $i$, is implicit. We refer to $\partial E / \partial \tilde{n}_i$ as the potential on the grid, and use $v_i$ as an abbreviation.

An example term is the Slater exchange energy which we compute as

$$E_x^{\text{slater}} = \sum_j dr \ C \tilde{n}_j^4 \quad (2.44)$$

where $dr$ is a volume element, and the corresponding derivative is given as

$$\frac{\partial E_x^{\text{slater}}}{\partial \tilde{n}_i} \frac{\partial \tilde{n}_i}{\partial P^{\mu\nu}} = \sum_j dr \ C \frac{4}{3} \frac{1}{\tilde{n}_j^2} \delta_{ij} \frac{\partial \tilde{n}_i}{\partial P^{\mu\nu}} \quad (2.45)$$

$$= dr v_i \frac{\partial \tilde{n}_i}{\partial P^{\mu\nu}}.$$

The computation of the term $\partial \tilde{n}_i / \partial P^{\mu\nu}$ needs to respect that the sum in Eq. 2.42 is only over a subset $\{\mu', \nu'\}$. This amounts to performing the integration

$$v_i \frac{\partial \tilde{n}_i}{\partial P^{\mu\nu}} dr = v_i \phi_\mu(r_i)\phi_\nu(r_i) dr \quad (2.46)$$

over exactly the same grid points that have been used in the mapping of $\phi_\mu(r)\phi_\nu(r)$. 


Only slightly more involved is the case where the density functional depends on the density and on the gradient of the density. The gradient is \( g_j = D_j(\{n_i\}) \), where the derivative operator \( D_j \) can be a function of all grid points, e.g. if the discrete Fourier transform is employed for computing the gradient, or e.g. a local function of the grid for a finite difference approximation \( g_j = (\tilde{n}_{j+1} - \tilde{n}_{j-1})/(2\Delta) \). Using the chain rule leads to

\[
\frac{\partial E(\tilde{n}_i, g_i)}{\partial P^\mu} = \frac{\partial E(\tilde{n}_i, g_i)}{\partial \tilde{n}_i} \frac{\partial \tilde{n}_i}{\partial P^\mu} + \frac{\partial E(\tilde{n}_i, g_i)}{\partial g_j} \frac{\partial g_j}{\partial \tilde{n}_i} \frac{\partial \tilde{n}_i}{\partial P^\mu}
\]

valid for all choices of \( D \). Computing \( \partial E(\tilde{n}_i, g_i)/\partial g_j \) and \( \partial D_j(\{n_i\})/\partial \tilde{n}_i \) is the numerical equivalent of performing a partial integration.

Furthermore, we consider, as an extension of the basic scheme, a multi-grid method in which \( \phi_n(r)\phi_n(r) \) is mapped to a different grid, according to the smoothness of the Gaussian product. The total density needs to be obtained on the finest grid using an operator that performs the interpolation. The value on the fine mesh is obtained as \( n_f^j = (P_f(\{n_c^i\})_j \) where \( f \) and \( c \) imply the fine and the coarse mesh respectively, and the interpolating operator \( P_f^c \) might depend on all grid points. Therefore the derivative will involve terms like

\[
v_j \frac{\partial I_j(\{n_c^i\})}{\partial n_c^i} \frac{\partial n_c^i}{\partial P^\mu}.
\]

The index \( j \) in the above expression runs over all grid points of the fine mesh, whereas the index \( i \) goes over the coarse mesh. The term \( v_j \partial I_j(\{n_c^i\})/\partial n_c^i \) can be interpreted as an operator \( R_c^j \) applied on \( v_j \) that constructs the potential on the coarse grid.

Finally, we note that in the particular cases of a Fourier space derivative for \( D_j \) and a Fourier interpolation for \( P_f^c \), it is advantageous to use a \( g \)-space representation for all densities and operators involved, as all derivatives such as \( \partial D_g(\{\tilde{n}_g\})/\partial \tilde{n}_g \) are diagonal (i.e. \( \propto \delta_{gg'} \)).

See section 2.7 for a more in depth treatement of the multigrid techniques and of the operators \( P_f^c \) anf \( R_c^j \), that, in all the methods implemented, are linear in \( \{n_c^i\} \).

### 2.5.2 Forces on the ions

The ionic forces can be evaluated by computing explicitly the gradient of the GPW energy as defined by Eqs. 2.20 and 2.29 with respect to the atomic positions. This derivative must
take the atom centered nature of the Gaussian basis set and the orthogonality constraints on the wave functions into account. We list the required derivatives in the following. These are computed analytically for all terms except for the Coulomb and exchange and correlation terms that are computed on the grids, consistent with their definition.

The derivatives of the density independent terms are given by

$$\nabla_i E^{\text{core}} = \sum_{j \neq i} \frac{R_j - R_i}{|R_j - R_i|^2} \left\{ \frac{Z_i Z_j}{|R_i - R_j|} \text{erfc} \left[ \frac{|R_i - R_j|}{R_i^2 + R_j^2} \right] + \frac{2}{\sqrt{\pi}} \frac{Z_i Z_j}{\sqrt{R_i^2 + R_j^2}} \exp \left[ - \frac{|R_i - R_j|^2}{R_i^2 + R_j^2} \right] \right\}$$

$$\nabla_i E^{\text{self}} = 0$$

All other terms depend directly on the density matrix $P^{\mu \nu}$, but involve only derivatives of Cartesian Gaussian functions, that can easily be computed, since these are again Cartesian Gaussian functions, but with different $l$ quantum number. In this derivation, we follow closely Ref. (Pul69) and introduce explicitly the derivatives $\nabla_i P^{\mu \nu}$ which are afterwards related to the orthogonality constraints on the wave function. We define $E^{\text{core}}$ and $H^{\text{core}}_{\mu \nu}$ as the energy and matrix elements due to the electronic kinetic energy, the short range part of the local pseudo potential, and the non-local pseudo potential to obtain

$$\nabla_i E^{\text{core}} = \sum_{\mu \nu} (\nabla_i P^{\mu \nu}) H^{\text{core}}_{\mu \nu} + \sum_{\mu \nu} P^{\mu \nu} (\nabla_i H^{\text{core}}_{\mu \nu})$$

$$= \sum_{\mu \nu} (\nabla_i P^{\mu \nu}) H^{\text{core}}_{\mu \nu}$$

$$+ \sum_{\mu \nu} P^{\mu \nu} \left[ 2 (\nabla_i \varphi_{\mu}(r)) - \frac{1}{2} \nabla^2 |\varphi_{\nu}(r)| \right]$$

$$+ 2 (\nabla_i \varphi_{\mu}(r)) |V_{LR}^{SR}(r)\varphi_{\nu}(r))$$

$$+ 2 (\nabla_i \varphi_{\mu}(r)) |V_{LR}^{PP}(r, r')\varphi_{\nu}(r'))$$

$$+ (\varphi_{\mu}(r)) |\nabla_i V_{LR}^{SR}(r)\varphi_{\nu}(r))$$

$$+ (\varphi_{\mu}(r)) |\nabla_i V_{LR}^{PP}(r, r')\varphi_{\nu}(r'))$$

(2.50)

where the number of terms is already reduced by regrouping terms, exploiting symmetry of $P^{\mu \nu}$ and $H^{\text{core}}_{\mu \nu}$. Furthermore, the translational invariance of the first derivatives is exploited for the force calculation in QUICKSTEP using identities such as e.g.

$$\langle \nabla_i \varphi_{\mu}(r) | - \frac{1}{2} \nabla^2 |\varphi_{\nu}(r)| \rangle = - \langle \varphi_{\mu}(r) | - \frac{1}{2} \nabla^2 |\nabla_j \varphi_{\nu}(r)| \rangle$$

$$\langle \nabla_i \varphi_{\mu}(r) | V(r) | \varphi_{\nu}(r) \rangle + \langle \varphi_{\mu}(r) | V(r) | \nabla_j \varphi_{\nu}(r) \rangle = - (\varphi_{\mu}(r) | \nabla K V(r) | \varphi_{\nu}(r))$$
where $\varphi_\mu(r)$, $\varphi_\nu(r)$ and $V(r)$ are located on the atoms $I$, $J$ and $K$ respectively.

The density dependent energy terms are computed using the chain rule with the density as an intermediate variable to yield the following derivatives

$$\nabla I E^H[\rho_{\text{tot}}] + \nabla I E^{\text{xc}}[\rho] = \sum_{\mu\nu} (\nabla I P^{\mu\nu}) V^{\text{tot}}_{\mu\nu}$$

$$+ 2 \sum_{\mu\nu} P^{\mu\nu} \int (\nabla I \varphi_\mu(r)) v^{\text{tot}}(r) \varphi_\nu(r)dr$$

$$+ \int (\nabla I n^I_c(r)) v^H(r) dr$$

(2.51)

where $v^{\text{tot}}(r) = v^H(r) + v^{\text{xc}}(r)$.

In the above equations, the terms involving $\nabla I P^{\mu\nu}$ can be collected and rewritten using the Kohn-Sham matrix $K_{\mu\nu}$ as

$$\sum_{\mu\nu} (\nabla I P^{\mu\nu}) (H_{\text{core}}^{\text{K-S}} + V^{\text{tot}}_{\mu\nu}) = \sum_{\mu\nu} (\nabla I P^{\mu\nu}) K_{\mu\nu}$$

(2.52)

The derivative of the density matrix can be eliminated by expanding the density matrix in terms of the wave function coefficients, inserting the Kohn-Sham equations (Eq. 2.53), and simplifying the expression using the derivative of the orthogonality constraints on the wave functions (Eq. 2.54).

$$\sum_\nu K_{\mu\nu} c_\nu^\rho = \varepsilon_\mu \sum_\nu S_{\mu\nu} c_\nu^\rho$$

(2.53)

$$\nabla I \sum_{\mu\nu} c_\mu^\rho S_{\mu\nu} c_\nu^\rho = 0$$

$$2 \sum_{\mu\nu} (\nabla I c_\mu^\rho) S_{\mu\nu} c_\nu^\rho = - \sum_{\mu\nu} c_\mu^\rho (\nabla I S_{\mu\nu}) c_\nu^\rho$$

(2.54)

This leads to

$$\sum_{\mu\nu} (\nabla I P^{\mu\nu}) K_{\mu\nu} = \sum_{\mu\nu} \sum_i \left[ (\nabla I c_\mu^{\rho_\mu}) S_{\mu\nu} c_\nu^{\rho_\nu} + c_\mu^{\rho_\nu} (\nabla I c_\nu^{\rho_\mu}) \right]$$

$$= 2 \sum_{\mu\nu} \sum_i \varepsilon_\mu \left( (\nabla I c_\mu^{\rho_\mu}) S_{\mu\nu} c_\nu^{\rho_\nu} \right)$$

$$- \sum_{\mu\nu} \sum_i \varepsilon_\mu c_\mu^{\rho_\nu} (\nabla I S_{\mu\nu})$$

$$- \sum_{\mu\nu} W^{\mu\nu} (\nabla I S_{\mu\nu})$$

$$= -2 \sum_{\mu\nu} W^{\mu\nu} (\nabla I \varphi_\mu(r) | \varphi_\nu(r) )$$

(2.55)
where the energy weighted density matrix $W^{\nu\nu}$ is introduced. It can be observed that this force contribution is easily calculated as it only involves derivatives of overlap matrix elements. This term was originally derived by Pulay (Pul69) and is only present if the basis set is atom position dependent.

### 2.6 Program structure and implementation

QUICKSTEP is tightly integrated in a larger program named CP2K and the Fortran 95 sources are made freely available (CP204). Currently, good portability and efficiency is achieved by relying on a few commonly available libraries such as BLAS (BLA), LAPACK (LAP), FFTW (FFT), MPI (MPI) and ScaLAPACK (Sea), and restricting ourselves to standard Fortran with OpenMP (Ope) extensions. In this section, we present our approach to structural aspects of the code, and describe how good efficiency can be obtained for collocating and integrating Gaussian products on a grid.

#### 2.6.1 Algorithm environments

In order to deal with the growing complexity of the code in which e.g. all algorithms must be able to run concurrently and interfaces evolve over time, a more object-oriented approach is employed in CP2K. Each major algorithm has an associated 'environment' i.e. a derived type that contains all necessary data structures to allow for abstracting and simplifying the interfaces and splitting the algorithms. Access to components of the environment is through get/set functions. In this way, a certain abstraction can be introduced so that it is e.g. possible to hide the actual method used to compute the forces on the atoms (force.env), and provide generic modules that perform the molecular dynamics or geometry optimization independent on whether forces and energies are calculated ab-initio, classically, or in some other way. The various phases of the algorithm are well separated as follows:

1. creation of a computational environment c.env, initializing to default values for the components
2. optionally, set components of c.env to adapt it to a specific task
3. call the routine that actually performs the algorithm with a simple interface and passing c.env as parameter

4. call the cleanup routine that destroys c.env

the points 2-3 can be repeated more than once, and the point 3 can easily be further split into smaller subroutines. This pattern is quite widespread in the code (qs.env for the whole DFT part, scf.env for the SCF cycle in QUICKSTEP, ks.env for the final assembly of the Kohn–Sham matrix,...).

The use of module environments is what has made QM/MM, Monte Carlo and perturbation simulations possible. The ability to have more than one simulation going on has been crucial. But this approach has a problem in Fortran 90. This derives from the fact that to use a function or a type you must "use" the module that contain their definition, so that the compiler knows their interface. This makes it impossible to build recursive functions/structures across modules.

It might seem a rather theoretical constraints, but many high level approaches need a way to update the energy and forces of sub systems to calculate its energy/force. This is the case with QM/MM and the perturbation approach.

If the composite system should be seen as a force.env, like the sub systems, one is quickly trapped in the recursive definition. Defining things in the same module allows recursive definitions, but it is not the perfect solution as you don’t want to define everything inside the same module.

2.6.2 F77 Interface

The application programming interface given by the algorithm environments is very flexible, but sometime one would like to have a simpler interface. To interface to external programs that don’t use Fortran (and see cp2k as library) the ideal thing would be a Fortran77 interface (that is easily called also by c).

To achieve this one should have a flat interface where only simple types are passed. The force.envs (and a couple of other environments) are identified by a unique integer (id.nr) and internally a mapping id.nr → environment is kept.
The development of this interface initially had the goal of making cp2k a linkable library. Now it is used also inside cp2k. It turned out that a big advantage of it (apart its simplicity) was the it could avoid the recursive call problem of the force.env. Indeed to call a Fortran77 function you don’t need to already have compiled it.

This was used this to have many replica of the system running in parallel (replica.env), and for the perturbation approach. I expect that it will be very useful for many higher level applications.

2.6.3 Input

The input is an important part of the program, as it is the interface with the user. A weak point of open source programs is very often the documentation, and for the input this is even more crucial.

The programmer of a feature normally knows how the input for its part of the program works, and the meaning of the various flags. As programmer he/she prefers to spend its time making the feature work, using it, rather than documenting it. Also if the program is really in active development the documentation can become outdated very easily.

The ability to have multiple computations going on in parallel means that one would like to have an automatic way to change the input and to generate valid inputs within the program, i.e. to have a more formalized input.

To try to address these problems we developed a new input structure. It has a recursive structure, composed of sections that contain other sections and keywords. Keywords have a value that is a simple type (logical, real, integer, character), or an array of them, and can have a default value.

The whole input structure has to be declared explicitly in the program (you cannot read a keyword if you have not declared its existence before). To make the presence of at least a minimum up-to-date documentation the declaration of the keyword/section has to include a small string that describe it. From the input structure an HTML manual and an XML description can be automatically generated.

The parser checks that the input conforms the input structure. Then the whole parsed input is kept in memory, in a way that reflects the recursive structure (the whole input
36  2 Ab-initio MD

is just like a section), and subsection can be passed around. The input can be queried (and modified) also using a string that represent the path of the keyword. Keeping the whole input in memory makes the handling of it very comfortable, and we thought that the memory usage of it (memory that can be paged out if not used) was acceptable.

The restart file of cp2k is just an input that contains all the informations to continue the simulation. If you would like to take only some information from the input two inputs can be merged, taking for example to positions from the restart input.

This uniform and automatic handling of the input was used to good effect especially in the (later versions of) Monte Carlo simulations and in the perturbation approach.

2.6.4 Memory management

Two techniques are employed to reduce the memory requirements of QUICKSTEP. To safely share instead of copying fairly large objects such as e.g. descriptors for plane wave grids or matrices, reference counting is employed. In this way, a variable within the objects is used to track how many pointers to the object are in use. Whenever a pointer to the object is assigned (i.e. the object is retained), this counter is increased, and the counter is decreased as soon as the object is released. The object is automatically deallocated as soon as the counter becomes zero.

The second technique is aimed at reducing memory fragmentation and time spent in dynamically allocating and deallocating large objects. Static allocation relies on the knowledge that at a given stage of an algorithm a variable can be overwritten and makes it difficult to change the algorithm, or to allow for variants that use a different sequence of operations. Pools have been introduced as a solution to this problem. Temporarily unused objects are given to a pool that stores these objects in a cache, and objects are created rapidly if the pool can return such a cached element.

2.6.5 Parallelization strategy

In order to make a program scale in parallel it is necessary to distribute at least part of the data and tasks between the different processors, without introducing a large amount of communication between the different CPUs. There are currently two layers of paral-
2.6 Program structure and implementation

Parallelization in CP2K. A standard message passing interface that is available on most parallel machines (MPI (MPI)) provides the main layer, and this layer is augmented using an application program interface that supports shared-memory parallel programming (OpenMP (Ope)). In our applications, OpenMP is typically used in addition to MPI with just a few threads per MPI task. The MPI tasks can be addressed through a one dimensional (1d) rank $m = 0..N-1$ or through two dimensional (2d) coordinates $(p, q) = (0..P-1, 0..Q-1)$ with $N = P \times Q$.

Most data related to the structure of the whole system, such as lists describing the molecules, atomic kinds, coordinates, layout of the grids and matrices are replicated. A 1d mapping $d$ that given the atom number $i$ returns the 1d rank $m = d(i)$ of the task that should hold/work on the data is typically used for atom related operations, such as e.g. computing $n_e(r)$.

A two dimensional mapping that given the indices of the atom pair $(i, j)$ returns the 2d coordinates $(p, q)$ of the task that owns that pair is employed for operations that should be performed on pairs of atoms, e.g. from computing the interaction between atoms $i$ and $j$. This 2d distribution is built up from two 1d distributions $d_1, d_2$ distributing over processor rows and columns respectively. For $i < j$ the couples $(i, j)$ and $(j, i)$ are local to the processor $(p, q)$ with

\[
\begin{align*}
  p &= d_1(i) \text{ and } q = d_2(j) \text{ if } (i + j \mod 2) = 0 \\
  p &= d_1(j) \text{ and } q = d_2(i) \text{ if } (i + j \mod 2) = 1
\end{align*}
\]

The $d_1$ and $d_2$ distributions keep in account the size of the basis of the atoms, and try to build a well balanced 2d distribution. The storage required for the full 1d distribution is small, so that replicating this data enables each processor to calculate rapidly the location of any $(i,j)$ pair. The atomic blocks of the sparse matrices with the same structure as the overlap matrix are distributed using this 2d distribution.

All dense matrix linear algebra is based on ScaLAPACK (Sca) and thus fully parallelized. These full matrices e.g. used to represent the coefficients of the orbitals, or the Kohn-Sham matrix during the diagonalization are distributed using the format imposed by ScaLAPACK, which is block cyclic in both dimensions with blocks of constant size (BCC'97). The sparsity of the overlap and Kohn-Sham matrix is however exploited in the orbital transformation method (section 2.9) and an efficient matrix multiply routine has been implemented to multiply a sparse matrix with a dense matrix (i.e. overlap/Kohn-Sham matrix and orbital coefficients). This implementation provides uniformly good
performance for the cases where the sparsity of the $S$ matrix is nearly absent or on the order of 10% as is encountered for the largest 3d systems (see also Table 2.3). The efficiency comes from exploiting the atomic block structure of $S$, which allows for relatively high peak performance, limiting the number of matrix multiplies by skipping zero atomic blocks, and distributing the blocks according to Eq. 2.56 which allows for reducing the communication for the orbital coefficients by a factor of $\min(P, Q)$ without introducing the need for communicating blocks of the sparse matrix.

The usual x-slices distribution, where y-z planes of the grid are distributed among the processors, is employed for plane waves grids, and the parallelization of the FFT follows the implementation used in CPMD (CPM; MH00). The grids used for the collocation are either fully replicated or are distributed in x-slices with overlapping borders. These overlapping borders allow for collocating the Gaussian products without the need for treating boundaries in a special manner during the collocation. Currently, the parallelization of this step appears to be the bottleneck to obtain good scaling on hundreds of CPUs.

\section*{2.7 Splines}

\subsection*{2.7.1 Multi grid}

Multi grid methods instead of just a fine grid $G_f$ use other coarser grids. These grid levels are ordered from the most coarse $G_c = G_1$ to the finest $G_f$ ($1 = c \leq f$). In 3D all the coarser grids do not cost much in term of memory (typically 0.14-0.4 times the memory of the fine grid). Adding these extra grids is useful because each one can represent a given wavelength in an optimal way (i.e. with a minimal number of points), and perform operations on this wavelength efficiently. Typically operations on each grid level are local and work on patches of neighboring points, and after a series of them one collects the result on the fine grid.

Multi grids methods can be used to solve linear equations on a grid, for example partial differential equations, but they aren’t yet used for this purpose in cp2k. We use multi grids to transfer the density from the Gaussian basis set to the grid trying to use a constant number of points per Gaussian, as described in (LHP99), and in QM/MM to
transfer the MM potential on the grid.

Multi grid is interesting only if there is an efficient way to transfer the operations
done on one grid level to the others. For \( i < j \) the transfer functions

\[
P^i_j : \mathcal{G}_i \rightarrow \mathcal{G}_j \\
R^j_i : \mathcal{G}_j \rightarrow \mathcal{G}_i
\]

(2.57) (2.58)

are called prolongation and restriction respectively.

If one wants that integration of a function \( h \) defined on a finer grid \( \mathcal{G}_j \) with a function \( g \) defined on a coarser grid \( \mathcal{G}_i \) gives the same result both transferring \( g \) to the fine grid and
summing there or (more efficiently) transferring \( h \) to the coarse grid and then summing
there one has

\[
\langle P^i_j g, h \rangle = \langle g, R^j_i h \rangle
\]

(2.59)

i.e. the projection is the dual of the restriction.

The prolongation can be seen as an interpolation: given the values on a coarse grid
try to find the values on a finer grid. In general one can also imagine a continuous function
that underlies the prolongation operation. A very good method for grids with periodic
boundary condition is the G-space interpolation. With a fast Fourier transform (FFT)
one can find the G-space representation \( h^3_k \) of the points on the grid. Then a continuous
representation of them would be

\[
n(r) = \sum_{ijk} \tilde{n}_{ijk} \exp(G(i, j, k) \cdot r),
\]

(2.60)

where \( G(i, j, k) = 2\pi h^{-1}[i, j, k], h^{-1} \) is the inverse of the cell vectors matrix, and \( i, j, k \) are
evenly distributed between the positive and negative values. The G-space interpolation
can be performed directly in the G-space, without going in the direct space. Indeed the
\( G(i, j, k) \) of the coarser grid are a subset of the ones of the finer grid, and the mapping
is trivial, taking care that for an even number of grid points you assign half the value to
\( N/2 \) and half to \(-N/2\).

The continuous function underlying the G-space interpolation is \( C^\infty \) (i.e. smooth,
ininitely often differentiable), and is the best interpolation scheme (wrt. to \( L^2 \) norm) for
points that come from a periodic \( C^\infty \) function. Unfortunately if the points come from a
function which is not smooth or for a non periodic function this is no longer true.
In cp2k non smoothness is present because at the core there is a jump in the derivative (cusp condition), and the exchange-correlation functionals, especially the gradient corrected ones, exacerbate the problem. This, as explained in the section 2.7 is due also to the pseudo potential we use in cp2k. Also introducing a cutoff for the Gaussian loses their smoothness. It was in this setting that I initially introduced the spline approach described in the next section. This turned out to be more useful that we thought and an extension of it was used to cope with the non periodicity (with respect to the QM cell) of the potential in a QM/MM setting.

2.7.2 Periodic uniform splines

A uniform cardinal B-Spline of order 3 in 3d is a function $R^3 \rightarrow R$

$$f(x, y, z) = \sum_{ijk} c_{ijk} N^3(x - i)N^3(y - j)N^3(z - k), \quad (2.61)$$

that is controlled by the coefficients $c_{ijk}$.

$N^3$ is a piecewise polynomial function in $C^2$ with compact support that can be seen as the convolution of the characteristic function of $[-1/2, 1/2]$ ($\chi_{[-1/2,1/2]}$) with itself three times.

$$N^3(t) = \begin{cases} 
\frac{1}{6}(t + 2)^3 & -2 \leq t < -1 \\
-\frac{1}{2}t^3 - t^2 + \frac{2}{3} & -1 \leq t < 0 \\
\frac{1}{2}t^3 - t^2 + \frac{2}{3} & 0 \leq t < 1 \\
-\frac{1}{6}(t - 2)^3 & 1 \leq t < 2 \\
0 & \text{otherwise}
\end{cases} \quad (2.62)$$

2.7.3 Periodic prolongation/restriction

With this the prolongation operation can be defined as follow:

1. find the coefficients $c_{ijk}$ that interpolate the values $v_{ijk}$ on the coarse grid

2. evaluate the spline (2.61) on the fine grid to obtain the final values $w_{ijk}$
We define the function

\[ S^i : \mathcal{G}_i \rightarrow \mathcal{G}_i \quad (S^i)_{klmnop} = N^3(||k - n||)N^3(||l - o||)N^3(||m - p||) \]  

(2.63)

where \( ||x|| \) is introduced because of periodic boundary conditions, and means the smallest distance, for example for the dimension \( x \)

\[ ||x|| = ((x + N_x/2) \mod N_x) - N_x/2, \]

(2.64)

where \( N_x \) is the number of grid points in the dimension of \( x \). \( S^i \) maps the coefficients \( c_{ijk} \) to the values \( v_{ijk} \). This matrix is very sparse because the \( N^3 \) is different from 0 only for the nearest neighbor, i.e. for an integer value \( i \)

\[ N^3(i) = \begin{cases} 
\frac{1}{6} & \text{if } i = -1 \\
\frac{2}{3} & \text{if } i = 0 \\
\frac{1}{6} & \text{if } i = 1 \\
0 & \text{otherwise}
\end{cases} \]

(2.65)

The application \( S^i c \) can be seen as the convolution of the grid with the 3x3x3 stencil (indexed from -1 to 1) with values

\[ S_4 = \left( \frac{2}{3} \right)^3 \left( \frac{1}{6} \right)^{|i|+|j|+|k|} \]

(2.66)
which has values
\[
\left[ \frac{8}{27}, \frac{2}{27}, \frac{1}{54}, \frac{1}{216} \right]
\]
for center, face centers, edges, and vertices of the 3x3x3 cube. This is the same \(S_4\) operation used in the XC section (2.4.4) and can be implemented very efficiently.

Then the first step of the prolongation is
\[
c = (S^i)^{-1}v
\]  
(2.68)
which we calculate iteratively with a conjugated gradient solver, using
\[
\left[ \frac{8}{27}, -\frac{2}{27}, -\frac{1}{54}, -\frac{1}{216} \right]
\]
(2.69)
as approximate inverse for the first guess, and
\[
\left[ 4.096, -1.28, 0.4, -0.125 \right]
\]
(2.70)
as pre-conditioner. The pre-conditioner is generated by the 1d-values \([-1.6/4, 1.6, -1.6/4]\) in each direction. It was found by minimizing the condition number of \(S^i\) multiplied by operators generated from 1d-values, and then (slightly) further optimized in the program. With this in 10-15 iterations, independently of the size of the grid, a convergence to less than \(10^{-10}\) for both argument and residual can be achieved.

To evaluate the spline on the fine grid we use commensurate grids for efficiency reasons, which means that each grid has exactly the double of the number of points in every direction than the previous grid level. In this case it is useful to introduce the (rectangular) matrix,
\[
(T^i_{i+1})_{kmnop} = N^3\left(\frac{k}{2} - n\right)N^3\left(\frac{l}{2} - o\right)N^3\left(\frac{m}{2} - p\right)
\]
(2.71)
which is very sparse as for half integer the only nonzero values are
\[
N^3\left(\frac{i}{2}\right)_{i=-4.4} = [0, \frac{1}{48}, \frac{1}{48}, \frac{23}{48}, \frac{23}{48}, \frac{1}{48}, \frac{1}{48}, 0]
\]
(2.72)
Thus we have
\[
P^i_{i+1} = T^i_{i+1}(S^i)^{-1},
\]
(2.73)
and
\[
P^i_{i+1} = (P^i_{i+1})^T = (S^i)^{-1}(T^i_{i+1})^T.
\]
(2.74)
The interpolation between the other grid levels can be defined as the product of the cascade prolongation/restrictions from grid $i$ to grid $j$

$$P_j^i = \prod_{k=j-1}^{i} P_{k+1}^j = P_{j-1}^j \cdots P_{i+1}^i P_i^j,$$  \hspace{1cm} (2.75)

and

$$R_i^j = \prod_{k=i}^{j-1} R_{k+1}^i = R_{i+1}^i \cdots R_{j-1}^i.$$ \hspace{1cm} (2.76)

This approach works very well with periodic boundary conditions. The coefficients of the spline can be seen as the G-space coefficients of a Fourier transform. Like them they depend in a unique and global way from the values on the grid (direct space): any coefficient depends on the values of all the grid, but with the splines the weight of far away points decreases faster than with G-space interpolation, splines are more localized.

