Fast and accurate electronic structure methods: large systems and applications to boron-carbon heterofullerenes

Inauguraldissertation

zur

Erlangung der Würde eines Doktors der Philosophie

vorgelegt der

Philosophisch-Naturwissenschaftlichen Fakultät

der Universität Basel

von

Stephan Mohr

aus Susch, Graubünden

Basel, 2013

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät
auf Antrag von

Prof. Dr. Stefan Goedecker
Dr. habil. Thierry Deutsch

Basel, den 18. Juni 2013

Prof. Dr. Jörg Schibler
Dekan

Contents

Acknowledgements

First of all I would like to express my deep gratitude to my supervisor, Prof. Dr. Stefan Goedecker, for offering me the possibility to work on such interesting topics and his continuous support during the period of my PhD.

Furthermore I would like to thank the entire BigDFT team in Basel and Grenoble for its precious assistance. In particular I would like to mention Dr. Luigi Genovese who has always been able to help in case any problems appeared, and Dr. Paul Boulanger and Dr. Laura Ratcliff, who are as well heavily involved in the development of the linear scaling version of BigDFT. In addition I would like to thank Jan Spörri for parallelizing large parts of the linear scaling version with OpenMP and Nazim Dugan for implementing a new parallelization scheme for the Poisson Solver.

With respect to the second part of the Thesis I am very grateful to Dr. Pascal Pochet for helpful discussions and inputs.

Last but not least I would like to thank all former and present members of Stefan Goedecker's group in Basel, who created always a pleasant ambiance for my work, and the technical staff – in particular the secretaries – at the University of Basel.

Part I

Linear scaling Density Functional Theory for large systems

CHAPTER 1

Introduction

Any piece of matter, be it a small isolated molecule or a large infinite periodic crystal, is in principle just a collection of nuclei and electrons. The interactions among them and consequently the properties of matter are governed by the fundamental laws of quantum mechanics. Since the basic equations describing these interactions are known, the determination of the properties of matter seems to be a simple task at first sight. Methods that use these laws are called ab-initio methods.

For some very simple examples – the most famous one probably being the hydrogen atom – the fundamental equations can be solved analytically. Even for slightly more complicated systems, an analytical solution is not possible any more and one therefore has to either use some approximations which allow an analytical solution or to solve the equations numerically on a computer. However, for most systems of interest, even the best supercomputers available nowadays are not capable to solve the quantum mechanical problem in its exact form. Consequently one has – even when using a numerical approach – to search for some simplifications in order to make the equations solvable while still keeping the quantum mechanical origin of the description.

The first fundamental approximation that is usually adopted is the so-called Born-Oppenheimer approximation which allows to treat the nuclei as classical particles. One is therefore left with the task of solving the electronic structure problem in a quantum-mechanical way. Unfortunately also this problem remains way too complicated in order to be solved exactly even numerically and one therefore has to adopt further simplifications.

There exist several such approximations, differing conceptually by how much they

stick to the fundamental quantum mechanical equations. Thus the accuracy and consequently also the speed of these methods vary a lot and the number of atoms that can be treated with them ranges from only a few ones to several millions.

One of the most famous approaches to accomplish the task of solving the electronic structure problem is the framework of Density Functional Theory (DFT). Here the approximation consists of turning the system of interacting electrons into a system of non-interacting quasi-electrons. Since its development in the 1960s, DFT has become one of the most popular electronic structure methods due to its good balance between accuracy and speed.

Even though DFT can offer a substantial speedup compared to other ab-initio methods, its usage is still limited to currently a few hundred atoms. The reason is its asymptotically cubic scaling, which makes calculations for really large system prohibitive.
Fortunately this problem can be circumvented by the introduction of so-called linear scaling algorithms. Of course these algorithms come again at the cost of some further approximations, but it can be shown that they are well justified and linear scaling DFT is consequently still – at least to the extent to which standard DFT is – a fully ab-initio method. Using these low-complexity algorithms it is possible to carry out DFT calculations for thousands or even millions of atoms, in this way pushing up the size of the systems that can be investigated with ab-initio methods.

Nevertheless DFT calculations for large systems remain a very sophisticated task and are only doable on large supercomputers. Due to the fact that the computational power of a single core does not increase any further and the overall power of the supercomputers nowadays stems from their massive parallelism, it is of utmost importance that any code that aims to run on such a machine is highly parallelized. Therefore an efficient parallel implementation is as important as using a good physical approach.

This first part of the Thesis describes the implementation of a linear scaling DFT code within the framework of the already existing BigDFT package. To this end the fundamental principles of electronic structure calculations and DFT in particular are presented first, followed by a brief outline on how the intrinsic cubic scaling of this approach can be linearized. After a short overview over the wavelet basis set that is used in BigDFT and which exhibits some very nice properties making it an ideal basis set for linear scaling calculations, the text focuses on the implementation of the linear scaling version of BigDFT. In this section the style will be a mixture of theoretical considerations and practical applications, in this way trying to illustrate the problems that had to be dealt with during the implementation. A strong focus will also be put on the parallelization of the code. The first part concludes with some benchmark results demonstrating the capabilities of the linear scaling version of BigDFT.

It must be noted that the even though the code is capable of giving accurate results at an almost perfect linear scaling, it is still under development and there are still some open problems that need to be addressed. An overview of these issues is given towards the end of this first part.

CHAPTER 2

Some basics about electronic structure calculations

2.1 The Born-Oppenheimer approximation

Due to the quantum-mechanical nature of the electrons and the nuclei which are the constituents of matter, an exact calculation of their interactions is – except for the most simple cases – not possible. Consequently one has to introduce some approximations in order to be able to solve the problem.
The range of approximations is very wide, but in general all of them rely on the Born-Oppenheimer approximation which will be derived in the following [1].

A priori, ab-initio calculations for a system composed of electrons and nuclei require to treat both of them quantum-mechanically, i.e. the combined electron-nuclei wave function $\Psi^{en}(\{\mathbf{R}_l\},\{\mathbf{r}_l\})$ has to be calculated, where $\{\mathbf{R}_l\}$ stands for the coordinates of all nuclei in the system and $\{\mathbf{r}_l\}$ for those of all electrons. This wave function is an eigenfunction of the combined electron-nuclei Hamiltonian

$$\mathcal{H}^{en}(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\Psi^{en}(\{\mathbf{R}_l\},\{\mathbf{r}_l\}) = E^{en}\Psi^{en}(\{\mathbf{R}_l\},\{\mathbf{r}_l\}), \tag{2.1}$$

where $\mathcal{H}^{en}(\{\mathbf{R}_l\},\{\mathbf{r}_l\})$ is defined as

$$\mathcal{H}^{en}(\{\mathbf{R}_l\},\{\mathbf{r}_l\}) = \mathcal{T}^{n}(\{\mathbf{R}_l\}) + \mathcal{H}(\{\mathbf{R}_l\},\{\mathbf{r}_l\}) \tag{2.2}$$

with

$$\begin{aligned}\mathcal{T}^n(\{\mathbf{R}_l\}) &= -\sum_{i=1}^{N}\frac{1}{2M_i}\nabla^2_{\mathbf{R}_i},\\ \mathcal{H}(\{\mathbf{R}_l\},\{\mathbf{r}_l\}) &= \sum_{i=1}^{N}\sum_{j=1}^{i-1}\frac{Z_iZ_j}{|\mathbf{R}_i-\mathbf{R}_j|}-\sum_{i=1}^{n}\frac{1}{2}\nabla^2_{\mathbf{r}_i}+\sum_{i=1}^{n}\sum_{j=1}^{i-1}\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|}-\sum_{i=1}^{n}\sum_{j=1}^{N}\frac{Z_j}{|\mathbf{r}_i-\mathbf{R}_j|}.\end{aligned} \tag{2.3}$$

Here M_i stands for the mass of the ith nucleus in atomic units, N for the total number of atoms and n for the total number of electrons. $\mathcal{T}^n(\{\mathbf{R}_l\})$ is the kinetic energy of the nuclei, and the terms of $\mathcal{H}(\{\mathbf{R}_l\},\{\mathbf{r}_l\})$ are the electrostatic repulsion among the nuclei, the kinetic energy of the electrons, the electrostatic repulsion among the electrons and the electrostatic attraction between the electrons and the nuclei.
The above operators were written in atomic units which are defined by setting $m_e = 1$, $e = 1$, $\hbar = 1$, $1/4\pi\epsilon_0 = 1$. This convention will always be used in the following unless otherwise stated. The other convention which will be used throughout the Thesis is that only non-complex quantities are considered.

Unfortunately the above eigenvalue equation is way too complicated to be solved directly. Therefore one has to adopt some approximations.
To this end one introduces the electronic wave functions $\Phi_k(\{\mathbf{R}_l\},\{\mathbf{r}_l\})$ which are eigenfunctions of the electronic Hamiltonian,

$$\mathcal{H}(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\Phi_k(\{\mathbf{R}_l\},\{\mathbf{r}_l\}) = \epsilon_k(\{\mathbf{R}_l\})\Phi_k(\{\mathbf{R}_l\},\{\mathbf{r}_l\}). \tag{2.4}$$

Due to the hermiticity of the operator $\mathcal{H}(\{\mathbf{R}_l\},\{\mathbf{r}_l\})$ its eigenfunction form a complete set with respect to the space of the electronic coordinates. Therefore the combined electron-nuclei wave function can be expanded in this basis:

$$\Psi^{en}(\{\mathbf{R}_l\},\{\mathbf{r}_l\}) = \sum_k \Phi_k(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\psi^n_k(\{\mathbf{R}_l\}). \tag{2.5}$$

Inserting this expansion into Eq. (2.1) yields

$$\begin{aligned}-\sum_{i=1}^{N}\sum_k\frac{1}{2M_i}\nabla^2_{\mathbf{R}_i}\Phi_k(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\psi^n_k(\{\mathbf{R}_l\}) + \sum_k\epsilon_k(\{\mathbf{R}_l\})\Phi_k(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\psi^n_k(\{\mathbf{R}_l\})\\ = E^{en}\sum_k\Phi_k(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\psi^n_k(\{\mathbf{R}_l\}).\end{aligned} \tag{2.6}$$

The next step is to multiply from left with $\Phi_j(\{\mathbf{R}_l\},\{\mathbf{r}_l\})$ and to integrate. Using the orthonormality relation

$$\int d\mathbf{r}_1\cdots\int d\mathbf{r}_n\,\Phi_j(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\Phi_k(\{\mathbf{R}_l\},\{\mathbf{r}_l\}) = \delta_{jk} \tag{2.7}$$

this yields

$$-\sum_{i=1}^{N}\sum_{k}\frac{1}{2M_i}\int \mathrm{d}\mathbf{r}_1\cdots\int \mathrm{d}\mathbf{r}_n\,\Phi_j(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\nabla^2_{\mathbf{R}_i}\Phi_k(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\psi^n_k(\{\mathbf{R}_l\}) + \epsilon_j(\{\mathbf{R}_l\})\psi^n_j(\{\mathbf{R}_l\}) = E^{en}\psi^n_j(\{\mathbf{R}_l\}). \quad (2.8)$$

Applying the product rule for the Laplace operator ∇^2 and again using the orthonormality relation one arrives at

$$\begin{aligned}&-\sum_{i=1}^{N}\frac{1}{2M_i}\nabla^2_{\mathbf{R}_i}\psi^n_j(\{\mathbf{R}_l\})\\ &\quad-\sum_{i=1}^{N}\sum_{k}\frac{1}{2M_i}\int \mathrm{d}\mathbf{r}_1\cdots\int \mathrm{d}\mathbf{r}_n\Big[\Phi_j(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\nabla_{\mathbf{R}_i}\Phi_k(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\nabla_{\mathbf{R}_i}\psi^n_k(\{\mathbf{R}_l\})\\ &\qquad+\Phi_j(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\nabla^2_{\mathbf{R}_i}\Phi_k(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\psi^n_k(\{\mathbf{R}_l\})\Big]\\ &\qquad+\epsilon_j(\{\mathbf{R}_l\})\psi^n_j(\{\mathbf{R}_l\}) = E^{en}\psi^n_j(\{\mathbf{R}_l\}). \quad (2.9)\end{aligned}$$

So far no approximation has been used and everything is still exact. However now the first simplification comes into play. In the so-called adiabatic approximation the sum that runs over k and in this way couples different electronic eigenstates is completely discarded and only the electronic ground state $\Phi_0(\{\mathbf{R}_l\},\{\mathbf{r}_l\}),\epsilon_0(\{\mathbf{R}_l\})$ is used throughout the entire equation; this will also allow to replace ψ^n_j by ψ^n_0:

$$\begin{aligned}&-\sum_{i=1}^{N}\frac{1}{2M_i}\nabla^2_{\mathbf{R}_i}\psi^n_0(\{\mathbf{R}_l\})\\ &\quad-\sum_{i=1}^{N}\frac{1}{2M_i}\int \mathrm{d}\mathbf{r}_1\cdots\int \mathrm{d}\mathbf{r}_n\Big[\Phi_0(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\nabla_{\mathbf{R}_i}\Phi_0(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\nabla_{\mathbf{R}_i}\psi^n_0(\{\mathbf{R}_l\})\\ &\qquad+\Phi_0(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\nabla^2_{\mathbf{R}_i}\Phi_0(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\psi^n_0(\{\mathbf{R}_l\})\Big]\\ &\qquad+\epsilon_0(\{\mathbf{R}_l\})\psi^n_0(\{\mathbf{R}_l\}) = E^{en}\psi^n_0(\{\mathbf{R}_l\}). \quad (2.10)\end{aligned}$$

This approximation is justified by the presence of the factor $\frac{1}{M_i}$, which causes the non-adiabatic coupling terms to be small due to the large value of M_i. However this holds only as long as there are no electronic energies ϵ_i being nearly degenerate. This can be seen by rewriting such a non-adiabatic coupling term:

$$\begin{aligned}&\int \mathrm{d}\mathbf{r}_1\cdots\int \mathrm{d}\mathbf{r}_n\,\Phi_j(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\nabla_{\mathbf{R}_i}\Phi_k(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\\ &\quad=\frac{1}{\epsilon_j-\epsilon_k}\int \mathrm{d}\mathbf{r}_1\cdots\int \mathrm{d}\mathbf{r}_n\,\Phi_j(\{\mathbf{R}_l\},\{\mathbf{r}_l\})[\mathcal{H}(\{\mathbf{R}_l\},\{\mathbf{r}_l\}),\nabla_{\mathbf{R}_i}]\Phi_k(\{\mathbf{R}_l\},\{\mathbf{r}_l\}). \quad (2.11)\end{aligned}$$

Since the result of the commutator is given by

$$[\mathcal{H}(\{\mathbf{R}_l\},\{\mathbf{r}_l\}),\nabla_{\mathbf{R}_i}] = Z_i \sum_{j=1}^{n} \frac{\mathbf{r}_j - \mathbf{R}_i}{|\mathbf{r}_j - \mathbf{R}_i|^3} \tag{2.12}$$

and the numerator in (2.11) is thus finite, it follows that these coupling terms become very large as soon as the energies ϵ_j and ϵ_k come close together.

However even with this simplification the second term in Eq. (2.10) remains still quite involved. Therefore, as a second approximation and again justified by the presence of the factor $\frac{1}{M_i}$, this term is completely discarded as well, in this way leading to

$$-\sum_{i=1}^{N} \frac{1}{2M_i} \nabla^2_{\mathbf{R}_i} \psi_0^n(\{\mathbf{R}_l\}) + \epsilon_0(\{\mathbf{R}_l\})\psi_0^n(\{\mathbf{R}_l\}) = E^{en}\psi_0^n(\{\mathbf{R}_l\}). \tag{2.13}$$

This is the final result of the so-called Born-Oppenheimer approximation [2]. The nucleonic wave function $\psi_0^n(\{\mathbf{R}_l\})$ is moving in the potential generated by the eigenvalues $\epsilon_0(\mathbf{R}_i)$ of the electronic ground state. For this reason the electronic ground state energy is also called the ground state potential energy surface or ground state Born-Oppenheimer surface.

Solving Eq. (2.13) gives the nucleonic wave function $\psi_0^n(\mathbf{R}_i)$ and the energy E^{en} of the combined system of electrons and nuclei. The combined electron-nuclei wave function is, according to Eq. (2.5), given by

$$\Psi^{en}(\{\mathbf{R}_l\},\{\mathbf{r}_l\}) = \Phi_0(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\psi_0^n(\{\mathbf{R}_l\}). \tag{2.14}$$

To conclude, the Born-Oppenheimer approximation states that one first has to solve for the electronic ground state while keeping the nuclei fixed and then use this result in order to move the nuclei.

2.2 Solving the electronic structure problem

It has been demonstrated in the previous section that within the Born-Oppenheimer approximation the electronic ground state has to be determined [1, 3] while keeping the nuclei fixed. The fundamental equation to solve this problem is the many-body Schrödinger equation

$$\mathcal{H}(\{\mathbf{R}_l\},\{\mathbf{r}_l\})\Phi(\{\mathbf{R}_l\},\{\mathbf{x}_l\}) = \epsilon(\{\mathbf{R}_l\})\Phi(\{\mathbf{R}_l\},\{\mathbf{x}_l\}). \tag{2.15}$$

Since electrons are not only characterized by their position $\mathbf{r}_l$, but also by their spin s_l, the combined variable $\mathbf{x}_l = (\mathbf{r}_l s_l)$ has been introduced. The Hamiltonian $\mathcal{H}(\{\mathbf{R}_l\}, \{\mathbf{r}_l\})$ is the same as in Eq. (2.3) and is independent of the spin.

The wave function $\Phi(\{\mathbf{R}_l\}, \{\mathbf{x}_l\})$ is normalized to one,

$$\int \mathrm{d}\mathbf{r}_1 \cdots \int \mathrm{d}\mathbf{r}_n \, |\Phi(\{\mathbf{R}_l\}, \{\mathbf{x}_l\})|^2 = 1, \tag{2.16}$$

and is – due to the nature of electrons being fermions and thus obeying the Pauli exclusion principle [4] – required to be antisymmetric with respect to the exchange of two electrons:

$$\Phi(\{\mathbf{R}_l\}, \mathbf{x}_1, \ldots \mathbf{x}_i, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_n) = -\Phi(\{\mathbf{R}_l\}, \mathbf{x}_1, \ldots \mathbf{x}_j, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_n). \tag{2.17}$$

The ground state of the electronic many-body system is given by the variational principle,

$$\epsilon_0 = \min_{\Phi} \langle \Phi(\{\mathbf{R}_l\}, \{\mathbf{x}_l\}) | \mathcal{H}(\{\mathbf{R}_l\}, \{\mathbf{r}_l\}) | \Phi(\{\mathbf{R}_l\}, \{\mathbf{x}_l\}) \rangle \,, \tag{2.18}$$

under the constraints (2.16) and (2.17).

What makes the solution of the electronic structure problem so difficult is its high dimensionality. A wave function $\Phi(\{\mathbf{R}_l\}, \{\mathbf{x}_l\})$ describing a system of n electrons is a quantity of dimension $4n$, which makes it impossible to work directly with it.

Instead of writing the energy in terms of the wave function it is also possible to express it in terms of so-called density matrices. This is a completely equivalent concept and will be used extensively in the context of the linear scaling algorithm.
The density matrix of a many-electron quantum state which is described by the many-electron wave function $\Phi(\{\mathbf{R}_l\}, \{\mathbf{x}_l\})$ is defined as

$$\gamma_n(\{\mathbf{R}_l\}; \mathbf{x}_1, \ldots, \mathbf{x}_n; \mathbf{x}'_1, \ldots, \mathbf{x}'_n) = \Phi(\{\mathbf{R}_l\}, \mathbf{x}_1, \ldots, \mathbf{x}_n) \Phi(\{\mathbf{R}_l\}, \mathbf{x}'_1, \ldots, \mathbf{x}'_n). \tag{2.19}$$

Furthermore it is useful to introduce the so-called reduced density matrices of first and second order:

$$\begin{aligned}
&\gamma_1(\{\mathbf{R}_l\}; \mathbf{x}_1; \mathbf{x}'_1) = n \int \mathrm{d}\mathbf{x}_2 \cdots \int \mathrm{d}\mathbf{x}_n \, \Phi(\{\mathbf{R}_l\}, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) \Phi(\{\mathbf{R}_l\}, \mathbf{x}'_1, \mathbf{x}_2, \ldots, \mathbf{x}_n), \\
&\gamma_2(\{\mathbf{R}_l\}; \mathbf{x}_1, \mathbf{x}_2; \mathbf{x}'_1, \mathbf{x}'_2) \\
&\qquad = n(n-1) \int \mathrm{d}\mathbf{x}_3 \cdots \int \mathrm{d}\mathbf{x}_n \, \Phi(\{\mathbf{R}_l\}, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_n) \Phi(\{\mathbf{R}_l\}, \mathbf{x}'_1, \mathbf{x}'_2, \mathbf{x}_3, \ldots, \mathbf{x}_n).
\end{aligned} \tag{2.20}$$

The spin charge density $\gamma(\mathbf{x})$, which will be of great importance later on, is given by the diagonal part of the reduced density matrix of first order, i.e.

$$\gamma(\mathbf{x}) = \gamma_1(\mathbf{x}; \mathbf{x}). \tag{2.21}$$

With these definitions all energy contributions appearing in the Hamiltonian of Eq. (2.3) can be expressed in terms of the reduced density matrices of first and second order:

$$\begin{aligned}
E_{n-n} &= \sum_{i=1}^{N}\sum_{j=1}^{i-1}\int \mathrm{d}\mathbf{x}_1\cdots\int \mathrm{d}\mathbf{x}_n\,\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\frac{Z_iZ_j}{|\mathbf{R}_i-\mathbf{R}_j|}\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\\
&= \sum_{i=1}^{N}\sum_{j=1}^{i-1}\frac{Z_iZ_j}{|\mathbf{R}_i-\mathbf{R}_j|}, \qquad (2.22a)\\
E_{kin} &= -\frac{1}{2}\sum_{i=1}^{n}\int \mathrm{d}\mathbf{x}_1\cdots\int \mathrm{d}\mathbf{x}_n\,\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\nabla^2_{\mathbf{r}_i}\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\\
&= -\frac{n}{2}\int \mathrm{d}\mathbf{x}_1\cdots\int \mathrm{d}\mathbf{x}_n\,\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\nabla^2_{\mathbf{r}_1}\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\\
&= -\frac{1}{2}\int \mathrm{d}\mathbf{x}_1\,\nabla^2_{\mathbf{r}_1}n\int \mathrm{d}\mathbf{x}_2\cdots\int \mathrm{d}\mathbf{x}_n\,\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\Phi(\{\mathbf{r}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\\
&= -\frac{1}{2}\int \nabla_{\mathbf{r}_1}\gamma(\mathbf{x}_1;\mathbf{x}_1')\Big|_{\mathbf{x}_1=\mathbf{x}_1'}\mathrm{d}\mathbf{x}_1, \qquad (2.22b)\\
E_{e-e} &= \sum_{i=1}^{n}\sum_{j=1}^{i-1}\int \mathrm{d}\mathbf{x}_1\cdots\int \mathrm{d}\mathbf{x}_n\,\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|}\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\\
&= \frac{n(n-1)}{2}\int \mathrm{d}\mathbf{x}_1\cdots\int \mathrm{d}\mathbf{x}_n\,\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\frac{1}{|\mathbf{r}_1-\mathbf{r}_2|}\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\\
&= \int \mathrm{d}\mathbf{x}_1\int \mathrm{d}\mathbf{x}_2\,\frac{1}{|\mathbf{r}_1-\mathbf{r}_2|}\frac{n(n-1)}{2}\\
&\quad\times\int \mathrm{d}\mathbf{x}_3\cdots\int \mathrm{d}\mathbf{x}_n\,\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\\
&= \iint \frac{1}{|\mathbf{r}_1-\mathbf{r}_2|}\gamma_2(\mathbf{x}_1,\mathbf{x}_2;\mathbf{x}_1,\mathbf{x}_2)\,\mathrm{d}\mathbf{x}_1\mathrm{d}\mathbf{x}_2, \qquad (2.22c)\\
E_{e-n} &= -\sum_{i=1}^{n}\sum_{j=1}^{N}\int \mathrm{d}\mathbf{x}_1\cdots\int \mathrm{d}\mathbf{x}_n\,\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\frac{Z_j}{|\mathbf{r}_i-\mathbf{R}_j|}\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\\
&= -n\sum_{j=1}^{N}\int \mathrm{d}\mathbf{x}_1\cdots\int \mathrm{d}\mathbf{x}_n\,\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\frac{Z_j}{|\mathbf{r}_1-\mathbf{R}_j|}\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\\
&= -\sum_{j=1}^{N}\int \mathrm{d}\mathbf{x}_1\,\frac{Z_j}{|\mathbf{r}_1-\mathbf{R}_j|}n\int \mathrm{d}\mathbf{x}_2\cdots\int \mathrm{d}\mathbf{x}_n\,\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\Phi(\{\mathbf{R}_l\},\mathbf{x}_1,\ldots,\mathbf{x}_n)\\
&= -\sum_{j=1}^{N}\int \frac{Z_j}{|\mathbf{r}_1-\mathbf{R}_j|}\gamma(\mathbf{x}_1;\mathbf{x}_1)\,\mathrm{d}\mathbf{x}_1. \qquad (2.22d)
\end{aligned}$$

The nuclei-nuclei interaction is given by the classical expression due to the normalization condition of Eq. (2.16). For the other terms the symmetry of the wave function –

see Eq. (2.17) – was employed to get rid of the sum. Furthermore integration by parts was used in order to shift the Laplace operator in the derivation of the kinetic energy.

One might wonder why the dimensionality problem can not be solved by the introduction of the density matrices, since the energy – which used to be expressed via the $4n$-dimensional wave function – is now expressed via the reduced density matrices of first and second order which have only dimension 4 and 8, respectively.
However this is not as simple as it might seem since there is the hidden constraint that these density matrices can be obtained from a n-electron wave function. This constraint is known as the n-representability problem. Whereas there is no known criterion which can ensure that a second order density matrix is n-representable, it can be shown [5,6] that for the first oder density matrix the eigenvalues must be in the interval $[0,1]$.

Due to these difficulties, it is – except for the most simple examples – not possible to exactly solve the electronic structure problem. Therefore one has to introduce some additional approximations. Popular choices for these approximate methods are Hartree-Fock (HF), Møller-Plesset perturbation theory of various order (MP2, MP3, MP4), Configuration-Interaction of various accuracy (CISD, CISD(T)) and Coupled Cluster of various accuracy (CCSD, CCSD(T)) [3]. Hartree-Fock is the fastest, but also the least accurate of these methods, whereas Coupled Cluster and Configuration Interaction are the most accurate, but also the most expensive ones. Møller-Plesset perturbation theory lies in between them from the viewpoint of both the accuracy and the cost.

A general problem of all these methods is their bad scaling which ranges from N^4 for HF over N^5 for MP2, N^6 for MP3, CISD and CCSD to N^7 for MP4, CISD(T) and CCSD(T), where N is the size of the basis set. Consequently these methods, in particular the more accurate ones, are only applicable to very small systems.
Even though there exist variants of these wave function methods which exhibit a linear scaling with respect to the number of atoms [7–9], the bad scaling with respect to the size of the basis set persists; consequently these approaches can in practice only be used in connection with a small basis set and are thus limited in accuracy.

An alternative to these methods is Density Functional Theory, which will be presented in detail in the following.

2.3 Some basics about Density Functional Theory

Density Functional Theory (DFT) [1, 10] is a very popular method to solve the electronic structure problem since it gives reasonable accuracy at moderate computational

costs. The scaling is proportional to the cube of the system size – this property will be analyzed in more detail in Sec. 2.4 –, which is better than all other methods that have briefly been mentioned in the previous section. Still the accuracy one gets with DFT is usually better than that of Hartree-Fock, which is the most favorable of these approaches from the viewpoint of the scaling.

For the remaining part of the discussion spin will be ignored for the sake of simplicity. Anyway spin can be neglected for the important class of closed shell systems which contain an even number of electrons; in this case one can get rid of the spin dependency by an integration over this degree of freedom. As an example, the spinless reduced density matrix of first order is given by

$$\rho_1(\mathbf{r}_1;\mathbf{r}_1') = \int \gamma_1(\mathbf{r}_1 s_1;\mathbf{r}_1' s_1)\,\mathrm{d}s_1. \tag{2.23}$$

Thus the many-body wave function depends only on $3n$ and the charge density on 3 spatial coordinates.
Also the condition for the first order density matrix to be n-representable is different for a closed shell system: Instead of lying in the interval $[0,1]$ the eigenvalues must now be contained in the interval $[0,2]$.

2.3.1 The Hohenberg-Kohn theorems

The fundamental basis upon which DFT is built is the first Hohenberg-Kohn theorem [11] which states the following: *The ground-state density $\rho_0(\mathbf{r})$ uniquely determines the potential, up to an arbitrary constant*.

To demonstrate this theorem the electronic Hamiltonian of Eq. (2.3) is first split up into its various contributions, namely the kinetic energy of the electrons $\mathcal{T}$, the electron-electron repulsion $\mathcal{V}_{ee}$ and the external potential represented by the one-body operator $\mathcal{V}_{ext}$:

$$\begin{aligned}
\mathcal{H} &= \mathcal{T} + \mathcal{V}_{ee} + \mathcal{V}_{ext},\\
\mathcal{T} &= -\frac{1}{2}\sum_{i=1}^{n} \nabla_{\mathbf{r}_i}^2,\\
\mathcal{V}_{ee} &= \sum_{i=1}^{n}\sum_{j=1}^{i-1} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|},\\
\mathcal{V}_{ext} &= \sum_{i=1}^{n} v_{ext}(\mathbf{r}_i).
\end{aligned} \tag{2.24}$$

If there is no external field present, the external potential is simply given by the potential generated by the nuclei, i.e. $v_{ext}(\mathbf{r}_i) = \sum_{j=1}^{N} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$. The interaction among the nuclei was removed from the Hamiltonian since it is not relevant for the electronic structure problem.
The proof of the first Hohenberg-Kohn theorem is done by contradiction. To this end one first assumes that there exist two external potentials $\mathcal{V}_{ext}^{(1)}$ and $\mathcal{V}_{ext}^{(2)}$ that differ by more than a constant and that give rise to the same ground state density. These two potentials would define two different Hamiltonians $\mathcal{H}^{(1)}$ and $\mathcal{H}^{(2)}$ with two different ground states $\Phi^{(1)}$ and $\Phi^{(2)}$. Since $\Phi^{(2)}$ is not the ground state for $\mathcal{H}^{(1)}$ it follows from the variational principle that

$$\langle \Phi^{(2)} | \mathcal{T} + \mathcal{V}_{ee} + \mathcal{V}_{ext}^{(1)} | \Phi^{(2)} \rangle > \langle \Phi^{(1)} | \mathcal{T} + \mathcal{V}_{ee} + \mathcal{V}_{ext}^{(1)} | \Phi^{(1)} \rangle \,. \tag{2.25}$$

The strict inequality in this equation is justified by the assumption that the ground state is non-degenerate. Since both wave functions yield the same charge density it follows from Eq. (2.22d) that (2.25) simplifies to

$$\langle \Phi^{(2)} | \mathcal{T} + \mathcal{V}_{ee} | \Phi^{(2)} \rangle > \langle \Phi^{(1)} | \mathcal{T} + \mathcal{V}_{ee} | \Phi^{(1)} \rangle \,. \tag{2.26}$$

However it is absolutely arbitrary which wave function is called 1 and which 2; therefore it is equally valid to write

$$\langle \Phi^{(1)} | \mathcal{T} + \mathcal{V}_{ee} | \Phi^{(1)} \rangle > \langle \Phi^{(2)} | \mathcal{T} + \mathcal{V}_{ee} | \Phi^{(2)} \rangle \,. \tag{2.27}$$

Adding Eqs. (2.26) and (2.27) yields

$$\langle \Phi^{(1)} + \Phi^{(2)} | \mathcal{T} + \mathcal{V}_{ee} | \Phi^{(1)} + \Phi^{(2)} \rangle > \langle \Phi^{(1)} + \Phi^{(2)} | \mathcal{T} + \mathcal{V}_{ee} | \Phi^{(1)} + \Phi^{(2)} \rangle \,, \tag{2.28}$$

which is a contradiction.
Consequently the assumption that there exist two external potentials that still yield the same density was wrong, thereby proofing the theorem.

This result is quite remarkable. As can be seen from Eq. (2.24) the Hamiltonian is fully determined by the ground state density (up to a constant shift) due to the first Hohenberg-Kohn theorem. As a consequence also the many-body wave functions for the ground state and all excited states are fully determined by the ground state density. Since the system is completely characterized by these wave functions, it follows that all its properties are uniquely determined by the ground state density.

In spite of the striking consequences of the first Hohenberg-Kohn theorem it does not provide a means to determine the ground state density. This issue is addressed by the second Hohenberg-Kohn theorem.

To this end the variational principle is considered again, which tells that the ground state is given by minimizing the energy over all wave functions Φ:

$$E = \min_{\Phi} \langle \Phi | \mathcal{T} + \mathcal{V}_{ee} + \mathcal{V}_{ext} | \Phi \rangle \,. \tag{2.29}$$

The minimization over Φ can now be split up into an outer loop minimizing over all densities ρ and an inner loop minimizing over all wave functions Φ yielding the charge density ρ [12]:

$$E = \min_{\rho} \left[\min_{\Phi \to \rho} \langle \Phi | \mathcal{T} + \mathcal{V}_{ee} + \mathcal{V}_{ext} | \Phi \rangle \right] . \tag{2.30}$$

The external potential depends only on the density and can therefore be taken out of the inner minimization loop, leading to

$$E = \min_{\rho} \left[\min_{\Phi \to \rho} \langle \Phi | \mathcal{T} + \mathcal{V}_{ee} | \Phi \rangle + \int \mathcal{V}_{ext}(\mathbf{r}) \rho(\mathbf{r}) \, \mathrm{d}\mathbf{r} \right] . \tag{2.31}$$

From the last equation it becomes clear that for a given density ρ, the ground state wave function is the one which minimizes $\mathcal{T} + \mathcal{V}_{ee}$ and yields ρ. Since this minimization does not depend on the external potential, it has to be a universal result for a given density. Thus it is possible to define the universal functional

$$F[\rho] = \min_{\Phi \to \rho} \langle \Phi | \mathcal{T} + \mathcal{V}_{ee} | \Phi \rangle \tag{2.32}$$

and to write the ground state energy as

$$E = \min_{\rho} \left[F[\rho] + \int \mathcal{V}_{ext}(\mathbf{r}) \rho(\mathbf{r}) \, \mathrm{d}\mathbf{r} \right] . \tag{2.33}$$

This demonstrates that the density obeys a variational principle and that the ground state density is the one which minimizes Eq. (2.33).
These last results are known as the second Hohenberg-Kohn theorem.

If the exact form of the functional $F[\rho]$ was known, it would be possible to directly use Eq. (2.33) in order to minimize the energy under the constraint of a fixed number of particles $n = \int \rho(\mathbf{r}) \, \mathrm{d}\mathbf{r}$, i.e. to minimize $E[\rho] - \mu n$, where $\mu = \partial E / \partial n$ is the chemical potential of the system. This would then lead to the Euler-Lagrange equation

$$\frac{\delta F[\rho]}{\delta \rho(\mathbf{r})} + \mathcal{V}_{ext}(\mathbf{r}) = \mu. \tag{2.34}$$

Unfortunately such a functional form for F is not known and as a consequence DFT calculations are usually done in the framework of Kohn-Sham DFT. Still Eq. (2.34) will be used later.

2.3.2 The Kohn-Sham formalism of DFT

In the Kohn-Sham formulation of DFT [13], the system of n interacting electrons is replaced by a system of n non-interacting quasi-electrons. The Kohn-Sham ansatz is based on two fundamental assumptions:

1. The exact ground state density emerging from the system of interacting electrons can be represented by the ground state density of the system of non-interacting quasi-electrons. This assumption is called "non-interacting-V-representability".
2. The Kohn-Sham Hamiltonian consists of the kinetic energy operator and an effective one-body potential operator $\tilde{\mathcal{V}}$.

These n independent quasi-electrons give rise to n orthonormal single-particle orbitals $\phi_i(\mathbf{r})$, out of which the many-electron wave function $\tilde{\Phi}(\mathbf{r}_1, \ldots, \mathbf{r}_n)$ can be constructed as one single Slater determinant [14]:

$$\tilde{\Phi}(\mathbf{r}_1, \ldots, \mathbf{r}_n) = \frac{1}{\sqrt{n!}} \begin{vmatrix} \phi_1(\mathbf{r}_1) & \cdots & \phi_n(\mathbf{r}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{r}_n) & \cdots & \phi_n(\mathbf{r}_n) \end{vmatrix}. \tag{2.35}$$

The tilde is used to distinguish between this wave function being constructed from single-particle orbitals and the true many-body wave functions Φ. The ansatz (2.35) automatically fulfills the normalization of Eq. (2.16) and the antisymmetry condition of Eq. (2.17).

The single particle density matrix of first order $\gamma_1(\mathbf{r}_1, \mathbf{r}_1')$, which is defined by Eq. (2.20), can in this case be directly expressed via the single particle orbitals.
As an example the density matrix for the case of two electrons is explicitly calculated. For such a system the wave function $\tilde{\Phi}(\mathbf{r}_1, \mathbf{r}_2)$ is given by

$$\tilde{\Phi}(\mathbf{r}_1, \mathbf{r}_2) = \frac{1}{\sqrt{2}} \left[\phi_1(\mathbf{r}_1)\phi_2(\mathbf{r}_2) - \phi_1(\mathbf{r}_2)\phi_2(\mathbf{r}_1)\right] \tag{2.36}$$

and the reduced density matrix of first order consequently by

$$\begin{aligned} \gamma_1(\mathbf{r}_1; \mathbf{r}_1') &= 2 \int \Phi(\mathbf{r}_1, \mathbf{r}_2)\Phi(\mathbf{r}_1', \mathbf{r}_2)\, \mathrm{d}\mathbf{r}_2 \\ &= \int \left[\phi_1(\mathbf{r}_1)\phi_2(\mathbf{r}_2) - \phi_1(\mathbf{r}_2)\phi_2(\mathbf{r}_1)\right] \left[\phi_1(\mathbf{r}_1')\phi_2(\mathbf{r}_2) - \phi_1(\mathbf{r}_2)\phi_2(\mathbf{r}_1')\right] \mathrm{d}\mathbf{r}_2 \\ &= \int \phi_1(\mathbf{r}_1)\phi_2(\mathbf{r}_2)\phi_1(\mathbf{r}_1')\phi_2(\mathbf{r}_2)\, \mathrm{d}\mathbf{r}_2 + \int \phi_1(\mathbf{r}_2)\phi_2(\mathbf{r}_1)\phi_1(\mathbf{r}_2)\phi_2(\mathbf{r}_1')\, \mathrm{d}\mathbf{r}_2 \\ &\quad - \int \phi_1(\mathbf{r}_1)\phi_2(\mathbf{r}_2)\phi_1(\mathbf{r}_2)\phi_2(\mathbf{r}_1')\, \mathrm{d}\mathbf{r}_2 - \int \phi_1(\mathbf{r}_2)\phi_2(\mathbf{r}_1)\phi_1(\mathbf{r}_1')\phi_2(\mathbf{r}_2)\, \mathrm{d}\mathbf{r}_2 \\ &= \phi_1(\mathbf{r}_1)\phi_1(\mathbf{r}_1') + \phi_2(\mathbf{r}_1)\phi_2(\mathbf{r}_1'), \end{aligned} \tag{2.37}$$

where the orthonormality of the single-particle orbitals, $\int \phi_i(\mathbf{r})\phi_j(\mathbf{r})\,\mathrm{d}\mathbf{r} = \delta_{ij}$, was used. For a system consisting of n electrons the same arguments apply and the density matrix is thus given by

$$\gamma_1(\mathbf{r}_1;\mathbf{r}_1') = \sum_{i=1}^{n} \phi_i(\mathbf{r}_1)\phi_i(\mathbf{r}_1'). \tag{2.38}$$

It follows from this result that the charge density, which is the diagonal part of the reduced density matrix of first order, can be calculated according to

$$\rho(\mathbf{r}) = \sum_{i=1}^{n} |\phi_i(\mathbf{r})|^2. \tag{2.39}$$

The next step is to write down an equation determining the single-particle orbitals ϕ_i and to find the form of the one-body potential $\tilde{\mathcal{V}}$.
To this end the variational principle for the many-body wave function – i.e. Eq. (2.29) – is rewritten for the case of the non-interacting electrons:

$$E = \min_{\tilde{\Phi}} \langle \tilde{\Phi}|\mathcal{T} + \tilde{\mathcal{V}}|\tilde{\Phi}\rangle\,. \tag{2.40}$$

From this it cannot be concluded – by comparing with (2.29) – that the potential is given by $\tilde{\mathcal{V}} = \mathcal{V}_{ee} + \mathcal{V}_{ext}$, since there is the constraint that the many-body wave function $\tilde{\Phi}$ is a Slater determinant constructed out of the single-particle orbitals ϕ_i; consequently $\langle\tilde{\Phi}|\mathcal{T}|\tilde{\Phi}\rangle$ is the kinetic energy of the system of non-interacting particles and is not necessarily identical to the true kinetic energy for the interacting system. This becomes also visible by explicitly writing the energy in terms of the single-particle orbitals, which follows by inserting (2.35) into Eq. (2.40) and carrying out the similar steps as for the derivation of the density matrix in (2.37):

$$E = \min_{\{\phi_1,\ldots,\phi_n\}} \sum_i \langle \phi_i(\mathbf{r})| -\frac{1}{2}\nabla^2 + \tilde{\mathcal{V}}|\phi_i(\mathbf{r})\rangle\,. \tag{2.41}$$

This means that the kinetic energy of the system of non-interacting particles can be expressed via the single-particle orbitals as $E_{kin} = \sum_i \langle\phi_i| - \frac{1}{2}\nabla^2|\phi_i\rangle$. The difference to the true kinetic energy $\langle\Phi|\mathcal{T}|\Phi\rangle$ has consequently to be hidden in the potential operator $\tilde{\mathcal{V}}$.

By building the functional derivatives $\delta E/\delta\phi_i$ under the normalization constraint $\langle\phi_i|\phi_i\rangle = 1$ it follows from Eq. (2.41) that the single-particle orbitals ϕ_i are given by the solution of the eigenvalue equation

$$\left(-\frac{1}{2}\nabla^2 + \tilde{\mathcal{V}}(\mathbf{r})\right)\phi_i(\mathbf{r}) = \epsilon_i\phi_i(\mathbf{r}). \tag{2.42}$$

Defining the Kohn-Sham Hamiltonian as

$$\mathcal{H}_{KS}(\mathbf{r}) = -\frac{1}{2}\nabla^2 + \tilde{\mathcal{V}}(\mathbf{r}) \tag{2.43}$$

the eigenvalue problem of (2.42) can thus be written as

$$\mathcal{H}_{KS}\phi_i(\mathbf{r}) = \epsilon_i\phi_i(\mathbf{r}). \tag{2.44}$$

From now on the single-particle orbitals ϕ_i and the corresponding eigenvalues ϵ_i will be called Kohn-Sham orbitals and Kohn-Sham eigenvalues, respectively.

What remains is the determination of the potential $\tilde{\mathcal{V}}$. Starting from Eq. (2.40) and applying the same steps as in the derivation of the second Hohenberg-Kohn theorem leads to the expression

$$E = \min_{\rho}\left[\tilde{T}[\rho] + \int \tilde{\mathcal{V}}(\mathbf{r})\rho(\mathbf{r})\,\mathrm{d}\mathbf{r}\right], \tag{2.45}$$

where the functional $\tilde{T}[\rho]$ gives the kinetic energy of the non-interacting particles and is defined as

$$\tilde{T}[\rho] = \min_{\tilde{\Phi}\to\rho}\langle\tilde{\Phi}|\mathcal{T}|\tilde{\Phi}\rangle\,. \tag{2.46}$$

Put into words, the Kohn-Sham wave function $\tilde{\Phi}$ for a given density $\rho(\mathbf{r})$ is consequently that wave function which minimizes the kinetic energy while yielding $\rho(\mathbf{r})$.
From Eq. (2.45) an Euler-Lagrange equation similar to (2.34) can now readily be derived and is given by

$$\frac{\delta\tilde{T}[\rho]}{\delta\rho(\mathbf{r})} + \tilde{\mathcal{V}} = \mu. \tag{2.47}$$

Keeping this result in mind, the next step is to rewrite the functional for the system of interacting electrons – i.e. (2.32) – in terms of the kinetic energy of the system of non-interacting particles and a remainder which is split up in the Hartree energy $U[\rho] = \frac{1}{2}\int\frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}'$ and the unknown exchange-correlation energy $E_{XC}[\rho]$. This last quantity represents the difference between the true kinetic energy and the one obtained from the single particle orbitals as well as the non-classical electron-electron interaction which is not present in the Hartree term. Consequently one can write

$$F[\rho] = \tilde{T}[\rho] + U[\rho] + E_{XC}[\rho]. \tag{2.48}$$

Inserting this into Eq. (2.34) and defining the exchange-correlation potential as $v_{XC}(\mathbf{r}) = \frac{\delta E_{XC}[\rho]}{\delta\rho(\mathbf{r})}$ yields

$$\frac{\delta\tilde{T}[\rho]}{\delta\rho(\mathbf{r})} + \int\frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,\mathrm{d}\mathbf{r}' + v_{XC}(\mathbf{r}) + \mathcal{V}_{ext}(\mathbf{r}) = \mu. \tag{2.49}$$

By comparing this result with Eq. (2.47) one gets an expression for the potential of the system of non-interacting particles:

$$\tilde{\mathcal{V}}(\mathbf{r}) = \mathcal{V}_{ext}(\mathbf{r}) + \int\frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,\mathrm{d}\mathbf{r}' + v_{XC}(\mathbf{r}). \tag{2.50}$$

It has to be noted that the sum of the Kohn-Sham eigenvalues, the so called band-structure energy

$$E_{BS} = \sum_{i=1}^{n} \langle \phi_i | \mathcal{H}_{KS} | \phi_i \rangle = \sum_{i=1}^{n} \epsilon_i, \tag{2.51}$$

is not identical to the total energy of the system, which is – according to Eqs. (2.33) and (2.48) – given by

$$E = -\frac{1}{2} \sum_{i=1}^{n} \int \phi_i(\mathbf{r}) \nabla^2 \phi_i(\mathbf{r}) \, \mathrm{d}\mathbf{r} + \int \mathcal{V}_{ext}(\mathbf{r}) \rho(\mathbf{r}) \, \mathrm{d}\mathbf{r} + \frac{1}{2} \iint \frac{\rho(\mathbf{r}) \rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \, \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}' + E_{XC}[\rho(\mathbf{r})]. \tag{2.52}$$

Comparing this with the Kohn-Sham Hamiltonian $\mathcal{H}_{KS}$ of Eq. (2.43), it follows that the total energy is related to the band-structure energy via

$$E = \sum_{i=1}^{n} \langle \phi_i | \mathcal{H}_{KS} | \phi_i \rangle - \frac{1}{2} \iint \frac{\rho(\mathbf{r}) \rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \, \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}' + E_{XC}[\rho(\mathbf{r})] - \int v_{XC}(\mathbf{r}) \rho(\mathbf{r}) \, \mathrm{d}\mathbf{r}. \tag{2.53}$$

The big unsolved problem of the Kohn-Sham formalism is that the exact form of the exchange-correlation functional $E_{XC}[\rho]$ is unknown. Therefore one has to use approximations to it.

2.3.2.1 Strategies for solving the Kohn-Sham equations

There are two possibilities to determine the Kohn-Sham orbitals. Either one directly solves the eigenvalue equation (2.44) by diagonalizing the Hamiltonian represented in a certain basis, or one iteratively minimizes the band-structure energy (2.51).
Both approaches will eventually lead to the same result. However the direct diagonalization is only feasible if the basis set is reasonably small.

Furthermore it must be noted that both approaches need to determine the solution in a self-consistent way, meaning that the density that one obtains from the final orbitals according to Eq. (2.39) must be identical to the one used for the construction of the potential (2.50) – and thus of the Hamiltonian – which has led to this solution.

If the system exhibits a large enough band gap, this condition will eventually be met if one directly updates the orbitals by minimizing the total energy [15] and straightforwardly reuses the new charge density as the input for the construction of a new potential and thus a new Hamiltonian.
For metallic systems, on the other hand, it is often required to use more orbitals than electrons (respectively more than half the number of electrons in the case of a closed shell system) and to smear out the Fermi surface with a finite electronic temperature [16, 17], in this way assigning fractional occupation numbers to the orbitals which

will then as well enter the calculation of the charge density. In such situations one first has to perform a few minimization steps for the expression $\sum_{i=1}^{n} \langle \phi_i | \mathcal{H}_{KS} | \phi_i \rangle$ using a fixed Hamiltonian, followed by an update of the occupation numbers and a mixing of the new charge density with the old one. The resulting charge density is then the input for the evaluation of the new potential and thus the construction of the new Hamiltonian. If the size of the basis set is small enough, the few minimization steps at a fixed potential can be replaced by a diagonalization of the Hamiltonian matrix in this basis.

2.3.3 Exchange-Correlation functionals

The simplest approximation to the unknown functional $E_{XC}[\rho]$ is the so-called Local Density approximation (LDA), which gives – in spite of its crudeness – remarkably good results. This approximation makes the assumption that the system under investigation can reasonably well be described by a homogeneous electron gas with the same charge density, where the nuclei are replaced by a uniform positively charged background. The LDA approximation is therefore by construction exact for the uniform electron gas.

The LDA exchange-correlation energy for a system with the charge density $\rho(\mathbf{r})$ is given by

$$E_{XC}^{LDA}[\rho(\mathbf{r})] = \int \rho(\mathbf{r}) \epsilon_{XC}^{hom}(\rho(\mathbf{r}))\, \mathrm{d}\mathbf{r}, \tag{2.54}$$

where $\epsilon_{XC}^{hom}(\rho(\mathbf{r}))$ is the exchange-correlation energy density of a homogeneous electron gas with the same charge density. The value of the exchange correlation functional is consequently completely local.
ϵ_{XC}^{hom} is further split up in an exchange part and a correlation part. Whereas the exchange part can be calculated analytically and is given by

$$\epsilon_{X}^{hom}(\rho(\mathbf{r})) = -\frac{3}{4}\left(\frac{3}{\pi}\rho(\mathbf{r})\right)^{1/3}, \tag{2.55}$$

the correlation part cannot be determined exactly. Furthermore there are different approximations for the case of high [18,19] and low [20,21] electronic densities.

LDA gives accurate results for systems that resemble the homogeneous electron gas, i.e. systems with charge densities which are only slowly varying, for instance solids. For systems where this condition is not fulfilled, for instance small molecules or atoms, the energy calculated with the LDA approximation is typically too high. As a consequence LDA yields in general a too large binding energies; furthermore bond lengths are typically underestimated.

An improvement of the accuracy can be reached by the so-called Generalized-Gradient Approximation (GGA) functionals. This class takes into account not only the density at a given point, but also its gradient:

$$E_{XC}^{GGA}[\rho(\mathbf{r})] = \int \rho(\mathbf{r})\epsilon_{XC}(\rho(\mathbf{r}), \nabla\rho(\mathbf{r}))\,\mathrm{d}\mathbf{r}. \tag{2.56}$$

More explicitly this is often written as

$$E_{XC}^{GGA}[\rho(\mathbf{r})] = \int \rho(\mathbf{r})\epsilon_X^{hom}(\rho(\mathbf{r}))F_{XC}(\rho(\mathbf{r}), \nabla\rho(\mathbf{r}))\,\mathrm{d}\mathbf{r}, \tag{2.57}$$

where $\epsilon_X^{hom}(\rho(\mathbf{r}))$ is again the exchange energy density of the homogeneous electron gas and F_{XC} is a dimensionless function. There are several propositions for the form of F_{XC} [22–24]; they all have in common that they yield the LDA result in the limit where the gradient is zero.

A further improvement can be reached by so-called SIC functionals, which stands for "self-interaction correction" [25, 26]. These functionals try to correct the non-physical interaction of an electron with itself that is present in standard functionals. This self-interaction stems from the Hartree term and should in principle be exactly canceled by the exchange-correlation term, but this cancellation is not perfect for most functionals.

Other important classes of functionals are the so-called meta-GGA functionals [27, 28], which depend in addition on the kinetic energy density $\sum_i \frac{1}{2}|\nabla\phi_i(\mathbf{r})|^2$, and hybrid functionals [29–31] which mix the exchange-correlation energy from DFT with some exchange energy from a Hartree-Fock calculation. With hybrid functionals one typically gets the most accurate results.

2.3.4 Pseudopotentials

In a DFT calculation a priori all electrons of a given atom have to be included in the description of the system. However it turns out that the electrons which are close to the core region are chemically inert, meaning that they are not involved in chemical reactions. Therefore it is advantageous to simulate these electrons by a so-called pseudopotential, i.e. one replaces the atomic nucleus and the core electrons by a pseudoatom whose charge is reduced by the number of core electrons.

This approach has several advantages. First of all it makes the calculation much faster simply due to the fact that the number of electrons is reduced.
Furthermore the orbitals of the core electrons would oscillate very rapidly close to the nuclei, thus requiring a very high resolution in this region. For an adaptive basis set,

as it is used in BigDFT, this would in principle be feasible, but it obviously increases the complexity a lot if many different resolution levels have to be used. For a basis that requires a uniform grid spacing over the entire simulation box the situation is even worse; here the high resolution required in the core region would make the calculation hopelessly slow.
In addition to these benefits one can make a virtue out of necessity and include relativistic effects into the pseudopotential [32] which would be absent otherwise.

Within the framework of such a pseudopotential calculation the total Kohn-Sham Hamiltonian is given by

$$\mathcal{H}_{KS} = -\frac{1}{2}\nabla^2 + \mathcal{V}_{KS}[\rho] + \mathcal{V}_{PSP} \tag{2.58}$$

with the Kohn-Sham potential

$$\mathcal{V}_{KS}[\rho] = \mathcal{V}_{ext}(\mathbf{r}) + \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,\mathrm{d}\mathbf{r}' + v_{XC}(\mathbf{r}) \tag{2.59}$$

and the pseudopotential term $\mathcal{V}_{PSP}$. In BigDFT the norm-conserving GTH-HGH pseudopotentials [33–35] are used, which consist of a local and a non-local term, i.e. $\mathcal{V}_{PSP} = \mathcal{V}_{local} + \mathcal{V}_{nonlocal}$:

$$\begin{aligned}
\mathcal{V}_{local}(\mathbf{r}) &= \frac{Z_{ion}}{r}\mathrm{erf}\left(\frac{r}{\sqrt{2}r_{loc}}\right) + e^{-r^2/2r_{loc}^2} \\
&\quad \times \left[C_1 + C_2\left(\frac{r}{r_{loc}}\right)^2 + C_3\left(\frac{r}{r_{loc}}\right)^4 + C_4\left(\frac{r}{r_{loc}}\right)^6\right], \\
\mathcal{V}_{nonlocal}(\mathbf{r}) &= \sum_l \sum_{i,j=1}^{3} h_{ij}^{(l)} \left|p_i^{(l)}\right\rangle \left\langle p_j^{(l)}\right| \\
&\quad \text{with} \quad \left\langle \mathbf{r} \middle| p_i^{(l)} \right\rangle = \frac{\sqrt{2}r^{l+2(i-1)}\exp\left[-\frac{1}{2}\left(\frac{r}{r_l}\right)^2\right]}{r_l^{l+(4i-1)/2}\sqrt{\Gamma\left(l+\frac{4i-1}{2}\right)}} \sum_{m=-l}^{l} Y_{l,m}(\theta,\phi).
\end{aligned} \tag{2.60}$$

$Y_{l,m}(\theta,\phi)$ are the spherical harmonics, r_{loc} is the localization radius of the local part and r_l the localization radius of a given projector.

It has to be noted that the electrostatic potential generated by the nuclei, which has so far been included in the external potential $\mathcal{V}_{ext}$, is now already contained in the pseudopotential term. Consequently the term $\mathcal{V}_{ext}$ in Eq. (2.59) now only describes real external potentials, e.g. an electric field.

2.4 Scaling of traditional Kohn-Sham DFT

One of the most important characteristic of any electronic structure method is – of course apart from its accuracy – the scaling with respect to the size of the system. In Sec. 2.2 it has been mentioned briefly that the scaling of popular wave function methods ranges from N^4 to N^7, where N is the size of the basis set. However, since only the scaling is noted without any absolute time, N can in principle be any measure of the system size which is directly related to the number of basis functions; a popular choice is the number of atoms.
Due to these large powers of N, calculations for big systems become extremely expensive.

Kohn-Sham DFT, on the other hand, exhibits a more favorable cubic scaling. This property will be analyzed in more detail in this section.

As shown in Sec. 2.3.2 the framework of Kohn-Sham DFT requires to solve for the single-particle orbitals ϕ_i given by Eq. (2.42). This procedure involves tasks exhibiting different scalings with respect to the size of the system, so the total time needed to calculate a system of size N can be written as

$$t_{tot}(N) = \sum_i c_i \gamma_i(N), \tag{2.61}$$

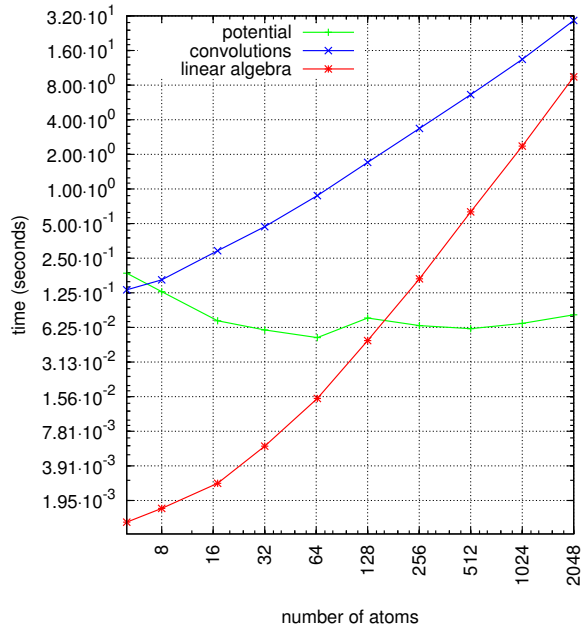
where the sum runs over all tasks and $c_i\gamma_i(N)$ is the time required by task i. $\gamma_i(N)$ gives the scaling of the task with respect to N and c_i is its prefactor that determines the absolute time. Thus for small systems the total time is mainly influenced by the magnitude of the prefactors, whereas for large systems those parts with the heaviest scaling dominate.

In the context of Kohn-Sham DFT the part with the worst scaling is related to the orthogonality that is imposed on the Kohn-Sham orbitals. Such an orthogonalization step requires to calculate the scalar product among all orbitals of the system, which is proportional to n^2 if there are n such orbitals. Since each orbital extends over the entire system, the cost of calculating one single scalar product is proportional to m, where m is the size of the basis set used to represent the orbitals. Consequently the overall scaling is proportional to n^2m. Since in general both n and m are proportional to the size of the system – represented by the number of atoms N – the scaling of the orthogonalization is proportional to N^3, i.e. $\gamma_{\text{ortho}}(N) = N^3$.

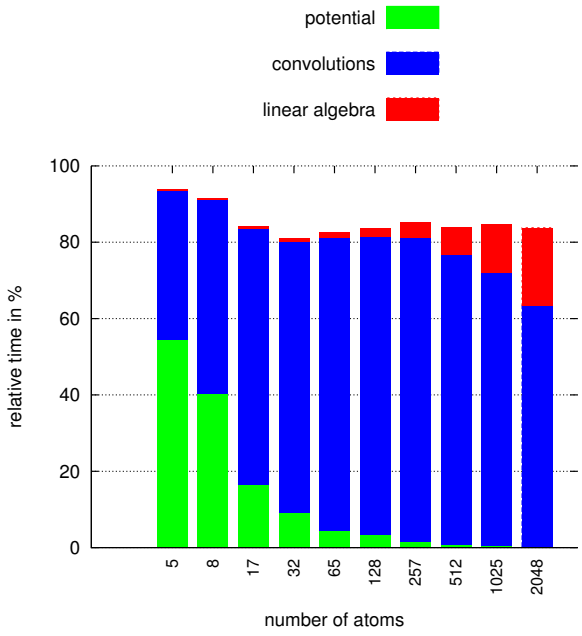
However, as already mentioned, the scaling only tells which part will preponderate for very large values of N. For smaller systems, it might well be that other parts dominate due to their larger prefactor.
To illustrate these issues the traditional cubic version of BigDFT was taken as an exam-

ple. The main tasks of this code can basically be split up in three categories. The calculation of the potential is dominated by the Poisson solver which exhibits an almost-linear $N \log N$ scaling. The next level are the quadratic parts of the convolutions which are, for instance, required to apply the kinetic energy operator or to calculate the charge density. Finally there is the linear algebra part – comprising the above mentioned orthogonalization – which has a cubic scaling. Together these three categories account for most of the total computation time; for the test case studied here their sum always amounts to more than 80% of the total time, independent of the size of the system.

The scaling of these three categories is shown in Fig. 2.1a for the case of alkane chains of various lengths. The runs were done in parallel such that each MPI task had to handle one orbital, i.e. the number of MPI tasks was directly proportional to the size



(a) The run time for the three categories potential, convolutions and linear algebra with respect to the number of atoms in an alkane chain. The time taken by the potential section remains roughly constant, whereas the convolutions scale quadratically and the linear algebra cubically.



(b) Relative times taken by the three tasks. Together they always account for more than 80% of the total time. Whereas the potential is only relevant for small systems and the linear algebra only plays an important role for very large systems, the convolutions take a large amount of time over a wide range.

Figure 2.1: Illustration of the scaling of the cubic version of BigDFT. The test was done for alkanes of different lengths; the smallest one consisted of 5 atoms, the largest one of 2048 atoms. The runs were executed in parallel such that each MPI task had to handle one orbital. Only the computation time is shown, i.e. the communication was excluded. The timings are given for one step in the minimization procedure of the energy.

of the system. Since only the computation time is plotted, i.e. neglecting the time taken by the communication, and since BigDFT exhibits a very efficient parallelization, it is therefore to be expected that the time taken by the potential section should remain roughly constant for all system sizes, whereas the time taken by the convolutions should increase linearly and that taken by the linear algebra quadratically. Since the timings are shown in a log-log plot, this should result in straight lines with slope 0, 1 and 2, respectively. As can be seen from the figure, this is actually the case as soon as a given size is reached.

The plot also gives some ideas on the prefactors which are basically given by the time taken for the smallest system. The prefactor for the potential part is the largest one, followed by the one for the convolutions, which however still has the same order of magnitude. The prefactor for the linear algebra, on the other hand, is orders of magnitude smaller.

As a consequence the relative importance of these three categories varies a lot as the size of the system is increased. This is illustrated in Fig. 2.1b, where the relative amount of time taken by these three sections is shown. As can be seen the time taken for the calculation of the potential is only relevant for very small system up to roughly 20 atoms. Due to the small prefactor for the linear algebra part there is then a very large range where the convolutions dominate, and only at around 1000 atoms the influence of the linear algebra starts to play an important role.

CHAPTER 3

Linear scaling Density Functional Theory

3.1 Theoretical background

Whereas the previous chapter has provided some insight into the basics of electronic structure calculations and the traditional Kohn-Sham ansatz of DFT, this chapter will focus on the foundations of linear scaling DFT methods.

3.1.1 Locality in DFT

If one wants to develop a method whose computational time scales only linearly with respect to the size of the system, it is necessary to make at some point the assumption that only quantities which are strictly localized are dealt with. To justify this assumption, it is in turn required that the properties of the latter ones are only weakly influenced by what is going on far away. If this condition is fulfilled, the error introduced by strictly localizing these quantities should be acceptable. This procedure is the key in developing a linear scaling algorithm.

A priori quantum mechanics is a non-local concept [1]. The wave functions that fully characterize a given system extend in general over the entire volume. An example that illustrates this non-locality is the antisymmetry of a many-electron wave function which must be fulfilled for any pair of electrons, no matter whether they are nearby or far away.

Fortunately there are however some quantities which do not directly require the determination of the extended wave functions. Examples in the context of Kohn-Sham DFT are the energy or the density matrix, which are both integrated quantities being invariant under unitary transformations among the Kohn-Sham orbitals, and which are sufficient to determine the ground state of the system. For such quantities the term "nearsightedness" has been coined by Kohn [36], meaning that their calculation at a given point $\mathbf{r}$ requires only information at points $\mathbf{r}'$ in a localized region around $\mathbf{r}$.
Consequently it should – as long as the quantities employed are well suited for this purpose – be possible to develop a fully ab-initio method that still scales only linearly with respect to the size of the system.

This concept of locality is not exploited by the standard Kohn-Sham scheme where all orbitals may extend over the entire system. One might argue that there exists a set of maximally localized Wannier orbitals which are related to the standard Kohn-Sham eigenorbitals via a unitary transformation and reflect in some sense the nearsightedness principle. Once the eigenorbitals ψ_i – the Kohn Sham orbitals will from now on be denoted by ψ and not ϕ as in the previous chapter since ϕ will get a different meaning – are found, the Wannier functions W_i can be generated as

$$W_i(\mathbf{r}) = \sum_j U_{ij}\psi_j(\mathbf{r}) \tag{3.1}$$

with a unitary matrix $\mathbf{U}$. But since this explicit construction of the Wannier functions requires first the exact shape of the extended eigenorbitals, it does not help in developing an algorithm that scales linearly with the size of the system.
Furthermore there is no simple unique prescription how the Wannier functions should be defined. A method by Marzari and Vanderbilt [37] minimizes the total spread of the orbitals $\sum_i \langle r^2\rangle - \langle \mathbf{r}\rangle_i^2$ in order to generate them. However one might also think of other criteria – minimizing the spread is just one possibility –, making the definition somehow arbitrary.