The coefficients define a continuous function that on the grid has exactly the values of the direct space, but that is defined everywhere, not just on the grid, and thus they can be used to interpolate the values, or transfer the function between grid levels. The continuous function defined by the cubic splines is $C^2$ (twice continuously differentiable). This is not optimal to interpolate smooth functions, but if the function to interpolate is not so regular (due for example to cutoff effects, or numerical instabilities) then the spline interpolation becomes better. In fact local discontinuities, being the spline more localized, spread around less as can be seen looking at the interpolation of a delta like function on a grid, which as was discussed in section 2.4.4 is similar to the situation that arises in the core of the atom with gradient corrected functions.

### 2.7.4 Non-periodic uniform splines

If one wants to go beyond the periodic boundary conditions the function $N^3$ cannot be used for the coefficients close to the border. Indeed using the $N^3$ function would force the function to go at 0 and with derivative 0 two units after the border, and what is worse (one can argue that what happens beyond the border is not relevant and is an artifact) a simple linear function cannot be interpolated exactly. This gives rise to border effects that cannot be neglected. This problem is important for QM/MM where the potential generated by the MM atoms is not periodic with respect to the QM cell. As already
stated the solution is to modify the form of the $N^3$ functions for the coefficients close to the border.

To find out how to modify the functions we will look at a generalization of the uniform cardinal splines. To simplify the discussion we will first look at a non-uniform B-Spline of order 3 in just 1 dimension. This is a parametric 1d line in a 2d dimensional space, i.e. a $\mathbb{R} \rightarrow \mathbb{R}^2$ function

$$g(u) = \sum_i P_i N^3(u - i), \quad (2.77)$$

where $P_i$ is an array (indexed by the integer $i$) of 2-dimensional vectors.

This looks complicated, but if one sets

$$P_i = [i, v_i] \quad (2.78)$$

then if we call the first component of $g$, $x$ and the second $h$

$$[x(u), h(u)] := g(u), \quad (2.79)$$

we see that the mapping $x(u)$ is the identity and

$$h(u) = h(x) = \sum_i v_i N(x - i) \quad (2.80)$$

and so $h$ is just a uniform cardinal spline.

Assuming that the lower boundary is at 0, we want to look at

$$P_i = [\max(0, i), v_{\max(0,i)}]. \quad (2.81)$$

As we can see for $u > 1 \quad u = x$, but for smaller values the correspondence breaks and the function gets really parametric. $x$ begins to change more and more slowly, and finally freezes at 0 when $u$ reaches $-1$. Now the correct way to redefine the $N^3(x - i)$ to functions $M_i(x)$ for $i$ close to the border (i.e. to 0) is

$$M_0(x(u)) = N^3(u + 2) + N^3(u + 1) + N^3(u) \quad (2.82)$$

$$M_1(x(u)) = N^3(u - 1)$$

$$M_2(x(u)) = N^3(u - 2)$$

and for $i > 2 \quad M_i(x) = N^3(x - i)$. 

Figure 2.6: The left panel shows the weights of the splines for \( i = -2..2 \). The dotted red splines (with \( i = -2..0 \)) have the same coefficient, so they have been summed up into the continuous black line. The right panel shows the value of \( x \) as function of \( u \)

To be able to directly represent \( M_{0..2}(x) \) one has to invert \( x(u) \)

\[
x^{-1}(t) = \begin{cases} 
\text{undefined} & t < 0 \\
\sqrt{6u - 1} & t < \frac{1}{6} \\
2\sqrt{2} \cos \left( \frac{1}{3} \pi + \arccos \left( \frac{3\sqrt{2}(t-1)}{4} \right) \right) + 1 & t < 1 \\
t & t \geq 1 
\end{cases}
\]  

(2.83)

With an explicit inverse one obtains a direct representation of the functions \( M_{0..2} \) shown in figure 2.7. We see that for the evaluation on a grid with spacing 1 only the weight exactly at the border has to be changed (to 1), whereas for an uniform refinement, i.e. to prolongate to a grid with spacing 1/2, the border and the points just before it have to be changed. Approximately \( M_{0..2} \) have the following values at the important points:

\[
\begin{align*}
M_0(0) &= 1 & M_0(\frac{1}{2}) &= 0.517977703393314356529532 \\
M_1(0) &= 0 & M_1(\frac{1}{2}) &= 0.46404459321371286940937 \\
M_2(0) &= 0 & M_2(\frac{1}{2}) &= 0.017977703393314356529531.
\end{align*}
\]  

(2.84)

Thus using the weights given by \( M_0, M_1, M_2 \) instead of the ones given by \( N^3 \) at the border the simplicity of the uniform spline schema can be kept and linear functions can
be correctly interpolated. The upper border is just symmetric.

In 3d we have to look at non-uniform B-Spline of order 3 in 3 dimensions, which are parametric 3d surfaces in a 4d dimensional space, i.e. a $\mathbb{R}^3 \rightarrow \mathbb{R}^4$ function

$$g(u,v,t) = \sum_{ijk} P_{ijk} N^3(u-i) N^3(v-j) N^3(t-k), \quad (2.85)$$

where $P_{ijk}$ is a 3d grid (indexed by the integer $i, j, k$) of 4-dimensional vectors.

Looking at it one can see that the fact that the weight functions are just a direct product of the 1d weighting functions is preserved with boundaries along the border of a box. Assuming that the lower left corner of the box is for $(i,j,k) = (0,0,0)$

$$f(x,y,z) = \sum_{ijk} v_{ijk} M_i(x) M_j(y) M_k(z), \quad (2.86)$$

with $M_i$ as defined in the 1d case.
2.7.5 Non-periodic prolongation/restriction

The prolongation and restriction operation can be calculated just as before

\[ P^i_j = \tilde{T}^i_j (\tilde{S}^i)^{-1} \]  \hspace{1cm} (2.87)
\[ R^i_i = (P^i_j)^T = ((\tilde{S}^i)^T)^{-1}(\tilde{T}^i_j)^T \]  \hspace{1cm} (2.88)

where \( \tilde{S}^i \) and \( \tilde{T}^i_j \) are different from \( S^i \) and \( T^i_j \) because they use \( M_i(x) \) instead of the \( N^3(x-i) \). This means that \( \tilde{S}^i \) differs from \( S^i \) only at the border where in each dimension 1 is used instead of 2/3 as weight and the 1/6 contribution from the neighboring point is ignored. This breaks the symmetry of \( S^i \) and makes the sum of the contributions of the weights \( c \) close to the border differ from 1. Likewise \( \tilde{T}^i_j \) differs from \( T^i_j \) only at the border using the values of (2.84).

The inversion of \( \tilde{S}^i \) is performed using the same approximate inverse as in the non-periodic case, but setting the weight to 1 instead of 2/3 at the border, and removing the 1/6 contribution from the point next to the border, as with \( \tilde{S} \). For the pre-conditioner the contribution from the weight \( c \) at the border are scaled in such a way that at the value at the border is 1 (i.e. not just setting the border to one, but also changing the contribution to the close-by \( v \)). With this method the same performance as in the periodic case can be achieved on big grids: \( \approx 12 \) iterations for \( 10^{-10} \) accuracy, \( \approx 20 \) for machine accuracy \( (10^{-14}) \). For small grids other approximate inverse and pre-conditioners (not based on the the periodic solution) would be better, but \( \approx 1/3 \) more iterations on the small grids is not costly, and not worth extra optimization.

Such a function can describe exactly hyper planes, is efficient to evaluate and has worked very well for the QM/MM implementation in cp2k.

2.7.5.1 Efficient mapping of product Gaussian on the real space grid

The transformation of the atomic orbital based density matrix to an electron density on a real space grid, and the complementary operation, i.e. the computation of matrix elements given a potential on a real space grid are important operations in the GPW method. This step needs to be performed at every SCF cycle, and might contribute significantly to the overall execution time of the algorithm. However, a Gaussian basis set allows for techniques that reduce the cost of this operation significantly if regular orthogonal grids
are employed. The described algorithm has a computational cost for computing integrals involving Cartesian Gaussian that scales effectively linearly in the \( l \) quantum number.

A first step in the algorithm is the transformation of the atomic orbital basis into primitive Cartesian Gaussian:

\[
g_{\alpha i} \equiv g_{\alpha i}^s(r) = x^i x \exp(-\eta x^2) y^i y \exp(-\eta y^2) z^i z \exp(-\eta z^2). \tag{2.89}
\]

The product of two primitive Gaussian \( g_{\alpha i}^s(r - A) \) and \( g_{\beta j}^s(r - B) \) is a Cartesian factor multiplied by a single Gaussian with center \( P = \frac{\eta_a A + \eta_b B}{\eta_a + \eta_b} \), exponent \( \eta_P = \eta_a + \eta_b \), and prefactor

\[
\exp\left(-\frac{\eta_a \eta_b}{\eta_a + \eta_b} |A - B|^2 \right). \tag{2.91}
\]

Based on this exponent, the total \( l \) quantum number, and the prefactor, the radius \( R \) of a sphere around \( P \) is computed where the Gaussian product is non-negligible to within some threshold, typically \( 10^{-10} - 10^{-14} \). Only grid points within this sphere are used in the integration and mapping. As shown in section 2.5.1, consistency between the Kohn-Sham matrix and the GPW energy can be maintained.

It is exploited that the product can be factorized in three parts that each depend only on a single variable \( x, y \) or \( z \). Each part can be precomputed on a 1d grid, the full value of the Gaussian product on any point of the 3d grid is a product of 3 precomputed numbers. In particular, the inner loop on the grid (e.g. \( x \) component) will just be a single multiplication of a constant (depending on \( y \) and \( z \)) with number stored in a 1d array (independent of \( y \) and \( z \)). Furthermore, all products that have the same center, independent of the \( l \) quantum number, can be computed simultaneously by expanding for each term the factors \( (x - A)^{l_x} (x - B)^{l_x} \) and summing the prefactors of identical terms. The inner loop contains in that case just \( l_x + l_y + 1 \) terms, each one can be precomputed as described above. Therefore \( O(l_x^3 l_y^3) \) terms are computed in approximately \( O(l_x + l_y) \) time, contributing greatly to the efficiency of the algorithm. In particular, basis sets involving high angular momentum are relatively inexpensive, even if derivatives for e.g. the kinetic energy density or the forces are required.

We describe in section 2.5.1 how to define consistently the Kohn-Sham matrix if multi grids are introduced in the GPW energy functional. In the multi grid method,
the exponent $\eta_p$ of the Gaussian product is used to select a grid so that the number of
grid points per $\sigma_p^2 = 1/2\eta_p$ is approximately independent of $\eta_p$. The accuracy of the
multi grid method is fixed specifying a single number (number of grid points per $\sigma_p$) which is conveniently expressed as the plane wave cutoff of the grid used for Gaussian
with $\eta_p = 1$. We employ 30 Ry as a default relative cutoff as this is both efficient and
accurate. Furthermore, it is not necessary to have multi grids for each exponent, and in
our experience it is sufficient to have $N$ grids with
\begin{equation}
E_{\text{cut}}^i = \frac{E_{\text{cut}}^{1-i}}{\alpha^{i-1}} \quad i=1..N
\end{equation}
where $\alpha = 3.0$. The necessary $N$ and $E_{\text{cut}}^{1}$ depend on the smallest and largest exponents
of the Gaussian basis set used, but are typically four and 280 Ry respectively.

2.8 Basis sets

The electron density, and all the operators to be representable in the computer need to
be expanded in a finite basis. This obviously introduces an error, and the exact form and
amount of the error depend on the basis set used. Plane waves, or any representation
that treats all the points of the space in the same way has a very natural way to control
the error: the grid spacing (or energy cutoff).

The density changes a lot more close to the core of the atom, so it make sense to use
a basis that has more variability close to the core. The Gaussian orbital functions that
we use in the GPW method are such a basis. Having a non-uniform spatial dependence,
actually a localized atom centered basis that can be integrated analytically as we do has
many advantages, but together with the advantages there are also drawbacks. The error
is not anymore controlled by just one easy parameter. Even if the variational principle
guarantees that the error in absolute energy will always decrease increasing the basis set,
one can get worse relative energies and forces.

This is connected with the Basis Set Superposition Error (BSSE, see also section 2.11.1),
which is an error due to the dependence of the basis from the atomic position. A non
uniform variational flexibility of the basis in space gives an artificial lowering of the energy
of those atomic configurations in which the exact solution can be better approximated.
Thus a smaller basis set with a uniform (almost constant) error can be better than a large
basis set, which is very accurate only in some configurations.
For this reason care is needed in the construction of basis sets of increasing accuracy. Significant experience exists with Gaussian basis sets and there are a number of formats (HT95; A.A). Whereas polarization and diffuse functions can normally be adopted from published basis sets, the valence part of the basis has to be generated for the usage with the GTH pseudo potentials. A systematically improving sequence of basis sets for use with the GTH pseudo potentials was optimized by Matthias Krack for all first- and second-row elements, using the procedure detailed below.

Exponents of a set of primitive Gaussian functions were optimized to yield the lowest pseudo atom energies for all first- and second-row elements with an atomic DFT code employing the appropriate GTH potential for each element. The atomic DFT code allows for the calculation of first analytic derivatives of the total atomic energy with respect to the Gaussian orbital exponents. A family basis set scheme was adopted using the same set of exponents for each angular momentum quantum number of the occupied valence states, i.e. s and p orbitals for the elements from H to Ar. A growing number of primitive Gaussian functions, typically four to six, was included into these sets to provide an increasingly good description of the pseudo atomic wave function.

Finally, these primitive Gaussian functions were contracted using the coefficients of the respective pseudo atomic wave functions. In addition, a split valence scheme was applied to enhance the flexibility of the valence basis part. The splitting was increased in line with the number of primitive Gaussian functions employed from double- (DZV) over triple- (TZV) up to quadruple-zeta valence (QZV). For instance, the basis set sequence for oxygen starts with four primitive Gaussian functions on the DZV level, uses five functions for TZV, and finally six on the QZV level. Moreover, these basis sets were augmented by polarization functions which were taken from the all-electron basis sets cc-pVXZ (X = D, T, Q) of Dunning (Dun89; WD93), but only the first p or d set of polarization functions was used depending on the actual element. In that way a new sequence of basis sets was created with an increasing number of primitive Gaussian functions and polarization functions for each first- and second-row element. The basis sets were labeled DZVP, TZVP, TZV2P, QZV2P, and QZV3P due the applied degree of splitting and the increasing number of provided polarization functions.

To show that the quality of the basis improves systematically within this sequence the optimized geometry with different basis sets of the following test set of 39 small molecules:
consisting of first- and second-row elements was compared with the results of with the
NUMOL results of Dickson and Becke (DB93). NUMOL is a purely numerical DFT code
and thus considered to be free of basis set effects. As can be seen in (VKM+05) the
smallest basis set DZVP gives on average slightly too long bond distances, but already
the TZVP basis set works fine for most of the molecules. Finally, the TZV2P, QZV2P,
and QZV3P show an satisfactory agreement for all bond distances, even if NUMOL uses
a frozen core approximation, and not a pseudopotential.

The O and H basis set are also throughly tested in condensed phase, in section 3.4.2.

If required, these basis sets can be further augmented by diffuse functions, analogous
to the aug-cc-pVXZ basis sets, resulting in a sequence aug-DZVP, aug-TZVP, aug-TZV2P,
aug-QZV2P, and aug-QZV3P. The inclusion of diffuse functions may significantly improve
the accuracy of certain molecular properties, reducing the BSSE error, especially in the
gas phase, but require treatment for the near linear dependencies that typically arise in
condensed phase calculations (see section 2.9). We will therefore not present any results
from calculations involving diffuse functions in the following.

2.9 Wave function optimization

The calculation of the electronic ground state amounts to the minimization of the elec¬
tronic energy (Eq. 2.20) with respect to the orthonormal one-particle orbitals or the
one-particle density matrix. Even though techniques exist to perform this minimization
in a time scaling linearly with the system size (Goe99), no such technique is currently
implemented in QUICKSTEP, and the algorithms show a cubic scaling with respect to the
number of atoms in the unit cell. Nevertheless, a careful design of the methods can make
them, especially on parallel computers, very efficient for e.g. molecular or condensed phase
systems of up to a few thousand atoms (see section 2.12). Two methods are currently
available in QUICKSTEP to minimize the total ground state energy of a system by an
iterative self-consistent field (SCF) procedure: a traditional diagonalization (TD) scheme
and an efficient orbital transformation (OT) method (VH03). In the following, the de-
scription is restricted to closed-shell systems, however, the generalization to open-shell (spin-polarized) systems is straightforward and QUICKSTEP can deal with both types of systems using both the TD and the OT method.

### 2.9.1 Traditional diagonalization

The TD scheme uses an eigensolver from a standard parallel program library (ScaLAPACK) to solve the general eigenvalue problem

\[ Kc = Sc\epsilon \]  

(2.93)

where \( K \) is the Kohn-Sham matrix and \( S \) is the overlap matrix. The eigenvectors \( c \) represent the orbital coefficients and the \( \epsilon \) are the corresponding eigenvalues. Unfortunately, the overlap matrix \( S \) is not the unit matrix, since QUICKSTEP employs an atom-centered basis set of non-orthogonal Gaussian-type orbital functions. The eigenvalue problem is therefore transformed from the generalized to the standard form

\[ Kc = UTU^T c' = \epsilon c' \]  

(2.95)

\[ K'c' = c' \epsilon \]  

(2.96)

using a Cholesky decomposition of the overlap matrix

\[ S = UTU \]  

(2.97)

as the default method for that purpose. Eq. 2.96 can be solved by diagonalization of \( K' \), and the orbital coefficients \( c \) in the non-orthogonal basis are obtained by backtransformation

\[ c' = Uc \]  

(2.98)

The names in brackets denote the employed ScaLAPACK routines.

Alternatively, a symmetric orthogonalization instead of a Cholesky decomposition can be applied by using

\[ U = S^{1/2} \]  

(2.99)

However, the calculation of \( S^{1/2} \) involves a diagonalization of \( S \) which is computationally more expensive than a Cholesky decomposition. On the other hand, linear dependencies in the basis set introduced by small Gaussian function exponents can be detected.
Eigenvalues of $S$ smaller than a threshold value (usually $10^{-5}$) indicate significant linear dependencies in the basis set and a filtering of the corresponding eigenvectors might help to circumvent numerical difficulties if they arise during the SCF iteration procedure. Both orthogonalization schemes are implemented in QUICKSTEP. For small systems the choice of the orthogonalization has no crucial impact on the performance, since it has to be performed only once for each configuration during the initialization of the SCF run. By contrast, the eigenvectors and eigenvalues of the full Kohn-Sham matrix $K'$ have to be calculated in each iteration step as indicated by Eq. 2.96 using a divide-and-conquer (pdsyevd) scheme or an expert driver (pdsyevx) which allows to request only the build of an eigenvector sub-set. The divide-and-conquer scheme is faster than the expert driver if all eigenvectors have to be computed. However, for the construction of the new density matrix

$$P = 2 c_{\text{occ}} c_{\text{occ}}^T$$ \quad (2.100)

only the occupied orbitals are needed. In that case the expert driver is superior, since for standard basis sets only 10–20% of the orbitals are occupied.

The TD scheme is usually combined with methods to improve the convergence of the SCF iteration procedure. The most efficient SCF convergence acceleration is achieved by the direct inversion in the iterative sub-space (DIIS) \cite{Pul82, Pul80} exploiting the commutator relation

$$e = KPS - SPK$$ \quad (2.101)

between the Kohn-Sham and the density matrix where the error matrix $e$ is zero for the converged density. The TD/DIIS scheme is an established method for electronic structure calculations. The DIIS method can be very efficient in the number of iterations required to reach convergence starting from a sufficiently pre-converged density which is significant if the cost of constructing the Kohn-Sham matrix is larger than the cost of diagonalization. Nevertheless, the cost for the TD/DIIS scales as $O(M^3)$, where $M$ is the size of the basis set. This implies that, even at fixed system size, increasing the number of basis functions results in a cubic growth of the computational cost. A further disadvantage of the DIIS is that the method might fail to converge or that a sufficiently pre-converged density cannot be obtained. This happens more frequently for electronically difficult systems such as e.g. spin-polarized systems with a small energy gap between the highest occupied and the lowest unoccupied orbital.
2.9.2 Orbital transformation method

The OT method is a direct minimization method that addresses both deficiencies of the TD/DIIS scheme, as the method is guaranteed to converge, and scales, depending on the preconditioner, as \(O(MN^2)\). As described in detail in Ref. (VH03), the OT method parameterizes the MO coefficients \(c\) using new variables \(x\) as

\[
c(x) = c_0 \cos(U) + x U^{-1} \sin(U),
\]

where the matrix \(U = (x^T S x)^{1/2}\), \(x\) satisfies the linear constraint \(x^T S c_0 = 0\) and \(c_0\) are constant initial vectors that satisfy \(c_0^T S c_0 = I\). As the constraint on \(x\) is linear, standard optimizers such as DIIS or conjugate gradients can be employed. For a robust minimizer, e.g. using conjugate gradients in combination with line search, convergence is guaranteed, thus leading to a robust electronic structure method. The computational cost of the OT method is normally dominated by the computation of the \(O(MN)\) terms \(Hc\) and \(Sx\), but is in principle \(O(MN^2)\) with a sparse preconditioner, and \(O(M^2N)\) if a non-sparse preconditioner is used. In Ref. (VH03) it was shown how to compute the matrix functionals \(\cos(U), U^{-1} \sin(U)\) and the derivatives with respect to \(x\) based on diagonalization of the \(N \times N\) matrix \(x^T S x\). We have found that the performance of the \(N \times N\) diagonalization on a parallel computer can be such that an alternative method based on expanding the matrix functionals is beneficial. In this method, the matrix functionals are Taylor expanded as:

\[
\cos(U) = \sum_{i=0}^{K} \frac{(-1)^i}{(2i)!} (x^T S x)^i,
\]

\[
U^{-1} \sin(U) = \sum_{i=0}^{K} \frac{(-1)^i}{(2i + 1)!} (x^T S x)^i.
\]

The appropriate order \(K\) of the Taylor expansion is determined at every step by estimating an upper bound of the eigenvalue spectrum of \(x^T S x\) using the maximum absolute row sum norm. Note that each term is quadratic in \(x\), and that low order Taylor expansion (e.g. \(K = 2, 3\)) can be accurate to machine precision, in particular, if a good initial guess \(c_0\) can be generated. This is frequently the case during molecular dynamics simulations.
Table 2.1: Shown is the execution time in seconds of a single diagonalization using ScaLAPACK routines, and the time needed by the OT routines for one SCF iteration. The calculations were performed on 32 CPUs of an IBM SP4. A sample of 64 H2O molecules has been employed, and the basis sets DZVP, TZVP, TZV2P, QZV2P, and QZV3P result in a total of 1472, 1856, 2560, 2944, and 3648 basis functions, respectively.

Example terms needed for the derivative \( \frac{\partial E(c(x))}{\partial c} \) are

\[
\begin{align*}
\frac{\partial E(c(x))}{\partial c} \left( \frac{\partial (x^T S x)}{\partial x} \right)^1 &= S x(M), \\
\frac{\partial E(c(x))}{\partial c} \left( \frac{\partial (x^T S x)}{\partial x} \right)^2 &= S x(O(M) + (M)O), \\
\frac{\partial E(c(x))}{\partial c} \left( \frac{\partial (x^T S x)}{\partial x} \right)^3 &= S x(O(OM + MO) + (MO)O),
\end{align*}
\]

where \( O = x^T S x \) and \( M = (x^T H c) + (x^T H c)^T \), and terms are grouped to suggest an efficient way of evaluation.

Table 2.1 compares directly the computational cost of one step of the OT algorithm with the cost of solving the generalized eigenvalue equation for the \( N \) lowest eigenvectors using the ScaLAPACK routines \texttt{pdsyevx}, \texttt{pdsygst}, and \texttt{pdtrsm}. It can be observed that the OT algorithm is far superior.

## 2.10 Molecular dynamics

Molecular dynamics (AT87b; FS02) is a well founded method to study properties of materials at a finite temperature. The idea is to let the atoms, treated classically, evolve in a way that tries to closely match the microscopic evolution of the system. Molecular dynamics can be used to calculate equilibrium properties, and transport properties.

Similar studies can be performed with ab initio molecular dynamics in which explicit electronic structure calculations are employed to compute potential energies and forces.
A significant advantage of ab initio molecular dynamics is that no parametrisation of an empirical potential is needed and as such a wide range of systems can be simulated, even if unexpected chemical events take place. The length and time scales of typical ab initio molecular dynamics simulations are currently given by approximately 10 to 1000 atoms and 1 to 100 ps.

### 2.10.1 Algorithms

To let the system evolve and sample the phase space, a method to go from one point in phase space to the next is needed. To do this on a computer the equation of motion

\[ M_A \ddot{R}_A(t) = -\nabla_A E(\{R_A\}) = -\nabla_A \min_{\rho(r)} E(\{R_A\}, \rho(r)) \]  

(2.108)

where \( M_A \) and \( R_A \) are the atomic mass and atomic coordinates of the atom \( A \) respectively; have to be discretized.

The points of two trajectories even started at infinitesimally close points will diverge one from the other exponentially, so it is clear that for an approximate method it is exceedingly difficult to try to reproduce the exact trajectory starting at given point of the phase space. Fortunately in chemistry (unlike in astronomy) one normally is not interested in reproducing exactly a given trajectory, knowing that the approximate trajectory stays close to some real trajectory (unknown a priori) for long times compared to the one given by the Lyapunov exponent is enough, as one is interested in statistical averages that should not depend too much on exactly which trajectory is chosen.

If one wants to perform micro-canonical (NVE) simulations important properties for the integrator are the time reversibility (as the equations of motion are), a property that contributes to give longtime energy conservation, and simplicity (conservation of phase space volume), which guarantees a correct sampling. A common algorithm that has these properties is the Velocity Verlet algorithm(Ver67; SABW82). It is simple to implement and widely used, it is also the algorithm we use. It is defined as:

\[ v(t + \frac{\Delta t}{2}) = v(t) + \frac{1}{2m} F(r(t)) \]  

(2.109)

\[ r(t + \Delta t) = r(t) + \Delta t v(t + \frac{\Delta t}{2}) \]  

(2.110)

\[ v(t + \Delta t) = v(t + \frac{\Delta t}{2}) + \frac{1}{2m} F(r(t + \Delta t)) \]  

(2.111)
A problem of the simplectic integration algorithms is that they sample the microcanonical (NVE) ensemble. Experiments are normally performed in the canonical (NVT) ensemble. Having the temperature that is not explicitly fixed and known a priori can be inconvenient (as we will see later).

A simple but effective way to equilibrate the system around a given temperature is to re-scale the velocities to the target temperature each time the instantaneous temperature (kinetic energy) is further away than a given threshold from the target temperature. This method is not well suited to sample the NVT ensemble, but can equilibrate the system.

To perform the simulations in the NVT ensemble a thermostat has to be used, there are many possibilities, and no perfect solution. From the theoretical point of view the Nose Hoover chain thermostat (MKT82), is interesting because (for chains longer than 1) it can give the exact distribution for the harmonic oscillator and it does conserve an energy like quantity.

If one gives up the idea of complete determinism (because of random interaction with the environment) an interesting MD based sampling method is the Langevin dynamics. Langevin dynamics (TLF77; CR81; vGB82) introduces a Gaussian distributed random force that is offset by a dissipative friction term. The Langevin dynamics has the canonical (NVT) distribution as equilibrium distribution, and it can be very efficient at equilibrating a system. This technique can be used to describe effects deriving by a coarse grained description of a system, and indeed this is the reason this technique was developed. A problem of it is that momentum, rotational momentum,... are not conserved, and the random noise disrupts dynamical information (how much depends on the amount of noise, and thus on the friction coefficient).

2.10.2 Wave function extrapolation

One advantage of MD is that atomic configurations are generated in a continuous fashion and one can hence predict initial trial wave function for the SCF calculation by doing multi-linear extrapolation using the previous wave functions (APJ92). However, a given eigenfunction \( \psi_{i} \) can change rapidly if the two occupied states \( \psi_{i} \) and \( \psi_{j} \) are energy resonant. We therefore extrapolate the density matrix instead, more precisely its contra-covariant representation \( PS \). Since \( PS(t_{n}) \) is approximately a projector onto the
occupied subspace it can be multiplied with \( \{ \psi_i(t_{n-1}) \} \) to yield, after orthogonalization, a initial trial wave function \( \{ \psi_i(t_n) \} \) for time \( t_n \). Thus, our scheme provides directly an initial wave function, as required for the OT method and e.g. plane wave schemes, at variance with a similar scheme published recently (PF04). A simple formula for \( C(t_n) \) that does not require the explicit evaluation of \( PS \) is obtained by using Eq. 2.100:

\[
C(t_n) \approx \sum_{m=1}^{K} (-1)^{m+1} \binom{K}{m} C(t_{n-m}) C(t_{n-m})^T S(t_{n-m}) C(t_{n-1}).
\]  

(2.112)

This can be evaluated in \( O(MN^2) \) time. The effect of this extrapolation in terms of the number of iterations needed per MD step and on the error in the energy at the first SCF step is shown in Fig. 2.9. Our experience has shown that \( K = 3 \) produces a stable and efficient algorithm.

### 2.10.3 BO-MD

All these methods use electronic structure part as a “black-box” returning energies and forces for a given ionic configuration. We exploit this in CP2K by introducing sufficiently abstract interfaces to the electronic structure code (see section 2.6) so that any of the methods commonly used in molecular simulation such as e.g. path integral MD or MD simulations in the isothermal ensemble (NVT) are readily available.

This is possible because in the Born-Oppenheimer approximation (see section 2.1) introduces a clear separation between the ionic propagation and the electronic structure part, treating the atomic coordinates classically, and the electrons quantum-mechanically.

We notice that for BO MD, the time step used in the propagation is only dependent on the frequency spectrum of the atomic system. The time reversibility guarantees that no long term drift of the constant of motion, i.e. the sum of the atomic kinetic and potential energy, can be observed, provided that the forces are the exact derivatives of the potential energy (see paragraph 4.3.3 of Ref. (FS02)). As shown in section 2.4, the computed forces are exact derivatives of the energy only if \( \frac{\partial E}{\partial \Pi^{\mu\nu}} = 0 \). In any practical calculation, the wave function optimization is such that the above criterion is only approximately true, and hence the resulting drift in the constant of motion can be one criterion to judge the quality of the simulation. This is at variance with Car-Parrinello (CP85) simulations (see section 3.1.1), where the CP constant of motion (i.e. ionic kinetic energy + potential energy)}
Figure 2.8: Shown are the kinetic energy (top), and the constant of motion (bottom) for a liquid of 64 H$_2$O molecules at approximately 330 K during an NVE simulation. Simulation parameters are: a TZVP//DZVP basis set, a 200 Ry density cutoff, a time step of 0.5 fs, OT DIIS minimizer with $PS$ extrapolation ($K = 3$), and $\epsilon_{SCF} = 10^{-6}$. The measured drift is as low as $2.0 \cdot 10^{-7}$ a.u. per ps and per atom.

energy + fictitious kinetic energy) is exactly conserved (in the Verlet sense), but where the energy transfer from the ionic system to the fictitious system needs to be controlled.

Shown in Fig. 2.8 are the results of a simulation on 64 H$_2$O where the drift in the constant of motion is as low as $2.0 \cdot 10^{-7}$ a.u. per ps and per atom. However, we have found in liquid water simulations with the same convergence criterion, that the drift depends on the basis set and that $1.0 \cdot 10^{-6}$ a.u. per ps and per atom can be expected for $\epsilon_{SCF} = 10^{-6}$.

As one application of MD simulation we have computed the hydrogen velocity-velocity auto-correlation function, and the corresponding power spectrum in Fig. 2.10 for the system described in the caption of Fig. 2.8. These results can be directly compared to the experimental vibrational spectrum of liquid H$_2$O. The experimental results for the bending and the two OH stretch modes are 1645, 3280 and 3490 cm$^{-1}$, and this compares favorably with the well defined peak at 1640 and the broad peak at 3330 cm$^{-1}$ found in our simulations.
Figure 2.9: Upper panel: Average number of iterations needed to achieve $\epsilon_{\text{SCF}} = 10^{-6}$ (squares) and convergence ($\epsilon_{\text{SCF}} = 10^{-7}$, diamonds) for an MD simulation of 64 H$_2$O at 300 K using a TZV2P basis set. Lower panel: Error in the total energy per atom of the initial wave function.
Figure 2.10: Upper panel: Hydrogen velocity-velocity auto-correlation function. Lower panel: Fourier transform of the auto-correlation function. The system is a liquid of 64 H$_2$O molecules at approximately 330 K during an NVE simulation. Simulation parameters are: a TZVP/DZVP basis set, a 200 Ry density cutoff, a time step of 0.5 fs, OT DIIS minimizer with PS extrapolation ($K = 3$), and $\epsilon_{\text{SCF}} = 10^{-6}$. 
2.11 Accuracy

To assess the accuracy of the code Matthias Krack compared the geometries for 39 first and second row small molecules with those calculated with NUMOL (DB93), as described in section 2.8, finding an excellent agreement (VKM+05).

For condensed phase systems the situation is more complex because standard benchmarks to judge the quality of an electronic structure method for the liquid phase do not yet exist. Therefore, in order to assess the accuracy of QUICKSTEP for calculations in the condensed phase, comparisons with the program CPMD (CPM) have been made by Joost VandeVondele. CPMD employs a plane wave basis set to represent the wave functions which is a natural basis set for periodic condensed phase systems. Furthermore, plane waves provide an orthogonal basis that allows to reach conveniently the basis set limit for a given pseudo potential only by increasing a single parameter, i.e. the plane wave cut-off. In principle, identical total energies can be obtained by QUICKSTEP and CPMD, if the same system is described by the same pseudo potentials. Nevertheless, this is a non-trivial and very stringent test for the accuracy of both methods, since the two methodologies and basis sets are very different. We have already mentioned that the GTH pseudo potentials are rather hard (see section 2.4.2) and that they require a higher plane wave cut-off than the pseudo potentials of Troullier and Martins (TM91) that are commonly used in CPMD calculations.

Figure 2.11 shows a correlation plot between forces computed with QUICKSTEP and CPMD for the same configuration of a liquid water sample with 32 water molecules in the simulation cell using an LDA density functional (GTH96). Five different methods are being compared. CPMD with a plane wave cut-off of 300 Ry and 100 Ry implying a density cut-off of 1200 Ry and 400 Ry, respectively. QUICKSTEP with a large basis (uc, uncontracted with high angular momentum terms, 254 basis functions per water molecule) and a density cut-off of 1200 Ry, and a DZVP basis set (23 basis functions per water molecule) and a density cut-off of 1200 Ry or 300 Ry. Panel A shows that perfect agreement between CPMD and QUICKSTEP can be achieved. The root mean square deviation (RMSD) between the CPMD and the QUICKSTEP forces is below $6 \cdot 10^{-5}$ a.u.. The panels B, C, and D, with RMSDs of $2 \cdot 10^{-3}$, $2 \cdot 10^{-4}$, and $7 \cdot 10^{-4}$ a.u., show that the quality of the forces as obtained with QUICKSTEP compares favorably with a typical plane waves calculation.
2.11 Accuracy

Figure 2.11: The four panels show the correlation between all forces $F_x$, $F_y$, and $F_z$ calculated for the same configuration of a liquid water sample containing 32 water molecules. Panel A shows that perfect agreement can be obtained between QUICKSTEP (QS) and CPMD using large basis sets (see text) for both methods; Panel B compares these converged results to CPMD results using slightly more typical settings; Panel C shows the effect of a smaller Gaussian basis set; Panel D additionally shows the effect of a smaller density cut-off.
Furthermore, in (VKM+05) we compared also the accuracy of the relative energies obtained from the different methods by computing a potential energy surface of 25 configurations of 64 water molecules sampled during an MD run, using the BLYP density functional (Bec88a; LYP88; MSSP89) finding that the reference potential energy surface itself fluctuates around zero with a Root Mean Square Deviation (RMSD) of $9.8 \cdot 10^{-3}$ a.u. and a Maximum Absolute Difference (MAD) of $18.6 \cdot 10^{-3}$ a.u.

These results show that good agreement can be obtained, but that both a sufficiently good basis set and a sufficiently high density cut-off are needed to reach an accuracy that is e.g. similar to a 100 Ry CPMD calculation. We have observed that convergence with respect to the density cut-off is significantly slower for BLYP calculations than for LDA calculations (see also section 2.4.4)

### 2.11.1 BSSE in liquid water

The Basis Set Superposition Error (BSSE), as explained section 2.11.1 is an error due to a non uniform variational flexibility of the basis which gives an unphysical lowering of the energy to the configurations in which the exact solution can be better approximated. This is unavoidable if a localized basis is employed.

To reduce its influence one can use large basis sets. In particular with Gaussian, that do not describe the tail region of the density correctly, as they decay too fast, the use of diffuse function to better describe the tail part reduces very much the error. Unfortunately diffuse functions increase the computational cost a lot, as they reduce the sparsity of $S$, and increase its condition number by introducing almost linearly dependent components in the basis. This has negative effects on the forces and is not practicable for the MD of condensed phase systems.

Another method to reduce the BSSE is the counterpoise method (BB70). It separates the systems in two fragments and assumes that the the lowering in energy in the combined system due to the BSSE is the same as the lowering in energy in the two fragments calculated in presence of the basis set of the other fragment. This method works very well in practice, but it gives only the bounding energy of two fragments, and the attempts at generalizing it to more than two fragments have not been very successful so far.

This means that one cannot easily get rid of the BSSE in condensed phase simulations
2.11 Accuracy

Figure 2.12: Computed errors in the binding energy of a single water molecule in the liquid as calculated with DZVP, TZV2P, QZV2P, QZV3P basis sets (circles, squares, triangles and crosses) relative to a reference binding energy that is a counterpoise corrected calculation using a QZV3P basis with added d and f functions (see text for details). The fluctuations around the constant shift have standard deviations of respectively 0.417, 0.157, 0.085, 0.079 kcal/mol.

performed with QUICKSTEP, however, the fact that we were able to reproduce the forces calculated with the CPMD program in the case of liquid water seem to indicates that BSSE is mostly a shift of the potential energy surface. To check this intuition and characterize the shift one can look at the binding energy of a single water molecule in a liquid sample containing 32 molecules.

$$E^{\text{binding}} = E(32H_2O) - E(1H_2O) - E(31H_2O)$$

(2.113)

which is sensitive to a proper description of both gas and liquid phase molecules, and might thus be expected to be sensitive to the BSSE error. The BSSE error is defined here approximately for a given basis as the difference in $E^{\text{binding}}$ computed using the given
basis set and a reference value obtained with a counterpoise corrected calculation using a QZV3P basis set augmented with diffuse functions and with extra d and f orbitals, which should be close to the basis set limit value of $E^{binding}$. We have computed this error for a large number of water molecules in liquid samples that were obtained from a MD simulation at around 320 K using the OLYP functional (VMK+04) and the results are shown in Fig. 2.12. As anticipated, a significant shift of this binding energy can be observed, but the error is to a very good extend independent of the configuration. This is consistent with the quality of the potential energy surface observed previously and provides an explanation for the observed rapid convergence with respect to basis set of structural and dynamical properties of liquid water (VMK+04).

2.12 Benchmarks

In this section, we show that the accuracy of QUICKSTEP can be achieved with high computational efficiency. We illustrate both the serial performance and the scalability on parallel computers using a high-end supercomputer as well as a modern cluster based on a PC-like architecture.

2.12.1 Liquid water

Liquid water is a convenient benchmark system since it can easily be scaled by doubling the number of water molecules in the unit cell, followed by classical equilibration to yield a system without additional symmetries. It is a standard benchmark system for the CPMD code (CPM) to check performance and scalability on various parallel computers. MD runs for pure liquid water at ambient conditions (300 K, 1 bar) have been conducted using input parameters as appropriate for quality production runs, i.e. GTH pseudo potentials, TZV2P basis sets for hydrogen and oxygen, a density cut-off of 280 Ry for the expansion of the electronic density, a threshold of $10^{-12}$ a.u. for the overlap integral between two primitive Gaussian functions, and the total energy of the system was converged to $10^{-7}$ a.u. at every MD time step (0.5 fs).

Table 2.2 lists the characteristics of the benchmark systems that range in size from 32 to 1024 water molecules, the largest system being several nanometres in all dimensions.
Table 2.2: Detailed characteristics of the employed benchmark systems for liquid water at ambient conditions (300 K, 1 bar). The edge length of the cubic simulation cell, the number of atoms, electrons, Gaussian-type orbitals ($M$), occupied orbitals ($N$), and plane waves, i.e. grid points, is listed.

<table>
<thead>
<tr>
<th>system</th>
<th>cell [Å]</th>
<th>atoms</th>
<th>electrons</th>
<th>$M$</th>
<th>$N$</th>
<th>grid points ($\times 10^6$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 H$_2$O</td>
<td>9.9</td>
<td>96</td>
<td>256</td>
<td>1280</td>
<td>128</td>
<td>1.3</td>
</tr>
<tr>
<td>64 H$_2$O</td>
<td>12.4</td>
<td>192</td>
<td>512</td>
<td>2560</td>
<td>256</td>
<td>2.0</td>
</tr>
<tr>
<td>128 H$_2$O</td>
<td>15.6</td>
<td>384</td>
<td>1024</td>
<td>5120</td>
<td>512</td>
<td>4.1</td>
</tr>
<tr>
<td>256 H$_2$O</td>
<td>19.7</td>
<td>768</td>
<td>2048</td>
<td>10240</td>
<td>1024</td>
<td>9.3</td>
</tr>
<tr>
<td>512 H$_2$O</td>
<td>24.9</td>
<td>1536</td>
<td>4096</td>
<td>20480</td>
<td>2048</td>
<td>16.0</td>
</tr>
<tr>
<td>1024 H$_2$O</td>
<td>31.3</td>
<td>3072</td>
<td>8192</td>
<td>40960</td>
<td>4096</td>
<td>32.8</td>
</tr>
</tbody>
</table>

Table 2.3: Occupation of the overlap matrix applying a threshold of $10^{-12}$ for the overlap contribution of two primitive Gaussian orbital functions.

<table>
<thead>
<tr>
<th>system</th>
<th>occupation</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 H$_2$O</td>
<td>100.0 %</td>
</tr>
<tr>
<td>64 H$_2$O</td>
<td>99.6 %</td>
</tr>
<tr>
<td>128 H$_2$O</td>
<td>85.1 %</td>
</tr>
<tr>
<td>256 H$_2$O</td>
<td>51.3 %</td>
</tr>
<tr>
<td>512 H$_2$O</td>
<td>25.8 %</td>
</tr>
<tr>
<td>1024 H$_2$O</td>
<td>12.9 %</td>
</tr>
</tbody>
</table>

The number of contracted Gaussian-type orbital basis functions is growing linearly from 1280 to 40960 functions, and $32.8 \times 10^6$ plane waves are required for the auxiliary basis of the largest system. Since matrices like the overlap and Kohn-Sham matrix grow quadratically with system size, it is indispensable to take advantage of the sparsity that is emerging with increasing system size. The fraction of non-zero blocks of the overlap matrix, for a threshold of $10^{-12}$, is shown in Table 2.3. For the smaller systems, all molecules in the unit cell interact with each other, which results in a dense overlap matrix, whereas starting from roughly 200 water molecules, the interaction sphere is completely confined in the united cell. Consequently, the number of non-zero elements grows only linearly with system size for systems larger than roughly 256 H$_2$O. As described previously, this matrix sparsity is exploited in QUICKSTEP. A further observation that can be made is that for the employed TZV2P basis the number of occupied orbitals $N$ is only 10% of total number.
of orbitals $M$ (see Table 2.2). Operations that deal only with the occupied orbitals ($MN$) require significantly less memory and time than similar operations on full matrices ($M^2$). It is therefore possible to test larger systems with the orbital transformation method (OT) than with the traditional diagonalization (TD) method.

As a measure of the performance we have reported the average time needed per MD step for the benchmark systems of Table 2.2 using both traditional diagonalization and the orbital transformation method. This time covers all aspects of the calculation, including the geometry dependent initializations, SCF iterations and force calculation. Results are reported in Figure 2.13 using an IBM Regatta p690+ system with 32 Power4+ (1.7 GHz) CPUs per node, 3.6 GB RAM per processor, and interconnected by an IBM High Performance Switch (HPS), and in Figure 2.14 for a PC cluster with dual AMD Athlon MP2200+ CPUs and a fast interconnect (SCI). A double logarithmic scale is employed to show the parallel scaling of the methods. From these results, we can conclude that the TD scheme allows for efficient simulation of small systems (32–64 H$_2$O) on a relatively small number of CPUs, and that medium sized systems (128–256 H$_2$O) run with fair efficiency on 32–64 CPUs, whereas large systems cannot run with the given memory constraints. The OT method provides significantly improved scaling for all systems, is up to 6 times faster, and the reduced memory requirements allow simulation of up to 1024 water molecules. With the current implementation, at least one or two H$_2$O molecules (40–80 basis functions) per CPU are needed to maintain efficiency. Whereas short relaxations are the limit for the larger systems, 10 ps of dynamics of a system containing 64 water molecules can be generated in roughly four days using one Regatta node.
Figure 2.13: Scaling of the CPU time per MD step using the traditional diagonalization (TD) scheme and the orbital transformation (OT) method for the benchmarks systems of Table 2.2. The calculations were performed on an IBM Regatta p690+ system with 32 Power4+ (1.7 GHz) per node interconnected by an IBM High Performance Switch (HPS). The dotted lines represent ideal scaling.
Figure 2.14: Scaling of the CPU time per MD step using the traditional diagonalization (TD) scheme and the orbital transformation (OT) method for the benchmarks systems of Table 2.2. The calculations were performed on a PC-Cluster with dual AMD Athlon MP2200+) using a SCI interconnect. The dotted lines represent ideal scaling.
Chapter 3

Ab-initio Liquid Water

Pure liquid water is a rather uniform and simple system (from the chemical point of view). It is still very interesting not only because of its ubiquity, and importance for life, but also for its anomalous properties. The empirical force fields for water have become more and more accurate over the years, but no one can yet yield a quantitative description of the thermodynamic properties of water over the whole liquid range. Indeed the large polarizability and strong dipole and its chemical activity, particularly its self-dissociation, pose a challenge to empirical potentials. ab-initio description of water can describe both the physical and chemical properties, but can only explore much shorter trajectories and smaller systems. Thus due to the challenges represented by this system there are still many open questions, and active research.

The first attempt at ab-initio molecular dynamics description of water used the Car-Parrinello method (LSPC93). The CP2K program, as described previously performs Born-Oppenheimer molecular dynamics runs, and can perform also Monte Carlo sampling, and, as will be presented in this chapter, was used to perform several simulations of liquid water.

A comparison of various sampling techniques (CP-MD, Monte Carlo and BO-MD) applied to liquid water to which I participated, and that I briefly review in 3.1 spurred us to evaluate the temperature and functional dependence in liquid water. Indeed as liquid water is very sensitive to temperature, imperfectly determined temperature can contribute to explain the spread of values of the peak Oxigen-Oxigen pair correlation functions found in the literature. The functional dependence as shown in section 3.2 was troubling
large, and shows that unfortunately DFT is not yet a “black-box” method, and that the
transferability and approximations involved in DFT methods, as with other methods,
need to be checked before accepting the results. We show that the self diffusion constant
of water cannot yet be determined in a dependable way from ab-initio simulations, and
is normally severely underestimated.

3.1 Investigation of Different Sampling Approaches

Recently Asthagiri et al. (APK03) and Grossman et al. (GSD+04) questioned the repro¬
ducibility of previous ab-initio simulations. For this reason we wanted to assess the
reproducibility of first principles simulations of liquid water using different approaches
to sample the trajectories (e.g., different electronic structure methods, different fictitious
electron masses, and different statistical mechanical ensembles), but maintaining the other
simulation parameters (density functional and, to the extent possible, temperature).

3.1.1 Car-Parrinello Method

The Car-Parrinello method (CP85) uses an extended Lagrangian formalism. In it the
electronic degrees of freedom $\Psi_i$ are minimized at the beginning, and then evolve together
with the atomic degrees of freedom. The electrons are given a fictitious mass $\mu$ and then
evolve according to the Lagrangian

$$
\mathcal{L} = \frac{1}{2} \sum_i (\dot{\psi}_i^* \dot{\psi}_i) + \sum_I M_I \dot{v}_I \cdot v_I - E[\psi_i, \mathbf{R}_I] + \sum_{ij} (\delta_{ij} - \langle \psi_i | \psi_j \rangle)$$

This evolution can be implemented easily if the basis set is orthogonal and does not depend
explicitly on the atomic positions, because then the forces on the electronic degrees of
freedom don’t contain terms depending on the change of the basis set. For this reason Car-
Parrinello methods normally use plane waves. Plane waves can be handled very efficiently,
as they are uniform, there is very optimized code that handles FFT transformations, so
that operations can be done either in real or reciprocal space depending on where they
are more efficient. The Car-Parrinello procedure works if there is adiabatic separation
between electronic and ionic degrees of freedom (i.e. if their characteristic frequencies are
sufficiently separated) (MH00). This can be achieved by choosing the fictitious mass of
the electrons small enough.
The detailed effects on how the fictitious mass influences the results, is not yet fully understood. The basic argument that explains why the Car-Parrinello approach works is that if there is a gap $E_{\text{gap}}$ between the occupied and unoccupied orbitals within the harmonic approximation the lowest frequency of oscillations of the orbital is

$$\omega = \sqrt{\frac{2E_{\text{gap}}}{\mu}}$$

(3.2)

and that in classical mechanics systems well separated in frequency don’t interact (PSB91; MH00). Tangney and Scandolo (PT02) argue that the change in the electronic spectrum due to the ionic movement changes this simple picture. In the rigid ion approximation they propose that to get a better agreement of the results the masses of the ions are augmented of a term proportional to the kinetic energy of the electrons of the atom. This would mean that the effective temperature of the CP-simulation would be higher than the one calculated with the kinetic energy of the ions. Tangney recently explored more in depth the dependence from the electronic mass $\mu$ (Tan06).

What can be said is that all these effects should disappear when the electronic mass goes to 0. Empirically what happens is that when the simulation is started the electrons have no kinetic energy, rapidly they acquire a small kinetic energy (the electrons need to move to stay close to the minimum), and then a constant drift towards higher temperatures sets in. This drift is given by the imperfect adiabatic separation between electronic and ionic degrees of freedom. If the drift is small one expects that the electrons sample a potential energy surface very close to the Born Oppenheimer surface. Even if the drift is small enough and one does not expect to see a big effect from such a small energy flux, doing long simulations it accumulates: the temperature of the electrons will continue to rise (and the one of the ions to diminish) until they equilibrate. Obviously if the electronic temperature is not small (with. to the gap) one can not anymore expect to be sampling the BO potential surface. To compensate the drift one can periodically quench the electrons (minimizing the wave functions), but this introduces a discontinuity in the trajectory of the electronic degrees of freedom. Another option is to use thermostats to compensate the drift. This works quite well, but it should be clear that the thermostat compensates just the effect of the accumulation of the energy flux, not the presence of the energy flux. If the flux is too big there will be artifacts in the simulation results, and if the electrons cannot follow the minimum hysteresis and thermostat dependent effects will arise.
3.1.2 Monte Carlo

Instead of trying to reproduce or approximate the microscopic dynamics of a system, one can try to directly sample the phase space with the correct probability. This replaces time averages with ensemble averages. If the system is ergodic this two approaches give the same result.

The advantage of doing an ensemble average is that any kind of moves (as long as they conserve the detailed balance) are allowed, also unphysical ones that, if correctly chosen, can equilibrate the system faster. Furthermore in a classical framework (that is valid for the ions) the contribution of the kinetic part can be calculated analytically, and one can sample just the configuration space. Another advantage of the Monte Carlo-Metropolis algorithm is that only the energy (and not the forces) of the configuration is needed. The drawback of the Monte Carlo method is that you have no dynamical informations, and that no extrapolation can be used to get the first guess of the wave-functions. As the ab-initio energy evaluation is expensive it pays off to try complicated moves.

In CP2K/QUICKSTEP, as the trial moves with the QM potential are very costly they are biased by performing trial trajectories (ISWS00; Gel03) with an empirical potential calculated with a spline interpolation of the QM potential itself(IPBV04). This algorithm was implemented by Matthew J. McGrath under the direction of J. Ilja Siepmann in cp2k, and uses for the ab-initio code Quickstep that I co-developed to evaluate the quantum energy.

3.1.3 Article

On this topic we (I.-F. W. Kuo, C. J. Mundy, M. J. McGrath, J. I. Siepmann, J. VandeVondele, M. Sprik, J. Hutter, B. Chen, M. L. Klein, F. Mohamed, M. Krack and M. Parrinello) published an article (KMM+04), for which I participated actively at the long discussion on the results and tests to do.

In this we perform ten simulations of 64 water molecules using the BLYP functional at a temperature around 315 K using various sampling techniques. The BLYP functional was widely used in the literature because it was thought to give a good description of water.
The results show that it is possible to reproduce the same results with different sampling techniques if enough care is taken. They also confirm that a fictitious mass of 800 au for micro-canonical simulations of tens of picoseconds leads to an unacceptably large drift of the electronic kinetic energy.

The simulation produce an Oxygen-Oxygen radial distribution function with a peak of about 3.0, and an underestimated self-diffusion constant of about 0.04 Å²/ps (experimentally $D_{\text{self}} = 0.35 \text{Å}^2/\text{ps}$ at T=318K). The time scale accessible through ab-initio simulations (tens of ps) is much too short to expect a convergence of the diffusion constant, a molecule does not have the time to diffuse much more than the width of itself. Furthermore calculations by Kohlenmeyer (Koh04) on classical polarizable models show that it can be extremely difficult to converge the diffusion constant. Even with this caution, it is possible to say that the BLYP model is, consistently with recent findings, sluggish-diffusive and over-structured.

Although it was possible to reach similar results with different sampling techniques, starting with exactly the same configuration, looking at the results in the literature it is clear that there is quite some spread of results, even within the same technique. This hints that there is some source of error that is not well understood. Thus came the idea of looking at the variability within a single technique and to try to better understand the sources of errors.

3.2 Temperature and functional dependence in water

After having seen that different sampling techniques could give the same result, we explored how sensitive the result were to changes in temperature and functional. For if it was possible to get the same results we saw that much care was needed to get the same result even within the same technique.

Part of this sections are taken from an article that we (Joost VandeVondele, Fawzi Mohamed, Matthias Krack, Jürg Hutter, Michiel Sprik, Michele Parrinello) published on the Journal of chemical physics 122, 014515 (2005). I have participated actively at the discussion on this article, I have implemented part of the functionals, and generally contributed to the program and article.
3.3 Introduction

Liquid water is perhaps the most popular system in computational chemistry in terms of the effort spent on modeling its properties. Numerous force fields have been designed ranging from empirical models with parameters fitted to experimental data to ab initio models which have been parameterized using the results of quantum chemistry calculations of clusters in vacuum. The modeling of water has assumed a similar stimulating role in the development of ab initio molecular dynamics methods for aqueous systems. Since the first Car-Parrinello simulations of bulk liquid water about ten years ago (LSPC93), a succession of studies has appeared (SHP96; SP99; VHK01; IV02; APK03; CIKP03; GSD04; KMM04) using a variety of density functionals and molecular dynamics schemes. Repeating the history of classical simulation, the results of these ab initio simulations can show significant disagreement with experiment and among each other. Obviously, differences between current density functionals are an important source of discrepancy. However, the reproducibility and accuracy of the ab initio molecular dynamics methodology have also been questioned (APK03; GSD04). This was prompted by the observation that, even if the same density functional is used, a noticeable degree of variability in the structural (the pair correlation function) and dynamical (the diffusion constant) properties remains. This suggests that these discrepancies should be contributed to more technical factors such as system size, duration of the run and differences in implementation of the molecular dynamics (MD) method, that include e.g. size and type of the basis set, pseudo potential, convergence criterium for the electronic structure calculation (Born-Oppenheimer MD) or fictitious mass (Car-Parrinello MD), time step, integrator and ensemble. Indeed, in Ref. (GSD04) it was shown that a proper choice of fictitious mass is vital for simulations of liquid water.

As ab initio simulations move towards more complex systems, this source of error needs to be understood. With a 300-fold increase of supercomputer power since the original publications (A.A04), a new generation of more systematic studies has become feasible, and we should be in a better position to resolve these issues. Thus, in a recent comprehensive study we have addressed the question of the convergence of the adiabatic force calculation. Results produced by the Car-Parrinello method were compared to results obtained by Born-Oppenheimer MD and Monte Carlo simulations (KMM04). In the latter two approaches the electronic structure is recomputed every time step by iterative electronic minimization, in contrast to the Car-Parrinello method which applies
3.4 Validation of methods and Gaussian basis sets in the condensed phase

3.4.1 Electronic structure computation and molecular dynamics

The simulations presented here have been performed using the methods described in chapter 2. The use of the hybrid Gaussian and plane wave (GPW) scheme explained in
section 2.4 that combines a Gaussian basis for the wave functions with an auxiliary plane wave basis set for the density as implemented in the CP2K/QUICKSTEP program (LHP97; CP204; VKM’05), and its linear scaling the Kohn-Sham matrix construction maintaining the accuracy of conventional schemes were important advantages. We used the pseudo potentials of the Goedecker-Teter-Hutter (GTH) type, described in section 2.4.2, with its parameters determined initially for the local density approximation (GTH96; HGH98) and adjusted for the various density functionals employed. Elements of the Kohn-Sham and overlap matrix smaller than $\epsilon_{\text{default}} \approx 10^{-12} - 10^{-14}$ were neglected, and a 280Ry density grid has been employed.

The majority of DFT based plane wave molecular dynamics studies of water have employed the Car-Parrinello method. Born-Oppenheimer (BO) MD, as in the present work has been rare (See for example Ref. (VHK+01)).

For the wave function optimization the orbital transform method (VH03) described in section 2.9.2 was employed together with the wave function extrapolation method discussed in 2.10.2 that greatly enhances the efficiency of the BO molecular dynamics. The nuclear equations of motion have been integrated using a standard velocity Verlet algorithm with a 0.5 fs time step and Hydrogen masses.

The model system consisted of 32 molecules in a cubic simulation cell with edges of 9.8528 Å under periodic boundary conditions. After equilibrating using classical force field methods, the initial configuration was equilibrated for at least 15ps more using the BLYP(Bec88a; LYP88) density functional, prior to use in the production runs. The final runs, of which the first few ps have been discarded for computation of properties, are all continuous NVE trajectories, i.e. without intermediate rescaling of the ionic velocities. We enforce a sufficiently strict convergence upon the wave functions (convergence criterium for the electronic gradient $\epsilon_{\text{SCF}} \approx 10^{-6} - 10^{-7}$, energy difference between final self-consistent field (SCF) cycles $10^{-10} - 10^{-12}$ a.u. per atom), so that the drift in the total energy during these simulations is approximately $10^{-6}$ a.u. / ps / atom. For the water systems considered, this is equivalent to a temperature change of approximately 1K over a 10ps simulation, which suggests that the correct NVE ensemble is sampled.