An alternative description, which is completely equivalent to using the orbitals ψ_i, but incorporates in a natural way the nearsightedness principle, is given by the use of the first order density matrix which was introduced in Eq. (2.20); from now on it will be denoted by $F(\mathbf{r},\mathbf{r}')$ instead of $\gamma_1(\{\mathbf{R}_i\};\mathbf{r}_1;\mathbf{r}_1')$. In the independent particle framework of Kohn-Sham DFT it is – according to Eq. (2.38) – given by

$$F(\mathbf{r},\mathbf{r}') = \sum_i f(\epsilon_i)\psi_i(\mathbf{r})\psi_i(\mathbf{r}'), \tag{3.2}$$

where the Fermi function $f(\epsilon_i)$ determines the occupation of the ith orbital and is given by

$$f(\epsilon_i) = \frac{1}{1+e^{(\epsilon_i-\mu)/(k_BT)}} \tag{3.3}$$

with the chemical potential μ, the Boltzmann constant k_B and the temperature T, which is in general assumed to be zero. For a system with finite gap, the density matrix of a system containing n electrons at zero temperature will only have n non-zero eigenvalues (which then have value one); consequently the density matrix has only rank n and can be constructed from the occupied states only:

$$F(\mathbf{r},\mathbf{r}') = \sum_{i=occ} \psi_i(\mathbf{r})\psi_i(\mathbf{r}'). \tag{3.4}$$

The central quantities of DFT that have been expressed so far in terms of the orbitals ψ_i,

$$\begin{aligned}
E_{kin} &= -\frac{1}{2}\sum_i f(\epsilon_i) \int \psi_i(\mathbf{r})\nabla^2\psi_i(\mathbf{r})\,\mathrm{d}\mathbf{r},\\
E_{pot} &= \sum_i f(\epsilon_i) \int \psi_i(\mathbf{r})\tilde{\mathcal{V}}(\mathbf{r})\psi(\mathbf{r})\,\mathrm{d}\mathbf{r},\\
E_{BS} &= E_{kin} + E_{pot} = \sum_i f(\epsilon_i) \int \psi_i(\mathbf{r})\mathcal{H}(\mathbf{r})\psi_i(\mathbf{r})\,\mathrm{d}\mathbf{r},\\
\rho(\mathbf{r}) &= \sum_i f(\epsilon_i)|\psi_i(\mathbf{r})|^2,
\end{aligned} \tag{3.5}$$

can – according to Eqs. (2.21) and (2.22) – also be expressed in terms of the density matrix:

$$\begin{aligned}
E_{kin} &= -\frac{1}{2}\int \nabla^2 F(\mathbf{r},\mathbf{r}')\Big|_{\mathbf{r}=\mathbf{r}'}\,\mathrm{d}\mathbf{r}',\\
E_{pot} &= \int \tilde{\mathcal{V}}(\mathbf{r}')F(\mathbf{r}',\mathbf{r}')\,\mathrm{d}\mathbf{r}',\\
E_{BS} &= E_{kin} + E_{pot} = \int \mathcal{H}(\mathbf{r}')F(\mathbf{r},\mathbf{r}')\Big|_{\mathbf{r}=\mathbf{r}'}\,\mathrm{d}\mathbf{r}',\\
\rho(\mathbf{r}) &= F(\mathbf{r},\mathbf{r}).
\end{aligned} \tag{3.6}$$

In case the above operators are discretized using a finite orthonormal basis set $\phi_\alpha(\mathbf{r})$, i.e.

$$\begin{aligned}
H_{\alpha\beta} &= \int \phi_\alpha(\mathbf{r})\mathcal{H}(\mathbf{r})\phi_\beta(\mathbf{r})\,\mathrm{d}\mathbf{r},\\
K_{\alpha\beta} &= \iint \phi_\alpha(\mathbf{r})F(\mathbf{r},\mathbf{r}')\phi_\beta(\mathbf{r}')\,\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}',
\end{aligned} \tag{3.7}$$

the band-structure energy and the total number of electrons $n = \int \rho(\mathbf{r})\,\mathrm{d}\mathbf{r}$ can be written as traces of these matrices:

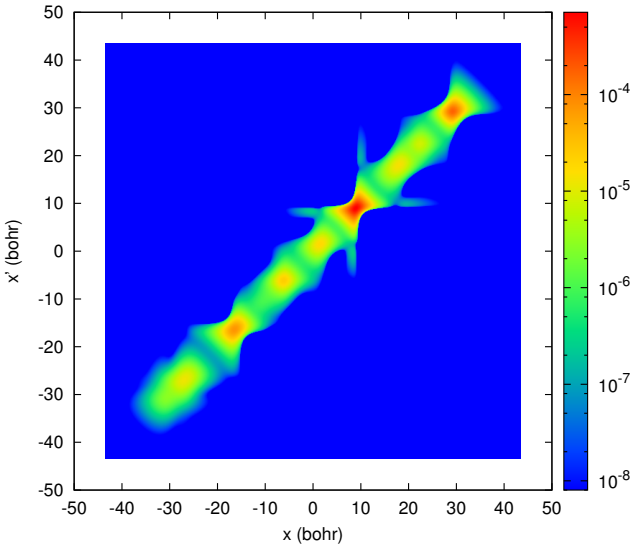
$$\begin{aligned}
E_{BS} &= \mathrm{tr}(\mathbf{KH}),\\
n &= \mathrm{tr}(\mathbf{K}).
\end{aligned} \tag{3.8}$$

It is worth noting that the second order density matrix is – in contrast to the energy expressions derived in Eqs. (2.22) – not required any more; this is a direct consequence of the independent particle framework.

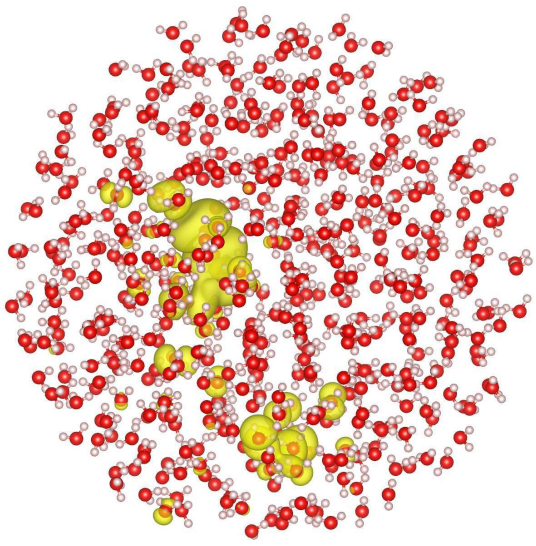
3.1.2 Decay properties of the density matrix

It has been demonstrated that for insulators and metals at finite temperature the matrix elements $F(\mathbf{r}, \mathbf{r}')$ decay exponentially with the distance $|\mathbf{r} - \mathbf{r}'|$ [38–44], whereas for metals at zero temperature they decay algebraically [45].
This property might be surprising at first sight since, according to Eq. (3.2), the density matrix can be constructed from the Kohn-Sham eigenorbitals, which are extended quantities. The reason for the decay properties of the density matrix lies in the interference among the various eigenfunctions, thereby canceling contributions where $\mathbf{r}$ and $\mathbf{r}'$ are far away.

To illustrate the decay properties of the density matrix, the latter one was explicitly constructed according to Eq. (3.2) from the Kohn-Sham orbitals that emerged from a traditional cubically scaling DFT calculation. Since the density matrix is a six-dimensional quantity, it can not be visualized directly. Therefore it has only been calculated along the x dimension; two points y_0 and z_0 for the y and z direction, respectively, were



(a) The density matrix with respect to x and x', as specified by Eq. (3.9). There is obviously an exponential decay with respect to the distance $|x - x'|$; within a distance of a few bohr the values of the density matrix decay by several orders of magnitude.



(b) The isosurface of the square of a Kohn-Sham orbital. The plotted isosurface has a value of $5 \cdot 10^{-8}$ and is consequently comparable to the light blue values in Fig. 3.1a. The locality which is present in the density matrix is completely missing.

Figure 3.1: Illustration of the decay properties of the density matrix and the extended nature of the Kohn-Sham orbitals. The calculation was carried out for a water droplet consisting of 1500 atoms and a diameter of a bit more than 60 bohr, using the traditional cubic version of BigDFT.

chosen and then the density matrix along the x direction was calculated as

$$F(x,x') = \sum_i f(\epsilon_i)\psi_i(x,y_0,z_0)\psi_i(x',y_0,z_0). \tag{3.9}$$

The origin was chosen to lie in the center of the simulation box and consequently $y_0 = z_0 = 0$ was used.
The resulting density matrix is shown in Fig. 3.1a. It is obvious that the values decay exponentially with the distance $|x - x'|$. This locality is not at all represented by the Kohn-Sham orbitals that were used for the construction of the density matrix and which can be fairly extended. The square of such an extended orbital is shown in Fig. 3.1b.

This intrinsic sparsity of the density matrix is the key in developing an algorithm that scales only linearly with the size of the system. Due to the rapid decay of the matrix elements $F(\mathbf{r},\mathbf{r}')$ with respect to the distance $|\mathbf{r} - \mathbf{r}'|$ it is justified to cut the density matrix at a given radius in order to enforce a strict sparsity, i.e. to explicitly set

$$F(\mathbf{r},\mathbf{r}') = 0 \qquad \text{for} \quad |\mathbf{r} - \mathbf{r}'| > \gamma, \tag{3.10}$$

where γ is some system-dependent constant that characterizes the decay behavior. This strict sparsity can then be exploited further to reach a linear scaling algorithm.

3.2 Strategies for linear scaling DFT

All linear scaling methods exploit in some way the decay properties of the density kernel or the Wannier functions, meaning that they assume that these quantities are zero outside of a given localization region and therefore only calculate them within this subvolume. For simplicity often a sphere is taken and the localization region is consequently described by a single parameter, namely the cutoff radius.

There are several different approaches to reach linear scaling [46]:

- The Fermi Operator Expansion (FOE) directly calculates the density matrix F as a function of the Hamiltonian $\mathcal{H}$, i.e. $F = f(\mathcal{H})$. One such representation is based on a series of Chebyshev polynomials [47,48], another one one a rational expansion [49]. The Chebyshev expansion will be described in more detail in Sec. 5.2.3.

- As mentioned, the zero temperature density matrix of an insulator does not have full rank and can be constructed from the occupied states only. The Fermi Operator Projection method [50,51] is similar to the FOE method, but uses the density matrix as a projection operator onto the occupied subspace, in this way generating a set of Wannier-like orbitals. In this way one does not have to deal with unoccupied states as it is the case for the FOE method.

- The idea of the divide-and-conquer method is to divide a large system into several smaller subsystems. After solving the problem separately in each of these subvolumes, the solution for the entire system is then patched together from the solutions of the subsystems. In its first formulation [52,53] this method was applied to the calculation of the charge density, in a later version [54] directly to the construction of the density matrix.

- In the density-matrix minimization approach [55] one determines the density matrix at zero temperature by minimizing a functional which ensures that the two essential properties of the density matrix – namely that it is idempotent and that its eigenvectors with eigenvalue 1 are the occupied eigenvectors of the Hamiltonian – are simultaneously fulfilled. The functional whose minimization leads to the desired properties is given by

$$\Omega = \mathrm{tr}[(3F^2 - 2F^3)(\mathcal{H} - \mu I)], \tag{3.11}$$

  where μ is the chemical potential and I the identity. The term in the first parenthesis is called the "McWeeny purification" [56] which drives the density matrix towards idempotency.

- The orbital minimization approach [57–61] does not directly calculate the density matrix, but expresses it via a set of Wannier functions according to Eq. (3.4). To obtain the latter ones the following functional has to be minimized:

$$\Omega = 2\sum_n \sum_{i,j} c_i^n H'_{ij} c_j^n - \sum_{n,m} \sum_{i,j} c_i^n H'_{ij} c_j^m \sum_l c_l^n c_l^m. \tag{3.12}$$

  Here c_i^n is the expansion coefficient of the nth Wannier function with respect to the ith basis function and H'_{ij} the matrix element of the shifted Hamiltonian $\mathcal{H} - \mu I$ with respect to the basis functions.

- The optimal basis density-matrix minimization approach [62,63] is in some sense a combination of the last two methods. First it generates a set of so-called support functions ϕ_α – which can be seen as some set of auxiliary basis functions, being in turn expressed in terms of an underlying basis set – and then it optimizes the density matrix in the basis of these support functions. In this way the dimensions of the density matrix are considerably reduced compared to a direct

representation in terms of the underlying basis set. The density matrix is written in separable form as

$$F(\mathbf{r},\mathbf{r}') = \sum_{\alpha,\beta} \phi_\alpha(\mathbf{r}) K_{\alpha\beta} \phi_\beta(\mathbf{r}') \tag{3.13}$$

and the matrix $\mathbf{K}$ is given by

$$\mathbf{K} = 3\mathbf{LSL} - 2\mathbf{LSLSL} \tag{3.14}$$

with $\mathbf{S}$ being the overlap matrix among the support functions. Using this representation of the density matrix a minimization of the total energy is carried out with respect to both the support functions ϕ_α and the matrix elements $L_{\alpha\beta}$.

3.3 Linear scaling in BigDFT

3.3.1 General ansatz – support functions and density kernel

The linear scaling version of BigDFT is based on the same ansatz for the density matrix as the optimal basis density-matrix minimization approach – an approach that has also been chosen by other linear scaling codes [64, 65]. Consequently it is written in separable form as

$$F(\mathbf{r},\mathbf{r}') = \sum_{\alpha,\beta} \phi^\alpha(\mathbf{r}) K_{\alpha\beta} \phi^\beta(\mathbf{r}'). \tag{3.15}$$

This separable form has the advantage that it is not necessary to work with a quantity exhibiting in total six dimension (twice a three-dimensional position), but rather with one that depends only on one single three-dimensional position.
The $\phi^\alpha(\mathbf{r})$ are called support functions and the matrix $\mathbf{K}$ the density kernel. A priori the support functions are not specified any further; in particular they are not required to be orthonormal.
In order to reach linear scaling, one has to make sure that the support functions are strictly localized and the density kernel is sparse.

The reason for using superscripts for the support functions and subscripts for the density kernel is not just an aesthetic one. As will be shown later, the support functions are expanded in terms of an underlying orthogonal basis. Therefore they can be identified as the coordinate vector with respect to this basis, thus being a contravariant quantity and consequently being denoted by an upper index. The density kernel, on the other hand, is a covariant quantity of second order and therefore denoted by two lower indices.

By defining covariant support functions $\phi_\alpha(\mathbf{r})$ which satisfy the relation

$$\int \phi_\alpha(\mathbf{r})\phi^\beta(\mathbf{r})\,\mathrm{d}\mathbf{r} = \delta_\alpha^\beta, \tag{3.16}$$

it becomes clear that the density kernel is actually the density matrix in the basis of these covariant support functions:

$$K_{\alpha\beta} = \iint \phi_\alpha(\mathbf{r})F(\mathbf{r},\mathbf{r}')\phi_\beta(\mathbf{r}')\,\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}'. \tag{3.17}$$

The transformation from contravariant to covariant quantities is done using the metric tensor $g_{\alpha\beta}$, which is symmetric, i.e. $g_{\alpha\beta} = g_{\beta\alpha}$. This transformation reads

$$\phi_\alpha = \sum_\beta \phi^\beta g_{\beta\alpha}. \tag{3.18}$$

In order to determine the metric tensor Eq. (3.16) can be exploited. Using the short Bra-ket notation and defining the overlap matrix among the contravariant support functions by $S^{\alpha\beta} = \langle\phi^\alpha|\phi^\beta\rangle$ one gets

$$\delta_\alpha^\gamma = \langle\phi^\gamma|\phi_\alpha\rangle = \sum_\beta \langle\phi^\gamma|\phi^\beta\rangle\, g_{\beta\alpha} = \sum_\beta S^{\gamma\beta}g_{\beta\alpha}, \tag{3.19}$$

from which one concludes that the metric tensor is the inverse of the overlap matrix among the contravariant support functions, i.e. $g_{\beta\alpha} = (S^{-1})^{\beta\alpha}$. Therefore it will from now on be denoted by $S_{\alpha\beta}$ and the relation reads

$$S_{\beta\alpha} = (S^{-1})^{\beta\alpha}. \tag{3.20}$$

This distinction between contravariant and covariant quantities is very important for the decay properties of the density kernel. It is clear from Eq. (3.17) that this decay is determined by the localization characteristics of the covariant support functions ϕ_α. Unfortunately it is not possible to directly control their decay properties. Even if the contravariant support functions ϕ^α – which are the ones that one has control over – are well localized, this is not necessarily the case for the covariant ones. If the covariant support functions decay only slowly, this behavior is inherited by the density kernel, in this way making it difficult to truncate it and finally hindering an efficient linear scaling implementation.

3.3.2 Physical quantities in terms of the support functions and the density kernel

The next step is to determine how physical quantities as the total number of electrons and the band-structure energy are related to the support functions and the density

kernel [66].
The first thing to note is that with the introduction of contravariant and covariant support functions the completeness relation becomes

$$1 = \sum_{\alpha} |\phi^{\alpha}\rangle \langle \phi_{\alpha}| = \sum_{\alpha} |\phi_{\alpha}\rangle \langle \phi^{\alpha}| = \sum_{\alpha,\beta} |\phi^{\alpha}\rangle S_{\alpha\beta} \langle \phi^{\beta}| = \sum_{\alpha,\beta} |\phi^{\alpha}\rangle (S^{-1})^{\alpha\beta} \langle \phi^{\beta}| . \tag{3.21}$$

This relation can be used to establish a relation between the density kernel in the basis of the contravariant support functions, $K^{\alpha\beta}$, and the one in the basis of the covariant ones, $K_{\alpha\beta}$:

$$\begin{aligned} K^{\alpha\beta} &= \langle \phi^{\alpha}|F|\phi^{\beta}\rangle \\ &= \sum_{\gamma,\delta,\epsilon,\zeta} \langle \phi^{\alpha}|\phi^{\gamma}\rangle S_{\gamma\delta} \langle \phi^{\delta}|F|\phi^{\epsilon}\rangle S_{\epsilon\zeta} \langle \phi^{\zeta}|\phi^{\beta}\rangle \\ &= \sum_{\gamma,\zeta} S^{\alpha\gamma} \langle \phi_{\gamma}|F|\phi_{\zeta}\rangle S^{\zeta\beta} \\ &= \sum_{\gamma,\zeta} S^{\alpha\gamma} K_{\gamma\zeta} S^{\zeta\beta}. \end{aligned} \tag{3.22}$$

However, in order to calculate the physical properties of the system, the density matrix has to be represented neither in the basis of the contravariant support functions ϕ^{α} nor in the basis of the covariant ones ϕ_{α}, but in a set of orthonormal support functions $\tilde{\phi}_{\alpha}$. Due to the orthonormality of the latter ones it is not required any more to distinguish between contravariant and covariant quantities, as follows from Eq. (3.18). Defining these orthonormal functions by means of a Löwdin orthonormalization,

$$|\tilde{\phi}_{\alpha}\rangle = \sum_{\beta} (S^{1/2})_{\alpha\beta} |\phi^{\beta}\rangle = \sum_{\beta} (S^{-1/2})^{\alpha\beta} |\phi^{\beta}\rangle , \tag{3.23}$$

one gets for the density kernel in this basis, denoted by $\tilde{K}_{\alpha\beta}$, the following expression:

$$\begin{aligned} \tilde{K}_{\alpha\beta} &= \langle \tilde{\phi}_{\alpha}|F|\tilde{\phi}_{\beta}\rangle \\ &= \sum_{\gamma,\delta,\epsilon,\zeta} \langle \tilde{\phi}_{\alpha}|\phi^{\gamma}\rangle S_{\gamma\delta} \langle \phi^{\delta}|F|\phi^{\epsilon}\rangle S_{\epsilon\zeta} \langle \phi^{\zeta}|\tilde{\phi}_{\beta}\rangle \\ &= \sum_{\gamma,\delta,\epsilon,\zeta,\eta,\theta,\iota,\kappa} (S^{-1/2})^{\alpha\eta} \langle \phi^{\eta}|\phi^{\gamma}\rangle S_{\gamma\delta} S^{\delta\iota} K_{\iota\kappa} S^{\kappa\epsilon} S_{\epsilon\zeta} \langle \phi^{\zeta}|\phi^{\theta}\rangle (S^{-1/2})^{\theta\beta} \\ &= \sum_{\gamma,\delta,\epsilon,\zeta,\eta,\theta,\iota,\kappa} (S^{-1/2})^{\alpha\eta} S^{\eta\gamma} (S^{-1})^{\gamma\delta} S^{\delta\iota} K_{\iota\kappa} S^{\kappa\epsilon} (S^{-1})^{\epsilon\zeta} S^{\zeta\theta} (S^{-1/2})^{\theta\beta} \\ &= \sum_{\iota,\kappa} (S^{1/2})^{\alpha\iota} K_{\iota\kappa} (S^{1/2})^{\kappa\beta}. \end{aligned} \tag{3.24}$$

Diagonalizing this matrix $\tilde{\mathbf{K}}$ will give the occupation numbers of the Kohn-Sham orbitals.

The physical relevant Hamiltonian matrix $\tilde{\mathbf{H}}$ whose diagonalization will yield the Kohn-Sham eigenvalues can be derived along the same lines. Denoting by $H^{\alpha\beta} = \langle\phi^\alpha|\mathcal{H}|\phi^\beta\rangle$ the Hamiltonian in the basis of the contravariant support functions one gets

$$\begin{aligned}
\tilde{H}_{\alpha\beta} &= \langle\tilde{\phi}_\alpha|\mathcal{H}|\tilde{\phi}_\beta\rangle \\
&= \sum_{\gamma,\delta,\epsilon,\zeta} \langle\tilde{\phi}_\alpha|\phi^\gamma\rangle\, S_{\gamma\delta}\, \langle\phi^\delta|\mathcal{H}|\phi^\epsilon\rangle\, S_{\epsilon\zeta}\, \langle\phi^\zeta|\tilde{\phi}_\beta\rangle \\
&= \sum_{\gamma,\delta,\epsilon,\zeta,\eta,\theta} (S^{-1/2})^{\alpha\eta}\, \langle\phi^\eta|\phi^\gamma\rangle\, S_{\gamma\delta} H^{\delta\epsilon} S_{\epsilon\zeta}\, \langle\phi^\zeta|\phi^\theta\rangle\, (S^{-1/2})^{\theta\beta} \\
&= \sum_{\gamma,\delta,\epsilon,\zeta,\eta,\theta} (S^{-1/2})^{\alpha\eta} S^{\eta\gamma} (S^{-1})^{\gamma\delta} H^{\delta\epsilon} (S^{-1})^{\epsilon\zeta} S^{\zeta\theta} (S^{-1/2})^{\theta\beta} \\
&= \sum_{\delta,\epsilon} (S^{-1/2})^{\alpha\delta} H^{\delta\epsilon} (S^{-1/2})^{\epsilon\beta}.
\end{aligned} \tag{3.25}$$

These two relations allow to write the total number of electrons and the band-structure energy, which are – according to Eq. (3.8) – given by $n = \mathrm{tr}(\tilde{\mathbf{K}})$ and $E_{BS} = \mathrm{tr}(\tilde{\mathbf{K}}\tilde{\mathbf{H}})$, respectively, in terms of the matrices $K_{\alpha\beta}$, $H^{\alpha\beta}$ and $S^{\alpha\beta}$:

$$\begin{aligned}
n &= \mathrm{tr}(\tilde{\mathbf{K}}) = \mathrm{tr}(\mathbf{S}^{1/2}\mathbf{K}\mathbf{S}^{1/2}) = \mathrm{tr}(\mathbf{K}\mathbf{S}), \\
E_{BS} &= \mathrm{tr}(\tilde{\mathbf{K}}\tilde{\mathbf{H}}) = \mathrm{tr}(\mathbf{S}^{1/2}\mathbf{K}\mathbf{S}^{1/2}\mathbf{S}^{-1/2}\mathbf{H}\mathbf{S}^{-1/2}) = \mathrm{tr}(\mathbf{K}\mathbf{H}).
\end{aligned} \tag{3.26}$$

Explicitly written out this reads

$$\begin{aligned}
n &= \sum_{\alpha,\beta} K_{\alpha\beta} S^{\alpha\beta} = \sum_{\alpha,\beta} K_{\alpha\beta}\, \langle\phi^\alpha|\phi^\beta\rangle\,, \\
E_{BS} &= \sum_{\alpha,\beta} K_{\alpha\beta} H^{\alpha\beta} = \sum_{\alpha,\beta} K_{\alpha\beta}\, \langle\phi^\alpha|\mathcal{H}|\phi^\beta\rangle\,.
\end{aligned} \tag{3.27}$$

Another important physical quantity, namely the total charge density, can readily be derived from Eq. (3.15) and is given by

$$\rho(\mathbf{r}) = F(\mathbf{r},\mathbf{r}) = \sum_{\alpha,\beta} \phi^\alpha(\mathbf{r}) K_{\alpha\beta} \phi^\beta(\mathbf{r}). \tag{3.28}$$

3.3.3 Idempotency of the density kernel

An important characteristic of the density kernel is its idempotency. This property can be derived from the idempotency of the density matrix which reads

$$\int F(\mathbf{r},\mathbf{r}'') F(\mathbf{r}'',\mathbf{r}')\, \mathrm{d}\mathbf{r}'' = F(\mathbf{r},\mathbf{r}'). \tag{3.29}$$

Writing both parts of the above equation in terms of the density kernel and the support functions gives for the left-hand side

$$\begin{aligned}\int F(\mathbf{r},\mathbf{r}'')F(\mathbf{r}'',\mathbf{r}')\,\mathrm{d}\mathbf{r}'' &= \sum_{\alpha,\beta,\mu,\nu}\int \phi^{\alpha}(\mathbf{r})K_{\alpha\mu}\phi^{\mu}(\mathbf{r}'')\phi^{\nu}(\mathbf{r}'')K_{\nu\beta}\phi^{\beta}(\mathbf{r}')\,\mathrm{d}\mathbf{r}'' \\ &= \sum_{\alpha,\beta,\mu,\nu}\phi^{\alpha}(\mathbf{r})K_{\alpha\mu}S^{\mu\nu}K_{\nu\beta}\phi^{\beta}(\mathbf{r}')\end{aligned} \tag{3.30}$$

and for the right-hand side

$$F(\mathbf{r},\mathbf{r}') = \sum_{\alpha,\beta}\phi^{\alpha}(\mathbf{r})K_{\alpha\beta}\phi^{\beta}(\mathbf{r}'). \tag{3.31}$$

By comparing Eqs. (3.30) and (3.31) one finds the relation

$$K_{\alpha\beta} = \sum_{\mu,\nu}K_{\alpha\mu}S^{\mu\nu}K_{\nu\beta} \tag{3.32}$$

or in more compact form

$$\mathbf{K} = \mathbf{KSK}. \tag{3.33}$$

Consequently any method that tries to determine the density matrix using the ansatz (3.15) has to make sure that Eq. (3.33) holds, be it by construction of the method or by additional constraints in the optimization procedure.

3.3.4 Relation to the traditional Kohn-Sham scheme

The ansatz of writing the density matrix in terms of the support functions and the density kernel can smoothly be transformed back into the traditional Kohn-Sham formulation. If the support functions were identical to the eigenfunctions of the Hamiltonian, i.e. $\phi_\alpha = \phi^\alpha = \psi_\alpha = \psi^\alpha$ – the contravariant and covariant quantities are identical in this case due to the orthonormality of the Kohn-Sham orbitals –, the kernel elements would be the occupation number times a Kronecker delta:

$$\begin{aligned}K^{(KS)}_{\alpha\beta} &= \iint \psi_\alpha(\mathbf{r})F(\mathbf{r},\mathbf{r}')\psi_\beta(\mathbf{r}')\,\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}' \\ &= f(\epsilon_\beta)\int \psi_\alpha(\mathbf{r})\psi_\beta(\mathbf{r})\,\mathrm{d}\mathbf{r} \\ &= f(\epsilon_\beta)\delta_{\alpha\beta}.\end{aligned} \tag{3.34}$$

In this way one is back at the standard Kohn-Sham formulation:

$$\begin{aligned} E_{BS}^{(KS)} &= \sum_{\alpha,\beta} K_{\alpha\beta}^{(KS)} H^{\alpha\beta} = \sum_{\alpha,\beta} f(\epsilon_\beta)\delta_{\alpha\beta}H_{\alpha\beta} = \sum_{\alpha} f(\epsilon_\alpha)H_{\alpha\alpha} = \sum_{\alpha} f(\epsilon_\alpha)\epsilon_\alpha, \\ \rho^{(KS)}(\mathbf{r}) &= \sum_{\alpha,\beta} \psi_\alpha(\mathbf{r})K_{\alpha\beta}^{(KS)}\psi_\beta(\mathbf{r}) = \sum_{\alpha,\beta} \psi_\alpha(\mathbf{r})f(\epsilon_\beta)\delta_{\alpha\beta}\psi_\beta(\mathbf{r}) = \sum_{\alpha} f(\epsilon_\alpha)|\psi_\alpha(\mathbf{r})|^2, \\ n^{(KS)} &= \sum_{\alpha,\beta} K_{\alpha\beta}^{(KS)} S^{\alpha\beta} = \sum_{\alpha,\beta} f(\epsilon_\beta)\delta_{\alpha\beta}\delta_{\alpha\beta} = \sum_{\alpha} f(\epsilon_\alpha). \end{aligned} \tag{3.35}$$

3.3.5 The Kohn-Sham orbitals in terms of the support functions

An alternative way of thinking is to directly express the Kohn-Sham orbitals ψ_i as a linear combination of the support functions:

$$\psi_i(\mathbf{r}) = \sum_{\alpha} c_{i\alpha}\phi^\alpha(\mathbf{r}). \tag{3.36}$$

This formulation is completely equivalent to Eq. (3.15) since

$$F(\mathbf{r},\mathbf{r}') = \sum_i f(\epsilon_i)\psi_i(\mathbf{r})\psi_i(\mathbf{r}') = \sum_i f(\epsilon_i)\sum_{\alpha,\beta} c_{i\alpha}c_{i\beta}\phi^\alpha(\mathbf{r})\phi^\beta(\mathbf{r}') = \sum_{\alpha,\beta}\phi^\alpha(\mathbf{r})K_{\alpha\beta}\phi^\beta(\mathbf{r}'), \tag{3.37}$$

where the density kernel is given by

$$K_{\alpha\beta} = \sum_i f(\epsilon_i)c_{i\alpha}c_{i\beta}. \tag{3.38}$$

The starting point to determine the expansion coefficients c_i^α is the eigenvalue equation (2.44) for the Kohn-Sham orbitals, $\mathcal{H}\psi_i(\mathbf{r}) = \epsilon_i\psi_i(\mathbf{r})$. Inserting Eq. (3.36) yields

$$\sum_\alpha c_{i\alpha}\mathcal{H}\phi^\alpha(\mathbf{r}) = \epsilon_i\sum_\alpha c_{i\alpha}\phi^\alpha(\mathbf{r}). \tag{3.39}$$

Multiplying from left with $\phi^\beta(\mathbf{r})$ and integrating gives

$$\sum_\alpha c_{i\alpha}\int \phi^\beta(\mathbf{r})\mathcal{H}\phi^\alpha(\mathbf{r})\,\mathrm{d}\mathbf{r} = \epsilon_i\sum_\alpha c_{i\alpha}\int \phi^\beta(\mathbf{r})\phi^\alpha(\mathbf{r})\,\mathrm{d}\mathbf{r}. \tag{3.40}$$

Introducing the usual notations $H^{\beta\alpha} = \int \phi^\beta(\mathbf{r})\mathcal{H}\phi^\alpha(\mathbf{r})\,\mathrm{d}\mathbf{r}$ and $S^{\beta\alpha} = \int \phi^\beta(\mathbf{r})\phi^\alpha(\mathbf{r})\,\mathrm{d}\mathbf{r}$ one finally gets

$$\sum_\alpha H^{\beta\alpha}c_{i\alpha} = \epsilon_i\sum_\alpha S^{\beta\alpha}c_{i\alpha}. \tag{3.41}$$

Thus the result is that the expansion coefficients $c_{i\alpha}$ are given by the solution of the generalized eigenvalue equation

$$\mathbf{H}\mathbf{c}_i = \epsilon_i\mathbf{S}\mathbf{c}_i. \tag{3.42}$$

3.3.6 Orthonormal versus non-orthonormal support functions

In order to develop a method that scales only linearly with respect to the size of the system it is mandatory to use a set of support functions being strictly localized. However, as already mentioned, the sparsity of the density kernel is governed by the decay properties of the covariant support functions and not the contravariant ones themselves. So it might happen that, even though the contravariant support functions are well localized, the covariant ones are fairly extended, thus causing the density kernel to be a rather dense matrix and in this way hindering the development of an efficient linear scaling code.

A solution to this problem would be to use a set of orthonormal support functions such that $S^{\alpha\beta} = \delta^{\alpha\beta}$. It is clear from Eq. (3.18) that this implies the equality of contravariant and covariant support functions, i.e. $\phi_\alpha = \phi^\alpha$. This simplifies the equations of the previous sections considerably since the overlap matrix appearing here and there can be discarded.
Moreover, it is not required any more to distinguish between contravariant and covariant quantities when considering, for instance, the density kernel. This is in particular important from the viewpoint of the decay properties and consequently the sparsity of the matrices. By using an orthonormal set of support functions it is guaranteed that the sparsity of the density kernel is not artificially reduced due to the covariant support functions being too extended.

However it is admittedly difficult to construct a set of support functions which is at the same time strictly localized and orthonormal, since these are in general two contradicting properties that are competing with each other. Actually there is only one class of functions known which exhibits at the same time the two characteristics of orthonormality and compact support, namely the Daubechies wavelets, which are the underlying basis set of BigDFT and will be discussed in more detail in Sec. 4.4.

From these considerations it becomes clear that the support functions are actually required to exhibit two properties which are not compatible, meaning that it is necessary to make some compromise for at least one of them. Since a stringent localization of the support functions must be strictly enforced in order to reach linear scaling, there will be a slight non-orthonormality of the support functions that has to be accepted. However, if the localization regions are chosen sufficiently large, the deviations of the overlap matrix from the identity matrix are fairly small. Therefore it is still justified to make the assumption that the decay properties of the contravariant and covariant support functions are identical, and thus also that the sparsity of the density kernel is not artificially enlarged by the contravariant ones.
Still the general notation using the overlap matrix and the distinction between contravariant and covariant quantities is kept for the further discussions.

3.3.7 Fixed versus optimized support functions

As can be seen from Eq. (3.26) the band-structure energy, which is one of the main outputs of an electronic structure calculation, depends on the support functions and the density kernel. Whereas the density kernel is characteristic for each system, this is a priori not necessarily the case for the support functions. Thus they can basically be classified in two categories, namely those which are fixed and those which are optimized in-situ during the calculation.
So far it has not been specified which category is most suited for the current purposes.

There might be the hope that it would be possible to use a fixed set of support functions and only optimize the density kernel, in this way saving the time needed for the in-situ optimization of the first ones.
However this approach is in general not suited from the viewpoint of the accuracy. First of all there is no simple recipe how to generate a good set of support functions beforehand, making their choice – and thus the final result of the calculation – somehow arbitrary. In addition it might be that a given set gives good results for one system, but fails for another one, i.e. the transferability would be completely lost in this way.

Furthermore it is not guaranteed that working with a fixed set of support functions is actually faster. To overcome the mentioned problems it would be necessary to use a rather large number of support functions. This large set will blow up the dimensions of many quantities – e.g. overlap matrix, Hamiltonian matrix, density kernel, etc. –, which will heavily increase the computation time.

By using, on the other hand, a set of support functions which is optimized in-situ for each system, it should be possible to work with a much smaller set, resulting in matrices whose dimensions are drastically reduced. Thus it might well happen that the time spent for the optimization of the support functions is more than compensated. Furthermore there will always be a natural transferability by construction in this way.

To summarize it seems to be more advantageous to optimize the set of support functions in-situ during the calculation, which will then result in a relatively small number of support functions still yielding an excellent accuracy.

CHAPTER 4

Wavelets – an ideal basis set for linear scaling methods

4.1 Importance of the basis set

In reality most quantities that are dealt with are continuous functions. However, when working on a computer, these quantities have to be transfered onto a finite grid and expressed in terms of a set of basis functions. Both the grid as well as the basis functions which are chosen have a big influence on the accuracy and the speed of the calculation.

The simplest choice for the grid would be a uniform grid that covers the entire simulations box. However this might result in a waste of computational resources if there are wide regions of space which are empty, meaning that there is nothing that needs to be expressed in terms of the basis set. A better solution would be to use an adaptive grid which only covers those regions of space which are of interest.
The choice of the basis set is closely related to the choice of the grid. Obviously an adaptive grid can only be used in connection with a basis set that allows such an adaptive resolution.

For the development of an efficient linear scaling code, the choice of the basis set is of utmost importance. If one had to specify some properties that the basis set should exhibit, one would probably list the following: It should have compact support in order to give the possibility to work with strictly localized quantities; it should be orthonormal in order to avoid the tedious work with the overlap matrix that would arise otherwise; and it should have systematic convergence properties such that one never

has to worry about the quality of the basis set.
As will be shown in the next section, there indeed exists a basis set which exhibits all these properties.

4.1.1 Wavelets – the third way

A popular choice for the basis set are plane waves. They have many nice properties that make them a good candidate for electronic structure calculations: They exhibit a systematic convergence, meaning that adding more plane waves will systematically increase the accuracy; many of the important parts can efficiently be done using Fast Fourier Transforms exhibiting a favorable $N \log N$ scaling; they form an orthonormal basis set; and they are strictly localized in Fourier space.
On the other hand there are also a few properties that are quite disadvantageous in particular for linear scaling calculations: There is no localization in real space, i.e. empty regions of space still have to be covered by the basis set; and there is no possibility to increase the resolution around the nuclei where usually higher accuracy is required than farther away.

Another popular choice for the basis set are Gaussians. They are in some sense the opposite of plane waves, meaning that they perform poorly in those fields where the plane waves are advantageous and vice versa. In more detail, they exhibit no systematic convergence properties since adding more and more Gaussians might lead to basis superposition errors; furthermore it is not as obvious as for plane waves how to generate a good basis set. On the other hand they have a natural localization in real space and allow for an adaptive resolution around the nuclei. In addition the number of basis functions that is required to get a certain level of accuracy is usually much smaller compared to the number of plane waves needed to obtain the same accuracy.

Another interesting possibility for the choice of the basis set is to use wavelets. They can in some sense combine the advantages of both plane waves and Gaussians. They are well localized in both Fourier space and real space; they form an orthonormal basis set; they allow for an adaptive resolution in certain regions of space; and they exhibit systematic convergence properties.

Comparing these properties with the wish list at the end of the previous section it becomes clear that wavelets are an ideal candidate for a basis set to be used in the context of linear scaling calculations.

4.2 Basic properties of wavelets

There are many different families of wavelets, each one exhibiting some special characteristics. BigDFT uses the so-called *least asymmetric Daubechies of order 16* [67]. However a much simpler wavelet family will be used in order to demonstrate some basic properties of wavelets.

4.2.1 An illustrating example – the Haar wavelet family

The conceptually simplest wavelet family is the so-called Haar wavelet family [68], which is shown in Fig. 4.1. Of course this wavelet family is way too crude to be useful in any numerical context, but it is well suited to illustrate some basic properties of wavelets [69].

Each wavelet family consists of a mother scaling function ϕ and a mother wavelet ψ. As can be seen from the figure, the wavelet is varying more rapidly than the scaling function.
In order to generate a basis set out of these mother functions, one can use scaling and shifting operations:

$$\begin{aligned} \phi_i^k(x) &\propto \phi(2^k x - i), \\ \psi_i^k(x) &\propto \psi(2^k x - i). \end{aligned} \tag{4.1}$$

According to this notation the index k describes the resolution – i.e. higher values of k represent skinnier functions –, whereas the index i stands for the localization in space. These scaled and shifted scaling functions and wavelets can now be used to approximately represent a continuous function, as will be shown in the following.

For simplicity first the case where the basis set consists only of scaling functions is considered. As an easy example a piecewise function f in the interval $[0,1]$ which can exactly be expanded in terms of 16 Haar scaling functions at resolution level 4 will be used for illustrating purposes. Thus the function f may be written as

$$f(x) = \sum_{i=0}^{15} s_i^4 \phi_i^4(x) \qquad \text{with} \quad s_i^4 = f(i/16). \tag{4.2}$$

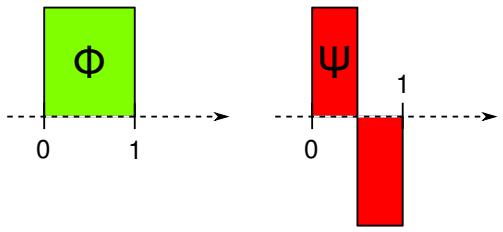


Figure 4.1: Plot of the Haar wavelet family, which is conceptually the simplest wavelet family. On the left side the scaling function ϕ is shown, on the right side the corresponding wavelet ψ. As can be seen the wavelet is varying more rapidly than the scaling function.

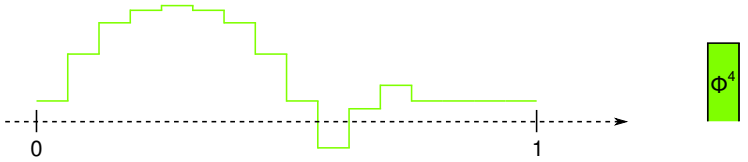


Figure 4.2: Representation of a piecewise constant function using 16 Haar scaling functions at resolution level 4.

An illustration of this expansion is shown in Fig. 4.2. For a function which is not piecewise constant the expansion in terms of the scaling functions is analogous, just that in general the equality between the original function and the scaling function representation is not absolutely exact – this would only hold true in the limit of infinitely skinny scaling functions.

Another, more interesting possibility is to expand the function f in terms of both scaling functions and wavelets. To this end it is necessary to determine a relation between scaling functions and wavelets at different resolution levels. As is depicted in Fig. 4.3 a scaling function at resolution level k can be written as a linear combination of a scaling function and a wavelet at resolution level $k-1$. So any linear combination of the two scaling functions $\phi_{2i}^k(x)$ and $\phi_{2i+1}^k(x)$ can be written as a linear combination of the scaling function $\phi_i^{k-1}(x)$ and the wavelet $\psi_i^{k-1}(x)$.
Consequently the function f from Eq. (4.2) may as well be written as

$$f(x) = \sum_{i=0}^{7} s_i^3 \phi_i^3(x) + \sum_{i=0}^{7} d_i^3 \psi_i^3(x). \tag{4.3}$$

The prescription how to get the expansion coefficients at level $k-1$ from the coefficients at level k can be determined in this simple case by looking at Fig. 4.3 and is given by

$$\begin{aligned} s_i^{k-1} &= \frac{1}{2} s_{2i}^k + \frac{1}{2} s_{2i+1}^k, \\ d_i^{k-1} &= \frac{1}{2} s_{2i}^k - \frac{1}{2} s_{2i+1}^k. \end{aligned} \tag{4.4}$$

This procedure is called "forward transform" or "wavelet analysis".

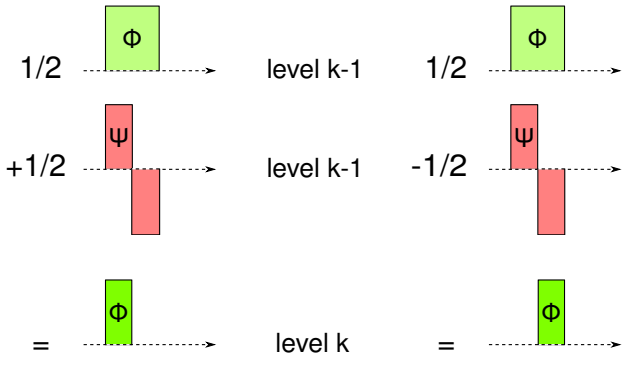
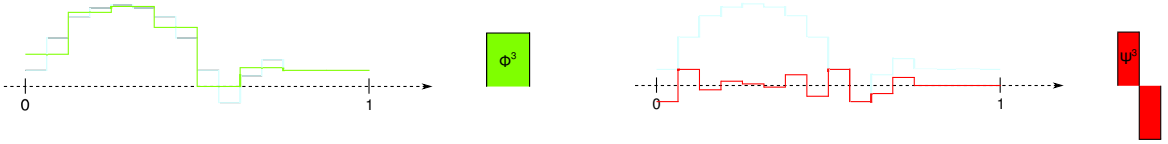


Figure 4.3: Each skinny scaling function at resolution level k can be written as a linear combination of a coarse scaling function and a coarse wavelet at resolution level $k-1$.



(a) The scaling function part of the function, which is much smoother than the original one, meaning that it is varying more slowly.

(b) The wavelet part of the function, which contains obviously the rapidly oscillating corrections to the scaling function representation.

Figure 4.4: Representation of the same function as in Fig. 4.2, however this time split up in scaling functions and wavelets at resolution level 3 according to Eq. (4.3). The original function is indicated in decent blue.

Eq. (4.4) demonstrates that the scaling function coefficients at the lower resolution level are given by a weighted sum of the scaling function coefficients at the higher resolution level, whereas the wavelet coefficients are given by a weighted difference. Therefore it is intuitively clear that the scaling function part in Eq. (4.3) represents a smoothened version of the function, whereas the wavelet part represents the rapidly varying corrections to this smoothened function.
This fact is illustrated in Fig. 4.4. The part which is given by the scaling functions only, i.e. the sum $\sum_{i=0}^{7} s_i^3 \phi_i^3(x)$, is shown in Fig. 4.4a and is clearly much smoother than the original function – at least to the extent to which a step function can be called smooth. On the other hand, the wavelet part, i.e. the sum $\sum_{i=0}^{7} d_i^3 \psi_i^3(x)$, is varying much faster, as can be seen from Fig. 4.4b.

Given a data set whose size is a power of 2, this procedure may now be applied recursively until one finally arrives at

$$f(x) = s_0^0 \phi_0^0(x) + d_0^0 \psi_0^0(x) + \sum_{i=0}^{1} d_i^1 \psi_i^1(x) + \sum_{i=0}^{3} d_i^2 \psi_i^2(x) + \sum_{i=0}^{7} d_i^3 \psi_i^3(x). \qquad (4.5)$$

This representation requires exactly the same number of expansion coefficients as the original one, namely 1 for the scaling function and 15 for the wavelets.
Still this representation is much more interesting than the one using only scaling functions. As depicted in Fig. 4.5 there is a region where the function f is constant. From the relations $s_i^4 = f(i/16)$ and $d_i^{k-1} = \frac{1}{2} s_{2i}^k - \frac{1}{2} s_{2i+1}^k$ (Eqs. (4.2) and (4.4)) it is clear that

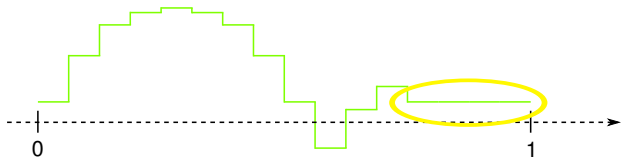


Figure 4.5: Since the function f is constant in the region marked in yellow, some wavelet coefficients will turn out to be zero in this region. Consequently the function can be compressed by using a mixed scaling function / wavelet expansion.

some of the wavelet coefficients will turn out to be zero – this fact is also visible in Fig. 4.4b, where the wavelet part of the function is zero for the region where f is constant – and the function f as given by Eq. (4.5) can be expressed by fewer than the 16 coefficients which are used there.
Thus the conclusion is that a mixed scaling function / wavelet expansion is very well suited to compress data that is only slowly varying.

Since the two representations (4.2) and (4.5) are completely equivalent, it is also possible to go back from the mixed scaling function / wavelet representation to an expansion using only scaling functions. Here the fact that a wavelet family fulfills the so-called refinement relations can be used, meaning that each scaling function and wavelet at resolution level $k-1$ can be written as a linear combination of scaling functions at resolution level k. This fact is again depicted for the Haar wavelet family in Fig. 4.6.
In this simple example the prescription how to get the expansion coefficients for the scaling functions at resolution level k from the scaling function and wavelet coefficients at resolution level $k-1$ can again be determined by looking at the figure and is given by

$$\begin{aligned} s_{2i}^{k} &= s_{i}^{k-1} + d_{i}^{k-1}, \\ s_{2i+1}^{k} &= s_{i}^{k-1} - d_{i}^{k-1}. \end{aligned} \tag{4.6}$$

This procedure is call a "backward transform" or "wavelet synthesis".

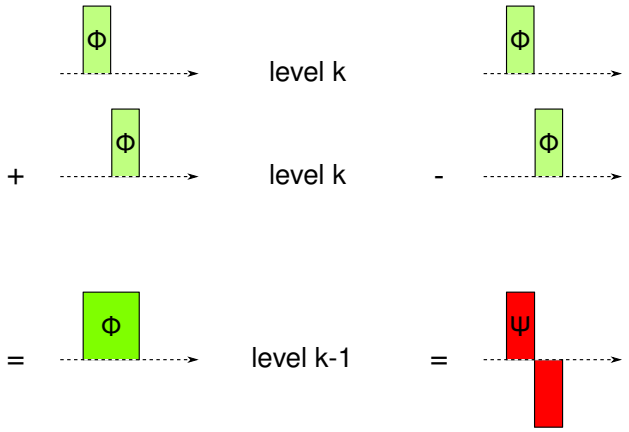


Figure 4.6: Illustration of the refinement relations for the Haar wavelet family: Each scaling function and wavelet at resolution level $k-1$ can be written as linear combination of scaling functions at resolution level k.

4.2.2 Basic formulas for wavelets

In this section some basic formulas that are valid for an orthogonal wavelet family are noted. A more detailed list can be found in an overview by Goedecker [69] and the

non-trivial proofs in the book by Daubechies [67].

An orthogonal wavelet family can be completely characterized by two filters h and g of finite length. Even though the functional form of the scaling functions and wavelets is missing, knowing these filters allows to completely specify the wavelet family.

4.2.2.1 Orthogonality and symmetry of the filters

The filters h and g fulfill the following orthogonality relations

$$\sum_l h_{l-2i} h_{l-2j} = \delta_{ij}, \tag{4.7a}$$

$$\sum_l g_{l-2i} g_{l-2j} = \delta_{ij}, \tag{4.7b}$$

$$\sum_l h_{l-2i} g_{l-2j} = 0, \tag{4.7c}$$

and the symmetry relation

$$g_{i+1} = (-1)^{i+1} h_{-i}. \tag{4.8}$$

4.2.2.2 Refinement relations

The refinement relations, which were descriptively shown in Fig. (4.6), are given by

$$\phi(x) = \sqrt{2} \sum_{j=-m}^{m} h_j \phi(2x - j), \tag{4.9a}$$

$$\psi(x) = \sqrt{2} \sum_{j=-m}^{m} g_j \phi(2x - j), \tag{4.9b}$$

or alternatively written by

$$\phi_i^k(x) = \sqrt{2} \sum_{j=-m}^{m} h_j \phi_{2i+j}^{k+1}(x), \tag{4.10a}$$

$$\psi_i^k(x) = \sqrt{2} \sum_{j=-m}^{m} g_j \phi_{2i+j}^{k+1}(x), \tag{4.10b}$$

where the notations $\phi_i^k(x) = \sqrt{2}^k \phi(2^k x - i)$ and $\psi_i^k(x) = \sqrt{2}^k \psi(2^k x - i)$ were used.

4.2.2.3 Forward and backward transform

The prescription how to calculate the new coefficients in the course of a forward transform – also called wavelet analysis – is given by

$$s_i^{k-1} = \sum_{j=-m}^{m} h_j s_{j+2i}^k, \tag{4.11a}$$

$$d_i^{k-1} = \sum_{j=-m}^{m} g_j s_{j+2i}^k, \tag{4.11b}$$

and the one for the backward transform – also called wavelet synthesis – is given

$$s_{2i}^{k+1} = \sum_{j=-m/2}^{m/2} h_{2j} s_{i-j}^k + g_{2j} d_{i-j}^k, \tag{4.12a}$$

$$s_{2i+1}^{k+1} = \sum_{j=-m/2}^{m/2} h_{2j+1} s_{i-j}^k + g_{2j+1} d_{i-j}^k. \tag{4.12b}$$

Eqs. (4.11) and (4.12) are the generalizations of Eqs. (4.4) and (4.6), respectively.

4.2.2.4 Orthogonality of the scaling functions and wavelets

Just as the filters, the scaling functions and wavelets satisfy as well orthogonality relations:

$$\int \phi_i^k(x)\phi_j^k(x)\,\mathrm{d}x = \delta_{ij}, \tag{4.13a}$$

$$\int \psi_i^k(x)\phi_j^q(x)\,\mathrm{d}x = 0, \quad k \geq q, \tag{4.13b}$$

$$\int \psi_i^k(x)\psi_j^q(x)\,\mathrm{d}x = \delta_{kq}\delta_{ij}. \tag{4.13c}$$

4.2.3 Wavelets in three dimensions

So far only scaling functions and wavelets in one dimension have been considered. However real applications typically require a three-dimensional basis set. Thus the one-dimensional scaling functions and wavelets have to be generalized to three dimensions.

The easiest way to construct such a three-dimensional basis set consists in forming products of one dimensional scaling function and wavelets. This gives rise to one

three-dimensional scaling function, which is a product of three one-dimensional scaling functions, and seven three-dimensional wavelets, which are products containing at least one one-dimensional wavelet:

$$\begin{aligned}
\phi_{i,j,k}(x,y,z) &= \phi(x-i)\phi(y-j)\phi(z-k),\\
\psi^{(1)}_{i,j,k}(x,y,z) &= \psi(x-i)\phi(y-j)\phi(z-k),\\
\psi^{(2)}_{i,j,k}(x,y,z) &= \phi(x-i)\psi(y-j)\phi(z-k),\\
\psi^{(3)}_{i,j,k}(x,y,z) &= \psi(x-i)\psi(y-j)\phi(z-k),\\
\psi^{(4)}_{i,j,k}(x,y,z) &= \phi(x-i)\phi(y-j)\psi(z-k),\\
\psi^{(5)}_{i,j,k}(x,y,z) &= \psi(x-i)\phi(y-j)\psi(z-k),\\
\psi^{(6)}_{i,j,k}(x,y,z) &= \phi(x-i)\psi(y-j)\psi(z-k),\\
\psi^{(7)}_{i,j,k}(x,y,z) &= \psi(x-i)\psi(y-j)\psi(z-k).
\end{aligned} \tag{4.14}$$

The three-dimensional scaling functions and wavelets fulfill as well orthogonality and refinement relations which are generalizations of the ones for the one-dimensional case.
Forward and backward transforms are done by first transforming along the x dimension, then along the y dimension and finally along the z axis, or any other order.

4.3 Calculating derivatives in a wavelet basis

Since the application of the kinetic energy operator $-\frac{1}{2}\nabla^2$ requires the calculation of derivatives, it is important that the basis set allows to perform this operation efficiently. Fortunately this is the case for wavelets.

It turns out that applying the derivative operator of any order l, $\frac{\partial^l}{\partial x^l}$, to a scaling function at position j_1 and projecting this quantity back onto a scaling function at position i_1 gives rise to a special filter of finite length that is denoted by $a_{i_1-j_1}$ and only depends on the difference $i_1 - j_1$. The filters for the other cases – applying the derivative to a

wavelet and projecting back onto a scaling function etc. – are defined analogously:

$$a_{i_1-j_1} = \int \phi(x-i_1)\frac{\partial^l}{\partial x^l}\phi(x-j_1)\,\mathrm{d}x, \tag{4.15a}$$

$$b_{i_1-j_1} = \int \psi(x-i_1)\frac{\partial^l}{\partial x^l}\phi(x-j_1)\,\mathrm{d}x, \tag{4.15b}$$

$$c_{i_1-j_1} = \int \phi(x-i_1)\frac{\partial^l}{\partial x^l}\psi(x-j_1)\,\mathrm{d}x, \tag{4.15c}$$

$$e_{i_1-j_1} = \int \psi(x-i_1)\frac{\partial^l}{\partial x^l}\psi(x-j_1)\,\mathrm{d}x. \tag{4.15d}$$

The calculation of these filters is shown in more detail in appendix A.1.

Once these filters are determined, the calculation of derivatives in the wavelet basis is not difficult any more. Given a quantity $\Psi(x,y,z)$ that is expanded in a three-dimensional wavelet basis,

$$\begin{aligned}\Psi(x,y,z) = &\sum_{j_1,j_2,j_3} sss_{j_1,j_2,j_3}\phi(x-j_1)\phi(y-j_2)\phi(z-j_3)\\ &+\sum_{j_1,j_2,j_3} dss_{j_1,j_2,j_3}\psi(x-j_1)\phi(y-j_2)\phi(z-j_3)\\ &+\sum_{j_1,j_2,j_3} sds_{j_1,j_2,j_3}\phi(x-j_1)\psi(y-j_2)\phi(z-j_3)\\ &+\sum_{j_1,j_2,j_3} dds_{j_1,j_2,j_3}\psi(x-j_1)\psi(y-j_2)\phi(z-j_3)\\ &+\sum_{j_1,j_2,j_3} ssd_{j_1,j_2,j_3}\phi(x-j_1)\phi(y-j_2)\psi(z-j_3)\\ &+\sum_{j_1,j_2,j_3} dsd_{j_1,j_2,j_3}\psi(x-j_1)\phi(y-j_2)\psi(z-j_3)\\ &+\sum_{j_1,j_2,j_3} sdd_{j_1,j_2,j_3}\phi(x-j_1)\psi(y-j_2)\psi(z-j_3)\\ &+\sum_{j_1,j_2,j_3} ddd_{j_1,j_2,j_3}\psi(x-j_1)\psi(y-j_2)\psi(z-j_3),\end{aligned} \tag{4.16}$$

where *sss*, *dss* etc. are the three-dimensional generalizations of the *s* and *d* coefficients introduced previously, applying the derivative operator is straightforward, as is shown in more detail in appendix B.1. Denoting by sss'_{i_1,i_2,i_3} the expansion coefficient for the scaling function $\phi(x-i_1)\phi(y-i_2)\phi(z-i_3)$ after the application of the derivative operator, then its value is given by

$$\begin{aligned}sss'_{i_1,i_2,i_3} &= \iiint \phi(x-i_1)\phi(y-i_2)\phi(z-i_3)\frac{\partial^l}{\partial x^l}\Psi(x,y,z)\,\mathrm{d}x\mathrm{d}y\mathrm{d}z\\ &= \sum_{j_1} a_{j_1-i_1} sss_{j_1,i_2,i_3} + \sum_{j_1} b_{j_1-i_1} dss_{j_1,i_2,i_3}.\end{aligned} \tag{4.17}$$

The coefficients for the three-dimensional wavelets and the other directions can be calculated along the same lines and are given in appendix B.1.

To conclude this means that the calculation of the derivatives requires simply a convolution with a filter of finite length.

4.4 Wavelets in BigDFT

As already mentioned the wavelet family used in BigDFT is the *least asymmetric Daubechies of order 16* family, which is an orthogonal family with compact support. The filters h and g which characterize the family have only non-zero entries in the interval from -7 to 8, which makes in total 16 elements. Since the extent of the scaling functions and wavelets is as well determined by the length of this filter, a scaling function or wavelet centered on a given grid point i does not extend farther than $i-7$ and $i+8$. A plot of the scaling function and the associated wavelet is given in Fig. 4.7.

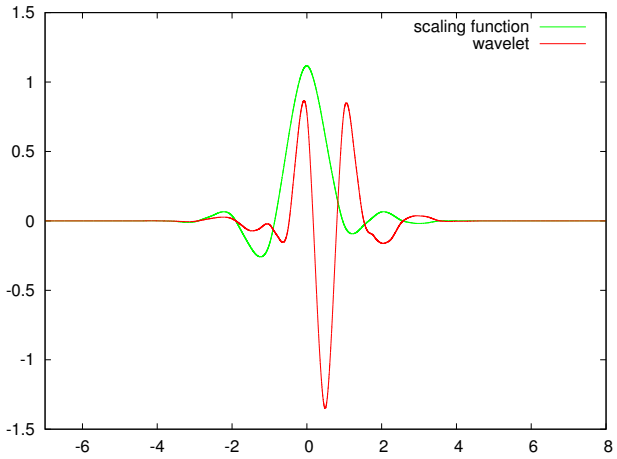


Figure 4.7: Plot of the scaling function and wavelet of the *least asymmetric Daubechies of order 16* wavelet family. Due to the filter which has non-zero entries only in the range from -7 to 8, the Daubechies are only different from zero in the same interval. The wavelet is varying more rapidly than the scaling function, which is in agreement with the discussion of the Haar wavelet at the beginning of this chapter.

4.4.1 The various resolution levels

A priori a wavelet basis allows to use as many resolution levels as desired. However in BigDFT there are only three levels of accuracy:

- A grid point carries one scaling function and seven wavelets. This is the case for all grid points that are close to the nuclei and therefore require a high resolution. Points which belong to this category are said to lie in the "fine region".
- A grid point carries only one scaling function. This is the case for all grid points that are farther away from the nuclei. Points which belong to this category are said to lie in the "coarse region". The resolution in the coarse region is half that of the fine region.
- A grid point carries neither scaling function nor wavelet. This is the case for all grid points that are even farther away from the nuclei than those of the coarse region. Since these points do not contribute to the representation of a quantity in the wavelet basis, they can be completely discarded.

It is worth noting that even though the resolution in the fine region is doubled compared to the coarse region, the grid spacing is the same in the entire simulation box. The resolution enhancement stems only from the additional wavelets in the fine region.

The prescription how to generate the coarse and the fine region is rather simple. The coarse region is defined as the union of spheres with a given radius which are centered on each nuclei, and the fine region is analogously defined by a union of spheres with a smaller radius. The radii of these spheres are given by the product of an atom-dependent constant and a user-specified factor. The grid which is constructed in this

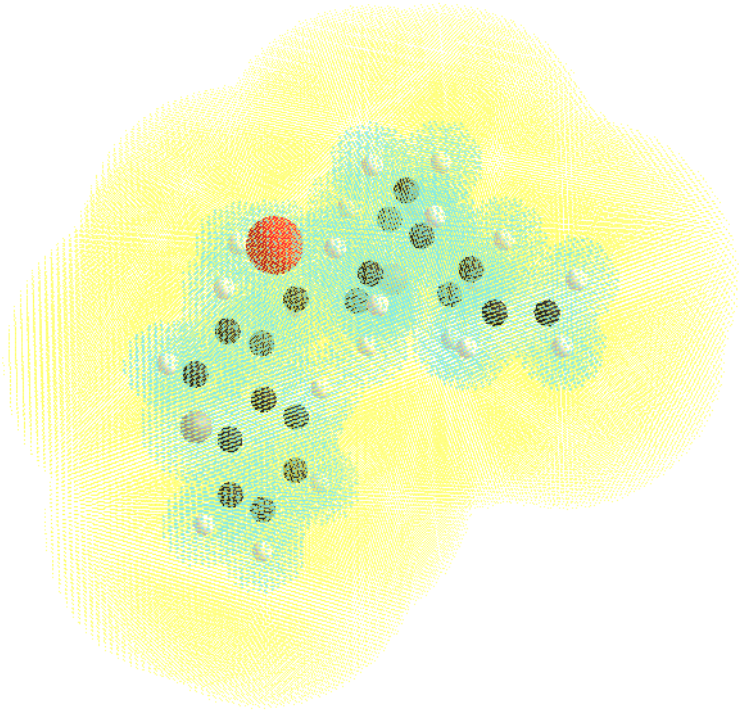


Figure 4.8: Visualization of the coarse and fine regions for cinchonidine which has the chemical formula $C_{19}H_{22}N_2O$. The yellow points represent the coarse grid, the light blue points the fine grid. Points which are neither in the coarse nor in the fine region are not shown.
The points belonging to the coarse region carry one scaling function, whereas the points belonging to the fine region carry in addition seven wavelets. It is obvious how the resolution is adaptively increased around the nuclei in this way.

way will be referred to as global grid or global region. A visualization of both regions for the case of a small molecule (cinchonidine, $C_{19}H_{22}N_2O$) is given in Fig. 4.8.

4.4.2 The wavelet basis in the traditional cubic version

In the traditional cubic version of BigDFT, the Kohn-Sham orbitals – denoted here by Ψ to avoid any confusion – are directly expanded in the basis of the Daubechies scaling functions and wavelets:

$$\Psi_i(\mathbf{r}) = \sum_{j_1,j_2,j_3} s^i_{j_1,j_2,j_3}\phi_{j_1,j_2,j_3}(\mathbf{r}) + \sum_{j_1,j_2,j_3}\sum_{\nu=1}^{7} d^i_{j_1,j_2,j_3,\nu}\psi^{(\nu)}_{j_1,j_2,j_3}(\mathbf{r}). \tag{4.18}$$

Here $s^i_{j_1,j_2,j_3}$ and $d^i_{j_1,j_2,j_3,\nu}$ correspond to the sss_{j_1,j_2,j_3} etc. coefficients in (4.16), and $\phi_{j_1,j_2,j_3}(\mathbf{r})$ and $\psi^{(\nu)}_{j_1,j_2,j_3}(\mathbf{r})$ are shorthand notations for $\phi(x-j_1)\phi(y-j_2)\phi(z-j_3)$ etc.
According to the definition in the previous section, a given grid point (j_1, j_2, j_3) will have both s and d coefficients if it belongs to the fine region, only a s coefficient if it belongs to the coarse region, and no coefficients if it belongs to the empty region. The Kohn-Sham orbitals can then be represented by a compressed form where only the non-zero coefficients are stored.
Thanks to the orthonormality of the Daubechies wavelet family, many operations can be directly done using only these compressed vectors holding the coefficients. For instance, a scalar product of two orbitals is simply given by the dot product of the two coefficient vectors. Another example is the application of the projectors which were described in Sec. 2.3.4 since they are as well expressed in this basis; thus also the application of the pseudopotential part is rather straightforward.

However there are also some quantities which are not represented in this compressed form. For instance the potential is evaluated on a uniform grid consisting of interpolating scaling functions with a grid spacing which is half that of the combined scaling function / wavelet representation. To go from one representation to the other one the forward and backward transforms described in 4.2.2.3 can be used.
Furthermore it turned out that it is not advantageous to evaluate the potential directly in the basis of these interpolating scaling functions. The reason is that one single scaling function is not very smooth. Instead it is better to evaluate the potential in the basis of some modified scaling functions – denoted by $\tilde{\phi}(x-i_1)$ – which exhibit a higher smoothness. In this way one single matrix element

$$V_{i_1i_2i_3,j_1j_2j_3} = \iiint \tilde{\phi}(x-i_1)\tilde{\phi}(y-i_2)\tilde{\phi}(z-i_3)\mathcal{V}(x,y,z)\tilde{\phi}(x-j_1)\tilde{\phi}(y-j_2)\tilde{\phi}(z-j_3)\,\mathrm{d}x\mathrm{d}y\mathrm{d}z \tag{4.19}$$

is not very accurate either, but the total expectation value

$$E_{pot} = \iiint \Psi(x,y,z)\mathcal{V}(x,y,z)\Psi(x,y,z)\,\mathrm{d}x\mathrm{d}y\mathrm{d}z \tag{4.20}$$

can be evaluated with very high accuracy [70].
The smoothening can be done on the fly as one transforms from the mixed scaling function / wavelet representation to the scaling function only representation [71].

4.4.3 The wavelet basis in the new linear version

The implementation of the linear scaling version of BigDFT is based on Eq. (3.15) or Eq. (3.36), respectively. Both the support functions ϕ^α and the density kernel $\mathbf{K}$ (or the coefficients $\mathbf{c}_i$, respectively) are optimized in order to get the best accuracy possible.

The support functions are in turn again expanded in the underlying basis set of scaling functions and wavelets, which are denoted here by φ_{j_1,j_2,j_3} and ψ_{j_1,j_2,j_3} in order to avoid confusions with the support functions ϕ^α. Consequently each support function can be written as

$$\phi^\alpha(\mathbf{r}) = \sum_{j_1,j_2,j_3} s^\alpha_{j_1,j_2,j_3}\varphi_{j_1,j_2,j_3}(\mathbf{r}) + \sum_{j_1,j_2,j_3}\sum_{\nu=1}^{7} d^\alpha_{j_1,j_2,j_3;\nu}\psi^{(\nu)}_{j_1,j_2,j_3}(\mathbf{r}). \tag{4.21}$$

Thanks to this expansion, the same considerations as for the Kohn-Sham orbitals apply, i.e. the support functions can as well be stored in a compressed format and many operations can be carried out straightforwardly thanks to the orthonormality of the scaling functions and wavelets.