The mean square displacement (MSD) curves were computed from the relative displacements of the Oxygen atoms only, averaged over all water molecules and all frames of the full trajectory as starting points. Frames of the trajectory have been stored every
The slope of the curves has been estimated by a linear fit to the data, starting from about 2ps to about half the length of the trajectory. This avoids the final part where the average is only over few uncorrelated frames, and the initial rapidly increasing part that is due to the motion of the molecules in their solvation cages. Nevertheless, the statistical uncertainty might still be relatively large, in particular if the system is not diffusive. The pair correlation functions \( g(r) \) have been computed using a binning width of 0.03Å, as suggested in Ref. (IV02), and are not smoothed.

### 3.4.2 Basis set convergence

In the GPW method, as in other localized basis set calculations, the quality of the Gaussian basis set is an important issue. In fact, there is little information about the performance of a Gaussian basis set in the context of molecular dynamics simulations of liquids. It is well known that the basis set superposition error (BSSE) can be a significant fraction of the binding energy of the water dimer, which could affect liquid simulations. (SBS01) However, the effect of BSSE on structural and dynamical properties of the liquid has not been investigated previously. From the results of section 2.11.1 we do not expect a big influence, but in order to quantify the influence of the basis set on the pair correlation functions and the diffusivity of the liquid, we have explicitly performed simulations of 32 water molecules using six different basis sets (DZVP, TZVP, TZV2P, QZV2P, OPT-TZVP, OPT-TZV2P) with the OLYP(HC02) functional.

The results are summarized in Table 3.1. The DZVP, TZVP, TZV2P and QZV2P basis sets have been constructed with a split valence method (VKM+05), whereas in the OPT-TZVP and OPT-TZV2P basis sets the contractions of both valence and polarization functions (over six Gaussian exponents) have been optimized explicitly to yield the lowest total energy for a liquid configuration, while retaining a low condition number of the overlap matrix. These optimized basis sets are computationally more demanding, and the corresponding trajectories are somewhat shorter (see Table 3.1). We have also computed two trajectories with a lower convergence (LC) of the wave function at every BO step.

The data reported in Table 3.1 show that MSDs and the maximum of the pair correlation function are in good agreement with each other, for all the methods. The small deviations are well within the statistical uncertainty (see also below), except perhaps the DZVP basis which appears to yield a slightly more diffusive liquid. The insensitivity of
3 Ab-initio Liquid Water

<table>
<thead>
<tr>
<th>Basis</th>
<th>BF</th>
<th>Time [ps]</th>
<th>T [K]</th>
<th>Drift</th>
<th>MSD [$\text{Å}^2\text{ps}^{-1}$]</th>
<th>$g(r)_{\text{QQ}}^{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DZVP</td>
<td>23</td>
<td>22.2</td>
<td>300</td>
<td>0.2</td>
<td>1.35</td>
<td>2.16</td>
</tr>
<tr>
<td>TZVP</td>
<td>29</td>
<td>18.7</td>
<td>311</td>
<td>0.6</td>
<td>0.78</td>
<td>2.33</td>
</tr>
<tr>
<td>TZV2P</td>
<td>40</td>
<td>21.2</td>
<td>311</td>
<td>0.8</td>
<td>0.77</td>
<td>2.27</td>
</tr>
<tr>
<td>QZV2P</td>
<td>46</td>
<td>16.1</td>
<td>306</td>
<td>0.5</td>
<td>0.78</td>
<td>2.19</td>
</tr>
<tr>
<td>OPT-TZVP</td>
<td>29</td>
<td>12.8</td>
<td>315</td>
<td>0.3</td>
<td>0.67</td>
<td>2.28</td>
</tr>
<tr>
<td>OPT-TZV2P</td>
<td>40</td>
<td>12.5</td>
<td>305</td>
<td>0.6</td>
<td>0.88</td>
<td>2.07</td>
</tr>
<tr>
<td>TZV2P-LC</td>
<td>40</td>
<td>24.6</td>
<td>308</td>
<td>-1.5</td>
<td>0.63</td>
<td>2.25</td>
</tr>
<tr>
<td>OPT-TZV2P-LC</td>
<td>40</td>
<td>21.4</td>
<td>306</td>
<td>-2.4</td>
<td>1.07</td>
<td>2.16</td>
</tr>
</tbody>
</table>

Table 3.1: Characteristics of an OLYP simulation (Gaussian basis functions per molecule (BF), time length, temperature, energy drift in a.u. $\times 10^{-6}/(\text{ps atom})$) and computed properties (Slope of the mean square displacement curves (MSD), maximum of the Oxygen-Oxygen pair correlation function $g(r)$) for six different basis sets with the same strict convergence criterion ($\epsilon_{\text{SCF}} = 10^{-7}$) and two simulations with a less strict convergence criterion ($\epsilon_{\text{SCF}} = 10^{-6}$, last two lines with the LC post-fix.).

The results is rather remarkable and deserves further study. In the remainder of this study we have employed TZV2P and QZV2P basis sets, with 1280 and 1472 basis functions for 32 water molecules respectively, for approximately $10^7$ SCF cycles.

3.5 Temperature dependence of properties of BLYP water

In order to investigate the effect of temperature on BLYP water we have set up 16 systems with different total energy, keeping all other parameters fixed. The total energy was chosen to yield final average temperatures in the range 280K-380K. The first 8 trajectories have been started from the same 300K equilibrium configuration (with different velocities), whereas the later 8 trajectories were initiated from one of the resulting configurations that equilibrated near 330K. All 16 simulations are approximately 20ps long. A TZV2P basis has been employed, and $\epsilon_{\text{SCF}} = 10^{-6}$ resulting in an energy drift in the order of $10^{-6}$ a.u./ps/atom. As shown if Fig. 3.1 the relationship between the total energy and
Figure 3.1: Relationship between total energy and average temperature for the sixteen NVE simulations of 32 water molecules as described in detail in the text. The solid line is a linear fit through the data.
the temperature of these runs is effectively linear. The maximum fluctuations around the expected value (fitted line) are on the order of 10K which can be considered the statistical uncertainty in the temperature for runs of 32 water molecules and 20ps duration.

The resulting Oxygen-Oxygen pair correlation functions are shown in Fig. 3.3. The change in shape of the curves clearly reflects the strong effect of temperature on the structure of water. The temperature dependence of the features of the curve, such as heights and positions of minima and maxima, is in agreement with the expected smoothing of the curves with increasing temperature.

In Fig. 3.2 this is quantified for the maximum of the first Oxygen-Oxygen peak. A wide range for the maximum can be observed (2.5 - 3.6), and we note that the scatter in the results is not small.

The MSDs obtained from these simulations, shown in Fig. 3.4, exhibit an even more pronounced sensitivity to temperature. The MSDs lie in the range 1.0 - 0.001 Å^2 ps^{-1}, with a roughly exponential dependency on the inverse temperature. Furthermore, it is apparent from the outliers that these BLYP simulations suffer from non-ergodic behavior on the 20ps time scale of the simulation at temperatures below approximately 325K.
Figure 3.3: Shown are five out sixteen Oxygen-Oxygen pair correlation functions, corresponding to simulations with temperatures of 292K, 318K, 329K, 351K, and 372K. The ordering of the curves follows the expected behavior, i.e. minima and maxima become less pronounced and positions of the first peak shift outwards for increasing temperatures. This is further quantified in Fig. 3.2. The dashed lines are the experimental results from Ref. (HSGHG00) and Ref. (Sop00).
Figure 3.4: Slopes from the MSD curves obtained for the sixteen water simulations described in the text. The grey line is an Arrhenius fit to data points > 325K. The activation energy obtained in this way is 21kJ/mole with an uncertainty of about 5kJ/mole. Significant deviations from the extrapolation can be observed in the lower temperature range. These deviations are indicative of non-ergodicity at lower temperatures.
3.6 Performance of six different density functionals

We note that this temperature might depend on simulation parameters such as system size and density functional. In this regime, configurations can remain in non-diffusive or more liquid-like state for long times. These low temperature results can therefore not be converged within the 20ps time scale and are very sensitive to the initial conditions, even if the simulation setup is identical. This makes simulations at these conditions very sensitive to all simulation details, including e.g. the equilibration or thermostating method. This behavior, combined with the difficulty of estimating the temperature of short runs, can be part of the explanation for the inaccuracies in results reported in literature. We note that also experimentally the diffusion and observed peak heights are strongly dependent on the temperature, especially if the meta-stable super-cooled water is considered as well. (GDH72; HRG\textsuperscript{+03}) Note that our simulations have been performed at constant volume, whereas these experiments are at constant pressure.

System dimensions are another set of critical parameters in simulation. In order to probe the response to an increase in system size, we have performed a further four simulations using 64 water molecules, using a computational setup as for the other simulations in this section. These trajectories are only 10ps - 13ps long. Three have average temperatures of 335K, 329K and 328K, whereas one has a temperature of 306K. Moreover, the density used in the literature is subject to small variation. In order to span this uncertainty the three runs at higher temperatures were carried out in cells of slightly different size (the edges of the cubic cell are 12.40Å, 12.42Å and 12.44Å). The cell size of the lower temperature run is 12.42Å. The results of these simulations are shown in Fig. 3.5 and Fig. 3.6 for pair correlation functions and MSD respectively. Based on this (limited) evidence, it seems that system size effects and the influence of small unit cell changes are not very pronounced for the properties considered. At the same time, the observed temperature dependence appears to be transferable to these larger systems.

3.6 Performance of six different density functionals

After this preparation, let us now return to the question of the accuracy of density functionals for liquid water. This has been tested previously (see e.g. (SHP96; BDHS00)), but the significant increase of computing power and the efficiency of our scheme allows for longer production runs and longer equilibration. Simulations have all been started from the same initial configuration, with the same initial velocities, same ba-
Figure 3.5: The Oxygen-Oxygen pair correlation functions for four simulations of 64 water molecules (solid lines) are compared with the results for 32 water molecules (dashed lines). The two more structured curves are for simulations with a temperature near 305K, whereas the four less structured curves are for temperatures near 330K. The three solid lines that are nearly indistinguishable are simulations with slight changes in the unit cell (12.40Å, 12.42Å, 12.44Å) and this indicates that such changes have little effect on the structure of liquid water. The relatively good agreement between the simulations at the same temperature but different particle number indicates that size effects are small and that the temperature dependence observed for 32 water molecules carries over to 64 water molecules.
Figure 3.6: Mean square displacements for the systems described in Fig. 3.5. The solid lines correspond to simulations with 64 water molecules, the dashed lines to simulations with 32 water molecules. The lower two lines correspond to temperatures near 305K, whereas the four upper curves are for simulations near 330K. There is no trend corresponding to the variations in unit cell.
Table 3.2: Characteristics (time length, temperature, energy drift in a.u.×10⁻⁶/(ps atom)) and computed properties (mean square displacement (MSD), maximum of the Oxygen-Oxygen pair correlation function) for a 32 molecule system comparing six different density functionals. Experimentally, at room temperature, the MSD is 1.3 Å²/ps, and \( g(r)_{OO}^{max} \) is 2.8.

sis (QZV2P) and convergence settings (\( \epsilon_{SCF} = 10^{-7} \)), but with different density functionals. The initial configuration and velocities were taken from a BLYP run at approximately 330K, so that the non-ergodic region, at least for the BLYP functional, is avoided. The length of all runs is again approximately 20ps. The functionals we have investigated are BLYP(Bec88a; LYP88), PBE(PBE96), TPSS(TPSS03), OLYP(HC02), HCTH120(HCTH98; BDHS00), HCTH407(HCTH98; BH01). The TPSS functional is a recent meta-GGA and explicitly introduces the kinetic energy density (\( \tau \)) in the exchange and correlation energy. A PBE pseudo potential has been employed for the TPSS simulations. Comparing bond lengths for a set of dimers to all electron calculations, and repeating the simulations of the liquid with a BLYP pseudo potential (data not shown), we verified that the influence of the pseudo potential is negligible.

The results are shown in Fig. 3.7 and Fig. 3.8 and summarized in Table 3.2. The strong variations between the different functionals are unambiguous. The MSD derived from the BLYP, PBE and TPSS simulations clearly point at a too sluggish dynamics, which is consistent with over structuring of the pair correlation functions. The three other functionals are significantly more diffusive, and correspondingly less structured. However, OLYP and HCTH407 exhibit too little structure to compare well with experiment. These results are even more striking if one considers the temperatures (Table 3.2) to which these runs equilibrate. Indeed, e.g. the diffusion of HCTH407 is about twice as fast as the diffusion of PBE, even though the later run is at about a 30K higher temperature. The
Figure 3.7: MSDs as obtained for simulations with the BLYP, PBE, TPSS, OLYP, HCTH120 and HCTH407 functionals using solid, dotted, dashed, long dashed, dash dotted, long dash dotted lines respectively.

fact that, starting from the same configuration and velocities, some simulations equilibrate to a lower temperature is consistent with their more 'liquid-like' behavior.

3.7 Conclusion

We have investigated the effect of temperature and different choices of density functional on results obtained from ab initio simulations of liquid water under carefully controlled conditions. These simulations show that current functionals may yield qualitatively different predictions for properties of the neat liquid. In light of the results available in literature about DFT performance in the gas phase, this is not a surprise. However, as extrapolation from dimer calculations to the liquid proves unreliable, the implication is that testing and deciding upon an appropriate functional is necessary when simulating aqueous systems. In spite of these shortcomings, and if the somewhat empirical approach to circumvent them is accepted, DFT provides a reasonable description of the liquid and is able to deal with the tiny energy scales involved. Our relatively long simulations con-
Figure 3.8: Oxygen-Oxygen pair correlation functions as obtained for simulations with the BLYP, PBE, TPSS, OLYP, HCTH120 and HCTH407 functionals indicated by solid, dotted, dashed, long dashed, dash dotted, long dash dotted lines respectively.
firm the results of previous work (APK03; GSD+04; KMM+04) that the application of either BLYP or PBE functionals leads to more structure and a more sluggish dynamics that anticipated in earlier simulations, based on shorter trajectories. The same holds for TPSS, a recent meta-GGA. From a practical point of view, a more diffusive liquid, such as obtained with OLYP, HCTH120, and HCTH407 is a significant advantage for modeling reactions in solution, as the time scales needed to sample solvent configurations become shorter, and equilibration is faster.

In agreement with experimental work, we have observed strong temperature dependence of structural and dynamical properties. This should be kept in mind while performing simulations of water at ambient conditions, i.e. close to the triple point. Part of the discrepancies in literature can be explained by the uncertainty in the temperature estimate of relatively short simulations, and the risk of poor equilibration violating equipartition. Furthermore, even for the runs of the order of 20ps as performed in this study, the observed non-ergodic behavior for the BLYP simulations at lower temperature should be taken into account when deciding on the simulation protocol. While our results confirm that the ab initio molecular dynamics technique is not yet a black box tool for the condensed phase, we remain convinced of the merits and strengths of the method.
Chapter 4

The Quantum Mechanics / Molecular Mechanics (QM/MM) Method

Even with efficient methods one would always like to be able to do even bigger systems. Empiric classical potential methods can handle much larger systems (and for longer times). Unfortunately an appropriate classical potential is not always known, and normally chemical reactions cannot be described. For a class of systems it is possible to describe at least most of the system with a classical potential, only for a small part of the system this is not possible. In this case using a hybrid method would be an optimal solution. This is exactly what a QM/MM approach tries to do, connecting these different all-atom representations: QM/MM partitions the system in a (small) QM part and the rest of the system that is treated at the MM level.

This method was pioneered by Warshel and Levitt (WL76), and has developed in a very active research area (SK86; MM95; Tho95; YTM01; CSM+03; LVR02; NGY05).

To have an hybrid QM/MM system, one has to subdivide the system $S_{QM+MM}$ in a QM part $S_QM$ and an MM part $S_{MM}$. Normally the easiest way to achieve this is to start from a rough classical description of the whole system $S_{QM+MM}$, and label some atoms of it, those that cannot be described well with the classical potential, as QM. These atoms then define $S_QM$ and $S_{MM}$. 
As QM description in the literature different levels of theory were adopted: semi empirical (FAB+00; DZHS85), ab initio Hartree-Fock (LHK99) post Hartree- Fock, (FO03) or DFT Hamiltonian, (CSM+03; LVR02; LHK99; ETHP99; LLL03). In this chapter after briefly reviewing the classical potentials in section 4.1 we will describe the integration of a QM/MM method in the cp2k package, that uses the previously described GPW method.

The electrostatic interaction between the QM and MM part is a critical part for an efficient evaluation of the QM/MM energy; in section 4.2 we will review the novel approach that we have taken to solve this problem. The treatment of the electrostatic in a fully consistent way with the periodicity of the MM cell is discussed in the outlook section.

If there are bonds between the QM and MM part then one cannot define the QM part as just the atoms labeled QM, because the electronic structure of the QM part would never converge to an electronic structure similar to the one that would be obtained treating the whole system as QM. In that case one needs to introduce extra “link” atoms in the QM system. This is an important issue for crystalline or biological systems, but not one on which we focus. A small review of the problem and partial solutions is given in section 4.4.

Parts of this chapter are taken form the article Teodoro Laino, Fawzi Mohamed, Alessandro Laio, and Michele Parrinello, An Efficient Real Space Multi grid QM/MM Electrostatic Coupling, J. Chem. Theory Comput. 2005, 1, 1176–1184. I actively participated at the development of the electrostatic interaction formulation, fitting of the potentials with the Gaussian, and the development of the spline based multi grid interpolation method, and generally to the development of the method and integration in cp2k. Teodoro Laino, apart participating to the previous activities did most of the work related to the implementation and MM integration.

### 4.1 Classical potentials

Many functional forms exist for empirical potentials, but for biological applications most of these are quite similar to the functional form that we use to describe the MM system,
4.1 Classical potentials

which is basically the one in (CCB+95):

\[
E_{\text{total}} = \sum_{\text{bonds}} K_r (r - r_{eq}) + \sum_{\text{angles}} K_\theta (\theta - \theta_{eq}) + \sum_{\text{dihedrals}} \sum_n \frac{V_n}{2} [1 + \cos(n\phi - \gamma)]
\]

\[
+ \sum_{i<j} \left[ \frac{A_{ij}}{R_{ij}^2} - \frac{B_{ij}}{R_{ij}^6} + \frac{q_i q_j}{\epsilon R_{ij}} \right]
\]  

(4.1)

where \(K_r\) is normally simply a constant times an harmonic term, \(K_\theta\) is either a constant times an harmonic potential or a constant times a cosine. The first three term compose the bonded forces: these potentials have a fixed topology, the molecules and bonds present in the system are part of the definition of the system one is studying, and they are not expected to change. The last terms constitute the non-bonded forces: Lennard-Jones and electrostatics.

Bond lengths (and sometime even rotational degrees of freedom) sometime can be frozen, thus removing short range oscillations that would need a small time step, or a more complex RESPA(TBM91) integration algorithm.

There are empirical potentials that allow for some simple reactions channels but they are difficult to parameterize (one needs to know the reaction coordinate, or to approximate a highly dimensional potential), and become easily computationally very costly. For this reasons, and because bond-breaking would make explorations of the systems at temperatures that break bonds, for example for simulated annealing or parallel tempering, more difficult, potentials for bio-molecules like AMBER(Cas02) and CHARMM(BBO+83) that need to be easy to compute and able to describe a molecule at normal conditions and in a range around them don’t allow bond-breaking.

The explicit notion of a molecule of these potentials is not present in the ab-initio potentials for which there are just atoms, the molecules just arise implicitly as groups of atoms that are strongly bound together, but they are not explicitly programmed in the potential. This is the reason ab-initio potentials can describe chemical reactions, but it also means that one cannot easily take advantage of the knowledge that some molecule are present, and stable, in the system.

One needs to be able to systematically find the parameters of the potential that describe the interactions with the close-by atoms of the same molecule that have not been frozen. This explains why simple harmonic potentials (or approximations of it for torsion or angular interactions) are used for the bonded forces. Using them it is also possible to
extract these values from ab-initio calculations.

The long range electrostatic interaction of the molecule, should also be reproduced, and for that these potentials have a non bonded electrostatic part that is composed by point charges. Normally the point charges are associated with the nuclei of the atoms, but they can be also at positions that are a function of them. The charges are often bigger than the partial charge of an atom calculated with ab-initio methods because their interaction should also reproduce polarization and other dynamic effects. As a consequence of that, and to make the interaction between bonded atoms only dependent on the bonded (harmonic) term and not on the charges, thus simplifying the fit of the parameters, some of these electrostatic interaction are removed from the potential.

To describe the van der Waals forces, and the repulsion of the cores of the atoms a Lennard-Jones potential is also used.

The kind of potential I just described covers the common ones like AMBER and CHARMM, and what is implemented in CP2K/FIST, the classical module of CP2K, mostly developed by Will Kuo and Chris Mundy. There are other potentials that try to include polarization, proton transfer,... and also these can be used instead of the MM potential in a hybrid QM/Pot approach (SS00) but we will not discuss those more complex potentials.

4.1.1 QM/MM interaction

The GPW method uses a grid to calculate the Hartree and exchange-correlation energy. Thus using it or any other method that uses a fine grid as part of the computation the QM atoms should be spatially confined in a small part of the system, otherwise having a QM box as big as the whole system would be computationally too expensive for many systems. If the QM box is different from the MM box, periodic boundary conditions on the QM box cannot be used. One can begin to use non-periodic boundary conditions for the QM box and, as it is easier to do so, this is the method that I will present. The issues connected with a fully MM-periodic solution will be discussed in the outlook section.

A QM/MM system is not just a QM system and an MM system separated, the QM and MM system have to interact with each other. The energy of the QM/MM system is
4.1 Classical potentials

given by

\[ E = E_{QM} + E_{MM} + E_{QM/MM} \]  \hspace{1cm} (4.2)

where \( E_{QM} \) is the non-interacting (gas-phase) energy calculated with the GPW method, \( E_{QM/MM} \) is the interaction energy and \( E_{MM} \) is the classical energy. \( E_{MM} \) is calculated on \( S_{QM+MM} \) excluding the bounded forces (bonds, bends, dihedrals) and charges that are fully in \( S_{QM} \), but keeping the Van der Waals interactions. This is very close to \( E_{MM}(S_{QM+MM}) - E_{MM}(S_{QM}) \), as will be explained more in depth in section 4.7.

The main interaction is the electrostatic interaction: the QM system is in the field generated by both the QM and MM part.

\[ E_{QM/MM} = \int V_{\text{loc}}^\text{LR}(r)\rho_{cl}(r)dr + \int \frac{n(r)\rho_{cl}(r')}{|r - r'|}drr' \]  \hspace{1cm} (4.3)

where \( V_{\text{loc}}^\text{LR}(r) \) is the long range part of the pseudo-potential of all the QM atoms, \( n(r) \) is the electronic density and \( \rho_{cl}(r) \) is the classical charge density that is defined as

\[ \rho_{cl}(r) = \sum_{I\in \text{MM}} \frac{q_I}{(\sqrt{\pi}r_I^c)^3} e^{-\left(\frac{|r - r_I|}{r_I^c}\right)^2}, \]  \hspace{1cm} (4.4)

where \( q_I \) is the classical charge of atom I, and \( r_I^c \) is an atomic parameter dependent on the atom type.

This means that the quantum system sees the classical atoms as Gaussian charges. The charges in the MM system are modeled as point charges, but physically an extended density of the order of the covalent radius would be a better model. Such an interaction would change only the very short range part of the interaction, a part that the classical forcefield describes anyhow with van der Waals and bonded interactions. For this reason in classical forcefields the more complex extended charge model is not used, apart in the particle-mesh method, but just as a technical detail to calculate the electrostatic interaction more efficiently.

When interacting with the QM part a point charge introduces the so called “electron-spill out” problem. An MM point charge can trap in its center some electron density of a close by QM atom, whereas the effective potential of a QM atom at the place of the MM atom would not.

Using a smeared charge for MM atoms is a simple method to avoid these problems. Laio et al. (LVR02) used a rational polynomial form for the potential of the MM atoms,
mostly for efficiency reasons, but a more natural choice, especially in Gaussian based program, is a Gaussian charge distribution, as was used in (AF03; DEB+02) at the boundary where covalent bonds are broken. Giving only a finite width to the classical charges is a quite crude approximation to handle a broken covalent bond, but we don’t use it for that purpose, the finite width is used to represent all the classical charges, preventing the electron spill-out problem due to the absence of the Pauli repulsion of the electrons of the MM atom, and as parameter to fit to describe better the interaction of QM atoms with close-by MM atoms, as in (LVR02).

4.1.2 Efficiency

The cost of a QM/MM calculation is composed by the cost of the MM-calculation, the cost of the QM calculation (often much bigger than the cost of the MM calculation), and the cost of the interaction energy 4.3.

Taking into account the definition of $V_{\text{kin}}^{LR}$ (2.22) the interaction energy becomes

$$E_{\text{QM/MM}} = \int \left( \sum_{\beta \in \text{QM}} - \frac{\alpha_{\text{PP}}}{\sqrt{\pi}} \right)^2 Z_{\text{ion}} \delta e^{-\alpha_{\text{PP}} |r-R_{\beta}|^2} + n(r) \right) \int \frac{\rho_{\text{QM}}(r')}{|r-r'|} dr' dr$$

This interaction can be calculated directly using Gaussian integrals, which done naively would scale like $O(N_{\text{QM}}^2 N_{\text{MM}})$ or using an intermediate representation for $V_{\text{MM}}$. Having a grid it is natural to use it to collocate $V_{\text{MM}}$. In this case the cost of the interaction energy scales like the number of grid points times the number of MM atoms, which should be proportional to $O(N_{\text{QM}} N_{\text{MM}})$: linear in both the number of QM atoms ($N_{\text{QM}}$), and the number of MM atoms ($N_{\text{MM}}$).

Unfortunately due to the large number of grid points $\approx 10^6$ and MM atoms $\approx 10^4$ or more for systems of biological interest, it is clear that a brute force evaluation of this term is impractical. Indeed in methods that use a grid, typically between 10% and 100% of the time used for the QM calculation is needed for this evaluation also using sophisticated techniques like hierarchical multi-pole (HMP) (ETHP99) or of clever implementations based on electrostatic cutoffs and potential form that can be efficiently evaluated (LVR02).

In the GPW method once $V_{\text{MM}}$ is on the QM grid, it can be treated like the other
4.2 GEEP: Gaussian Expansion of the Electrostatic Potential

To efficiently evaluate

\[
V^{\text{MM}}(r) = \int \frac{\rho_d(r')}{|r - r'|} dr' \\
= \int \sum_{l \in \text{MM}} \frac{q_l}{(\sqrt{\pi r_f^2})^2} e^{-(r - R_l)^2} \frac{1}{|r - r'|} dr' \\
= \sum_{l \in \text{MM}} \frac{\text{erf}(\frac{|r - R_l|}{r_f})}{|r - R_l|} \\
= \sum_{l \in \text{MM}} v_{r_f}^{\text{MM}}(r - R_l)
\]

we devised a novel schema (LMLP'05).

The basic idea is to express the potential

\[
v_{r_f}^{\text{MM}}(r) = \frac{\text{erf}(\frac{|r|}{r_f})}{|r|}
\]

as sum of Gaussians, plus a remainder function \( R_{\text{low}} \) that is smooth

\[
v_{r_f}^{\text{MM}}(r) = \sum_{i=1}^{N_g} A_i e^{-\frac{x_i^2}{r_f^2}} + R_{\text{low}}(r).
\]

The Gaussians can be collocated efficiently with the multi grid techniques used for the GPW method, and \( R_{\text{low}} \) should be collocated on a very coarse grid.

To really make \( R_{\text{low}} \) smooth the Gaussian parameters are optimized to remove the high G components of \( v_{r_f}^{\text{MM}} \). This was achieved by performing a least square fit in the G-space for \( g > G_{\text{cut}} \) using the analytical representation of the Fourier transform of a Gaussian

\[
\int e^{-i\omega \cdot \mathbf{g}} e^{-\frac{x^2}{a^2}} = e^{-\frac{2\pi^2 \omega^2}{4}}
\]
and of $v_{ij}^{\text{MM}}$

$$\hat{v}_{ij}^{\text{MM}}(g) = \frac{4\pi}{g^2} e^{-\frac{g^2r_i^2}{4}}$$

(solution of the Poisson equation with a single Gaussian charge).

To perform this least square minimization we developed a Mathematica notebook that is available together with cp2k (CP204).

To derive optimal coefficients one can also use the fact that

$$a v_{a}^{\text{MM}}(a r) = b v_{b}^{\text{MM}}(b r)$$

and so if $A_i^a$ and $B_i^a$ are the coefficients of a fit for $v_{a}^{\text{MM}}$ with $g > G_{\text{cut}}^a$ then

$$A_i^b = \frac{a}{b} A_i^a \quad B_i^b = \frac{b}{a} B_i^a \quad G_{\text{cut}}^b = \frac{a}{b} G_{\text{cut}}^a \quad R_{\text{low}}^b(r) = \frac{a}{b} R_{\text{low}}^a(\frac{a}{b} r)$$

is a fit for $v_{b}^{\text{MM}}$ with $g > G_{\text{cut}}^b$ with, which means that a fit for a smaller radius can (a little sub-optimally) be adapted to larger $r_i$.