In order to keep the support functions strictly localized, the coefficients $s^\alpha_{j_1,j_2,j_3}$ and $d^\alpha_{j_1,j_2,j_3;\nu}$ that represent a scaling function or wavelet, respectively, on the grid point (j_1,j_2,j_3) are set to zero if this grid point lies outside of the localization region of the support function. For simplicity, the localization regions are atom-centered spheres with a cutoff radius r_{cut}, but there is no fundamental constraint that would prevent them to have a different shape in the future, e.g. to be centered in between two atoms. Thus the condition for the localization region is given by

$$\left.\begin{aligned} s^\alpha_{j_1,j_2,j_3} &= 0 \\ d^\alpha_{j_1,j_2,j_3;\nu} &= 0 \end{aligned}\right\} \quad \text{if} \quad |\mathbf{R}_{j_1,j_2,j_3} - \mathbf{R}_\alpha| > r_{\mathrm{cut}}, \tag{4.22}$$

where $\mathbf{R}_{j_1,j_2,j_3}$ is the position of the grid point (j_1,j_2,j_3) and $\mathbf{R}_\alpha$ is the center of the localization region for the support functions ϕ^α.

The localization regions are always a subset of the global grid, i.e. a given grid point

of the localization region is only occupied if it is occupied as well for the global region. This is in particular important for the distinction between coarse and fine region, meaning that a grid point of the localization region can only belong to the fine region if it belongs as well to the fine region for the global grid.
As an illustration, Fig. 4.9 shows the same system as in Fig. 4.8, but this time including one such localization region.

The choice of the cutoff radius is one of the most important parameters from the viewpoint of both the accuracy and the speed and can be specified manually. It will depend both on the type of the atom on which the localization region is centered and on the accuracy that should be obtained. By choosing a too small radius it is not possible to get a set of support functions that can yield meaningful results since the density matrix – or the Kohn-Sham orbitals, respectively – cannot reasonably be represented anymore. Choosing the cutoff radius too large, on the other hand, will increase the computational time without giving significantly better results.

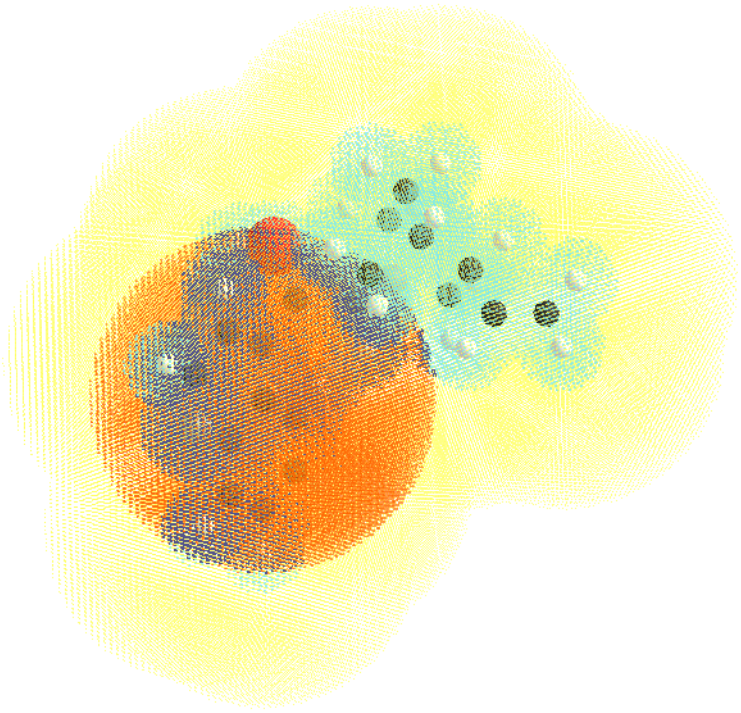


Figure 4.9: Visualization of the various grids used for the linear scaling version, for the same system as shown in Fig. 4.8. The yellow and the light blue points are the coarse and fine grid, respectively, of the global region. The orange and dark blue points are the coarse and fine grid, respectively, for the localization region. It becomes clear that the localization region is built on top of the global grid, i.e. a grid point of the localization region is only occupied if it is occupied as well for the global grid. In particular this applies to the distinction between coarse and fine region.

4.4.3.1 Enlarging the localization regions for the application of the Hamiltonian

The introduction of finite localization regions for the support functions involves some subtleties with respect to the application of the Hamiltonian onto the latter ones. The problematic part are the convolutions required for the evaluation of the kinetic energy.

Since these convolutions involve eight neighboring grid points, the value of $\langle \phi^\alpha | \nabla^2 \phi^\beta \rangle$ is not the same as $\langle \phi^\beta | \nabla^2 \phi^\alpha \rangle$. As a consequence the Hamiltonian matrix will not be symmetric.

This problem can be overcome by the introduction of a buffer zone of eight grid points around each support function. This buffer is initialized to zero, but the application of the Hamiltonian will fill it with non-zero values, as illustrated in Fig. 4.10. When calculating the scalar products in order to build the Hamiltonian matrix, it is important to keep the buffer zones; in this way the symmetry of the matrix is restored.

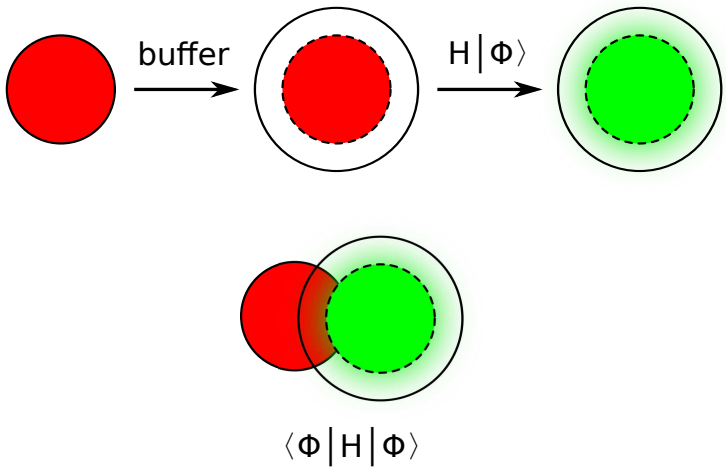


Figure 4.10: Illustration of the correct way to calculate the matrix elements $\langle \phi^\alpha | \mathcal{H} | \phi^\beta \rangle$. In a first step a buffer of eight grid points is added around each support function and initialized to zero. The convolutions performed in the course of the Hamiltonian application will then fill it with non-zero values. When building the scalar product with another support function, this buffer has to be retained.

CHAPTER 5

Detailed implementation of a linear scaling algorithm in BigDFT

5.1 Optimization of the support functions

According to the previous discussions, the number of support functions should be kept as small as possible. Therefore it is important that they are of very high quality.

The best possible choice would be to find some set of well localized Wannier functions, since they are equivalent to the extended Kohn-Sham orbitals that would emerge from a traditional cubic calculation. In this way the linear scaling version should give exactly the same results as the cubic one.
Of course it is not possible to simply generate the Wannier functions on-the-fly without any additional constraints, but there is still the hope that a set of support functions can be generated which has some resemblance with them.

There are four different modes how the support functions can be optimized: The trace minimization mode, the energy minimization mode, the mixed mode and the hybrid mode.
In principle they only differ in the target function that has to be minimized; apart from that the procedure is the same for all modes. First one calculates the unconstrained gradient of the target function Ω that has to be minimized with respect to the support functions:

$$g_\alpha(\mathbf{r}) = \frac{\delta\Omega}{\delta\phi^\alpha(\mathbf{r})}. \tag{5.1}$$

It has to be stressed that the derivative of the scalar Ω with respect to the contravariant support function $\phi^\alpha(\mathbf{r})$ yields a covariant gradient $g_\alpha(\mathbf{r})$. Therefore, before updating the support functions with this gradient, it first has to be converted to contravariant form, as follows from the discussion in Sec. 3.3.1:

$$g^\alpha(\mathbf{r}) = \sum_\beta S^{\alpha\beta} g_\beta(\mathbf{r}), \tag{5.2}$$

where **S** is the overlap matrix among the support functions.
Next one has to apply the orthonormality constraint to this gradient, as will be described in more detail in Sec. 5.1.8. Then the support functions are updated with this constrained gradient using any optimization method; in practice steepest descent or DIIS [72] is employed. In the last step one has to orthogonalize the support functions, as will be described in Sec. 5.1.7. After this the cycle starts over again.

It has to be noted that the support functions are always optimized at a fixed potential, i.e. in a non-self-consistent way. The potential is only updated in a second step, after the support functions have been optimized to some extent. The update of the potential is related to an optimization of the density kernel and will be described in more detail in Sec. 5.2.

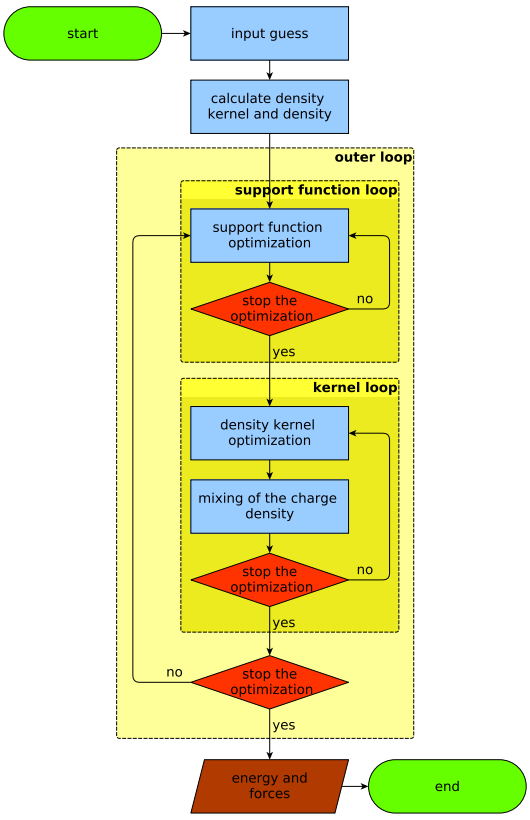


Figure 5.1: Flowchart to illustrate the basic linear scaling approach. There is one outer loop and two inner loops.
In the first inner loop the support functions are optimized until the exit criterion or the maximal number of iterations is reached. In the second inner loop the density kernel is optimized, followed by a mixing of the density, again until the exit criterion or the maximal number of iterations is reached.
These two loops are then iterated in the outer loop until overall convergence is reached. This overall convergence is based on the mean difference of the charge density per grid point which must come below a given threshold.
The input guess which precedes the outer loop will be described in more detail in Sec. 5.1.5.

As a consequence it is not advisable to do many steps in the optimization of the support functions, but rather only a few ones and then to update the potential. Therefore this optimization typically stops since the maximal number of iterations has been reached and not since some convergence criterion has been undercut.

These two tasks – first the optimization of the support functions and then that of the density kernel – build the two inner loops of the algorithm. They are contained in an outer loop and are executed alternately until overall convergence is reached. This convergence of the outer loop is based on the mean difference of the charge density per grid point between two iterations which must come below a given threshold.

A schematical overview of the method is given in Fig. 5.1. As will be shown in the next sections, this simple scheme is valid for both the trace minimization and the energy minimization mode.

5.1.1 Trace minimization

The trace minimization mode was the first one which was implemented. Its discussion will be rather detailed since many aspects will then apply as well to the other modes.

5.1.1.1 Keeping the support functions localized

As has been mentioned the optimization of the support functions is done in a non-self-consistent way using a fixed potential; only after some iterations in this optimization procedure the charge density and the potential are updated. This is similar to the mixing approach in the cubic version, where the trace of the Hamiltonian is minimized at a fixed potential, meaning that the target function is given by

$$\Omega = \sum_i f_i \langle \psi_i | \mathcal{H} | \psi_i \rangle \,. \tag{5.3}$$

Here f_i is the occupation number of orbital i which remains – as well as the potential – fixed during this minimization.
One could now formally apply the same procedure to the support functions and minimize the trace of the fixed Hamiltonian, i.e. the target function would be given by

$$\Omega = \sum_\alpha \langle \phi^\alpha | \mathcal{H} | \phi^\alpha \rangle \,. \tag{5.4}$$

The occupation numbers can all be set to one, since one is only interested in the support functions and not in any physical meaningful value of Ω.

If there were no localization constraints, this would be equivalent to the mixing approach in standard cubic DFT. Thus this procedure would finally lead to the ordinary Kohn-Sham orbitals (occupied ones and virtual ones) and the kernel elements in Eq. (3.15) – or the coefficients in Eq. (3.36), respectively – would be Kronecker deltas.

Unfortunately the situation is not that simple. In order to reach a linear scaling algorithm it is necessary to keep the support functions strictly localized. However several factors – e.g. the kinetic energy operator and the orthonormalization – tend to spread them out. Of course the strict localization can easily be achieved by cutting any contribution that extends outside of the localization region, but it is clear that the quality of the support functions will deteriorate if too much is cut in this way. However, since the number of support functions should be kept small, they need to be of high quality. As a consequence one has to find a way to keep them well localized while still preserving their high quality.

One possibility is to add a confining potential to the Kohn-Sham Hamiltonian which will tend to push the support functions back into their localization regions as soon as they spread out too much. As functional form for this confining potential a simple quartic function was chosen:

$$\mathcal{V}^\alpha(\mathbf{r}) = c^\alpha(\mathbf{r} - \mathbf{R}^\alpha)^4, \tag{5.5}$$

where $\mathbf{R}^\alpha$ is the center of the localization region in which the support function ϕ^α is contained. This functional form has the advantage that the confinement remains very small around the origin, but grows rapidly towards the edge of the localization region. In addition its evaluation is computationally cheap and can be added to the Kohn-Sham potential on the fly.
An illustration of the effect of this confining potential is given in Fig. 5.2. As can be seen the effective potential is indistinguishable from the true DFT potential close to the center, but increases rapidly towards the edges.

Using this effective potential one gets a new Hamiltonian which is different for each localization region. In the most general case each support function has its own local-

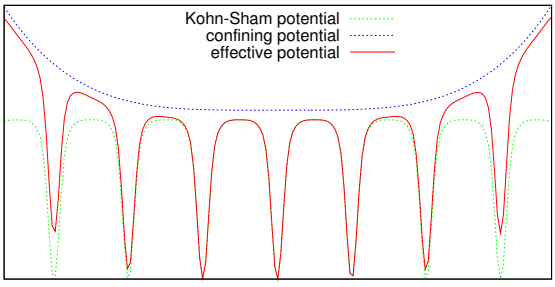


Figure 5.2: Illustration of the effect of the confining quartic potential. The plot shows the potential along one axis. Since an alkane was used for this plot, the Kohn-Sham potential is a periodic function. It is obvious that the effective potential is indistinguishable from the true Kohn-Sham potential at the center of the localization region, but increases rapidly towards the edges.

ization region and thus its own Hamiltonian

$$\mathcal{H}^\alpha = \mathcal{H}_{KS} + \mathcal{V}^\alpha. \tag{5.6}$$

Consequently the target function is now given by

$$\Omega^{tr} = \sum_\alpha \langle \phi^\alpha | \mathcal{H}^\alpha | \phi^\alpha \rangle \,. \tag{5.7}$$

From this expression the unconstrained gradient, which is – after applying the orthogonality constraint – used for the optimization of the support functions, can be readily calculated. Taking into account the distinction among covariant and contravariant quantities it is given by

$$\frac{1}{2} |g^\alpha\rangle = \frac{1}{2} \sum_\beta S^{\alpha\beta} \frac{\delta \Omega^{\mathrm{tr}}}{\delta \phi^\beta} = \sum_\beta S^{\alpha\beta} \mathcal{H}^\beta \, |\phi^\beta\rangle \,. \tag{5.8}$$

5.1.1.2 Improved convergence speed

There is – apart from the localization which is preserved –yet another reason why the trace minimization with the confining potential is advantageous, namely an improved convergence speed.

If all support functions are optimized using the same Hamiltonian – i.e. according to Eq. (5.4) – and without any localization constraints, then they are invariant under unitary transformations among themselves. Thus these transformations correspond to zero eigenvalues of the Hessian matrix which characterizes the optimization of the support functions. However, as soon as localization regions are introduced, this unitary invariance is slightly broken, and consequently these zero eigenvalue become finite, but still remain very small.
This fact is problematic for the convergence speed, which is characterized by the condition number κ. The exact relation between the convergence speed and the condition number depends on the specific optimization algorithm that is used, but the number of iterations which is required to converge is always monotonically increasing with κ, i.e. the larger the value of κ the slower is the convergence. Denoting by λ_{min} and λ_{max} the smallest and largest eigenvalue of the Hessian, respectively, the condition number is given by

$$\kappa = \frac{\lambda_{max}}{\lambda_{min}}. \tag{5.9}$$

Whereas the eigenvalues which are exactly zero do not enter this equation, the slightly non-zero eigenvalues do. Therefore the condition number can explode if λ_{min} is very tiny, which is the case if the unitary invariance is slightly violated.

On the other hand, the unitary invariance is strongly violated by the introduction of the confining potential. In this way these small eigenvalues will become larger, and thus the condition number will decrease, in this way accelerating the convergence.

This fact is illustrated in Fig. 5.3, where the mean gradient norm of the support functions is shown as a function of the number of iterations in the optimization procedure. The test system was a water droplet consisting of 150 atoms – thus amounting to 300 support functions – and the optimization algorithm was steepest descent with gradient feedback. The cutoff radius was set to a large value of 20 bohr in order to avoid the orthogonality problem, which will be discussed in more detail in Sec. 5.1.6. The prefactor for the confinement was chosen to be $1.25 \cdot 10^{-4}\,\mathrm{hartree/bohr^4}$, consequently the confining potential had a value of 20 hartree at the edges of the localization region.
As can be seen, the gradient norm decreases faster for the case where the confinement is used compared to the one where the Hamiltonian is the same for all localization regions, thus confirming that it is advantageous to artificially increase the small eigenvalues of the Hessian by strongly violating the unitary invariance

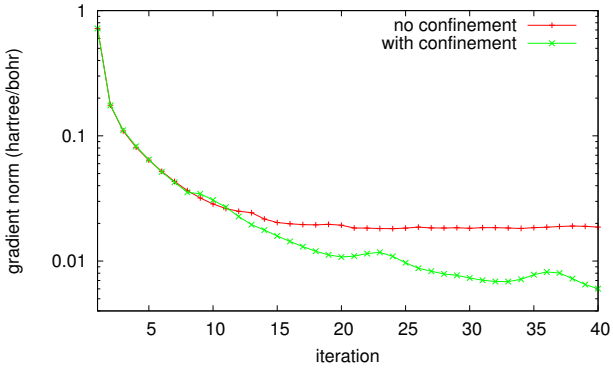


Figure 5.3: The gradient norm of the support functions as a function of the number of iterations in the optimization procedure. The introduction of the confining potential – thereby strongly violating the unitary invariance – helps to improve the convergence speed.

5.1.1.3 Preconditioning

An efficient preconditioning scheme is of utmost importance to get a fast and reliable minimization. To derive the preconditioning prescription one starts with the gradient, which is – including the normalization constraint – given by

$$|g^\alpha\rangle = \mathcal{H}\,|\phi^\alpha\rangle - \epsilon^\alpha\,|\phi^\alpha\rangle \tag{5.10}$$

with the Rayleigh quotient $\epsilon^\alpha = \langle\phi^\alpha|\mathcal{H}|\phi^\alpha\rangle$ and a Hamiltonian which is not further specified, i.e. the ordinary Kohn-Sham Hamiltonian as well as the one including the confinement can be used. For simplicity both the overlap matrix and the factor $\frac{1}{2}$ were omitted in the above expression. At a certain stage of the minimization procedure it can be assumed that an approximate solution for ϕ^α and ϵ^α has been found and that the true solution can be written as $\phi^\alpha + \Delta\phi^\alpha$ and $\epsilon^\alpha + \Delta\epsilon^\alpha$, respectively. Since the

error in the Rayleigh quotient is proportional to the square of the error in the support functions, it is justified to assume that $\Delta\epsilon^\alpha$ is zero and one is hence left with

$$|g^\alpha\rangle = \mathcal{H}\,|\phi^\alpha + \Delta\phi^\alpha\rangle - \epsilon^\alpha\,|\phi^\alpha + \Delta\phi^\alpha\rangle = 0. \tag{5.11}$$

A rearrangement of the terms gives

$$(\mathcal{H} - \epsilon^\alpha)\,|\Delta\phi^\alpha\rangle = -(\mathcal{H} - \epsilon^\alpha)\,|\phi^\alpha\rangle = -\,|g^\alpha\rangle\,. \tag{5.12}$$

Solving this equation for $|\Delta\phi^\alpha\rangle$ – symbolically written as $|\Delta\phi^\alpha\rangle = -(\mathcal{H} - \epsilon^\alpha)^{-1}\,|g^\alpha\rangle$ – yields as result a modification of the original gradient $|g^\alpha\rangle$. Consequently $|\Delta\phi^\alpha\rangle$ is called the preconditioned gradient and will from now on be denoted by $|\tilde{g}^\alpha\rangle$.
Using this preconditioned gradient will then allow powerful optimization steps with a step size of the order of one.

In practice it is not necessary to take the entire Hamiltonian to solve equation (5.12), but only the most important part which turns out to be the kinetic energy. So one is left with the task of solving

$$\left(-\frac{1}{2}\nabla^2 - \epsilon^\alpha\right)|\tilde{g}^\alpha\rangle = -\,|g^\alpha\rangle\,. \tag{5.13}$$

This is true as long as the standard Kohn-Sham Hamiltonian is used. However, it turned out to be important to include the confining potential into the preconditioning prescription as soon as such a confinement is present. This will only slightly modify Eq. (5.13) and the equation that must be solved for $|\tilde{g}^\alpha\rangle$ becomes

$$\left(-\frac{1}{2}\nabla^2 + c^\alpha(\mathbf{r} - \mathbf{R}^\alpha)^4 - \epsilon^\alpha\right)|\tilde{g}^\alpha\rangle = -\,|g^\alpha\rangle\,. \tag{5.14}$$

In practice both Eqs. (5.13) and (5.14) need only be solved approximately by performing a few steps of a Conjugent-Gradient procedure [73]. To this end the operators have to be applied to the gradient, i.e. one has to calculate $\left(-\frac{1}{2}\nabla^2 - \epsilon^\alpha\right)|g^\alpha\rangle$ and $\left(-\frac{1}{2}\nabla^2 + c^\alpha(\mathbf{r} - \mathbf{R}^\alpha)^4 - \epsilon^\alpha\right)|g^\alpha\rangle$, respectively. It has already been demonstrated that the application of the kinetic energy operator can efficiently be done thanks to the underlying wavelets basis and is given by convolutions of the scaling function / wavelet coefficients with some filters of finite length.
The application of the confinement operator is a bit more involved, but can as well be accomplished in a similar way, as will be shown in the following.

First the operator is split up in its contribution along the x, y and z direction. Denoting by the subscript 0 the center of the localization region this yields

$$\begin{aligned}(\mathbf{r} - \mathbf{R}_0)^4 &= \Big((x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2\Big)^2 \\ &= (x - x_0)^4 + (y - y_0)^4 + (z - z_0)^4 \\ &\quad + 2(x - x_0)^2(y - y_0)^2 + 2(x - x_0)^2(z - z_0)^2 + 2(y - y_0)^2(z - z_0)^2.\end{aligned} \tag{5.15}$$

Now all the six terms can be applied independently, and the procedure is analogous to the case of the derivative operators. Assuming a quantity $\Psi(x,y,z)$ that is expanded in a scaling function / wavelet basis according to Eq. (4.16), then, as an example, the scaling function coefficients after the application of the term $(x-x_0)^4$ are given by

$$\begin{aligned} sss'^{\{(x-x_0)^4\}}_{i_1,i_2,i_3} &= \iiint \phi(x-i_1)\phi(y-i_2)\phi(z-i_3)(x-x_0)^4\Psi(x,y,z)\,\mathrm{d}x\mathrm{d}y\mathrm{d}z \\ &= \sum_{j_1} a_{j_1-i_1} sss_{j_1,i_2,i_3} + \sum_{j_1} b_{j_1-i_1} dss_{j_1,i_2,i_3}, \end{aligned} \tag{5.16}$$

where a and b are filters of finite length that represent the application of the quartic potential. The prescriptions for the other coefficients and dimensions are similar. A detailed calculation of the filter elements is given in appendix A.2. Eq. (5.16) is completely identical to the case of the derivative operator, i.e. Eq. (4.17), just with different filters; consequently its derivation is the same as shown in appendix B.1 for that case.

The evaluation of the mixed terms, e.g. $(x-x_0)^2(y-y_0)^2$, is a bit more involved, but conceptually the same:

$$\begin{aligned} &sss'^{\{(x-x_0)^2(y-y_0)^2\}}_{i_1,i_2,i_3} \\ &\quad= \iiint \phi(x-i_1)\phi(y-i_2)\phi(z-i_3)(x-x_0)^2(y-y_0)^2\Psi(x,y,z)\,\mathrm{d}x\mathrm{d}y\mathrm{d}z \\ &\quad= \sum_{j_2} a_{j_2-i_2}\sigma\sigma\sigma^{i_1;a}_{i_1,j_2,i_3} + \sum_{j_2} a_{j_2-i_2}\delta\sigma\sigma^{i_1;b}_{j_1,j_2,i_3} + \sum_{j_2} b_{j_2-i_2}\sigma\delta\sigma^{i_1;a}_{i_1,j_2,i_3} + \sum_{j_2} b_{j_2-i_2}\delta\delta\sigma^{i_1;b}_{j_1,j_2,i_3} \end{aligned} \tag{5.17}$$

with

$$\begin{aligned} \sigma\sigma\sigma^{i_1;a}_{i_1,j_2,i_3} &= \sum_{j_1} a_{j_1-i_1} sss_{j_1,j_2,i_3}, & \delta\sigma\sigma^{i_1;b}_{i_1,j_2,i_3} &= \sum_{j_1} b_{j_1-i_1} dss_{j_1,j_2,i_3}, \\ \sigma\delta\sigma^{i_1;a}_{i_1,j_2,i_3} &= \sum_{j_1} a_{j_1-i_1} sds_{j_1,j_2,i_3}, & \delta\delta\sigma^{i_1;b}_{i_1,j_2,i_3} &= \sum_{j_1} b_{j_1-i_1} dds_{j_1,j_2,i_3}, \end{aligned} \tag{5.18}$$

where again a and b are some filters, representing this time the application of a quadratic potential. A detailed derivation is given in appendix B.2.

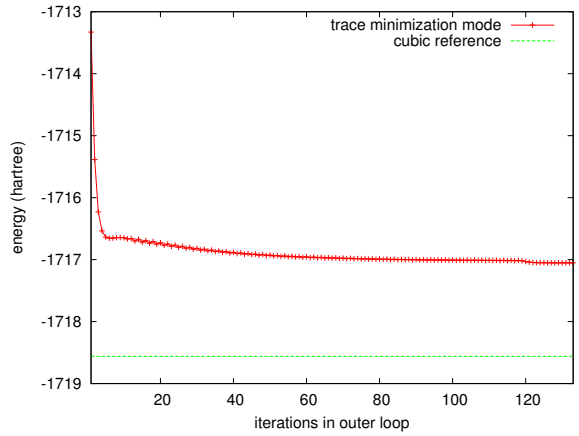
5.1.1.4 Moderate accuracy

In spite of the striking advantages of the trace minimization mode – namely that it keeps the support function well localized while still optimizing them and in addition accelerates the convergence – it is not well suited in practice.

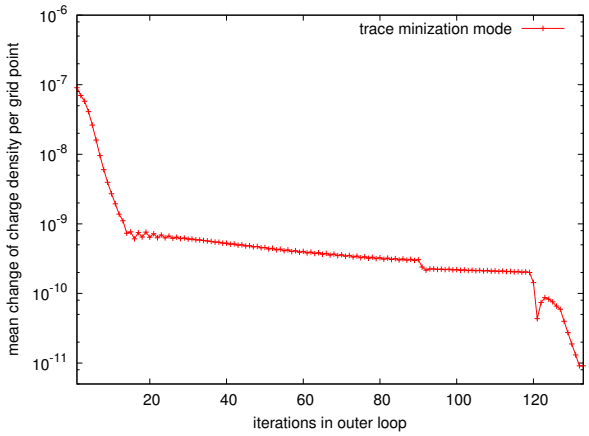
First of all the accuracy obtained in this way is only moderate. The reason is probably that the effect of the confining potential is too strong and as a consequence the support functions are not of high enough quality in order to allow a reasonable representation

of the density matrix or the Kohn-Sham orbitals, respectively, and to yield accurate results.
Furthermore the outer loop – i.e. the loop in which first the optimization of the support functions and then that of the density kernel is executed and which terminates as soon as a self-consistent solution has been found – shows an extreme slow convergence towards the end of the calculation.

To illustrate the problem a test run was done for a water droplet consisting of 300 atoms and exhibiting a diameter of about 35 bohr. The droplet was not relaxed, but still the forces on the atoms are only at around $10^{-1}\,\mathrm{hartree/bohr}$ and it is thus well suited for these tests. The cutoff radii for the support functions were set to 9 bohr for both atom kinds and the prefactor for the confining potential was chosen to be $1.5 \cdot 10^{-3}\,\mathrm{hartree/bohr^4}$, which corresponds to a value for the confinement of 9.84 hartree at the edges of the localization region. The optimization of the density kernel is done using the FOE method which will be presented in more detail in Sec. 5.2.3 and which is able to give accurate results, i.e. the moderate accuracy obtained here is entirely due to the trace minimization mode used for the optimization of the support functions.



(a) The energy calculated by the linear version as a function of the number of iterations in the outer loop. The deviations from the cubic reference value are substantial.



(b) The mean change of the charge density per grid point as a function of the number of iterations in the outer loop. Whereas the convergence is pleasing in the beginning, it becomes very slow after about 15 iterations.

Figure 5.4: Results for a run using the trace minimization mode for the optimization of the support functions. The x axis stands for the iterations of the outer loop, i.e. in each such iteration first the support functions and then the density kernel are optimized. The run was done for a water droplet consisting of 300 atoms; the cutoff radii for the support functions were set to 9 bohr and the confinement prefactors to $1.5 \cdot 10^{-3}\,\mathrm{hartree/bohr^4}$. The overall results are not satisfying. The sharp kink at nearly 120 iterations is due to a fixation of the support functions and is discussed in more detail in the text.

The results of the run are shown in Fig. 5.4. It is obvious that the total energy calculated by the linear scaling version deviates considerably from the value of the cubic reference calculation. In addition the convergence of the density in the outer loop is – except for the first few iterations – extremely slow.
Towards the very end of the run the minimization of the target function becomes unstable, i.e. the trace increases even if the step size for the optimization is decreased. This phenomenon will be discussed in more detail in Sec. 5.1.6. In such a case the code stops the optimization and fixes the support functions since a further improvement does not seem possible any more; as a consequence only the density kernel is optimized in this fixed set of support functions and the charge density quickly converges.

5.1.2 Energy minimization mode

As has been demonstrated in the previous section the presence of the confining potential has several – at least theoretical – advantages, but it will still prevent the linear version from yielding a result that is of equal quality than the cubic reference calculation.

So it seems that it is only possible to get highly accurate results if no confinement is used.
Furthermore switching off the confinement offers another interesting possibility. Since in this case the Hamiltonian is the same for all localization regions, it is not mandatory any more to minimize the trace. Instead it is possible to directly minimize the band-structure energy according to Eq. (3.27), i.e. the target function is given by

$$\Omega^{en} = \sum_{\alpha,\beta} K_{\alpha\beta} \langle \phi^\alpha | \mathcal{H} | \phi^\beta \rangle \,. \tag{5.19}$$

Since in this way exactly the same quantity as in the cubic version is minimized – just this time in the basis of the support functions –, it is to be expected that the accuracy which is obtained is much better.

The covariant gradient corresponding to this target function can readily be derived from Eq. (5.19) and is given by

$$\frac{1}{2} |g_\alpha\rangle = \sum_\beta K_{\alpha\beta} \mathcal{H} |\phi^\beta\rangle \,. \tag{5.20}$$

The contravariant gradient is – according to (5.2) – consequently given by

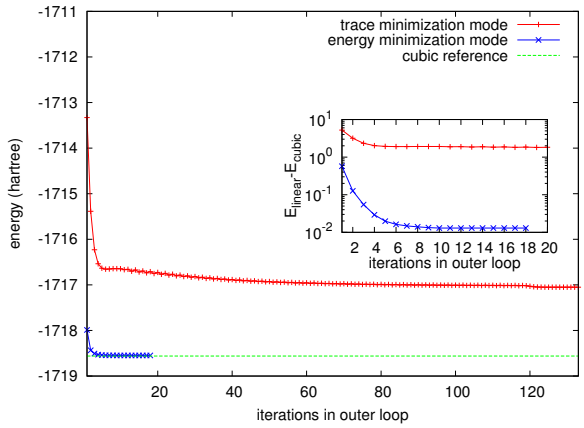
$$\frac{1}{2} |g^\alpha\rangle = \frac{1}{2} \sum_\beta S^{\alpha\beta} |g_\beta\rangle = \sum_{\beta,\gamma} S^{\alpha\beta} K_{\beta\gamma} \mathcal{H} |\phi^\gamma\rangle = \sum_\gamma K'^\alpha_{\ \gamma} |\phi^\gamma\rangle \tag{5.21}$$

with the modified density kernel $K'^{\alpha}_{\gamma} = \sum_{\beta} S^{\alpha\beta} K_{\beta\gamma}$.

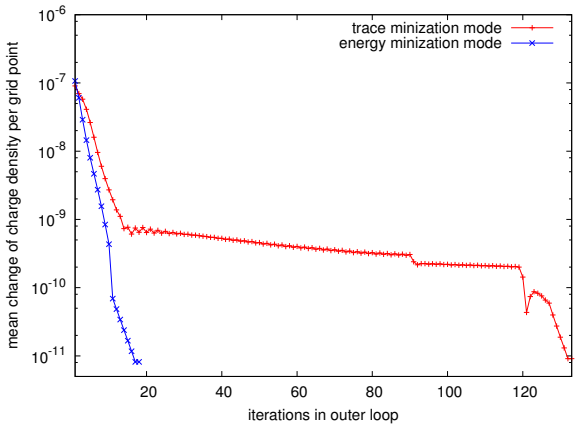
Apart from this modification the procedure is still the same as for the trace minimization mode, i.e. the flowchart in Fig. 5.1 is still valid.

The results of a run with this minimization mode are shown in Fig. 5.5, again for the case of the water droplet. In order to be able to compare these results with the ones obtained by the trace minimization mode, exactly the same parameters were used.
It is obvious that the energy comes much closer to the cubic value than with the trace minimization mode. In addition the convergence is much faster compared to that approach.

From these data it seems to be clear that the energy minimization mode is superior to the trace minimization mode.
On the other hand this method exhibits as well one severe shortcoming, namely that there is no more force that keeps the support functions well localized. In addition there are several driving forces that tend to delocalize the support functions, in particular the kinetic energy operator and the orthonormalization; this issue will be discussed in more detail in Sec. 5.1.6. It turns out that this can easily cause a breakdown of the minimization procedure optimizing the support functions in the first inner loop in



(a) The energy calculated by the linear scaling version as a function of the number of iterations in the outer loop. The energy minimization mode comes much closer to the value of the cubic reference calculation than the trace minimization mode.



(b) The mean change of the charge density per grid point as a function of the number of iterations in the outer loop. The energy minimization mode converges much faster. However this is – at least in parts – due to the early breakdown of the optimization procedure.

Figure 5.5: Comparison of the results for a run using the energy minimization mode and one using the trace minimization mode. The same test system and parameters were used as in Fig. 5.4. The results are much better for the energy minimization mode compared to the other one, from the viewpoint of both the accuracy and the convergence speed.

Fig. 5.1 since the support functions do not fit anymore into the localization regions. In such a case the target function Ω^{en} will always increase, no matter how small the step size used for the optimization algorithm – in practice often steepest descent – is chosen. Thus the code stops the optimization – i.e. it fixes the support functions – and the remaining optimization of the density kernel until the achievement of the overall convergence has to be carried out with the given set at that stage.
This situation actually arose for the test run, where the optimization broke down at the 12th iteration of the outer loop.

Since in general the support functions exhibit already a very good quality if such a breakdown occurs, it is still possible to come reasonably close to the cubic reference calculation, as has been demonstrated by the test run. Thus it is not that problematic for one single run.
However it can cause considerable problems if one wants to determine energy differences between several structures. There is the hope that the energies calculated by the linear version have a more or less constant offset compared to the values from the cubic reference calculations, and thus the energy differences between different structures should be pretty much the same for the cubic and the linear version since this offset cancels. However if the optimization breakdown does not occur at the same stage of quality for the various configurations, then this offset may be different and the energy differences thus not as good as hoped for.

5.1.3 Mixed mode

It is intuitively clear that the breakdown in the inner loop optimizing the support functions, which was described briefly in the previous section, happens more likely if the support functions are not yet well adapted to their chemical environment, since in this case they will still undergo heavy modifications.

Therefore it might be advantageous to combine the two modes presented so far, meaning that the first few iterations of the outer loop use the trace minimization mode and the remaining ones the energy minimization mode. As a result the support functions should already be adapted to their chemical environment to some extent while still being localized when the energy minimization mode starts, and therefore not undergo heavy modifications after the confining potential is switched off. In this way the problems related to this breakdown can hopefully be diminished, meaning that they do not occur or at least happen much later such that the quality of the support functions is always pretty much the same.

A flowchart of this approach is shown in Fig. 5.6. As can be seen, the procedure is

basically the same as for the trace minimization and the energy minimization modes whose flowcharts are shown in Fig. 5.1, with the exception that there are now two different loops for the optimization of the support functions. The switch from the trace minimization mode to the energy minimization mode is either done as soon as the maximal number of iterations using the first mode is reached or as soon as the mean change in the charge density per grid point between two iterations of the outer loop is below a given threshold.

A comparison of all three approaches is shown in Fig. 5.7, again using the same test system and parameters. The prefactor for the confinement which is used in the first iterations of the mixed mode – i.e. where the trace is minimized – was set to $3 \cdot 10^{-3}\,\mathrm{hartree/bohr^4}$, which is higher compared to the one used for the pure trace minimization mode; the idea is to strongly confine the support functions in the beginning in order to be in a position to tolerate some spreading which will inevitably occur as soon as the confinement is switched off. In this test the threshold for the switch from trace minimization to energy minimization was chosen such that two iterations of trace minimization were performed.
It is obvious that the mixed mode leads to the same final energy as the approach where

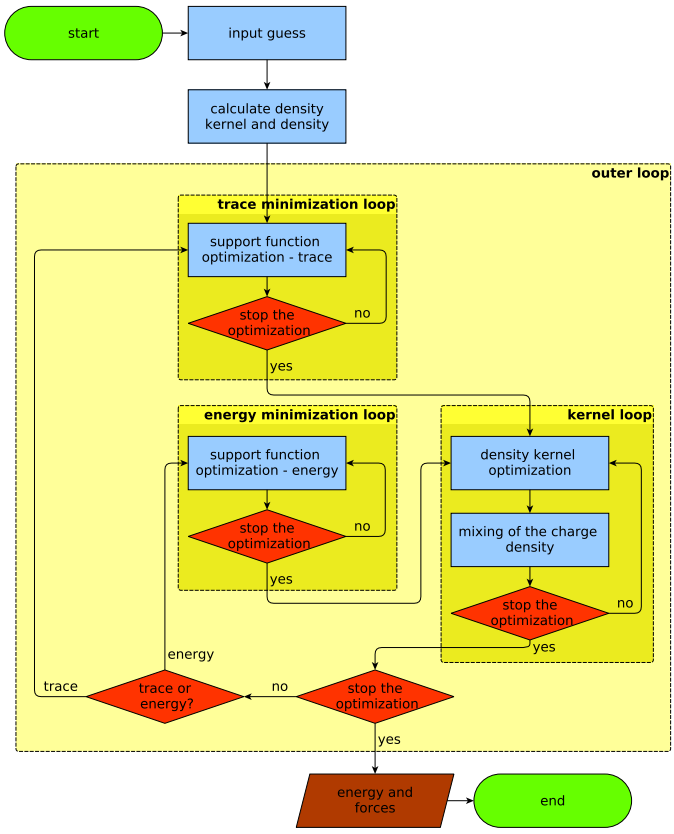
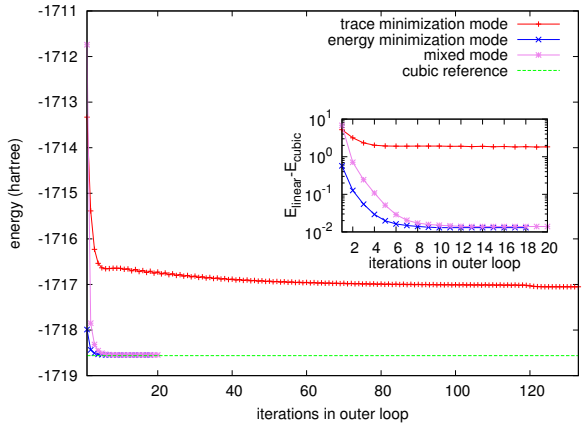
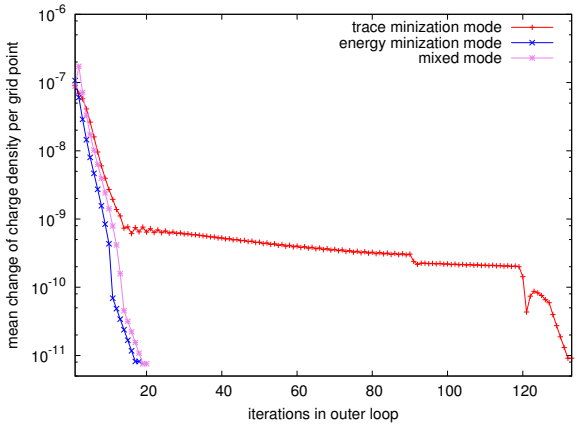


Figure 5.6: Flowchart to illustrate the mixed mode, which is a combination of trace minimization and energy minimization. The flowchart is similar to the one shown in Fig. 5.1, apart from the fact that there are now two loops for the optimization of the support functions. In the beginning they are optimized using the trace minimization mode including the confining potential in order to let them adapt themselves to the chemical environment while still remaining well localized. After a few iterations of the outer loop using this mode, the remaining iterations employ the energy minimization mode for the optimization of the support functions in order to get more accurate results.
The kernel loop is not modified and still the same as in Fig. 5.1.



(a) The energy calculated by the linear version as a function of the number of iterations in the outer loop. The final result for the mixed mode is more or less the same as for the energy minimization mode, even though the convergence is slightly delayed.



(b) The mean change of the charge density per grid point as a function of the number of iterations in the outer loop. The mixed mode and the energy minimization mode exhibit a similar convergence speed, but the mixed mode is slightly shifted to the right.

Figure 5.7: Comparison of the results for one run with the mixed mode, one with the energy minimization mode and one with the trace minimization mode. The same test system and parameters were used as in Figs. 5.4 and 5.5. The results for the mixed mode and the energy minimization mode are comparable from the viewpoint of both the accuracy and the convergence speed.

only the energy is minimized, however with a slightly delayed convergence.

On the other hand this approach has as well some shortcomings.
First of all the breakdown of the optimization occurs as well, even if it happens later. For this test the breakdown occurred at the 15th iteration of the outer loop, which is only slightly better compared to the energy minimization mode; thus only a scant improvement of the stability can be gained.
However the real problems lie somewhere else. The switch from the trace minimization mode to the energy minimization mode is rather drastic, which might potentially lead to some problems. In addition it is somehow arbitrary when this switch should take place. As mentioned it is based on the same criterion as the overall convergence of the outer loop, namely the mean change in the charge density per grid point. However choosing this threshold too small – i.e. more iterations using the trace minimization mode are performed – will yield a worse result, as illustrated in Fig. 5.8. The first curve, where the switch was performed at a mean change of 10^{-7}, is identical to the one shown in Fig. 5.7a and resulted in two iterations with trace minimization. For the next one the switch was performed at 10^{-8}, resulting in 8 iterations of trace minimization, and for the last one the switch was done at 10^{-9}, resulting in 15 iterations of trace minimization. It is obvious that the final results of the three runs do not coincide and

that the deviation from the cubic reference calculation is larger the more trace minimization steps were done, meaning that the error introduced by confining the support functions for too long could not be cured anymore.

Consequently there is a thin line between choosing too few iterations – causing the mixed mode to be similar to the energy minimization mode including its shortcomings – and too many iterations – making it similar to the trace minimization mode and yielding only moderate results –, which makes the usage of the mixed mode a bit involved.

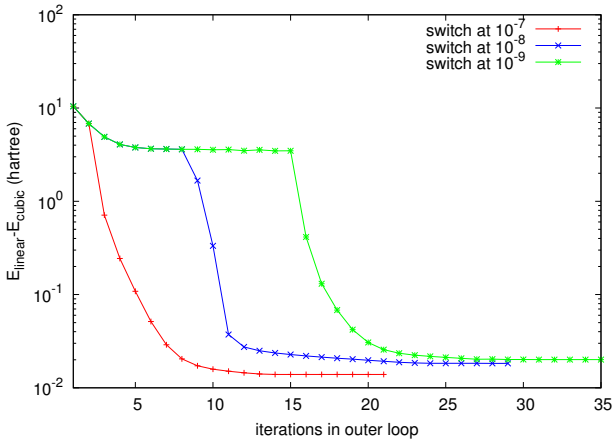


Figure 5.8: Comparison of different thresholds to switch from the trace minimization to the energy minimization mode. The switch was done as soon as the mean change of the charge density per grid point between two iterations of the outer loop was below this threshold. It is obvious that choosing this value too small will result in runs which are less accurate and in addition converge more slowly.

5.1.4 Hybrid mode

The previous section has shown that it might be a good idea to combine the trace minimization and the energy minimization in order to retard the breakdown of the support function optimization. On the other hand a drastic switch from the one to the other is not desirable, and in addition the switching criterion is not evident.
Thus it is probably better to use an approach that can smoothly transform one method into the other one.

To this end the two target functions are briefly noted again. For the trace minimization mode it is given by

$$\Omega^{tr} = \sum_{\alpha} \langle \phi^{\alpha} | \mathcal{H}^{\alpha} | \phi^{\alpha} \rangle \,, \tag{5.22}$$

where $\mathcal{H}^{\alpha}$ is the Hamiltonian including the confining potential, whereas for the energy minimization mode it is given by

$$\Omega^{en} = \sum_{\alpha,\beta} K_{\alpha\beta} \, \langle \phi^{\alpha} | \mathcal{H} | \phi^{\beta} \rangle \,. \tag{5.23}$$

These two methods can now be combined to get the so-called hybrid mode with the target function

$$\Omega^{hy} = \sum_{\alpha} K_{\alpha\alpha} \langle \phi^{\alpha} | \mathcal{H}^{\alpha} | \phi^{\alpha} \rangle + \sum_{\beta \neq \alpha} K_{\alpha\beta} \langle \phi^{\alpha} | \mathcal{H} | \phi^{\beta} \rangle \,. \tag{5.24}$$

By decreasing the prefactor of the confining potential – i.e. the value of c^{α} in Eq. (5.5) –, the hybrid target function Ω^{hy} will be smoothly transformed into the energy target function Ω^{en}. The prescription how the confinement should be reduced will be described in Sec. 5.1.4.1.

A flowchart of this method is shown in Fig. 5.9. The method is quite similar to the trace or energy minimization approach which are illustrated in Fig. 5.1, with the difference that the confinement is continuously adjusted each time the loop which optimizes the support functions is re-entered.

A comparison of the hybrid mode with all the other modes – again using the same system and parameters – is shown in Fig. 5.10. For the hybrid mode the prefactor for the initial confining potential was set to $3 \cdot 10^{-3}\,\mathrm{hartree/bohr^4}$; during the run it was then continuously decreased and had a value of $3.89 \cdot 10^{-13}\,\mathrm{hartree/bohr^4}$ in the iteration where the support functions were optimized the last time.

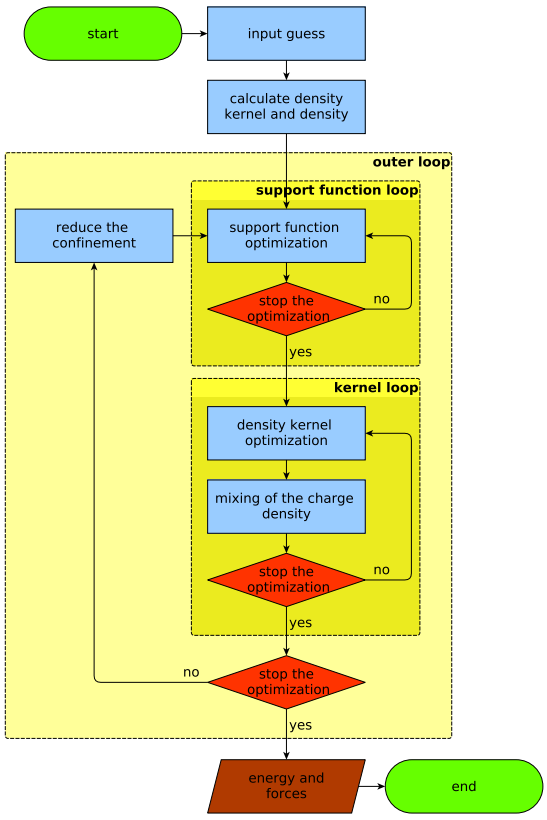
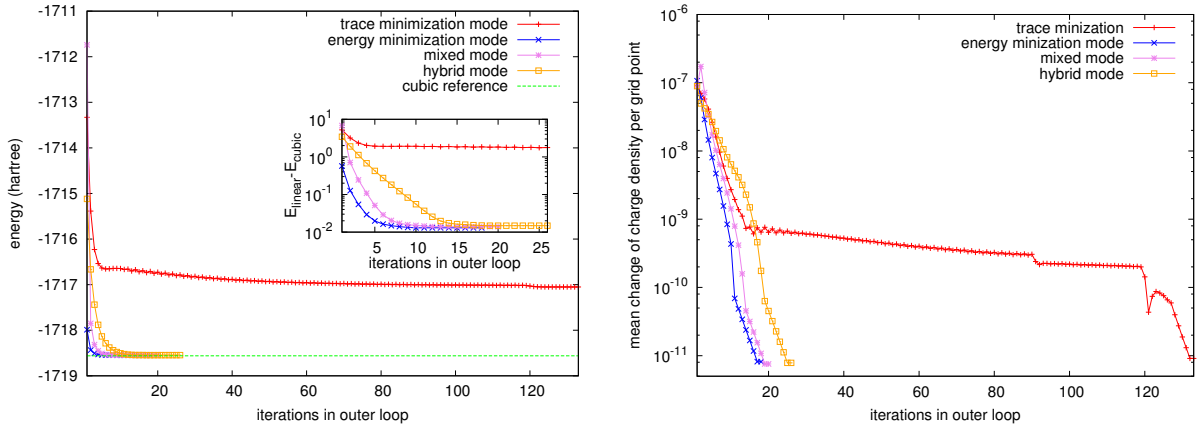


Figure 5.9: Flowchart to illustrate the hybrid mode. The procedure is very similar to the trace minimization mode and the energy minimization mode – whose flowcharts are shown in Fig. 5.1 –, i.e. there is one outer loop and two inner loops optimizing first the support functions and then the density kernel, respectively. The only difference is that the value of the confinement is reduced each time the loop optimizing the support functions is re-entered. The kernel loop is not affected by the hybrid mode and is still the same as for the other modes.

As can be seen the final energy is again comparable to the energy minimization and the mixed mode, but the convergence is slightly slower. Unfortunately the breakdown of the support function optimization happens as well, but at least much later compared to the to other modes, namely at the 21st iteration of the outer loop.

To summarize one can conclude that the hybrid mode gives identical results as the energy minimization mode and the mixed mode; furthermore it seems to exhibit a higher stability, however at the cost of a slightly slower convergence.



(a) The energy calculated by the linear scaling version as a function of the number of iterations in the outer loop. The final result of the hybrid mode is identical to the ones yielded by the energy minimization mode and the mixed mode, but the number of iterations that is required to reach convergence is slightly larger.

(b) The mean change of the charge density per grid point as a function of the number of iterations in the outer loop. The convergence speed of the hybrid mode is identical to the energy minimization mode and the mixed mode in the end, but slightly slower in the beginning.

Figure 5.10: Comparison of the results for a of run using the hybrid mode with the results of the runs using the other modes. Again the same test system was used as in Figs. 5.4, 5.5 and 5.7. Compared to the energy minimization mode and the mixed mode, the results for the hybrid mode are comparable from the viewpoint of the energy, but the convergence is slightly slower.

5.1.4.1 How to reduce the confinement

The value of the confining potential will be reduced during the calculation, thereby smoothly transforming the target function from the hybrid expression to the energy expression. The question is how the prefactor c^{α} should be reduced.

The most naive way would be to reduce the parameter as soon as the support functions are converged for the value c^{α} that is currently used, indicated by the norm of

the gradient which has to come below a given threshold. Unfortunately this does not work due to the influence of the localization regions, which will prevent the gradient from reaching small values even though the energy saturates; some more details on this issue will be discussed as well in the Secs. 5.1.6 and 6.5.1. Therefore a slightly different approach is used.

To derive this prescription it is assumed that the difference of the target function between iteration n and $n+1$ of the minimization procedure can be approximated to first order by the gradient of the target function with respect to the support functions times the change in the support functions, i.e.

$$\Delta\Omega'_{(n)} = \sum_{\alpha} \langle g^{\alpha}_{(n)} | \Delta\phi^{\alpha}_{(n)} \rangle \,, \tag{5.25}$$

where $|\Delta\phi^{\alpha}_{(n)}\rangle$ is the change which the support functions will undergo between iteration n and $n+1$, i.e.

$$|\Delta\phi^{\alpha}_{(n)}\rangle = |\phi^{\alpha}_{(n+1)}\rangle - |\phi^{\alpha}_{(n)}\rangle \,, \tag{5.26}$$

and $|g^{\alpha}_{(n)}\rangle$ is the gradient including the orthonormality constraint at iteration n, which will be derived in detail in Sec. 5.1.8.

As already mentioned, the gradient of the target function will never go to zero due to the influence of the localization regions which is competing with the orthogonality constraint imposed on the support functions. Since the expected decrease of the target function is directly proportional to the gradient, the same considerations apply as well. Consequently $\Delta\Omega'_{(n)}$ will not go down to zero. On the other hand the actual change in the target function which is observed and which is denoted by $\Delta\Omega_{(n)}$ will go to zero if the limit for the localization region and the currently used confining potential is reached, meaning that a further optimization is not possible anymore. If this is the case, the only possibility to further minimize the target function is to decrease the value of the confining potential.

One can now make a virtue out of necessity and use these properties in order to derive a prescription how the prefactor of the confinement should be reduced. At each step of the minimization procedure the ratio of the actual decrease in the target function and the previous estimate is determined:

$$\kappa = \frac{\Delta\Omega_{(n)}}{\Delta\Omega'_{(n)}}. \tag{5.27}$$

When re-entering the optimization loop for the support functions the next time, the prefactor for the confinement is then multiplied with the last value of κ of the previous optimization loop, i.e.

$$c^{\alpha}_{new} = \kappa c^{\alpha}_{old}. \tag{5.28}$$

If κ is of the order of one, this means that there is still some scope for the support functions with the current value of the confining potential and it should therefore be kept as it is. If, on the other hand, κ is much smaller, this means that it is hardly possible to further improve the support functions and the magnitude of the confining potential should consequently be decreased.

An illustration of the effect of reducing the confinement in this way is shown in Fig. 5.11, where the value of κ is plotted as a function of the number of iterations in the outer loop. The red curve shows the behavior if the strength of the confinement is reduced as specified by Eq. (5.28), whereas the green curve shows the evolution of κ if the confinement is held constant.
As can be seen, the second possibility is much less stable, indicated by the value of κ becoming negative, meaning that the actual change of the target function, $\Delta\Omega_{(n)}$, became positive, whereas the estimate, $\Delta\Omega'_{(n)}$, was negative.
The red curve, on the other hand, shows the expected behavior. In the beginning the value of κ is close to one, meaning that the support functions can still be optimized with the strong confinement, whereas towards the end it becomes very small, meaning that the confinement should be reduced.

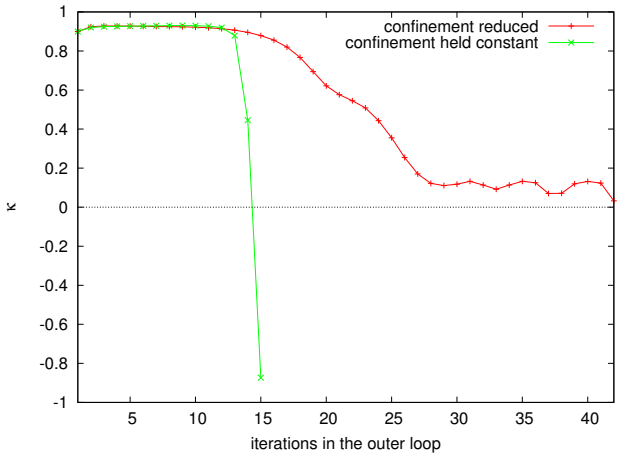


Figure 5.11: The value of κ – as defined by Eq. (5.27) – as a function of the number of iterations in the outer loop. The test system was an alkane consisting of 302 atoms; the cutoff radius for the support functions was set to 9 bohr and the initial prefactor for the confinement to $3 \cdot 10^{-3}$ hartree/bohr^4. A negative value of κ means that the target function increased even though the estimation predicted a decrease.

5.1.4.2 Preconditioning with the hybrid method

The minimization of the target function will only be efficient if a good preconditioning scheme is available. Finding such a good scheme is often some trial-and-error process. Therefore it is not guaranteed that the following scheme will be the ultimate solution, but still it is a reasonable proposal.

If there was no confinement, the preconditioning would be done by solving the equa-

tion

$$\left(-\frac{1}{2}\nabla^2+\epsilon^\alpha\right)|\tilde{g}^\alpha\rangle=|g^\alpha\rangle \tag{5.29}$$

for the preconditioned gradient $|\tilde{g}^\alpha\rangle$. However this prescription tends to spread out the support functions, which is not desirable if a confinement is used. Therefore it turned out that it is better to include the confining potential in the above equation in the latter case, as has been described in Sec. 5.1.1.3:

$$\left(-\frac{1}{2}\nabla^2+\epsilon^\alpha+c^\alpha(\mathbf{r}-\mathbf{R}^\alpha)^4\right)|\tilde{g}^\alpha\rangle=|g^\alpha\rangle\,. \tag{5.30}$$

When minimizing the new hybrid target function, there are contributions from both Hamiltonians with and without the confinement. However the preconditioning cannot be done independently for both parts, since it is applied to the gradient after the orthonormality constraint, which will mix these two contributions.

One possible solution to circumvent this problem is the following:
First a gradient $|G^\alpha\rangle$ is defined which would arise if there were only contributions from the Hamiltonian including the confinement:

$$|G^\alpha\rangle=\mathcal{H}^\alpha|\phi^\alpha\rangle-\lambda^\alpha|\phi^\alpha\rangle \tag{5.31}$$

with $\lambda^\alpha=\langle\phi^\alpha|\mathcal{H}^\alpha|\phi^\alpha\rangle$, i.e. only a normalization constraint, but no orthogonality constraint is applied.
Next the projectors on this confining gradient and its orthogonal complement are defined:

$$\begin{aligned}P_c^\alpha&=\frac{1}{\langle G^\alpha|G^\alpha\rangle}|G^\alpha\rangle\langle G^\alpha|\,,\\ P_{nc}^\alpha&=1-P_c^\alpha=1-\frac{1}{\langle G^\alpha|G^\alpha\rangle}|G^\alpha\rangle\langle G^\alpha|\,,\end{aligned} \tag{5.32}$$

where "c" stand for confining and "nc" for non-confining. Using these projectors the gradient is split up in two parts:

$$\begin{aligned}|g_c^\alpha\rangle&=P_c^\alpha|g^\alpha\rangle\,,\\ |g_{nc}^\alpha\rangle&=P_{nc}^\alpha|g^\alpha\rangle\,.\end{aligned} \tag{5.33}$$

Now two separate preconditioning equations can be solved for both parts:

$$\begin{aligned}\left(-\frac{1}{2}\nabla^2+\epsilon^\alpha+c^\alpha(\mathbf{r}-\mathbf{R}^\alpha)^4\right)|\tilde{g}_c^\alpha\rangle&=|g_c^\alpha\rangle\,,\\ \left(-\frac{1}{2}\nabla^2+\epsilon^\alpha\right)|\tilde{g}_{nc}^\alpha\rangle&=|g_{nc}^\alpha\rangle\,.\end{aligned} \tag{5.34}$$

After these equations have been solved for $|\tilde{g}_c^\alpha\rangle$ and $|\tilde{g}_{nc}^\alpha\rangle$, the two solutions can be added to get the final preconditioned gradient:

$$|\tilde{g}^\alpha\rangle = |\tilde{g}_c^\alpha\rangle + |\tilde{g}_{nc}^\alpha\rangle\,. \tag{5.35}$$

In the limit where c^α is zero, this will be equivalent to solving directly Eq. (5.29):

$$\begin{aligned} |\tilde{g}^\alpha\rangle &= \left(-\frac{1}{2}\nabla^2 + \epsilon^\alpha\right)^{-1} |g^\alpha\rangle \\ &= \left(-\frac{1}{2}\nabla^2 + \epsilon^\alpha\right)^{-1} |g_c^\alpha + g_{nc}^\alpha\rangle \\ &= \left(-\frac{1}{2}\nabla^2 + \epsilon^\alpha\right)^{-1} |g_c^\alpha\rangle + \left(-\frac{1}{2}\nabla^2 + \epsilon^\alpha\right)^{-1} |g_{nc}^\alpha\rangle \\ &= |\tilde{g}_c^\alpha\rangle + |\tilde{g}_{nc}^\alpha\rangle\,. \end{aligned} \tag{5.36}$$

It must be noted that this prescription is just a suggestion and it might well be that other procedures are more suited in practice. This can also be seen from Fig. 5.12, which shows a comparison of the performance of the three available preconditioning schemes, namely the one used for the trace minimization mode (Eq. (5.30)), the one for the energy minimization mode (Eq. (5.29)) and the one just described. The target function was in all cases the same – namely the hybrid expression – and only the preconditioning was done in a different way. As can be seen the difference among the three methods is rather small and one can thus not conclude that one approach is superior to the other ones.

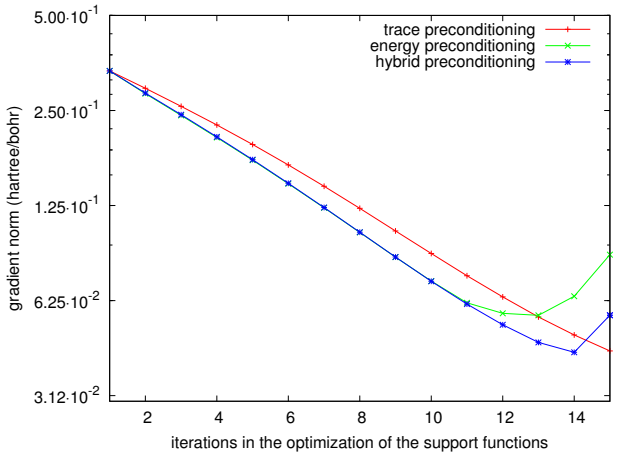


Figure 5.12: Comparison of the three available preconditioning scheme. "trace preconditioning" stands for the preconditioning according to Eq. (5.30), "energy preconditioning" for the one according to Eq. (5.29), and "hybrid preconditioning" for the one described in Sec. 5.1.4.2 with the final result of Eqs. (5.34) and (5.35). At the 16-iteration the optimization for the run using the hybrid preconditioning broke down. The test system was an alkane consisting of 302 atoms; the cutoff for the support functions was set to 9 bohr and the prefactor for the confinement to $3 \cdot 10^{-3}\,\mathrm{hartree/bohr^4}$.

5.1.5 Input guess

In the previous section several methods to optimize the support functions have been presented. However, in order to be able to carry out any optimization, first a reasonable input guess has to be created.

In the cubic version of BigDFT, the input guess for the Kohn-Sham orbitals is done as a linear combination of atomic orbitals (LCAO). This means that first the atomic orbitals – denoted by χ^α – for all atoms are generated and the charge density is calculated as the superposition of the atomic charge densities $\rho_I(\mathbf{r})$:

$$\rho(\mathbf{r}) = \sum_I \rho_I(\mathbf{r}) = \sum_\alpha f_\alpha |\chi^\alpha(\mathbf{r})|^2, \tag{5.37}$$

where f_α denotes the occupation number of the atomic orbital χ^α. Using this charge density a Hamiltonian $\mathcal{H}$ can be constructed and be represented in the basis of the atomic orbitals:

$$H^{\alpha\beta} = \langle \chi^\alpha | \mathcal{H} | \chi^\beta \rangle . \tag{5.38}$$

Defining furthermore the overlap matrix $S^{\alpha\beta} = \langle \chi^\alpha | \chi^\beta \rangle$ and solving the generalized eigenvalue problem

$$\mathbf{H}\mathbf{c}_i = \epsilon_i \mathbf{S}\mathbf{c}_i \tag{5.39}$$

gives the expansion coefficients of the Kohn-Sham orbital ψ_i in terms of the atomic orbitals:

$$\psi_i(\mathbf{r}) = \sum_\alpha c_{i\alpha} \chi^\alpha(\mathbf{r}). \tag{5.40}$$

This approach is analogous to the one briefly presented in Sec. 3.3.5 for the linear scaling version, just with the difference that the support functions ϕ^α are this time replaced by the atomic orbitals χ^α.

Since this method generates in general quite good input guesses for the Kohn-Sham orbitals, there is the hope that a similar method can also be used for the linear version, i.e. that the support functions can as well be written as a linear combination of the atomic orbitals:

$$\phi^\alpha(\mathbf{r}) = \sum_\beta d^\alpha_\beta \chi^\beta(\mathbf{r}). \tag{5.41}$$

Because the solution of the eigenvalue problem in Eq. (5.39) leads in general to extended orbitals which do not fit into a given localization region, the procedure has to be slightly adapted. Since, as has been discussed, the trace minimization is able to optimize the support functions to some degree while still keeping them well localized, it is an obvious choice to perform the same procedure in the basis of the atomic orbitals. Consequently one has to construct for each localization region γ its own Hamiltonian matrix

$$H^{\gamma;\alpha\beta} = \langle \chi^\alpha | \mathcal{H}^\gamma | \chi^\beta \rangle , \tag{5.42}$$

where the Hamiltonian $\mathcal{H}^\gamma$ is given by Eq. (5.6). Now a trace minimization can be performed in the basis of the atomic orbitals, i.e. the expansion coefficients $\mathbf{d}^\alpha$ of the support functions in terms of the atomic orbitals are given by the minimization of

$$\Omega = \sum_\alpha \langle \mathbf{d}^\alpha | \mathbf{H}^\alpha | \mathbf{d}^\alpha \rangle \tag{5.43}$$

under the constraint

$$\mathbf{d}^T \mathbf{S} \mathbf{d} = \mathbf{I}, \tag{5.44}$$

where $\mathbf{d}$ is the matrix constructed out of the vectors $\mathbf{d}^\alpha$, $\mathbf{S}$ the overlap matrix among the atomic orbitals and $\mathbf{I}$ the identity matrix. In this way the input guess for the support functions should be well adapted to its chemical environment while still remaining fairly localized. Furthermore this method has the advantage that there is no constraint on the number of support functions that can be generated.

In spite of these striking advantages it turned out that an alternative approach that simply uses the atomic orbitals in their original form yields an input guess of equal quality. Furthermore it is of course much cheaper since the entire minimization procedure can be avoided.
The limitation that the number of support functions which can be generated in this way is at the moment restricted to be a complete shell of the atom on which they will be centered does not seem to be a serious issue. The other restriction, namely that there is no flexibility with respect to the localization of the support functions – e.g. to center them in between to atoms – might be an issue to be addressed in the future.

Since these restrictions are – at least at the moment – not heavy limitations, the input guess using simply the atomic orbitals is consequently the method of choice

5.1.6 Orthogonality problem

It has been demonstrated that any method that eventually minimizes the energy – i.e. the energy minimization mode, the mixed mode and the hybrid mode – yields considerably better results than only minimizing the trace using the confinement. However, as has been shown as well, minimizing the energy may lead to an early breakdown of the optimization of the support functions since there is no more force that counteracts their extension.

One of the main reasons for the spreading of the support functions is the orthogonality that is imposed on them. If it happens that they become too extended in the course of the optimization procedure due to the orthogonalization it may occur that all of a

sudden the chosen localization radius becomes too small and consequently a large part of the support functions has to be cut at the boundaries of the localization regions in order to retain the strict localization. This may lead to a deterioration of the quality of the support functions and as a consequence the value of the target function will raise instead of decrease.

These problems due to the orthogonalization are illustrated in Fig. 5.13. Here the target function – which is the energy in this case – is shown as a function of the iterations in the inner loop optimizing the support functions. In order to isolate the effect of the orthogonalization, the support functions were optimized using steepest descent with a step size of zero, i.e. they just repeatedly underwent orthogonalizations without being modified otherwise. This means that the target function is actually shown as a function of the number of orthogonalizations.
If there were no localization constraints, the support functions should be exactly – up to numerical noise – orthonormal after the first step and subsequent orthogonalizations should not modify the energy any more. However, if there are localization constraints, the support functions will in the most general case not be exactly orthonormal and thus the target function keeps changing in every step.

As can be seen the energy increases considerably after each orthonormalization step for small cutoff radii. For instance for the smallest cutoff radius of 5 bohr, the energy increased by 0.039 hartree in the first step, which corresponds to 0.009% of the total value, and even the subsequent steps raise the energy each time by values of the order of 10^{-3} hartree.
For larger cutoffs the energy increase diminishes, as expected, but it still remains a considerable problem. Furthermore the plot shows that – even if the effect becomes smaller and smaller – the energy increases in each step since, as explained before, the

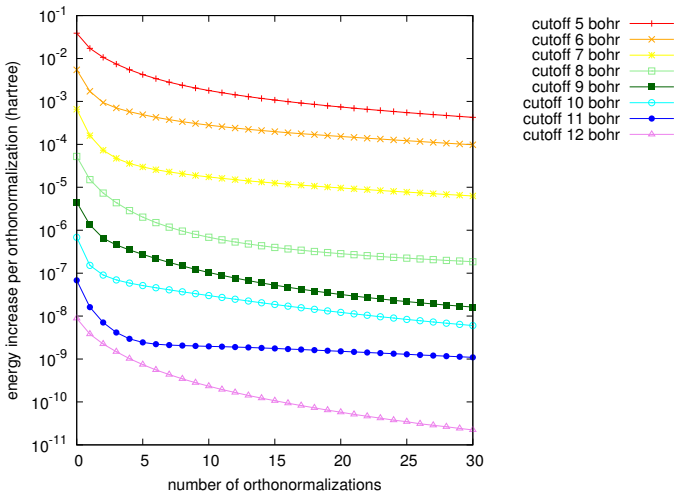


Figure 5.13: Illustration of the orthogonality problem for a system consisting of 25 water molecules. The plot shows the energy increase after each orthonormalization step. In between the orthonormalizations the support functions were not modified except for the inevitable cutting at the boundaries. As expected the problem is worst for small cutoff radii. Furthermore the energy increase becomes smaller with each orthonormalization step that is performed, but it never vanishes completely.

support functions will never be exactly orthogonal.

It is obvious that this energy increase will create problems at some point. If the support functions are already rather well converged, then only very little can be gained by optimizing them according to the gradient. On the other hand the orthonormalization will always raise the value of the target function. Thus it will happen that the value of the target function starts to increase even if the gradient is not yet zero.
Reducing the step size for the optimization, which is usually done as soon as an optimization runs into trouble, will in general not help, but rather aggravate the problem since the energy increase stemming from the orthogonalization will preponderate even more.

If modifying the step size should help, then it should rather be enlarged in the hope to optimize the support functions to a larger extent such that the energy increase due to the orthogonalization can be compensated.
However it is admittedly quite hazardous to increase the step size for an optimization that gets into a mess. For this reason such an energy increase which can not be eliminated by reducing the step size – and thus being identified as a problem stemming from the orthogonalization – is considered as a breakdown of the optimization procedure and leads to a fixing of the support functions, meaning that the following density kernel optimizations until the achievement of overall convergence are all carried out using this fixed set of support functions.

Due to these difficulties its is also quite involved to find a convergence criterion for the support function optimization that depends on the gradient. More details on this are given in Sec. 6.5.1.

These problems are in some sense the price that has to be payed if one wants to work with a set of orthonormal support functions. It is not surprising that they arise since – as already mentioned in Sec. 3.3.6 – orthogonality and localization are in general two contradicting properties.
However completely releasing the orthogonality would open new problems and bottlenecks – for instance necessitating the introduction of a larger cutoff radius for the density kernel or complicating the calculation of $\mathbf{S}^{-1}$ and $\mathbf{S}^{-1/2}$, which are used in various locations – that are probably not easier to overcome than this one.
A possible solution might be to only relax the orthogonality as soon as a further optimization of the support functions with the orthogonality constraint is not possible anymore. If they are already reasonably converged and will thus not undergo heavy changes from that point on, the approximate orthogonality should be preserved in this way.

5.1.7 Orthogonalization

As already mentioned several times the support functions are required to be orthonormal. This is accomplished by means of a Löwdin orthonormalization [74], which generates a set of orthonormal support functions $\tilde{\phi}^\alpha$ out of the non-orthonormal ones ϕ^α:

$$|\tilde{\phi}^\alpha\rangle = \sum_\beta (S^{-1/2})^{\alpha\beta} |\phi^\beta\rangle \tag{5.45}$$

with the overlap matrix $S^{\alpha\beta} = \langle\phi^\alpha|\phi^\beta\rangle$. Whereas the calculation of the latter can be done with linear scaling thanks to the strict localization of the support functions – more details on this are given in Sec. 6.3.1.1 – the calculation of $\mathbf{S}^{-1/2}$ remains a bottleneck since it requires a diagonalization of the matrix. Consequently a way to circumvent this obstacle has to be found.

5.1.7.1 Taylor approximation

Even if the value of $\mathbf{S}^{-1/2}$ is calculated exactly, it will – due to the strict localization of the support functions – not be possible to exactly orthogonalize the support functions since in general the orthonormalized support function $\tilde{\phi}^\alpha$ does not fit into the same localization region as the non-orthonormal one ϕ^α. Therefore one can make a virtue out of necessity and try to replace the exact calculation of $\mathbf{S}^{-1/2}$ by an approximation which can be calculated much faster. If the error introduced by this approximation is of the same order of magnitude as the one caused by the localization constraint, this approach should be an acceptable way to go.

Luckily the situation is such that one is dealing with support functions being only slightly non-orthonormal, thus yielding an overlap matrix which is still close to the identity. Thus there is the hope that the error introduced by approximating $\mathbf{S}^{-1/2}$ by a first order Taylor expansion is not too large. Consequently the value of $\mathbf{S}^{-1/2}$ is approximated by

$$(S^{-1/2})^{\alpha\beta} = ([I + (S - I)]^{-1/2})^{\alpha\beta} \approx ((I - \frac{1}{2}(S - I))^{\alpha\beta} = \frac{3}{2}\delta^{\alpha\beta} - \frac{1}{2}S^{\alpha\beta}. \tag{5.46}$$

A comparison of this way of approximating $\mathbf{S}^{-1/2}$ and its exact calculation is shown in Fig. 5.14, where the maximal deviation of the overlap matrix from the identity before and after the orthonormalization is shown as a function of the cutoff radius for the support functions. The test was done for a water droplet consisting of 1500 atoms, giving rise to an overlap matrix of dimension 3000×3000.

First of all it can be seen that the maximal deviation from the identity matrix before

the orthogonalization is quite independent of the cutoff radius, meaning that all orthogonalizations start from approximately the same conditions and can thus well be compared. Furthermore it is obvious that the difference from the identity matrix after the orthonormalization is more or less independent of the localization radius for the Taylor approximation, whereas the one for the exact calculation of $\mathbf{S}^{-1/2}$ decreases as the cutoff radius is increased.
This is intuitively clear, since for the exact calculation the only source of error comes from the cutting of the support functions due to the localization constraint; obviously this error tends to zero as the cutoff radius is increased. On the other hand the error introduced by the Taylor expansion does not only depend on the localization constraint, but also on how much the overlap matrix deviates from the identity matrix before $\mathbf{S}^{-1/2}$ is calculated; as mentioned, this deviation is more or less independent of the localization radius. Thus one can conclude that as long as the two curves are close together, the error stemming from the localization constraint dominates, whereas the one caused by the Taylor approximation prevails as soon as they spread apart.
As can be seen the differences between the version that calculates $\mathbf{S}^{-1/2}$ exactly and the one that approximates it using the Taylor expansion are – at least for typical cutoff radii which are around 10 bohr – not huge. In view of the enormous time saving offered by the Taylor expansion – see in this context also Fig. 5.17 –, it therefore seems to be a good strategy to use this approximation.

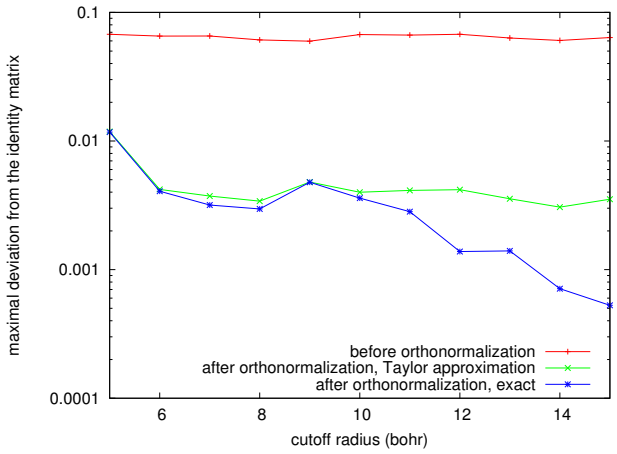


Figure 5.14: Comparison of the two orthogonalization procedures, i.e. the one which exactly calculates $\mathbf{S}^{-1/2}$ and the one which approximates it using a first order Taylor expansion, illustrated by the maximal deviation of the overlap matrix after the orthogonalization from the identity. The red curve shows the same quantity before the orthogonalization. The test was done for a water droplet consisting of 1500 atoms and a total matrix size of 3000×3000.

5.1.7.2 Submatrix method

Unfortunately the procedure described in the previous section is not applicable right after the input guess when the support functions are not yet close to being orthonormal, thus yielding an overlap matrix that is considerably distinct from the identity. Using the Taylor approximation in such a case is not advisable since it assumes only slightly non-orthonormal support functions.

However always using the exact calculation of $\mathbf{S}^{-1/2}$ in this situation would create a bottleneck causing severe problems for large systems. Therefore a procedure was developed which avoids the diagonalization of the entire matrix while still approximating $\mathbf{S}^{-1/2}$ very accurately.

A schematical overview of this so-called submatrix method is shown in Fig. 5.15. The basic idea behind it is that the value of $\mathbf{S}^{-1/2}$ is calculated independently for each column of the matrix.
Therefore one starts by first determining the "active space" for each column, meaning that one selects all matrix elements that correspond to support functions with which

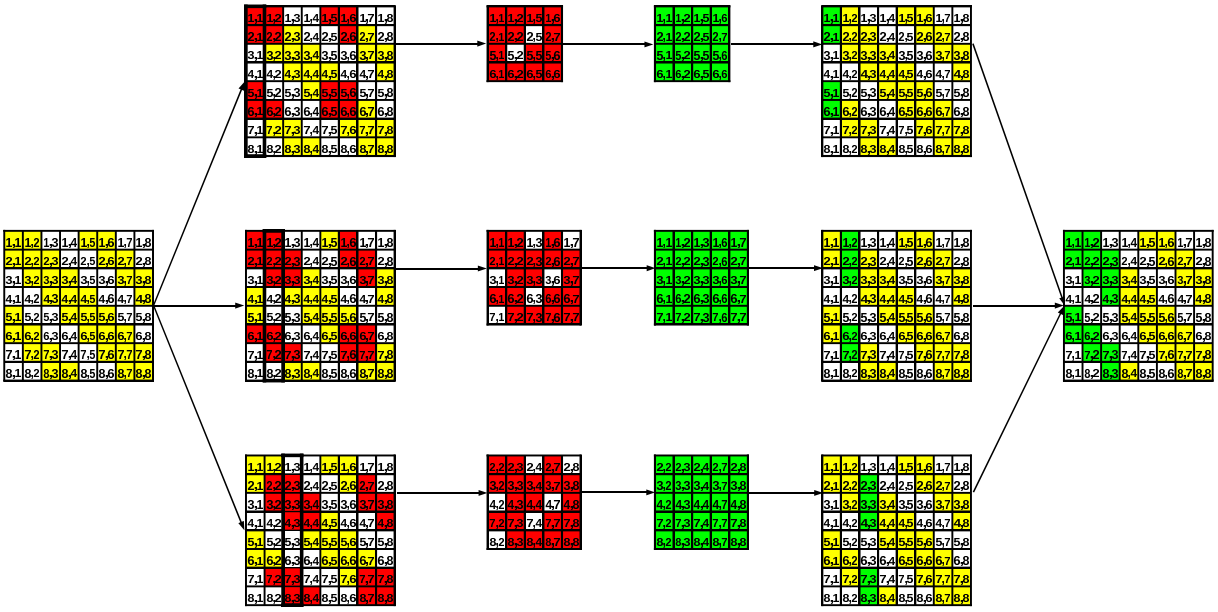


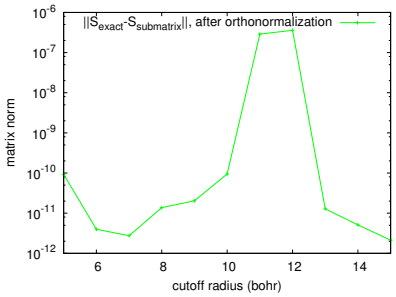
Figure 5.15: Schematical view of the submatrix method which allows to calculate $\mathbf{S}^{-1/2}$ without diagonalizing the entire matrix. The procedure is shown for the three first columns of the matrix; for the other ones the procedure is analogous.
On the very left side the overlap matrix with its sparsity pattern – i.e. only the yellow fields are non-zero – is shown. Now for each column which is processed – visualized by the bold frame – the "active space" (indicated by red) is selected, given by those matrix elements corresponding to support functions with which the support function represented by the current column overlaps. For instance, the first support function has overlaps with the support functions 1, 2, 5 and 6, and consequently the matrix elements belonging to these support functions form the active space. This active space is then cut out of the large matrix and filled into a smaller matrix $\mathbf{s}$, padding with zero the empty entries. For this smaller matrix the value of $\mathbf{s}^{-1/2}$ is calculated exactly, as indicated by the green matrices. Now the result of this smaller matrix is inserted back into the large matrix, but only filling the column that has been chosen initially. The matrix elements which are not covered by the submatrix $\mathbf{s}$ due to the sparsity of $\mathbf{S}$ are padded with zeros.

the support function represented by the current column overlaps. This active space defines a smaller submatrix $\mathbf{s}$ which is then cut out of the large matrix and for which the exact value of $\mathbf{s}^{-1/2}$ can be calculated with much less computational effort compared to the large matrix. Afterwards the result for $\mathbf{s}^{-1/2}$ is inserted back into that column of the large matrix which is processed, padding with zero those elements which were not part of the small matrix $\mathbf{s}$. If $\mathbf{S}^{-1/2}$ is directly stored in the same sparse format as $\mathbf{S}$, this padding is not required.
Since the columns can be treated independently, this method can be easily parallelized and is therefore quite efficient on large parallel architectures.

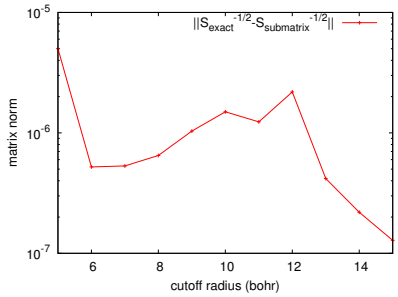
An overview of the performance of this method is shown in Fig. 5.16. The first plot, Fig. 5.16a, shows the matrix norm of the difference between the overlap matrix after the orthogonalization using the exact calculation of $\mathbf{S}^{-1/2}$ and the one using the submatrix method, i.e. it plots the value

$$\kappa = ||\mathbf{S}_{\text{exact}} - \mathbf{S}_{\text{submatrix}}|| = \sqrt{\sum_{\alpha,\beta} \left| S^{\alpha\beta}_{\text{exact}} - S^{\alpha\beta}_{\text{submatrix}} \right|^2}. \tag{5.47}$$

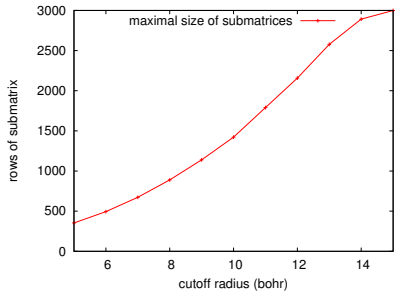
Here quite some variations with respect to the cutoff radius can be observed, but still the maximum is only of the order of 10^{-7}. This is actually an excellent value in view of the fact that the maximal error introduced by the localization constraints is – depend-



(a) The matrix norm of the difference between the overlap matrix after the orthogonalization using the exact method and the one using the submatrix approximation, i.e. Eq. (5.47). Even if there is a rather large variation with respect to the cutoff radius, the values are always very small.



(b) The matrix norm of the difference between the exact calculation of $\mathbf{S}^{-1/2}$ and its approximation using the submatrix method. The variations with respect to the cutoff radius are rather small and the matrix norm is always quite small, demonstrating the accuracy of the submatrix method.



(c) The maximal size (i.e. the maximal number of columns) of the submatrices as a function of the cutoff radius. As expected there is a strong increase. Since the total matrix has a size of 3000×3000, the curve starts to saturate towards large radii.

Figure 5.16: An overview of the performance of the submatrix method. The tests were again done for the water droplet containing 1500 atoms. Figs. 5.16a and 5.16b show the accuracy of the method, whereas Fig. 5.16c is rather a performance issue.

ing on the localization radius – of the order of 10^{-4} to 10^{-3}, as has been demonstrated in Fig. 5.14. This means that an orthogonalization using the submatrix method can be considered as exact.

An even much better way to estimate the accuracy of the submatrix method itself is to plot the value of $\left|\left|\mathbf{S}_{\text{exact}}^{-1/2} - \mathbf{S}_{\text{submatrix}}^{-1/2}\right|\right|$, i.e. the matrix norm of the difference between the two ways to calculate $\mathbf{S}^{-1/2}$. In this way the error introduced by using the submatrix method can be isolated and is not mixed up with the error introduced by the localization constraint. This plot is shown in Fig. 5.16b. As can be seen this matrix norm shows only a weak variation with respect to the cutoff radius and is always smaller than 10^{-5}, once more demonstrating the accuracy of the submatrix method.

Last but no least Fig. 5.16c shows the maximal size of the the submatrices. As expected this number exhibits a strong increase with respect to the cutoff radius. Still it is appealing that the error introduced by the submatrix method does not seem to depend heavily on the size of the submatrices.

In view of these impressing results one might wonder why it is not possible to use the submatrix method throughout the entire calculation, replacing the Taylor approximation. The reason is quite simple: The Taylor approximation method is still much faster – by orders of magnitude – than the submatrix method. This fact is illustrated in Fig. 5.17, where the time required by the three approaches to calculate $\mathbf{S}^{-1/2}$ – exact, Taylor approximation and submatrix method – are shown as a function of the cutoff radius.
Furthermore the scaling with respect to the cutoff radius is much better for the Taylor approximation method. Here the time is directly proportional to the number of

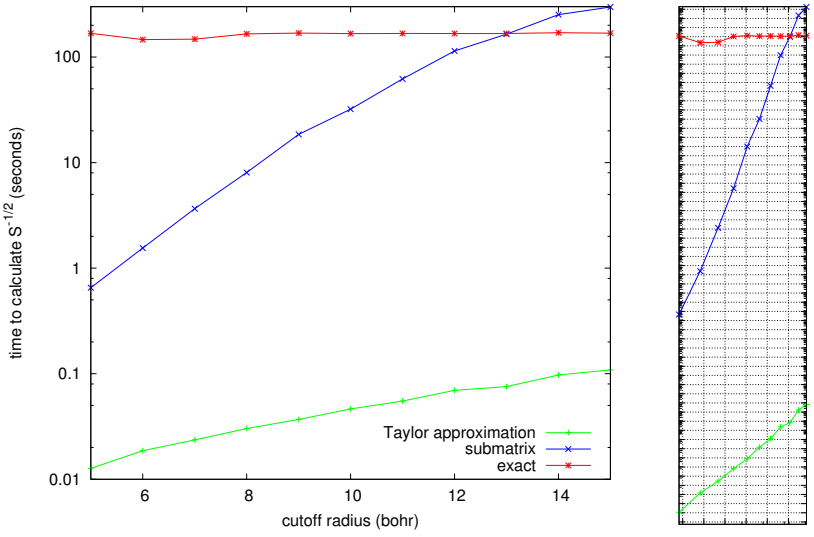


Figure 5.17: The time required for the calculation of $\mathbf{S}^{-1/2}$ by the three methods "Taylor approximation", "submatrix" and "exact", with respect to the cutoff radius. The right plot is the same on a log-log scale, demonstrating that the scaling of the submatrix method is the cube of the scaling of the Taylor approximation.

non-zero matrix elements, which is in turn related to the number of overlaps a given support function exhibits with the other ones and depends – apart from the cutoff radius – as well on the geometry. For the submatrix method, on the other hand, the most time consuming part is the diagonalization of the submatrix, which scales cubically with respect to the size of the matrix. This size is, in turn, again related to the number of overlaps a given support functions exhibits. Consequently the scaling of the submatrix method is the cube of the scaling of the Taylor approximation. This fact is demonstrated by the log-log plot in Fig. 5.17.
The exact calculation of $\mathbf{S}^{-1/2}$ requires the diagonalization of the entire overlap matrix and its time requirement is therefore independent of the cutoff radius. Thus it can happen that the submatrix method becomes slower than the exact calculation of $\mathbf{S}^{-1/2}$ for very large cutoff radii.

It is worth noting that both the Taylor approximation method and the submatrix method can rather easy be parallelized, whereas this is much more difficult for the exact method. Consequently the crossover point between the exact and the submatrix method depends on the number of MPI tasks that are used. For the current test 1500 MPI tasks were used, meaning that each MPI task had to handle two columns of the matrix.

5.1.8 Orthonormality constraint

In order to keep the support functions as orthogonal as possible in the course of the optimization procedure, it is important to incorporate an orthogonality constraint into the gradient. For its derivation one has to temporarily abandon the orthonormality and only reapply it after the gradient has been calculated. A way how this can be done if the Hamiltonian is orbital-dependent – which corresponds to the case where the support functions are optimized using a confining potential, i.e. the trace minimization mode, the mixed mode and the hybrid mode – was outlined by Goedecker and Umrigar [75] and will be shown in the following.

5.1.8.1 Derivation for orthonormal orbitals

For this derivation it is assumed that the support functions can be orthonormalized exactly; consequently the distinction between covariant and contravariant quantities is not necessary and everything is written with a lower index. The generalization to non-orthogonal support functions will be shown in Sec. 5.1.8.2.

The derivation starts with the construction of a set of orthonormal support functions

$\tilde{\phi}_\alpha$ out of the non-orthonormal set ϕ_α by means of a Löwdin orthogonalization:

$$\tilde{\phi}_\alpha = \sum_\beta (S^{-1/2})_{\alpha\beta}\phi_\beta \tag{5.48}$$

with the overlap matrix $S_{\alpha\beta} = \int \phi_\alpha(\mathbf{r})\phi_\beta(\mathbf{r})\,\mathrm{d}\mathbf{r}$. Since one wants to calculate the derivative for a set of support functions which are only slightly non-orthonormal it is justified to approximate the calculation of $\mathbf{S}^{-1/2}$ by a first order Taylor approximation and to write $\mathbf{S}^{-1/2} = [\mathbf{I} + (\mathbf{S} - \mathbf{I})]^{-1/2} \approx \mathbf{I} - \frac{1}{2}(\mathbf{S} - \mathbf{I}) = \frac{3}{2}\mathbf{I} - \frac{1}{2}\mathbf{S}$. The Löwdin orthogonalization then reads

$$\tilde{\phi}_\alpha = \sum_\beta (\frac{3}{2}\delta_{\alpha\beta} - \frac{1}{2}S_{\alpha\beta})\phi_\beta. \tag{5.49}$$

The total gradient of the target function with respect to the support function ϕ^α can now be calculated by applying the chain rule:

$$\frac{\delta\Omega}{\delta\phi_\alpha(\mathbf{r})} = \sum_\beta \int \frac{\delta\Omega}{\delta\tilde{\phi}_\beta(\mathbf{r}')}\frac{\delta\tilde{\phi}_\beta(\mathbf{r}')}{\delta\phi_\alpha(\mathbf{r})}\,\mathrm{d}\mathbf{r}'. \tag{5.50}$$

The first part is just the unconstrained gradient that depends on the specific functional form of the target function Ω,

$$g_\beta(\mathbf{r}) = \frac{1}{2}\frac{\delta\Omega}{\delta\tilde{\phi}_\beta(\mathbf{r})}. \tag{5.51}$$

For the second part one gets

$$\begin{aligned}
\frac{\delta\tilde{\phi}_\beta(\mathbf{r}')}{\delta\phi_\alpha(\mathbf{r})} &= \frac{3}{2}\delta_{\alpha\beta}\delta(\mathbf{r}-\mathbf{r}') - \frac{1}{2}S_{\alpha\beta}\delta(\mathbf{r}-\mathbf{r}') - \frac{1}{2}\sum_\gamma \phi_\gamma(\mathbf{r}')\frac{\delta}{\delta\phi_\alpha(\mathbf{r})}\int \phi_\beta(\mathbf{r}'')\phi_\gamma(\mathbf{r}'')\,\mathrm{d}\mathbf{r}'' \\
&= \frac{3}{2}\delta_{\alpha\beta}\delta(\mathbf{r}-\mathbf{r}') - \frac{1}{2}S_{\alpha\beta}\delta(\mathbf{r}-\mathbf{r}') - \frac{1}{2}\sum_\gamma \phi_\gamma(\mathbf{r}')[\delta_{\alpha\beta}\phi_\gamma(\mathbf{r}) + \delta_{\alpha\gamma}\phi_\beta(\mathbf{r})] \\
&= \frac{3}{2}\delta_{\alpha\beta}\delta(\mathbf{r}-\mathbf{r}') - \frac{1}{2}S_{\alpha\beta}\delta(\mathbf{r}-\mathbf{r}') - \frac{1}{2}\delta_{\alpha\beta}\sum_\gamma \phi_\gamma(\mathbf{r}')\phi_\gamma(\mathbf{r}) - \frac{1}{2}\phi_\alpha(\mathbf{r}')\phi_\beta(\mathbf{r}) \\
&= \delta_{\alpha\beta}\delta(\mathbf{r}-\mathbf{r}') - \frac{1}{2}\delta_{\alpha\beta}\sum_\gamma \phi_\gamma(\mathbf{r}')\phi_\gamma(\mathbf{r}) - \frac{1}{2}\phi_\alpha(\mathbf{r}')\phi_\beta(\mathbf{r}),
\end{aligned} \tag{5.52}$$

where in the last step the fact that the derivative is calculated for a set of orthonormal orbitals was used and therefore the overlap matrix is equal to the identity matrix, i.e.

$\mathbf{S} = \mathbf{I}$. Inserting the results (5.51) and (5.52) in the total gradient (5.50) one gets

$$\begin{aligned}\frac{1}{2}\frac{\delta\Omega}{\delta\phi_\alpha(\mathbf{r})} &= \sum_\beta \int g_\beta(\mathbf{r}')\delta_{\alpha\beta}\delta(\mathbf{r}-\mathbf{r}')\,\mathrm{d}\mathbf{r}' \\ &\quad - \frac{1}{2}\sum_{\beta,\gamma}\int \delta_{\alpha\beta}g_\beta(\mathbf{r}')\phi_\gamma(\mathbf{r}')\phi_\gamma(\mathbf{r})\,\mathrm{d}\mathbf{r}' \\ &\quad - \frac{1}{2}\sum_{\beta}\int g_\beta(\mathbf{r}')\phi_\alpha(\mathbf{r}')\phi_\beta(\mathbf{r})\,\mathrm{d}\mathbf{r}' \\ &= g_\alpha(\mathbf{r}) - \frac{1}{2}\sum_\gamma\left(\int g_\alpha(\mathbf{r}')\phi_\gamma(\mathbf{r}')\,\mathrm{d}\mathbf{r}'\right)\phi_\gamma(\mathbf{r}) - \frac{1}{2}\sum_\beta\left(\int g_\beta(\mathbf{r}')\phi_\alpha(\mathbf{r}')\,\mathrm{d}\mathbf{r}'\right)\phi_\beta(\mathbf{r}).\end{aligned} \tag{5.53}$$

Defining the Lagrange multiplier matrix which enforces this constraint by

$$\Lambda_{\alpha\beta} = \int g_\alpha(\mathbf{r})\phi_\beta(\mathbf{r})\,\mathrm{d}\mathbf{r} \tag{5.54}$$

the above result can be written in a more compact form as

$$\frac{1}{2}\frac{\delta\Omega}{\delta\phi_\alpha(\mathbf{r})} = g_\alpha(\mathbf{r}) - \frac{1}{2}\sum_\beta \Lambda_{\alpha\beta}\phi_\beta(\mathbf{r}) - \frac{1}{2}\sum_\beta \Lambda_{\beta\alpha}\phi_\beta(\mathbf{r}). \tag{5.55}$$

If the matrix Λ was symmetric – which would be the case if the form of the gradient did not depend on the specific value of α, i.e. without using any confinement – the two sums could be combined into one.

5.1.8.2 Generalization to non-orthonormal orbitals

The above derivation – with the final result (5.55) – was done for a set of orthonormal support functions. Now it has to be generalized to non-orthogonal ones. This can most easily done in the space of the coefficients of the underlying wavelet basis. To this end each support function is written explicitly in this basis:

$$\phi_\alpha(\mathbf{r}) = \sum_i c_{i\alpha}\chi_i(\mathbf{r}), \tag{5.56}$$

where χ_i stands for both scaling functions and wavelets. The coefficients $c_{i\alpha}$ form a matrix of dimension $n_{\text{basis}} \times n_{\text{sup.f.}}$, where n_{basis} is the total number of underlying basis functions (scaling functions and wavelets) and $n_{\text{sup.f.}}$ the total number of support functions. Denoting the quantities for the orthonormal support functions by a tilde, the Löwdin orthogonalization can – thanks to the orthonormality of the wavelet basis – be written as

$$\tilde{\mathbf{c}} = \mathbf{c}\mathbf{S}^{-1/2}, \tag{5.57}$$

where $\mathbf{S} = \mathbf{c}^T\mathbf{c}$ is the overlap matrix of dimension $n_{\text{sup.f.}} \times n_{\text{sup.f.}}$. Expressing the Hamiltonian as a matrix $\mathbf{H}$ of dimension $n_{\text{basis}} \times n_{\text{basis}}$ the orthogonality constraint of Eq. (5.55) – now as well represented by a matrix $\tilde{\mathbf{G}}$ of dimension $n_{\text{basis}} \times n_{\text{sup.f.}}$ – can be written as

$$\tilde{\mathbf{G}} = \mathbf{H}\tilde{\mathbf{c}} - \frac{1}{2}\tilde{\mathbf{c}}\tilde{\mathbf{\Lambda}} - \frac{1}{2}\tilde{\mathbf{c}}\tilde{\mathbf{\Lambda}}^T, \tag{5.58}$$

where the matrix $\tilde{\mathbf{\Lambda}}$ of dimension $n_{\text{sup.f.}} \times n_{\text{sup.f.}}$ is this time given by $\tilde{\mathbf{\Lambda}} = \tilde{\mathbf{c}}^T\tilde{\mathbf{g}}$ with $\tilde{\mathbf{g}}$ being the expansion coefficients of the unconstrained gradient in the underlying basis. Again the tilde indicates that these are the quantities for the case of orthonormal support functions.
The gradient matrix for the orthogonal case, $\tilde{\mathbf{G}}$, is related to the one for the non-orthogonal case, $\mathbf{G}$, in the same way as the coefficients, namely

$$\tilde{\mathbf{G}} = \mathbf{G}\mathbf{S}^{-1/2}. \tag{5.59}$$

This equation can now be solved for $\mathbf{G}$, yielding

$$\begin{aligned}
\mathbf{G} &= \left[\mathbf{H}\tilde{\mathbf{c}} - \frac{1}{2}\tilde{\mathbf{c}}\tilde{\mathbf{\Lambda}} - \frac{1}{2}\tilde{\mathbf{c}}\tilde{\mathbf{\Lambda}}^T\right]\mathbf{S}^{1/2} \\
&= \left[\mathbf{H}\tilde{\mathbf{c}} - \frac{1}{2}\tilde{\mathbf{c}}(\tilde{\mathbf{c}}^T\tilde{\mathbf{g}}) - \frac{1}{2}\tilde{\mathbf{c}}(\tilde{\mathbf{g}}^T\tilde{\mathbf{c}})\right]\mathbf{S}^{1/2} \\
&= \left[\mathbf{H}(\mathbf{c}\mathbf{S}^{-1/2}) - \frac{1}{2}(\mathbf{c}\mathbf{S}^{-1/2})(\mathbf{S}^{-1/2}\mathbf{c}^T\mathbf{g}\mathbf{S}^{-1/2}) - \frac{1}{2}(\mathbf{c}\mathbf{S}^{-1/2})(\mathbf{S}^{-1/2}\mathbf{g}^T\mathbf{c}\mathbf{S}^{-1/2})\right]\mathbf{S}^{1/2} \\
&= \mathbf{H}\mathbf{c} - \frac{1}{2}\mathbf{c}\mathbf{S}^{-1}\mathbf{c}^T\mathbf{g} - \frac{1}{2}\mathbf{c}\mathbf{S}^{-1}\mathbf{g}^T\mathbf{c} \\
&= \mathbf{H}\mathbf{c} - \frac{1}{2}\mathbf{c}\mathbf{S}^{-1}\mathbf{\Lambda} - \frac{1}{2}\mathbf{c}\mathbf{S}^{-1}\mathbf{\Lambda}^T,
\end{aligned} \tag{5.60}$$

where the symmetry of $\mathbf{S}^{-1/2}$ was used. This is exactly the same expression as for the orthonormal case, i.e. Eq. (5.58), except for the fact that the Lagrange multiplier matrix has to be multiplied first with the inverse of the overlap matrix.

Whereas the correction for the non-orthonormality is quite simple in principle, its implementation might be problematic since it requires to invert the overlap matrix which could become a bottleneck. However, after the discussion in Sec. 5.1.7, it seems to be plausible to approximate the inverse as well by a first order Taylor expansion,

$$\mathbf{S}^{-1} = (\mathbf{I} + (\mathbf{S} - \mathbf{I}))^{-1} \approx \mathbf{I} - (\mathbf{S} - \mathbf{I}) = 2\mathbf{I} - \mathbf{S}, \tag{5.61}$$

thereby circumventing this problem.

Since the effect of the slight non-orthonormality is not very strong, it can be specified

manually whether the inverse of the overlap matrix should be applied to the Lagrange multiplier matrix or not. Furthermore it can be chosen whether the calculation of $\mathbf{S}^{-1}$ should be done exactly or only approximately using the Taylor expansion.

5.1.9 The number of support functions

As already mentioned several times, the goal is to use only a very small number of support functions. If they are of good enough quality, adding more support functions will improve the final result only little while still increasing the computational demand considerably.

The number of support functions can be specified for each atom type in the system. Due to the nature of the input guess consisting of the atomic orbitals, it is at the moment only possible to use the numbers 1, 4, 9 and 16, which correspond to the sum of s-, p-, d- and f-orbitals, respectively. Usually it is enough to use a minimal basis set, consisting of those orbitals for which the atom exhibits a non-zero occupation.

To investigate the effect of using more support functions than just such a minimal set, a test for an alkane consisting of 152 atoms was performed. For this system the minimal basis set consists of 4 support functions for each carbon atom and 1 for each hydrogen atom, denoted by 4/1. The next step would be to add as well the p-orbitals for the hydrogen atoms – denoted by 4/4 –, and a further increase in the accuracy is expected when taking in addition the d-orbitals for the carbon atoms, denoted by 9/4.
The results of this test are shown in Fig. 5.18. Very surprisingly the system behaves in a non-variational way, i.e. the energy does not systematically decrease if the number of

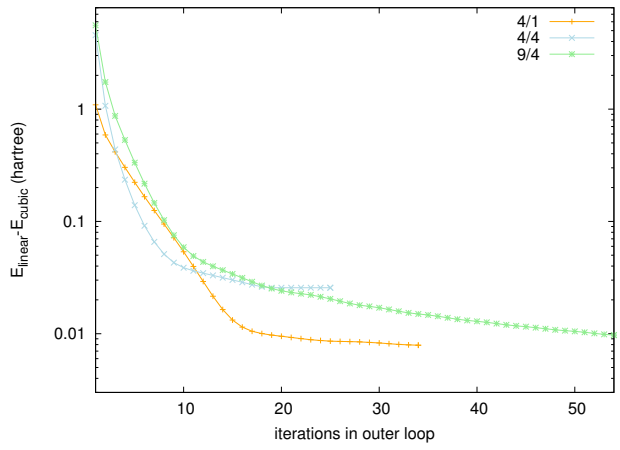


Figure 5.18: Comparison of the effect of increasing the number of support function, tested for an alkane consisting of 152 atoms. The hybrid mode and the FOE method were used for the optimization of the support functions and the density kernel, respectively; the cutoff radius was set to 9 bohr and the initial prefactor for the confinement to $3.0 \cdot 10^{-3}$ hartree/bohr^4. 4/1 means 4 support functions per carbon atom and 1 per hydrogen atom; the meaning of 4/4 and 9/4 is analogous. The energy difference between the linear and the cubic version, $E_{\text{linear}} - E_{\text{cubic}}$, is clearly non-variational.

support functions is increased. This means that the additional support functions have in some way deteriorated the quality of the original ones.

It turns out that the problem is related to the orthonormalization of the atomic orbitals which are used as the input guess for the support functions. It seems that the more support functions are contained in one localization region, the more they are spread out in the course of the orthonormalization, thus requiring to cut a lot at the boundaries. Since this cutting affects as well the original support functions, it becomes clear why adding more and more support functions can deteriorate their quality and lead to worse results.

In order to prevent this deterioration of the support functions, the orthonormalization procedure has to be modified, as will be shown in the following. In a first step the minimal basis is orthonormalized without taking into account the remaining support functions:

$$\tilde{\phi}^\alpha = \sum_{\beta \in M} (\mathbf{S}'^{-1/2})^{\alpha\beta} \phi^\beta \quad \forall \alpha \in M, \tag{5.62}$$

where M denotes the minimal set and $\mathbf{S}'$ is the overlap matrix of this subset. This generates by definition the same orthonormal set as the minimal approach, i.e. without the additional support functions. In a second step the latter ones are orthogonalized with respect to the minimal basis by means of the Gram-Schmidt procedure:

$$\tilde{\phi}^\alpha = \phi^\alpha - \sum_{\beta \in M} S^{\alpha\beta} \phi^\beta \quad \forall \alpha \notin M, \tag{5.63}$$

where $\mathbf{S}$ is this time the overlap matrix among all support functions. Finally the additional support functions are orthonormalized without modifying the other ones:

$$\tilde{\tilde{\phi}}^\alpha = \sum_{\beta \notin M} (\mathbf{S}''^{-1/2})^{\alpha\beta} \tilde{\phi}^\beta \quad \forall \alpha \notin M, \tag{5.64}$$

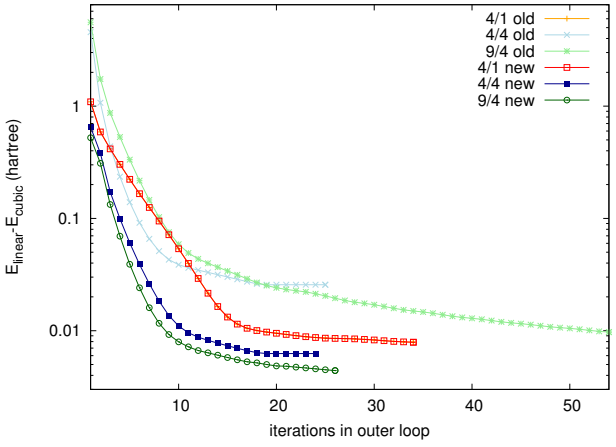


Figure 5.19: Illustration of the effect of the two different orthonormalization methods for the input guess of the support functions on the results when a larger set of support functions than just the minimal one is used. The same system and parameters were used as in Fig. 5.18. It is obvious that the results using the new method are clearly superior; furthermore the variationality is perfectly restored. In addition the lines for "4/1 old" and "4/1 new" coincide, as it should be.

where $\mathbf{S}''$ is the overlap matrix among the support functions which do not belong to the minimal set.
If one is using more than just one additional level – e.g. using s-, p- and d-orbitals instead of just s-orbitals – this procedure can be applied recursively.
In this way one gets an orthogonal set of support functions which is by construction an enhancement of the minimal basis, meaning that the variationality is restored.

This is confirmed by the results shown in Fig. 5.19, which compares this new orthonormalization with the data of Fig. 5.18.
First of all it is obvious that the results for the minimal basis are identical to the ones yielded by the old version, as it should be. Furthermore, as expected, the variationality is perfectly restored and the results for the settings 4/4 and 9/4 are considerably better compared to the old orthonormalization method.

These results are quite astonishing taking into account that the only difference between the two versions is the very first orthonormalization. This demonstrates the huge impact of the input guess on the final results.

However it is questionable whether it is worth – at least for this system – to take more support functions than the minimal set. Tab. 5.1 shows the energy difference per atom between the linear and the cubic version and the average time required for one iteration of the outer loop. As can be seen, the energy difference is already very small ($1.41\,{}^{\mathrm{meV}}/_{\mathrm{atom}}$) for the minimal basis set. Increasing the number of support functions further gives only little improvement of the accuracy, but increases the run time considerably.

	$E_{\mathrm{linear}} - E_{\mathrm{cubic}}$ (${}^{\mathrm{meV}}/_{\mathrm{atom}}$)	time (seconds)
4/1	1.41	6.7
4/4	1.11	15.5
9/4	0.79	26.8

Table 5.1: The energy difference between a run using the linear version and one using the cubic version, together with the average time needed for one iteration in the outer loop. The energy difference, this time shown in ${}^{\mathrm{meV}}/_{\mathrm{atom}}$, is already very small for the minimal basis set. The test system was the same that was used for Fig. 5.19.

5.2 Kernel optimization

Unlike the optimization of the support functions, where the various modes may lead to different results, the situation for the optimization of the density kernel is contrary. All methods to optimize the density kernel should finally yield the same result even

though they may be quite different in spirit.

The optimization of the density kernel is always done using a fixed set of support functions. Still it is an iterative procedure – even if some of the methods presented in this section, for instance the direct diagonalization, seem to be non-iterative at first sight –, since the density kernel is optimized in a self-consistent way, meaning that after each update of the density kernel a new charge density and a new potential are determined, which are then the input for the next optimization step. This is in contrast to the optimization of the support functions which is done in a non-self-consistent way.

Still it is in general better not to optimize the density kernel to a fully self-consistent solution, but rather to stop after a few iterations and then to further optimize the support functions with the new potential. Otherwise one may just have found a self-consistent solution in a basis which is not yet of high quality, which will most likely simply increase the time to solution without giving a better final result.
A general overview can also be gained from the various flowcharts in Sec. 5.1.

The most straightforward approach would be to directly calculate the gradient of the energy with respect to the elements of the density kernel, i.e. $\frac{\partial E}{\partial K^{\alpha\beta}}$, and then to update the density kernel with this gradient. However this procedure would violate the idempotency of the density kernel; thus each such update would require an additional operation that restores this property.

For the linear scaling version of BigDFT several approaches for the optimization of the density kernel were implemented which all have in common that they automatically yield an idempotent density kernel. These various methods will first be explained in detail, followed by a comparison of their performances.

5.2.1 Direct diagonalization

The most straightforward way to optimize the density kernel is a direct diagonalization of the Hamiltonian matrix in the basis of the support functions, i.e. $H^{\alpha\beta} = \langle \phi^\alpha | \mathcal{H} | \phi^\beta \rangle$. By solving the generalized eigenvalue problem of Eq. (3.42),

$$\mathbf{H}\mathbf{c}_i = \epsilon_i \mathbf{S}\mathbf{c}_i, \tag{5.65}$$

one gets the eigenvectors $\mathbf{c}_i$ which are – according to Eq. (3.36) – the expansion coefficients of the Kohn-Sham orbitals in terms of the support functions, i.e. $|\psi_i\rangle = \sum_\alpha c_{i\alpha} |\phi^\alpha\rangle$. Thus the density kernel can – in agreement with Eq. (3.38) – be constructed out of them according to

$$K_{\alpha\beta} = \sum_i c_{i\alpha} c_{i\beta}. \tag{5.66}$$

This new density kernel automatically satisfies the requirement of idempotency. From the normalization of the coefficients $\mathbf{c}_i^T \mathbf{S} \mathbf{c}_j = \delta_{ij}$ – which is automatically provided by the eigensolver – it follows that

$$\sum_{\mu,\nu} K_{\alpha\mu} S^{\mu\nu} K_{\nu\beta} = \sum_{\mu,\nu} \sum_{i,j} c_{i\alpha} c_{i\mu} S^{\mu\nu} c_{j\nu} c_{j\beta} = \sum_{i,j} c_{i\alpha} \delta_{ij} c_{j\beta} = \sum_i c_{i\alpha} c_{i\beta} = K_{\alpha\beta}, \tag{5.67}$$

or written in more compact form

$$\mathbf{KSK} = \mathbf{K}, \tag{5.68}$$

which is exactly the idempotency condition (3.33).

This normalization of the coefficients also ensures the orthonormality of the fictitious Kohn-Sham orbitals, since

$$\langle \psi_i | \psi_j \rangle = \sum_{\alpha,\beta} c_{i\alpha} c_{j\beta} \langle \phi^\alpha | \phi^\beta \rangle = \sum_{\alpha,\beta} c_{i\alpha} S^{\alpha\beta} c_{j\beta} = \delta_{ij}. \tag{5.69}$$

The new kernel as calculated by (5.66) then permits the determination of the new charge density according to Eq. (3.28). However this new charge density cannot be used directly; instead a mixing with the old density has to be performed first. In the simplest case of linear mixing, the final charge density $\tilde{\rho}$ which will be used in the subsequent step is given by

$$\tilde{\rho} = \alpha \rho^{\text{new}} + (1 - \alpha) \rho^{\text{old}}, \tag{5.70}$$

where α is the mixing parameter and lies between 0 and 1. There exist also more elaborate mixing prescriptions than this simple one – e.g. the Pulay mixing [72], which is as well implemented – that typically give a faster convergence. The choice of the mixing scheme can be specified manually by the user.

Instead of mixing the charge density it is also possible to mix the potential which is calculated out of it. Apart from this difference the procedure is exactly the same. However, as will be shown later in Sec. 5.2.4, it turned out that mixing the charge density gives usually a faster convergence.

Even if this straightforward approach is very fast for small systems, it becomes prohibitive for larger ones due to the cubic scaling of the diagonalization.
An improvement of the scaling could be achieved by exploiting the sparsity of the matrix. Unfortunately most of the packages that can diagonalize large sparse matrices (e.g. Anasazi [76] and SLEPc [77]) are mainly designed to extract only a few eigenvalues and eigenvector, whereas the approach of the direct diagonalization requires a considerably larger number of eigenvectors. Furthermore it is rather difficult to efficiently parallelize this operation, meaning that the code will exhibit a bad performance

on highly parallel architectures.
As a consequence this method remains limited to systems where the matrix dimensions do not exceed a few thousand.

5.2.2 Direct minimization

Another possibility is to directly optimize the expansion coefficients $\mathbf{c}_i$ which appear in the representation of the Kohn-Sham orbitals in the basis of the support functions according to Eq. (3.36).
To this end one starts with the gradient of the Kohn-Sham orbitals, which can be derived by calculating the derivative of the band-structure energy $E_{BS} = \sum_i \langle\psi_i|\mathcal{H}|\psi_i\rangle$ and applying the orthogonality constraint:

$$|g_i\rangle = \mathcal{H}\,|\psi_i\rangle - \sum_j \Lambda_{ij}\,|\psi_j\rangle\,, \tag{5.71}$$

where the Lagrange multiplier matrix $\Lambda_{ij} = \langle\psi_i|\mathcal{H}|\psi_j\rangle$ enforces the orthonormality constraint. This is the same prescription as Eq. (5.55) – the factor $\frac{1}{2}$ in front of $|g_i\rangle$ has been omitted for simplicity since it just corresponds to a scaling of the gradient –, just with the difference that the Lagrange multiplier matrix is this time symmetric due to the independence of the Hamiltonian on the orbitals and as a consequence the two terms of the right hand side of (5.55) can be combined into one single expression.

The goal is to write this gradient in terms of the support in the same way the orbitals are represented in this basis, meaning that one has to determine the coefficients $\mathbf{d}_i$ of the expansion

$$|g_i\rangle = \sum_\alpha d_{i\alpha}\,|\phi^\alpha\rangle\,. \tag{5.72}$$

The first step is to insert the expansion of the Kohn-Sham orbitals, i.e. $|\psi_i\rangle = \sum_\alpha c_{i\alpha}\,|\phi^\alpha\rangle$, into the formula for the Lagrange multiplier matrix Λ:

$$\Lambda_{ij} = \sum_{\alpha,\beta} c_{i\alpha}c_{j\beta}\,\langle\phi^\alpha|\mathcal{H}|\phi^\beta\rangle\,. \tag{5.73}$$

Since both the support functions ϕ^α and the coefficients $c_{i\alpha}$ are available, this matrix can be evaluated straightforwardly. Inserting the same expansion into the expression for the gradient of (5.71) leads to

$$|g_i\rangle = \sum_\alpha c_{i\alpha}\mathcal{H}\,|\phi^\alpha\rangle - \sum_j\sum_\alpha \Lambda_{ij}c_{j\alpha}\,|\phi^\alpha\rangle\,. \tag{5.74}$$

This expression has to be equal to the representation of the gradient according to (5.72), i.e. one gets the relation

$$\sum_{\alpha} d_{i\alpha} \left|\phi^{\alpha}\right\rangle = \sum_{\alpha} c_{i\alpha} \mathcal{H} \left|\phi^{\alpha}\right\rangle - \sum_{j} \sum_{\alpha} \Lambda_{ij} c_{j\alpha} \left|\phi^{\alpha}\right\rangle . \tag{5.75}$$

Multiplying from left with $\langle\phi^{\beta}|$ and using the notations $S^{\alpha\beta} = \langle\phi^{\alpha}|\phi^{\beta}\rangle$ and $H^{\alpha\beta} = \langle\phi^{\alpha}|\mathcal{H}|\phi^{\beta}\rangle$ yields

$$\sum_{\alpha} S^{\beta\alpha} d_{i\alpha} = \sum_{\alpha} H^{\beta\alpha} c_{i\alpha} - \sum_{j} \sum_{\alpha} S^{\beta\alpha} c_{j\alpha} \Lambda_{ij}. \tag{5.76}$$

The sums over j and α on the right hand side of (5.76) can be evaluated independently of the rest of the equation. Thus one can define the vector $\mathbf{b}_i$ by

$$b_i^{\beta} = \sum_{\alpha} H^{\beta\alpha} c_{i\alpha} - \sum_{j} \sum_{\alpha} S^{\beta\alpha} c_{j\alpha} \Lambda_{ij} \tag{5.77}$$

which then leads to the expression

$$\sum_{\alpha} S^{\beta\alpha} d_{i\alpha} = b_i^{\beta}. \tag{5.78}$$

Thus the final result is that the expansion coefficients $\mathbf{d}_i$ for the gradient are given by the solution of the linear system of equations

$$\mathbf{S}\mathbf{d}_i = \mathbf{b}_i. \tag{5.79}$$

After solving this equation the coefficients $\mathbf{c}_i$ can be optimized with any optimization procedure, giving in this way an improved representation of the Kohn-Sham orbitals in terms of the support functions.

What remains is the orthonormalization of the coefficients, since the orthonormality constraint in Eq. (5.71) preserves this property only to first order. Therefore it has to be ensured explicitly after each optimization step that

$$\mathbf{c}_i^T \mathbf{S} \mathbf{c}_j = \delta_{ij}. \tag{5.80}$$

This is accomplished using the Löwdin method, which reads in this case

$$\tilde{\mathbf{c}}_i = \sum_{j} \left[\mathbf{c}^T \mathbf{S} \mathbf{c}\right]_{ij}^{-1/2} \mathbf{c}_j, \tag{5.81}$$

where $\tilde{\mathbf{c}}_i$ are the coefficients after the orthonormalization and $\mathbf{c}_i$ the ones before.

Using these normalized coefficients the density kernel can then be evaluated in exactly the same way as for the direct diagonalization approach. Thanks to the normalization

of the coefficients it again automatically fulfills the requirement of idempotency.

From the viewpoint of the scaling with respect to the size of the system, this approach has a priori again a cubic scaling due to the linear algebra contained in it. Eq. (5.79) requires to invert the overlap matrix in order to solve the linear system of equation, and the orthonormalization necessitates the calculation of $\left[\mathbf{c}^T\mathbf{S}\mathbf{c}\right]_{ij}^{-1/2}$; both operations will scale cubically with respect to the size of the matrices.
However there are a few differences compared to the direct diagonalization approach described in Sec. 5.2.1. First of all solving a linear system of equations is typically faster than diagonalizing a matrix, i.e. the coefficients can be obtained in less time. Furthermore the dimension of the matrix which has to be diagonalized for the Löwdin procedure is equal to the number of Kohn-Sham orbitals, whereas the Hamiltonian matrix which has to be diagonalized in the direct diagonalization approach has the dimension of the number of support functions.

A clear improvement of the scaling could be reached by exploiting the sparsity properties of the matrices. Furthermore it could be possible to use the fact that both $\mathbf{S}$ and $\mathbf{c}^T\mathbf{S}\mathbf{c}$ are not very distinct from the identity matrix and the calculation of $\mathbf{S}^{-1}$ and $[\mathbf{c}^T\mathbf{S}\mathbf{c}]^{-1/2}$ might consequently be approximated in some form, for instance again by using a Taylor expansion.

5.2.3 Fermi Operator Expansion

Unlike the direct diagonalization and the direct minimization approach which both first determine the expansion coefficients $\mathbf{c}$ of the Kohn-Sham orbitals and then calculate the density kernel $\mathbf{K}$ out of them, the Fermi Operator Expansion (FOE) [47, 48] directly calculates the density kernel in the basis of the support functions.
There exist several different flavors of such an expansion; the one used in the linear scaling version of BigDFT is the so-called Chebyshev Fermi Operator Expansion.

5.2.3.1 Chebyshev expansion

The basic idea of the FOE method is to express the density matrix as a function of the Hamiltonian, i.e. $F = f(\mathcal{H})$. In terms of the support functions, this would then correspond to an expression of the density kernel in terms of the Hamiltonian matrix, i.e. $\mathbf{K} = f(\mathbf{H})$.
One such expression which is particularly simple is a polynomial expansion of order

n_{pl} in the Hamiltonian matrix:

$$\mathbf{K} \approx \mathbf{p}(\mathbf{H}) = \sum_{i=0}^{n_{pl}} c_i \mathbf{H}^i. \tag{5.82}$$

Unfortunately polynomials of high degree can become numerically unstable. However this problem can be circumvented by using a Chebyshev polynomial representation [78]:

$$\mathbf{p}(\mathbf{H}) = \frac{c_0}{2}\mathbf{I} + \sum_{i=1}^{n_{pl}} c_i \mathbf{T}^i(\mathbf{H}), \tag{5.83}$$

where $\mathbf{I}$ is the identity matrix, which was written as $\mathbf{H}^0$ in Eq. (5.82), and $\mathbf{T}^i(\mathbf{H})$ the Chebyshev matrix polynomials of degree i. These polynomials are only defined in the interval $[-1, 1]$, which requires that the Hamiltonian has to be scaled and shifted such that its eigenvalue spectrum lies within this range. If ϵ_{min} and ϵ_{max} are the smallest and largest eigenvalue, respectively, which would result from diagonalizing the Hamiltonian matrix according to $\mathbf{H}\mathbf{c}_i = \epsilon_i \mathbf{S}\mathbf{c}_i$, then the scaled Hamiltonian $\tilde{\mathbf{H}}$ has to be built according to

$$\tilde{\mathbf{H}} = \sigma(\mathbf{H} - \tau\mathbf{S}), \qquad \text{with} \quad \sigma = \frac{2}{\epsilon_{max} - \epsilon_{min}}, \quad \tau = \frac{\epsilon_{min} + \epsilon_{max}}{2}, \tag{5.84}$$

where $\mathbf{S}$ is again the overlap matrix.
A way to determine the lowest and highest eigenvalue without diagonalizing the entire matrix will be shown in Sec. 5.2.3.3.

The Chebyshev polynomials appearing in (5.83) can be calculated from the following recursion relation:

$$\begin{aligned} \mathbf{T}^0(\tilde{\mathbf{H}}) &= \mathbf{I}, \\ \mathbf{T}^1(\tilde{\mathbf{H}}) &= \tilde{\mathbf{H}}, \\ \mathbf{T}^{j+1}(\tilde{\mathbf{H}}) &= 2\tilde{\mathbf{H}}\mathbf{T}^j(\tilde{\mathbf{H}}) - \mathbf{T}^{j-1}(\tilde{\mathbf{H}}). \end{aligned} \tag{5.85}$$

What remains is to calculate the expansion coefficients c_i. To this end one has to recall that the density matrix is a projection operator onto the occupied subspace of the Kohn-Sham orbitals:

$$\langle \psi_i | F | \psi_j \rangle = f(\epsilon_j)\delta_{ij}, \tag{5.86}$$

where the function $f(\epsilon_j)$ is the Fermi distribution describing the occupation of the orbital $|\psi_j\rangle$ and is given by

$$f(\epsilon) = \frac{1}{1 + e^{(\epsilon-\mu)/k_B T}}, \tag{5.87}$$

where μ is the chemical potential, k_B Boltzmann's constant and T the temperature. For systems with a finite band gap the temperature is usually set to zero, in which case

the chemical potential corresponds to the Fermi energy and the Fermi distribution becomes a step function with its values being either 1 or 0. The Fermi energy has to be adjusted such that the sum of the occupation numbers – which corresponds to the trace of the density kernel – is equal to the number of electrons in the system. A way to accomplish this is shown in Sec. 5.2.3.4.

Evaluating the polynomial $p(\mathcal{H})$ in the same eigenfunction representation as the density matrix in (5.86) gives

$$\langle\psi_i|p(\mathcal{H})|\psi_j\rangle = p(\epsilon_j)\delta_{ij}, \tag{5.88}$$

where

$$p(\epsilon) = \frac{c_0}{2} + \sum_{i=1}^{n_{pl}} c_i T^i(\epsilon). \tag{5.89}$$

By comparing Eqs. (5.86) and (5.88) it becomes clear that the polynomial expansion $p(\epsilon)$ has to approximate the Fermi distribution $f(\epsilon)$ in the interval $[-1,1]$. Thus the coefficients c_i are simply the expansion coefficients of the Fermi distribution with respect to the Chebyshev polynomials in the interval $[-1,1]$.

Once the expansion coefficients c_i are determined, which is negligible from the viewpoint of the time consumption, the expansion of the density kernel according to Eq. (5.83) can be carried out using only matrix-vector multiplications. If the lth column of the Chebyshev matrix $\mathbf{T}$ is denoted by $\mathbf{t}_l$, then these vectors fulfill – according to Eq. (5.85) – the recursion relation

$$\begin{aligned} \mathbf{t}_l^0 &= \mathbf{e}_l, \\ \mathbf{t}_l^1 &= \tilde{\mathbf{H}}\mathbf{e}_l, \\ \mathbf{t}_l^{j+1} &= 2\tilde{\mathbf{H}}\mathbf{t}_l^j - \mathbf{t}_l^{j-1}, \end{aligned} \tag{5.90}$$

where $\mathbf{e}_l$ is the lth column of the identity matrix. The lth column of the density kernel, denoted by $\mathbf{k}_l$, is then given by the linear combination of all the columns $\mathbf{t}_l$ according to Eq. (5.83), i.e.

$$\mathbf{k}_l = \frac{c_0}{2}\mathbf{t}_l^0 + \sum_{i=1}^{n_{pl}} c_i \mathbf{t}_l^i. \tag{5.91}$$

This demonstrates that the density kernel can be constructed using only matrix vector multiplications.

Due to the fact that the Hamiltonian was scaled and shifted such that its eigenvalues lie in the interval $[-1,1]$, the band-structure energy is not simply given by $\mathrm{tr}(\mathbf{K}\tilde{\mathbf{H}})$, but the shifting and scaling operations have to be undone. Thus the correct value is given by

$$E_{BS} = \frac{\mathrm{tr}(\mathbf{K}\tilde{\mathbf{H}})}{\sigma} + \tau\,\mathrm{tr}(\mathbf{K}\mathbf{S}), \tag{5.92}$$

where σ and τ are defined in Eq (5.84)

However the procedure presented so far will – even if the matrix vector multiplications are relatively cheap operations – lead to a cubic scaling. Exploiting the sparsity of the matrix $\tilde{\mathbf{H}}$ will reduce the scaling, but it will still remain quadratic since both the number of columns of the Chebyshev matrices and their length is proportional to the size of the system.
Thus true linear scaling can only be achieved by introducing a localization region for each column and setting all elements to zero if they lie outside of this region. As will be shown later in Sec. 6.2.1.2 the final result is not that sensitive with respect to the choice of this cutoff radius and saturates quite rapidly.

In practice it turns out that it is more advantageous to replace the Fermi distribution by the function

$$f(\epsilon) = \frac{1}{2}\left[1 - \mathrm{erf}\left(\frac{\epsilon - \mu}{\Delta\epsilon}\right)\right], \tag{5.93}$$

since it approaches the limits 0 and 1 faster as one goes away from the chemical potential.
It has to be stressed that even for calculations performed at zero temperature, where the Fermi distribution is a step function, it is important to use a function corresponding to finite temperature. Otherwise it would be hard to represent the Fermi distribution as a polynomial due to the introduction of Gibbs oscillations at the Fermi energy that spoil the Chebyshev fit.
For the function given by Eq. (5.93) this means that $\Delta\epsilon$ should not become too small; typically it is a fraction of the band gap [48]. Larger values give lower accuracy, whereas smaller values give higher accuracy. However, as mentioned, it has to be ensured that the value does not become too small in order to maintain the good quality of the Chebyshev fit.

The last thing to show is that the density kernel calculated in this way again automatically fulfills the requirement of idempotency. To demonstrate this one can use the fact that the kernel calculated by the Fermi Operator Expansion is by construction the density matrix in the basis of the support functions, i.e. $K_{\alpha\beta} = \langle\phi_\alpha|F|\phi_\beta\rangle$. Using the completeness relation $\sum_\mu |\phi^\mu\rangle\langle\phi_\mu| = \sum_\mu |\phi_\mu\rangle\langle\phi^\mu| = 1$ and the idempotency property of the density matrix gives

$$\begin{aligned}\sum_{\mu,\nu} K_{\alpha\mu} S^{\mu\nu} K_{\nu\beta} &= \sum_{\mu,\nu} \left\langle \phi_\alpha \,\middle|\, F \,\middle|\, \phi_\mu \right\rangle \left\langle \phi^\mu \,\middle|\, \phi^\nu \right\rangle \left\langle \phi_\nu \,\middle|\, F \,\middle|\, \phi_\beta \right\rangle \\ &= \left\langle \phi_\alpha \,\middle|\, FF \,\middle|\, \phi_\beta \right\rangle \\ &= \left\langle \phi_\alpha \,\middle|\, F \,\middle|\, \phi_\beta \right\rangle \\ &= K^{\alpha\beta},\end{aligned} \tag{5.94}$$

which shows that the density kernel indeed fulfills the idempotency requirement.

5.2.3.2 Generalization to non-orthonormal support functions

If the set of support functions in which the density matrix is represented is non-orthonormal, almost all the central equations remain identical if the Hamiltonian matrix $\tilde{\mathbf{H}}$ is replaced with a modified matrix $\tilde{\mathbf{H}}'$ which is – as follows from Eq. (3.25) – given by

$$\tilde{\mathbf{H}}' = \mathbf{S}^{-1/2}\tilde{\mathbf{H}}\mathbf{S}^{-1/2}, \tag{5.95}$$

where $S^{\alpha\beta} = \langle\phi^\alpha|\phi^\beta\rangle$ is, as usual, the overlap matrix among the support functions. The only equation which is modified by the introduction of non-orthonormal orbitals is the expression for the total number of electrons. Whereas this quantity is simply given by the trace of the density kernel in the orthonormal case, the expression has now to be replaced by

$$n = \mathrm{tr}(\mathbf{KS}), \tag{5.96}$$

as follows from Eq. (3.26).

However exactly evaluating $\mathbf{S}^{-1/2}$ would again require a diagonalization of the overlap matrix, thus spoiling the linear scaling that has been obtained by exploiting the sparsity properties and introducing the localization regions for the vectors of the Chebyshev matrices. Therefore its calculation is again approximated by a first order Taylor expansion, which can be evaluated in very little time. Since the overlap matrix is very close to the identity thanks to the quasi-orthonormality which is imposed on the support functions, the error introduced in this way should not be too large.

5.2.3.3 Guessing lower and upper bounds for the eigenvalue spectrum

As has been mentioned it is necessary to know the lowest and highest eigenvalue of the Hamiltonian matrix, ϵ_{min} and ϵ_{max}, in order to be able to shift its spectrum into the interval $[-1, 1]$. However determining these two values by a diagonalization would be wasteful. Since one does not have to determine the exact values of ϵ_{min} and ϵ_{max}, but only an lower and upper bound, ϵ_{low} and ϵ_{up}, such that $\epsilon_{low} \leq \epsilon_{min}$ and $\epsilon_{up} \geq \epsilon_{max}$, a faster approach can be used.