The results of this optimization procedure for $r_i^c = 1.1$ Å and 0.44 Å are shown in table 4.1, and graphically for $r_i^c = 1.1$ Å in figures 4.1 and 4.2. It can be seen that with $r_i^c = 1.1$ three Gaussian are enough to have a $G_{\text{cut}} = 1$ which lets collocate $R_{\text{low}}$ on a grid with spacing an order of magnitude bigger.

Given the Gaussian parameters $A_i^f$ and $B_i^f$, $R_{\text{low}}^f$ is an analytic function, but as it has to be evaluated for each MM atom it is more efficient to fit it with a cubic spline, and calculate its value from the tabulated spline.

<table>
<thead>
<tr>
<th>no. of</th>
<th>$r_i^c = 1.1$ Å</th>
<th>$r_i^c = 0.44$ Å</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>$A_i$ (au) $B_i$ (Bohr)</td>
<td>$A_i$ (au) $B_i$ (Bohr)</td>
</tr>
<tr>
<td>1</td>
<td>0.103103 4.243060</td>
<td>0.230734 1.454390</td>
</tr>
<tr>
<td>2</td>
<td>0.125023 2.966300</td>
<td>0.270339 1.094850</td>
</tr>
<tr>
<td>3</td>
<td>0.098613 2.254250</td>
<td>0.075855 4.906710</td>
</tr>
<tr>
<td>4</td>
<td>0.190667 0.883485</td>
<td>1.965640 2.658160</td>
</tr>
<tr>
<td>5</td>
<td>0.173730 1.965640</td>
<td>3.591640 1.27689</td>
</tr>
<tr>
<td>6</td>
<td>0.127689 3.591640</td>
<td>0.095104 3.591640</td>
</tr>
</tbody>
</table>

Table 4.1: Optimal coefficients of the Gaussian for the two values of $r_i^c$ used for water.
Figure 4.1: Gaussian expansion of the electrostatic potential (GEEP) in real space. The picture shows the components of the fit of table 4.1 for the value $r_f^r = 1.1 \text{ Å}$. 
Figure 4.2: Fourier transform of $V_{\text{MM}}$ (in red) and Fourier transform of the residual function $R_{\text{low}}$ (in green). It can be seen that in this case ($r_c = 1.1 \ \text{Å}$) the residual function has a $G_{\text{cut}} = 1.0$. 
4.2.1 QM/MM Forces

The $V^{\text{MM}}$ potential can be added to the other grid potentials like $v_n^{\text{rc}}$. In this case if the forces on the QM atoms are calculated with the GPW method as highlighted in section 2.5, then they correctly take into account the QM/MM interaction

$$E_{\text{QM/MM}} = \int \rho_{qm}(\mathbf{r}) \sum_{I \in \text{MM}} \left( \sum_i A_I^I e^{-\frac{(|\mathbf{r} - \mathbf{R}_I|^2)}{\beta_I^2}} + R_{\text{low}}^I (\mathbf{r} - \mathbf{R}_I) \right) \, dV.$$  \hspace{1cm} (4.13)

For the forces on the MM atoms one has to derive (4.13) with respect to the displacement of an MM atom $I$ which gives:

$$\int \rho_{qm}(\mathbf{r}) \left( \sum_i A_I^I \nabla_r e^{-\frac{(|\mathbf{r} - \mathbf{R}_I|^2)}{\beta_I^2}} + \nabla_r R_{\text{low}}^I (\mathbf{r} - \mathbf{R}_I) \right) \, dV,$$

where $\nabla_r$ we mean the gradient with respect to $\mathbf{r}$ that in this case is equal to $-\nabla_{R_I}$.

This can be performed on the grid, and taking into account that the restriction is the prolongation operation one can (like in section 2.5) transfer $\rho_{qm}$ to the coarser grid level and perform the integration on the grid where the Gaussian (or $R_{\text{low}}$) is defined.

4.3 Algorithm

The complete sequence of operations to perform a QM/MM simulation then is

1. setup of the MM system (topology,...)
2. remove the classical interactions between QM atoms from the interactions to compute
3. tabulate $R_{\text{low}}$ from 0 to the cell diameter for each $r_j^I$ (if two Gaussian fits derive from the same coefficients only one table is needed)
4. setup of the QM system
5. calculate the MM energy and forces
6. collocate $V^{\text{MM}}$ on the grid:
(a) for each MM atom take the optimal Gaussian coefficients for its $r_j^*$-
(b) collocate each Gaussian on its optimal multi grid (skipping atoms for which
the Gaussian has no overlap with the QM-grid)
(c) for each MM atom calculate $R_{\text{low}}$ on the finest grid
(d) collect the potential on the finest grid using the non periodic spline prolongation
described in section 2.7

7. calculate the QM energy and forces performing the SCF adding the grid potential
$V^\text{MM}$ to the other grid potentials ($V^\text{H}$ and $V^\text{XC}$)

8. transfer $\rho_{\text{qm}}$ to the coarser grid levels with the restriction operation (section 2.7)

9. integrate the derivatives of the Gaussian and of $R_{\text{low}}$ each on its grid and add the
contribution to the MM atoms.

To recalculate the forces after having moved the atoms repeat from point 5 (skipping
the initial setup).

4.4 Bonds at the boundary

If there is a covalent bond between the QM and MM subsystem, an electron pair should
be shared between the QM and MM part, but as the MM part has no electrons this does
not work. It is clear that without some sort of extra interaction the QM system will never
converge to an electronic state that resembles the electronic state of the system treated
fully at the QM level.

This is a highly nontrivial problem, as one must be very careful not to count inter¬
actions of the atoms at the boundary twice.

It has not been our focus to address this problem. A common approach is to introduce
link atoms in the QM system to which the bond can be done. A very simple solution
is to cap the bonds with hydrogen atoms. This lets the QM system make a bond, but
typically the bond with an hydrogen has different properties than the original bond. In
particular it will have a different bond length. A way to handle this problem is to express
the position of the hydrogen bond as function of the coordinates of the atoms involved in
the bond (typically at a fixed position along the bond), and then redistribute the forces on the link atom through the constraints. This has been implemented in cp2k. Using this method one has to be very careful to either remove, or at least control the interactions of the added H-atom with the close-by MM atoms.

A better solution to this problem would be to use a modified pseudo potential for the link atom, that tries to reproduce bond length and strength of the original bond, as done in (ZLY99). This would let the link atom stay at the same place as an MM atom.

### 4.5 Validation

The calculation of the forces within the present implementation has been compared with the calculation of the forces using the method published elsewhere, (LVR02) which is a
Figure 4.4: On the left: energy conservation of a system composed of 3 water molecules equilibrated at 400 K during 1 ps of simulation. The red line shows the total energy for the QM/MM run, the green line represents a pure classical run, and the blue line shows a pure quantum run. The total energies have been shifted for better visualization. No drift is observed, and all energy conservations are consistent. On the right: potential energy during the same run. Its variation is 3 orders of magnitude larger than the total energy variation.

QM/MM implemented in the CPMD code. Comparison with the CPMD-QM/MM code is complicated by the fact that in this last scheme a multi-polar expansion is used for the long-range part of the QM/MM electrostatic coupling, leading to inaccuracies. For this reason we compare only forces on atoms of the first MM solvation shell, which are treated exactly also in the CPMD-QM/MM code. The realistic problem was made up of 215 classical SPC water molecules and 1 QM water molecule. Although the system size is relatively small, the number of molecules present is comparable to the number of molecules normally treated exactly, and not through the multi-polar expansion, in CPMD-QM/MM. In Figure 4.3 we show the relative error between the CPMD-QM/MM code and the present implementations. The highest relative errors (less than 1.0%) correspond to forces that have small modulus ($10^{-3}$ au). The average relative error is $\approx 0.01\%$ with a speed-up in the energy and derivative evaluation of a factor of 40 wrt. the CPMD program.

An important benchmark for QM/MM codes that are aimed at molecular dynamics (MD) simulation is their ability to conserve energy. The system studied was composed of 3 water molecules (2 MM and 1 QM for the QM/MM run) previously equilibrated at 400 K. The simulation time was 1 ps, and results are shown in Figure 4.4. For comparison the energy of the pure classical and the quantum run are shown in the same picture. No
4.6 Results and Discussion

Consistent with checks done in previous work (ETHP99; LVR02), we test the accuracy of our implementation by computing the pair correlation function of a QM system embedded in a classical solvent. As found elsewhere (LVR02), the smoothing radius plays an important role in determining the bond properties of the system, and the choice of this parameter can have dramatic effects on pair correlation functions. The use of a different functional form (compare (4.7) with Eq 3 of Laio et al. (LVR02)) forced us to re-parameterize the $r^*_j$.

For the classical water molecules, the cutoff radii $r^*_j$ were chosen in order to reproduce the coordination number and the main peaks of the classical SPC water pair correlation function. A system of 2560 water molecules (2559 classical SPC water and 1 quantum water) in a cubic box, subject to periodic boundary conditions, was investigated. The system was previously equilibrated at $\rho = 1g/cm^3$, $T = 298K$. One SPC water molecule was then replaced by a QM water molecule. The GTH pseudo potential (GTH96) described in section 2.4.2 was used to describe the core charge distribution, and BLYP exchange-correlation density functional (Bec88a; LYP88) was employed in all the calculations, in conjunction with a cell plane-wave cutoff of $E_{cut} = 280Ry$. Several runs, with different values of the radius parameter $r^*_j$ were performed. The optimized radii are 0.44 Å for hydrogen and 1.20 Å for oxygen and allow the full classical SPC pair correlation function to be reproduced, as shown in Figure 4.5. Due to the different functional form of Eq 3,
the optimal values found with our implementation are different from the ones previously published (LVR02).

To test the transferability of the $r_j^f$ parameters determined for water, we also evaluated the pair correlation function of a QM dipeptide (GLY-ALA) zwitterion solvated in 3352 SPC water. We aimed at reproducing the pair correlation function obtained with the AMBER force field (Cas02). The pair correlation functions obtained with the present QM/MM implementation are indeed extremely close to the full classical results (see Figure 4.6).

4.7 Outlook - Periodic QM/MM

The method described so far does not treat the boundary conditions in a fully consistent way. In fact what one wants is either a QM cluster in an infinite mm crystal, or a periodic repeated QM/MM system. A better approach for the first type of system is to adopt an energy subtraction method (SS00) which without taking into account eventual link atoms can be written as

$$ E = E_{\text{MM}}(S_{\text{MM}} + S_{\text{QM}})_p + E_{\text{QM}}(S_{\text{QM}})_n - E_{\text{MM}}(S_{\text{MM}})_n, \quad (4.15) $$

where $p$ means periodic boundary conditions, $np$ without periodic boundary conditions and $S_{\text{MM}}$ and $S_{\text{QM}}$ are the MM and QM subsystems. This is not perfect, because the periodic system is the one deformed by the different QM potential used in the cluster. The error of this approximation depends on how well the MM potential describes the $S_{\text{QM}}$ cluster. The fact that one has to describe the QM part at MM level, and with a description that closely matches the QM one is a clear drawback of this method. What we do in our method is more like

$$ E = E_{\text{MM}}(S_{\text{MM}} + S_{\text{QM}})_p + E_{\text{QM}}(S_{\text{QM}})_n - E_{\text{MM}}(S_{\text{MM}})_p, \quad (4.16) $$

which means that the periodic copies of the MM systems have a hole in them. Actually in our formulation to avoid completely the definition of $S_{\text{QM}}$ at the MM level we replace the interaction between the MM and QM systems $S_{\text{MM}}$ and $S_{\text{QM}}$ calculated using the MM level interaction with a direct QM/MM interaction

$$ E = E_{\text{MM}}(S_{\text{MM}})_p + E_{\text{QM}}(S_{\text{QM}})_n + E_{\text{QM/MM}}(S_{\text{QM}}, S_{\text{MM}})_p. \quad (4.17) $$
Figure 4.5: H-O and O-O pair correlation functions for QM water. QM/MM values are compared with the full classical SPC calculation. The QM/MM calculations are performed with $r_f^q$ equal to 1.2 Å for oxygen and 0.44 Å for hydrogen. The quantum box and the classical box employed in the simulation have a cube box size of 10.0 Å and 42.0 Å, respectively. The roughness of the QM/MM curve is due to the much shorter simulation time. The bin size for the evaluation of the pair correlation function is of 0.2 Å.
The unphysical hole in periodic copies is the price to pay for not wanting to describe $S_{QM}$ with the MM forcefield, and calculating $E_{QM}(S_{QM})$ as cluster with this naive approach.

A more consistent way to treat the QM/MM system would be

$$E = E_{MM}(S_{MM} + S_{QM})_p + E_{QM}(S_{QM})_p - E_{MM}(S_{MM})_p,$$  \hspace{1cm} (4.18)

where basically the calculation at MM level of the QM system $E_{MM}(S_{QM})$ is replaced by the QM one $E_{QM}(S_{QM})$, or actually in our formulation

$$E = E_{MM}(S_{MM})_p + E_{QM}(S_{QM})_p + E_{QM/MM}(S_{QM}, S_{MM})_p.$$  \hspace{1cm} (4.19)

This is a fully consistent and physically sound representation where the QM system $S_{QM}$ does not need to be represented at the MM level.
4.7 Outlook - Periodic QM/MM

To use it one has first to calculate the QM/MM interaction energy with periodic boundary conditions

\[
E_{\text{QM/MM}}(S_{\text{QM}}, S_{\text{MM}}) = \sum_{ijk} \int \rho_{\text{qm}}(r) V_{\text{MM}}(r - R_{ijk}) dr
\]

\[
= \int \rho_{\text{qm}}(r) \sum_{I \in \text{MM}} \sum_{ijk} v_{ij}^{\text{MM}}(r - R_I - R_{ijk}) dr,
\]

(4.20)

where \( R_{ijk} = h[i, j, k]^T \) are vectors of the MM lattice, and \( h \) is the MM cell matrix.

Thus the potential of each MM atom is given by the sum of the contributions of all its periodic images

\[
\sum_{ijk} v_{ij}^{\text{MM}}(r - R_{ijk}) = \sum_{ijk} \sum_{m=1}^{N_g} A_m e^{-\frac{(r - R_{ijk})^2}{2b_m}} + \sum_{ijk} R_{\text{low}}(r - R_{ijk}).
\]

(4.21)

The summation of the \( N_g \) Gaussian and their periodic copies can be done in real space as done during the collocation of the basis set Gaussian in the GPW method (see section 2.7.5.1). Actually often it is even faster than with basis set because due to the large size of the MM box only the closest image contributes, and the MM Gaussian have no angular part. The sum on the \( R_{\text{low}} \) part can be done in the reciprocal-space (G-space).

\[
\tilde{R}_{\text{low}}(g) = \int \sum_{ijk} R_{\text{low}}(r - R_{ijk}) e^{-i\mathbf{g} \cdot \mathbf{r}} dr = \frac{4\pi}{g^2} e^{-\frac{\theta^2}{4}} - \sum_{m=1}^{N_g} A_m \pi^{3/2} B_m e^{-\frac{\theta^2}{4m}}
\]

(4.22)

and

\[
\tilde{R}_{\text{low}}(\mathbf{r}) = \frac{1}{\Theta} \sum_{0 < g_{ijk} < G_{\text{cut}}} \tilde{R}_{\text{low}}(g_{ijk}) e^{i\mathbf{g}_{ijk} \cdot \mathbf{r}}
\]

(4.23)

where \( \Theta \) is the volume of the MM cell, and \( g_{ijk} = 2\pi h^{-1}[i, j, k]^T \) are the reciprocal vectors with the correct periodicity. Note that because of symmetries only terms like

\[
e^{i\mathbf{g}_{ijk} \cdot \mathbf{r}} + e^{-i\mathbf{g}_{ijk} \cdot \mathbf{r}} = 2\cos(\mathbf{g}_{ijk} \cdot \mathbf{r})
\]

(4.24)

are nonzero, and one can rewrite the potential as

\[
\tilde{R}_{\text{low}}(\mathbf{r}) = \frac{1}{\Theta} \sum_{g_{ijk} < G_{\text{cut}}} \tilde{R}_{\text{low}}(g_{ijk}) \cos(g_{ijk} \cdot \mathbf{r})
\]

(4.25)

Also \( E_{\text{QM}}(S_{\text{QM}})_p \), i.e. the QM energy with periodic boundary conditions with respect to the MM cell has to be calculated. If the electrostatic problem is solved in the QM box
it will have the wrong periodicity. To correct for it one can try to use a simplified $\tilde{\rho}_{qm}$, that reproduces the long range behavior of the electrostatic interaction of $\rho_{qm}$, and then calculate the energy difference between the two periodicities with the fitted density. Blöchl (Blö95) uses a density that is obtained fitting Gaussian charges at the atom positions. This method is the one that Teodoro Laino implemented in CP2K and that we did describe in (LMLP06).

This works well as we did show in (LMLP06), but the fitting procedure of the Blöchl method can become costly. Another method, that could be used in the future, uses the same philosophy as the multi grid interpolation schema. The idea is to transfer the charge density $\rho_{qm}$ from the coarsest QM grid $G_{QM}^1$ to the MM grid $G_{MM}$, solve the Poisson equation on the MM grid and on the QM grid obtaining $V_{MM}^H$ and $V_{QM}^H$. Then transfer $V_{QM}^H$ from $G_{QM}^1$ to $G_{MM}$ to have $G_{MM}(V_{QM}^H)$, and from it calculate

$$G_{QM}^1(V_{MM}^H - G_{MM}(V_{QM}^H))$$

(4.26)

on the coarse QM grid, and from there push it down to all grid levels. This potential basically corrects the wrong long range part of the potential calculated on the QM grid by replacing it with a potential that has the correct periodicity, as it is calculated on the MM grid.

For this to work one has to describe how to go from $G_{QM}^1$ to $G_{MM}$ and vice-versa. The idea is to use the spline method described in section 2.7, and define the prolongation as

$$P_{MM}^{QM} = T_{MM}^{QM}(S_1)^{-1},$$

(4.27)

and in the opposite direction

$$P_{QM}^{MM} = (P_{MM}^{QM})^T = ((S_1)^T)^{-1}(T_{MM}^{QM})^T.$$  

(4.28)

where $S_1$ is exactly the same as the one for the non periodic boundary conditions and $T_{MM}^{QM}$ uses, just like $T_2^1$, the $M_i$ instead of $N^3$, but as $G_{QM}^1$ and $G_{MM}$ grids are not necessarily commensurate, and might be shifted, the $M_i$ are not evaluated just at half integers.

For $i_{qm}$ and $i_{mm}$ indices in the QM and MM grids

$$h_{qm} \frac{i_{qm}}{N_{qm}} - h_{mm} \frac{i_{mm}}{N_{mm}} = \Delta x$$

(4.29)

where $h_{qm}$ and $h_{mm}$ are the cell vectors of the QM and MM system, $N_{qm}$ and $N_{mm}$ are the number of QM and MM grid cells. Each index component should be divided by the
The number of grid cells in that direction and then multiplied with the cell matrix. The shift between the two cells is given by $\Delta x$. From the previous expression one can calculate for all the MM grid points $i_{mm}$ that are fully inside the QM grid, the (not necessarily integer) points
\[(i_{qm})_k = (h_{qm}^{-1}(\Delta x + h_{mm} \frac{i_{mm}}{N_{nm}}))_k(N_{qm})_k \quad (4.30)\]
at which one has to evaluate the $M$ to obtain $T$.

One can see the QM density on the MM grid as some sort of $\tilde{\rho}_{qm}$ that fits the long-range part of the QM density, and thus see the similarities with the Blöchl method.

This method has a cost of $O(N_{QM} + N_{MM})$ (i.e. it depends explicitly on the number of MM atoms through the MM grid), and should be performed at each SCF step. It could be a competitive alternative to the Blöchl method.

The periodic method presented in this section is fully consistent, but if one wants to study a single charged surface defect on a crystal surface, or other systems that have a large interaction radius it might be better to try to remove the periodic interaction between the $S_{QM}$. To avoid the problem of the hole in the periodic copies one can start with the electrostatic potential of an unperturbed periodic MM crystal $\tilde{V}_{p}^{MM}$. This potential can be calculated on a very small sample if the crystal is uniform. Then introducing $\tilde{S}_{QM+MM}$, the unperturbed MM crystal in the QM and MM region, one can calculate a potential that contain the madelung (periodic) interaction from
\[\tilde{V}_{p}^{MM} - V_{np}^{MM} (\tilde{S}_{QM+MM}), \quad (4.31)\]
where basically one creates the potential of a crystal with a hole. The potential can be added to the cluster calculations
\[E_{MM}(\tilde{S}_{QM} + \tilde{S}_{MM})_{np} + E_{QM}(S_{QM})_{np} - E_{MM}(S_{QM})_{np}, \quad (4.32)\]
so that the cluster is embedded in a periodic crystal. As the potential of the crystal stays frozen, it is better to constrain the atoms at the external border of $S_{MM1}$ at their place.

### 4.8 Conclusions

The QM/MM hybrid method was discussed, and a new algorithm for evaluating the QM/MM coupling term with a fast linear scaling implementation was presented. The
evaluation of the electrostatic potential on a grid is proportional to the number of MM atoms times the number of grid points. In real systems the linear scaling evaluation of the potential is therefore characterized by a prefactor $\approx 10^6$. In this scheme the prefactor is instead $\approx 10^3$. The number of floating point operations is reduced several orders of magnitude, and the computational time is 10-100 times smaller.

The algorithm is now implemented in the package CP2K, released under GPL license, and freely available on the Internet (CP204). The scheme was validated by checking the energy conservation, and for a realistic system numerical accuracy was verified by comparing the forces with the analytical method, with a mean relative error of 0.01%. In addition, we computed the pair correlation function of a QM water molecule in classical water and of a QM zwitterionic dipeptide in classical water. The modified Coulomb interaction and the multi grid approach reproduce correctly the structural properties of a QM water molecule solvated in classical water and the parameters obtained therein can be used effectively to describe the properties of an organic molecule containing both negatively and positively charged moieties, as in the case of the zwitterion.

All tests address the correctness of results. The performance analysis confirms the present algorithm as the state of the art for the evaluation of QM/MM interaction coupling. Moreover, at variance with the majority of present-day QM/MM methods, our scheme does not rely on electrostatic cutoffs and so avoids all related problems. Consequently, the present method offers a fast, easy-to-use code for QM/MM calculations of large biological and inorganic systems.

The extension of this method for a more consistent treatment of the periodic boundary conditions was also discussed, and an alternative method to re-couple the QM interactions with the correct MM periodicity was proposed.
Chapter 5

EP: Energy through DFT-Perturbation Theory

MM methods, as we have seen normally have a fixed topology, a concept of molecule. They work in many cases because often we know that the system we want to study decomposes in fragments that interact weakly. QM methods normally don’t have this, which can be an advantage, but it also means that they cannot take advantage of the fact that we know that some interactions are weak. In this chapter we present a technique that tries to take advantage of this knowledge to compute the energy efficiently.

Such methods have some similarities with hybrid methods like QM/MM as in some way they can be seen as hybrid methods that combine QM methods with QM methods. We are not interested in methods that use higher level QM methods like MP2 or coupled cluster, rather we would like to simplify DFT for weakly interacting fragments.

The idea is to subdivide the system in fragments that can be solved independently, and take into account the interaction between fragments in an approximate way. Domain decompositions approaches (Yan91; Goe99) subdivide the system in real space to be able to reach linear scaling, and are quite different from the approach we want to look at. The recently proposed density functional embedding schema (IKH06) based on the approach proposed by (Cor91) and developed by (WW93; Wes97; NLBW05) has the same goals and some similarities with our approach, but the molecular systems interact via an embedding potential derived from orbital free DFT, whereas in our approach also the interaction energy is calculated in an orbital based approach. This is closer to the
MI-SCF methods (NTSI01) that constrain the molecular orbitals to be localized on the atomic basis set of the fragments.

All this methods have some problems to describe well hydrogen as there is some charge transfer involved. Hydrogen bonds are very important, but are also the strongest of the “weak” interactions. D. Benoit and M. Parrinello (BSP01), have shown that second order DFT perturbation theory can describe hydrogen bonds and silica quite well from orthogonalized frozen orbitals. Recently Khaliullin, et al. (KHGB06) applied the same idea to MI-SCF solutions, and were able to get a very good accuracy. In the present work we use the wave-functions of a non-interacting molecular system as starting point for the second order perturbation theory.

The present chapter is more theoretical than the others because I have not yet implemented a big application using this approach. Nevertheless the perturbation approach presented here in itself is still attractive. It can be very useful for analysis, as seeing the perturbation response due to the interactions can give a better insight in the system, and it could be useful to build empirical potentials. Its simple structure can be useful to develop new methods, for example to study direct wave function propagation in the original Car-Parrinello spirit.

5.1 Derivative Notation

Calculating the second order expansion of the energy can become quite involved, especially because in this case, unlike in (BSP01), the basis set (Gaussian) is explicitly dependent of the position of the atoms. To perform the derivatives I developed a notation that has been very useful.

A derivative of $f$ with respect to $x$ gives a linear function that approximate $f$ at the derivation place. It is possible to use $dx$ as the argument of this linear function, or, as in differential geometry, as the dual vector of $\partial_x$ that extract the argument of the linear function that represent the derivative from a tangential vector.

This is especially useful with non commuting operators:

\[
\frac{\partial}{\partial x} (x \cdot a \cdot x) dx = dx \cdot a \cdot x + x \cdot a \cdot dx
\] (5.1)
Another useful thing is to use different alphabets, roman (i,j,...) for electrons and
greek for the atomic basis functions (µ, ν,...), sum between up and down indexes and
use the overlap as metric tensor to transform from up indexes do down indexes 
\( S_{ij} = \int \psi_i(r)\psi_j(r)dr \) and \( (S^{-1})^{ij} \) is its inverse

For example the coefficients \( C_i \) of a wave function \( \psi_i \) should be treated, if possible,
as rectangular matrix (for efficiency reasons) and multiplication both to the left and to
the right is linear in the coefficients \( C_i \).

This notation can be applied both to the abstract bra/ket notation, described in the
appendix A.2, and to the matrix form, which is closer to what must me implemented in
practice. It avoids all tensor products, place holders,... that I had to use the first time I
calculated the derivatives.

In some cases one knows the argument \( y \) (tangential vector) at which to evaluate the
linear function given by the derivative \( \frac{\partial}{\partial x} \). To specify it is passed as argument to the
differential \( dx \) following the idea that \( dx \) is the dual vector that extracts the \( x \) derivative,
but using square brackets to give some hint that it is a functional application.

\[
\frac{\partial}{\partial x} (x \cdot a \cdot x)dx[y] = dx[y] \cdot a \cdot x + x \cdot a \cdot dx[y] = y \cdot a \cdot x + x \cdot a \cdot y
\]  

When the final result is a real number like the energy, or expressions like

\[
F = \text{Tr}(HAA^\dagger)
\]  

then the derivatives are like

\[
\frac{\partial F}{\partial A} dA = \frac{\partial}{\partial A} \text{Tr}(H(AA^\dagger)) = \frac{\partial}{\partial A} \text{Tr}(A^\dagger HA),
\]  

where manipulations that use the linearity of \( \text{Tr} \) and the fact that any circular permutation
of its arguments does not change the value are very useful. Assuming that \( H \) does not
depend on \( A \) this becomes

\[
\text{Tr}(A^\dagger HdA + dA^\dagger HA) = \text{Tr}(A^\dagger HdA + c.c.)
\]  

where the \( +c.c. \) in the last expression means plus complex conjugate.

To verify the exactness of the formulas the Mathematica package for non-commutative
algebra (HSM01) has been useful.
5.2 Energy expression

We use an atomic basis $\phi_\mu$ that is non orthogonal

$$\langle \phi_\mu | \phi_\nu \rangle = S_{\mu \nu}$$

and with it we express the orbitals

$$\psi_i = \sum_\mu C_i^\mu \phi_\mu,$$

where $C_i^\mu$ is a rectangular $n_{mo} \times n_{ao}$ (molecular orbitals $\times$ atom orbitals) matrix that contains the orbital coefficients.

We want to write everything for nonorthogonal orbitals, so we introduce the overlap between orbitals

$$\bar{S}_{ij} = \langle \psi_i | \psi_j \rangle = \sum_{\mu \nu} C_i^{\mu \nu} S_{\mu \nu} C_j^\nu = (C^\dagger SC)_{ij},$$

and

$$\psi^i = \psi_i^* = \sum_{\mu j} (\bar{S}^{-1})^{ij} C_j^\mu \phi_\mu,$$

de the dual of the $\psi_i$. With it

$$\bar{S}_{ij} = \langle \psi_i | \psi_j \rangle = \delta_{ij}, \quad \langle \psi_i | \psi^j \rangle = \delta_{ij},$$

$$\langle \psi^i | \psi^j \rangle = \sum_j (\bar{S}^{-1})^{ij} \langle \psi_j | \psi_j \rangle = \sum_j (\bar{S}^{-1})^{ij} | \psi_j \rangle.$$  

The ao-density matrix $P$ is defined as

$$P = C S^{-1} C^\dagger \quad \iff \quad P^{\mu \nu} = \sum_{ij} C_i^{\mu} (S^{-1})^{ij} C_j^\nu$$

and density matrix is

$$\rho = \sum_i |\psi_i\rangle \langle \psi_i| = \sum_{ij} |\psi_i\rangle (S^{-1})^{ij} \langle \psi_j| = \sum_{\mu \nu} |\phi_\mu\rangle P^{\mu \nu} \langle \phi_\nu|,$$

where we have assumed single occupation of each orbital, so that the density is the projection into the occupied density orbital. It is not difficult to introduce a factor 2 at the right places if needed to account for spin degeneracy.
The energy can be written as:

\[
E = \frac{1}{2} \sum_i \int \left( \langle \psi_i | r \rangle \nabla^2 \langle r | \psi_i \rangle + \langle \psi_i | r \rangle \nabla^2 \langle r | \psi_i \rangle \right) dr
+ \frac{1}{2} \int \frac{\langle r | \rho | r' \rangle \langle r' | \rho | r' \rangle}{|r - r'|} drdr' + E_{xc}
+ \int V_{\text{ext}}(r) \langle r | \rho | r \rangle dr + \int V_{\text{pp}}(r, r') \langle r | \rho | r' \rangle drdr',
\]

where \( E_{xc} = E_{xc}[\rho] \) is the exchange correlation functional, \( V_{\text{ext}} \) is the external potential and \( V_{\text{pp}} \) is the pseudo-potential potential.