To this end one first guesses reasonable values for ϵ_{low} and ϵ_{up} and scales and shifts the Hamiltonian according to Eq. (5.84). With this Hamiltonian two "penalty-kernels" according to Eq. (5.83) can be calculated; the term penalty-kernel is used since the expansion coefficients are this time not a Chebyshev fit to the Fermi distribution, but rather a fit to a penalty function. For the upper penalty-kernel – meaning that it is used to determine the upper bound of the eigenvalue spectrum –, this penalty function is

zero throughout the entire spectrum, but starts to blow up as soon as it comes in the neighborhood of ϵ_{up}. Analogously the lower penalty-function is very large for values below and around the value of ϵ_{low}, but rapidly decays to zero for larger values. For simplicity two exponential functions were used, which exhibit these desired properties.
A plot of two such penalty functions is shown in Fig. 5.20. The exponential decay and increase, respectively, around the lower and upper bounds are clearly visible. The flat part is due to the fact that the value of the penalty functions is set equal to noise level of the Chebyshev fit as soon as it falls below this value.

Consequently the upper penalty-kernel can be seen as an operator that assigns a non-zero occupation to all eigenstates that lie above ϵ_{up} and a zero occupation to all states that lie below it, and in the same way the lower penalty-kernel assigns non-zero occupation numbers to states that lie below ϵ_{low} and zero occupations to states above it.
Since the sum of the occupation numbers is given by the trace of the kernel, it can now easily be determined whether the bounds ϵ_{low} and ϵ_{up} cover the entire eigenvalue spectrum. If the trace of the lower penalty-kernel is zero, this means that there are no states with eigenvalues below ϵ_{low} and that consequently ϵ_{low} is smaller than ϵ_{min}; thus this bound is fine in such a case. On the other hand, if the trace is different from zero, this means that there is at least one eigenvalue below ϵ_{low}; in such a situation, the value of ϵ_{low} has to be decreased and a new penalty-kernel is calculated.
The determination of the upper bound is of course done in a completely analogous way.

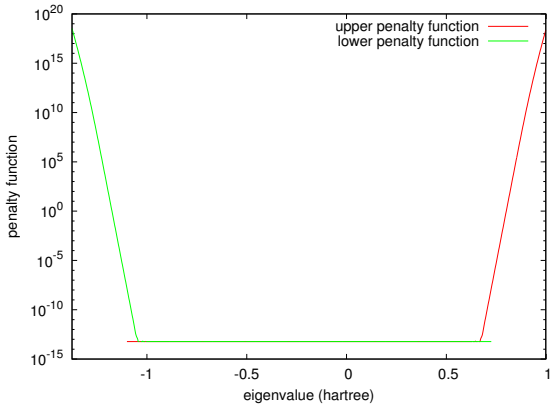


Figure 5.20: Plot of the two penalty functions used to estimate the bounds of the eigenvalue spectrum. The upper penalty function is zero throughout the entire interval, but blows up in the neighborhood of the upper bound. The lower penalty function exhibits the analogous behavior around the lower bound. The lower and upper bound for this test were set to -1.1 and 0.8, respectively. The flat part corresponds to the noise level of the Chebyshev fit, which is set to be the lower bound for the functions.

5.2.3.4 Determining the Fermi energy

The sum of the eigenvalues of the density matrix – which is also equal to its trace –, gives the sum of all occupation numbers of the Kohn-Sham orbitals. Of course this value must be equal to the number of electrons in the system which is denoted by

n. Consequently the density kernel has to be calculated several times; if the trace is smaller than the number of electrons then the Fermi energy was too low, if it is larger than the number of electrons then the Fermi energy was too high.

The condition that the trace of the density kernel equals the number of electrons in the system corresponds to determining the root of the function

$$g(\mu) = \mathrm{tr}[\mathbf{K}(\mu)\mathbf{S}] - n, \tag{5.97}$$

which is a monotonically increasing function that ranges from $-n$ to $n_{\text{sup.f.}} - n$, where $n_{\text{sup.f.}}$ is the total number of support functions. The dependence of the density kernel on the Fermi energy μ has this time been explicitly noted as $\mathbf{K}(\mu)$.

In the beginning, when the guess for the Fermi energy might be quite far away from the correct value, this search for the root of Eq. (5.97) can be accomplished using the bisection method, which is a stable, but rather slow approach. Given two values μ_1 and μ_2 for which the function $g(\mu)$ is negative and positive, respectively, a new guess for the Fermi energy is calculated according to

$$\mu_3^{bs} = \frac{\mu_1 + \mu_2}{2}. \tag{5.98}$$

It turned out that the convergence can be accelerated by taking the average of the root proposed by the bisection method and the one proposed by the secant method which is given by

$$\mu_3^{sm} = \mu_2 - g(\mu_2)\frac{\mu_2 - \mu_1}{g(\mu_2) - g(\mu_1)}, \tag{5.99}$$

meaning that the final solution is given by

$$\mu_3^{bs+sm} = \frac{\mu_3^{bs} + \mu_3^{sm}}{2}. \tag{5.100}$$

A further acceleration – however at the cost of some stability – can be reached by using a cubic interpolation, which is possible as soon as one has four pairs $(\mu_i, g(\mu_i))$. To this end one first determines the cubic polynomial

$$p(\mu) = a\mu^3 + b\mu^2 + c\mu + d \tag{5.101}$$

that goes through the points $\{(\mu_1, g(\mu_1)), (\mu_2, g(\mu_2)), (\mu_3, g(\mu_3)), (\mu_4, g(\mu_4))\}$. The coefficients a, b, c, d are given by the solution of the following linear system of equations:

$$\begin{pmatrix} \mu_1^3 & \mu_1^2 & \mu_1 & 1 \\ \mu_2^3 & \mu_2^2 & \mu_2 & 1 \\ \mu_3^3 & \mu_3^2 & \mu_3 & 1 \\ \mu_4^3 & \mu_4^2 & \mu_4 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} g(\mu_1) \\ g(\mu_2) \\ g(\mu_3) \\ g(\mu_4) \end{pmatrix}. \tag{5.102}$$

The three (possibly complex) roots of the resulting interpolating polynomial can be determined analytically and are given by

$$\begin{aligned}
\mu_{(1)} &= -\frac{b}{3a} - \frac{R}{3a} - \frac{b^2 - 3ac}{3aR}, \\
\mu_{(2)} &= -\frac{b}{3a} + \frac{R(1+i\sqrt{3})}{6a} - \frac{(1-i\sqrt{3})(b^2-3ac)}{6aR}, \\
\mu_{(3)} &= -\frac{b}{3a} + \frac{R(1-i\sqrt{3})}{6a} - \frac{(1+i\sqrt{3})(b^2-3ac)}{6aR},
\end{aligned} \tag{5.103}$$

with

$$\begin{aligned}
R &= \sqrt[3]{\frac{1}{2}(Q + 2b^3 - 9abc + 27a^2d)}, \\
Q &= \sqrt{(2b^3 - 9abc + 27a^2d)^2 - 4(b^2 - 3ac)^3}.
\end{aligned} \tag{5.104}$$

Out of these three solutions the one which is real and closest to the old Fermi energy is selected.

As already mentioned the interpolation method comes at the cost of a decreased stability. First of all the matrix in Eq. (5.102) can become virtually singular if the values μ_i are all close together, making the solution of this equation numerically unstable.
In addition it might happen that the function $g(\mu)$ is not monotonically increasing on a very small scale due to the nature of the polynomial fit that is used to represent the Fermi function or the one of Eq. (5.93), respectively. If the values used for the interpolation happen to lie in such a region it is not sensible any more to use the cubic interpolation and consequently better to go back to the combination of bisection and secant approach.

5.2.4 Comparison of the different methods

The various approaches that were presented in order to optimize the density kernel can be compared from two different perspectives, namely the accuracy and the speed. As will be shown in the following sections, the FOE method is slightly less accurate than the other methods, however only by an amount which is easily acceptable. On the other hand, it is by far the fastest method, in particular for very large systems.

5.2.4.1 Accuracy of the kernel methods

In principle all approaches should lead to the same final result since they all aim at the same goal, namely the calculation of the density kernel in the basis of the support

functions. However there are a few subtleties.

The most fundamental difference is that the diagonalization methods and the FOE method determine the exact density kernel for the current potential, whereas the direct minimization approach only improves it and calculates a new potential before the minimization has been completed. However, this should not affect the final result when self-consistency has been reached.

Furthermore only the diagonalization methods and the direct minimization take correctly into account the slight non-orthogonality of the support functions by solving the generalized eigenvalue problem or the linear system of equations, respectively, without adopting any approximations with respect to the overlap matrix. Also the orthogonalization of the expansion coefficients which has to be carried out for the direct minimization is done exactly.

The Fermi Operator Expansion method, on the other hand, approximates the value of $\mathbf{S}^{-1/2}$ by a first order Taylor expansion. For this reason some small error is introduced for this approach compared to the two other ones.
Furthermore it is assumed that the matrix $\tilde{\mathbf{H}}' = \mathbf{S}^{-1/2}\tilde{\mathbf{H}}\mathbf{S}^{-1/2}$ which is used for the calculation of the density kernel has the same sparsity pattern as the matrices $\mathbf{S}$ or $\tilde{\mathbf{H}}$, which is in general not true. This will introduce an additional error.
It has to be stressed that these inaccuracies are not a shortcoming of the method, but rather errors introduced in trying to reach linear scaling. If the same approximations were applied as well to the other methods – which would be necessary to improve the scaling – the same problems would arise as well.

However there are also some approximations for the FOE method which are inherent to this approach. As explained in Sec. 5.2.3.1, the function that describes the occupation of the eigenstates – i.e. the modified Fermi distribution – corresponds to a finite temperature distribution, whereas the other methods calculate the density kernel at zero temperature. Furthermore the approximation of this function as a polynomial of finite degree introduces as well some small errors.

Due to all these reason, it is to be expected that the two direct diagonalization methods and the direct minimization approach give identical results, whereas the FOE method will perform slightly worse.
In order to validate this assumption, a direct comparison of all four methods was performed for alkanes of varying lengths; the number of atoms ranged from 152 to 1052. The cutoff radius for the support functions was set to 8 bohr, and they were optimized using the hybrid mode with an initial prefactor of $4.9 \cdot 10^{-3}\,\mathrm{hartree/bohr^4}$ for the confinement. To compare the various approaches, the difference between the final energy as calculated by the linear versions using these methods and the one from a cubic refer-

ence calculation was evaluated and divided by the number of atoms in order to have a size-independent quantity.

The results of this test are shown in Fig. 5.21. First of all it can be seen that the energy difference is more or less independent of the length of the alkanes for all four methods, as it should be the case.
Furthermore it is obvious that indeed the diagonalization methods and the direct minimization give almost identical results, whereas the FOE method is slightly worse.
However the deviation between the FOE method and the other ones – which is of the order of $10^{-5}\,\mathrm{hartree/atom}$ – has to be compared with the deviation of the linear results from the cubic reference calculation, which is of the order of $10^{-4}\,\mathrm{hartree/atom}$. Thus the difference between the FOE approach and the other linear methods is one order of magnitude smaller than then overall deviation of the linear results from the cubic one. Therefore it seems to be well justified to use the FOE method in practice. This is important since – as will be shown in Sec. 5.2.4.2 – only the FOE method is at the moment capable to perform calculations which exhibit a strict linear scaling with respect to the size of the system.

In addition the results of these test runs validate the approximations that were used in order to reach linear scaling for the FOE method, and the same approximations can thus also be applied to the other methods in the future.

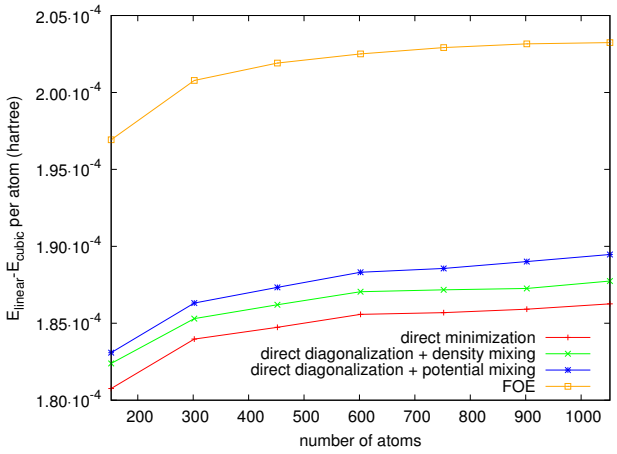


Figure 5.21: Comparison of the accuracy of the various methods to calculate the density kernel. The plot shows the energy difference per atom between the linear version and the cubic one for alkanes of various length. Whereas the diagonalization methods and the direct minimization give very similar results, the FOE method is slightly worse. However the difference is very small; the deviation of the FOE approach from the other ones is roughly one order of magnitude smaller than the deviation of the linear results from the cubic one.

5.2.4.2 Scaling of the kernel methods

As has been demonstrated in the previous section all methods to optimize the density kernel yield more or less identical results. For practical applications one can thus simply choose the fastest approach out of them.

Due to the fact that at the moment only the FOE method has completely eliminated the cubically scaling linear algebra parts, it is to be expected that this method outperforms the other approaches, in particular for large systems.
In order to validate this assumption, the runs which were used in Sec. 5.2.4.1 to check the accuracy of the various methods are now analyzed from the viewpoint of the runtime.

The results are shown in Fig. 5.22. As can be seen the time needed by the four methods is more or less the same for the smallest system, consisting of 152 atoms. This means that the problematic linear algebra parts are not yet important for this size. However the differences between the methods become dramatic as the number of atoms increases. Whereas the FOE method shows a strict linear scaling, the other methods perform much worse. For the largest system, where 2102 support functions were used, the FOE method is 9 times faster than the direct diagonalization with density mixing, which is the second-best method.
This is an impressive demonstration of the importance to remove the bottleneck generated by the cubically scaling linear algebra.

As a consequence the FOE approach is at the moment the only method which is suited to be used for very large systems and will therefore also be employed for the scaling tests in Sec. 6.4.

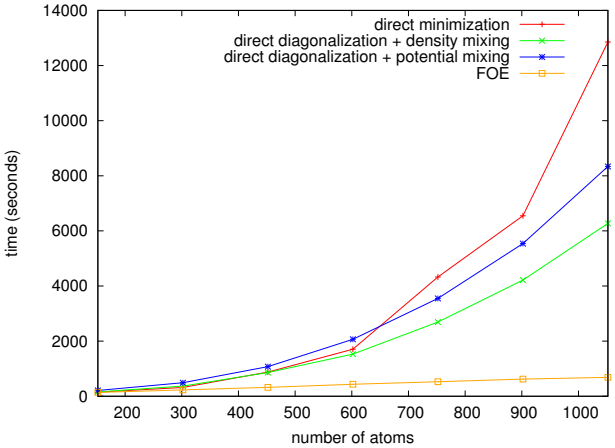


Figure 5.22: Comparison of the total runtime for the four different methods to optimize the density kernel, using the same test system and parameters as in Fig. 5.21. Whereas all methods are close together for the smallest system, it is obvious that only the FOE method scales linearly with respect to the size of the system. This is due to the fact that the FOE approach is the only one which has fully removed the cubically scaling linear algebra. Furthermore it can be seen that the direct diagonalization with density maxing is faster than with potential mixing.

5.3 Forces

So far only the total energy has been considered when comparing the linear scaling version with its cubic counterpart. However the calculation of the forces acting on the nuclei is as well an important output of an electronic structure calculation. Whereas their determination is rather straightforward for the cubic version of BigDFT, this is unfortunately not the case for the linear scaling version.

5.3.1 The Hellmann-Feynman theorem

The evaluation of the forces in a standard cubic DFT calculation is based on the Hellman-Feynman theorem [79, 80], which will be derived in the following.

To this end one assumes that the Hamiltonian depends on some external parameter λ, thus giving rise to λ-dependent eigenvalues and eigenvectors

$$H(\lambda)\psi(\lambda,\mathbf{x}) = \epsilon(\lambda)\psi(\lambda,\mathbf{x}). \tag{5.105}$$

Here $\mathbf{x}$ is any set of coordinates, i.e. ψ can be a single- or many-electron wave function. The Hellman-Feynman theorem then states that the derivative of $\epsilon(\lambda)$ with respect to the external parameter λ is given by

$$\begin{aligned}
\frac{\partial\epsilon(\lambda)}{\partial\lambda} &= \frac{\partial}{\partial\lambda}\int \psi(\lambda,\mathbf{x})H(\lambda)\psi(\lambda,\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&= \int \psi(\lambda,\mathbf{x})\frac{H(\lambda)}{\partial\lambda}\psi(\lambda,\mathbf{x})\,\mathrm{d}\mathbf{x} + \int \frac{\psi(\lambda,\mathbf{x})}{\partial\lambda}H(\lambda)\psi(\lambda,\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&\quad + \int \psi(\lambda,\mathbf{x})H(\lambda)\frac{\partial\psi(\lambda,\mathbf{x})}{\partial\lambda}\,\mathrm{d}\mathbf{x} \\
&= \int \psi(\lambda,\mathbf{x})\frac{H(\lambda)}{\partial\lambda}\psi(\lambda,\mathbf{x})\,\mathrm{d}\mathbf{x} + \epsilon(\lambda)\int \frac{\psi(\lambda,\mathbf{x})}{\lambda}\psi(\lambda,\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&\quad + \epsilon(\lambda)\int \psi(\lambda,\mathbf{x})\frac{\partial\psi(\lambda,\mathbf{x})}{\partial\lambda}\,\mathrm{d}\mathbf{x} \\
&= \int \psi(\lambda,\mathbf{x})\frac{H(\lambda)}{\partial\lambda}\psi(\lambda,\mathbf{x})\,\mathrm{d}\mathbf{x} + \epsilon(\lambda)\frac{\partial}{\partial\lambda}\int \psi(\lambda,\mathbf{x})\psi(\lambda,\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&= \int \psi(\lambda,\mathbf{x})\frac{H(\lambda)}{\partial\lambda}\psi(\lambda,\mathbf{x})\,\mathrm{d}\mathbf{x}.
\end{aligned} \tag{5.106}$$

In the last step the fact that $\int \psi(\lambda,\mathbf{x})\psi(\lambda,\mathbf{x})\,\mathrm{d}\mathbf{x} = 1$ and thus $\frac{\partial}{\partial\lambda}\int \psi(\lambda,\mathbf{x})\psi(\lambda,\mathbf{x})\,\mathrm{d}\mathbf{x} = 0$ was used.

In one wants to calculate the forces acting on the nuclei, the parameter λ corresponds

to the atomic coordinates. Consequently the Hellman-Feynman theorem can be applied to this special case. Again going back to the very beginning and taking the electronic Hamiltonian $\mathcal{H}$ of Eq. (2.3) and the true many-electron wavefunction Φ – i.e. the variable $\mathbf{x}$ now corresponds to all the electronic coordinates $\{\mathbf{r}_1, \ldots, \mathbf{r}_n\}$ – the Hellman-Feynman theorem yields for the forces acting on atom n

$$\begin{aligned}
\mathbf{F}_n &= -\frac{\partial E(\{\mathbf{R}_l\})}{\partial \mathbf{R}_n} \\
&= -\int \mathrm{d}\mathbf{r}_1 \cdots \int \mathrm{d}\mathbf{r}_n \, \Phi(\{\mathbf{R}_l\}, \mathbf{r}_1, \ldots, \mathbf{r}_n) \frac{\partial \mathcal{H}(\{\mathbf{R}_l\}, \mathbf{r}_1, \ldots, \mathbf{r}_n)}{\partial \mathbf{R}_n} \Phi(\{\mathbf{R}_l\}, \mathbf{r}_1, \ldots, \mathbf{r}_n) \\
&= -\sum_{i=1}^{N} \sum_{j=1}^{i-1} \int \mathrm{d}\mathbf{r}_1 \cdots \int \mathrm{d}\mathbf{r}_n \, \Phi(\{\mathbf{R}_l\}, \mathbf{r}_1, \ldots, \mathbf{r}_n) \frac{\partial}{\partial \mathbf{R}_n} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} \Phi(\{\mathbf{R}_l\}, \mathbf{r}_1, \ldots, \mathbf{r}_n) \\
&\quad + \sum_{i=1}^{n} \sum_{j=1}^{N} \int \mathrm{d}\mathbf{r}_1 \cdots \int \mathrm{d}\mathbf{r}_n \, \Phi(\{\mathbf{R}_l\}, \mathbf{r}_1, \ldots, \mathbf{r}_n) \frac{\partial}{\partial \mathbf{R}_n} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|} \Phi(\{\mathbf{R}_l\}, \mathbf{r}_1, \ldots, \mathbf{r}_n) \\
&= -\sum_{i=1}^{N} \sum_{j=1}^{i-1} \frac{\partial}{\partial \mathbf{R}_n} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} + \sum_{j=1}^{N} \int \rho(\mathbf{r}) \frac{\partial}{\partial \mathbf{R}_n} \frac{Z_j}{|\mathbf{r} - \mathbf{R}_j|} \, \mathrm{d}\mathbf{r} \\
&= \sum_{j \neq n} \frac{Z_n Z_j (\mathbf{R}_n - \mathbf{R}_j)}{|\mathbf{R}_n - \mathbf{R}_j|^3} + Z_n \int \rho(\mathbf{r}) \frac{\mathbf{r} - \mathbf{R}_n}{|\mathbf{r} - \mathbf{R}_n|^3} \, \mathrm{d}\mathbf{r},
\end{aligned} \tag{5.107}$$

where the results of Eqs. (2.22a) and (2.22d) were used.
This means that the force depends only on the charge density and is identical to the expression that would arise from a classical charge distribution.

5.3.2 Forces in Density Functional Theory

The derivation of the Hellmann-Feynman theorem was assuming that the energy can be written as $E = \langle \Phi | \mathcal{H} | \Phi \rangle$. Whereas this is true for the many-electron wave function Φ, it is not the case for the Kohn-Sham orbitals ψ_i appearing in DFT.
Here the part of the energy which can be written in this way is the band-structure energy $E_{BS} = \sum_i \langle \psi_i | \mathcal{H} | \psi_i \rangle$. The total energy, however, is given by

$$E(\{\mathbf{R}_l\}) = \sum_i \langle \psi_i(\{\mathbf{R}_l\}) | \mathcal{H}(\{\mathbf{R}_l\}) | \psi_i(\{\mathbf{R}_l\}) \rangle + E_{DC}[\rho], \tag{5.108}$$

where the double counting energy is – according to Eq. (2.53) – given by

$$E_{DC}[\rho] = -\frac{1}{2} \iint \frac{\rho(\mathbf{r}) \rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \, \mathrm{d}\mathbf{r} \mathrm{d}\mathbf{r}' + E_{XC}[\rho(\mathbf{r})] - \int v_{XC}(\mathbf{r}) \rho(\mathbf{r}) \, \mathrm{d}\mathbf{r}. \tag{5.109}$$

Thus the forces are given by

$$\mathbf{F}_n = -\frac{\mathrm{d}E(\{\mathbf{R}_l\})}{\mathrm{d}\mathbf{R}_n} = -\frac{\partial E(\{\mathbf{R}_l\})}{\partial \mathbf{R}_n} - \int \frac{\delta E(\{\mathbf{R}_l\})}{\delta \rho(\mathbf{r})} \frac{\partial \rho(\mathbf{r})}{\partial \mathbf{R}_n} \,\mathrm{d}\mathbf{r}. \tag{5.110}$$

However as soon as a fully self-consistent solution has been found the total energy is a minimum with respect to the electronic density according to Eq. (2.33) and consequently

$$\frac{\delta E}{\delta \rho(\mathbf{r})} = 0. \tag{5.111}$$

Therefore the contribution to the forces stemming from the double counting term is zero and the results of the previous section are still valid, i.e. the forces are given by Eq. (5.107) with the total charge density calculated from the superposition of the individual orbital charge densities,

$$\rho(\mathbf{r}) = \sum_i \rho_i(\mathbf{r}) = \sum_i |\psi_i(\mathbf{r})|^2. \tag{5.112}$$

5.3.3 Forces due to the pseudopotential

So far the formulas for the forces have been derived for the case of all-electron calculations, meaning that the number of Kohn-Sham orbitals is equal to the number of electrons in the system (or equal to half the number in the case of a closed shell calculation).
However BigDFT uses pseudopotentials to simulate the core electrons, and consequently the number of Kohn-Sham electrons is smaller than the total number of electrons. This has also an impact on the calculation of the forces acting on the nuclei.

The first term in Eq. (5.107), which is simply the interaction among the nuclei, is still correct within the pseudopotential framework. The second term, however, needs to be modified since the interactions between the nuclei and the electrons are now described by the pseudopotential [71].

According to Sec. 2.3.4 the pseudopotential is split up in a local and a non-local part. The energy contribution stemming from the local part for an atom located at position $\mathbf{R}_i$ is given by

$$E_{local}(\mathbf{R}_i) = \int \mathcal{V}_{local}(|\mathbf{r} - \mathbf{R}_i|)\rho(\mathbf{r}) \,\mathrm{d}\mathbf{r}. \tag{5.113}$$

The local potential can further be split up in a long-range and a short range part:

$$\begin{aligned}
&\mathcal{V}_{local}(\lambda) = \mathcal{V}_L(\lambda) + \mathcal{V}_S(\lambda),\\
&\mathcal{V}_L(\lambda) = \frac{Z_{ion}}{\lambda}\mathrm{erf}\left(\frac{\lambda}{\sqrt{2}r_{loc}}\right),\\
&\mathcal{V}_S(\lambda) = e^{-r^2/2r_{loc}^2}\left[C_1 + C_2\left(\frac{r}{r_{loc}}\right)^2 + C_3\left(\frac{r}{r_{loc}}\right)^4 + C_4\left(\frac{r}{r_{loc}}\right)^6\right].
\end{aligned} \tag{5.114}$$

By defining a "long-range charge density" ρ_L such that

$$\nabla^2_{\mathbf{r}}\mathcal{V}_L(|\mathbf{r}-\mathbf{R}_i|) = -4\pi\rho_L(|\mathbf{r}-\mathbf{R}_i|), \tag{5.115}$$

the local energy contribution can be rewritten as

$$E_{local}(\mathbf{R}_i) = \int \rho_L(|\mathbf{r}-\mathbf{R}_i|)\mathcal{V}_H(\mathbf{r})\,\mathrm{d}\mathbf{r} + \int \mathcal{V}_S(|\mathbf{r}-\mathbf{R}_i|)\rho(\mathbf{r})\,\mathrm{d}\mathbf{r}, \tag{5.116}$$

where $\mathcal{V}_H$ is the Hartree potential which is the solution of

$$\nabla^2\mathcal{V}_H(\mathbf{r}) = -4\pi\rho(\mathbf{r}). \tag{5.117}$$

From (5.114) and (5.115) the value of ρ_L can be calculated; the final result is given by

$$\rho_L(\lambda) = -\frac{1}{(2\pi)^{3/2}}\frac{Z_i}{r_{loc}^3}e^{-\lambda^2/2r_{loc}^2}. \tag{5.118}$$

The forces stemming from this local part of the pseudopotential can now be determined by calculating the derivative with respect to the atomic coordinates and are given by

$$\mathbf{F}_i^{local} = \frac{\partial E_{local}}{\partial \mathbf{R}_i} = \frac{1}{r_{loc}}\int \frac{\mathbf{r}-\mathbf{R}_i}{|\mathbf{r}-\mathbf{R}_i|}\left[\rho_L'(|\mathbf{r}-\mathbf{R}_i|)\mathcal{V}_H(\mathbf{r}) + \mathcal{V}_S'(|\mathbf{r}-\mathbf{R}_i|)\rho(\mathbf{r})\right]\,\mathrm{d}\mathbf{r} \tag{5.119}$$

with

$$\begin{aligned}
&\rho_L'(\lambda) = \frac{1}{(2\pi)^{3/2}}\frac{Z_i}{r_{loc}^4}\lambda e^{-\lambda^2/2r_{loc}^2},\\
&\mathcal{V}_S' = \frac{\lambda}{r_{loc}}e^{-\lambda^2/2r_{loc}^2}\\
&\quad\times\left[(2C_2 - C_1) + (4C_3 - C_2)\left(\frac{\lambda}{r_l}\right)^2 + (6C_4 - C_3)\left(\frac{\lambda}{r_{loc}}\right)^4 - C_4\left(\frac{\lambda}{r_{loc}}\right)^6\right].
\end{aligned} \tag{5.120}$$

Since both ρ_L' and $\mathcal{V}_S'$ are localized functions due to the presence of the Gaussians, the integral of Eq. (5.119) only has to be performed in a relatively small region around the atom.

The nonlocal part of the pseudopotential is – according to Eq. (2.60) – given by

$$E_{nonlocal}(\mathbf{R}_i) = \sum_j \sum_l \sum_{m,n=1}^{3} \langle \psi_j | p_m^{(l)}(\mathbf{R}_i) \rangle \, h_{mn}^{(l)} \, \langle p_n^{(l)}(\mathbf{R}_i) | \psi_j \rangle \,, \tag{5.121}$$

where the dependence of the projector on the atomic position has been explicitly noted. The forces are thus given by

$$\begin{aligned} \mathbf{F}_i^{nonlocal} = -\sum_j \sum_l \sum_{m,n=1}^{3} \Bigg[& \left\langle \psi_j \left| \frac{\partial p_m^{(l)}(\mathbf{R}_i)}{\partial \mathbf{R}_i} \right\rangle h_{mn}^{(l)} \left\langle p_n^{(l)}(\mathbf{R}_i) \right| \psi_j \right\rangle \\ & - \left\langle \psi_j \left| p_m^{(l)}(\mathbf{R}_i) \right\rangle h_{mn}^{(l)} \left\langle \frac{\partial p_n^{(l)}(\mathbf{R}_i)}{\mathbf{R}_i} \right| \psi_j \right\rangle \Bigg] . \end{aligned} \tag{5.122}$$

Expressing the derivatives of the projectors in the same wavelet basis as the Kohn-Sham orbitals, the evaluation of the scalar products in Eq. (5.122) is straightforward.

5.3.4 Forces in terms of the support functions and the density kernel

In the context of the linear scaling version one does not have the Kohn-Sham orbitals at hand in order to evaluate the forces, but only the support functions and the density kernel. However their calculation can still be done straightforwardly in terms of these quantities.

The evaluation of the local part of the forces can actually be carried out in the same way as for the cubic version, since it requires only the charge density and the Hartree potential which are both available.
In order to calculate the nonlocal forces, the Kohn-Sham orbitals have to be replaced by their representation in terms of the support functions, $\psi_j = \sum_\alpha c_{j\alpha} \phi^\alpha$. Afterwards the sum over j can be evaluated, leading to

$$\begin{aligned} \mathbf{F}_i^{nonlocal} = -\sum_{\alpha,\beta} \sum_l \sum_{m,n=1}^{3} K_{\alpha\beta} \Bigg[& \left\langle \phi^\alpha \left| \frac{\partial p_m^{(l)}(\mathbf{R}_i)}{\partial \mathbf{R}_i} \right\rangle h_{mn}^{(l)} \left\langle p_n^{(l)}(\mathbf{R}_i) \right| \phi^\beta \right\rangle \\ & - \left\langle \phi^\alpha \left| p_m^{(l)}(\mathbf{R}_i) \right\rangle h_{mn}^{(l)} \left\langle \frac{\partial p_n^{(l)}(\mathbf{R}_i)}{\mathbf{R}_i} \right| \phi^\beta \right\rangle \Bigg] , \end{aligned} \tag{5.123}$$

where the density kernel is – according to Eq. (3.38) – given by $K_{\alpha\beta} = \sum_j c_{j\alpha} c_{j\beta}$.

5.3.5 Pulay forces

For the linear scaling version the calculation of the forces is more involved than for the cubic version. In contrast to the latter case, where the Kohn-Sham orbitals are directly represented in the wavelet basis whose functional form is independent of the positions of the atoms, the orbitals are this time expanded in terms of the support functions ϕ^α:

$$\psi_i(\mathbf{r}) = \sum_\alpha c_{i\alpha}\phi^\alpha(\mathbf{r}). \tag{5.124}$$

The problem is that these support functions depend explicitly on the atomic positions.

Thus the force acting on atom n is – taking into account the results of Sec. 5.3.2, stating that it is only necessary to consider the band-structure part of the energy for the calulation of the forces – this time given by

$$\begin{aligned}
\mathbf{F}_n &= -\frac{\partial}{\partial \mathbf{R}_n}\sum_i \int \psi_i(\{\mathbf{R}_l\},\mathbf{r})\mathcal{H}(\{\mathbf{R}_l\},\mathbf{r})\psi_i(\{\mathbf{R}_l\},\mathbf{r})\,\mathrm{d}\mathbf{r} \\
&= -\sum_i \int \psi_i(\{\mathbf{R}_l\},\mathbf{r})\frac{\partial \mathcal{H}(\{\mathbf{R}_l\},\mathbf{r})}{\partial \mathbf{R}_n}\psi_i(\{\mathbf{R}_l\},\mathbf{r})\,\mathrm{d}\mathbf{r} \\
&\quad -\sum_i \int \frac{\partial \psi_i(\{\mathbf{R}_l\},\mathbf{r})}{\partial \mathbf{R}_n}\mathcal{H}(\{\mathbf{R}_l\},\mathbf{r})\psi_i(\{\mathbf{R}_l\},\mathbf{r})\,\mathrm{d}\mathbf{r} \\
&\quad -\sum_i \int \psi_i(\{\mathbf{R}_l\},\mathbf{r})\mathcal{H}(\{\mathbf{R}_l\},\mathbf{r})\frac{\partial \psi_i(\{\mathbf{R}_l\},\mathbf{r})}{\partial \mathbf{R}_n}\,\mathrm{d}\mathbf{r}.
\end{aligned} \tag{5.125}$$

The first term is the standard Hellman-Feynman force term which has been calculated previously. The two other terms, which do not vanish any more, are known as Pulay forces [81]. They are present as soon as a basis set which explicitly depends on the atomic positions – such as Gaussians – is used and are therefore not limited to the case of a linear scaling DFT code, but can also appear in a traditional one exhibiting a cubic scaling.

It has to be stressed that, even for support functions that depend explicitly on the atomic positions, the Pulay terms can be exactly zero. This would be the case if the support functions formed a complete set such that the linear combinations of Eq. (5.124) would yield the exact eigenfunctions of $\mathcal{H}$. In this case the equation $\mathcal{H}\psi_i = \epsilon_i\psi_i$ would hold and the representation of the Kohn-Sham orbitals in the basis of the support functions would be equivalent to their representation in terms of the underlying wavelet basis, which is free of any Pulay terms.

The next step is to see whether it is possible to calculate the Pulay corrections to the forces. The band-structure energy is – this time written in terms of the density kernel

and the support functions instead of the fictitious Kohn-Sham orbitals – given by the expression

$$E_{BS}(\{\mathbf{R}_l\}) = \sum_{\alpha,\beta} K_{\alpha\beta}(\{\mathbf{R}_l\}) \int \phi^{\alpha}(\{\mathbf{R}_l\},\mathbf{r})\mathcal{H}(\{\mathbf{R}_l\},\mathbf{r})\phi^{\beta}(\{\mathbf{R}_l\},\mathbf{r})\,\mathrm{d}\mathbf{r}, \tag{5.126}$$

where this time the dependence of the density kernel, the support functions and the Hamiltonian on the atomic coordinates is written. Using this expression the forces acting on atom n are given by

$$\mathbf{F}_n = -\frac{\mathrm{d}E_{BS}}{\mathrm{d}\mathbf{R}_n} = -\frac{\partial E_{BS}}{\partial \mathbf{R}_n} - \sum_{\alpha,\beta} \frac{\partial E_{BS}}{\partial K_{\alpha\beta}} \frac{\partial K_{\alpha\beta}}{\partial \mathbf{R}_n} - \sum_{\alpha} \int \frac{\delta E_{BS}}{\delta \phi^{\alpha}(\mathbf{r})} \frac{\partial \phi^{\alpha}(\mathbf{r})}{\partial \mathbf{R}_n}\,\mathrm{d}\mathbf{r}. \tag{5.127}$$

The first term describing the explicit dependence of the energy on the atomic coordinates is the standard Hellmann-Feynman term, whereas the second and third terms describing the implicit dependence of the energy on the atomic coordinates are the Pulay corrections.

In case the energy is perfectly converged to zero with respect to both the density kernel and the support functions, i.e.

$$\frac{\partial E_{BS}}{\partial K_{\alpha\beta}} = 0 \quad \text{and} \quad \frac{\delta E_{BS}}{\delta \phi^{\alpha}(\mathbf{r})} = 0 \tag{5.128}$$

for all values of α and β, then the additional Pulay terms are zero and one is left with the standard Hellman-Feynman force. This is the second case – in addition to the one where the support functions permit an exact representation of the Kohn-Sham orbitals – which gives no Pulay forces even though a basis set explicitly depending on the atomic positions is used.

In practice neither of these two situations will arise. Thus one has in general to deal with Pulay forces. Explicitly writing out Eq. (5.127) gives for the forces acting on atom n

$$\begin{aligned}
\mathbf{F}_n = &-\sum_{\alpha,\beta} K_{\alpha\beta}(\{\mathbf{R}_l\}) \int \phi^{\alpha}(\{\mathbf{R}_l\},\mathbf{r}) \frac{\partial \mathcal{H}(\{\mathbf{R}_l\},\mathbf{r})}{\partial \mathbf{R}_n} \phi^{\beta}(\{\mathbf{R}_l\},\mathbf{r})\,\mathrm{d}\mathbf{r} \\
&-\sum_{\alpha,\beta} \frac{\partial K_{\alpha\beta}(\{\mathbf{R}_l\})}{\partial \mathbf{R}_n} \int \phi^{\alpha}(\{\mathbf{R}_l\},\mathbf{r})\mathcal{H}(\{\mathbf{R}_l\},\mathbf{r})\phi^{\beta}(\{\mathbf{R}_l\},\mathbf{r})\,\mathrm{d}\mathbf{r} \\
&-\sum_{\alpha,\beta} K_{\alpha\beta}(\{\mathbf{R}_l\}) \left[\int \frac{\partial \phi^{\alpha}(\{\mathbf{R}_l\},\mathbf{r})}{\partial \mathbf{R}_n} \mathcal{H}(\{\mathbf{R}_l\},\mathbf{r})\phi^{\beta}(\{\mathbf{R}_l\},\mathbf{r})\,\mathrm{d}\mathbf{r} \right. \\
&\left. + \int \phi^{\alpha}(\{\mathbf{R}_l\},\mathbf{r})\mathcal{H}(\{\mathbf{R}_l\},\mathbf{r}) \frac{\partial \phi^{\beta}(\{\mathbf{R}_l\},\mathbf{r})}{\partial \mathbf{R}_n}\,\mathrm{d}\mathbf{r} \right].
\end{aligned} \tag{5.129}$$

The first term, which contains the derivative of the Hamiltonian, is the standard Hellman-Feynman term and can be calculated as shown previously. The last two terms can as well be calculated since the derivatives of the support functions appearing in them can be determined thanks to the fact that the support functions are represented in terms of the underlying wavelet basis. What remains is the second term containing the derivative of the density kernel; unfortunately this quantity can not be evaluated straightforwardly.

A possible solution for the problem might be to neglect this term and to calculate only those Pulay corrections stemming from the terms containing the derivatives of the support functions. However this is probably a rather bad idea, since it might well happen that some of the Pulay forces caused by the terms containing the derivatives of the density kernel cancel at least partially with those contributions caused by the terms containing the derivatives of the support functions; thus a neglect of only one part could lead to an increase of the Pulay forces instead of a reduction.
Therefore it is probably better to neglect as well those terms containing the derivative of the support functions and to keep only the Hellman-Feynman forces:

$$\mathbf{F}_n \approx \sum_{\alpha,\beta} K_{\alpha\beta}(\{\mathbf{R}_l\}) \int \phi^{\alpha}(\{\mathbf{R}_l\},\mathbf{r}) \frac{\partial \mathcal{H}(\{\mathbf{R}_l\},\mathbf{r})}{\partial \mathbf{R}_n} \phi^{\beta}(\{\mathbf{R}_l\},\mathbf{r})\,\mathrm{d}\mathbf{r}. \tag{5.130}$$

A check whether this is a sensible approximation is performed in Sec. 6.1.2; as will be shown the accuracy obtained in this way is usually sufficient in practice.

Still it must be noted that simply neglecting the other terms is only a stopgap and there might well be cases where they are important. Therefore it remains a task for the future to see whether their calculation is possible nevertheless.

CHAPTER 6

Benchmarking the linear scaling version of BigDFT

6.1 Accuracy of the linear scaling version

The accuracy of the linear scaling version will be checked from three viewpoints, always comparing its results with those obtained by a reference calculation using the cubic version. The first check will be a comparison of the energy, the second one will compare the forces, and the last one will analyze the outcome of a geometry optimization.

6.1.1 Accuracy of the energy

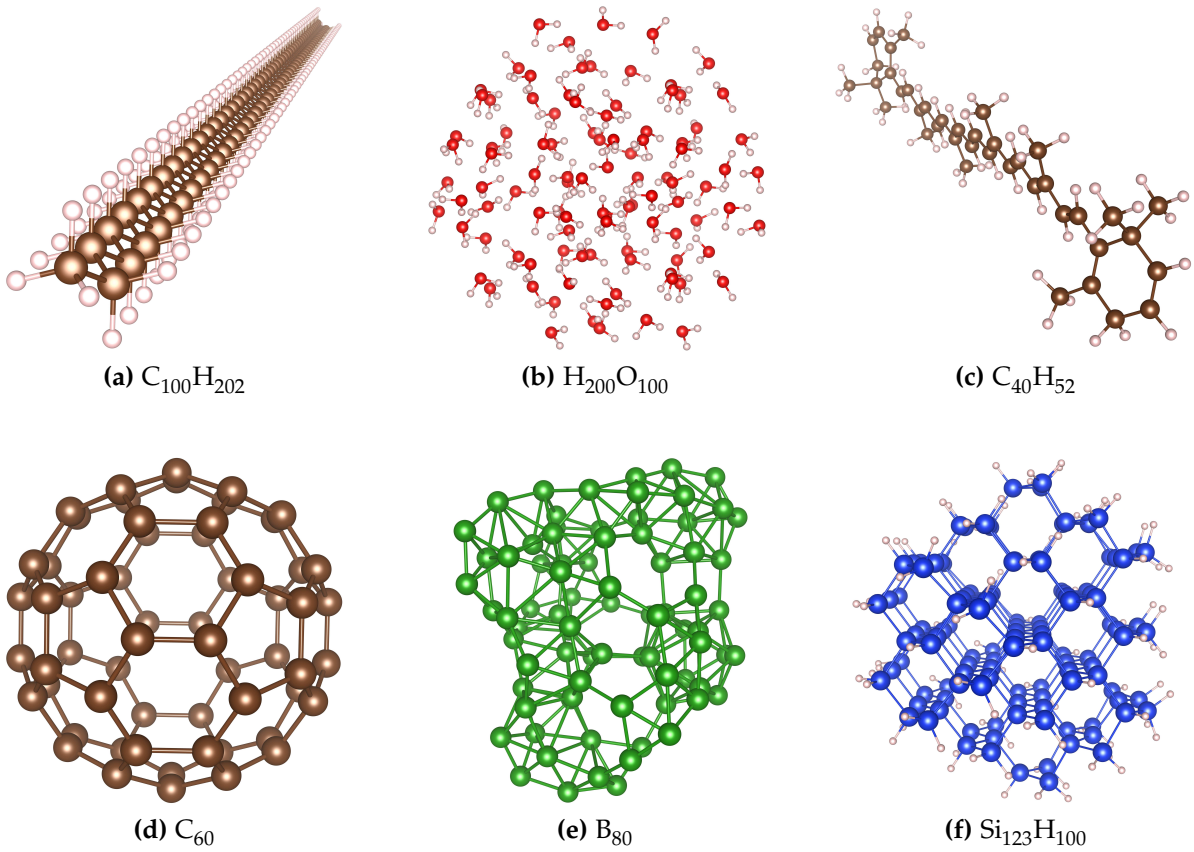
The easiest way to estimate the accuracy of the linear scaling version is to compare the final energy with the one obtained with the cubic scaling version of the code. This comparison will be carried out for several systems in order to get a broad overview over the performance of the linear scaling version.

The 6 systems that were used for the tests are an alkane $C_{100}H_{202}$, a water droplet $H_{200}O_{100}$, a carotene molecule $C_{40}H_{52}$, a carbon fullerene C_{60}, a boron cluster B_{80} and a silicon-hydrogen cluster $Si_{123}H_{100}$. Figures of these systems are shown in Fig. 6.1.

In order to test the accuracy, four calculations were done in total for each system: First one using the linear version and one using the cubic version, respectively, for the origi-

nal geometry, and then the same for a slightly modified configuration where all atoms were randomly shifted a bit. The linear scaling calculations were all done using the hybrid mode for the optimization of the support functions and the FOE method for the optimization of the density kernel. As a further complication, the input parameters affecting the performance of the linear scaling version were identical for all systems, namely 9 bohr for the cutoff radius of the support functions and $3 \cdot 10^{-3}$ hartree/bohr^4 for the initial prefactor of the confinement. The parameter for the density mixing – which is a well an important parameter and depends on the gap of the system – was set to a small value of 0.1, which is a conservative estimation that should hopefully fit all configurations. The number of support functions was chosen to be a minimal basis set.

The results of these runs are shown in Tab. 6.1. As can be seen the energies calculated by the linear and the cubic version are always rather close together, which is pretty



(a) $C_{100}H_{202}$ **(b)** $H_{200}O_{100}$ **(c)** $C_{40}H_{52}$

(d) C_{60} **(e)** B_{80} **(f)** $Si_{123}H_{100}$

Figure 6.1: The six systems which were used for the accuracy tests. Not all configurations were fully relaxed to their configurational ground state. The water droplet exhibits the largest force norm of $2.43 \cdot 10^{-1}$ hartree/bohr, whereas the carbon fullerene, which is the system exhibiting the smallest value, is relaxed to a force norm of $1.51 \cdot 10^{-6}$ hartree/bohr. The other systems have force norms of the order of 10^{-3} hartree/bohr.

	configuration 1			configuration 2		
	E_{linear} (eV)	E_{cubic} (eV)	ΔE ($^{\text{meV}}/_{\text{atom}}$)	E_{linear} (eV)	E_{cubic} (eV)	ΔE ($^{\text{meV}}/_{\text{atom}}$)
$C_{100}H_{202}$	-18733.538	-18734.014	1.58	-18730.898	-18731.378	1.59
$H_{200}O_{100}$	-46806.816	-46807.425	2.03	-46804.219	-46804.845	2.09
$C_{40}H_{52}$	-7027.518	-7028.182	7.22	-7026.605	-7027.277	7.31
C_{60}	-9301.260	-9302.436	19.61	-9300.353	-9301.578	20.42
B_{80}	-6138.687	-6141.208	31.52	-6138.278	-6140.744	30.83
$Si_{123}H_{100}$	-14808.389	-14815.567	32.19	-14807.388	-14814.599	32.33

Table 6.1: The results of the accuracy test runs for the six systems shown in Fig. 6.1. For both configurations, the first column shows the energy calculated by the linear version, the second one the energy calculated by the cubic version, and the third one the difference between the two values. As can be seen, this difference varies considerably, from about $1\,^{\text{meV}}/_{\text{atom}}$ for the alkane up to more than $30\,^{\text{meV}}/_{\text{atom}}$ for the boron and the silicon-hydrogen cluster.

nice in view of the fact that the chosen parameters for the linear scaling version were always identical. However there is still quite some variation in the quality of the results for the various systems, which becomes visible when looking at the energy differences between the linear and the cubic version, which range from about $1\,^{\text{meV}}/_{\text{atom}}$, which is an excellent value, to more than $30\,^{\text{meV}}/_{\text{atom}}$, which is still good, but nevertheless considerably worse than the other result. As will be shown in Sec. 6.2.1, the energy will converge exponentially towards the cubic value if the cutoff radius is increased; thus it is possible to systematically improve these results.

On the other hand, one is in general not that much interested in absolute energy values, but rather in energy differences between two structures. Consequently there is hope that the energy offset between the linear and the cubic version remains more or less constant for different configurations and that therefore energy differences are very

	energy difference		
	linear (eV)	cubic (eV)	Δ ($^{\text{meV}}/_{\text{atom}}$)
$C_{100}H_{202}$	-2.640	-2.636	-0.01
$H_{200}O_{100}$	-2.598	-2.579	-0.06
$C_{40}H_{52}$	-0.913	-0.905	-0.09
C_{60}	-0.907	-0.858	-0.81
B_{80}	-0.408	-0.464	0.69
$Si_{123}H_{100}$	-1.001	-0.968	-0.15

Table 6.2: The energy differences between the two configurations for the same systems as in Tab. 6.1. The first column gives the value obtained by using the linear version, the second one the value calculated with the cubic version. The last column shows the difference between the two values. It is obvious that the energy differences are much more accurate than the absolute values shown in Tab. 6.1.

accurate even for those systems exhibiting a large offset.
This is actually confirmed by the values shown in Tab. 6.2, which shows the energy difference between the two configurations for both the linear and the cubic version. As can be seen, the values are very similar; the largest deviation is only $-0.81\,\mathrm{meV/atom}$, which is considerably better than the absolute differences shown in Tab. 6.1.

To summarize, this test shows that it is possible to get very accurate results – in particular energy differences – with a standard set of parameters, which is of great importance for practical applications.

6.1.2 Accuracy of the forces

In Sec. 5.3.5 it has been shown that it is not possible to calculate the Pulay forces which arise for the linear scaling version. Consequently the forces will not exactly be the negative derivative of the energy as they should. However, even if it was possible to calculate the additional Pulay forces, the forces would still not be identical to the ones obtained by a cubic reference calculation due to the effect of the strict localization of the support functions and the cutoff radius used for the density kernel construction. If this additional error – which is inherent to any linear scaling code – is larger than the error introduced by the neglect of the Pulay forces, then this approximation is well justified.

In order to check whether the forces are accurate or not one should thus not compare the linear forces with their cubic counterpart – i.e. to look at the value of $|\mathbf{F}_{\mathrm{linear}} - \mathbf{F}_{\mathrm{cubic}}|$ – but rather verify whether the forces are the negative derivative of the energy.
To do so, an initial configuration $\mathbf{R}^{(a)}$ and a final configuration $\mathbf{R}^{(b)}$ are chosen, where $\mathbf{R}$ stands for all the atomic positions and is thus a vector of length $3N$. Now the initial configuration is slowly transformed into the final one using small steps of length $\Delta\mathbf{R}$. The energy difference between the two configurations can then be approximated by

$$\Delta E = \int_a^b \mathbf{F}(\mathbf{r})\,\mathrm{d}\mathbf{r} \approx \sum_{\mu} \mathbf{F}(\mathbf{R}^{(a)} + \mu\Delta\mathbf{R})\Delta\mathbf{R}, \tag{6.1}$$

where the force vector $\mathbf{F}$ has as well dimension $3N$ and μ indicates the intermediate steps to get from configuration a to configuration b.
This approximation can then be compared with the exact value obtained by directly calculating the energy differences, i.e. $E(\mathbf{R}^{(b)}) - E(\mathbf{R}^{(a)})$. If the forces are exactly the negative derivative of the energy, these two values will agree up to the noise level of the calculation.

In order to estimate the magnitudes of the various error sources, this test was done for five different setups:

1. First a reference calculation was done with the traditional cubically scaling Kohn-Sham scheme where all orbitals are allowed to extend over the entire simulation box. This will give the noise level of this test setup, since everything is absolutely exact.

2. Next a calculation was done with the linear scaling version, but neither a cutoff for the support functions and the density kernel construction nor a confining potential was used. In this way one has to get back the same results as for the reference calculation.

3. The third step was to switch on the confining potential – in order to avoid instabilities due to large gradients, it was necessary to reduce the step size (and thus to increase the number of iterations) used in the optimization of the support functions. In this way errors from the neglect of the Pulay terms should be introduced since the support functions depend now on the atomic positions. On the other hand there is still no error from the localization constraints.

4. The fourth step was to introduce a finite localization radius of 15 bohr for the kernel construction, but not yet for the support functions.

5. Finally the localization constraints for both the support functions – a value of 9 bohr was chosen – and the density kernel construction were applied. In this way one will get errors from both the strict localization and the neglect of the Pulay forces.

This test was done for an alkane consisting of 92 atoms. Even though this is not a very large number, the introduction of finite cutoff radii for the support functions and the density kernel construction will have a strong effect due to the chain-like structure of the molecule. The support functions were optimized using the hybrid mode, and the FOE approach was used for the determination of the density kernel. The step size to go from configuration $\mathbf{R}^{(a)}$ to configuration $\mathbf{R}^{(b)}$ was set to 0.005 bohr.

The results for all four setups are shown in Tab. 6.3. For the cubic reference calculation the discrepancy between the energy difference and the integral is of the order of 10^{-6}; this seems to be the noise level for this test, since there are no approximations in the cubic version and the forces have to be exactly the negative derivative of the energy. Therefore all linear setups that yield differences which are as well of this order of magnitude can be considered as exact.

It is obvious that this is more or less the case for all linear setups that do not apply

any localization constraint on the support functions, independent of whether they use a confinement or not. This shows that the influence of the Pulay forces – which should arise as soon as one introduces a confinement making the support functions dependent on the atomic positions and are thus present in the third, fourth and fifth setup – is very small.
The introduction of the finite cutoff radius for the construction of the density kernel does not deteriorate the quality of the results either; as will be shown later in Sec. 6.2.1.2 the accuracy with respect to this cutoff radius saturates quite rapidly and a value of 15 bohr is enough to reduce the error to the noise level. However, a much sharper drop in the accuracy appears as soon as the finite localization is imposed on the support functions by applying a strict cutoff radius of 9 bohr. Here the discrepancy between the energy difference and the force integral increases suddenly by almost two orders of magnitude.

In summary, this test demonstrates that the error introduced by the strict localization of the support functions is much larger than that caused by the neglect of the Pulay forces. Therefore it seems to be well justified to adopt this approximation and to use Eq. (5.130) to determine the forces.

	ΔE (hartree)	$\int \mathbf{F}(\mathbf{r})\,\mathrm{d}\mathbf{r}$ (hartree)	difference (hartree)
cubic	0.2082569	0.2082653	$-8.4 \cdot 10^{-6}$
linear global, no confinement	0.2082571	0.2082583	$-1.2 \cdot 10^{-6}$
linear global, with confinement	0.2082568	0.2082592	$-2.4 \cdot 10^{-6}$
linear global, FOE cutoff	0.2082671	0.2082592	$7.9 \cdot 10^{-6}$
linear global, both cutoffs	0.2083683	0.2084804	$-1.1 \cdot 10^{-4}$

Table 6.3: Overview of the various methods that were used to test the quality of the forces. "cubic" is the reference calculation with the cubic version and has to be considered as exact, i.e. the error seen here is the inevitable noise that is always present in DFT calculations, together with the one introduced by the approximation of the integral as a finite sum. "linear global, no confinement" is the linear version using the global localization region (i.e. employing an infinite cutoff radius) and no confinement. "linear global, with confinement" is the same, but this time using a confinement for the support functions (prefactor set to $3.0 \cdot 10^{-3}$ $\mathrm{hartree}/\mathrm{bohr}^4$). "linear global, FOE cutoff" adds in addition a cutoff radius of 15 bohr for the construction of the density kernel. "linear global, both cutoffs" finally uses in addition a cutoff radius of 9 bohr for the support functions and is thus the standard linear version. It is obvious that the effect of strictly localizing the support functions is much stronger than those of the confinement and the cutoff for the density kernel construction.

6.1.3 Geometry optimizations

Whereas one single electronic structure calculation only determines the electronic ground state for a given nucleonic configuration, a geometry optimization searches for minima on the Born-Oppenheimer surface. To this end, the atoms are moved according to the forces – which are the output of the electronic structure calculation – acting on them, in this way lowering the energy of the system until a point is found where the forces vanish. This means that the electronic structure problem has to be solved for many nucleonic configurations until finally the energetically most favorable one is reached.
The linear scaling version of BigDFT can use any of the geometry optimization methods available for the cubic version, which include SD [73], CG [73], DIIS [72], FIRE [82], and various flavors of BFGS [73].

It is clear that a reasonable geometry optimization is only possible if the forces are reliable. Therefore geometry optimizations can also be used to validate the accuracy of the forces. One starts with a given configuration where the forces acting on the atoms are non-zero, and relaxes the structure both with cubic and the linear version. If the forces of the linear scaling version are accurate enough, the structure should evolve in a similar way as with the cubic version.

The system which was used to carry out this test was the alkane $C_{100}H_{202}$; this system is still small enough such that the cubic version runs fast enough, but at the same time a good candidate to test the linear scaling version due to its chain-like structure. The support functions were optimized with the hybrid mode, and the density kernel was determined using the FOE approach. The cutoffs were set to 9 bohr for the support functions and 15 bohr for the density kernel construction, and the initial prefactor for the confinement was set to $3 \cdot 10^{-3}$ hartree/bohr^4. Four support functions were centered on each carbon atom and one on each hydrogen atom, i.e. a minimal basis set was

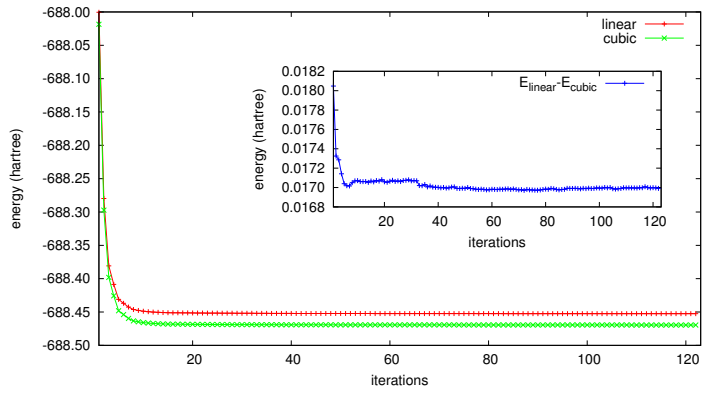


Figure 6.2: The energy as a function of the iterations in the geometry optimization for both the linear and the cubic version. The values decrease in a similar manner, which becomes even more clear by the small inset showing the energy difference at each iteration. The geometry optimization was performed using the BFGS algorithm for the movements of the atoms.

used.

The energies calculated by both the cubic and the linear version as a function of the iterations in the geometry optimization are shown in Fig. 6.2. It is obvious that both curves decrease in a similar manner, i.e. the offset between the cubic and the linear version seems to remains more or less constant throughout the entire calculation. This can be best seen from the inset which shows the difference between the two curves; this value lies always in between 0.0170 hartree and 0.0182 hartree, which is a very small variation in view of the total energy of about -688 hartree.

Another quantity that can be compared is the force acting on the atoms at each iteration. Since the forces are – at least if they can be calculated correctly – the negative derivative of the energy with respect to the atomic coordinates, the constant offset which is present in the energies should vanish and the forces should consequently be identical for both the linear and the cubic version. If there are differences between the two curves, this indicates that the forces of the linear version are not the negative derivative of the energy, which is a consequence of the neglect of the Pulay forces and the finite cutoff radii.
The results for the same test system are shown in Fig. 6.3. In the large plot the force norm at each iteration is shown; it is obvious that the two curves exhibit an excellent agreement. The small inset, which shows the norm of the difference of the forces, i.e. the value of $|\mathbf{F}_{\text{linear}} - \mathbf{F}_{\text{cubic}}|$, supports this observation; in view of the fact that the noise level for this calculation was of the order of 10^{-4} hartree/bohr to 10^{-5} hartree/bohr, the agreement is actually marvellous.

The last quantity that can be studied is the root mean square displacement (RMSD) between the linear and the cubic structure at each iteration of the geometry optimization,

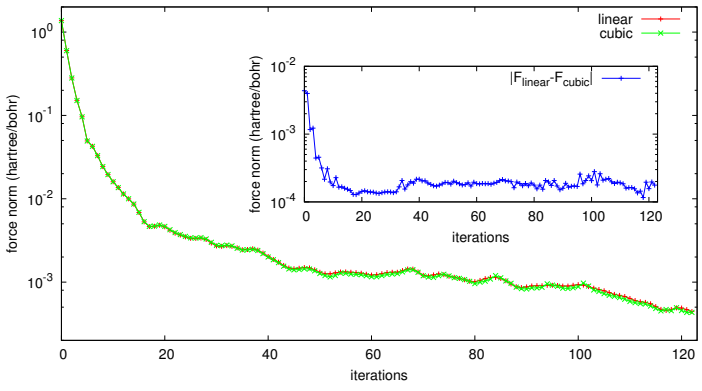


Figure 6.3: The force norm of the linear and the cubic version as a function of the iterations in the geometry optimization. The agreement between the two curves is excellent. This becomes even more visible in the inset, which shows the norm of the difference between the two forces, i.e. the value $|\mathbf{F}_{\text{linear}} - \mathbf{F}_{\text{cubic}}|$. The numbers were taken from the same test as the ones for the energy comparison.

defined by

$$\mathrm{RMSD}(i) = \frac{1}{\sqrt{N}} \sqrt{\sum_{j=1}^{N} \left(\mathbf{R}^j_{\mathrm{linear}}(i) - \mathbf{R}^j_{\mathrm{cubic}}(i)\right)^2}, \tag{6.2}$$

where N is the number of atoms and $\mathbf{R}^j_{\mathrm{linear}}(i)$ and $\mathbf{R}^j_{\mathrm{cubic}}(i)$ are the positions of atom j at the ith step of the geometry optimization for the linear and cubic version, respectively. In the beginning the RMSD is zero since both the linear and the cubic version start from the same initial structure. In the course of the geometry optimization this value will increase, but if it remains small throughout the entire optimization, this means that the two structures evolve in a very similar manner.
The results, again for the same test system, are shown in Fig. 6.4. As expected, the RMSD slightly increases during the optimization, but saturates towards the end and remains always below 0.01 bohr. This is an extremely small value which is not even visible by eye and lies below the typical error of DFT calculations. Consequently the final configuration of the linear and the cubic geometry optimization can be considered as identical.

Together these results demonstrate once more that the linear version yields vary accurate forces which are of high enough quality to be used for geometry optimizations.

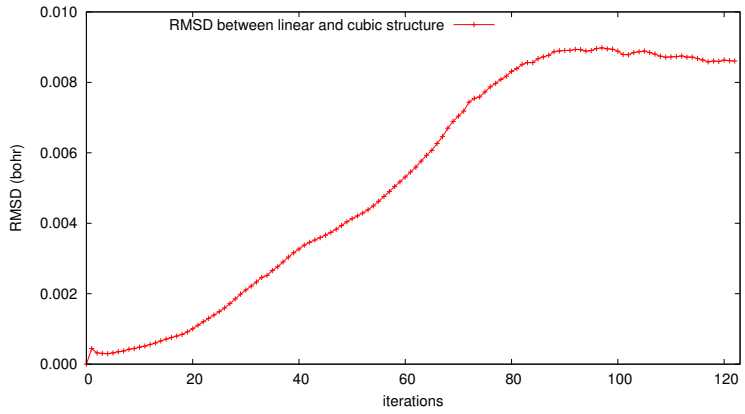


Figure 6.4: The RMSD between the linear and the cubic structures as a function of the iterations in the geometry optimization, again for the same run as the other two plots. The value slightly increases, but always stays below 0.01 bohr, which is admittedly very small.

6.2 Performance with respect to the cutoff radius

This section will address the question of how the performance of the codes varies with respect to the cutoff radii. First the accuracy will be investigated – where it is to be

expected that the results converge towards those of a cubic reference calculation – and in a second part the runtime as a function of the cutoff radius will be examined.

6.2.1 Convergence with respect to the cutoff radius

An important property that a linear scaling DFT code should exhibit is the ability to reproduce the results from a reference calculation using the cubically scaling version of the code in the limit of an infinitely large cutoff radius. It is obvious that more time will be required to determine the solution due to the larger prefactor of the linear scaling method, but the results should be identical up to the noise level.

To compare the results of a run using the linear scaling version with the reference calculation employing the cubic one, three quantities were considered:

- The difference of the total energy $E_{\text{linear}} - E_{\text{cubic}}$, which should go to zero as the cutoff radius goes towards infinity.
- The norm of the difference of the forces $|\mathbf{F}_{\text{linear}} - \mathbf{F}_{\text{cubic}}|$, which should as well go zero as the radius goes towards infinity.
- The consistency between energy and forces as described in Sec. 6.1.2, i.e. the quantity $\Delta E - \int \mathbf{F}(\mathbf{r})\,\mathrm{d}\mathbf{r}$. Here the value for the linear scaling run should go down to the same noise level as the cubic reference.

In the most general case, there is not only one cutoff radius for the support functions, but two, namely one for the contravariant ones and one for the covariant ones, as was explained in more detail in Sec. 3.3.1. The cutoff for the contravariant ones determines the sparsity of the overlap and Hamiltonian matrix, whereas the cutoff for the covariant ones is related to the sparsity of the density kernel.
However, if the support functions are orthonormal, the contravariant and covariant quantities are identical and can thus be characterized by one single cutoff radius. Thus it is also reasonable to assume that the density kernel exhibits the same sparsity as the Hamiltonian and overlap matrix. As already mentioned several times, the support functions can not be exactly orthogonal due to the strict localization constraint, but the deviations of the overlap matrix from the identity are so small that it is well justified to use only one single cutoff radius, even though the introduction of a separate parameter for the density kernel might still be an option for the future in order to increase the flexibility.

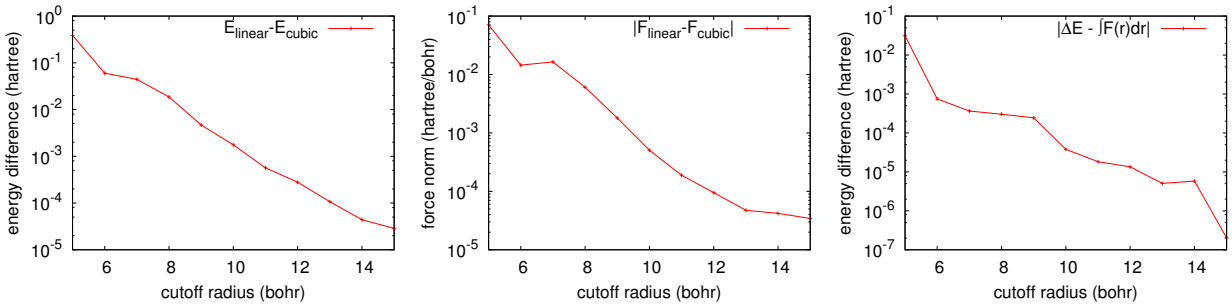
For the case of the Fermi Operator Expansion, there is – as briefly mentioned in Sec. 5.2.3.1 – however still a second cutoff radius. This parameter determines at which

distance the vectors are cut while building the expansion of the density kernel. As will be shown in Sec. 6.2.1.2, the influence of this cutoff radius is much smaller compared to the one for the support functions.

6.2.1.1 Cutoff radius for the support functions

To investigate the convergence properties with respect to the cutoff radius for the support functions, an alkane consisting of 92 atoms was chosen. This is not a large amount, but due to the chain-like geometry a finite cutoff still has a big influence; furthermore it is still small enough for the cubic version. The runs were performed using the hybrid mode for the optimization of the support functions and the FOE approach for the determination of the density kernel. The initial prefactor for the confinement was chosen such that the confining potential had a value of 20 hartree at the edges of the localization region and the cutoff radius for the construction of the density kernel was set to 20 bohr. The cutoff radius for the support functions was then varied in the range from 5 bohr to 15 bohr.

The results for all three quantities that were investigated – i.e. the energy, the forces and the consistency between energy and forces – are shown in Fig. 6.5. The energy differences between the runs using the linear version and the one using the cubic version, $E_{\text{linear}} - E_{\text{cubic}}$, are shown in Fig. 6.5a. The general trend exhibits a clear exponential convergence with respect to the cutoff radius; however the line has a slight kink at a



(a) The energy difference between a run with the linear version and the cubic reference version, $E_{\text{linear}} - E_{\text{cubic}}$.

(b) The norm of the difference vector between the forces of a run with the linear version and the cubic reference version, $|\mathbf{F}_{\text{linear}} - \mathbf{F}_{\text{cubic}}|$.

(c) The consistency between the energy difference and the corresponding path integral, $|\Delta E - \int \mathbf{F}(\mathbf{r})\, \mathrm{d}\mathbf{r}|$, for the linear version.

Figure 6.5: A comparison between the linear and the cubic version for various quantities as a function of the cutoff radius for the support functions. The test system was an alkane consisting of 92 atoms. An exponential convergence with respect to the cutoff radius is obvious. The kinks which are present at 6 bohr for the energy and the forces are due to a later breakdown of the support function optimization for this cutoff radius.

cutoff of 6 bohr. It turned out that this is due to the fact that for some reason the orthogonality problem was showing up much later for this cutoff radius and the support functions could consequently be optimized better, resulting in a slightly lower energy than expected for this cutoff.

Fig. 6.5b shows the norm of the differences in the forces, $|\mathbf{F}_{\text{linear}} - \mathbf{F}_{\text{cubic}}|$. Again an exponential convergence can be observed as a general trend, but the kink at 6 bohr is this time much more pronounced, indicating that the forces are more sensitive to the quality of the support functions than the energy. In addition there is a slight flattening of the curve in the region of the larger cutoff radii. The noise level for the chosen parameters was of the order of $10^{-6}\,\text{hartree}/\text{bohr}$.

Last but not least Fig. 6.5c shows the consistency between the energy and the forces, i.e. the value of $|\Delta E - \int \mathbf{F}(\mathbf{r})\,\mathrm{d}\mathbf{r}|$, as explained in Sec. 6.1.2. Since both the energy and the forces exhibit an exponential convergence, it is not surprising that the same also applies to this quantity. This curve is a bit less straight than the other two, but still the trend of an exponential decay is obvious.

To conclude these results show on the one hand that the linear version can reproduce the results of the cubic reference in the limit of large cutoff radii, and on the other hand they demonstrate the enormous influence of the cutoff radius on the accuracy of the calculation.

6.2.1.2 Cutoff radius for the Fermi Operator Expansion

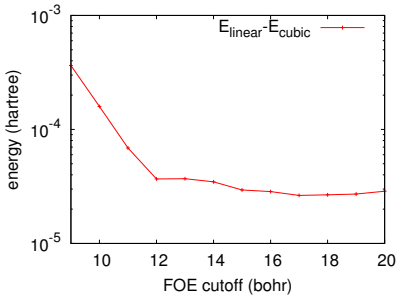
As explained in more detail in Sec. 5.2.3 the Fermi Operator Expansion uses a cutoff radius for the calculation of the matrix vector multiplications which finally build up the density kernel. The smaller this cutoff is, the less accurate the matrix vector multiplications are performed.
Therefore it is to be expected that this cutoff affects the accuracy of the FOE method and consequently also the overall results of the calculation.

To investigate the effect of this cutoff radius, several runs with different cutoff radii were performed for the same test system, namely the alkane consisting of 92 atoms. The cutoff radius for the support functions was set to 15 bohr, which is a very large value and should – according to the results of the previous section – not affect the accuracy considerably. The other parameters were chosen identical to the ones used for the runs in the previous section. The value of the FOE cutoff was varied from 9 bohr to 20 bohr.

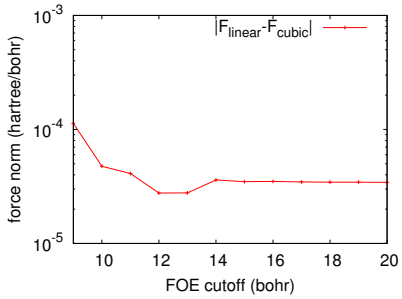
The results are shown in Fig. 6.6; this time only the energy and the forces were inves-

tigated. It is obvious that the influence of this cutoff radius is much smaller than that of the one for the support functions.
Fig. 6.6a shows the energy difference between the linear runs and the cubic reference calculation; the curve shows a nice exponential decay in the beginning, but then starts to saturate at a cutoff of about 12 bohr.
The situation for the norm of the force difference between the linear runs and the cubic one is similar, as can be seen from Fig. 6.6b. Again the decay of the curve stagnates at a cutoff of 12 bohr while being exponential for smaller values.

To conclude it seems that the choice of the cutoff radius for the FOE method has – at least if it is chosen large enough to reach the zone where the curves are flat – only little influence on the accuracy. Increasing it further beyond the start of the plateau will only make the calculation more costly without giving any improvement of the results.



(a) The energy difference between a run with the linear version and the cubic reference version, $E_{\text{linear}} - E_{\text{cubic}}$.



(b) The norm of the difference vector of the forces of a run with the linear version and the cubic reference version, $|\mathbf{F}_{\text{linear}} - \mathbf{F}_{\text{cubic}}|$.

Figure 6.6: Convergence of energy and forces as a function of the cutoff radius used for the kernel construction in the FOE method. The test system was the same as in Fig. 6.5. The cutoff radius for the support functions was set to the large value of 15 bohr such that it should not affect the results considerably. Both the energy and the force exhibit an exponential decay in the beginning, but then saturate quite fast at around 12 bohr.

6.2.2 Runtime as a function of the cutoff radius for the support functions

Increasing the cutoff radius for the support functions has two effects. On the one hand it simply enlarges the volume of the localization regions, thus requiring more work to be done for one support function within its own localization region, on the other hand it also increases the overlaps among the support functions and consequently affects the

operations which are related to this part.
Whereas the first effect is independent of the system under investigation since the shape of the support functions is always the same and thus the volume will increase as the third power of the cutoff radius, the second effect depends heavily on the geometry of the configuration. For a system which extends only along one dimension the number of overlaps will obviously increase much more slowly than for a configuration that extends also in the other dimensions.

As an illustration the number of overlaps among the support functions – i.e. the number of non-zero elements in the overlap matrix – was calculated as a function of the cutoff radius. Three different systems were taken for this test: An alkane consisting of 602 atoms, a graphene sheet with hydrogenated dangling bonds, amounting to 572 atoms in total, and a water droplet containing 600 atoms. In this way the test set contained a one-dimensional, a two-dimensional and a three-dimensional system.
The support functions were optimized using the hybrid mode and the FOE approach was employed for the determination of the density kernel. Whereas this is not important from the viewpoint of the number of overlaps among the support functions, it will be of great importance later on when the timings will be compared, since the FOE method depends heavily on the sparsity of the Hamiltonian matrix.

The results of this test are shown in Fig. 6.7. In order to be able to compare the three runs, the numbers of overlaps were all scaled down to the value 1 for the smallest cutoff radius. As expected the increase of the number of overlaps is strongly correlated with the dimensionality of the system. The three-dimensional water droplet exhibits the fastest increase, followed by the two-dimensional graphene sheet and the one-dimensional alkane.

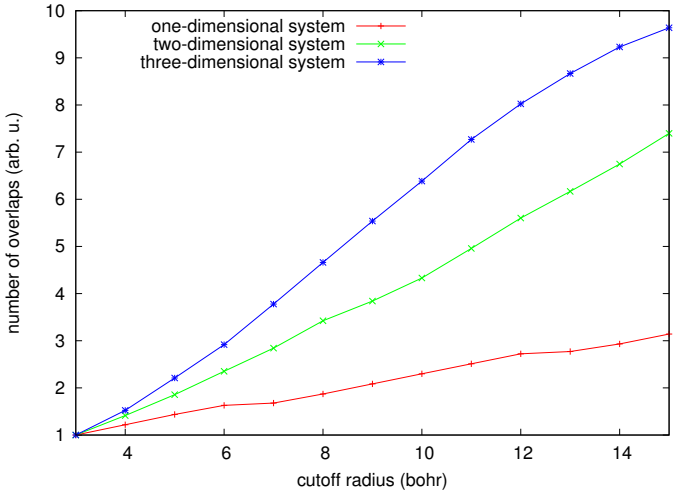


Figure 6.7: The number of overlaps among the support functions as a function of their cutoff radius. The numbers are all scaled down to 1 for the smallest cutoff radius in order to allow a comparison. Obviously the overlaps increase much more rapidly the higher the dimensionality of the system is.

Now the question is whether this strong dependence on the geometry is as well present in the total runtime – meaning that the operations related to the overlaps of the support functions are dominating – or whether there is an overall cubic scaling – independent of the geometry – stemming from the increase of the localization regions alone.
It has to be noted that there was a slight difference in the input parameters between the three systems, namely the choice of the mixing parameter – and consequently the number of iterations in the kernel loop – which had to be adjusted due to the different HOMO-LUMO gaps. Thus there might be a slight variation in the weights the various operations – e.g. support function optimization versus density kernel optimization – exhibit, but this should not affect the general statement.

The above question in answered in Fig. 6.8, where the total runtime for the same three systems is plotted as a function of the cutoff radius. In order to allow a fair comparison, the timings were again scaled down such that they exhibit the value 1 for the smallest cutoff radius. Furthermore the number of iterations in the outer loop was limited to 10 such that potential convergence problem – which might appear in particular for the small radii – cannot spoil the comparison.
As can be seen the increase of the timings is very different for the three systems and again ordered according to their dimensionality. This demonstrates that the runtimes are much more influenced by those parts depending on the overlaps of the support functions among each other than by those which depend only on the support functions inside their localization regions alone.

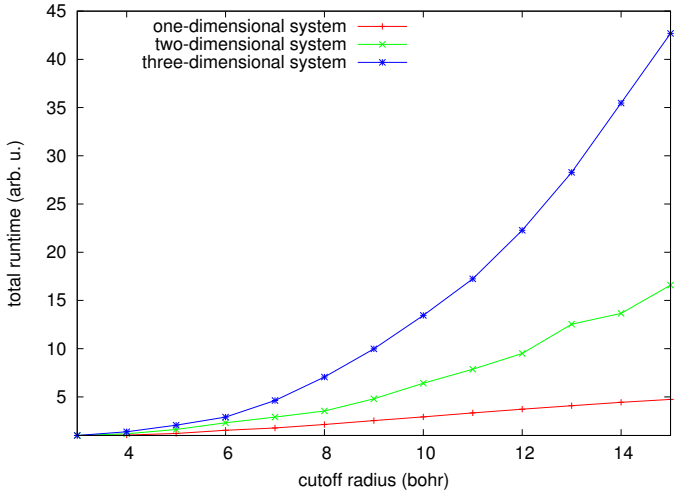


Figure 6.8: The total runtime for the three systems already used in Fig. 6.7 as a function of the cutoff radius of the support functions. In order to allow a comparison, the times were scaled down to 1 for the smallest cutoff radius. It is obvious how the geometry of the systems affects the results.

6.3 Parallelization

Treating hundreds or even thousands of atoms at the level of Density Functional Theory is a very demanding task. Even with an algorithm that only scales linearly with respect to the size of the system, the computational cost remains tremendous. Consequently the calculations are only feasible within a reasonable time frame if the overall workload can be split up among many processors.
Thus an efficient parallelization scheme is of utmost importance if the code is supposed to exploit the massive parallelism offered by nowadays supercomputers.

BigDFT exhibits two level of parallelization, namely distributed memory parallelization using MPI [83] and shared memory parallelization using OpenMP [84]. In addition some parts can exploit the massive parallelism offered by GPUs [85], however this is – at least for the moment – not relevant for the linear scaling version.
Whereas OpenMP can rather easily be added on top of any distributed memory parallelization, the MPI parallelization has to be planned carefully.

6.3.1 MPI parallelization

Since MPI is a shared memory programming model, one first has to think about how the data should be distributed among the various MPI tasks. At the moment there are basically two main quantities that are distributed, namely the support functions and the charge density or potential.

With respect to the support functions, it is a natural choice to simply split up the total number of support functions among the MPI tasks, i.e. each task only handles a few ones. Since the support functions are all quite similar in size, this should lead to an efficient parallelization.
On the other hand, the charge density – or the potential, respectively – has to be cut into pieces in order to be distributed. Since the charge density is stored in an orthorhombic box, the most convenient way is to split it up in planes and to distribute them among the MPI tasks. The axis for this distribution is chosen to be the z direction, thus the parallelization is most efficient if the system has the largest extent along this dimension.

As soon as the data distribution has been settled, the question of how to parallelize the computation can be addressed. Some operations related to the support functions can be done completely independent of each other – i.e. no communication among the various MPI tasks is required –, in this way exhibiting a natural parallelization. Examples are the application of the kinetic energy operator, the evaluation of the pseudopotential

part or the preconditioning.
However there are also some operations – both with respect to the support functions and the charge density – which require some communication among the MPI tasks and which can become a bottleneck if this communication is not done in an efficient way. Some of these problems will be discussed in the following.

6.3.1.1 Calculation of scalar products

One of the most common operations that requires communication among the various MPI tasks is the calculation of scalar products, for instance to build the overlap matrix:

$$S^{\alpha\beta} = \int \phi^{\alpha}(\mathbf{r})\phi^{\beta}(\mathbf{r})\,\mathrm{d}\mathbf{r}. \tag{6.3}$$

Since the support functions are distributed among the MPI tasks, it is obvious that they must be communicated in some way in order perform this operation.

The most obvious way would be to directly exchange in a point-to-point fashion those parts of the support functions which overlap with each other. Afterwards the scalar products can be calculated locally on each single MPI task.
Assuming, for simplicity, that each MPI task handles one support function, then this MPI task has to calculate one line of the overlap matrix. For instance, if MPI task 0 holds the support function number 1, then it has to calculate the first line and thus to determine the matrix elements $\int \phi_1(\mathbf{r})\phi_j(\mathbf{r})\,\mathrm{d}\mathbf{r}$ for all values of j. If the sparsity of the matrix is known, some matrix elements are zero and need not be calculated, but in general there still remain quite a lot which are non-zero.
In order to calculate all these integrals, task number 0 has to receive those parts of all the other support functions with which support function 1 overlaps. Though this is conceptually a straightforward approach, it has several severe drawbacks.

First of all the amount of data that has to be communicated is tremendous since the support functions have in general quite a notable overlap. This will also result in a very poor ratio between computation and communication; if each MPI task handles only one support function, then each element that is communicated is only used in one single operation.
Furthermore there is an enormous load unbalancing since support functions that are localized in the center of the system have in general a larger number of overlaps with other support functions than those which are localized at the boundary of the simulation cell, at least for free boundary conditions where no periodic images have to be considered.
Last but not least the large amount of data required by this point-to-point communication scheme is split up in a huge number of small messages, which could result in a

large overhead due to the latency of the network.

Some of these problems are illustrated in Fig. 6.9 for the case of a water droplet consisting of 300 atoms. The plot shows – as a function of the cutoff radius – the overall amount of data that has to be communicated and the total number of point-to-point messages that are required for one single calculation of the overlap matrix. The total number of MPI tasks used for this test was equal to the number of support functions such that each MPI task had to handle exactly one support function; this amounted to totally 600 MPI tasks.
It is obvious that the total amount of data that has to be communicated with this point-to-point approach is enormous. For instance, for a cutoff radius of 9 bohr – which is a typical value used in a practical application – roughly 8.96 GB have to communicated. This has to be compared with the total size of all support functions, which amounts to only 0.20 GB. So the total amount of data that needs to be communicated is about 45 times larger than the data itself.
Furthermore it can be seen that the total number of messages that have to be sent in total is indeed huge. Again specifically looking at a cutoff of 9 bohr, they amount to 142'842. It is very likely that such a large number will stress the network.

Due to all these problems it was necessary to develop a different approach. The basic idea behind it is to use a data layout where the support functions are "transposed".

For reasons of simplicity this layout is first briefly described for the case where no localization constraints are present, i.e. all support functions extend over the entire simulation box. In such a situation the support functions are represented by vectors containing the expansion coefficients with respect to the underlying wavelet basis and

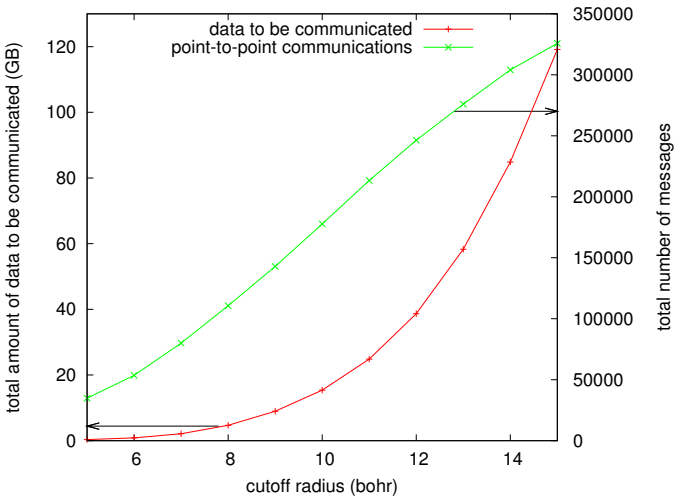


Figure 6.9: Illustration of the problems related to the point-to-point communication scheme. The test system was a water droplet consisting of 300 atoms; the number of MPI tasks used was equal to the number of support functions, namely 600. The total amount of data to be communicated (red curve, left axis) and the total number of point-to-point communications (green curve, right axis) are shown with respect to the cutoff radius. Both numbers are huge in view of the rather small dimensions of the system.

have – due to the absence of the localization constraint – all the same length.
In the standard data distribution, each MPI task holds a few of these vectors. In the transposed scheme, each MPI task holds only a subset of the coefficients, but in turn from all support functions. If the vectors representing the support functions were collected together to form a matrix, this new layout would correspond to a transposition of the matrix.
An illustration of the two layouts is given in Fig. 6.10 for the case of four MPI tasks and four support functions.

If the support functions are available in this transposed layout, each MPI task can calculate the partial scalar products among all support functions locally and in this way build up a partial overlap matrix. Afterwards these partial overlap matrices have to be summed up among all MPI tasks. Since all support functions extend over the entire simulation box, each support function overlaps with all the other ones and as a consequence the number of scalar products is the same on each MPI task, leading to a perfect load balancing.

In the general case where each support function is strictly localized not only the length of the vectors holding the support functions is slightly different, but also the number of overlaps per support function is not always the same. Therefore the simple transposition scheme is not applicable anymore.
In order to carry over the concept to this case the transposed layout has to be viewed from a different perspective. Instead of thinking of a matrix transposition, this layout can be seen as a distribution where each MPI task is responsible for a given region of the global simulation box and gets from all support functions those parts extending into this region.

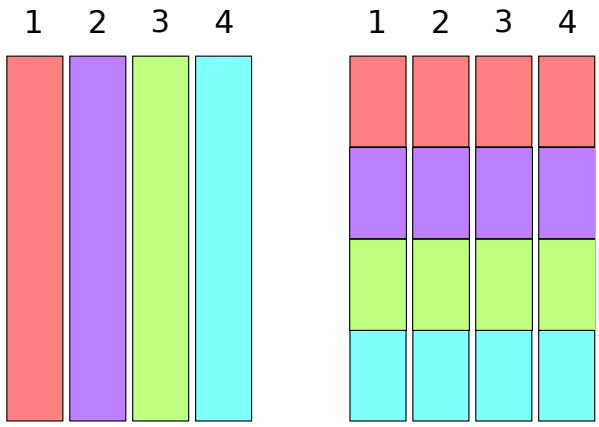


Figure 6.10: Illustration of the data layouts for the case where all support functions extend over the entire simulation cell, i.e. each support function can be written as a vector of some length N. On the left-hand side the standard data distribution is shown where – in this case – each MPI task holds one support function, i.e. one entire vector of length N. On the right-hand side the transposed data layout is shown where each MPI task holds components of all support functions, but in turn only a subset of them, namely N/n_{task}, where n_{task} is the total number of MPI tasks.

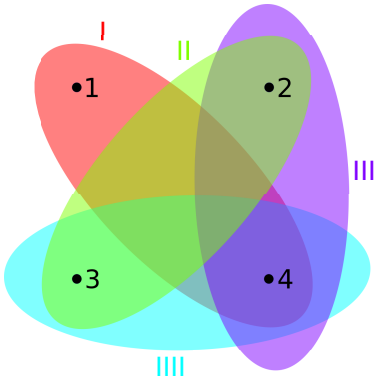
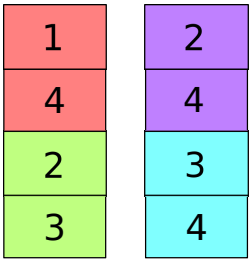


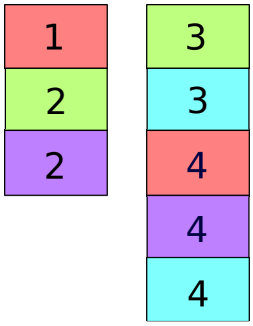
Figure 6.11: A simple example which will be used to illustrate the optimal transposed layout of the support functions. The system consists of four grid points labeled by Arabic numerals and four support functions labeled by Roman numerals. The support functions are constructed such that they all extend over two grid points.

This picture can now be transfered more easily to the general case. The entire simulation box has to be partitioned among all MPI tasks, and the support functions are then distributed to the various MPI tasks such that each one can calculate a partial overlap matrix for its region. This means that each MPI task has to receive those parts of all support functions which extend into that region of the simulation box for which the MPI task is responsible. These partial matrices are then again summed up to build the entire overlap matrix.
This partitioning of the box has to be done such that the load balancing among the MPI tasks is optimal, which in general does not correspond to a uniform distribution of the global simulation box.

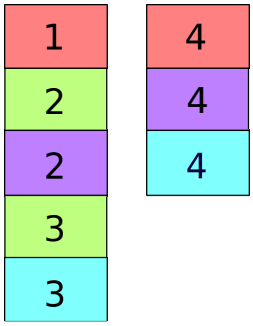
The exact procedure can most easily be explained with a small example, which is shown in Fig. 6.11. For simplicity the system consists of only four grid points (denoted by Arabic numerals) and four support functions (denoted by Roman numerals) which



(a) The standard layout where each MPI task has the entire data of some support functions.



(b) The naive transposed layout where each MPI tasks has all the data for some grid points.



(c) Optimized transposed layout similar to 6.12b, but this time with an optimal load balancing.

Figure 6.12: Schematic view of the different data layouts for the support functions of the simple example shown in Fig. 6.11. Fig. 6.12a is the standard layout, whereas Figs. 6.12b and 6.12c are transposed layouts.

all extend over two grid points.
The various data layouts that emerge from this small example are shown in Fig. 6.12. If there are two MPI tasks, each of them will handle two support functions in the standard data layout, i.e. task 0 will handle the support functions I and II and task 1 the support functions III and IIII. This is illustrated in Fig. 6.12a.

In order to build the transposed layout, the most naive implementation would simply split up the four grid points among the two MPI tasks, i.e. the first one would handle the grid points 1 and 2 and the second one the grid points 3 and 4. This approach is illustrated in Fig. 6.12b.

However this is not the optimal distribution from the viewpoint of the total workload per MPI task. In order to calculate the partial overlap matrix each MPI task has to iterate through all grid points it handles and perform multiplications among all support functions touching a given grid point.
If the transposed layout was constructed as described, this would mean that task 0 has to perform one multiplication for grid point 1 and four multiplications for grid point 2, whereas task 1 has to perform four multiplications for grid point 3 and nine for grid point 4. As a consequence there is an enormous load unbalancing of totally 5 versus 13 multiplications.

A much better solution, which will give an optimal load balancing, is illustrated in Fig. 6.12c. In this layout task 0 handles the grid points 1, 2 and 3, whereas task 1 only handles the grid point number 4. In this way both MPI tasks have to perform 9 multiplications.

In order to determine the shape of this optimized layout, a weight is assigned to each grid point, given by the square of the number of orbitals touching it. For the simple example these weights would consequently have the values one, four, four and nine, respectively. The total weight which is given by the sum of all partial weights is then split up among the MPI tasks as evenly as possible. For the simple test case this means that each process gets assigned a total weight of nine. Now the grid points can finally be assigned to the MPI tasks such that the total number of operations induced by the partitioning comes as close as possible to the target weight.
Since the calculation of the partial overlap matrices requires that all support functions for a given grid point are handled by the same MPI task, it is not possible to split up grid points. Consequently it will in general not be possible to exactly reach the target weight. However, since there are usually much more grid points than MPI tasks, these deviations from the optimal load balancing will remain small.

A demonstration of the superiority of the transposition scheme over the point-to-point approach from the viewpoint of the load balancing is given in Fig. 6.13. The plot shows

the minimal and maximal number of multiplications that have to be performed by a single MPI task in order to calculate the overlap matrix, again for the same system as in Fig. 6.9. For the point-to-point approach each MPI task has to calculate one line of the final overlap matrix, whereas for the transposition approach each task has to calculate an entire partial overlap matrix. Of course summing up the total number of multiplications among all MPI tasks gives the same result for both approaches.
The closer these two lines representing the minimal and maximal number of multiplications come to the optimal value – which is given by the total number of multiplications divided by the number of MPI tasks – the better the load balancing is. It is obvious that the transposition approach outperforms the point-to-point approach by far. For the typical cutoff radius of 9 bohr, the ratio between the maximum and the minimum is 4.11 for the point-to-point approach, whereas for the transposition approach it is only 1.11. To estimate the additional runtime caused by the load unbalancing, it is also interesting to compare the maximal value with the optimal one. For the point-to-point approach this ratio is 1.83, whereas for the transposed approach it is only 1.05.

The next observation is that calculating the scalar products using the transposed layouts requires considerably less data to be communicated than the point-to-point fashion.
Since the transposed layout is just a redistribution of the standard layout, the total amount of data that has to be communicated is always equal to the total size of all support functions. In general this is much less data compared to the amount that is communicated in the point-to-point approach, where a lot of data has to be duplicated. A direct comparison of the total amount of data that has to be communicated for the point-to-point and the transposition approach is shown in Fig. 6.14. Again the same system and number of MPI tasks as before were used. It is obvious that for the trans-

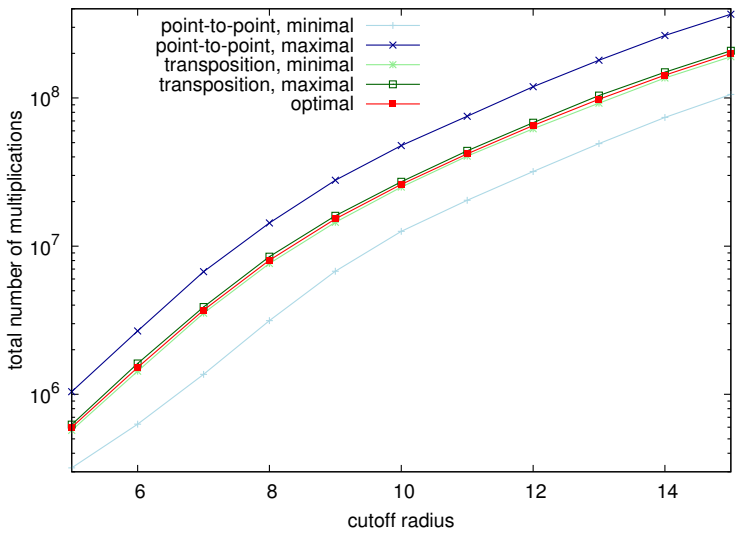


Figure 6.13: Minimal and maximal number of multiplications to be done by a single MPI task in order to calculate the overlap matrix among the support functions, together with the optimal value which is given by the total number of multiplications divided by the number of MPI tasks. The spread for the point-to-point approach is much larger than for the transposition approach, which comes close to the optimal value and thus exhibits a good load balancing.

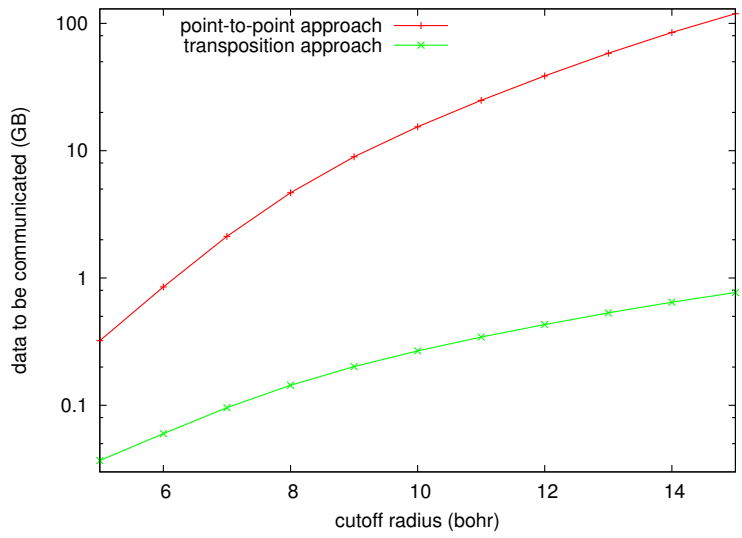


Figure 6.14: Comparison of the total amount of data that has to be communicated for the point-to-point approach and the transposition approach. The amount of data is smaller by orders of magnitude for the transposition approach. The test system was again identical to the one which was used before and whose other benchmarks are shown in Figs. 6.9 and 6.13.

position approach the amount of data that has to be communicated is much smaller.

Furthermore the transposition approach has the advantage that the communication can be done in a much more efficient way. After some local rearrangement of the data for each MPI task, it can in principle be communicated with one single MPI call (MPI_Alltoallv); in practice there are actually two since the coarse and fine parts are handled separately. After the data has been received again some local rearrangement is required in order to reach the correct data layout. These three steps – local rearrangement for the communication, the communication itself and the local rearrangement to

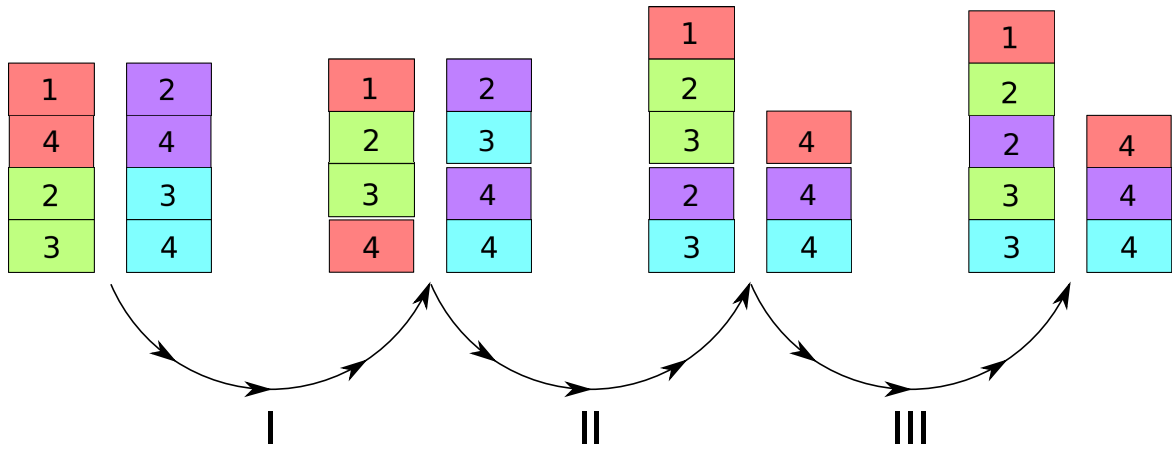


Figure 6.15: Illustration of the transposition process for the small test example of Fig. 6.11. In step I, the data is rearranged locally on each MPI task. Afterwards it can be communicated using one single collective call (MPI_Alltoallv), as shown in step II. Finally, in step III, it has again to be rearranged locally in order to reach the final layout.

get the correct layout – are illustrated in Fig. 6.15 again for the small example.
Due to the latency of the network, two MPI calls will most likely be more efficient than the tens of thousands of small messages that have to be sent around for the point-to-point approach.
For the latter case such a communication in one single step is not possible straightforwardly since a given data point is in general sent to more than one MPI task. If such a scheme was wanted, one would have to use large buffers where the data is often duplicated, thus leading to a large memory overhead.

In summary it is clear that this transposition approach eliminates all problems that arise with the point-to-point approach. Again looking specifically at the water droplet consisting of 300 atoms and a cutoff radius of 9 bohr the numbers for a run using 600 MPI tasks are impressive:

- The amount of data that is communicated is reduced by a factor of 45.
- The number of MPI calls is reduced from 142'842 to 2.
- The load unbalancing is reduced from 1.83 to 1.05.

Due to these numbers it is not surprising that the calculation of scalar products can be done much more efficiently using the transposition approach.

6.3.1.2 Calculation of the charge density

The calculation of the charge density is another task that requires communication among the various MPI tasks. The formula for its calculation is – according to Eq. (3.28) – given by

$$\rho(\mathbf{r}) = \sum_{\alpha,\beta} \phi^{\alpha}(\mathbf{r}) K_{\alpha\beta} \phi^{\beta}(\mathbf{r}). \tag{6.4}$$

This looks formally very similar to the calculation of the overlap matrix, since again the product of different support functions has to be computed.
However there are also a few differences. Whereas the overlap matrix is calculated with the support functions being stored in the compressed scaling function / wavelet basis, the charge density is calculated with the support functions given in a dense representation in a orthorhombic box. Furthermore the calculated charged density must match the shape of the Poisson Solver [86,87], which is parallelized in planes along the z axis.

This latter requirement was the reason to first implement the calculation of the charge density in such a way that each MPI task directly calculates the charge density of those planes that it will use later for the Poisson Solver. This has the advantage that there is

no more communication requirement after the calculation of the charge density.
Recalling that the transposition approach discussed previously can be seen as a partitioning of the simulation box among the MPI tasks, it becomes clear that this way of calculating the charge density is already such a transposition method.
Thus its performance from the viewpoint of the total amount of data that has to be communicated is quite advantageous. Since the overall simulation box is split up in disjoint planes, each element of a given support function has to be sent to exactly one MPI task, as illustrated in Fig. 6.16. Consequently the total amount of data that has to be communicated is equal to the total size of all support functions, in agreement with the discussion in Sec. 6.3.1.1.

However it turned out that this scheme has several shortcomings from the viewpoint of the load balancing.
First of all it might well happen that there are more MPI tasks than planes in the z direction. As a consequence some MPI tasks will be idle, leaving more work for the remaining ones.
In addition, even if the planes can be well distributed among the MPI tasks, the load balancing will still be very poor. This is simply due to the fact that the number of support functions extending to the planes at the edges of the simulation box is very small and as a consequence the sum in Eq. (6.4) runs only over a few terms in those regions. This is in strong contrast to the planes in the center of the simulation box, where many support functions overlap and the sum runs consequently over many terms.

Due to these reasons it turned out that it is more advantageous to perform the calculation of the charge density along the same lines as the calculation of the overlap matrix, i.e. to give up the strict partitioning into planes and to split the entire simulation box such that the load balancing is optimal. In this way it is also guaranteed that there are no more MPI tasks being idle.

The way the simulation box is partitioned among the MPI tasks is completely analogous to the case of the overlap matrix, i.e. on assigns to each grid point a weight which is given by the square of the number of support functions touching this grid point

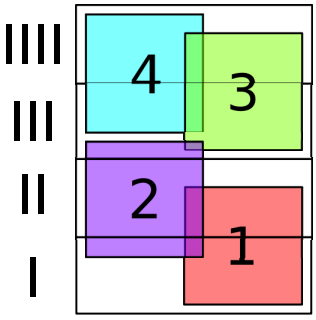


Figure 6.16: Illustration how the support functions (rectangles with Arabic numerals) must be distributed to the MPI tasks which handle the planes (underlying rectangles with Roman numerals). Since the planes are disjoint, each element of a given support functions has to be sent to exactly one MPI task. Consequently the total amount of data that has to be communicated is equal to the size of all support functions.

and then distributes the grid points among the MPI tasks such that the total weight is distributed evenly.

Since this way of partitioning the simulation box gives a different layout than that required by the Poisson Solver, an additional communication step is needed after the calculation of the charge density.
However, since only a redistribution of the charge density is required, the total amount of data to be communicated equals the size of this quantity and is thus rather small compared to the amount that had to be communicated before in order to distribute the support functions. Furthermore the communication can again be accomplished by one single MPI call. Due to these reasons it is not to be expected that this additional communication will create problems.

A comparison of the two ways of partitioning the simulation box is given in Fig. 6.17. The plot shows the minimal and maximal number of operations per MPI task for both approaches as a function of the cutoff radius; an operation corresponds to the calculation of one element in the sum in Eq. (6.4), i.e. two multiplications.
As test system again the water droplet consisting of 300 atoms was used. With the chosen grid spacing of 0.28 bohr this amounted to 431 planes in the z direction. Since 600 MPI tasks were used – which is reasonable since there are 600 support functions – some MPI tasks did not participate in the calculation of the charge density for the simple partitioning. Therefore the minimum number of operations per MPI task is zero for this approach and cannot be displayed in the plot due to the use of a logarithmic scale.

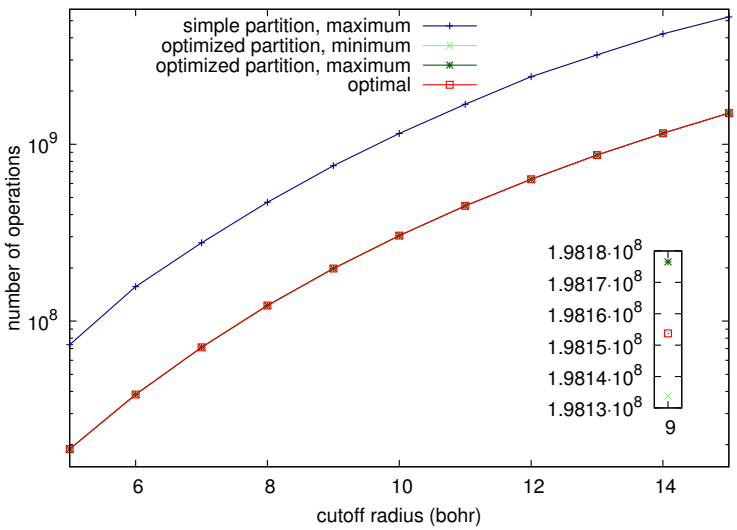


Figure 6.17: Minimal and maximal number of operations to be done by a single MPI task in order to calculate the charge density, together with the optimal value which is given by the total number of operations divided by the number of MPI tasks. The minimal value for the simple partitioning is zero and cannot be displayed on the logarithmic scale. For the optimized partitioning the minimal and maximal value almost coincide with the optimal value and are hardly visible. The inset shows these three values for a cutoff of 9 bohr on a non-logarithmic scale.

However it becomes clear from the figure that the maximal number is much larger than the optimal value which is given by the total number of operations divided by the number of MPI tasks. The ratio between the maximum and the optimum is close to four, independent of the cutoff radius. Again looking specifically at the value for a cutoff radius of 9 bohr, the maximal number is 3.81 times larger then the optimal value. On the other hand the situation looks much better for the optimized partitioning. Here the minimal and maximal values almost coincide with the optimal one and the three curves are basically indistinguishable. A small inset shows the three values for a cutoff radius of 9 bohr, where it becomes clear how excellent the load balancing is. The minimal value is at 0.9999 of the optimal value and the maximum at 1.0001.

The reason why the load balancing is much better compared to the case of the calculation of the overlap matrix which was shown in Fig. 6.13 is simply the much larger number of grid points – the support functions are given on a grid having half the grid spacing compared to the original one and in a cube instead of a sphere – which allows a better distribution of the total weight.

6.3.1.3 Linear combinations

Building linear combinations among the support functions is another operation that requires to communicate them among the MPI tasks. One example is the Löwdin orthonormalization:

$$\tilde{\phi}^{\alpha}(\mathbf{r}) = \sum_{\beta} (S^{-1/2})^{\alpha\beta} \phi^{\beta}(\mathbf{r}). \tag{6.5}$$

The overlap matrix is already available on each MPI task; thus this prescription is formally again very similar to the calculation of scalar products since each support function needs to receive those parts of the other support functions with which it overlaps. Consequently again the same considerations as in Sec. 6.3.1.1 apply and it turns out that the transposition approach is much more efficient than the direct point-to-point approach.
Furthermore, since the two operations – for the specific case of the orthonormalization the computation of the overlap matrix and the calculation of the linear combinations – are often close together, the transposed layout can directly be reused.

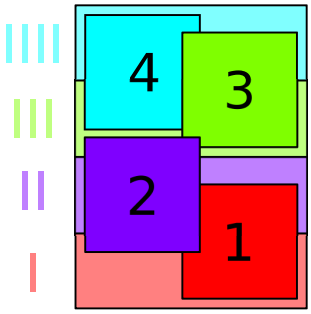
6.3.1.4 Gathering the potential to apply the Hamiltonian

As already mentioned the charge density and the potential are stored in an orthorhombic box being distributed in planes along the z axis in order to meet the parallelization of the Poisson Solver [86, 87].

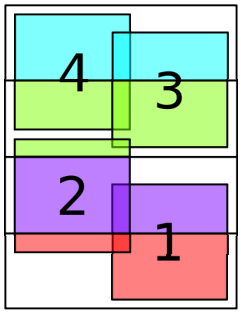
However, in order to apply the Hamiltonian onto a support function one needs the potential in a subbox that comprises this support function. If the support functions extended over the entire volume, then this subbox would be the identical to the global simulation box and the potential could be gathered using one single collective MPI call. This is the case for the cubic version, where the Kohn-Sham orbitals take over the role of the support functions.
However, for the linear scaling version, this subbox is in general only a small fraction of the global box and gathering the entire potential would be wasteful.

The situation is illustrated for a small example in Fig. 6.18. As is shown in Fig. 6.18a there are four planes which are distributed among four MPI tasks (Roman numerals) and four orbitals (Arabic numerals) that are as well distributed among the four MPI tasks.
The way the potential has to be gathered in order to be able to apply the Hamiltonian is shown in Fig. 6.18b. It can be seen that a given MPI task needs in general also parts of the potential being treated by other MPI tasks. Furthermore a given part of the potential is in general needed by more than one MPI tasks.
As a consequence the communication pattern is highly complicated and it is therefore the best solution to accomplish each communication step separately using a point-to-point scheme.

Since the potential that has to be communicated from one MPI task to another one is in general only a subblock of the potential on the MPI task from which it originates, it is not possible to directly communicate the entire array. Instead one either has to



(a) The potential distributed in planes (Roman numerals) and the four support functions (Arabic numerals).



(b) The parts of the total potential that are needed for the Hamiltonian application onto the support function.

Figure 6.18: Illustration of the data layout for the communication required to gather the potential for the application of the Hamiltonian. This small example consists of four support functions and four planes, both being distributed among four MPI tasks.

send each line of the subblock separately, thus leading to many small messages and therefore to a large overhead due to the latency, or to copy the subblock to a work array which can then be sent as one large array, in this way blowing up the memory requirements.

Fortunately there exists in addition a third way, namely the use of MPI derived data types [83]. In this way the entire block can be sent as one message without the need to copy it first to a work array. Due to the enormous saving this approach offers, it is the method of choice.

6.3.2 OpenMP parallelization

In contrast to MPI, OpenMP is a shared memory model, meaning that all threads within a team can access the same shared memory. Therefore one does not have to worry about the detailed data layout and communication bottlenecks, and OpenMP parallelization can often be added on top of an existing code without the need of a fundamental redesign.
OpenMP is frequently used in connection with the existing MPI parallelization, meaning that each MPI task is again parallelized over several OpenMP threads.

There are several reasons why an MPI task should be further split up in several OpenMP threads. One advantage is that the parallelism of modern supercomputers can be further exploited. Increasing the number of MPI tasks beyond the number of support functions will only give a moderate speedup, since the additional tasks will be idle for large sections of the program. On the other hand, by using several OpenMP threads for each MPI task, it is possible to go beyond this limitation and consequently exploit more cores, resulting in a considerable speedup of the calculation.

In addition, using several OpenMP threads is useful to avoid a potential loss of computing power due to memory limitations. As there are often memory requirements that can not be distributed among the MPI tasks – i.e. each MPI task has a given basic requirement – it is sometimes not possible to use as many MPI tasks as one has cores on a compute node since this would exceed the available memory. Thus one has to reduce the number of MPI tasks per node, thereby only using a fraction of the available cores. Here OpenMP can help: If each of the remaining MPI tasks spawns several OpenMP threads, it is again possible to exploit the full computing power of the node that would be lost otherwise.

Furthermore it can also be advantageous to use fewer MPI tasks and more OpenMP threads in order to reduce the overhead caused by the communication. In general

the performance of the MPI parallelization deteriorates as the number of MPI tasks is increased since the time taken by the communication becomes longer; consequently decreasing the number of MPI tasks and using more OpenMP threads can help in such situations.

The linear scaling version of BigDFT contains OpenMP statements in various places. Fig. 6.19 shows the speedup that can be gained by using several OpenMP threads for a various number of MPI tasks. The test system was a water droplet consisting of 300 atoms, in this way amounting to 600 support functions. The number of MPI tasks was varied from 75 to 300, and the number of OpenMP threads was ranging from 1 to 16. Due to the limited size of the supercomputer, a calculation using 300 MPI tasks and 16 OpenMP threads was not possible.

First of all it has to be noted that the sharp kink when going from 8 to 16 threads is caused by the architecture of the compute node. Consequently it is not really meaningful to conclude anything about the performance of the OpenMP parallelization from the values for 16 threads.
Furthermore it is evident that the OpenMP speedup is better the smaller the number of MPI tasks is. This is not surprising, since in these cases there is more work per MPI task which can be split up among the OpenMP threads. In addition the time spent for the communication is smaller if fewer MPI tasks are used, consequently increasing the ratio between computation and communication. For the most favorable case – the one using 75 MPI tasks – the speedup one gets by going from 1 to 8 threads is about 5.8, whereas for the worst case – the one using 300 MPI tasks – it is about 4.

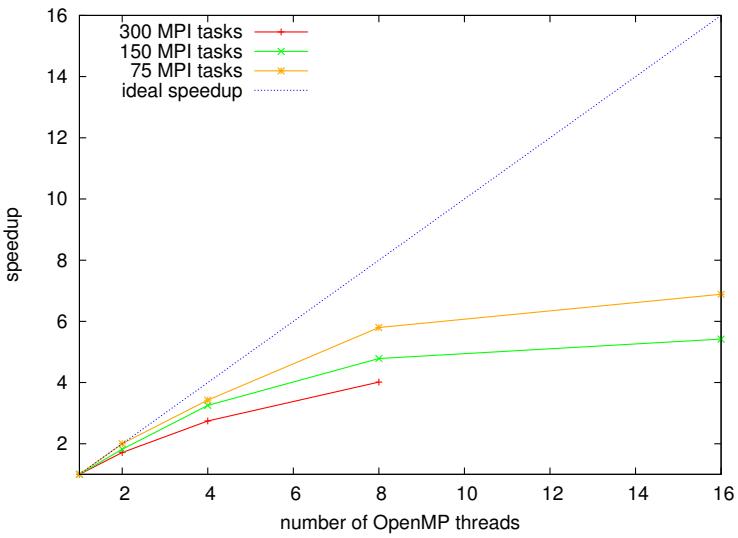


Figure 6.19: Speedup of the OpenMP parallelization for a water droplet consisting of 300 atoms, amounting to 600 support functions. The speedup is better for a smaller number of MPI tasks since in this way there is more work that can be split among the OpenMP threads. The sharp kink when going from 8 to 16 threads is due to the architecture of the compute node. Using 300 MPI tasks and 16 OpenMP threads was not possible since this would have surmounted the total number of cores on the supercomputer.

It is interesting to fit Amdahl's law [88] to the data points up to 8 threads. If the fraction of the code which can be parallelized is denoted by p, then Amdahl's law states that the maximal speedup that can be gained by using N cores in parallel is given by

$$S(N) = \frac{1}{(1-p) + \frac{p}{N}}. \tag{6.6}$$

In this way the data from Fig. 6.19 gives a non-parallel fraction of 14% for 300 MPI tasks, 9% for 150 MPI tasks and 5% for 75 MPI tasks. This non-parallel amount is mainly caused by the communication required for the MPI parallelization.

However it is astonishing that these numbers are smaller than the actual communication time required for the run with one thread. This means that even the communication time can be sped up by using more OpenMP threads.
This is a consequence of the limited injection bandwidth of the network – the amount of data that can be transfered from the compute node to the network within a certain time interval – over which the communication takes places. If only one thread is employed, then there are – for the architecture used for this test – 16 MPI tasks per node. If they all send their data at the same time, this operation will be limited by the injection bandwidth, leading to a slowdown of the communication. If, on the other hand, more threads are used, the number of MPI tasks per node is reduced, consequently increasing the available injection bandwidth per task and removing this bottleneck.

6.3.3 Scaling with the number of processors

As already mentioned it is of utmost importance that the code exhibits a good parallelization due to the enormous amount of work a DFT calculation requires. In order to measure the performance of the code as the number of cores is increased two possibilities exist:

- the so-called "strong scaling" indicates how the time to solution varies with the number of cores for a fixed problem size
- the so-called "weak scaling" indicates how the time to solution varies with the number of cores for a fixed problem size per core

Given an ideal parallelization, the strong scaling should lead to a runtime proportional to the inverse of the number of cores that are used, whereas the weak scaling should – assuming a perfect linear scaling with respect to the total size of the system – lead to a constant runtime.

6.3.3.1 Strong scaling

Measuring the strong scaling for a wide range of processors is a bit involved for technical reasons. The most difficult problem is the memory limitation. Whereas even for large systems the memory requirement is not a bottleneck if many cores are used since the data can be distributed over many compute nodes, it becomes problematic if the number of cores – and consequently also the number of compute nodes – is decreased. Using only a fraction of the available cores per node and in this way extending the calculation over more nodes would increase the total amount of memory that is available and thus be a workaround. However this would falsify the results by increasing the available memory bandwidth per core and therefore privilege those runs where a small number of cores is used.

For this reason the strong scaling for the chosen system – a water droplet consisting of 960 atoms – could only be measured from 160 to 3840 cores. Still this is a rather wide range and consequently a meaningful indication of the scaling properties of the code. The support functions were optimized using the hybrid mode and the FOE approach was employed for the optimization of the density kernel. The cutoff radii were set to 8 bohr and 12 bohr, respectively, and the initial prefactor for the confinement to $4.9 \cdot 10^{-3}\,\mathrm{hartree/bohr^4}$.
The speedup that resulted by varying the numbers of cores – defined as $s(N) = t_{160}/t_N$, where t_x is the runtime for x cores – is shown in Fig. 6.20. The test was performed for 2 and 4 OpenMP threads, meaning that the number of MPI tasks was ranging from 80 to 1920 and 40 to 960, respectively.
The runs using 4 OpenMP threads, which performed slightly better in this test than those using 2 threads, gave a speedup of about 15 by going from 160 to 3840 cores.

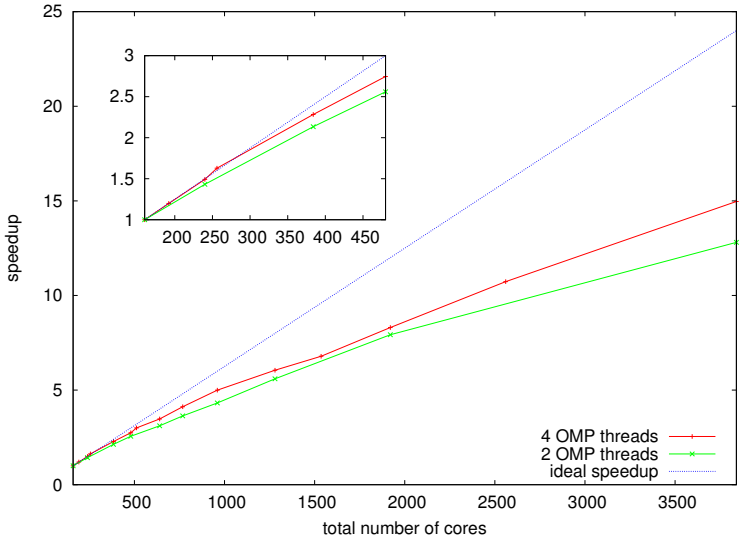


Figure 6.20: Plot of the effective speedup for a water droplet consisting of 960 atoms, together with the ideal speedup. The smallest number of cores used was 160, the largest 3840. A smaller number of cores was not possible due to memory limitations. Even for this large number of cores, a speedup of about 15 could be reached, which corresponds to a an efficiency of 62%. As shown by the inset, the scaling for the smaller values is even better.

Comparing this with the ideal value of 24 one gets an efficiency of roughly 62%. This is an excellent value considering the large number of cores used.
Furthermore it is obvious that the speedup is almost perfect for a small number of cores, as shown by the small inset in the figure – from 160 cores to 480 cores the speedup is roughly 2.75, i.e. about 92% of the ideal value. Since it is to be expected that the scaling towards a very small number of cores is rather getting better than worse, it is a fair assumption that one would get at least 90% efficiency by going from 1 to 160 cores. This would then result in an overall speedup of more than 2000 by going from 1 to 3840 cores, which demonstrates that the code exhibits an excellent parallelization.

Another important issue is the memory usage as a function of the numbers of cores. The memory is in general not perfectly balanced among all MPI tasks and usually the master task exhibits a memory requirement which lies above the average value. This is due to the fact that in the transposed layout this task handles a region at the border of the simulation box. Since the number of overlaps among the support functions is rather small in this region, the master task has to cover a portion of the simulation box which is larger than the average in order to reach the optimal load balancing, as was explained in more detail in Sec. 6.3.1.1.
The memory peak for the MPI master task – again for the same system as the timings – is shown in Fig. 6.21. As expected the memory usage decreases as the number of cores is increased.

Since the memory requirements depend only on the number of MPI tasks and not

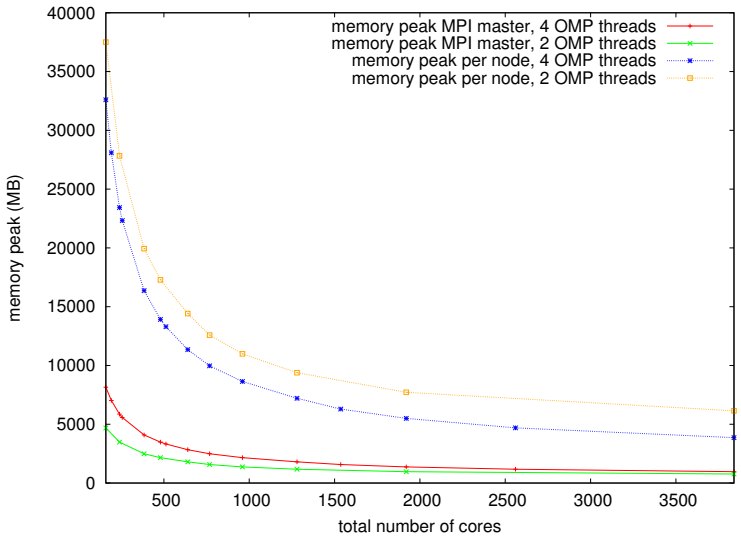


Figure 6.21: Memory peak for the MPI master task as a function of the total number of cores used, for the same system as the timings in Fig. 6.20. Since the memory usage depends only on the number of MPI tasks and not on the number of OpenMP threads, the memory requirements for a given number of cores is always larger for the run with 4 OpenMP threads. However – as indicated by the blue dotted line – the memory peak per node is always smaller for the runs with 4 OpenMP threads compared to the ones with 2 threads.

on the number of OpenMP threads, the memory peak for a given number of cores is always larger for a run with 4 OpenMP threads compared to another one with 2 OpenMP threads.
However the run with 4 OpenMP threads still uses the memory in a more efficient way. What is ultimately limiting is not the memory peak for one single MPI task, but the memory peak for one compute node. This value is indicated by the dotted lines in Fig. 6.21; for simplicity it was assumed that all MPI tasks on the compute node have the same memory requirement as the master task, which is in general too pessimistic. As can be seen the values for the runs with 4 OpenMP threads are always lower than those for the runs with 2 OpenMP threads. This behavior can be explained by the fact that there are some quantities which have to be stored by all MPI tasks; therefore reducing the number of MPI tasks per node by increasing the number of OpenMP threads diminishes the overall memory requirements. This demonstrates that OpenMP is a good option if the available memory is a bottleneck.

6.3.3.2 Weak scaling

Measuring the weak scaling is not such a problem from the viewpoint of the memory requirements as the strong scaling, in particular for the case of a linear scaling code where the memory requirements increase only linearly with respect to the size of the system. If the number of cores increases in the same way as the number of atoms, the memory needs per core are expected to remain constant. Consequently the range of the number of atoms that can be used for the test is basically only limited by the number of available cores.

The results of a test for water droplets of different sizes are shown in Fig. 6.22. The smallest droplet consisted of 150 atoms, the largest one of 3600 atoms. As for the strong scaling the support functions were optimized using the hybrid mode and the

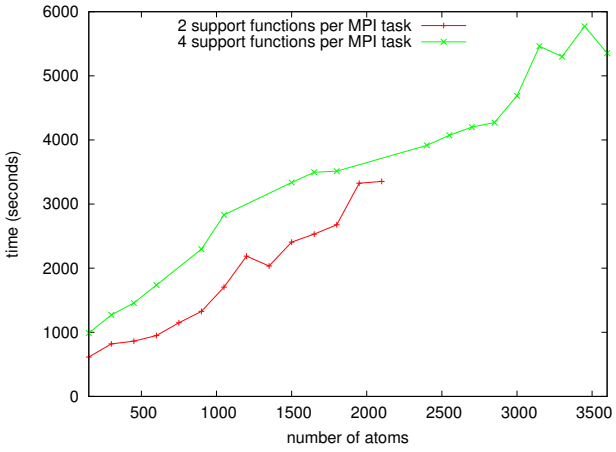


Figure 6.22: Weak scaling for water droplets of different sizes. The bad scaling has mainly two reasons. First of all the code only starts to scale linearly for rather large droplets (cf. Secs. 6.4.2 and 6.4.3). In addition the communication time increases with the number of MPI tasks even if the amount of data to be communicated per task remains constant. The runs were performed with 2 OpenMP threads per MPI task.

FOE approach was employed for the optimization of the density kernel. The cutoff radii were again set to 8 bohr and 12 bohr, respectively, and the initial prefactor for the confinement to $4.9 \cdot 10^{-3}$ hartree/bohr^4.
There are two different curves, one representing the runs with two support functions per MPI task and another one for the runs with four support functions per MPI task. Due to the limited size of the supercomputer, the runs where each MPI task holds two support functions could not go beyond 2100 atoms.
For a code that strictly scales linearly with respect to the number of atoms and furthermore exhibits a perfect parallelization, it is to be expected that the weak scaling gives a horizontal line.

This is obviously not the case here. There are mainly two reasons for this rather bad behavior. First of all, the water droplets must be rather larger in order to reach the linear scaling regime, as will be shown in more detail in Secs. 6.4.2 and 6.4.3. As a consequence it is not surprising that the run time increases for a small number of atoms even if the number of support functions per MPI task is held constant.
Later on, in regions where the code is expected to scale linearly with respect to the size of the system, the lines are still not horizontal. This is mainly due to the communication among the various MPI tasks. Whereas the overall amount of data that is communicated should increase linearly with respect to the size of the system – thus keeping the amount of data per MPI task constant –, there is an additional overhead due to the fact that more and more MPI tasks are involved in the communication. As a consequence the communication will take more and more time as the number of atoms increases.

These assumptions are confirmed by the plot in Fig. 6.23. Here the time taken by the linear algebra – comprising the FOE part which depends heavily of the sparsity of the matrices and is thus a good indicator of whether the linear scaling regime has been

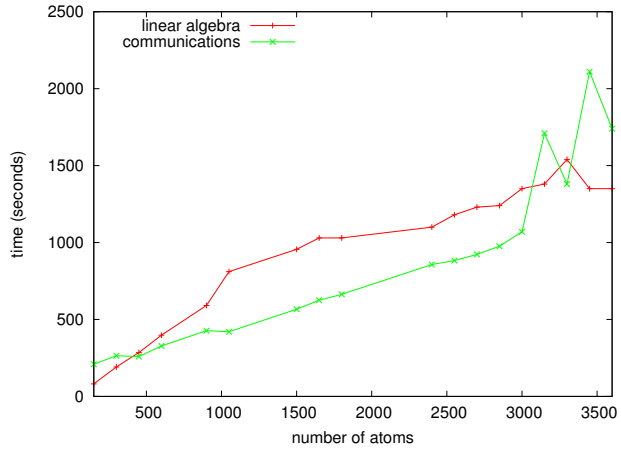


Figure 6.23: The time taken by the linear algebra and the communication for the run using 4 support functions per MPI task shown in Fig. 6.22. The time required by the linear algebra is an indicator of whether the linear scaling regime has been reached; as expected it increases rapidly in the beginning but flattens out for the large systems. The communication, on the other hand, exhibits a continuous increase over the entire range. Some possible explanations for the strong oscillations above 3000 atoms are given in the text.

reached – and by the communication are shown separately for the run with 4 support functions per MPI task. It is obvious how the linear algebra part exhibits a very steep increase in the beginning, but flattens towards the large number of atoms. The communication, on the other hand, increases uniformly over the entire range. The strong oscillations above 3000 atoms might be caused by other calculations running at the same time on the cluster, in this way increasing the total traffic on the network, or by a less favorable arrangement of the used compute nodes over the cluster.

6.4 Scaling with respect to the size of the system

The scaling with respect to the size of the system is the ultimate test to verify whether the goal – a DFT code that scales linearly – has been reached. From the discussions in the previous sections it is clear that only the Fermi Operator Expansion offers at the moment the possibility to calculate the density kernel with linear scaling. Therefore the following scaling benchmarks were all done with this method.

However it will be very hard to reach an absolutely perfect linear scaling, as there will always be some very small portions of code exhibiting a worse scaling. A simple example is the setup of the sparsity pattern of the matrices, which requires to determine which support functions overlap with each other; this procedure will result in a quadratic scaling in a straightforward implementation.
Therefore one simply has to make sure that the prefactors of these parts are as small as possible in order to minimize their bad influence on the overall scaling.

Another interesting question is that of the crossover point between the linear and the cubic scaling version, i.e. the system size at which the linear scaling version will be faster than the cubic one. This crossover point depends on the prefactors for the two versions; the larger the prefactor for the linear version is compared to the one for the cubic version, the higher will be the crossover point. It is clear that the crossover point depends also heavily on the accuracy that has to be obtained, i.e. how close the linear results should come to the cubic ones. The higher the accuracy requirements are, the larger the cutoff radius must be chosen and consequently the more the crossover point will be shifted towards larger numbers.

Furthermore it has to be noted that the convergence speed of the linear version depends also on the choice of the mixing parameter which has to be specified unless the direct minimization approach is used. Choosing a large value will accelerate the calculation, but as well decrease the stability. For the following benchmarks rather

conservative values were chosen, in particular for the one presented in Sec. 6.4.2.
The same considerations as for the mixing parameter apply as well to the choice of the mode with which the support functions are optimized. For the following benchmarks the hybrid mode is used, which in general exhibits a high reliability, but is often slightly slower than the energy minimization and the mixed mode, as follows from the discussion in Sec. 5.1.4.
Whereas all these issues do not affect the scaling behavior of the linear version, they heavily affect its prefactor, which will eventually determine the crossover point. Thus the crossover points which will be presented in the following sections might be lowered by choosing more aggressive parameters.

Furthermore it must be noted that the FOE method is used in connection with a mixing approach. This is in contrast to the cubic version where the direct minimization approach was used since the HOMO-LUMO gaps for the used test systems are large enough. Using as well a mixing approach for the cubic version would considerably slow down the calculations, in this way lowering the crossover point.

Apart from the mentioned parameters, the prefactor for the linear scaling version depends also heavily on the geometry of the structure under consideration. The best case is a large chain-like system which has a large extension in one direction and only small extensions in the other two dimensions. In such a case there are only very few overlaps between the support functions, giving rise to extremely sparse matrices and consequently a fast linear scaling code. The other extreme is a compact system where many overlaps among the support functions are present; for such a system the matrices are much less sparse, in this way increasing the prefactor considerably.
For the cubic version, on the other hand, the geometry is not that important since anyway all orbitals extend over the entire simulation box. As a consequence the crossover point will be much lower for a chain-like structure than for a compact one.

Due to these reasons the most extreme cases – chain-like alkanes and compact water droplets – will be used for the benchmarks.

6.4.1 The best case – a chain-like system

First the optimal case of the chain-like alkanes C_nH_{2n+2} is considered, for which the linear scaling version is expected to perform best. The hybrid mode was used for the optimization of the support functions and the FOE method for the optimization of the density kernel; the cutoffs were set to 8 bohr and 12 bohr, respectively. The prefactor for the initial confinement was chosen to be $4.9 \cdot 10^{-3}\,\mathrm{hartree/bohr^4}$.

For the linear scaling version the number of atoms ranged from 152 to 12002; beyond this number the available memory was not sufficient any more. For the cubic version the largest system that could be treated without running out of memory consisted of 1502 atoms. In order to set this number as high as possible and following the discussion in Sec. 6.3.2, 8 OpenMP threads and 302 MPI tasks were used, amounting to 2416 cores in total.

6.4.1.1 Timings

The runtimes for both the linear and the cubic versions as a function of the number of atoms is shown in Fig. 6.24. Before looking at the results in more detail it is necessary to note a few things which might otherwise lead to a wrong interpretation.
For the smallest molecule $C_{50}H_{102}$ the number of MPI tasks is equal to the number of support functions used in the linear version, but twice as large as the number of orbitals used in the cubic version. Since the parallelization over the support functions – or the Kohn-Sham orbitals, respectively – is one of the main concepts to exploit the MPI parallelism, it is clear that the available resources may be utilized much more efficiently by the linear version than by the cubic one. However the additional MPI tasks which are idle most of the time will become usable as soon as the number of atoms increases, resulting in a flatter rise of the run time than expected for the cubic version. Furthermore it has to be noted that the large number of OpenMP threads will tendentially perform better for the bigger systems. For the smallest molecule $C_{50}H_{152}$ the code is already heavily parallelized using the 302 MPI tasks and the speedup stemming from the 8 OpenMP threads is only moderate; however, as one goes to larger systems,

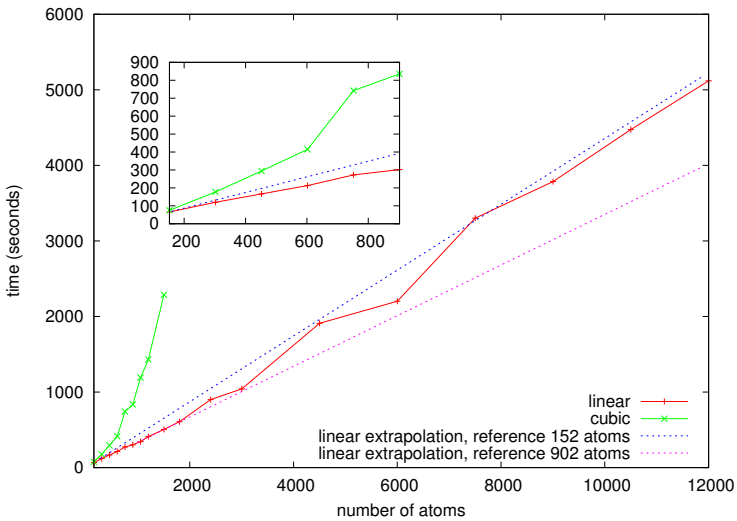


Figure 6.24: The total CPU time for both the linear and the cubic version as a function of the number of atoms for alkanes of various length. The shortest molecule consisted of 152 atoms, the longest of 12002 atoms. The cubic version could not go beyond 1502 atoms due to memory limitations. The kink in the cubic version for 752 atoms is due to an additional iteration which was required for the optimization of the Kohn-Sham orbitals. The strong oscillations of the linear version will be addressed in Sec. 6.4.1.3.

the workload per MPI task increases and these 8 threads can thus offer a substantial speedup.