### 5.3 Second order expansion

The energy depends both on the ionic positions \( \mathbf{R} \) and the electronic ao-coefficients \( C \). The ground state energy (the one of interest) is the minimum with respect to \( C \) of the energy. If one has some \( C \) that are a guess for the optimal ones it is possible to do a Taylor expansions of the energy around \( C \)

\[
E(\mathbf{R}, C + \delta C) = E(\psi^0) + \frac{\partial E}{\partial C}[\delta C] + \frac{1}{2} \frac{\partial^2 E}{\partial C \partial C}[\delta C][\delta C] + o(\delta C^2)
\]

Now the idea is to minimize this expansion to second order, and find the optimal \( \delta C \). This can be found by solving the system of equations

\[
\frac{\partial E}{\partial C} dC + \frac{\partial^2 E}{\partial C \partial C} dC[\delta C] dC = 0
\]

that has formally the solution

\[
\delta C = -\left( \frac{\partial^2 E}{\partial C \partial C} \right)^{-1} \frac{\partial E}{\partial C}.
\]

This is a simple idea, and is basically what was done in (BSP01), and more recently in (KHGB06), only that we derive everything for non orthogonal orbitals.

#### 5.3.1 Derivatives

Using

\[
S^{-1}S = I \quad \Rightarrow \quad \frac{\partial S^{-1}}{\partial C} = -S^{-1} \frac{\partial S}{\partial C} S^{-1},
\]
and the definition of $\bar{S}$, it is possible to see that

$$\frac{\partial P}{\partial C}dC = \frac{\partial}{\partial C}(C\bar{S}^{-1}C^\dagger)dC = (I - PS)dC\bar{S}^{-1}C^\dagger + c.c. \quad (5.20)$$

and thus

$$\frac{\partial E}{\partial C}dC = \frac{\partial E}{\partial P}dP\left[\frac{\partial P}{\partial C}\right] = \text{Tr}(H\frac{\partial P}{\partial C})$$

$$= \text{Tr}(H((I - PS)dC\bar{S}^{-1}C^\dagger + c.c.)) = 2\text{Tr}(\bar{S}^{-1}C^\dagger H((I - PS)dC),$$

where

$$H = \frac{\partial E}{\partial P} \quad (5.22)$$

is the Kohn-Sham matrix.

If $V$ is the subspace spanned by the $C$ then $(I - PS)$ is a projection into the orthogonal subspace $V^\perp$. Eq. 5.21 is exactly what one expects: only changes in the occupied subspace, that at first level are perpendicular to all the $C$, can change the energy. This means that $dC$ has to be in $V^\perp$, which in differential geometry is the tangential space at the point $C$. Also note that this is exactly 0 (as expected) at the minimum when $HC = SCA$ for a diagonal matrix $A$.

This is the equation obtained in (SWP80), and a variant of which where the orthogonality in the fragment is kept into account (NTSI01) is used by the MI-SCF approaches.

The second derivative is more complex

$$\frac{\partial E}{\partial C\partial C}(dC_1dC_2) = \text{Tr}(H\frac{\partial^2 P}{\partial C\partial C}dC_1dC_2) + \frac{\partial^2 E}{\partial P\partial P}dP_1[\frac{\partial P}{\partial C}dC_1]dP_2[\frac{\partial P}{\partial C}dC_2] \quad (5.23)$$

First let’s look at the term

$$\frac{\partial P}{\partial C\partial C}dC_1dC_2 = C\bar{S}^{-1}dC_1^\dagger(-S + SPS)dC_2\bar{S}^{-1}C^\dagger$$

$$- C\bar{S}^{-1}dC_1^\dagger S\bar{S}^{-1}dC_2^\dagger(I - SP)$$

$$- C\bar{S}^{-1}dC_1^\dagger S\bar{S}^{-1}dC_2^\dagger(I - SP)$$

$$+ (I - PS)dC_1\bar{S}^{-1}dC_2^\dagger(I - SP) + c.c. \quad (5.24)$$

note that $-S + SPS = -(I - SP)S(I - PS)$. The second term if $dC_1 = C$ is the negative of the first order term, whereas if $dC$ is in range of $(I - PS)$ it vanishes. The third term is just like the second. These two terms connect the tangential space $V^\perp$ with the subspace $V$ of the $C$, and can be neglected if one restrict it to the tangential space, as in differential
5.3 Second order expansion

geometry. As they are connected with the correction to the second order to the curvature due to the orthogonality constraints they could be useful for a method that propagates the wave functions to get a solution that is closer to the geodesic than a tangential vector. We neglect them, thus

$$\text{Tr}(H \frac{\partial^2 P}{\partial C \partial C} dC_1 dC_2) =$$

$$\text{Tr} \left( HCS^{-1} dC_1 (-S + SPS) dC_2 S^{-1} C^\dagger \right)$$

$$+ H(I - PS) dC_1 S^{-1} dC_2^\dagger (I - SP) + c.c. \right) \quad (5.25)$$

In the second term of (5.23) for $dP$ is is possible to use the expansion we found before

$$\frac{\partial^2 E}{\partial P \partial P} dP_1[(I - PS) dC_1 S^{-1} C^\dagger + c.c. | dP_2[(I - PS) dC_2 S^{-1} C^\dagger + c.c.], \quad (5.26)$$

and the derivative is the second order kernel that appears in all perturbation calculations,

$$\frac{\partial^2 E}{\partial P \partial P} = \langle r | P_1 | r' \rangle \langle r' | P_2 | r' \rangle \frac{dE}{|r - r'|} + \frac{\partial^2 E_{XC}}{\partial P \partial P} dP_1 dP_2. \quad (5.27)$$

The details of the $E_{XC}$ derivative are explained in the appendix A.4.

Now if we call $P_1$ the place where $dP_1$ is evaluated

$$P_1 = (I - PS) dC_1 S^{-1} C^\dagger + c.c. \quad (5.28)$$

then if we define $V_{P_1}$ as the second derivative in the direction $P_1$

$$\frac{\partial^2 E}{\partial P \partial P} dP_1 | P_1 | dP_2 = \text{Tr} \left( V_{P_1} dP_2 \right), \quad (5.29)$$

which can be expressed in more direct way as

$$V^{P_1}_{\alpha \lambda} = \int \phi_\kappa(r) \left[ \int \phi_\mu^{(r')} P^\mu_r P^{\nu'}_r \phi_\nu(r') d\tau + v^{P_1}_{xc}(r) \right] \phi_\lambda(r) dr$$

$$+ \frac{1}{2} \int v^{P_1}_{xc}(\tau(r)) \nabla \phi_\kappa(r) \cdot \nabla \phi_\lambda(r) dr, \quad (5.30)$$

where $v^{P_1}_{xc}(r)$ and $v^{P_1}_{xc}(\tau(r))$ are the second derivative of $E_{XC}$ as defined in the section A.4.

Thus finally we have

$$\frac{\partial E}{\partial C \partial C} dC_1 dC_2 = \text{Tr} \left( dC_1^\dagger (I - SP) \right)$$

$$- S dC_1 S^{-1} C^\dagger HCS^{-1} + H(I - PS) dC_1 S^{-1} C_2^\dagger + V_{P_1} C S^{-1} + c.c. \right) \quad (5.31)$$

$$+ c.c. \right) \quad (5.32)$$

$$\right) \quad (5.33)$$

where $P_1$ depends on $dC_1$. 

5.3.2 Equation to solve

Taking the previous equations together the equations to solve are

\[
\text{Tr} \left( dC_2^\dagger (I - SP) \right) \\
- SdC_1[C_1]\tilde{S}^{-1}C_1^\dagger HCS^{-1} + H(I - SS)dC_1[C_1]\tilde{S}^{-1} + V_P C\tilde{S}^{-1} \\
\left] + \text{c.c.} \right) = -\text{Tr} \left( dC_2^\dagger (I - SP)HCS^{-1} + \text{c.c.} \right)
\]

for all the \( dC_2 \) in the tangential space, which can be simplified to

\[
(I - SP)[ \\
- SC_1\tilde{S}^{-1}C_1^\dagger HCS^{-1} + H(I - PS)C_1\tilde{S}^{-1} + V_P C\tilde{S}^{-1} \\
\left] = -(I - SP)HCS^{-1}
\]

which, apart the \( S \) and \( \tilde{S}^{-1} \) due to the non-orthogonality of both basis set and orbitals, is the expression found in (BSP01).

As can be seen in Fig. 5.1 this method can describe the H-bond very well. Unfortunately solving this system of equations is not substantially easier than finding the exact minimum of the energy with the orbital transformation method (see section 2.9.2).

Both methods find an optimal value for the parameter \( C_1 \) in the tangential space

\[
(C_1)^\dagger SC = 0
\]

but whereas with the perturbation you find the minimum of the approximate quadratic expansion of the energy, with OT you find the real minimum of the energy using a one to one mapping between the tangential space and the orthogonal subspaces.

As normally OT is not much more expensive than the second order expansion (actually it is more optimized) there is no reason to do more or less the same work to find an approximate solution.

5.4 Partial convergence and MD

If one assumes that the error in the forces is random, then one has a random force that produces a random walk in the velocities. The mean of the random force is 0, but actually the most probable thing is that the velocity grows with \( \sqrt{t} \), which means a linear drift
Figure 5.1: Second order expansion energy on water dimer by stretching the O-O distance, starting from the minimum energy Cs symmetric configuration. The energy of the frozen density configuration $e_0$ (+), the perturbatively corrected $e_{\text{perturb}}$ (x), and the full convergence energy $e_{\text{pot}}$ (*) are plotted. A very small systematic error is visible in the repulsive well, but the in the agreement in the important region is excellent. The minimum energy configuration along with the density correction is shown just above the graphs.

in the energy. If the force has a random noise with variance $\sigma^2$, then the mean square displacement of a component of $v$ after $n$ steps without external potential should be roughly

$$\sqrt{\langle (\Delta v)^2 \rangle} = \frac{\sigma \cdot \sqrt{n_{\text{steps}}}}{m},$$

where $m$ is the mass of the particle, and the drift in the total energy per atom should be

$$\text{drift}_E = \frac{1}{2} \sigma^2 n_{\text{steps}} \cdot 3 \frac{1}{n_{\text{at}}} \sum_i \frac{1}{m_i}.$$

Actually the error on the force is not truly random, and not all the drift is due to a random force, there are other factors like the accuracy of the integrator, etc. Note for example that the drift due to a random force gets worse with a smaller time step (more steps per picosecond), in reality the error on the force, especially at smaller time steps, cannot be considered random.
Anyway if the error on the forces is assumed to be $\pm 10^{-3}$ and to be uniformly distributed, one has $\sigma = 2 \cdot 10^{-3}/12$, and the drift for 64 water molecules with 0.5 fs time step should be

$$
\frac{1}{2} \left( \frac{2 \cdot 10^{-3}}{12} \right)^2 \cdot 2000 \cdot 3 \cdot \frac{1}{64} \left( \frac{128}{1822} + \frac{64}{16 \cdot 1822} \right) \approx 1.25 \cdot 10^{-7} \text{ au} \left( \frac{\text{au}}{\text{ps atom}} \right)
$$

which is more or less what we find in section 2.10.3 ($2 \cdot 10^{-7}$).

Normally the error on the forces is more or less the square root of the error energy, thus one has to converge the energy very well ($< 10^{-6}$ Hartrees) to have forces that are enough accurate to perform molecular dynamics ($< 10^{-3}$ au). The accuracy of DFT is around $10^{-3}$ Hartree, and one could converge just a little less than it if the forces would be enough accurate. The present method is simple enough to make all the calculation of the derivatives at any convergence level possible, which makes this method interesting again. In theory one could apply the same ideas also to OT starting from the same $C$, but the derivatives are more complex.

### 5.4.1 Non-convergence forces

The energy is a function of the wave functions coefficients $C$ and of the atomic positions $R$. Now for the ground state energy is

$$
E_g = \min_{\psi} E(C, R)
$$

but if a minimization routine $\hat{C}(C, R)$ is used and one knows the path of how $E$ is approximated then it is possible to define

$$
E_{nc} = E(\hat{C}(C, R), R)
$$

and consistent forces can be calculated:

$$
\frac{\partial E_{nc}}{\partial R} = \frac{\partial E}{\partial C} \frac{\partial \hat{C}}{\partial C} \frac{\partial C}{\partial R} + \frac{\partial E}{\partial R} \bigg|_{C=\hat{C}}.
$$

Actually someone could argue that as $\hat{C}$ uses a second order expansion of the energy we should use that expansion to calculate the energy. That might indeed be more consistent, but in fact it does not matter, the two energies expressions are by definition equal up to second order, and the whole procedure makes sense only if that approximation works well. Thus to spare us some cumbersome derivatives we use the full energy and not the second order expansion, this way $\partial E/\partial C$ and $\partial E/\partial R$ are already known.
5.4.2 Perturbation starting point: fragment calculations

In our method we have assumed that the system decomposes in fragments that can be calculated separately. Then from the solution of each fragment one might assemble the starting point $C$ for the perturbation. The resulting $C$ is blocked (nonzero only in fragment), but not orthonormal: for some $i, j \ (C_i)^\dagger S C_j \neq \delta_{ij}$. This happens when $i$ and $j$ are in different subsystems, even if frozen densities are used. The loss of orthonormality is a common problem, in subdivision methods, and normally one either ignores this error (as in many semi-empirical methods), or orthogonalizes the orbitals. As we have developed a method that can inherently accept nonorthogonal orbitals, we can use them as they are.

The energy expression for the fragments $\tilde{E}$ is a sum of energies of the single fragments, and does not need to be the same as the one for the full system: different functional, cutoff, cell, boundary conditions might be used. Along with $\tilde{E}$ we introduce the subsystem (block diagonal) ao-overlap matrix $\tilde{S}$, Kohn Sham matrix $\tilde{H}$ and density matrix $\tilde{P} = CC^\dagger$.

To calculate the non convergence forces one needs $\partial C/\partial R$. If one uses frozen densities as starting point $\partial C/\partial R$ becomes very simple, but in general it is difficult to calculate it efficiently, something that is problematic also for other approaches.

In this case as the energy of the subsystem $\tilde{E}$ is at the minimum with respect to the $C$ it is possible to find a method to calculate efficiently the $\partial C/\partial R$ derivative that is needed. In fact we do not really need $\partial C/\partial R$ alone, but rather

$$\frac{\partial E_{ac}}{\partial C} dC \frac{\partial C}{\partial R} = \text{Tr}(K_{cr}^\dagger \frac{\partial C}{\partial R} + \text{c.c.}), \quad (5.43)$$

where $K_{cr}$ is a $n_{ao} \times n_{ao}$ matrix (like $C$) that contains the coefficients that multiply $\partial C/\partial R$.

If we are at the minimum (wrt. to the fragment calculation)

$$\frac{\partial E}{\partial C} dC = 0 \quad \iff \quad 2(I - \tilde{S}\tilde{P})\tilde{H}C\tilde{S}^{-1} = 0 \quad (5.44)$$

and

$$\frac{\partial^2 E}{\partial C \partial R} + \frac{\partial^2 E}{\partial C \partial R} = 0, \quad (5.45)$$

from which we can extract the $\partial C/\partial R$ derivative

$$\text{Tr}(K_{cr}^\dagger \frac{\partial C}{\partial R} + \text{c.c.}) = \left(\frac{\partial^2 E}{\partial C \partial C}\right)^{-1} dC_1[K_{cr}]dC_2 \frac{\partial^2 E}{\partial C \partial R}, \quad (5.46)$$
In this equation the energy derivative can be calculated using

$$\frac{\partial P}{\partial C\partial R} dC = -(I - PS) dC S^{-1} C^\dagger \frac{\partial S}{\partial R} P - C S^{-1} dC^\dagger (I - SP) \frac{\partial S}{\partial R} P + \text{c.c.} \quad (5.47)$$

and taking into account that we are at the minimum (5.44), we have that

$$\frac{\partial E}{\partial C\partial R} dC = \text{Tr}(dC^\dagger (I - S\tilde{P})(\frac{\partial \tilde{H}}{\partial R} - \frac{\partial \tilde{S}}{\partial R} \tilde{P}H)CS^{-1} + \text{c.c.}), \quad (5.48)$$

where $\partial H/\partial R$ can be calculated using the Kohn-Sham operator $\tilde{H}$, for which $H_{\mu\nu} = \langle \phi_\mu | \tilde{H} | \phi_\nu \rangle$, as

$$\frac{\partial H}{\partial R} = \langle \nabla R \phi_\mu | \tilde{H} | \phi_\nu \rangle + \langle \phi_\mu | \nabla R | \phi_\nu \rangle + \langle \phi_\nu | \frac{\partial \tilde{H}}{\partial R} | \phi_\mu \rangle. \quad (5.49)$$

That are all derivatives that need already to be calculated. Note that you have to consider $\tilde{H}$ as function of the density $n$ and of $R$ when performing the derivative, and thus you have only a term coming from the pseudo potential which is symmetric to the one that comes from moving the density.

The inverse in (5.46) has to be calculated iteratively, and it is not acceptable to solve it again for each $R$ derivative. Fortunately it is possible to calculate the $dC_1$ (or $K_{cr}$) side first. $K_{cr}$ is not block diagonal, but looking at (5.46) you can see that the $dC_2$ part and the operator are in the subsystems (block diagonal), and thus only the block diagonal part of $K_{cr}$ is relevant.

The way the procedure is implemented the $n_{mo} \times n_{ao}$ matrixes are full matrixes distributed in a two dimensional periodic block-cyclic way (see section 2.6.5), and extracting sub-blocks is not efficient, and the fact that only block diagonal elements are needed is not used to perform calculations more efficiently. On the other side the sparse $n_{ao} \times n_{ao}$ matrixes have a better storage form and can be more easily transferred to the subsystems in parallel. Thus we want to transform

$$(-\frac{\partial^2 E}{\partial C^\dagger \partial C})^{-1} dC_1 [K_{cr}] dC_2 = \text{Tr}(K_{cr}^\dagger \frac{\partial^2 E}{\partial C^\dagger \partial C})^{-1} dC_2 + \text{c.c.} \quad (5.50)$$

so that a block diagonal $n_{ao} \times n_{ao}$ matrix appears. Using $I_{mo\times ao} = C^\dagger \tilde{S}C$ it can be done

$$(-\frac{\partial^2 E}{\partial C^\dagger \partial C})^{-1} dC_1 [K_{cr}] dC_2 = \text{Tr}(C^\dagger \tilde{S} \text{diag}(CK_{cr}^\dagger)(\frac{\partial^2 E}{\partial C^\dagger \partial C})^{-1} dC_2 + \text{c.c.}), \quad (5.51)$$

where $\text{diag}(CK_{cr}^\dagger)$ is the subsystem (block) diagonal part of $CK_{cr}^\dagger$. Now we can define the coefficients of the $\partial E/(\partial C\partial R)$ as

$$K_{ecr} = (-\frac{\partial^2 E}{\partial C^\dagger \partial C})^{-1} \text{diag}(K_{cr}^\dagger) \tilde{S}C, \quad (5.52)$$
which can be solved iteratively with high precision because each subsystem is solved by itself.

Then the $R$ derivatives can be calculated as

$$\text{Tr}(K_{\text{ecr}}^T(I - \hat{S}\hat{P})(\frac{\partial \hat{H}}{\partial R} - \frac{\partial \hat{S}}{\partial R} \hat{P}H)C + c.c.)$$

$$= \text{Tr} \left( \left( CK_{\text{ecr}}^T(I - \hat{S}\hat{P}) + c.c. \right) \frac{\partial \hat{H}}{\partial R} \right) + \text{Tr} \left( - (\hat{P}HC)K_{\text{ecr}}^T(I - \hat{S}\hat{P}) + c.c. \right) \frac{\partial \hat{S}}{\partial R} \right)$$

$$= \text{Tr}(K_{hr} \frac{\partial \hat{H}}{\partial R}) + \text{Tr}(K_{sr} \frac{\partial \hat{S}}{\partial R}) \quad (5.53)$$

where $K_{hr}$ and $K_{sr}$ are the coefficients of $\partial \hat{H}/\partial R$ and $\partial \hat{S}/\partial R$, and this trace can be efficiently calculated exactly as in the GPW method, i.e. transforming the derivative of the atom position in a gradient on the basis functions and summing up right away the contribution to the forces, without generating the $\partial \hat{H}/\partial R$ and $\partial \hat{S}/\partial R$ matrices.

### 5.5 Minimizer Derivatives

In this section the $n_{mo} \times n_{mo}$ matrixes $x, p, q, r, c, sc$ and $b$ are seen as $n_{mo}n_{mo}$ vectors, and so that $x^Tb = \text{Tr}(x^Tb)$.

To fully define the method we have to describe the minimization method $\hat{C}$ and its derivatives. In our case we have to solve the linear system of equations given by (5.35). These can and should be solved iteratively, because calculating the matrix explicitly is not efficient. As iterative solver we use the conjugated gradient algorithm with the kinetic energy as preconditioner. The simplicity of the algorithm and of the second order expansion make it possible to calculate the derivatives at each iteration.

#### 5.5.1 Conjugated gradient

The conjugated gradient method is an effective method to solve the linear system

$$Ax = b \quad (5.54)$$

for symmetric positive definite $A$, in our case

$$A = \frac{\partial^2 E}{\partial C \partial C} \quad b = -\frac{\partial E}{\partial C} \quad (5.55)$$
where both are evaluated at C.

The method proceeds by generating vector sequences $x^i$ of successive approximations to the solution, residuals $r^i$ corresponding to them, and search directions $p^i$. The length of these sequences can become large, but only a small number of vectors needs to be kept in memory. The use of a preconditioner $M^{-1}$ that reduces the condition number of $A$ (the ratio between the largest and smallest eigenvalue) or at least compresses all the eigenvalues but the extremal ones, can greatly enhance the convergence of the algorithm.

The conjugated gradient algorithm with preconditioner $M^{-1}$ is defined as

$$
\begin{align*}
  r^0 &= b - Ax^0 \\
  p^0 &= 0 \\
  \text{do i = 1..} & \\
  & z^{i-1} = M^{-1}r^{i-1} \\
  & \text{if } p^{i-1} = 0 \text{ then } \\
  & p^i = z^{i-1} \\
  & \text{else } \\
  & \beta_{i-1} = <r^{i-1}, z^{i-1}> / <r^{i-2}, z^{i-2}> \\
  & p^i = z^{i-1} + \beta_{i-1} p^{i-1} \\
  & \text{end if } \\
  & q^i = Ap^i \\
  & \alpha_i = <r^{i-1}, z^{i-1}> / <p^i, q^i> \\
  & x^i = x^{i-1} + \alpha_i p^i \\
  & r^i = r^{i-1} - \alpha_i q^i
\end{align*}
$$

where $p$ is the unnormalized search direction, and $r$ updated just to spare computation ($r^i = b - Ax^i$). This version, which is the one you find normally the literature (BBC+94) has been optimized to reduce as much as possible the scalar products, and numerical error, but as result the underlying mathematical structure is less clear.

For example, as one would expect, for $p$ just the direction, not the length is important, even if this is not true in the formulation of the algorithm just presented. The correction

$$
\hat{c}^i = \alpha_i p^i
$$

(5.56)
for the solution $x$ if $\alpha \neq 0$ has the same direction as $p$. So if for the moment, to simplify the discussion, we ignore the rare case $\alpha = 0$, one can to use the $c^{i-1}$ instead of the $p^{i-1}$ when generating the new search direction. In this case one can introduce a procedure $F$ that produces a new correction $c_{new}$ from the old ones.

### 5.5.2 CG step function

To define the function $F(R, C, x, c)$ (atom positions, wave functions coefficients, solution, last correction) that produces the new correction $c_{new}$, the formula to calculate the $p^i$ has to be changed. In the conjugated gradient the directions $p^i$ are constructed in such a way that $(p^i)^T A p^{i-1} = 0$. Actually this implies orthogonality also to all the previous $A p^i$. Using this knowledge an equivalent method to construct the new $p^i$ is

$$p^i = M^{-1} r^{i-1} - \frac{(A p^{i-1})^T M^{-1} r^{i-1}}{(p^{i-1})^T A p^{i-1}} p^{i-1}$$

which is invariant with respect to a rescaling of $p^{i-1}$, and thus also $c^{i-1} = \alpha_i p^{i-1}$ can be used.

For $\alpha_i$ one can use that $A p^{i-1}$ and $p^i$ are orthogonal to simplify $(p^i)^T A p^i$ to $(p^i)^T A M^{-1} r^{i-1}$, and making as added bonus $\alpha_i p^i$ invariant with respect to rescaling of $p^i$.

Thus $F$ can be expressed as

$$F(R, C, x, c) = \{ r = b - A x$$

if $c = 0$ then
$$\beta = 0$$
else
$$\beta = \frac{c^T A M^{-1} r}{c^T A c}$$
end if
$$p = M^{-1} r + \beta c$$
$$\alpha = \frac{r^T M^{-1} r}{p^T A M^{-1} r}$$
return $\alpha p$  }
Using $F$ the algorithm can be expressed simply as
\begin{equation}
\begin{aligned}
c^0 &= 0 \\
do i &= 1..n_{\text{steps}} \\
c^i &= F(R,C, \sum_{j=0}^{i-1} c^j, c^{i-1}) \\
\end{aligned}
\end{equation}

\subsection{5.5.3 Direct derivative accumulation}

Using the previous version of the algorithm it is possible to calculate $\delta c^i$, the $\partial c^i / \partial R$ derivative, iteratively starting with $c^0 = 0$ and keeping just the last derivative and the sum of all the derivatives so far, as can be done with the $c^i$ in (5.59). Thus an optimized algorithm that calculates also the derivatives would be
\begin{equation}
\begin{aligned}
c^0 &= sc = \delta c^0 = \delta sc^0 = 0 \\
do i &= 1..n_{\text{steps}} \\
\delta c^i &= \frac{\partial F}{\partial R}(R,C,sc^{i-1},c^{i-1}) + \frac{\partial F}{\partial C}(R,C,sc^{i-1},c^{i-1})\frac{\partial C}{\partial R} \\
&\quad + \frac{\partial F}{\partial x}(R,C,sc^{i-1},c^{i-1})sc^{i-1} + \frac{\partial F}{\partial c}(R,C,sc^{i-1},c^{i-1})\delta c^{i-1} \\
c^i &= F(R,C,sc^{i-1},c^{i-1}) \\
\delta sc^i &= \delta sc^{i-1} + \delta c^i \\
sc^i &= sc^{i-1} + c^i \\
\end{aligned}
\end{equation}

Unfortunately this algorithm cannot be used in our case because for efficiency reasons one should calculate the $R$ derivatives as little as possible, and thus $\delta c_{\text{last}}$ and $\delta sc$ have suspended computations in them, and are not just one object.

The derivatives that appear in the minimizer can be brought to the from
\begin{equation}
\begin{aligned}
\text{Tr}(W_{sr} \frac{\partial S}{\partial R}) \quad &\text{Tr}(W_{hr} \frac{\partial H}{\partial R}) \quad \text{Tr}(W_{cr} \frac{\partial C}{\partial R}) \quad \text{Tr}(W_{mr} \frac{\partial M^{-1}}{\partial R}) \\
\end{aligned}
\end{equation}

plus something like
\begin{equation}
\begin{aligned}
\text{Tr}(W_{sr} \frac{\partial V_{\text{grid}}^{PI}}{\partial R}) \\
\text{Tr}(P^I \frac{\partial V_{\text{grid}}^{W_{er}}}{\partial R}) \\
\end{aligned}
\end{equation}
the exact structure of this last term is explained later and in appendix A.5. Here $W_{sr}, W_{hr}, W_{vr}, W_{cr}, W_{mr}$ are some coefficient matrixes. We use these matrixes and not the coefficients of $\partial A/\partial R$, because $A$ is a huge matrix $(n_{ao}n_{mo} \times n_{ao}n_{mo})$ that we do not want to store explicitly, so clearly this is the case also for its coefficient matrix. The problem is that to calculate the $W_*$ one should also know where the derivative of $sc$ will be used. This can be taken from (5.42), but there is a catch, the place where to evaluate the derivatives is not known until at the end. One might think that using the second order expansion would avoid this problem, but it is not so, everything is linear, but one still has terms like

$$\frac{\partial^2 E}{\partial C \partial C} \delta c_1[c^{i+1}] \delta c_2[\delta c^i]$$

(5.62)

that cannot be evaluated efficiently before $c^{i+1}$ is known.

This is so because in each $\delta c^i$ there are derivatives like

$$\sum_{j=1}^i \gamma_j (M^{-1} \frac{\partial^2 E}{\partial C \partial C}) \frac{\partial C}{\partial R}$$

(5.63)

that are non computable until the end because both $\frac{\partial^2 E}{\partial C \partial C}$ and $\frac{\partial C}{\partial R}$ can be efficiently evaluated only if the left hand side is known. Thus to calculate the $\delta c^i$ term one would have to calculate $O(i)$ times $M^{-1} \frac{\partial^2 E}{\partial C \partial C}$, making the algorithm quadratic in the number of steps of conjugated gradient.