With these considerations in mind it is now possible to have a closer look at the figure. It is obvious that the linear version exhibits a much more favorable scaling than the cubic one. A linear extrapolation starting from the timing for 152 atoms pretends a superlinear behavior in the beginning and an almost perfect linear scaling towards the large numbers, whereas an extrapolation starting from the value for 902 atoms reveals the small sections of the code which do not scale in a perfectly linear manner. This fact is probably due to the usage of the large number of 8 OpenMP threads, resulting in a slightly superlinear scaling for the smallest systems.
The cubic version shows rather a quadratic than a cubic scaling; it is actually even slightly better than quadratic, probably again due to the better exploitation of the parallelism for the larger systems. This demonstrates that the 1502 atoms which could be treated before running out of memory were still not enough in order to reach the range where the cubically scaling linear algebra dominates and the system is thus still in the wide range where – in agreement with the discussion in Sec. 2.4 – the quadratically scaling convolutions prevail.

To estimate the crossover point one has to look at the small inset in Fig. 6.24. It can be seen that even for the smallest system consisting of 152 atoms the total CPU time is slightly smaller for the linear version compared to the cubic one. This would indicate that the crossover point is located at less than 152 atoms. On the other hand, as already mentioned, the cubic version can not fully exploit the MPI parallelism for this small system and is therefore slightly disadvantaged compared to the linear version. The first data point where both versions can fully exploit the MPI parallelism is at 302 atoms; however the cubic version is here already considerably slower – around 50% – than the linear one. Since, according to Sec. 2.4, the scaling of the cubic version is actually rather quadratic for such small systems, the conclusion is that the true crossover point is located at around 200 atoms.
However, as mentioned in the beginning of this section about the scaling, the crossover point depends strongly on the choice of various parameters and may thus vary a lot.

6.4.1.2 Memory

Besides the timing it is also interesting to have a look at the scaling of the memory requirements for the linear and the cubic version. The memory peak for the MPI master task is shown in Fig. 6.25. Due to an integer overflow in the memory profiler the peak values could only be determined up to 1052 atoms for the cubic version and 9002 atoms for the linear version.

Still it is obvious that the linear version has a memory requirement that increases only linearly with respect to the size of the system, whereas the one of the cubic version increases much faster. In addition the plot demonstrates that even for the smallest system the memory peak of the linear version is only at about 60% of that of the cubic version, i.e. the crossover point from the viewpoint of memory usage is much lower than that from the viewpoint of the CPU time.
Furthermore, whereas the crossover point for the runtimes is considerably affected by both the cutoff radii and the other parameters as the mixing constant or the optimization mode for the support functions, the crossover point for the memory usage depends essentially only on the cutoff radius.

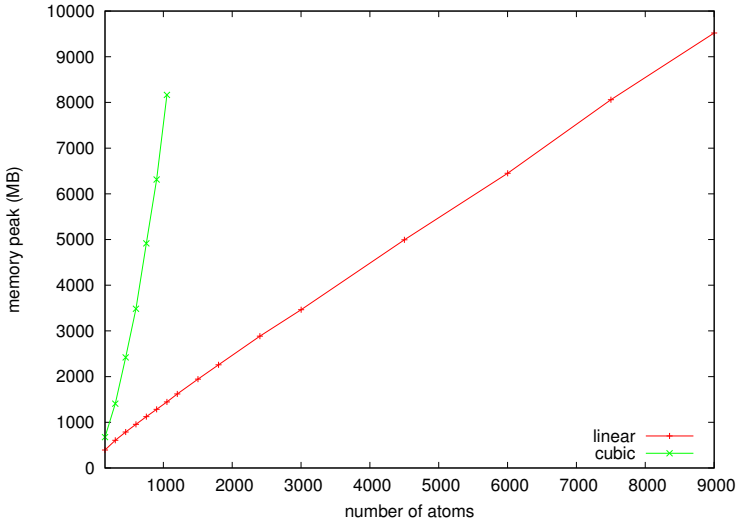


Figure 6.25: The memory peak of the MPI master task as a function of the number of atoms for the same alkanes as in Fig. 6.24. Due to an integer overflow in the memory profiler the values are only available up to 1052 atoms for the cubic version and 9002 for the linear one. The linear version shows a strict linear increase, whereas the cubic version exhibits a much faster rise of the memory requirements.

6.4.1.3 The Poisson Solver – problematic for large chain-like structures

As can be seen from Fig. 6.24 the timings for the linear scaling version exhibit considerable oscillations, in particular for the larger systems. It turns out that these variations stem mainly from the Poisson Solver, as demonstrated by Fig. 6.26. Here the total time consumption of the Poisson Solver (both communication and computation) is subtracted from the total run time and shown separately. It is obvious that the oscillations are almost entirely due to these parts.
Furthermore the remaining time shows an almost perfect linear scaling, even when taking the values for 902 atoms as reference. Only for a very large number of atoms there are some small deviations from the straight line which must be caused by some other small sections of the code which are not yet fully linearized. Thus one can conclude that the Poisson solver is – at least for the alkanes – the main cause why the strict linear scaling is slightly spoiled.

The rather bad performance of the Poisson Solver has several reasons.
First of all the scaling is by construction not strictly linear, since the Poisson solver has an intrinsic $N \log N$ scaling due to the Fast Fourier Transforms (FFTs) that are used.
Secondly the oscillations in the timing are most probably caused by the zero-padding required for the FFTs, which puts some restrictions on the size of the data sets which can be processed. Thus it might happen that in some cases even a small increase in the size of the system blows up the dimensions of the Poisson Solver considerably, whereas in other situations an increase of the system does not even alter them.
Last but not least it has to be noted that the parallelization of the Poisson solver over planes is not well suited for long chain-like structures as the alkanes. As an example, the dimensions of the computational box for the largest alkane were $161 \times 165 \times 59673$, meaning that the number of planes in the z direction is several hundred times larger than the one in the other two directions. As a consequence the parallelization over planes is very efficient as long as they are aligned along the z direction; however the Poisson Solver also has to perform operations in planes along the other dimensions, which leads to a severe load unbalancing if the number of planes is smaller then the number of MPI tasks.
A solution to this problem would be to parallelize the Poisson solver over lines instead of planes. In this way the workload could be distributed among the MPI tasks in a much more efficient way. Some preliminary results for this approach will be presented in Sec. 6.5.5.

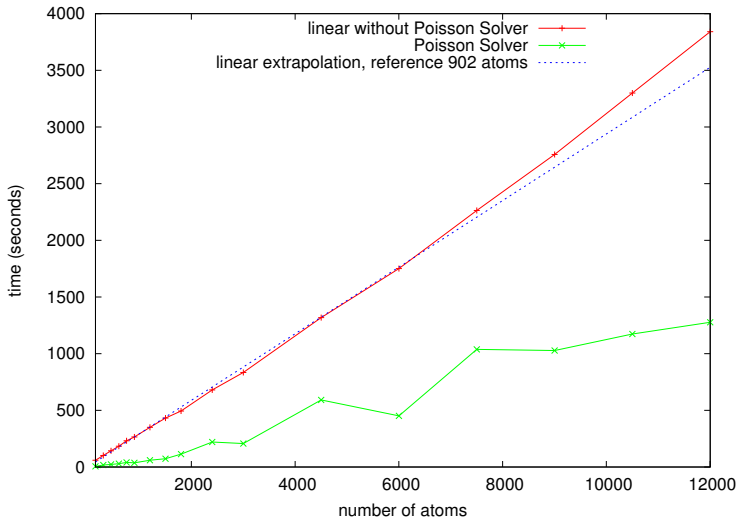


Figure 6.26: Plot of the total CPU time for the linear version – i.e. the same quantity as in Fig. 6.24 – but this time split up in the time taken by the Poisson solver (both communication and computation) and the remaining part. It is obvious that the Poisson solver part is responsible for both the large oscillations in the runtime and the spoiling of the linear scaling which are visible in Fig. 6.24.

6.4.2 The worst case – a compact system

The worst case for the linear scaling version is a compact system, exhibiting much overlap among the support functions and thus leading to rather dense matrices. As a specific example water droplets of different size were considered; the smallest droplet consisted of 150 atoms, the largest one of 8400 atoms. For the cubic version is was not possible to go beyond 1500 atoms due to memory limitations. Again 8 OpenMP threads were used in order to push this limit up as much as possible. The number of MPI tasks was set to 300, thus being equal to the number of support functions used for the smallest droplet. Consequently 2400 cores were used in total.
As for the alkanes the hybrid mode in connection with the FOE method was used; the cutoffs were again 8 bohr and 12 bohr, respectively, and the prefactor for the initial confinement was once more set to $4.9 \cdot 10^{-3}$ hartree/bohr^4.

6.4.2.1 Timings

In Fig. 6.27 the total CPU time is plotted as a function of the number of atoms for both the linear and the cubic version. Again it is important to note that the available MPI parallelization is smaller for the cubic version, but this time the effect is not as heavy as for the alkanes. Whereas in that case the number of orbitals was only half the number of support functions, here the number of orbitals is two third the number of support functions. Consequently the cubic version is still a bit disadvantaged, but not as heavily as for the alkanes. Furthermore it is again to be expected that the large number of 8 OpenMP threads gives a better speedup the larger the systems are, thus slightly improving the scaling.

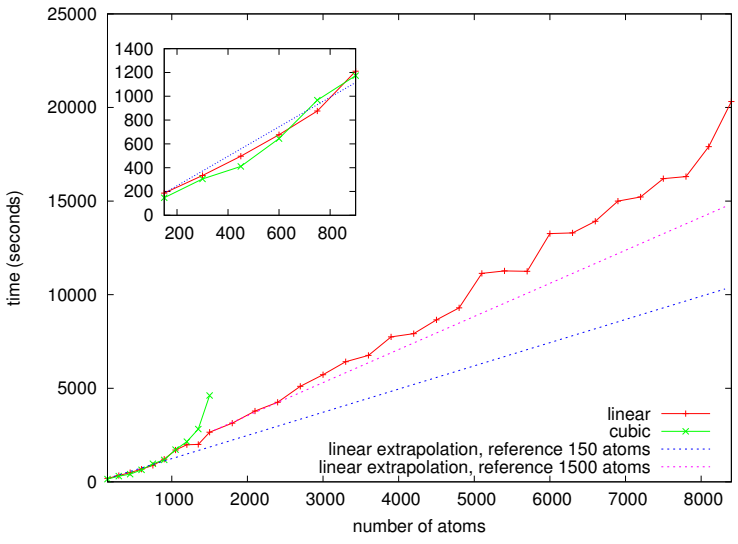


Figure 6.27: The total CPU time as a function of the number of atoms in water droplets of different size. The smallest droplet consisted of 150 atoms, the largest one of 8400 atoms. The cubic version could not go beyond 1500 atoms due to memory limitations. The scaling for the linear and the cubic version is similar in the beginning; the linear version starts to show the desired scaling only at a larger number of atoms. Still there are some deviations from the perfect linear scaling for the largest systems.

First of all it is obvious that the scaling of the linear and the cubic version is very similar in the beginning and seems to be rather linear. Whereas this is to be expected for the linear scaling version, it is surprising for the cubic one; however this behavior can most probably be explained by the better exploitation of the available cores for the larger systems. Furthermore the similar scaling of the linear and the cubic version means that the system is still to small in order to allow the linear version to exploit the localization properties of the support functions and the corresponding sparsity of the matrices. This becomes also visible by the linear extrapolation from the value for 150 atoms, which lies considerably below the actual numbers.
Only at about 1000 atoms the system seems to be large enough such that the linear version can take advantage of the localization and sparsity properties and thus starts to exhibit a better scaling than the cubic version. However the extrapolation from the value for 1500 atoms shows that the scaling is still not perfectly linear, but nevertheless it is obviously much better than that of the cubic version.

The determination of the crossover point is this time not so easy, since both versions exhibit a similar behavior in the beginning. From Fig. 6.27 it can be concluded that the point where the linear scaling version starts to perform considerably better than the cubic one is located at around 1000 atoms. Due to this rather large number, the considerations about the exploitation of the parallelism are not as important as for the alkanes.
The fact that the crossover point is much higher compared to the case of the alkanes demonstrates the huge impact of the geometry. However, it must be noted that – as for the alkanes – the crossover point might be lowered by choosing more aggressive parameters.

6.4.2.2 Memory

The memory peak for the MPI master task is shown in Fig. 6.28. Due to an integer overflow in the memory profiler the values are only available up to 5100 atoms for the linear version and 1050 atoms for the cubic version.

As for the case of the alkanes it is obvious that the memory requirements increase strictly linearly for the liner scaling version, whereas they grow much faster for the cubic version.
Again there is no crossover point, i.e. the linear version requires always less memory than the cubic one. For the smallest droplet the peak of the linear version is only 55% of that of the cubic version.

This demonstrates that the memory requirements are not that much dependent – even though there is still some effect – on the geometry as the runtime.

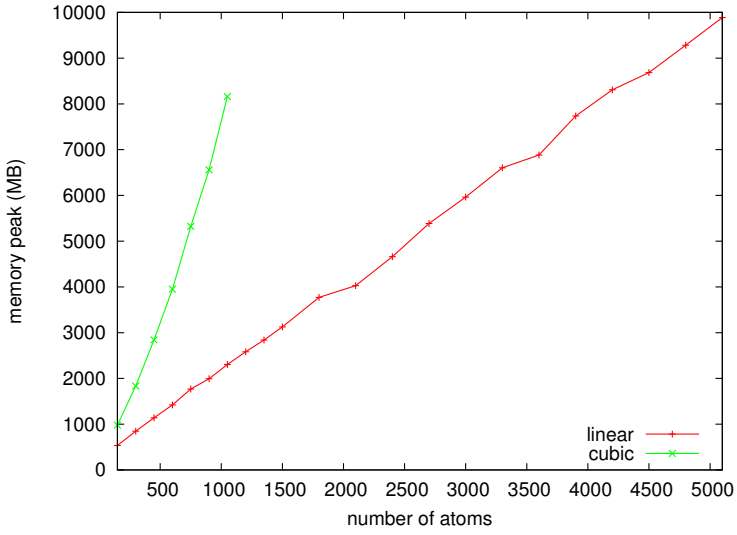


Figure 6.28: The memory peak of the MPI master task as a function of the number of atoms for the same runs as in Fig. 6.27. The linear version shows a strict linear increase, whereas the memory requirements of the cubic version increase much faster. Due to an integer overflow in the memory profiler the values are only available up to 5100 atoms for the linear and 1050 atoms for the cubic version.

6.4.3 Impact of the geometry on the sparsity properties

The results of the Secs. 6.4.1 and 6.4.2 have already demonstrated that the geometry has a huge impact on the runtime of the linear scaling version. This is mainly due to the fact that the overlaps among the support functions – and thus the sparsity of the matrices – depend strongly on the geometrical arrangement.

An impressing demonstration of the impact of the geometry on the sparsity of the matrices is shown in Fig. 6.29. The plot shows on the right axis the relative sparsity of the matrices – i.e. the number of zero-elements divided by the total number of elements – as a function of the number of atoms; on the left axis it shows the relative time taken by the FOE part, which depends heavily on the sparsity of the matrices, again as a function of the number of atoms. As soon as this relative time stops to increase, it can be concluded that the system has reached the linear scaling regime.
The data for the plot was extracted from the same runs which were used to illustrate the scaling behavior in the Secs. 6.4.1 and 6.4.2, thus representing the most favorable case of the chain-like alkanes and the worst case of the compact water droplets. Since the number of MPI task was chosen such that in both cases each MPI task had to handle one support function for the smallest system and in addition also all the other crucial parameters (as the cutoff radii) were identical, these runs allow a fair comparison. The only parameters that were different are the mixing parameter and the number of iterations in the kernel loop, which were smaller and larger, respectively, for the water droplets; this slightly increases the relative amount taken by the FOE part for these systems, but still the general statement will remain valid.

First of all it becomes clear from the figure that the matrices for the alkanes are much sparser than those for the water droplets even though the same localization radii were

used for both systems, which is simply a consequence of their geometrical arrangement.
Furthermore it is obvious how the sparsity has an impact on the relative time taken by the FOE part. For the smallest system – 152 atoms for the alkane and 150 atoms for the water droplet – the sparsity of the matrices is still rather small for both systems, namely 75.84% and 50.12%, respectively. Consequently also the relative time taken by the FOE part – 7.4% and 10.5%, respectively – is close together.

However the situation changes drastically as the number of atoms is increased. For the alkanes the sparsity grows very rapidly and already reaches 94.87% for 752 atoms. At this point the relative time taken by the FOE part starts to remains constant at a few percent. The decrease in the beginning is probably due to a better exploitation of the parallelism; the 8 OpenMP threads that were used were most likely not able to perform well for the smallest systems.

For the water droplet, on the other hand, the same level of sparsity is only reached for a much larger number of atoms; a value of 94.54% is attained for 2400 atoms. Therefore the relative time taken by the FOE part grows in a similar manner as the sparsity and saturates at much higher values than for the alkanes; it only starts to remain constant at a sparsity level of about 95%.
Again it can be observed that the FOE time increases a bit more slowly than expected in the very beginning which is probably once more due to a better exploitation of the parallelism.

As a rule of thumb it can be concluded from these tests that, in order to reach the lin-

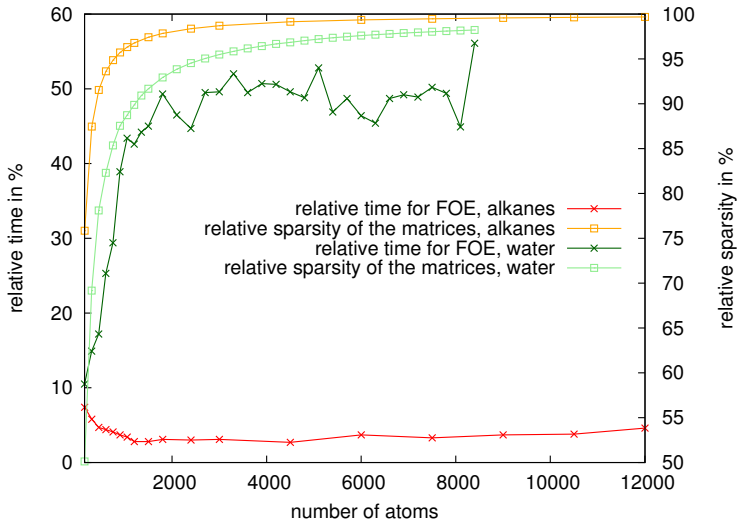


Figure 6.29: The relative time taken by the FOE part (left axis) and the relative sparsity of the matrices (right axis), extracted from the same runs that were used in Figs. 6.24–6.28. The matrices for the alkanes are much sparser than those for the water droplet, demonstrating the huge impact of the geometry of the system on this property. As a rule of thumb it can be concluded that the matrices need to exhibit a relative sparsity of at least 95% in order to reach the linear scaling regime, as indicated by the relative FOE time remaining constant.

ear scaling regime, the matrices need to exhibit a relative sparsity of at least 95%. The number of atoms for which this value is attained depends strongly on the geometry, and consequently does as well the size for which the linear scaling regime is reached.

6.5 Open problems

In spite of the already quite impressing performance of the linear scaling version of BigDFT, there are still some issues that need to be further investigated and improved.

6.5.1 Convergence criterion for the support functions

A very important point which needs to be addressed is to find a good convergence criterion for the optimization of the support functions. So far they are optimized until one runs into the orthogonality problem causing a breakdown of the optimization procedure. As soon as this happens, the support functions can de facto not be improved any further, so the code stops the optimization and one consequently works with a fixed set of support functions from this point on.

In general the support functions are already of good enough quality when this breakdown occurs in order to still yield highly accurate results. The problem is rather that sometimes this breakdown happens slightly later, making the already good support functions even better. Whereas this is not a problem for one single calculation, it can be problematic if results from different calculations have to be compared. In order to prevent this problem, it would be necessary to stop the optimization of the support functions always at the same stage of quality.

The most obvious way to define a convergence criterion would be to rely on the gradient of target function with respect to the support functions. However this will not work due to the strict localization that is enforced, which will cause the target function to saturate even if the gradient does not go to zero. Thus the gradient can not be used as a good convergence criterion.

Another simple possibility would be to stop the optimization as soon as the difference of the target function between two subsequent iterations is below a given threshold. However this is not a good criterion either, since in this way a slow convergence might pretend that this threshold has already been reached and the optimization is then stopped too early.

A slightly more sophisticated approach, which combines both the gradient and the difference in the target function between two iterations, will be shown in the following. To first order it can be assumed that the difference in the target function is given by the gradient, $|g^\alpha\rangle$, times the change in the support functions between two iterations, $|\Delta\phi^\alpha\rangle$:

$$\Delta\Omega \approx \sum_\alpha \langle g^\alpha | \Delta\phi^\alpha \rangle \,. \tag{6.7}$$

Assuming that the support functions are optimized using steepest descent with a step size α, i.e. $|\Delta\phi^\alpha\rangle = -\lambda\,|g^\alpha\rangle$, this estimated difference becomes

$$\Delta\Omega = -\lambda \sum_\alpha \langle g^\alpha | g^\alpha \rangle \,. \tag{6.8}$$

Without the localization constraint it should be possible to converge $\Delta\Omega$ to any small value. However, following the discussion in Sec. 5.1.6, this is not possible, mainly due to the orthogonality that is imposed and which is competing with the strict localization. Consequently, if the overall change in the value of the target function between two iterations is supposed to be negative, it is necessary that $\Delta\Omega$ must be larger in magnitude than the increase caused by the orthogonalization which follows the update of the support functions. As soon as the opposite happens, the support functions have to be considered as converged. If the increase caused by the orthogonalization is denoted by ξ, the convergence condition thus reads

$$-\Delta\Omega < \xi. \tag{6.9}$$

Inserting Eq. (6.8) yields

$$\lambda \sum_\alpha \langle g^\alpha | g^\alpha \rangle < \xi, \tag{6.10}$$

from which a convergence criterion for the mean gradient norm can be derived:

$$\sqrt{\frac{\sum_\alpha \langle g^\alpha | g^\alpha \rangle}{N_{\mathrm{sf}}}} < \sqrt{\frac{\xi}{\lambda N_{\mathrm{sf}}}}, \tag{6.11}$$

where N_{sf} is the number of support functions.

Even though this scheme is fully implemented, it is still not completely clear yet whether it is a suitable criterion in practice. Thus the optimization is at the moment typically stopped after two iterations and the support functions are permanently fixed as soon as one runs into the orthogonality problem.

6.5.2 Strict treatment of the quasi-orthogonality

As already mentioned the support functions are only quasi-orthogonal due to the strict localization that is imposed on them. This is in principle not a problem since it simply

introduces some overlap matrices $\mathbf{S}$ – or $\mathbf{S}^{-1}$ and $\mathbf{S}^{-1/2}$, respectively – here and there. In addition it has been shown that simplifying the calculation of $\mathbf{S}^{-1/2}$ by a first order Taylor expansion might be an acceptable compromise between accuracy and speed .

Still there are several open questions related to this topic. First of all it is not completely clear whether this correction due to the quasi-orthogonality is needed everywhere at all. It might well be that it is only required when the density kernel is calculated, but negligible for the optimization of the support functions.
Furthermore more investigations are required in order to fully validate the approximation of $\mathbf{S}^{-1/2}$ for the kernel optimization and to explore whether there are better options for this task.

6.5.3 Releasing the orthogonality constraint

As shown in Sec. 5.1.6 the strict orthogonality that is imposed on the support functions can cause problems towards the end of the optimization procedure. A possible solution would be to release this constraint as soon as it starts to cause problems.
However it has to be made sure that still the overlap matrix among the orbitals does not become too distinct from the identity, since otherwise the various approximations that rely on this quasi-orthogonality are doomed to failure.

6.5.4 Preconditioning

Finding an efficient preconditioning scheme is a rather involved task. Since it does not modify any result, but simply acts as a convergence accelerator, it is difficult to determine whether the approach being currently used is well suited.

Whereas the preconditioning procedures for the pure trace minimization and for the pure energy minimization are rather straightforward, the prescription for the hybrid method is just one proposition. It might well be that there exist other approaches which give better results.

6.5.5 Optimizations for extreme conditions

As has been demonstrated there are some sections of the code which exhibit problems when they are used under extreme conditions.

One example is the Poisson Solver for very long alkanes and many MPI tasks. As mentioned the problem could be alleviated by a parallelization over lines instead of over planes. Some preliminary results of this approach are shown in Fig. 6.30. Here the time taken by the Poisson Solver (both communication and computation) is shown as a function of the number of atoms in alkanes of various length. The runs were executed using 3004 MPI tasks. Since this number considerably exceeds the dimensions of the Poisson Solver in the x- and y-direction, it is clear that the old parallelization over planes is problematic and causes a severe load unbalancing which manifests itself in the timings. The new parallelization over lines, on the other hand, can much better exploit the parallelism and is thus considerably faster. For the largest system consisting of 7502 atoms, the new version outperforms the old one by a factor of about 15.

Another issue is the reduction of the memory requirement for very large systems. Since some quantities – e.g. the sparse matrices – are stored by each MPI task, they can create severe bottlenecks. Consequently one has to think about a way of distributing these quantities among the various MPI tasks.

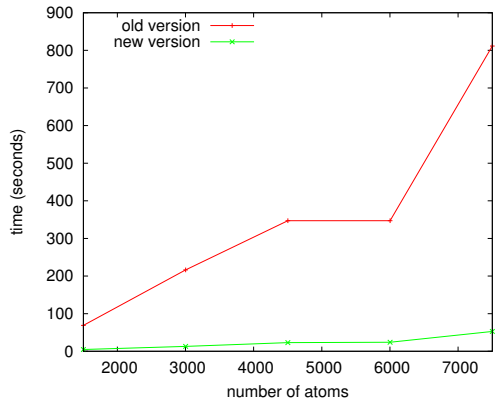


Figure 6.30: Comparison of the time taken by the Poisson Solver for the old version, which is parallelized over planes, and the new version, which is parallelized over lines, for alkanes of various lengths. The support functions were optimized using the hybrid mode and the density kernel was constructed using the FOE method; the cutoff radii were set to 8 bohr and 12 bohr, respectively, and the initial prefactor for the confinement was set to $4.9 \cdot 10^{-3}$ hartree/bohr^4. 3004 MPI tasks and 4 OpenMP threads were used, thus amounting to totally 12016 cores.

6.5.6 More sparse algebra

Whereas all methods to optimize the support functions systematically exploit their localization properties, the situation for the kernel optimization is different. So far only the FOE method makes full usage of the sparsity properties of the matrices, thus reaching a strict linear scaling. All other methods, i.e. the diagonalization methods and the direct minimization, do not fully exploit this sparsity.

It will be rather hard to make the diagonalization methods scale linearly. Two packages that allow to solve for eigenvectors of large sparse matrices were tested, namely Anasazi [76] and SLEPc [77]. Even if they are in principle able to deal with very large

matrices, they are mainly designed to calculate only a few eigenvectors from a matrix. This is in strong contrast to the requirements of the direct diagonalization methods, which need to know a large part of the spectrum of the matrices.
The situation for the direct minimization is better, since the diagonalization is only required for the overlap matrix among the fictitious Kohn-Sham orbitals during the orthonormalization. It might be that this diagonalization can be avoided by again using a Taylor approximation.

6.5.7 More feelings for the parameters

Another important point is the tuning of the various input parameters, e.g. localization radius, confining potential, number of iterations in the inner loops, etc. Even if the number of parameters is not huge, they may have a big impact on the speed and accuracy of the calculation.
Even if some knowledge about these parameters is available at present, more tests are required for a full understanding.

6.5.8 More functionals

At the moment the linear scaling code is only able to work with LDA functionals. A generalization to GGA functionals should not be too difficult, since it basically just requires a modified distribution of the charge density such that each MPI task is able to calculate the gradient of this quantity, which then enters into the calculation of the functional.

Hybrid functionals, which contain some portions of exact Hartree-Fock exchange, can in principle as well be handled. Denoting the Kohn-Sham orbitals by ψ_i, this exchange energy is given by [1]

$$E_X^{HF} = \sum_{i,j} \iint \frac{\psi_i(\mathbf{r})\psi_j(\mathbf{r}')\psi_j(\mathbf{r})\psi_i(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}'. \tag{6.12}$$

By replacing the Kohn-Sham orbitals by their representation in terms of the support functions according to Eq. (3.36) one gets

$$\begin{aligned} E_X^{HF} &= \sum_{i,j} \sum_{\alpha,\beta,\gamma,\delta} c_{i\alpha}c_{j\beta}c_{j\gamma}c_{i\delta} \iint \frac{\phi^\alpha(\mathbf{r})\phi^\beta(\mathbf{r}')\phi^\gamma(\mathbf{r})\phi^\delta(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}' \\ &= \sum_{\alpha,\beta,\gamma,\delta} K_{\alpha\delta}K_{\beta\gamma} \iint \frac{\phi^\alpha(\mathbf{r})\phi^\beta(\mathbf{r}')\phi^\gamma(\mathbf{r})\phi^\delta(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}', \end{aligned} \tag{6.13}$$

which allows to evaluate the Hartree-Fock exchange energy in terms of the density kernel and the support functions.

The same arguments apply also to meta-GGA functionals, where the kinetic energy density may as well be written in terms of these quantities:

$$\frac{1}{2}\sum_i |\nabla\psi_i(\mathbf{r})|^2 = \frac{1}{2}\sum_i\sum_{\alpha,\beta} c_{i\alpha}c_{i\beta}\nabla\phi^\alpha(\mathbf{r})\nabla\phi^\beta(\mathbf{r}) = \frac{1}{2}\sum_{\alpha,\beta} K_{\alpha\beta}\nabla\phi^\alpha(\mathbf{r})\nabla\phi^\beta(\mathbf{r}). \tag{6.14}$$

However the situation is more complicated when it comes to functionals which explicitly depend on a single Kohn-Sham orbital or the orbital-density, as SIC functionals. These quantities can again be expanded in terms of the support functions, but this time the coefficients $c_{i\alpha}$ are explicitly required and can not be replaced by the density kernel:

$$\begin{aligned}\psi_i(\mathbf{r}) &= \sum_\alpha c_{i\alpha}\phi^\alpha(\mathbf{r}),\\ \rho_i(\mathbf{r}) &= \sum_{\alpha,\beta} c_{i\alpha}c_{i\beta}\phi^\alpha(\mathbf{r})\phi^\beta(\mathbf{r}).\end{aligned} \tag{6.15}$$

If the density kernel is optimized by the direct diagonalization or the direct minimization approach, then these coefficients are available. For the FOE approach, on the other hand, only the density kernel is known, and there is no way to get access to the expansion coefficients of the Kohn-Sham orbitals. Consequently such orbital-dependent functionals cannot be used in connection with the FOE approach.

6.5.9 Improve the quality of the forces

Even though it has been shown in the Secs. 6.1.2 and 6.1.3 that the forces calculated by the linear version are quite accurate, there still seems to be some room for improvement.

First of all it might be worth to see whether it is nonetheless possible to determine the Pulay forces which are neglected at the moment.
Furthermore it turned out that the noise level of the forces is usually higher compared to the cubic version, in particular for small cutoff radii. Finding a way to reduce this noise would allow to use smaller cutoff radii in practical applications and thus accelerate the calculations a lot.

6.5.10 More boundary conditions

At the moment the linear version can only handle free boundary conditions. A generalization to wire, surface and periodic boundary conditions would considerably enlarge the range of possible applications.

6.5.11 Technical optimizations

Even though some sections of the code are already highly optimized – e.g. the matrix vector multiplications for the FOE or the communication for the calculation of scalar products and the charge density – there are some other sections which can still considerably be improved.
This is in particular the case for the direct minimization approach, where some preliminary results are already available and are illustrated in Fig. 6.31. This plot shows the scaling of the direct minimization mode as a function of the number of MPI tasks for the old unoptimized version and the new one after the improvements. The timings for the smallest system were both scaled down to one since the tests were executed on different machines; because the optimizations do not only affect the prefactor of the method, but also its scaling, the improvements are still visible. It is amazing how much could be gained by the technical optimizations.

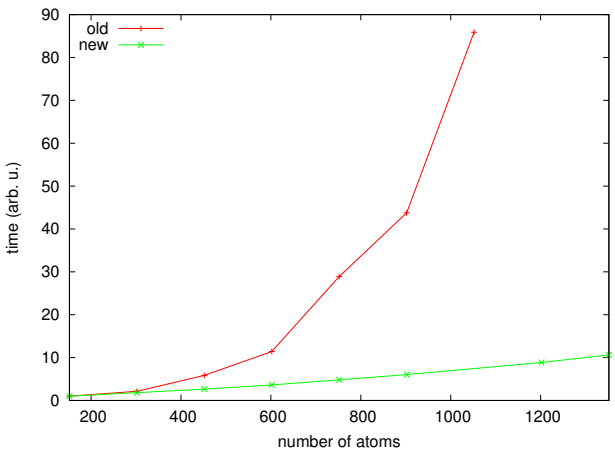


Figure 6.31: Scaling of the direct minimization mode before and after the technical optimizations as a function of the number of atoms in alkanes of various lengths. The support functions were optimized with the hybrid mode using a cutoff radius of 8 bohr and an initial prefactor for the confinement of $4.9 \cdot 10^{-3}\,\mathrm{hartree/bohr^4}$. In order to allow a fair comparison, the timings for the smallest systems were scaled down to one since the tests were performed on different machines. The runs were executed using 302 MPI tasks and 2 OpenMP threads.

CHAPTER 7

Conclusions and outlook

The first part of this Thesis described in detail the various steps – from the theoretical background over the practical implementations, the technical challenges and finally the check of the results – that had to be taken in order to develop a DFT code which scales only linearly with respect to the size of the system.

First it was demonstrated that the solution of the electronic structure problem can be efficiently solved within the framework of DFT, which gives a good balance between cost and accuracy. Next it was shown that the intrinsic cubic scaling of this formalism can be reduced to a strict linear scaling by exploiting the decay properties of the density matrix describing the system. After a very short introduction to wavelets, which form the underlying basis set and are predestined for linear scaling calculations, the largest part of the text was devoted to the description of the implementation of a linear scaling version within the framework of the BigDFT package.

Representing it in terms of a set of support functions – which are optimized in-situ – and the density kernel allows to determine the density matrix in an accurate and efficient way and paves the way towards a linear scaling algorithm. Several approaches have been developed for the optimization of the support functions and the density kernel; it will be a task for the future to further determine which ones are best suited for the actual purpose.
Most of the bottlenecks that appeared on the way towards an algorithm with the desired scaling properties could be eliminated and it is therefore possible to perform calculations whose time requirements scale only linearly with respect to the size of the system. The benchmarks towards the end of the text have furthermore demonstrated

that the code still yields very accurate results. In addition it is highly parallelized and can scale up to thousands of cores.

Even though there are still some open questions whose solutions are expected to improve the accuracy and the speed further, these preliminary results are already very encouraging.

Apart from further improving and stabilizing the code, the next step will be to to see its performance for realistic applications. Thanks to the fact that the linear scaling version is able to give reasonable results with a standard set of parameters it is to be expected that this transition from the benchmark systems which have been considered so far to the real ones will be possible straightforwardly.

To this end it will also be necessary on the one hand to remove some bottlenecks which are still present under certain circumstances and one the other hand to enhance the functionality of the code. As these issues are mainly technical things and should not create fundamental problems it is to be expected that these steps can be taken smoothly.

Part II

Boron aggregation in the ground states of boron-carbon fullerenes

CHAPTER 8

Introduction

Unlike the first part of the Thesis, which was only describing basic developments, this second part is about a real application, namely the structural investigation of boron-carbon fullerenes.

Since its discovery by Kroto et al. in 1985 [89] the C_{60} fullerene has found a wide range of applications as a building block in the field of nanoscience. For instance it is possible to directly form solids out of it [90] or to dope it by adding substitutional or endohedral atoms, e.g. in the context of hydrogen storage [91].
For future applications it would be advantageous to have more such basic building blocks which could then be selected depending on the specific needs. One possibility is to modify the original carbon fullerene by substitutional doping. A very popular choice for the dopant atoms are boron and nitrogen since they are neighbors to carbon in the periodic table; thus they are comparable in size and electronic properties and it is to be expected that they can be integrated into the carbon geometry without affecting the overall shape too much. Various boron-carbon heterofullerenes have been observed experimentally [92, 93]. The existence of cross-linked $N_{12}C_{48}$ fullerenes could explain experimental measurements of thin solid films [94]. The case of boron is of particular importance as it is the p-type counterpart of the n-type nitrogen doping in fullerenes and graphene used to tune their electronic or catalytic properties [95, 96].

It is clear that such heterofullerenes may only be useful as building blocks in practice if they are energetically stable. Therefore is is important to get more insight into the structural properties of these compounds, i.e. to determine the energetically most favorable configuration for a given stoichiometry and to get some knowledge about

the surrounding energy landscape. If the energetically most favorable structure is reasonably separated from the other ones, it is to be expected that it can be produced experimentally.
In order to determine the energetically lowest structure, it is in principle necessary to perform an unbiased search over all possible stable configurations. Of course this is not feasible for such a large system, since this number increases in general exponentially with the number of atoms. As a consequence one either has to use an algorithm which is able to find the most favorable configuration without searching over all possibilities – a task which became possible only recently [97–100] – or one has to constrain the search beforehand in some way, thereby only investigating a small portion of the entire energy landscape.

For the stoichiometries $C_{60-n}B_n$ this constraint typically consisted in starting the search from the perfect C_{60} fullerene and substituting n carbon atoms by boron. Garg et al. [101] extensively investigated the geometries $C_{60-n}B_n$ for $n = 1 - 12$. They concluded that the boron is arranged in such a way that a pentagon ring does not contain more than one boron atom and a hexagon not more than two boron atoms (at non-adjacent sites). Putting more boron atoms in a ring increases the bond lengths and decreases the stability. A study by Viani and Santos [102] on various smaller fullerenes again confirmed that boron atoms are most preferably situated at opposite sites in a hexagon, thereby increasing the bond lengths in their neighborhood. For the heterofullerene $B_{12}C_{48}$, which will be one of the two stoichiometries investigated here, Manaa et al. [103] did a detailed study; they claimed that the best structure was the same that was previously found for $N_{12}C_{48}$ [104], thereby again confirming the previous results stating that the boron atoms should be distributed over the entire carbon cage and isolated.

This second part of the Thesis describes how an extensive and unbiased search for energetically low structures for the stoichiometries $B_{12}C_{48}$ and $B_{12}C_{50}$ was performed, however this time trying to explore a wider range of the energy landscape by not restricting the investigation to the structural motif of substituted C_{60} fullerenes where the boron atoms are isolated. In this way it was possible to discover many new structures which are energetically more favorable than those which have been known so far. Furthermore they belong to a completely new class of structures. This demonstrates that many energetically low configurations have been missed so far.

However, before presenting the results of this investigation, some basic introduction into the field of structure prediction will be given.

CHAPTER 9

Short introduction to structure prediction

9.1 Some basic terms

In order to determine stable configurations of molecules or solids, one has to search for minima on the potential energy surface which has been introduced in Sec. 2.1. These minima are characterized by two conditions, namely the forces acting on the atoms which must be zero and the eigenvalues of the Hessian which must all be positive.
Any point on the potential energy surface which exhibits these two properties is a local minimum, whereas the global minimum, on the other hand, is the energetical minimum of all local minima.

In this context it is also useful to introduce the term basin of attraction. A basin of attraction for a given minimum consists of all configurations which would relax into this local minimum if a geometry optimization with a sufficiently small step size was performed.

Another important term is that of a funnel. A funnel consists of a subset of all minima – or basins of attraction, respectively – which are all connected by a certain maximal barrier height. A system whose internal energy is higher than this maximal barrier is thus able to move freely inside the funnel and access all the minima contained within it. In order to leave the funnel, a higher barrier would have to be crossed, necessitating the system to exhibit a larger internal energy.
Of course the definition of a funnel depends on the barrier height which is considered

and the assignment of the various minima to different funnels is thus somehow arbitrary.

An illustration of all these terms is given in Fig. 9.1 for a simple one-dimensional function. There are several local minima – denoted by yellow stars – with their corresponding basins of attraction which are separated by the vertical blue bars. The global minimum is marked with a green star. Furthermore the high barrier in the middle suggests to define two different funnels, as indicated by the red arrows.

These terms do not only apply to the search of stable configurations for solids or molecules, where the function whose minima are searched for is the energy as a functions of their coordinates, but to any other quantity whose minima have to be determined. Global optimization is consequently a very general problem.

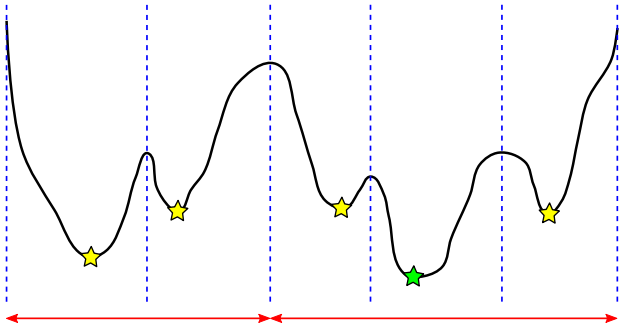


Figure 9.1: An illustration of the basic terms by a simple one-dimensional energy landscape. The vertical blue bars separate the basins of attractions belonging to the local minima indicated by the yellow stars. The green stars represents the global minimum. The two funnels being separated by the high central barrier are indicated by the red arrows.

9.1.1 Difficulties of a global optimization

Whereas a local optimization is rather straightforward – one starts from a given configuration and simply minimizes the target function using any optimization method with a sufficiently small step size – a global optimization is a very involved task. In order to determine the global minimum with absolute certainty, it would in principle be necessary to first identify all local minima and then to determine the global minimum out of them. This is problematic for two reasons: Firstly it is hard to determine whether really all local minima have been found or whether some have been missed, and secondly the number of local minima is in general tremendous, making a systematic search impossible in practice.
As a consequence the determination of global minima is a very challenging task, and it is therefore advisable to rather call the result of such an investigation a putative global minimum in order to take into account the mentioned uncertainties.

Furthermore systems with several deep funnels are even more challenging for any

global optimization algorithm, since one can very easily get trapped in a funnel which does not contain the global minimum. In such a situation it is very hard to get out of it and enter the correct funnel. This problem can at least partially be overcome by starting several global optimization searches from many different initial configurations, out of which at least one will hopefully end up in the correct funnel.

9.2 Global optimization methods

In spite of the mentioned difficulties there exist several methods trying to determine the global minimum of a given target function. Some important ones, which may also be applied to structural optimizations [105], will be briefly presented in the following sections.

9.2.1 Genetic algorithms

Genetic algorithms try to mimic Darwin's theory of evolution, meaning that they employ the concept of the "survival of the fittest". To this end one starts with an initial set of structures, which will then undergo random modifications, known as mutations. Furthermore it is possible to combine two initial structures into a new one, called a crossover. Out of all of the structures generated in this way – i.e. the initial set and the new ones – one then selects those which are the fittest; in the context of a structural optimization the fitness would be measured by the energy of the configurations.
Repeating this process of creating new structures and selecting the best ones will result in a set of configurations which will more and more exhibit the desired property, namely come close to the global minimum.

However applying genetic algorithms to a structural optimization is not as straightforward as it might seem at first sight. First of all completely random mutations of a structure will in most cases result in rather unphysical configurations. Thus it might be necessary to restrict these random modifications somehow, in this way biasing the algorithm. Furthermore it is not obvious how two structures can be combined into a new one. A possibility would be two cut the two initial configurations into two pieces and then glue together the latter ones in order to build a new structure, but it is of course questionable whether a reasonable configuration can be constructed from two fragments in this way.

9.2.2 Simulated annealing

At a sufficiently low temperature the probability of the system being in any other state than the ground state is vanishingly small due to the tiny Boltzmann weight $e^{-(E_i-E_0)/k_BT}$ of the excited states. Consequently one could perform moves one the potential energy surface and accept or reject these new configurations in such a way that one finally obtains a low-temperature Boltzmann distribution.

However this method will in general not work in practice. Unless one uses very violent moves – which will then, on the other hand, in most cases lead to very unphysical configurations – the system will be trapped in another than the global minimum since it can not overcome the surrounding barriers due to the low temperature.

Simulating annealing circumvents this problem by starting the simulation at a high temperature and then to gradually decrease it. In this way the system should hopefully have enough energy in order to overcome the mentioned barriers in the beginning even when only physical meaningful moves are used and already be close to the global minimum – or at least in the correct funnel – when the temperature becomes small. The simplest prescription is to let the system evolve according to Molecular Dynamics (MD). Due to the ergodicity the Boltzmann distribution will eventually be reached. Thus simulated annealing is in some sense imitating was is happening in nature during a cooling precess.

9.2.3 Basin hopping

The key idea behind the basin hopping method is to use a simplified potential energy surface compared to the original one. To this end the energy of an entire basin of attraction is set equal to the energy of the corresponding local minimum, resulting in a piecewise constant energy landscape. An illustration for the sample function shown in Fig. 9.1 is given in Fig. 9.2.
In this way two neighboring local minima are not separated any more by potentially

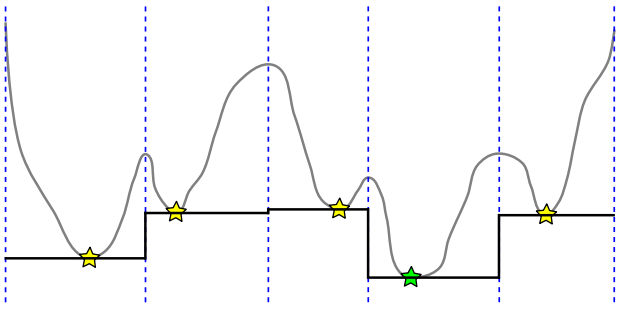


Figure 9.2: The same function as shown in Fig. 9.1, however this time modified according to the basin hopping algorithm. For each basin of attraction, the function takes on the value of the corresponding local minimum. In this way, the high barrier separating the two funnels has disappeared.

high barriers, and it is therefore much easier to switch from one minimum to the other one. This can also nicely be seen in the illustration, where the high barrier in the middle has completely disappeared, thereby also removing the two-funnel character of the system.

In the context of a Monte Carlo simulation, where a new trial configuration is accepted or rejected according to its Boltzmann factor $e^{(E_{\text{trial}}-E_{\text{old}})/k_BT}$, one is thus only comparing energies of local minima. Since these are in general not very distinct, the Boltzmann factor is consequently not getting too small. On the other hand, when using the original energy landscape, it might happen that E_{trial} is much higher than the energy of the local minimum to whose basin of attraction this trial configuration belongs, and the Boltzmann factor becomes therefore very tiny, making it very unlikely that this trial step is accepted.
The generation of the modified energy landscape can be done on-the-fly. For each trial configuration one performs a local geometry optimization whose final result will then be used to determine whether the trial configuration is accepted or not.

The temperature, which is a free parameter in the basin hopping method, has not been specified so far. Thus it could be used to combine basin hopping with a simulated annealing scheme by continuously lowering the temperature as the simulation progresses. However this is rarely done in practice.

9.2.4 Minima Hopping

The Minima Hopping method [100] is a global optimization method which is neither based on genetic algorithms nor on thermodynamics. Still it is closely related to the basin hopping method since it employs the same modified energy landscape.

However the new trial configurations – which are generated by a short MD trajectory starting from the current local minimum – are not accepted or rejected based on their Boltzmann weight, but rather based on whether their energy is higher or lower than a threshold energy which is continuously adjusted such that on average half of all trial steps are accepted. Furthermore the trial steps become heavier – by means of a higher kinetic energy used for the MD part – if a given minimum is visited several times, be it since the trial step did not lead out of the basin of attraction or be it since the algorithm simply came back after visiting some other minima.

Together these two feedback mechanisms ensure that repeated visits of the same minimum are avoided and the energy landscape can consequently be explored efficiently.

Since the escape steps are in addition always directed towards low lying barriers, this gives a fast trend towards the global minimum.

9.3 Structural stability

For the discussion of the structural stability it is in principle necessary to take into account temperature and pressure. However for the moment it is assumed that both quantities are zero; in such a case a configuration is by definition in its ground state if its energy is minimal. In the language of the previous section, the ground state thus corresponds to the global minimum of the potential energy surface. Since a system always tends to lower its energy, it can be concluded that such a configuration must consequently be stable, i.e. the system has no tendency to modify its state.

For the other local minima the situation is more complicated. Since in these cases the energy is as well at a minimum – although only locally – the system has again no tendency to leave this local minimum. On the other hand, there are by definition other minima which are lower in energy than the current one. Even if they are separated by potentially high barriers, there is a chance that the system will eventually end up in one of them. Thus such configurations which are a local – but not a global – minimum of the potential energy surface are called metastable.

As mentioned, the question whether a given system will be found in its ground state or in any other metastable state does not only depend on the heights of the barriers separating the various minima, but also on the temperature of the system and the external pressure which is applied. The higher the temperature is, the more the system will undergo thermal fluctuations and therefore more easily cross barriers. This crossing of barriers may occur in both directions, i.e. it is also possible that the system leaves the ground state and ends up in a metastable one. However, since the energy difference from the ground state to a given barrier is larger compared to the one from a metastable state to the same barrier, the stability of the ground state is still higher than that of the metastable one.

Even if a complete description of the system is in principle only possible if all minima and the corresponding barriers are known, some basic estimation about the stability of a system can still be gained from the minima alone. If the energy gap between the ground state and the excited states is rather larger, this implies that the lowest minimum is separated from the other ones by rather high energy barriers. Thus it is more likely that the system will be found in the ground state.

CHAPTER 10

New structural motifs for boron-carbon fullerenes

10.1 Methodology to determine low energy structures

As written in more detail in the introduction to this second part, it has been believed so far that the ground state for boron-carbon nanocages is given by structures that are identical to carbon fullerenes, just with some carbon atoms substituted by boron. Furthermore it has been assumed that the boron atoms are distributed across the entire surface of the fullerene and isolated, meaning that they are always separated by at least one carbon atom. Thus all investigations done so far have been biased in this direction. In the language of the global optimization problematic which was briefly introduced in Sec. 9.1 this means that only one funnel has been explored.

However there is no guarantee that the global minimum is really located in this funnel. Therefore it is necessary to explore also other parts of the energy landscape belonging to other structural motifs and funnels. Thus it was attempted in the course of this work [106] to explore the potential energy surface as unbiased as possible by generating a very wide range of initial structures, out of which at least one will hopefully belong to the funnel containing the global minimum.

The input structures for the two stoichiometries $B_{12}C_{48}$ and $B_{12}C_{50}$, which were investigated in this study, were generated using several approaches.
For $B_{12}C_{48}$, the most obvious approach, which had also been used in previous studies, was to replace 12 carbon atoms by boron in a perfect C_{60} fullerene. Another approach

consisted in replacing 10 atoms in a C_{58} fullerene and to add two additional boron atoms at the centers of both pentagons and hexagons as they were demonstrated to be the building blocks [107, 108] in the B_{80} fullerene [109]. A third one was to cut out 12 adjacent atoms from a C_{60} fullerene and to fill the hole with a compact boron icosahedron or a boron patch. In the last approach structures of high symmetry were generated for the stoichiometry C_{48} and 12 boron atoms were added at locations where it seemed appropriate by intuition.
For the stoichiometry $B_{12}C_{50}$ the first approach for the generation of initial structures was to take those configurations which turned out to be optimal for $B_{12}C_{48}$ and to manually add two additional carbon atoms. The second one consisted in replacing ten carbon atoms by boron in a C_{60} fullerene and to add two additional interstitial boron atoms at positions where it seemed appropriate by intuition. In the last approach 12 carbon atoms were replaced by boron starting from a C_{62} fullerene; for this last one two examples of C_{62} fullerenes were taken which were first described by Ayuela et al. [110] and Qian et al. [111] and turned out to be the most stable C_{62} isomers in a study by Cui et al. [112].

For all of the structures generated in this way a local geometry optimization was performed first in order to sort out the least favorable compounds. For the most interesting ones among the remaining configurations some short runs using the minima hopping method – which was presented in Sec. 9.2.4 – were performed in order to see some trends towards an energy lowering. With these informations some manual modifications – e.g. exchanging a boron and a carbon atom – were applied in order to speed up the exploration in a second step of minima hopping. Finally a systematical exchange of boron and carbon atoms up to second-nearest neighbors for the most favorable structures emerging from this process was performed.
As was stated, the minima hopping method allows, in principle, to find the global minimum of the potential energy surface; however, due to the complexity of the current system, this search would last very long if it is started in the wrong funnel and one can thus only determine the pseudo-global minimum in a given region of the potential energy surface within a reasonable time frame. However, thanks to the fact that the minima hopping searches were started from many distinct minima, this problem could hopefully be overcome since there is a larger chance that at least one of the initial structures belongs to the funnel containing the ground state.

In this way more than thousand configurations for both stoichiometries could be generated without the restriction of sticking too much to a given structural motif.

All calculations were done at the level of DFT, using the BigDFT package [71] which was presented in detail in the first part of the Thesis. All calculations were done using the traditional cubic version of the code since the system is still too small in order to

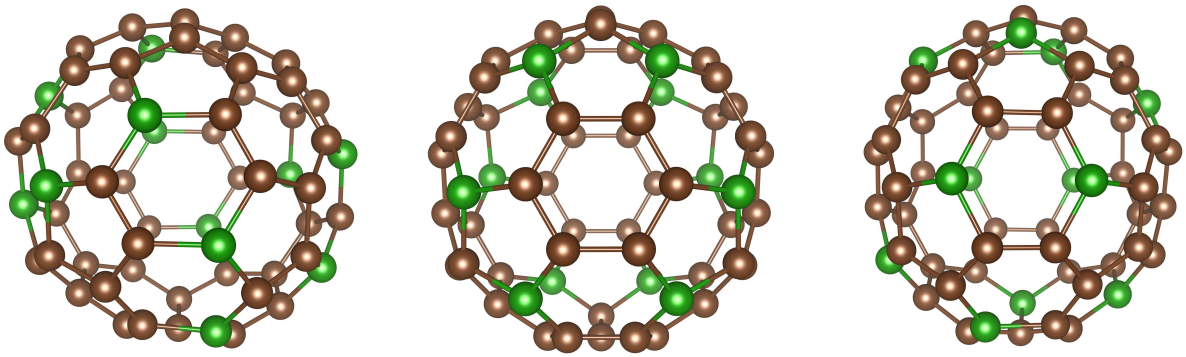
profit from the linear scaling version. The exchange correlation part was described by the PBE functional [24], which has shown to give highly reliable energy differences between different structural motifs in boron [113] and is therefore used [114] in this work. The grid spacing was set to 0.24 bohr, which allowed – together with the chosen radii for the coarse and fine regions – a convergence of the energy to 10^{-5} hartree, and all systems were relaxed until the maximal force component on any atom was within the noise level of the calculation, which was of the order of 1 meV/Å.

10.2 Energy landscape for $B_{12}C_{48}$

10.2.1 Putative ground state known so far

As already mentioned several times it has been assumed so far that the ground state for the stoichiometry $B_{12}C_{48}$ is given by structures which exhibit the same shape as the Buckminster fullerene C_{60}, just with some carbon atoms substituted by boron. Furthermore it has been believed that the boron atoms should be distributed across the entire surface and isolated, meaning that they are always separated by at least one carbon atom. In the following this class of structures will be referred to as "diluted".

The three energetically lowest structures that have been found so far [115] are shown in Fig. 10.1. In the most favorable one, which is shown in Fig. 10.1a, 6 boron atoms are situated in pentagons at the top and the bottom of the fullerene, respectively, while the



(a) The lowest structure. **(b)** The second-lowest structure. **(c)** The third-lowest structure.

Figure 10.1: The three energetically lowest structures found so far for the stoichiometry $B_{12}C_{48}$. They have in common that the boron atoms are distributed over the entire surface and isolated, i.e. always separated by at least one carbon atom. Furthermore they exhibit the same overall shape as the C_{60} fullerene.

remaining 6 boron atoms are distributed around the equator within the remaining six pentagons in a way that exhibits a S_6 symmetry; the same structure proved to be the most favorable structure as well for $N_{12}C_{48}$ [104]. In the figure, these top and bottom parts are actually located at the top right and bottom left, respectively. Several studies confirmed that this is the energetically most favorable compound [101,116].
A second type of structure – shown in Fig. 10.1b – of D_{3d} symmetry with two boron atoms per pentagon was considerably higher in energy, as well as a third structure – shown in Fig. 10.1c – of S_6 symmetry in which the boron atoms are distributed in pairs per hexagon and which was also known for $N_{12}C_{48}$ [117]. The energy differences between the lowest structure and the two other isomers were calculated [115] to be 0.65 eV and 1.13 eV, respectively.

10.2.2 New structural motifs

In the course of the survey of the energy landscape several structures were found which are considerably lower in energy than the ones which have been proposed so far. Whereas the new configurations agree with them in the overall shape exhibiting a cage-like structure, they differ substantially by the fact that the boron is not distributed over the entire cluster, but aggregated in a patch, thereby separating the surface of the compound in a boron-rich and a boron-poor part. This is in strong contrast to the widely accepted belief that the boron atoms should be isolated [118], i.e. be always separated by one or several carbon atoms. This new structural motif will be referred to as "patched".

The low energy part of the spectrum that was explored is shown in Fig. 10.2, together with figures of the 12 lowest isomers that have been found and the three lowest diluted structures. In total 143 new structures being lower in energy than the most favorable configuration known so far have been found. However, since the main focus was put on determining the ground state and not on systematically exploring the entire energy landscape, it is almost guaranteed that there are even more minima in the range between the new putative ground state and the lowest diluted structure.
The energy levels are colored on a scale from 0 to 1 which describes the relative amount of carbon atoms being first neighbors to boron. Thus a value of 0 (red) means that the boron atoms are only surrounded by boron – which can obviously not happen –, whereas a value of 1 (blue) means that the boron atoms are only surrounded by carbon. Consequently the coloring of the patched structures tends towards red values, while the coloring of the diluted ones tends towards blue values.

As has been mentioned previously the structures being lowest in energy have in com-

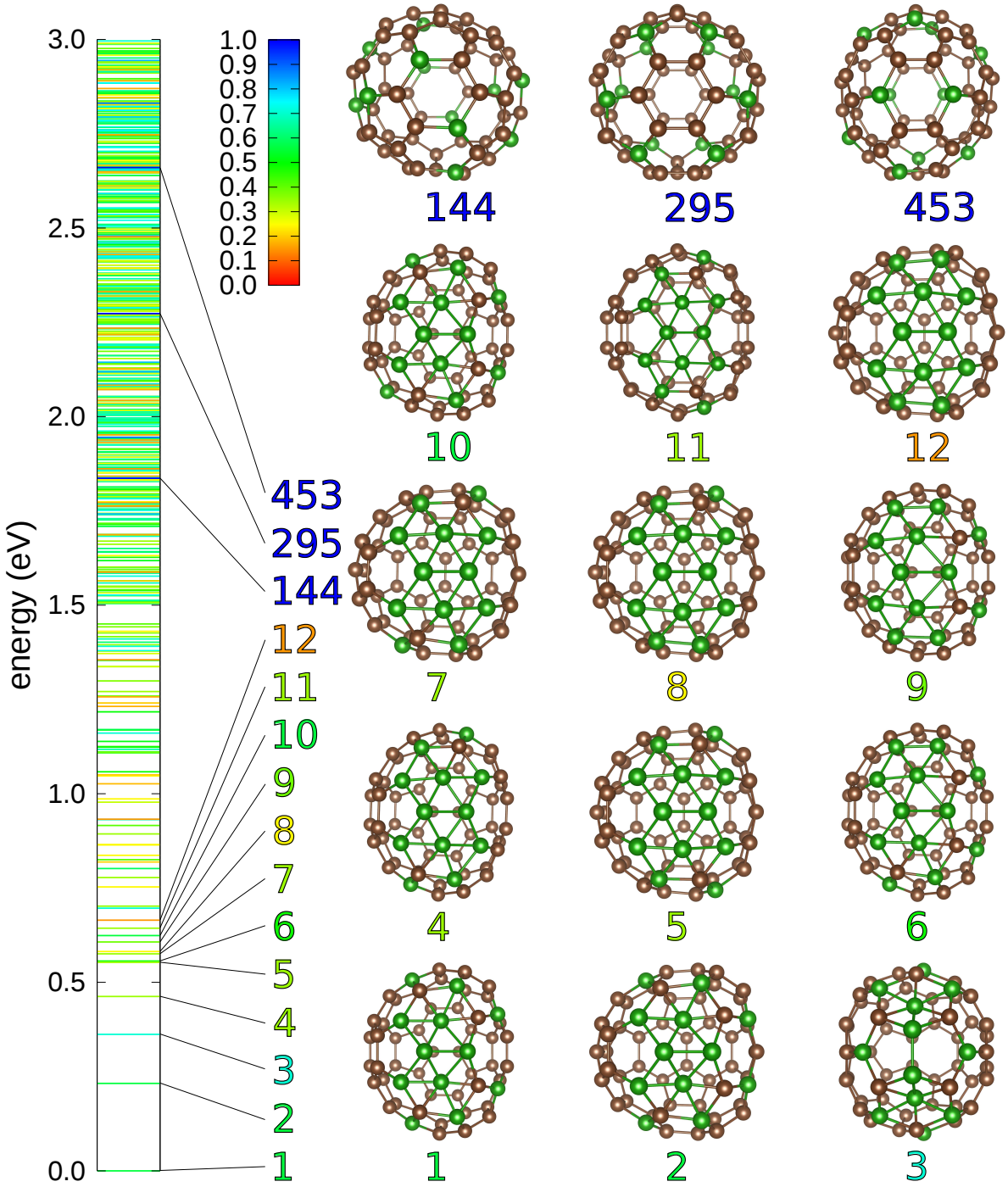


Figure 10.2: Plot of the 12 energetically most favorable structures and the three lowest diluted configurations of $B_{12}C_{48}$. On the left side the lower part – only up to 3 eV above the putative ground state – of the energy spectrum is shown. The coloring scheme is explained in the text. It must be noted that the spectrum is most likely not complete – in particular for higher energies – as the focus was put on determining the ground state and not on exploring the entire energy landscape.

mon that the boron atoms are aggregated at one single location on the surface of the cluster, in this way forming a flat patch. However it is interesting to see that for the lowest structures the boron part does not form a compact patch, but rather one which is slightly frayed at the boundaries. This results in astonishing configurations where carbon atoms have four boron atoms as first neighbors. Furthermore it is surprising that the lowest structure exhibits a heptagon, which is usually less favorable than the penta- and hexagons, and twice two adjacent pentagons, which is in general very disadvantageous [119]. However the pentagons do not only consist of carbon, but there are some substitutional boron atoms contained within them.

The energy spectrum exhibits rather large separations at the bottom and gets narrower for values which are more than roughly 0.5 eV above the new putative ground state. Furthermore there is a clear spacing between the ground state and the first excited state.

Tab. 10.1 gives some more details about the structures depicted in Fig. 10.2.
The first column shows the energy separation ΔE of the configurations with respect to the energetically lowest one. Comparing with the class of the diluted structures, it can be seen that the new putative ground state is 1.8 eV lower in energy than the most favorable configuration of that structural motif. Since, as mentioned, the latter one is identical to the putative ground state identified by Manaa et al., it follows that the

Table 10.1: Some details about the same minima as shown in Fig. 10.2, i.e. the 12 energetically most favorable ones and the three lowest diluted structures of the stoichiometry $B_{12}C_{48}$. The first column shows the energy separation ΔE to the new putative ground state, the second the HOMO-LUMO gap, the third the formation energy ΔH with respect to the bulk phases of boron and carbon, the fourth the point group and the fifth the RMSD with respect to the new putative ground state. The minima are labeled according to their energetical ordering.

	ΔE (eV)	gap (eV)	ΔH $\left(\frac{\text{eV}}{\text{atom}}\right)$	PG	RMSD (bohr)
1st	0.000	0.457	0.340	C_s	0.000
2nd	0.232	0.477	0.344	C_s	1.003
3rd	0.362	0.646	0.346	C_s	1.717
4th	0.462	0.645	0.348	C_1	1.092
5th	0.553	0.295	0.350	C_s	1.589
6th	0.557	0.236	0.350	C_1	0.558
7th	0.575	0.247	0.350	C_1	1.486
8th	0.582	0.452	0.350	C_1	1.579
9th	0.607	0.604	0.350	C_s	0.783
10th	0.624	0.378	0.351	C_1	0.533
11th	0.644	0.616	0.351	C_s	1.344
12th	0.665	0.644	0.351	C_s	1.490
144th	1.836	0.494	0.371	S_6	3.465
295th	2.273	0.316	0.378	D_{3d}	3.501
453th	2.661	0.302	0.385	S_6	3.442

new putative ground state is 1.8 eV below the one previously proposed. The second lowest structure where the boron is diluted is only the 295th lowest structure among all minima that have been found with an energy separation of 2.3 eV, being identical to the second-lowest structure identified by Manaa et al., and the third lowest diluted structure – being identical to the third lowest structure found by Manaa et al. – turned out to be the 453rd lowest structure in this study with an energy separation of 2.7 eV. These results confirm in some sense the findings of previous studies as the same diluted structures turned out to be the most favorable ones; on the other hand they demonstrate that many minima have been missed so far by restricting the search to the structural motif of diluted cages.

In the second column the HOMO-LUMO gaps of all these structures are presented. The values range from 0.2 eV to 0.6 eV, thus being rather small, and do not exhibit any special pattern. In particular there is no notable difference between the class of the patched and the diluted structures.

The next column shows the formation energies per atom ΔH with respect to the bulk conformations of boron and carbon (α-boron and cubic diamond, respectively), which is defined by $\Delta H = (E - n_B E_B^0 - n_C E_C^0)/(n_B + n_C)$ with E being the energy of the compound and E_X^0 and n_X being the energy per atom of the reference configurations and the number of atoms, respectively. As expected, the formation energy is clearly positive and does not give any useful information right here; however it can be used in order to compare different stoichiometries, as will be done later when the results for $B_{12}C_{50}$ are presented.

In the fourth column the symmetry classes of the structures are noted. Whereas the diluted structures exhibit rather high symmetries (point groups S_6 and D_{3d}), the new structures are much less symmetric (point groups C_s and C_1).

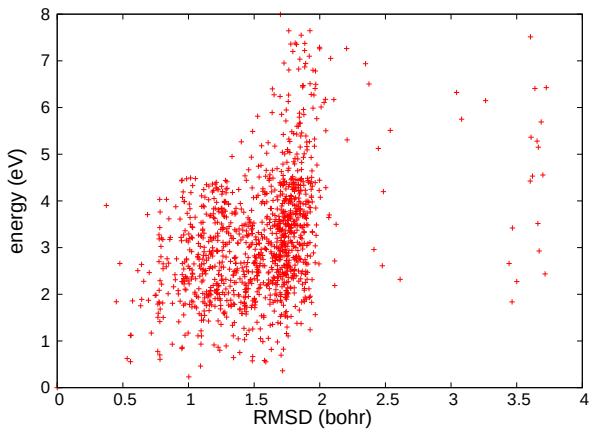


Figure 10.3: The energy difference versus the RMSD – both with respect to the putative ground state – of all structures of the stoichiometry $B_{12}C_{48}$ up to 8 eV above the putative global minimum. There is a broad range containing structures whose energies are completely uncorrelated to the value of the RMSD. However there is a sharp boundary at about 2 bohr, and the structures exhibiting a larger RMSD are not as favorable energetically as the ones belonging to the first group.