### 5.5.4 Inverse CG step

The solution to the problems of the direct CG derivative accumulation is to first calculate $sc^{n\text{steps}}$ an then calculate the derivative going backwards from the last $c^i$ to the first. To do this one can use the function

$$G(R, C, x, c) = \{ \quad z_{old} = M^{-1}(b - A(x - c)) \quad \} \quad \text{p}_{old} = c - c^T Ac z_{old}^{old} \quad \text{r}_0 = b - Ax^0 \quad \gamma = \frac{r_0^T \text{p}_{old}}{\text{p}_{old}^T A \text{p}_{old}} \quad \text{return } \gamma \text{p}_{old} \quad \}$$

that given a partial solution and last correction, steps back and finds the previous correction. In it $p_{\text{old}}$ is extracted from c using the definition of c and the fact that $p_{\text{old}}^T Ac = 0$. 
Then $p_{odd}$ is rescaled using the knowledge that $Ac_{odd}$ removes the component in the direction $Ap_{odd}$ of the residual $r_0$ using $A^{-1}$ as scalar product. The whole procedure will fail if $c$ is $0$.

With this procedure once we can go back the conjugated gradient with a cost in time and storage equal to the one of the forward iteration.

1. get $c^{n\text{steps}}$ and $s c^{n\text{steps}} = \sum_{j=1}^{n\text{steps}} c^j$ from the forward CG solution
2. do $i = n_{\text{step}}, 1$
   - (a) $c^{i-1} = G(R, C, s c^i, c^i)$
   - (b) $s c^{i-1} = s c^i - c^i$

### 5.5.5 Inverse derivative accumulation

In this iteration one can calculate the derivatives of $c^i$ by assuming that they have the form given by (5.60) plus two terms like

$$\text{Tr}(W_{c^{i-1}} \frac{\partial c^{i-1}}{\partial R}), \quad \text{Tr}(W_{sc^{i-1}} \frac{\partial s c^{i-1}}{\partial R}).$$

(5.65)

In each iteration one can reduce these two terms to the others plus terms in $\partial c^{i-2}/\partial R$ and $\partial s c^{i-2}/\partial R$. Basically the algorithm can be seen as an application of the chain rule starting with $c^i$ and $s c^i$ and going back until $c^0 = \delta c^0 = sc^0 = \delta sc^0 = 0$.

The forces due to all the $W_s$, excluding $W_{r_{i-1}}$ and $W_{s_{i-1}}$, can be calculated immediately, the choice to accumulate them is just a computational convenience. In fact one has to calculate immediately the forces due to the $W_{r_i}$, because those forces (for efficiency reasons) depend both on the right hand side and on the left hand side of the coefficients of $\delta A$, and cannot be accumulated efficiently. The details are explained in the appendix A.5.

So the algorithm to collect the derivatives can be written as

1. get $c^{n\text{steps}}$ and $s c^{n\text{steps}} = \sum_{j=1}^{n\text{steps}} c^j$ from forward CG solution
2. $W_{hR} = W_{sr} = W_{cr} = W_{cR} = W_{mr} = W_{ci} = 0$
3. \( W_{sc^i} = \frac{\partial E}{\partial C} \bigg|_{C=C+sc^{i}\text{steps}} \), which is the left hand side for the force evaluation of the \( sc \) term.

4. do \( i = n_{\text{step}} \cdot 1 \)
   
   (a) \( c^{i-1} = G(R, C, sc^i, c^i) \)
   
   (b) \( sc^{i-1} = sc^i - c^i \)
   
   (c) expand

   \[
   \delta c_i = \frac{\partial F}{\partial R} (R, C, sc^{i-1}, c^{i-1}) + \frac{\partial F}{\partial C} (R, C, sc^{i-1}, c^{i-1}) \frac{\partial C}{\partial R} + \frac{\partial F}{\partial x} (R, C, sc^{i-1}, c^{i-1}) \delta sc^{i-1} + \frac{\partial F}{\partial c} (R, C, sc^{i-1}, c^{i-1}) \delta c^{i-1}
   \]

   into the \( W_{i} \), and with it transfer the derivatives in \( c^i \) and \( sc^i \) in derivatives in \( c^{i-1} \) and \( sc^{i-1} \).

5. set the \( c^0 \) and \( sc^0 \) derivatives to 0

   After this algorithm one has the matrixes \( W_{hr}, W_{sr}, W_{cr}, W_{vr}, W_{mr} \) with which to evaluate the component of the force dependent on \( sc \).

### 5.5.6 \( F \) derivatives

To really be able to perform the reduction of the \( c^i \) one needs to know the reduction explicitly.

The \( R \) derivative is

\[
\text{Tr}(W \frac{\partial F}{\partial R}) = \text{Tr}(W \frac{\partial F}{\partial A} dA \frac{\partial A}{\partial R}) + \text{Tr}(W \frac{\partial F}{\partial b} db \frac{\partial b}{\partial R}) + \text{Tr}(W \frac{\partial F}{\partial M^{-1}} dM^{-1} \frac{\partial M^{-1}}{\partial R})
\]

(5.66)

the \( C \) derivative is

\[
\text{Tr}(W \frac{\partial F}{\partial C}) = \text{Tr}(W \frac{\partial F}{\partial A} dA \frac{\partial A}{\partial C}) + \text{Tr}(W \frac{\partial F}{\partial b} db \frac{\partial b}{\partial C}) + \text{Tr}(W \frac{\partial F}{\partial M^{-1}} dM^{-1} \frac{\partial M^{-1}}{\partial C})
\]

(5.67)

and then there are still

\[
\text{Tr}(W \frac{\partial F}{\partial sc^i})
\]

(5.68)
and

\[ \text{Tr}(W \frac{\partial F}{\partial \alpha}) \]  \hspace{1cm} \text{(5.69)}

The complete expansion of the formulas is in the appendix A.5

### 5.5.7 Numerical stability

The normal conjugated gradient algorithm is very stable, and robust. Actually it is guaranteed that it will find the correct solution if it will iterate long enough. The routine to compute the derivatives is not so stable with very high iteration numbers. This is so because numerical errors accumulate and it is not guaranteed that exactly the same search directions will be constructed. For normal iteration numbers (∼10) the error from my numerical tests remains very well controlled (∼10^{-13}) even for rather ill conditioned problems.

In those tests if convergence is achieved before \( n_{\text{steps}} \), the procedure was stopped earlier, and the derivatives calculated with that reduced number of steps. This should be done in each case, even if the algorithm description I did not explicitly said so.

A problem, as discussed before arises if \( \alpha = 0 \), and we are still far away from convergence. In that rare case even a brute force re-solution of the conjugated gradient up to the problematic iteration would be acceptable.

### 5.6 The big picture

Putting everything together the method to calculate energy and forces of a weakly interacting system would be

1. solve each fragment exactly and find the starting \( C \)
2. transfer \( C \) from the fragments to the whole interacting system
3. calculate \( e_0 \), the frozen density energy
4. solve (5.35) approximately by doing just \( n_{\text{step}} \) steps of conjugated gradient.
5. step the conjugated gradient backward accumulating the derivative coefficients $W_{hr}, W_{hr}, W_{er}, W_{mr}$

6. add the $e0$ forces to the $W_*$

7. calculate $Tr(W_{er} \partial C/\partial R)$ in the subsystems by solving the perturbation equations exactly

8. calculate the other $W_*$ terms just as with the normal force evaluation

This algorithm builds upon the strengths of the perturbation approach: the ability to calculate all the derivatives rather easily. Thanks to this it is possible to compute the dependence of the wave functions on the position.

To find the minimum of the energy to second order one has to solve a linear system of equations. These can be solved iteratively (because calculating the matrix explicitly is not efficient) using the conjugated gradient algorithm.

The easy structure of the algorithm makes it possible to calculate the derivative at each optimization level, and thus to have consistent forces without full convergence.

## 5.7 Conclusion and Outlook

In this chapter a perturbation approach with a very intuitive representation was developed. Derivatives, and a notation to express them are calculated in a fully non-orthogonal setting.

This method could be very useful for analysis, as one obtains the perturbation with respect to a well defined state without extra computational effort. One could also think to build an empirical polarizable force field with a structure similar to the taylor expansion we developed. Such a method would have the orthogonality constraints of the orbitals automatically embedded in its formulation. This implies the need of a non sparse $C^* = CS^{-1}$ or some approximation of it. Thus the simplifications that one should do to have an efficient yet accurate representation are not obvious and more research is needed, but this could offer a consistent way to take into account the Pauli repulsion.
A possible application of this to weakly interacting systems was presented. For this application the non convergence forces are calculated. It is shown how to calculate the $\partial C/\partial \mathbf{R}$ term in fragments in an efficient way. Furthermore it is shown how coherent derivatives can be kept in a conjugated gradient optimization with a moderate overhead, also when the derivatives are partially suspended.

The perturbation approach presented in this chapter has many nice properties, and I think a future, but if used to accelerate the calculation, and not for analysis or forcefield creation, perturbation based methods will probably always be inferior to OT, unless one takes advantage of the simpler structure of the perturbation equations. This because OT minimizes the full energy on the tangential space, not an approximate second order expansion.

The algorithm presented here tries take advantage of the formally simpler structure of the perturbation equation. This algorithm should have a (not very big) advantage on OT, and lets one test all the derivatives and look at convergence characteristics. Later one could think about more aggressive optimizations, for example enforcing of sparsity going in the direction MI-SCF. Another future search direction, that also can take advantage of the simple structure of the perturbation equations, might be to try to directly evolve of the wave functions (direction Car-Parrinello). With a direct propagation of the wave functions one is not anymore interested in having the exact solution at each time step, but only to propagate it in such a way that the error stays bounded. In that case it will probably be better to keep into account the second order correction due to the curvature of the space of (5.24) that we have neglected, at least a posteriori.

It would also be interesting to see how far the same procedure to keep into account the non convergence could be applied to the OT method.
Chapter 6

Conclusion

Ab-initio molecular dynamics is a powerful method to study the properties of molecular systems. It can be used to explore states where not much is known, check assumptions, as starting point for more empirical theories. Unfortunately ab-initio molecular dynamics is computationally costly, and so not applicable to all the systems one would like. In this thesis we have presented some solutions that are practically realized in the openly available, and actively developed cp2k program (CP204) to apply this method to systems that are of common interest.

GPW

DFT, and the GPW method are presented in detail, along with the reasons their efficiency. The $E^{xc}$ functionals, are analyzed, their evaluation is discussed, and some partial solutions to reduce the ripples of the energy functional, inspired by the new multi grid spline methods of section 2.7, are proposed. A better solution is to change the pseudo potential, or go to an all electron approach like the newly implemented GAPW where the density does not approach 0 at the core of the atoms.

The newly developed spline based multi grid interpolation showed very good characteristics especially without periodic boundary conditions.

The parallelization strategy, and the algorithms developed for it, like the new sparse-matrix full-matrix multiplication are described, as well as more technical aspects like the
input, error handling, and algorithm environments.

Then the method is validated and its performance checked, and it is shown that it represent a robust and efficient method to apply to real systems.

**Ab-initio Liquid Water**

As an example of real world application ab-initio liquid water was studied. We review of a comparison between various sampling methods and perform extensive tests on temperature and functional dependence of Liquid water.

The uncertainty on temperature along with the sensitivity of water to temperature changes can explain the spread of values found in the literature.

Results seem to confirm that BLYP water is over-structured, and that almost all functionals are sluggish diffusive. Quantum effects of the ion motion (that are known to be important in water) could explain the difference.

Unfortunately functional dependence turns out to be rather large which shows that DFT is not yet a black-box method, as any other method, a check of the applicability of the method to the system, and sensitivity of results on the various approximations still need to be performed, DFT is no silver bullet.

**QM/MM**

To simulate bigger systems and to take into account the screening or periodicity effects due to the environment often the hybrid QM/MM method is a valid solution. A novel method to calculate the coulomb interaction in the QM/MM between the MM and QM is explained and validated. This method that uses an expansion in Gaussian of the coulomb potential, is both accurate and fast.

The method presented can be extended with a fully consistent periodic treatment that does not need to describe the QM subsystem at MM level, unlike many other energy difference based methods. For the periodic extension a new decoupling schema to treat the electrostatic of the QM part is proposed.
The breaking of covalent bonds, that was not the focus of our work, is not solved in a really satisfactory way in our method, probably a pseudo potential approach would be a better solution for it.

**Perturbation approach**

Finally an approach based on a second order expansion of the energy in wave functions is studied. This approach is a very attractive and natural way to treat systems that decompose in fragments (molecules), and can be useful for analysis purposes, and maybe empirical potential fitting.

The theory is developed in a fully non orthogonal setting, and some possible applications are highlighted, in particular the possibility of applying it to weakly interacting systems. In this case non convergence forces can be calculated, and enabling one to do molecular dynamics with partially converged wave function.

For this purpose a method to calculate the derivatives in a conjugated gradient procedure, even if some of these derivatives will be calculated only at the end is developed, along with a method to calculate the $\partial C/\partial R$ derivatives at a cost comparable with a force evaluation.

Optimizations connected to sparsity enforcement could be applied to this perturbation approach. This approach could also be the basis for direct wave function propagation.
Chapter 7

Acknowledgments

In such a long work there are always a lot of people that directly or indirectly help you, and it is difficult to thank them all. First and foremost I would like to thank Michele Parrinello for having given me the chance to do this thesis, for his helpful comments and reviews, and for his example as man of science that I deeply respect.

Joost VandeVondele not only for his work on cp2k, and on the water simulations that I presented in 3.2, but also for his enthusiasm, friendship, and for having introduced me to climbing.

Then I would like to thank a lot Teodoro Laino for his work on QM/MM, huge help in the thankless task of transitioning from the old input parser to the new one, and his friendship.

Juerg Hutter for stimulating discussions, his support, and the support of CP2K program and team. Matthias Krack for useful discussions, and his support. Victoria Buch for her friendship, and for having introduced me to the, still mysterious to me, world of protonated water, and acid clusters.

Chris Mundy and Will Kuo for their development of the classical part of CP2K and precious help with QM/MM.

The whole group of Michele Parrinello, for the very nice time and stimulating discussion, in particular Alessandro Laio for insightful discussions on QM/MM, wonderful hikes and his friendship. Beautiful hikes and climbs that I also did with Francesco Gervasio
and Davide Donadio. Then Thomas Kühne, for the new enthusiasm brought in our group on cp2k, Daniele Passerone, and all the other members of the group.

A thank also to all the other fellow PhD. students: Federica Trudu, Gabriele Petraglio, Davide Branduardi, Paolo Elvati and Daniel Aktah.

Last but not least a thank to my family, in particular my mother Elena Mohamed, for all the support, and to my brothers Semir Mohamed and Akram Mohamed. Finally a big thank also to my wife Laura Pecchioli for all her support and understanding.
List of Figures

2.1 Electrostatic energy convergence ........................................ 17

2.2 Behavior of $n$ and $v^{xc}$ close to the core of an O atom .......... 21

2.3 Error in energy and forces due to $E^{xc}$ wiggles with various xc computation methods ......................................................... 24

2.4 Scaling of KS matrix construction ........................................ 28

2.5 The $N^3$ function ............................................................. 41

2.6 Parametric cubic splines weights close to the border and $x(u)$ .......... 45

2.7 The border functions $M_{0.2}$ as function of $x$ ......................... 46

2.8 Energy conservation of 64 H$_2$O MD .................................... 59

2.9 Wave functions extrapolation efficiency .................................. 60

2.10 H velocity-velocity autocorrelation function of 64 H$_2$O .......... 61

2.11 QS vs CPMD forces ......................................................... 63

2.12 Liquid Water binding energy (BSSE) .................................... 65

2.13 IBM SP4 parallel scaling (diagonalization and orbital transformation) . . . 69

2.14 PC-Cluster parallel scaling (diagonalization and orbital transformation) . . 70

3.1 T vs. $E_{tot}$ for BLYP water .............................................. 81
3.2 Temperature dependence of the first peak of the O-O pair correlation function in BLYP water ........................................... 82
3.3 O-O pair correlation functions, corresponding to simulations of BLYP water at different temperatures .......................... 83
3.4 Slopes from the MSD curves obtained for the BLYP water simulations ... 84
3.5 O-O pair correlation functions, size dependence, 64 vs 32 water molecules. 86
3.6 BLYP water mean square displacements ................................... 87
3.7 Functional dependence of mean square displacement in water ........... 89
3.8 Functional dependence of O-O pair correlation functions in water ...... 90
4.1 Gaussian expansion of the electrostatic potential (GEEP) in real space. . . 101
4.2 Fourier transform of $V^{MM}$ and Fourier transform of the residual function $R_{\text{low}}$. ................................................................. 102
4.3 Test of QM/MM forces, QS vs CMPD. ........................................... 105
4.4 QM/MM energy conservation during an MD run of 3H$_2$O. ................. 106
4.5 QM/MM vs MM, water H-O and O-O pair correlation functions. ........ 109
4.6 QM/MM vs MM, $H_{\text{pept}} - O_{\text{wat}}$ and $O_{\text{pept}} - H_{\text{wat}}$ pair correlation functions for the ALA-GLY dipeptide in SPC water. ......................... 110
5.1 Perturbation approach, water dimer H-bond. ............................ 123
List of Tables

2.1 Scaling of orbital transform and diagonalization wrt. basis set 55
2.2 H$_2$O benchmark system characteristics 67
2.3 Occupation of the overlap matrix 67
3.1 Basis Set convergence test for OLYP Liquid Water 80
3.2 Functional dependence in Liquid Water 88
4.1 QM/MM-GEEP, Optimal coefficients of the Gaussian expansion 100
Bibliography


[APK03] D. Asthagiri, L. R. Pratt, and J. D. Kress, Free energy of liquid water on the basis of quasichemical theory and ab initio molecular dynamics, Phys. Rev. E 68 (2003), 041505. 72, 76, 91


[LHK99] P. D. Lyne, M. Hodoscek, and M. Karplus, A hybrid qm-mm potential employing
hartree-fock or density functional methods in the quantum region, J. Phys.
Chem. A 103 (1999), 3462–3471. 94

[LHP97] G. Lippert, J. Hutter, and M. Parrinello, A hybrid Gaussian and plane wave
12, 78

[LHP99] The Gaussian and augmented-plane-wave density functional method
for ab initio molecular dynamics simulations, Theor. Chem. Acc. 103 (1999),
no. 2, 124–140. 12, 23, 38

[LLL03] M. J. Loferer, H. H. Loeffler, and K. R. Liedl, A qm-mm interface between
charmm and turbomole: Implementation and application to systems in bulk
1249. 94

[LMLP05] Teodoro Laino, Fawzi Mohamed, Alessandro Laio, and Michele Parrinello,
An efficient real space multigrid qm/mm electrostatic coupling, J. Chem. Th.
Comput. 1 (2005), 1176–1184. 99

[LMLP06] T. Laino, F. Mohamed, A. Laio, and M. Parrinello, An efficient linear-scaling
electrostatic coupling for treating periodic boundary conditions in qm/mm

[LP85] M. Levy and J. P. Perdew, Hellmann-feynman, virial and scaling requisite for
the exact universal density functional. shape of the correlation potential and

[LSPC93] K. Laasonen, M. Sprik, M. Parrinello, and R. J. Car, “ab initio” liquid water,

coupling scheme for hybrid carparrinello molecular dynamics simulations,

[LYP88] C. T. Lee, W. T. Yang, and R. G. Parr, Development of the colle-
salvetti correlation-energy formula into a functional of the electron-density,


[Ver67] L. Verlet, Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules, Phys. Rev. 159 (1967), 98. 56


Appendix A

A.1 Differential Geometry

A very basic exposition of the ideas of differential geometry that are useful to us. No formal definition or explanation on how to calculate is presented, see (Str84) for a very compact formal definition, and (Spi76) for a big review of Differential Geometry (assumes many things as known).

Differential geometry can be used to describe differentiable functions on manifolds. A manifold is a metric space that is locally isomorphic to $\mathbb{R}^n$. In numerical computations the metric can be important because it gives the concept of distance, and in minimizing the error one wants that the convergence is uniform in all directions. For example for two subspaces $V_1$ and $V_2$ a possible measure of distance is the maximum of the minimum angle between a $v \in V_1$ and a $w \in V_2$. Actually as shown in (AES98) the Euclidean metric induces the Euclidean metric on the Grassman manifold. This is a good metric, and there is no real need to change it, so won’t discuss metrics further.

The differential of a function can be seen a local linear approximation of the function at a given point $p$. To have a linear approximation, one needs a (linear) vector space at the point $p$. This is trivial for function from $\mathbb{R}^n$ (or $\mathbb{C}^n$) to $\mathbb{R}^m$ (or $\mathbb{C}^m$). In this case $\mathbb{R}^n$ and $\mathbb{R}^m$ can be seen as a vector spaces between which defining a linear function. On a manifold in general the structure of vector space is not given, but at every point $p$ a vector space $T_p$ called tangential space can be constructed. To describe linear functions
The dual space can be useful, and it has been given a special name: cotangential space.

In this case the differential of a function \( f : M \to N \) at a point \( p \in M \) can be seen as linear function between \( T_p \) and \( T_{f(p)} \) (we will look only at cases where \( N = \mathbb{R}^m \), and thus the tangential space of \( N \) is always \( \mathbb{R}^m \), and does not depend on \( f(p) \)).

The tangential space can be a wholly abstract space, often it is constructed on the space of functions that are equal in some neighborhood of \( p \). What is needed is some connection between the points of \( T_p \) and the points of the manifold. Such a connection is given by geodesics. A geodesic is the shortest path between two points. With geodesic one can build a (locally) one to one relationship between the points of \( M \) and the vectors of \( T_p \): to each vector \( v \in T_p \) one associates the geodesic parameterized with constant velocity \( g(t) \) with \( g(0) = p \) and \( g'(0) = v \), and through it the point \( g(1) \) (i.e. the point in the direction of \( v \) at distance \( |v| \)).

Another thing that is often needed (even if we don’t need it explicitly) is a mean to compare two vectors of different tangential spaces \( T_p \) and \( T_q \) of the same manifold. To do this there is the parallel transport. This is a mean to map (transport) a vector \( v \in T_p \) to a vector \( \tilde{v} \in T_q \).

Now we can formulate the Newton algorithm in the differential geometry framework. Let \( f : M \to \mathbb{R} \) be a differentiable function

- Approximate the function \( f \) at the point \( p \) (or actually \( \hat{f} : T_p \to \mathbb{R} \), such that after the correspondence using geodesic \( \hat{f} = f \)), with a quadratic function \( \hat{f} : T_p \to \mathbb{R} \).
- find the \( v \in T_p \) that minimizes \( \hat{f} \)
- go to the point corresponding to \( v \) (using the geodesic) and, if the error is too big, repeat the procedure from there.

It should be clear that the approximation \( \hat{f} \) can be obtained with a Taylor expansion using the derivatives of \( f \) along a geodesic (i.e. using the derivatives of \( \hat{f} \)).

If a manifold \( M \) of dimension \( n \) is embedded in \( \mathbb{R}^m \) (i.e. it can be seen as sub manifold of \( \mathbb{R}^m \)), there is a nice representation of the tangential space \( T_p \): it can be seen as \( n \) dimensional hyper plane tangent to the surface defined by \( M \). If \( f \) can be extended in some way to the whole \( \mathbb{R}^m \), then the derivative of \( f \) in \( M \) is basically the restriction
of the derivative to the tangential space (i.e. the cotangential vector that describes this restriction).

## A.2 bra/ket notation

In the bra/ket notation $|\psi_i\rangle$ represents a wave function. This function is a (ket) vector in the infinitely dimensional Hilbert space of functions that are square integrable. The use of functions that are not square integrable on the whole space (like plane waves), or of distributions like the $\delta$ function need some care, and redefinition of the meaning of the various terms, but is feasible. We won’t formalize the exact meaning of a ket vector, and will be quite liberal in the way we use it ( sometime as distribution, sometime as function) following the physicists tradition.

Once one has defined the ket vector $|\psi_i\rangle$, a bra-vector $\langle\psi_i|$ is simply the complex conjugate of the function represented by $|\psi_i\rangle$. Bra and ket are denoted this way because one can define a scalar product between functions

$$\langle f, g \rangle = \int f(X)g(X)dX$$  \hspace{1cm} (A.1)

and a bra vector $\langle f |$ with a ket vector $ketg$ build a bracket $\langle f |g \rangle = \langle f, g \rangle$.

If instead of just one vector $|f\rangle$, one has a series of them $|f_i\rangle$ that build a basis it is possible to define the dual vector $\langle f_i^* |$. This is defined in such a way that

$$\langle f_i^* | f_j \rangle = \delta_{ij} \hspace{1cm} \langle f_i | f_j^* \rangle = \delta_{ij}$$  \hspace{1cm} (A.2)

For an orthonormalized basis $\langle f_i = \langle f_i^* |$ and $|f_i\rangle = |f_i^* \rangle$. Often the dual vector is also written as $\langle f^i |$. I did not use this convention because there are indices (like the electron number) that are neither covariant nor contravariant, and I found it confusing to mix them, the rule that you always have to contract between "up" indices and "down" down is not valid anymore.

A linear operators $L$ can be applied on $|f\rangle$ to yield $|Lf\rangle$. The adjoint operator $L^\dagger$ is the operator such that

$$\forall \langle f, \forall \phi \rangle \hspace{1cm} \langle L^\dagger f | \phi \rangle = \langle f | L \phi \rangle.$$  \hspace{1cm} (A.3)

If the operator is self-adjoint ($L^\dagger = L$), as almost all the operators in quantum mechanics,
then
\[ \langle Lf|g \rangle = \langle f|Lg \rangle = \langle f|L|g \rangle. \] (A.4)

If \( f_i \) are a basis then one can represent the identity operator
\[ \infty = \sum_i |f_i\rangle \langle f_i^*| = \sum_i |f_i^*\rangle \langle f_i| \] (A.5)
and any other operator
\[ L = \sum_i |L f_i\rangle \langle f_i^*| = \sum_i |L f_i^*\rangle \langle f_i|. \] (A.6)

This is one of the reasons this notation is useful, it is naturally basis set independent.

If the \( f_i \) only define a subspace, one can define their dual \( f_i^* \) within that subspace. Any extension of the \( f_i \) that is orthogonal to all the \( f_i \) would have those dual vectors. At this point \( \sum_i |f_i\rangle \langle f_i^*| \) is not anymore the identity operator, but a projection. This is the case for the density matrix
\[ \rho = n_e \sum_i |\psi_i\rangle \langle \psi_i^*| \] (A.7)
where \( n_e \) is the number of electrons per orbital (1 or 2). Most of the time we assume that \( n_e = 1 \) so that \( \rho \) is a projection in the occupied subspace.

For any basis set \( f_i \)
\[ \sum_i \langle f_i^*|L|f_i \rangle = \sum_i \langle f_i|L|f_i^* \rangle = \text{Tr}(L) \] (A.8)

### A.3 \( \langle \psi_i | \) and \( | \psi_i \rangle \) derivatives

In these calculations I used the two linear independent \( \langle \psi_i | \) and \( | \psi_i \rangle \) derivatives as basis of the tangential space. Strictly speaking in the case we look at \( \psi \) can be taken as real, but I feel that working with them is more general, and makes some aspects of the mathematics behind it more easily visible. Furthermore it makes the finding of errors easier (the result has to satisfy more symmetries).

To understand why these derivatives come up, and their properties, it is very useful to look at the simplest possible case: the one of a complex function \( f \) of one variable
z = x + iy. The complex derivative \( \partial_z \) is a differential operator that can be defined with respect to \( x \) and \( y \) as

\[
\partial_z = \frac{\partial_x - i\partial_y}{2}
\]  

(A.9)

\( f \) can also be seen as function from \( \mathbb{R}^2 \rightarrow \mathbb{R}^2 \)

\[
f(z) = f_1(x + iy) + if_2(x + iy),
\]  

(A.10)

and its first derivative with respect to \( x \) and \( y \) is

\[
\begin{bmatrix}
\partial_x f_1 & \partial_x f_2 \\
\partial_y f_1 & \partial_y f_2
\end{bmatrix}.
\]  

(A.11)

As it can be seen the derivative are four real numbers, and they cannot, in general be reduced to just two real number (or a complex number). Complex analysis looks at a very special kind of functions, analytic functions (any algebraic function of \( z \) is analytic almost everywhere it is defined). For these functions the differential operator

\[
\partial_z = \frac{\partial_x + i\partial_y}{2}
\]  

(A.12)

is always zero (Cauchy-Riemann differential equation), a thing that has very deep consequences, but at the moment don’t interest us.

If one wants to look at a non-analytic function \( f \) like \( |z|^2 = zz \) then he cannot use only the \( z \) derivative, but he need also the \( \bar{z} \) derivative. Actually the case \( f(z) = |z|^2 \) is special because the result is real and so the two derivatives are one the complex conjugate of the other.

All what we said is still valid if we substitute \( \partial_z \) with \( \delta/\delta\psi \) and \( \partial_z \) with \( \delta/\delta\bar{\psi} \), in fact basically all the properties of \( \partial_z \) and \( \partial_z \) carry over (there is a small difference, I assumed that \( \delta/\delta\bar{\psi} \) give back already the differentials (like \( dz\partial_z \)).

A.3.1 Energy Derivatives
To support fractional occupation we introduce also

\[
\overrightarrow{\rho} = \sum_i n_i |\psi_i\rangle \langle \psi_i^*| = \sum_{i,j} n_i |\psi_i\rangle S_{ij}^{-1} \langle \psi_j|
\]

\[
\overrightarrow{\rho} = \sum_i |\psi_i^*\rangle \langle \psi_i| n_i = \sum_{i,j} |\psi_j\rangle S_{ji}^{-1} \langle \psi_i| n_i
\]

\[
\bar{\rho} = \frac{\rho + \overrightarrow{\rho}}{2}
\]

(\text{A.13}) \quad (\text{A.14}) \quad (\text{A.15})

\(\overrightarrow{\rho}\) and \(\overrightarrow{\rho}\) are almost projections (apart squaring the \(n_i\), and \(\bar{\rho}\) is their symmetric part.

The way I introduce fractional occupation (it isn’t smearing yet) is quite unorthodox, (the density is not necessarily simply a sum of scaled orbitals), but it has the right number of electrons and in the minimum the orbitals become orthogonal, and one gets the expected result (or if the \(n_i\) are all equal).

The energy functional can be written as

\[
E = -\frac{1}{4} \sum_i \int (\langle \psi_i^* | \mathbf{r} \rangle \nabla^2 | \mathbf{r} \rangle | \psi_i \rangle n_i + n_i \langle \psi_i | \mathbf{r} \rangle \nabla^2 (| \mathbf{r} \rangle | \psi_i^* \rangle n_i) \rangle d\mathbf{r}
\]

\[
+ \frac{1}{2} \int \frac{\langle \mathbf{r} | \bar{\rho} | \mathbf{r}' \rangle \langle \mathbf{r}' | \bar{\rho} | \mathbf{r}' \rangle }{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r} d\mathbf{r}'
\]

\[
+ E_{xc} + \int v_n(\mathbf{r}) \langle \mathbf{r} | \bar{\rho} | \mathbf{r} \rangle d\mathbf{r},
\]

(\text{A.16})

If we call \(V\) the subspace spanned by the \(\psi_i\), and

\[
\pi = \pi_V = \Pi - \rho
\]

(\text{A.17})

is the projection in \(V^\perp\). Using the following shorthands

\[
\overrightarrow{\pi}_i = n_i \Pi - \overrightarrow{\rho}
\]

\[
\langle \psi_i^* | = \sum_j S_{ij}^{-1} \langle \psi_j | n_j
\]

\[
\bar{\pi}_i = n_i \Pi - \bar{\rho}
\]

\[
\langle \psi_i^* | = \sum_j n_j |\psi_j\rangle S_{ji}^{-1}
\]

(\text{A.18}) \quad (\text{A.19})
we can write the derivatives we need as follows:

\[
\partial_{\psi_{i}}E\langle d\psi_{i}\rangle = \partial_{\rho}E\rho\left[ \frac{1}{2}\left( |\psi_{i}^{*}\rangle \langle d\psi_{i}^{*}| \vec{\rho} + |\psi_{i}^{*}\rangle \langle d\psi_{i}^{*}| \pi \right) \right] \quad (A.20)
\]

\[
\partial_{\psi_{m}}\partial_{\psi_{i}}E\langle d\psi_{i}\rangle = \partial_{\rho}E\rho\left[ \frac{1}{2}\left( - |\psi_{m}^{*}\rangle \langle d\psi_{m}^{*}| \pi - |\psi_{m}^{*}\rangle \langle d\psi_{m}^{*}| \pi + |\psi_{i}^{*}\rangle \langle d\psi_{i}^{*}| \pi - |\psi_{i}^{*}\rangle \langle d\psi_{i}^{*}| \pi \right) \right] \quad (A.21)
\]

\[
\partial_{\psi_{m}}\partial_{\psi_{i}}E\langle d\psi_{m}\rangle = \rho\partial_{\rho}E\rho\left[ \frac{1}{2}\left( |\psi_{m}^{*}\rangle \langle d\psi_{m}^{*}| \pi + |\psi_{m}^{*}\rangle \langle d\psi_{m}^{*}| \pi \right) \right] \quad (A.22)
\]

If we introduce the Kohn Sham operator

\[
H_{\rho} = \partial_{\rho}E = \int \left[ -\frac{1}{2}\vec{r}^{2} + \int \frac{\langle r'|\rho|r'\rangle}{|r - r'|} dr' + v_{n}(r)\right] \langle r| \otimes |r\rangle dr + \partial_{\rho}E_{xc}d\rho, \quad (A.23)
\]

and the following operators (that also appear in density functional perturbation theory)

\[
\hat{O}_{lm} = \partial_{\rho}\partial_{\rho}E\rho\langle |\psi_{l}\rangle \rangle d\rho(|\psi_{m}^{*}\rangle) \\
= \int \frac{\langle r|\psi_{l}\rangle \langle r'|\psi_{m}^{*}\rangle}{|r - r'|} |r\rangle \otimes |r\rangle drdr' + \partial_{\rho}\partial_{\rho}E_{xc}d\rho(|\psi_{l}\rangle \rangle d\rho(|\psi_{m}^{*}\rangle) \quad (A.24)
\]

\[
O_{lm} = \partial_{\rho}\partial_{\rho}E\rho\langle |\psi_{l}\rangle \rangle d\rho(|\psi_{m}^{*}\rangle) \\
= \int \frac{\langle r|\psi_{l}\rangle \langle r'|\psi_{m}^{*}\rangle}{|r - r'|} |r\rangle \otimes |r\rangle drdr' + \partial_{\rho}\partial_{\rho}E_{xc}d\rho(|\psi_{l}\rangle \rangle d\rho(|\psi_{m}^{*}\rangle). \quad (A.25)
\]
we can rewrite the derivatives as follow:

$$\partial_{\psi_1} E\langle d\psi_1 | = \frac{1}{2} \langle d\psi_1 | \pi^* H_0 | \psi_1^* \rangle + \frac{1}{2} \langle d\psi_1 | \pi H_0 | \psi_1^* \rangle$$ (A.26)

$$\partial_{\psi_m} \partial_{\psi_1} E\langle d\psi_m | \langle d\psi_1 | =$$

$$- \frac{1}{2} \left( \langle d\psi_1 | \pi^* H_0 | \psi_m^* \rangle \langle d\psi_m | \psi_1^* \rangle + \langle d\psi_1 | \psi_m^* \rangle \langle d\psi_m | \pi H_0 | \psi_1^* \rangle \right)$$

$$+ \langle d\psi_1 | \pi H_0 | \psi_m^* \rangle \langle d\psi_m | \psi_1^* \rangle + \langle d\psi_1 | \psi_m^* \rangle \langle d\psi_m | \pi H_0 | \psi_1^* \rangle \right)$$ (A.27)

$$+ \partial_\rho \partial_\rho E d\rho \left( \frac{1}{2} \langle \psi_1^* \rangle \langle d\psi_1 | \pi^* H_0 \rangle + \frac{1}{2} \langle \psi_1^* \rangle \langle \psi_1^* | \pi \rangle \right)$$

$$d\rho \left( \frac{1}{2} \langle \psi_m^* \rangle \langle d\psi_m | \pi^* \rangle + \frac{1}{2} \langle \psi_m^* \rangle \langle \psi_m^* | \pi \rangle \right)$$

$$\partial_{\psi_m} \partial_{\psi_1} E\langle d\psi_m | \langle d\psi_1 | =$$

$$\frac{1}{2} \left( \langle d\psi_1 | \pi^* H_0 \pi | d\psi_m \rangle S_m^{-1} - \langle d\psi_1 | \pi | d\psi_m \rangle \langle \psi_m^* | H_0 | \psi_1^* \rangle \right)$$

$$\langle d\psi_1 | \pi H_0 \pi | d\psi_m \rangle S_m^{-1} - \langle d\psi_1 | \pi | d\psi_m \rangle \langle \psi_m^* | H_0 | \psi_1^* \rangle \right)$$ (A.28)

$$+ \partial_\rho \partial_\rho E d\rho \left[ \frac{1}{2} \left( \langle \psi_1^* \rangle \langle d\psi_1 | \pi^* H_0 \rangle + \langle \psi_1^* \rangle \langle \psi_1^* | \pi \rangle \right) \right]$$

$$d\rho \left[ \frac{1}{2} \left( \langle \psi_m^* | \pi H_0 \pi | d\psi_m \rangle + \langle \psi_m^* | \pi \rangle | d\psi_m \rangle \langle \psi_m^* | \pi \rangle \right) \right]$$

### A.4 XC derivatives

For the perturbation kernel we need the derivatives

$$\frac{\partial E^{XC}}{\partial P} dP = \frac{\partial E^{XC}}{\partial n} d\tilde{n} \frac{\partial \tilde{n}}{\partial P} dP$$

and

$$\frac{\partial^2 E^{XC}}{\partial P \partial P} dP dP = \frac{\partial^2 E^{XC}}{\partial n \partial n} d\tilde{n} \frac{\partial \tilde{n}}{\partial P} dP d\tilde{n} \frac{\partial \tilde{n}}{\partial P} dP$$

where $\tilde{n}$ is the density on the grid. The first derivative is simply $v_{xc}(r)$ and the integration of $dn \frac{\partial \tilde{n}}{\partial P} dP$ is the integration procedure dual of the mapping 2.7.5.1 performed on the grid.

The exchange-correlation functionals we will handle have, as described in section 2.4.4 the common general form

$$E^{XC}[n] = \int e^{xc}(n_1(r), n_1(r), \nabla n_1(r), \nabla n_1(r), \tau_1, \tau_1) dr.$$

(A.29)
If the implemented functionals have the analytical derivatives of $e^{xc}$, as it is the case for the most important functionals in cp2k, then for the first derivative one has to integrate $\partial e^{xc} / \partial n_\uparrow$, $\partial e^{xc} / \partial \nabla n_\uparrow$ and $\partial e^{xc} / \partial \nabla \tau_\uparrow$ with the corresponding partial derivatives

$$\frac{\partial n_\uparrow}{\partial n_\uparrow} dn_\uparrow = dn_\uparrow$$  \hspace{1cm} (A.30)

$$\frac{\partial \nabla n_\uparrow}{\partial n_\uparrow} dn_\uparrow = \nabla dn_\uparrow$$  \hspace{1cm} (A.31)

$$\frac{\partial \tau_\uparrow}{\partial \tau_\uparrow} d\tau_\uparrow = d\tau_\uparrow$$  \hspace{1cm} (A.32)

together with the symmetric ones obtained by swapping $\uparrow$ and $\downarrow$, as the others derivatives are 0.

The $\tau$ terms as

$$\tau_\uparrow(r) = \frac{1}{2} \sum_i \nabla \Psi_i^\uparrow(r) \cdot \nabla \Psi_i^\uparrow(r) = \frac{1}{2} \sum_{\mu\nu} P^{\mu\nu}_i \nabla \phi_\mu(r) \cdot \nabla \phi_\nu(r)$$  \hspace{1cm} (A.33)

are easy because the derivative is performed on the Gaussian when integrating the function on the grid as explained in section 2.4.4.

Normally the functionals do not depend on $\nabla n_\downarrow$ directly, but rather on $|\nabla n_\uparrow|$, $|\nabla n_\downarrow|$, $|\nabla n_\uparrow + \nabla n_\downarrow|$. Instead of $|\nabla n_\uparrow + \nabla n_\downarrow|$ one can use $\nabla n_\uparrow \cdot \nabla n_\downarrow$ as one can be expressed in the term of the other with $|\nabla n_\uparrow|$ and $|\nabla n_\downarrow|$. These have derivatives

$$\frac{\partial |\nabla n_\uparrow|}{\partial n_\uparrow} dn_\uparrow = \frac{\nabla n_\uparrow \cdot \nabla dn_\uparrow}{|\nabla n_\uparrow|}$$  \hspace{1cm} (A.34)

$$\frac{\partial |\nabla n_\uparrow + \nabla n_\downarrow|}{\partial n_\uparrow} dn_\uparrow = \frac{(\nabla n_\uparrow + \nabla n_\downarrow) \cdot \nabla dn_\uparrow}{|\nabla n_\uparrow + \nabla n_\downarrow|}$$  \hspace{1cm} (A.35)

$$\frac{\partial \nabla n_\uparrow \cdot \nabla n_\downarrow}{\partial n_\uparrow} dn_\uparrow = \nabla dn_\uparrow \cdot \nabla n_\downarrow$$  \hspace{1cm} (A.36)

plus their symmetric ones.

To avoid the calculation of $\nabla dn_\uparrow$, one can push the derivative on the other side by performing a partial integration. As shown in section 2.5 this can be done consistently also using numerical derivatives, and actually comes automatically by deriving the numerical derivative with respect to its argument.

For example for the derivative with respect to $|\nabla n_\uparrow|$ we have

$$\frac{\partial e^{xc}}{\partial |\nabla n_\uparrow|} \frac{\nabla n_\uparrow \cdot \nabla dn_\uparrow}{|\nabla n_\uparrow|} = -\nabla \cdot \left( \frac{\partial e^{xc}}{\partial |\nabla n_\uparrow|} \frac{\nabla n_\uparrow}{|\nabla n_\uparrow|} \right)dn_\uparrow.$$  \hspace{1cm} (A.37)
With this transformations the derivative will contain only \( dn \) and \( d\tau \) terms and not its derivatives \( \nabla dn \), and can be represented as two potentials on the grid with a component for each spin, namely \( v^{xc} \).

To have good performance as many partial integrations as possible should be done at once by summing the terms before deriving.

The second derivative is not really more complicated, apart having more terms. The only thing that one should not forget is that \( |V_n|, |\nabla n_1|, |\nabla n_1 + \nabla n_1| \) and \( \nabla n_1 \cdot \nabla n_1 \) have a nonzero second derivative with respect to the density that are

\[
\frac{\partial^2 |V_n|}{\partial n_1 \partial n_1} dn_1 dn_1 = \frac{\nabla dn_1 \cdot \nabla dn_1}{|\nabla n_1|^3} - \frac{(\nabla n_1 \cdot \nabla dn_1)(\nabla n_1 \cdot \nabla dn_1)}{|\nabla n_1|^3}
\]

\[
\frac{\partial^2 |\nabla n_1 + \nabla n_1|}{\partial n_1 \partial n_1} dn_1 dn_1 = \frac{\nabla dn_1 + \nabla \cdot \nabla dn_1}{|\nabla n_1 + \nabla n_1|^3} - \frac{(\nabla n_1 + \nabla n_1) \cdot \nabla dn_1}{|\nabla n_1 + \nabla n_1|^3} ((\nabla n_1 + \nabla n_1) \cdot \nabla dn_1)
\]

\[
\frac{\partial^2 \nabla n_1 \cdot \nabla n_1}{\partial n_1 \partial n_1} dn_1 dn_1 = \nabla dn_1 \cdot \nabla dn_1
\]

plus their symmetric ones.

With these partial derivatives one can build up xc part of \( V^{P_1} \) which is separated in two terms \( v^{P_1}_{xc}(r) \) and \( v^{P_2}_{xc}(\tau) \) both on the grid, each with a spin up and down component, just like the first derivative \( v^{xc} \), in fact once one direction is chosen a second derivative becomes a first derivative.

The procedure to perform it is

1. for all (first and second) derivatives of \( v^{xc} \)

   (a) if the derivative is one of the special first derivatives \( |\nabla n_1|, |\nabla n_1|, |\nabla n_1 + \nabla n_1| \) and \( \nabla n_1 \cdot \nabla n_1 \) multiply the derivative with the term found previously where \( dn_1, d\tau \) are replaced with the ones calculated with \( P_1 \) (thus one might use \( |\nabla n_1| \) directly as it is known without performing its derivative) and then collect all the terms \( dn_2, \nabla dn_2 \) and \( d\tau_2 \).
(b) if it is a second derivative
\[ \frac{\partial \epsilon_{xc}}{\partial X \partial Y} dX dY \] (A.42)
do the following
i. if \( X = Y \) it is a diagonal term, do \( dX [P_1] \), i.e. multiply the term with \( n_1, \tau_1, |\nabla n_1|,... \) depending if \( X \) is \( n, \tau, |\nabla n|,... \) and treat the resulting term just like the \( Y \) derivative in the \( \nu^{xc} \) buildup, collecting the terms in \( dn_2, \nabla dn_2 \) and \( d\tau_2 \)
ii. if \( X \neq Y \) it is an off diagonal term and the procedure of the previous point should be performed also with \( X \) and \( Y \) swapped.

2. finally partial integration should be applied to the \( \nabla dn_2 \) terms and thus find \( v_{xc}^{P_1}(r) dn_2 \) and \( v_{xc}^{P_3}(\tau) d\tau_2 \).

This was the way I had implemented the derivatives in CP2K at the beginning, but there was not a safe way to extract the information about which derivatives were present where and thus the loop was never debugged and had errors for derivatives involving the gradient of the density. Now the part handling gradient derivatives has been rewritten by Thomas Chassaing and Marcella Iannuzzi (I did not need much the gradient corrections) after I changed the whole xc structure to store meta informations about which derivatives are present where, and (somewhat unsatisfactorily) it was solved in hard coding each \( dX dY \) derivative and not dynamically looking at each partial derivative. To cope with \( \tau \) derivatives that at the moment are not handled one would probably need to revert to the dynamical approach because the possible term combinations are too many.

### A.5 Derivatives of a CG step

In this section the \( n_{mo} \times n_{mo} \) matrices are seen as \( n_{mo} n_{mo} \) vectors, unless otherwise stated, thus \( p^T c = \text{Tr}(p^T c) \).

The algorithm to compute conjugated gradient and its derivatives explained in section 5.5 needs to transform the derivatives to the normal form
\[ \text{Tr}(W_{\epsilon} \frac{\partial}{\partial R}) \] (A.43)
described in (5.60) and transfer the \( W_{\epsilon} \) and \( W_{sc} \) of (5.65) to lower indices.
A.5.1  \( \textit{db} \) derivative

To do this first let’s look at
\[
\text{Tr}(dC^\dagger db) = -D_R \frac{\partial E}{\partial C} dC
\]  \hspace{1cm} (A.44)

where \( D_R \) is the total derivative with respect to \( R \). Using 5.47, i.e.
\[
\frac{\partial P}{\partial C} dC = -(I - PS) dCS^{-1} C^\dagger \frac{\partial S}{\partial R} P - CS^{-1} dC^\dagger (I - SP) \frac{\partial S}{\partial R} P + c.c.
\]  \hspace{1cm} (A.45)
it becomes
\[
\text{Tr}(dC^\dagger db) = -\text{Tr} \left( \frac{\partial H}{\partial R} (I - PS) dCS^{-1} C^\dagger + c.c. \right) + \frac{\partial^2 E}{\partial P \partial R} dC_2 [(I - PS) \frac{\partial C}{\partial R} S^{-1} C^\dagger + c.c.] dP [(I - PS) dCS^{-1} C^\dagger + c.c.] + c.c.
\]  \hspace{1cm} (A.46)

and its \( W_{sr} \) are
\[
W_{sr} = PH(I - PS) dCS^{-1} C^\dagger + PHCS^{-1} dC^\dagger (I - SP) + c.c.
\]  \hspace{1cm} (A.47)
\[
W_{hr} = (I - PS) dCS^{-1} C + c.c.
\]  \hspace{1cm} (A.48)
\[
W_{cr} = (I - SP) V_{P_1} CS^{-1} + c.c.,
\]  \hspace{1cm} (A.49)

where \( P_1 = (I - PS) dCS^{-1} C^\dagger + c.c. \), and the second order perturbation kernel \( V_{P_1} \) is defined as
\[
\text{Tr}(V_{P_1} dP_2) = \frac{\partial^2 E}{\partial P \partial P} dP_1 [P_1] dP_2
\]  \hspace{1cm} (A.50)

and can be calculated with equation (5.30), and \( \partial H / \partial R \) is defined in (5.49). The presence of \( W_{cr} \) is what, multiplied repeatedly with \( A = \partial^2 E / (\partial C \partial C) \), gives rise to the incalculable terms that force us to do the inverse derivative accumulation.

A.5.2  \( \textit{dA} \) derivative

The \( \textit{dA} \) derivative is
\[
\text{Tr}(dC_1^\dagger dAdC_2) = D_R \frac{\partial^2 E}{\partial C \partial C} dC_1 dC_2
\]  \hspace{1cm} (A.51)
which is more complicated. Using

\[
\frac{\partial^3 P}{\partial C \partial C \partial R} dC_1 dC_2 = -\frac{\partial P}{\partial C} dC_1 \frac{\partial S}{\partial R} \frac{\partial P}{\partial C} dC_2 - \frac{\partial^2 P}{\partial C \partial R} dC_1 dC_2 \frac{\partial S}{\partial R} P + c.c.
\]

\[
= -((I - PS) dC_1 S^{-1} C^\dagger + c.c.) \frac{\partial S}{\partial R} ((I - PS) dC_2 S^{-1} C^\dagger + c.c.)
\]

\[
- (CS^{-1} dC_1(-S +SPS) dC_2 S^{-1} C^\dagger
\]

\[
+ (I - PS) dC_1 S^{-1} dC_2 (I - SP) + c.c.) \frac{\partial S}{\partial R} P + c.c.
\]

\[(A.52)\]

where like before the part that couples \( V \) and \( V_{\perp} \) has been removed, we can express the \( H \) part of the derivative

\[
D \cdot \text{Tr}(H \frac{\partial P}{\partial C \partial C} dC_1 dC_2) = \text{Tr}(\frac{\partial H}{\partial R} \frac{\partial^2 P}{\partial C \partial C} dC_1 dC_2) + \text{Tr}(H \frac{\partial^3 P}{\partial C \partial C \partial R} dC_1 dC_2) + \text{Tr}(H \frac{\partial^3 P}{\partial C \partial C \partial R} dC_1 dC_2 dC_3 \frac{\partial C}{\partial R})
\]

\[(A.53)\]

thus from the first term

\[
W_{hr} = \frac{\partial^2 P}{\partial C \partial C} dC_1 dC_2
\]

\[(A.54)\]

the second term

\[
W_{sr} = -\frac{\partial P}{\partial C} dC_2 H \frac{\partial P}{\partial C} dC_1 - PH \frac{\partial^2 P}{\partial C \partial C} dC_1 dC_2 + c.c.
\]

\[(A.55)\]

and the third term

\[
W_{cr} = H \frac{\partial^3 P}{\partial C \partial C \partial C} dC_1 dC_2 dC_3
\]

\[(A.56)\]

d\( A \) contains also a \( \partial^2 E/(\partial n \partial n) \) part. that has derivative

\[
\text{Tr}(\frac{\partial V_P}{\partial R} \frac{\partial P}{\partial C} dC_2) + \text{Tr}(V_P \frac{\partial^2 P}{\partial C \partial R} dC_2) + \text{Tr}(\frac{\partial V_P}{\partial R} \frac{\partial P}{\partial C} dC_1) + \text{Tr}(V_P \frac{\partial^2 P}{\partial C \partial R} dC_1)
\]

\[(A.57)\]

where \( P = \partial P/\partial C dC_1 \). With \( \partial V_P/\partial R \) it is meant the integration with \( -\nabla_I(\phi_p(R)\phi_p(R)) \) of the potential \( \nu_{\perp}^{P} (n) \) and the electrostatic one \( \nu_{\perp}^{P} \) on the grid, and the integration of \( \nu_{\perp}^{P} (\tau) \) with \( -\nabla_I(\nabla \phi_p(R) \cdot \nabla \phi_p(R)) \), just like it is done when calculating the forces due to \( \nu_{\perp} \). These derivatives have to be calculated immediately because the values of \( P_1 \) and \( P_2 \) change from time to time. Fortunately as we will see there are just three such derivatives at each iteration level.
The other two terms contribute to $W_{sr}$

\[
W_{sr} = -PV^2 (I - PS)dC_2 S^{-1}C^T - PV^2 CS^{-1}dC_2^T(I - SP) - PV^2 (I - PS)dC_1 S^{-1}C^T - PV^2 CS^{-1}dC_1^T(I - SP) + c.c. \quad (A.58)
\]

Finally the last term of $dA$ is

\[
\frac{\partial^3 E^{XC}}{\partial n_1 \partial n_2 \partial n_3} d_{n_1} d_{n_2} d_{n_3} \sum_{\mu \nu} [\nabla R^\mu (\phi_n) P^{\mu \nu} (\phi_n) + \sum_{\mu \nu} (\phi_n) P^{\mu \nu} (\nabla R^\mu (\phi_n)) + \sum_{\mu \nu} (\phi_n)((I - PS) \frac{\partial C}{\partial R} S^{-1}C^T)_{\mu \nu} (\phi_n) + c.c.]
\]

that is similar to $V^F$

### A.5.3 $F$ derivative

The $R$ derivative of $F$ for any coefficient vector $w$ (actually a $n_{mo} \times n_{ao}$ matrix) is

\[
\text{Tr}(w \frac{\partial F}{\partial R}) = \text{Tr}(w \frac{\partial F}{\partial b} db \frac{\partial b}{\partial R}) + \text{Tr}(w \frac{\partial F}{\partial A} dA \frac{\partial A}{\partial R}) + \text{Tr}(w \frac{\partial F}{\partial M} dM \frac{\partial M^{-1}}{\partial R}). \quad (A.60)
\]

Its first term is

\[
\text{Tr}(w \frac{\partial F}{\partial b} db) = \text{Tr}(\alpha w^T pr^T M^{-1}(-\frac{2I}{r^T M^{-1} r} - \frac{AM^{-1}}{p^T AM^{-1} r}) db) + \text{Tr}(\frac{cAM^{-1} db}{c^T A c} (\alpha \frac{r^T M^{-1} A c}{p^T AM^{-1} r} w^T p - w^T c)) + \text{Tr}(\alpha w^T db) \quad (A.61)
\]

and the second one

\[
\text{Tr}(w \frac{\partial F}{\partial A} dA) =
\]

\[
\left(\alpha \frac{w^T pr^T M^{-1}(-\frac{2I}{r^T M^{-1} r} + \frac{AM^{-1}}{p^T AM^{-1} r})}{p^T AM^{-1} r} + \frac{\alpha}{c^T A c} (-\frac{w^T pr^T M^{-1} A c}{p^T AM^{-1} r} + w^T c) c^T AM^{-1}
\right) \text{dAx} + \left(\alpha \frac{w^T p}{p^T AM^{-1} r} p^T AM^{-1} - \alpha w^T M^{-1} dA x - \frac{w^T p}{p^T AM^{-1} r} p^T dAM^{-1} r \right)
\]

\[
+ \left(\alpha \frac{w^T r^T M^{-1} A c}{p^T AM^{-1} r} (r^T M^{-1} - \beta c^T) - \frac{\alpha w^T c}{c^T A c} (r^T M^{-1} + \beta c^T)\right) dAc
\]  

\[
(A.62)
\]
where the terms can be collected differently, but the you still need 3 different evaluations of \(dA\).

The term \(M^{-1}\) derivative we ignore it assuming that the preconditioner is just \(I\).

For the \(C\) derivative
\[
\text{Tr}(w \frac{\partial F}{\partial C} dC) = \text{Tr}(w \frac{\partial F}{\partial A} dA[\frac{\partial A}{\partial C} dC]) + \text{Tr}(w \frac{\partial F}{\partial b} db[\frac{\partial b}{\partial C}]) + \text{Tr}(w \frac{\partial F}{\partial M^{-1} dM^{-1}} \frac{\partial M^{-1}}{\partial C})
\]
(A.63)

the only missing term is the
\[
\frac{\partial^3 P}{\partial C \partial C \partial C}
\]
(A.64)

needed for \(\partial A/\partial C\).

Finally there are still
\[
\text{Tr}(w \frac{\partial F}{\partial c} dc) = \alpha(w^T + \frac{w^T c^T}{p^t A M^{-1} r} r^T M^{-1} A) (\frac{c}{c^T A c} (r^T M^{-1} A + 2c^T A) + 1) dc
\]
(A.65)

and
\[
\text{Tr}(w \frac{\partial F}{\partial x} dx) = \alpha w^T p r^T M^{-1} (-\frac{2}{r^T M^{-1} r} + A (1 + \frac{c^T A}{c^T A c}))(Adx
\]
\[
+ \alpha \frac{w^T p}{r^T A M^{-1} r} p^T A M^{-1} Adx + \alpha w^T (1 - \frac{c^T A}{c^T A c} M^{-1} Adx
\]
(A.66)

that complete the definition of the algorithm.
Curriculum Vitae

Name  Fawzi Roberto MOHAMED

Born  9.9.1976, Sorengo, Ticino, Switzerland

Civil status  married

Nationality  swiss/tunisian

Languages  italian, french, german and english

Studies

• 1991-1995 Liceo Lugano 2 Matura (type C, scientific)

• 10.1995-10.1997 1. & 2. Vordiplom at ETH Zürich in physics


• 4.2000-10.2000 Diplomarbeit by Prof. Blatter and Prof. Troyer, during which a C++ program for the simulation of the Interaction of vortices in superconductors with $\kappa$ close to $1/\sqrt{2}$ was developed

• 10.2000 Diplom im Rechnergestützten Wissenschaften der ETH Zürich

• 11.2000-8.2001 Worked at the Uptime Object Factory at an object oriented communication protocol (Objective-C/Java).

• 11.2001-2006 PhD. at the ETH Zürich with Professor M. Parrinello, on DFT MD simulation, connected with the development of the open source DFT package CP2K/Quickstep written in Fortran 90

• 1.3.2006 scholarship as researcher at the Scuola Normale di Pisa

• 18.10.2006 passed PhD. Exam at ETH Zürich
List of Publications

Parts of this thesis are published in journal articles and conference proceedings.