Finally in the last column the RMSD [120] values of the structures with respect to the new putative ground state are presented. It is obvious that there is a clear separation between the patched structures and the diluted ones. Within the patched ones, however, there is no relation between the energy and the RMSD.

A more complete overview of the energies and the corresponding RMSD values is shown in Fig. 10.3. Here for all structures being separated from the putative ground state by less than 8 eV the energy difference and the RMSD with respect to the latter one are plotted. As can be seen there is a broad range with structures whose energies are completely uncorrelated to the value of the RMSD. However this range is sharply bounded at a value of about 2 bohr, separating it from energetically higher configurations. These results suggest that there is a deep, but relatively flat and broad funnel ranging up to the mentioned boundary and being clearly separated from the other structural motifs which exhibit higher energies. The configurations exhibiting the largest RMSD values belong to a class where the boron atoms are arranged in two patches at opposite sides of the cluster and to the class of the diluted structures.

In Tab. 10.2 the minimal and maximal bond lengths are given for the same 15 structures which were presented in detail in Fig. 10.2 and Tab. 10.1. As can be seen the minimal and maximal boron-boron bond lengths are often shorter and longer than their coun-

Table 10.2: The minimal and maximal bond lengths in Å for the 12 energetically most favorable configurations and the three lowest diluted structures of the stoichiometry $B_{12}C_{48}$, i.e. the same structures which are shown in Fig. 10.2 and whose details are presented in Tab. 10.1. For the diluted structures there are no direct boron-boron bonds.

	bond lengths (Å)					
	B-B		C-C		B-C	
	min	max	min	max	min	max
1st	1.638	1.790	1.388	1.471	1.539	1.628
2nd	1.663	1.765	1.385	1.485	1.544	1.618
3rd	1.666	1.681	1.391	1.495	1.530	1.630
4th	1.641	1.810	1.390	1.480	1.521	1.679
5th	1.687	1.735	1.386	1.474	1.522	1.630
6th	1.619	1.787	1.391	1.494	1.526	1.776
7th	1.648	1.803	1.389	1.479	1.496	1.659
8th	1.644	1.761	1.390	1.481	1.514	1.621
9th	1.626	1.770	1.392	1.491	1.535	1.795
10th	1.637	1.793	1.382	1.492	1.529	1.631
11th	1.657	1.765	1.395	1.485	1.518	1.639
12th	1.656	1.762	1.390	1.474	1.506	1.557
144th	-	-	1.390	1.502	1.541	1.577
295th	-	-	1.380	1.490	1.548	1.587
453th	-	-	1.393	1.498	1.538	1.587

terparts in the pure B_{80} fullerene proposed by Szwacki et al. [109], which range from 1.674 Å to 1.728 Å. On the other hand the minimal carbon-carbon bond lengths are still close to their value found in a pure C_{60} fullerene (1.398 Å); however the maximal value is often slightly larger than that of C_{60} (1.452 Å). Still it seems that the carbon parts of the clusters are not much distorted by the presence of the boron atoms. The boron-carbon bond lengths lie in between the other two categories, as expected.

Finally the results of a Mulliken charge analysis for the same 15 configurations are presented in Fig. 10.4. It turned out that there is a strong correlation between the net charge of an atom and its surrounding. The x axis corresponds to the relative amount of opposite atom kinds being first neighbors to a given atom, i.e. a value of 0 means that a given atom is only surrounded by atoms of the same kind, whereas a value of 1 means that a given atom is only surrounded by atoms of the opposite kind. The y axis shows the net charge of the given atom.
Whereas for $x = 0$ – i.e. boron atoms are only surrounded by boron and carbon atoms only by carbon – the net charge is approximately zero for both kinds, they behave differently for increasing values of x. The higher the value of x is – i.e. boron atoms are more and more surrounded by carbon and carbon atoms more and more by boron, respectively –, the more the carbon atoms get negatively charged, whereas the boron atoms get positively charged. Furthermore it seems that these results are valid for both structural motifs – i.e. diluted and patched – as there are no notable deviations from the pattern.

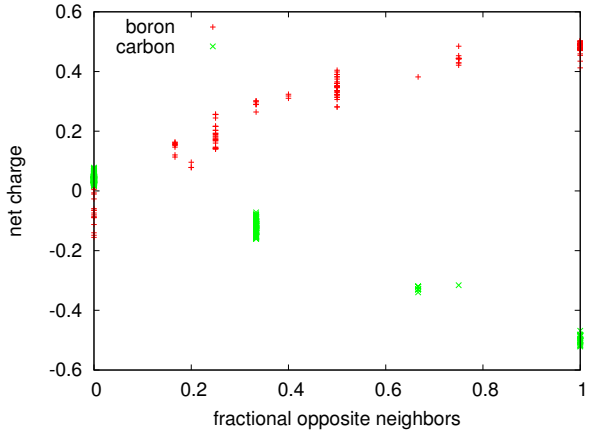


Figure 10.4: The net charge per atom, as calculated by the Mulliken charge analysis, for the 15 clusters shown in Fig. 10.2. The x coordinate denotes the fraction of opposite atom kinds surrounding a given atom, as explained in the text. There is a clear connection between the coordinate x and the magnitude of the net charge. Boron atoms get positively charged when the value of x is increased, whereas carbon atoms get negatively charged.

10.3 Energy landscape for $B_{12}C_{50}$

Looking at the energetically most favorable configuration of $B_{12}C_{48}$, it can be seen that the heptagon which is present in this compound could be filled up by adding two additional carbon atoms, as illustrated in Fig. 10.5. In this way the heptagon would be modified into one hexagon and one pentagon, and furthermore the two adjacent pentagons which are present twice on both sides of the heptagon are turned into a pentagon and a hexagon each. Since this configuration consists only of pentagons and hexagons and furthermore respects the isolated pentagon rule, it is plausible to assume that it is an excellent candidate for a global minimum of the stoichiometry $B_{12}C_{50}$.

In order to confirm this assumption an extended search for the ground state of these compounds was performed as well. The way how the initial structures were generated and the further procedure has already been explained in Sec. 10.1.

It turned out that the structure that was manually constructed from the ground state of $B_{12}C_{48}$ by adding the two carbon atoms as just described is the energetically most favorable configuration. The lower part of the energy spectrum – again up to 3 eV above the putative ground state – is shown in Fig. 10.6, together with figures of the 12 structures being lowest in energy and the two most favorable diluted configurations. Again the same coloring scheme as in Fig. 10.2 is used.
As for the case of $B_{12}C_{48}$ it is clear that all low lying minima correspond to the same structural motif where the boron is aggregated at one location, separating the surface of the cluster in a boron-rich and a boron-poor part. Furthermore it can be seen that once more the patch formed by the boron atoms is not compact, but rather frayed at the boundaries. As for the case of $B_{12}C_{48}$ the putative ground state is well separated from the first excited state; however the energy levels are slightly closer together this time.

The most favorable configurations belonging to the structural motif of the diluted clusters are only the 673rd and the 965th lowest structures, respectively, according to the ordering that was found; they exhibit an energy separation from the lowest one of

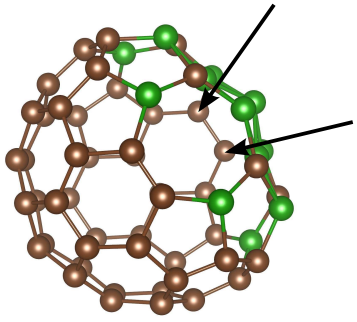


Figure 10.5: The energetically most favorable structure for the stoichiometry $B_{12}C_{48}$ contains a heptagon. Inserting two additional carbon atoms at the positions marked by arrows would turn the heptagon into a hexagon and a pentagon and furthermore modify the two adjacent pentagons which are present on both side of the heptagon into a pentagon and a hexagon each.

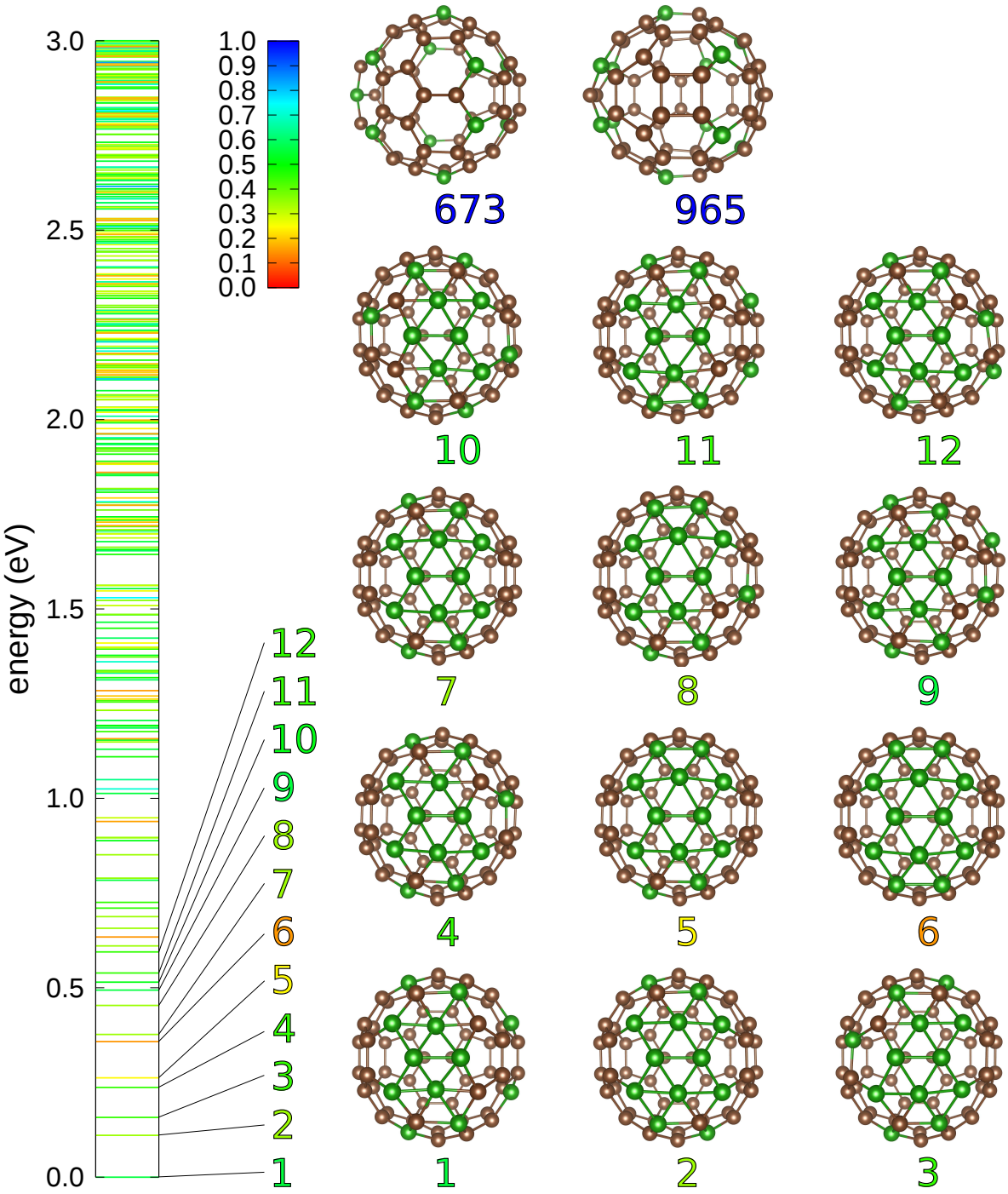


Figure 10.6: The lower part of the energy spectrum – only up to 3 eV above the putative ground state – of $B_{12}C_{50}$, together with figures of the 12 energetically lowest structures and the two most favorable diluted configurations. The energies of the latter ones are too high to be displayed on the spectrum. Again the same coloring scheme as in Fig. 10.2 is used. As for $B_{12}C_{48}$ it must be noted that the spectrum is most likely not complete.

Table 10.3: A more detailed description – analogous to Tab. 10.1 – of the 12 energetically most favorable configurations and the two lowest diluted structures of the stoichiometry $B_{12}C_{50}$, i.e. the ones depicted in Fig. 10.6: the energy separation ΔE to the putative ground state, the HOMO-LUMO gap, the formation energy ΔH with respect to the bulk phases of boron and carbon, the point group and the RMSD with respect to the putative ground state.

	ΔE (eV)	gap (eV)	ΔH $\left(\frac{\text{eV}}{\text{atom}}\right)$	PG	RMSD (bohr)
1st	0.000	0.534	0.322	C_s	0.000
2nd	0.110	0.416	0.324	C_2	1.363
3rd	0.158	0.434	0.325	C_1	1.209
4th	0.237	0.283	0.326	C_1	0.976
5th	0.262	0.338	0.327	C_1	1.285
6th	0.358	0.308	0.328	C_{2v}	1.390
7th	0.377	0.154	0.328	C_s	1.168
8th	0.453	0.215	0.330	C_1	1.113
9th	0.494	0.249	0.330	C_1	0.530
10th	0.515	0.380	0.331	C_1	1.451
11th	0.539	0.227	0.331	C_1	0.539
12th	0.595	0.363	0.332	C_1	0.931
673th	3.921	0.204	0.386	C_s	3.289
965th	5.227	0.458	0.407	C_s	3.080

already 3.9 eV and 5.2 eV. However it has to be emphasized that there was no specific search for diluted structures, as the focus was mainly put on determining the ground state, so it might well be that there are still some other ones which are slightly more favorable. Furthermore it must be noted again that the energy spectrum is most likely not complete – in particular for the higher energies – as the goal was not to explore the entire energy landscape, but only the low energy part.

In Tab. 10.3 some more details about the structures shown in Fig. 10.6 are presented, in analogy to Tab. 10.1.
The first column gives the energy separation ΔE of the configurations with respect to the lowest one. As can be seen the spacing between the energy levels is slightly smaller than for $B_{12}C_{48}$. On the other hand the difference to the diluted configurations is, as mentioned, considerably larger this time.

The second column gives the HOMO-LUMO gaps; compared to the values for $B_{12}C_{48}$ they are – except for the few lowest structures – tendentially smaller. However there is again no notable difference between the patched and the diluted structures.

The third column shows the formation energy per atom with respect to the bulk conformations of boron and carbon, respectively. These values are slightly lower than their counterparts for $B_{12}C_{48}$ in Tab. 10.1, indicating that it is more likely to encounter experimentally the stoichiometry $B_{12}C_{50}$ than $B_{12}C_{48}$.

The point groups of the configurations are presented in the fourth column. Comparing with the results for the stoichiometry $B_{12}C_{48}$ shown in Tab. 10.1 one can see that there is a wider variation of symmetry classes; the energetically most favorable structures have the point groups C_1, C_2, C_s and C_{2v}. On the other hand the diluted structures are of lower symmetry than their counterparts for $B_{12}C_{48}$ and exhibit only the point group C_s.

In the last column the RMSD of the configurations with respect to the energetically lowest one are presented. Again it can be seen that there is a clear separation between the patched and the diluted configurations, whereas there is no notable correlation between the energy and the RMSD within the class of the patched structures.

A more complete overview of the energy versus the RMSD of all structures up to a value of 8 eV above the putative ground state is shown in Fig. 10.7. The results are similar to those found for $B_{12}C_{48}$. Again there is a wide range which contains structures whose energies are completely uncorrelated to the value of the RMSD and which is sharply bounded at a bit less than 2 bohr; the configurations lying beyond this range are considerably less favorable. The structures exhibiting the largest RMSD values belong to a class where the boron atoms are arranged in a band-like manner around the cage and to the diluted class.

Tab. 10.4 shows the bond lengths for those 14 structures which were presented in detail in Fig. 10.6 and Tab 10.3; the results are similar to the ones obtained for $B_{12}C_{48}$. Again the minimal carbon-carbon bond lengths are close to the value in pure C_{60}, whereas the maximal ones are in general slightly larger; the boron-boron bonds exhibit a wider variation and their minimal and maximal values are often shorter and longer, respectively, than the corresponding values in B_{80}; and the carbon-boron bond lengths lie in between the other two numbers.

]

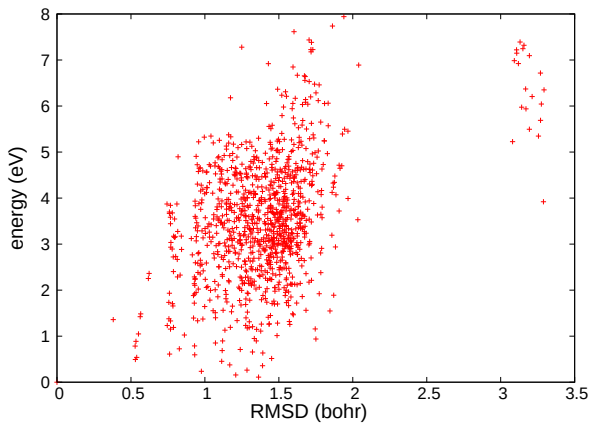


Figure 10.7: The energy difference versus the RMSD of all structures for the stoichiometry $B_{12}C_{50}$ up to 8 eV above the putative ground state. Again there is a broad range containing structures whose energies are completely uncorrelated to the value of the RMSD and which is limited by a sharp boundary at a bit less than 2 bohr. The structures exhibiting the largest RMSD belong to a class where the boron atoms are arranged in a band-like manner around the cage or to the diluted class.

Table 10.4: The minimal and maximal bond lengths for the 12 energetically lowest structures and the two most favorable diluted ones for the stoichiometry $B_{12}C_{50}$, i.e. the same configurations which are shown in Fig. 10.6 and whose details are presented in Tab. 10.3. For the diluted structures the boron-boron bond lengths are not meaningful.

	bond lengths (Å)					
	B-B		C-C		B-C	
	min	max	min	max	min	max
1st	1.664	1.749	1.384	1.479	1.537	1.664
2nd	1.670	1.752	1.393	1.482	1.536	1.645
3rd	1.665	1.790	1.393	1.489	1.517	1.693
4th	1.669	1.776	1.394	1.489	1.512	1.698
5th	1.666	1.751	1.394	1.496	1.535	1.630
6th	1.659	1.761	1.395	1.496	1.533	1.566
7th	1.665	1.750	1.390	1.487	1.532	1.661
8th	1.650	1.778	1.393	1.505	1.522	1.656
9th	1.651	1.776	1.389	1.485	1.525	1.674
10th	1.657	1.814	1.395	1.483	1.531	1.653
11th	1.592	1.807	1.392	1.474	1.544	1.729
12th	1.651	1.843	1.388	1.497	1.496	1.679
673th	—	—	1.375	1.492	1.521	1.588
965th	—	—	1.383	1.487	1.500	1.573

Finally in Fig. 10.8 the results of a Mulliken charge analysis for the same structures are presented. In analogy to Fig. 10.4 the net charge per atom is plotted as a function of the coordinate x denoting the fraction of opposite atom kinds surrounding a given atom. As for $B_{12}C_{48}$ there is once more a clear connection between the neighborhood of an atom and its net charge. Atoms which are only surrounded by the same kind are more or less neutral, whereas – with a magnitude which increases with the value of x – boron atoms being surrounded by carbon are positively and carbon atoms being surrounded by boron negatively charged.

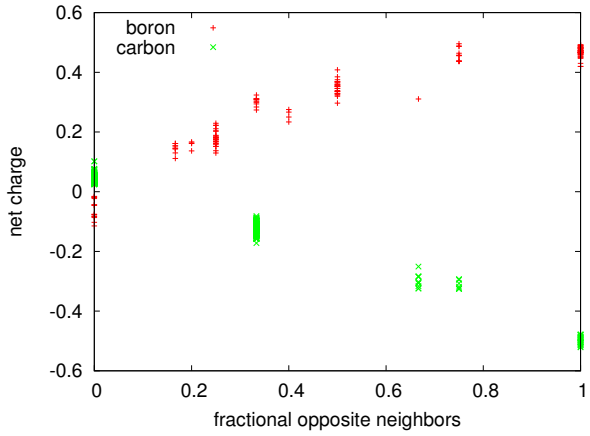


Figure 10.8: The net charge per atom – in analogy to Fig. 10.4 – as a function of the coordinate x denoting the fraction of opposite atom kinds surrounding the given atom for the 14 structures shown in Fig. 10.6. There is a clear connection between the value of x and the magnitude of the net charge. Carbon atoms surrounded by boron are negatively charged, whereas boron atoms surrounded by carbon are positively charged.

CHAPTER 11

Conclusions and outlook

An extended study of the heterofullerene $B_{12}C_{48}$ revealed many new minima that are considerably lower in energy than those which have been proposed so far. In addition it turned out that the energetically most favorable configurations belong to a completely new structural motif than those previously known. Whereas up to now it has been believed that the boron atoms should be distributed over the entire cage and isolated – referred to as diluted –, it seems that it is more favorable if they are aggregated in a single patch on the surface being slightly frayed at the boundaries. This demonstrates that the ground state for heterofullerenes is not necessarily related to the Buckminster fullerene.

Starting from the lowest configuration of $B_{12}C_{48}$ an extensive survey of the energy landscape for the stoichiometry $B_{12}C_{50}$ was performed as well. Also here the new structural motif of the patched structures is more favorable than that of the diluted ones. Since this stoichiometry exhibits a lower formation energy than $B_{12}C_{48}$, it should be more likely to encounter it experimentally.

Calculations of the HOMO-LUMO gaps demonstrated that the structures which were found for both $B_{12}C_{48}$ and $B_{12}C_{50}$ are insulators, however with small gaps of considerably less than 1 eV. A Mulliken charge analysis showed that there is a strong correlation between the net charge of a given atom and its surrounding, leading to an increasingly positive (negative) charge of a boron (carbon) atom the more it is surrounded by atoms of the opposite kind.

In a broader context these findings show that doping in sp^2-materials is not yet well understood and that no universally valid rules are available to predict which struc-

tural motifs are the most stable ones in such doped structures. The results could also give guidance to synthesis efforts [121] for such heterofullerenes. The steep rise of the energy of metastable configurations as a function of the distance from the ground state shown in Figs. 10.3 and 10.7 indicates that there is a substantial driving force towards low energy motifs with patches and suggests that a synthesis of patched structures should be possible. The energy gap between the ground state and the lowest metastable state (0.2 eV for $B_{12}C_{48}$ and 0.1 eV for $B_{12}C_{50}$) is however much smaller than in C_{60} (1.6 eV) and reaching the ground state might therefore be difficult.
A synthesis procedure based on the substitution [93] of carbon atoms by boron is unlikely to succeed for heterofullerenes containing a larger number of boron atoms which then form patches. Planar boron clusters that are structurally similar to the boron patches found in the ground states can however be synthesized experimentally [122] and might form growth nuclei for such heterofullerenes in a carbon rich spark or vapor chamber.

APPENDIX A

Calculation of the wavelet filters for different operators

This section describes in detail how the wavelets filters of various operators $\hat{\mathcal{O}}$ can be calculated for an orthogonal wavelet family. It is only necessary to calculate the filters for the scaling functions, i.e. $\langle\phi|\hat{\mathcal{O}}|\phi\rangle$; the ones for the other possibilities – i.e. $\langle\psi|\hat{\mathcal{O}}|\phi\rangle$, $\langle\phi|\hat{\mathcal{O}}|\psi\rangle$ and $\langle\psi|\hat{\mathcal{O}}|\psi\rangle$ – can be derived from the first ones.

A.1 Derivative filters

The first class of filters which are calculated are the derivative filters of arbitrary order l, i.e. the operator is given by $\hat{\mathcal{O}} = \frac{\partial^l}{\partial x^l}$. The specific value of l is not important for the derivation and the final result is valid for any l [69, 123].

A.1.1 The basic filter

The basic filter among the scaling functions is denoted by $a_i = \langle\phi(x-i)|\frac{\partial^l}{\partial x^l}|\phi(x)\rangle$. It is more convenient in the following to use an integral notation and therefore to write $a_i = \int \phi(x-i)\frac{\partial^l}{\partial x^l}\phi(x)\,\mathrm{d}x$.

Using – according to Eq. (4.9) – the refinement relation $\phi(x) = \sqrt{2}\sum_\mu h_\mu \phi(2x-\mu)$ and applying the variable substitutions $x' = 2x$ and $x'' = x' - \mu$ this leads to

$$\begin{aligned} a_i =& \int \phi(x-i)\frac{\partial^l}{\partial x^l}\phi(x)\,\mathrm{d}x \\ =& 2\sum_{\nu,\mu} h_\nu h_\mu \int \phi(2x-2i-\nu)\frac{\partial^l}{\partial x^l}\phi(2x-\mu)\,\mathrm{d}x \\ =& 2\sum_{\nu,\mu} h_\nu h_\mu 2^{l-1} \int \phi(x'-2i-\nu)\frac{\partial^l}{\partial x'^l}\phi(x'-\mu)\,\mathrm{d}x' \\ =& 2^l \sum_{\nu,\mu} h_\nu h_\mu \int \phi(x''-2i-\nu+\mu)\frac{\partial^l}{\partial x''^l}\phi(x'')\,\mathrm{d}x'' \\ =& 2^l \sum_{\nu,\mu} h_\nu h_\mu a_{2i+\nu-\mu}. \end{aligned} \tag{A.1}$$

This means that the filter values a_i are the elements of the eigenvector $\mathbf{a}$ associated with the eigenvalue 2^{-l},

$$\sum_j A_{ij} a_j = 2^{-l} a_i, \tag{A.2}$$

where the matrix elements A_{ij} are given by

$$A_{ij} = \sum_{\nu,\mu} h_\nu h_\mu \delta_{j,2i+\nu-\mu}. \tag{A.3}$$

A.1.2 The remaining filters

The calculation of the filter element $b_i = \langle \psi(x-i) | \frac{\partial^l}{\partial x^l} | \phi(x) \rangle$ can be calculated from the values of the filter a. Again using the refinement relations $\phi(x) = \sqrt{2}\sum_\mu h_\mu \phi(2x-\mu)$ and $\psi(x) = \sqrt{2}\sum_\mu g_\mu \phi(2x-\mu)$ one gets

$$\begin{aligned} b_i =& \int \psi(x-i)\frac{\partial^l}{\partial x^l}\phi(x)\,\mathrm{d}x \\ =& 2\sum_{\nu,\mu} g_\nu h_\mu \int \phi(2x-2i-\nu)\frac{\partial^l}{\partial x^l}\phi(2x-\mu)\,\mathrm{d}x \\ =& 2\sum_{\nu,\mu} g_\nu h_\mu 2^{l-1} \int \phi(x'-2i-\nu)\frac{\partial^l}{\partial x'^l}\phi(x'-\mu)\,\mathrm{d}x' \\ =& 2^l \sum_{\nu,\mu} g_\nu h_\mu \int \phi(x''-2i-\nu+\mu)\frac{\partial^l}{\partial x''^l}\phi(x'')\,\mathrm{d}x''. \end{aligned} \tag{A.4}$$

The integral $\int \phi(x'' - 2i - \nu + \mu)\frac{\partial^l}{\partial x''^l}\phi(x'')\,\mathrm{d}x''$ has been calculated previously, and one thus gets the relation

$$b_i = 2^l \sum_{\nu,\mu} g_\nu h_\mu a_{2i+\nu-\mu}. \tag{A.5}$$

The filters c and e can be derived along the same lines; the final result is given by

$$c_i = \int \phi(x-i)\frac{\partial^l}{\partial x^l}\psi(x)\,\mathrm{d}x = 2^l \sum_{\nu,\mu} h_\nu g_\mu a_{2i+\nu-\mu}, \tag{A.6}$$

$$e_i = \int \psi(x-i)\frac{\partial^l}{\partial x^l}\psi(x)\,\mathrm{d}x = 2^l \sum_{\nu,\mu} g_\nu g_\mu a_{2i+\nu-\mu}. \tag{A.7}$$

A.1.3 The filters for the general case

The above filters were derived for the special case where the grid spacing is equal to 1 and one scaling function or wavelet is located at the origin. The most general case is to consider the filters for variable grid spacings and positions, i.e.

$$a_{i,j;h} = \int \phi\left(\frac{x}{h} - i\right)\frac{\partial^l}{\partial x^l}\phi\left(\frac{x}{h} - j\right)\,\mathrm{d}x. \tag{A.8}$$

Using the variable substitution $x' = \frac{x}{h} - j$, which implies $\frac{\partial}{\partial x} = \frac{\partial}{\partial x'}\frac{\partial x'}{\partial x} = \frac{1}{h}\frac{\partial}{\partial x'}$ and $\mathrm{d}x = h\mathrm{d}x'$, this can be reduced to the general case

$$a_{i,j;h} = h^{-l+1}\int \phi\left(x' + j - i\right)\frac{\partial^l}{\partial x'^l}\phi\left(x'\right)\,\mathrm{d}x' = h^{-l+1}a_{i-j}. \tag{A.9}$$

This demonstrates that the value of the filter depends only on the difference $i - j$. The same arguments apply as well for the other filters:

$$b_{i,j;h} = h^{-l+1}\int \psi\left(\frac{x}{h} - i\right)\frac{\partial^l}{\partial x^l}\phi\left(\frac{x}{h} - j\right)\,\mathrm{d}x = h^{-l+1}b_{i-j}, \tag{A.10}$$

$$c_{i,j;h} = h^{-l+1}\int \phi\left(\frac{x}{h} - i\right)\frac{\partial^l}{\partial x^l}\psi\left(\frac{x}{h} - j\right)\,\mathrm{d}x = h^{-l+1}c_{i-j}, \tag{A.11}$$

$$e_{i,j;h} = h^{-l+1}\int \psi\left(\frac{x}{h} - i\right)\frac{\partial^l}{\partial x^l}\psi\left(\frac{x}{h} - j\right)\,\mathrm{d}x = h^{-l+1}e_{i-j}. \tag{A.12}$$

From these results it follows that the filters fulfill the following symmetry relations:

$$\begin{aligned} a_{i-j} &= a_{j-i}, & b_{i-j} &= c_{j-i}, \\ c_{i-j} &= b_{j-i}, & e_{i-j} &= e_{j-i}. \end{aligned} \tag{A.13}$$

A.2 Position operators filters

A.2.1 The basic filters

The ultimate goal is to calculate the filter for the operator x^4, i.e. the matrix element $\langle \phi(x-i) \,|\, x^4 \,|\, \phi(x) \rangle$. To this end one first has to derive the filters corresponding to the operators x, x^2 and x^3.

A.2.1.1 Basic filter – the linear operator

The procedure is analogous to the calculation of the filter for the derivative operator, i.e. one uses the refinement relation $\phi(x) = \sqrt{2}\sum_\mu h_\mu \phi(2x-\mu)$. However, since the operator x is – in contrast to $\frac{\partial^l}{\partial x^l}$ – not translational invariant, the variable substitution which is done in the course of the calculation introduces some additional terms. Denoting by $a_i^{(1)} = \int \phi(x-i)x\phi(x)\,\mathrm{d}x$ the basic filter one gets

$$\begin{aligned}
a_i^{(1)} =& \int \phi(x-i)x\phi(x)\,\mathrm{d}x \\
=& 2\sum_{\nu,\mu} h_\nu h_\mu \int \phi(2x-2i-\nu)x\phi(2x-\mu)\,\mathrm{d}x \\
=& \frac{1}{2}\sum_{\nu,\mu} h_\nu h_\mu \int \phi(x'-2i-\nu)x'\phi(x'-\mu)\,\mathrm{d}x' \\
=& \frac{1}{2}\sum_{\nu,\mu} h_\nu h_\mu \int \phi(x''-2i-\nu+\mu)(x''+\mu)\phi(x'')\,\mathrm{d}x' \\
=& \frac{1}{2}\sum_{\lambda,\mu} h_{\lambda+\mu-2i} h_\mu \int \phi(x''-\lambda)(x''+\mu)\phi(x'')\,\mathrm{d}x'' \\
=& \frac{1}{2}\sum_{\lambda,\mu} h_{\lambda+\mu-2i} h_\mu \left[\int \phi(x''-\lambda)x''\phi(x'')\,\mathrm{d}x'' + \mu\int \phi(x''-\lambda)\phi(x'')\,\mathrm{d}x''\right].
\end{aligned} \tag{A.14}$$

Using the orthonormality relation $\int \phi(x-\lambda)\phi(x)\,\mathrm{d}x = \delta_{\lambda 0}$ this leads to

$$\begin{aligned}
a_i^{(1)} =& \frac{1}{2}\sum_{\lambda,\mu} h_{\lambda+\mu-2i} h_\mu a_\lambda^{(1)} + \frac{1}{2}\sum_{\lambda,\mu} \mu h_{\lambda+\mu-2i} h_\mu \delta_{\lambda 0} \\
=& \sum_\lambda M_{i\lambda} a_\lambda^{(1)} + c_i
\end{aligned} \tag{A.15}$$

with the shorthand notations

$$M_{i\lambda} = \frac{1}{2}\sum_\mu h_{\lambda+\mu-2i} h_\mu, \quad c_i = \frac{1}{2}\sum_\mu \mu h_{\mu-2i} h_\mu. \tag{A.16}$$

So one gets the relation

$$\sum_{\lambda} (\delta_{i\lambda} - M_{i\lambda})\, a_{\lambda}^{(1)} = c_i, \tag{A.17}$$

which means that the filter elements a_i are the solution of the linear system of equations

$$\mathbf{Sa} = \mathbf{c} \qquad \text{with} \quad S_{i\lambda} = \delta_{i\lambda} - M_{i\lambda}. \tag{A.18}$$

A.2.1.2 Basic filter – the quadratic operator

The next step is to calculate the filter for the operator x^2. As will turn out its evaluation requires the results of the previous section, i.e. the filter $a_i^{(1)}$. Denoting the basic filter by $a_i^{(2)}$ one gets

$$\begin{aligned}
a_i^{(2)} =& \int \phi(x-i)x^2\phi(x)\,\mathrm{d}x \\
=& 2\sum_{\nu,\mu} h_\nu h_\mu \int \phi(2x-2i-\nu)x^2\phi(2x-\mu)\,\mathrm{d}x \\
=& \frac{1}{4}\sum_{\nu,\mu} h_\nu h_\mu \int \phi(x'-2i-\nu)x'^2\phi(x'-\mu)\,\mathrm{d}x' \\
=& \frac{1}{4}\sum_{\nu,\mu} h_\nu h_\mu \int \phi(x''-2i-\nu+\mu)(x''+\mu)^2\phi(x'')\,\mathrm{d}x' \\
=& \frac{1}{4}\sum_{\lambda,\mu} h_{\lambda+\mu-2i} h_\mu \int \phi(x''-\lambda)(x''+\mu)^2\phi(x'')\,\mathrm{d}x'' \\
=& \frac{1}{4}\sum_{\lambda,\mu} h_{\lambda+\mu-2i} h_\mu \left[\int \phi(x''-\lambda)x''^2\phi(x'')\,\mathrm{d}x'' + 2\mu\int \phi(x''-\lambda)x''\phi(x'')\,\mathrm{d}x''\right. \\
& \left. +\mu^2\int \phi(x''-\lambda)\phi(x'')\,\mathrm{d}x''\right].
\end{aligned} \tag{A.19}$$

Again using the orthonormality relation $\int \phi(x-\lambda)\phi(x)\,\mathrm{d}x = \delta_{\lambda 0}$ and the results for the calculation of the operator x this leads to

$$\begin{aligned}
a_i^{(2)} =& \frac{1}{4}\sum_{\lambda,\mu} h_{\lambda+\mu-2i} h_\mu a_\lambda^{(2)} + \frac{1}{2}\sum_{\lambda,\mu} \mu h_{\lambda+\mu-2i} h_\mu a_\lambda^{(1)} + \frac{1}{4}\sum_{\lambda,\mu} \mu^2 h_{\lambda+\mu-2i} h_\mu \delta_{\lambda 0} \\
=& \sum_{\lambda} M_{i\lambda} a_\lambda^{(2)} + b_i + c_i
\end{aligned} \tag{A.20}$$

with the shorthand notations

$$M_{i\lambda} = \frac{1}{4}\sum_{\mu} h_{\lambda+\mu-2i} h_\mu, \quad b_i = \frac{1}{2}\sum_{\lambda,\mu} \mu h_{\lambda+\mu-2i} h_\mu a_\lambda^{(1)}, \qquad c_i = \frac{1}{4}\sum_{\mu} \mu^2 h_{\mu-2i} h_\mu. \tag{A.21}$$

So there is again a similar relation as before, namely

$$\sum_{\lambda} (\delta_{i\lambda} - M_{i\lambda})\, a_{\lambda}^{(2)} = b_i + c_i, \tag{A.22}$$

which means that one has to solve the linear system of equations

$$\mathbf{Sa} = \mathbf{d} \qquad \text{with} \quad S_{i\lambda} = \delta_{i\lambda} - M_{i\lambda}, \quad d_i = b_i + c_i. \tag{A.23}$$

A.2.1.3 Basic filter – the cubic operator

Now one can proceed to the operator x^3, which again requires the results of the previous sections. Denoting the basic filter by $a_i^{(3)}$ one gets

$$\begin{aligned}
a_i^{(3)} =& \int \phi(x-i)x^3\phi(x)\,\mathrm{d}x \\
=& 2\sum_{\nu,\mu} h_\nu h_\mu \int \phi(2x-2i-\nu)x^3\phi(2x-\mu)\,\mathrm{d}x \\
=& \frac{1}{8}\sum_{\nu,\mu} h_\nu h_\mu \int \phi(x'-2i-\nu)x'^3\phi(x'-\mu)\,\mathrm{d}x' \\
=& \frac{1}{8}\sum_{\nu,\mu} h_\nu h_\mu \int \phi(x''-2i-\nu+\mu)(x''+\mu)^3\phi(x'')\,\mathrm{d}x' \\
=& \frac{1}{8}\sum_{\lambda,\mu} h_{\lambda+\mu-2i} h_\mu \int \phi(x''-\lambda)(x''+\mu)^3\phi(x'')\,\mathrm{d}x'' \\
=& \frac{1}{8}\sum_{\lambda,\mu} h_{\lambda+\mu-2i} h_\mu \left[\int \phi(x''-\lambda)x''^3\phi(x'')\,\mathrm{d}x'' + 3\mu\int \phi(x''-\lambda)x''^2\phi(x'')\,\mathrm{d}x''\right. \\
& \left. +3\mu^2\int \phi(x''-\lambda)x''\phi(x'')\,\mathrm{d}x'' + \mu^3\int \phi(x''-\lambda)\phi(x'')\,\mathrm{d}x''\right].
\end{aligned} \tag{A.24}$$

Using the orthonormality relation $\int \phi(x-\lambda)\phi(x)\,\mathrm{d}x = \delta_{\lambda 0}$ this leads to

$$\begin{aligned}
=& \frac{1}{8}\sum_{\lambda,\mu} h_{\lambda+\mu-2i} h_\mu a_\lambda^{(3)} + \frac{3}{8}\sum_{\lambda,\mu} \mu h_{\lambda+\mu-2i} h_\mu a_\lambda^{(2)} + \\
& + \frac{3}{8}\sum_{\lambda,\mu} \mu^2 h_{\lambda+\mu-2i} h_\mu a_\lambda^{(1)} + \frac{1}{8}\sum_{\lambda,\mu} \mu^3 h_{\lambda+\mu-2i} h_\mu \delta_{0\lambda} \\
=& \sum_{\lambda} M_{i\lambda} a_\lambda^{(3)} + b_i + c_i + d_i
\end{aligned} \tag{A.25}$$

with the shorthand notations

$$\begin{aligned} M_{i\lambda} &= \frac{1}{8}\sum_{\mu} h_{\lambda+\mu-2i}h_{\mu}, \quad b_i = \frac{3}{8}\sum_{\lambda,\mu} \mu h_{\lambda+\mu-2i}h_{\mu}a_{\lambda}^{(2)}, \\ c_i &= \frac{3}{8}\sum_{\lambda,\mu} \mu^2 h_{\lambda+\mu-2i}h_{\mu}a_{\lambda}^{(1)}, \quad d_i = \frac{1}{8}\sum_{\mu} \mu^3 h_{\mu-2i}h_{\mu}. \end{aligned} \tag{A.26}$$

Thus there is again a similar relation as before, namely

$$\sum_{\lambda} (\delta_{i\lambda} - M_{i\lambda})\, a_{\lambda}^{(3)} = b_i + c_i + d_i, \tag{A.27}$$

meaning that one has to solve the linear system of equations

$$\mathbf{Sa} = \mathbf{e} \qquad \text{with} \quad S_{i\lambda} = \delta_{i\lambda} - M_{i\lambda}, \quad e_i = b_i + c_i + d_i. \tag{A.28}$$

A.2.1.4 Basic filter – the quartic operator

Now it is finally possible to calculate the matrix elements for the operator x^4. Denoting the basic filter by $a_i^{(4)}$ one gets

$$\begin{aligned} a_i^{(4)} =& \int \phi(x-i)x^4\phi(x)\,\mathrm{d}x \\ =& 2\sum_{\nu,\mu} h_\nu h_\mu \int \phi(2x-2i-\nu)x^4\phi(2x-\mu)\,\mathrm{d}x \\ =& \frac{1}{16}\sum_{\nu,\mu} h_\nu h_\mu \int \phi(x'-2i-\nu)x'^4\phi(x'-\mu)\,\mathrm{d}x' \\ =& \frac{1}{16}\sum_{\nu,\mu} h_\nu h_\mu \int \phi(x''-2i-\nu+\mu)(x''+\mu)^4\phi(x'')\,\mathrm{d}x' \\ =& \frac{1}{16}\sum_{\lambda,\mu} h_{\lambda+\mu-2i} h_\mu \int \phi(x''-\lambda)(x''+\mu)^4\phi(x'')\,\mathrm{d}x'' \\ =& \frac{1}{16}\sum_{\lambda,\mu} h_{\lambda+\mu-2i} h_\mu \left[\int \phi(x''-\lambda)x''^4\phi(x'')\,\mathrm{d}x'' + 4\mu\int \phi(x''-\lambda)x''^3\phi(x'')\,\mathrm{d}x''\right. \\ & + 6\mu^2\int \phi(x''-\lambda)x''^2\phi(x'')\,\mathrm{d}x'' + 4\mu^3\int \phi(x''-\lambda)x''\phi(x'')\,\mathrm{d}x'' \\ & \left. + \mu^4\int \phi(x''-\lambda)\phi(x'')\,\mathrm{d}x''\right]. \end{aligned} \tag{A.29}$$

Using the orthonormality relation $\int \phi(x-\lambda)\phi(x)\,\mathrm{d}x = \delta_{\lambda 0}$ this leads to

$$\begin{aligned} a_i^{(4)} =& \frac{1}{16}\sum_{\lambda,\mu} h_{\lambda+\mu-2i}h_\mu a_\lambda^{(4)} + \frac{1}{4}\sum_{\lambda,\mu}\mu h_{\lambda+\mu-2i}h_\mu a_\lambda^{(3)} + \frac{3}{8}\sum_{\lambda,\mu}\mu^2 h_{\lambda+\mu-2i}h_\mu a_\lambda^{(2)} \\ &+ \frac{1}{4}\sum_{\lambda,\mu}\mu^3 h_{\lambda+\mu-2i}h_\mu a_\lambda^{(1)} + \frac{1}{16}\sum_{\lambda,\mu}\mu^4 h_{\lambda+\mu-2i}h_\mu \delta_{0\lambda} \\ =& \sum_k M_{ik}a_k^{(4)} + b_i + c_i + d_i + e_i \end{aligned} \tag{A.30}$$

with the shorthand notations

$$\begin{aligned} M_{i\lambda} &= \frac{1}{16}\sum_\mu h_{\lambda+\mu-2i}h_\mu, \quad b_i = \frac{1}{4}\sum_{\lambda,\mu}\mu h_{\lambda+\mu-2i}h_\mu a_\lambda^{(3)}, \\ c_i &= \frac{3}{8}\sum_{\lambda,\mu}\mu^2 h_{\lambda+\mu-2i}h_\mu a_\lambda^{(2)}, \quad d_i = \frac{1}{4}\sum_{\lambda,\mu}\mu^3 h_{\lambda+\mu-2i}h_\mu a_\lambda^{(1)}, \\ e_i &= \frac{1}{16}\sum_\mu \mu^4 h_{\mu-2i}h_\mu. \end{aligned} \tag{A.31}$$

From this one gets the relation

$$\sum_\lambda (\delta_{i\lambda} - M_{i\lambda})\, a_\lambda^{(4)} = b_i + c_i + d_i + e_i, \tag{A.32}$$

which means that one has to solve the linear system of equations

$$\mathbf{Sa} = \mathbf{f} \qquad \text{with} \quad S_{i\lambda} = \delta_{i\lambda} - M_{i\lambda}, \quad f_i = b_i + c_i + d_i + e_i. \tag{A.33}$$

A.2.2 The remaining filters

The remaining filters can be derived from the basic ones as it was the case for the derivative filters. However there are this time some additional terms stemming from the fact that the operators are not translational invariant.

A.2.2.1 Remaining filters – the linear operator

To derive the filter $b_i^{(1)} = \langle\psi|x|\phi\rangle$ one can again use the refinement relations $\phi(x) = \sqrt{2}\sum_\mu h_\mu\phi(2x-\mu)$ and $\psi(x) = \sqrt{2}\sum_\mu g_\mu\phi(2x-\mu)$. This leads to

$$\begin{aligned}
b_i^{(1)} =& \int \psi(x-i)x\phi(x)\,\mathrm{d}x \\
=& 2\sum_{\nu,\mu} g_\nu h_\mu \int \phi(2x-2i-\nu)x\phi(2x-\mu)\,\mathrm{d}x \\
=& \frac{1}{2}\sum_{\nu,\mu} g_\nu h_\mu \int \phi(x'-2i-\nu)x'\phi(x'-\mu)\,\mathrm{d}x' \\
=& \frac{1}{2}\sum_{\nu,\mu} g_\nu h_\mu \int \phi(x''-2i-\nu+\mu)(x''+\mu)\phi(x'')\,\mathrm{d}x'' \\
=& \frac{1}{2}\sum_{\nu,\mu} g_\nu h_\mu \left[\int \phi(x''-2i-\nu+\mu)x''\phi(x'')\,\mathrm{d}x''\right. \\
& \left. +\mu\int \phi(x''-2i-\nu+\mu)\phi(x'')\,\mathrm{d}x''\right].
\end{aligned} \tag{A.34}$$

The value of the integral $\int \phi(x''-2i-\nu+\mu)x''\phi(x'')\,\mathrm{d}x''$ has been determined previously. Using furthermore the orthonormality of the scaling functions one thus gets

$$b_i^{(1)} = \frac{1}{2}\sum_{\nu,\mu} g_\nu h_\mu \left[a^{(1)}_{2i+\nu-\mu} + \mu\delta_{2i+\nu-\mu,0}\right]. \tag{A.35}$$

The filters $c_i^{(1)}$ and $e_i^{(1)}$ can be derived along the same lines, leading to

$$c_i^{(1)} = \int \phi(x-i)x\psi(x)\,\mathrm{d}x = \frac{1}{2}\sum_{\nu,\mu} h_\nu g_\mu \left[a^{(1)}_{2i+\nu-\mu} + \mu\delta_{2i+\nu-\mu,0}\right], \tag{A.36}$$

$$e_i^{(1)} = \int \psi(x-i)x\psi(x)\,\mathrm{d}x = \frac{1}{2}\sum_{\nu,\mu} g_\nu g_\mu \left[a^{(1)}_{2i+\nu-\mu} + \mu\delta_{2i+\nu-\mu,0}\right]. \tag{A.37}$$

A.2.2.2 Remaining filters – the quadratic operator

The derivation of the remaining filters for the operator x^2 is completely analogous to the case of the operator x:

$$\begin{aligned}
b_i^{(2)} &= \int \psi(x-i)x^2\phi(x)\,\mathrm{d}x \\
&= 2\sum_{\nu,\mu} g_\nu h_\mu \int \phi(2x-2i-\nu)x^2\phi(2x-\mu)\,\mathrm{d}x \\
&= \frac{1}{4}\sum_{\nu,\mu} g_\nu h_\mu \int \phi(x'-2i-\nu)x'^2\phi(x'-\mu)\,\mathrm{d}x' \\
&= \frac{1}{4}\sum_{\nu,\mu} g_\nu h_\mu \int \phi(x'-2i-\nu+\mu)(x''+\mu)^2\phi(x'')\,\mathrm{d}x'' \\
&= \frac{1}{4}\sum_{\nu,\mu} g_\nu h_\mu \left[\int \phi(x'-2i-\nu+\mu)x''^2\phi(x'')\,\mathrm{d}x''\right. \\
&\qquad +2\mu\int \phi(x'-2i-\nu+\mu)x''\phi(x'')\,\mathrm{d}x'' \\
&\qquad \left.+\mu^2\int \phi(x'-2i-\nu+\mu)\phi(x'')\,\mathrm{d}x''\right].
\end{aligned} \tag{A.38}$$

The two integrals have already been determined. Together with the orthonormality of the scaling functions this yields

$$b_i^{(2)} = \frac{1}{4}\sum_{\nu,\mu} g_\nu h_\mu \left[a^{(2)}_{2i+\nu-\mu} + 2\mu a^{(1)}_{2i+\nu-\mu} + \mu^2\delta_{2i+\nu-\mu,0}\right]. \tag{A.39}$$

In a similar way one gets for the other two filters

$$c_i^{(2)} = \int \phi(x-i)x^2\psi(x)\,\mathrm{d}x = \frac{1}{4}\sum_{\nu,\mu} h_\nu g_\mu \left[a^{(2)}_{2i+\nu-\mu} + 2\mu a^{(1)}_{2i+\nu-\mu} + \mu^2\delta_{2i+\nu-\mu,0}\right], \tag{A.40}$$

$$e_i^{(2)} = \int \psi(x-i)x^2\psi(x)\,\mathrm{d}x = \frac{1}{4}\sum_{\nu,\mu} g_\nu g_\mu \left[a^{(2)}_{2i+\nu-\mu} + 2\mu a^{(1)}_{2i+\nu-\mu} + \mu^2\delta_{2i+\nu-\mu,0}\right]. \tag{A.41}$$

A.2.2.3 Remaining filters – the cubic operator

The remaining filters for the operator x^3 can be derived along the same lines:

$$\begin{aligned}
b_i^{(3)} =& \int \psi(x-i)x^3\phi(x)\,\mathrm{d}x \\
=& 2\sum_{\nu,\mu} g_\nu h_\mu \int \phi(2x-2i-\nu)x^3\phi(2x-\mu)\,\mathrm{d}x \\
=& \frac{1}{8}\sum_{\nu,\mu} g_\nu h_\mu \int \phi(x'-2i-\nu)x'^3\phi(x'-\mu)\,\mathrm{d}x' \\
=& \frac{1}{8}\sum_{\nu,\mu} g_\nu h_\mu \int \phi(x'-2i-\nu+\mu)(x''+\mu)^3\phi(x'')\,\mathrm{d}x'' \\
=& \frac{1}{8}\sum_{\nu,\mu} g_\nu h_\mu \left[\int \phi(x'-2i-\nu+\mu)x''^3\phi(x'')\,\mathrm{d}x''\right. \\
& +3\mu \int \phi(x'-2i-\nu+\mu)x''^2\phi(x'')\,\mathrm{d}x'' \\
& +3\mu^2 \int \phi(x'-2i-\nu+\mu)x''\phi(x'')\,\mathrm{d}x'' \\
& \left.+\mu^3 \int \phi(x'-2i-\nu+\mu)\phi(x'')\,\mathrm{d}x''\right].
\end{aligned} \tag{A.42}$$

Again using the previous results and the orthonormality leads to

$$b_i^{(3)} = \frac{1}{8}\sum_{\nu,\mu} g_\nu h_\mu \left[a^{(3)}_{2i+\nu-\mu} + 3\mu a^{(2)}_{2i+\nu-\mu} + 3\mu^2 a^{(1)}_{2i+\nu-\mu} + \mu^3\delta_{2i+\nu-\mu,0}\right]. \tag{A.43}$$

In a similar way one gets for the other two filters

$$\begin{aligned}
c_i^{(3)} =& \int \phi(x-i)x^3\psi(x)\,\mathrm{d}x \\
=& \frac{1}{8}\sum_{\nu,\mu} h_\nu g_\mu \left[a^{(3)}_{2i+\nu-\mu} + 3\mu a^{(2)}_{2i+\nu-\mu} + 3\mu^2 a^{(1)}_{2i+\nu-\mu} + \mu^3\delta_{2i+\nu-\mu,0}\right], \qquad \text{(A.44)} \\
e_i^{(3)} =& \int \psi(x-i)x^3\psi(x)\,\mathrm{d}x \\
=& \frac{1}{8}\sum_{\nu,\mu} g_\nu g_\mu \left[a^{(3)}_{2i+\nu-\mu} + 3\mu a^{(2)}_{2i+\nu-\mu} + 3\mu^2 a^{(1)}_{2i+\nu-\mu} + \mu^3\delta_{2i+\nu-\mu,0}\right]. \qquad \text{(A.45)}
\end{aligned}$$

A.2.2.4 Remaining filters – the quartic operator

What remains is the calculation of the remaining filters for the operator x^4. Proceeding in an analogous way as for the other cases leads to

$$\begin{aligned}
b_i^{(4)} =& \int \psi(x-i)x^4\phi(x)\,\mathrm{d}x \\
=& 2\sum_{\nu,\mu} g_\nu h_\mu \int \phi(2x-2i-\nu)x^4\phi(2x-\mu)\,\mathrm{d}x \\
=& \frac{1}{16}\sum_{\nu,\mu} g_\nu h_\mu \int \phi(x'-2i-\nu)x'^4\phi(x'-\mu)\,\mathrm{d}x' \\
=& \frac{1}{16}\sum_{\nu,\mu} g_\nu h_\mu \int \phi(x'-2i-\nu+\mu)(x''+\mu)^4\phi(x'')\,\mathrm{d}x'' \\
=& \frac{1}{16}\sum_{\nu,\mu} g_\nu h_\mu \left[\int \phi(x'-2i-\nu+\mu)x''^4\phi(x'')\,\mathrm{d}x''\right. \\
&+4\mu \int \phi(x'-2i-\nu+\mu)x''^3\phi(x'')\,\mathrm{d}x'' \\
&+6\mu^2 \int \phi(x'-2i-\nu+\mu)x''^2\phi(x'')\,\mathrm{d}x'' \\
&+4\mu^3 \int \phi(x'-2i-\nu+\mu)x''\phi(x'')\,\mathrm{d}x'' \\
&\left.+\mu^4 \int \phi(x'-2i-\nu+\mu)\phi(x'')\,\mathrm{d}x''\right].
\end{aligned} \tag{A.46}$$

With the previous results and the orthonormality condition this gives

$$b_i^{(4)} = \frac{1}{16}\sum_{\nu,\mu} g_\nu h_\mu \left[a^{(4)}_{2i+\nu-\mu} + 4\mu a^{(3)}_{2i+\nu-\mu} + 6\mu^2 a^{(2)}_{2i+\nu-\mu} + 4\mu^3 a^{(1)}_{2i+\nu-\mu} + \mu^4\delta_{2i+\nu-\mu,0}\right]. \tag{A.47}$$

The other two filters can be derived along the same lines, leading to

$$\begin{aligned}
c_i^{(4)} =& \int \phi(x-i)x^4\psi(x)\,\mathrm{d}x \\
=& \frac{1}{16}\sum_{\nu,\mu} h_\nu g_\mu \left[a^{(4)}_{2i+\nu-\mu} + 4\mu a^{(3)}_{2i+\nu-\mu} + 6\mu^2 a^{(2)}_{2i+\nu-\mu} + 4\mu^3 a^{(1)}_{2i+\nu-\mu} + \mu^4\delta_{2i+\nu-\mu,0}\right],
\end{aligned} \tag{A.48}$$

$$\begin{aligned}
e_i^{(4)} =& \int \psi(x-i)x^4\psi(x)\,\mathrm{d}x \\
=& \frac{1}{16}\sum_{\nu,\mu} g_\nu g_\mu \left[a^{(4)}_{2i+\nu-\mu} + 4\mu a^{(3)}_{2i+\nu-\mu} + 6\mu^2 a^{(2)}_{2i+\nu-\mu} + 4\mu^3 a^{(1)}_{2i+\nu-\mu} + \mu^4\delta_{2i+\nu-\mu,0}\right].
\end{aligned} \tag{A.49}$$

A.2.3 The Filters for the general case

The calculation of the filters for the general case – i.e. variable grid spacing and location – is not as simple as for the derivatives, since the operator is this time not translational invariant.

A.2.3.1 General case – the linear operator

For the evaluation of the linear operator in the most general case one has to calculate

$$a^{(1)}_{i,j;h} = \int \phi\left(\frac{x}{h} - i\right)(x - x_0)\phi\left(\frac{x}{h} - j\right)\,\mathrm{d}x. \tag{A.50}$$

Again using the variable substitution $x' = \frac{x}{h} - j$ this leads to

$$\begin{aligned} a^{(1)}_{i,j;h} &= h\int \phi(x' + j - i)(h(x' + j) - x_0)\phi(x')\,\mathrm{d}x' \\ &= h\Bigg[h\int \phi(x' + j - i)x'\phi(x')\,\mathrm{d}x' \\ &\quad + hj\int \phi(x' + j - i)\phi(x')\,\mathrm{d}x' \\ &\quad - x_0\int \phi(x' + j - i)\phi(x')\,\mathrm{d}x'\Bigg] \\ &= h\left[ha^{(1)}_{i-j} + hj\delta_{i-j,0} - x_0\delta_{i-j,0}\right] \\ &= h\left[ha^{(1)}_{i-j} + (hj - x_0)\delta_{i-j,0}\right], \end{aligned} \tag{A.51}$$

where the results of Sec. A.2.1.1 and the orthonormality relations were used. As for the case of the derivatives, this filter again only depends on the difference $i - j$.
The other filters can be determined analogously and are given by

$$b^{(2)}_{i,j;h} = h^2 b^{(1)}_{i-j}, \tag{A.52}$$

$$c^{(2)}_{i,j;h} = h^2 c^{(1)}_{i-j}, \tag{A.53}$$

$$e^{(1)}_{i,j;h} = h\left[he^{(1)}_{i-j} + (hj - x_0)\delta_{i-j,0}\right]. \tag{A.54}$$

A.2.3.2 General case – the quadratic operator

The most general case for the quadratic operator reads

$$a^{(2)}_{i,j;h} = \int \phi\left(\frac{x}{h} - i\right)(x - x_0)^2\phi\left(\frac{x}{h} - j\right)\,\mathrm{d}x, \tag{A.55}$$

which becomes with the variable substitution $x' = \frac{x}{h} - j$

$$a^{(2)}_{i,j;h} = h \int \phi(x' + j - i)(h(x' + j) - x_0)^2 \phi(x')\, \mathrm{d}x'. \tag{A.56}$$

Expanding the term $(h(x' + j) - x_0)^2$ gives

$$\begin{aligned}(h(x' + j) - x_0)^2 &= h^2(x' + j)^2 - 2hx_0(x' + j) + x_0^2 \\ &= h^2x'^2 + 2h^2x'j + h^2j^2 - 2hx_0x' - 2hx_0j + x_0^2.\end{aligned} \tag{A.57}$$

Inserting this expansion in (A.56) yields a bunch of terms of the form

$$\int \phi(x' + j - i)x'^l \phi(x')\, \mathrm{d}x', \tag{A.58}$$

where l ranges from 0 to 2. Using the results of Sec. A.2.1 and the orthonormality of the scaling functions one therefore gets the final result

$$\begin{aligned}a^{(2)}_{i,j;h} =& h\left[h^2a^{(2)}_{i-j} + 2h^2ja^{(1)}_{i-j} + h^2j^2\delta_{i-j,0} - 2hx_0a^{(1)}_{i-j} - 2hx_0j\delta_{i-j,0} + x_0^2\delta_{i-j,0}\right] \\ =& h\left[h^2a^{(2)}_{i-j} + 2h(hj - x_0)a^{(1)}_{i-j} + (hj - x_0)^2\delta_{i-j,0}\right],\end{aligned} \tag{A.59}$$

which again only depends on the difference $i - j$.
Analogously one gets for the other cases

$$b^{(2)}_{i,j;h} = h\left[h^2b^{(2)}_{i-j} + 2h(hj - x_0)b^{(1)}_{i-j}\right], \tag{A.60}$$

$$c^{(2)}_{i,j;h} = h\left[h^2c^{(2)}_{i-j} + 2h(hj - x_0)c^{(1)}_{i-j}\right], \tag{A.61}$$

$$e^{(2)}_{i,j;h} = h\left[h^2e^{(2)}_{i-j} + 2h(hj - x_0)e^{(1)}_{i-j} + (hj - x_0)^2\delta_{i-j,0}\right]. \tag{A.62}$$

A.2.3.3 General case – the cubic operator

The most general case for the cubic operator reads

$$a^{(3)}_{i,j;h} = \int \phi\left(\frac{x}{h} - i\right)(x - x_0)^3 \phi\left(\frac{x}{h} - j\right)\, \mathrm{d}x, \tag{A.63}$$

which becomes, using again the variable substitution $x' = \frac{x}{h} - j$,

$$a^{(3)}_{i,j;h} = h \int \phi(x' + j - i)(h(x' + j) - x_0)^3 \phi(x')\, \mathrm{d}x'. \tag{A.64}$$

Expanding the term $(h(x' + j) - x_0)^3$ gives

$$\begin{aligned}(h(x' + j) - x_0)^3 &= h^3(x' + j)^3 - 3h^2x_0(x' + j)^2 + 3hx_0^2(x' + j) - x_0^3 \\ &= h^3x'^3 + 3h^3x'^2j + 3h^3x'j^2 + h^3j^3 \\ &\quad - 3h^2x_0x'^2 - 6h^2x_0x'j - 3h^2x_0j^2 + 3hx_0^2x' + 3hx_0^2j - x_0^3.\end{aligned} \tag{A.65}$$

Inserting this expansion in (A.64) yields again a bunch of terms of the form

$$\int \phi(x'+j-i)x'^l\phi(x')\,\mathrm{d}x', \tag{A.66}$$

where l ranges this time from 0 to 3. With the results of Sec. A.2.1 and the orthonormality of the scaling functions this gives

$$\begin{aligned} a^{(3)}_{i,j;h} &= h\left[h^3a^{(3)}_{i-j} + 3h^3ja^{(2)}_{i-j} + 3h^3j^2a^{(1)}_{i-j} + h^3j^3\delta_{i-j,0}\right.\\ &\quad \left. -3h^2x_0a^{(2)}_{i-j} - 6h^2x_0ja^{(1)}_{i-j} - 3h^2x_0j^2\delta_{i-j,0} + 3hx_0^2a^{(1)}_{i-j} + 3hx_0^2j\delta_{i-j,0} - x_0^3\delta_{i-j,0}\right]\\ &= h\left[h^3a^{(3)}_{i-j} + 3h^2(hj-x_0)a^{(2)}_{i-j} + 3h(hj-x_0)^2a^{(1)}_{i-j} + (hj-x_0)^3\delta_{i-j,0}\right], \end{aligned} \tag{A.67}$$

which again only depends on the difference $i-j$.
For the other cases one gets in the same way

$$\begin{aligned} b^{(3)}_{i,j;h} &= h\left[h^3b^{(3)}_{i-j} + 3h^2(hj-x_0)b^{(2)}_{i-j} + 3h(hj-x_0)^2b^{(1)}_{i-j}\right],\\ c^{(3)}_{i,j;h} &= h\left[h^3c^{(3)}_{i-j} + 3h^2(hj-x_0)c^{(2)}_{i-j} + 3h(hj-x_0)^2c^{(1)}_{i-j}\right],\\ e^{(3)}_{i,j;h} &= h\left[h^3e^{(3)}_{i-j} + 3h^2(hj-x_0)e^{(2)}_{i-j} + 3h(hj-x_0)^2e^{(1)}_{i-j} + (hj-x_0)^3\delta_{i-j,0}\right]. \end{aligned} \tag{A.68}$$

A.2.3.4 General case – the quartic operator

For the evaluation of the quartic operator in the most general case one has to calculate

$$a^{(4)}_{i,j;h} = \int \phi\left(\frac{x}{h}-i\right)(x-x_0)^4\phi\left(\frac{x}{h}-j\right)\,\mathrm{d}x. \tag{A.69}$$

Again using the variable substitution $x' = \frac{x}{h} - j$ this leads to

$$a^{(4)}_{i,j;h} = h\int \phi(x'+j-i)(h(x'+j)-x_0)^4\phi(x')\,\mathrm{d}x'. \tag{A.70}$$

Expanding the term $(h(x'+j)-x_0)^4$ gives

$$\begin{aligned} (h(x'+j)-x_0)^4 &= h^4(x'+j)^4 - 4h^3x_0(x'+j)^3 + 6h^2x_0^2(x'+j)^2 - 4hx_0^3(x'+j) + x_0^4\\ &= h^4x'^4 + 4h^4x'^3j + 6h^4x'^2j^2 + 4h^4x'j^3 + h^4j^4\\ &\quad - 4h^3x_0x'^3 - 12h^3x_0x'^2j - 12h^3x_0x'j^2 - 4h^3x_0j^3\\ &\quad + 6h^2x_0^2x'^2 + 12h^2x_0^2x'j + 6h^2x_0^2j^2 - 4hx_0^3x' - 4hx_0^3j + x_0^4. \end{aligned} \tag{A.71}$$

Inserting this expansion into (A.70) yields a bunch of terms of the form

$$\int \phi(x'+j-i)x'^l\phi(x')\,\mathrm{d}x', \tag{A.72}$$

where l ranges now from 0 to 4. Again using the results of Sec. A.2.1 and the orthonormality of the scaling functions yields the final result

$$\begin{aligned}
a^{(4)}_{i,j;h} &= h\Big[h^4 a^{(4)}_{i-j} + 4h^4 j a^{(3)}_{i-j} + 6h^4 j^2 a^{(2)}_{i-j} + 4h^4 j^3 a^{(1)}_{i-j} + h^4 j^4 \delta_{i-j,0} \\
&\quad -4h^3 x_0 a^{(3)}_{i-j} - 12h^3 x_0 j a^{(2)}_{i-j} - 12h^3 x_0 j^2 a^{(1)}_{i-j} - 4h^3 x_0 j^3 \delta_{i-j,0} \\
&\quad +6h^2 x_0^2 a^{(2)}_{i-j} + 12h^2 x_0^2 j a^{(1)}_{i-j} + 6h^2 x_0^2 j^2 \delta_{i-j,0} - 4h x_0^3 a^{(1)}_{i-j} - 4h x_0^3 j \delta_{i-j,0} + x_0^4 \delta_{i-j,0}\Big] \\
&= h\Big[h^4 a^{(4)}_{i-j} + 4h^3 (hj - x_0) a^{(3)}_{i-j} + 6h^2 (hj - x_0)^2 a^{(2)}_{i-j} \\
&\quad +4h(hj - x_0)^3 a^{(1)}_{i-j} + (hj - x_0)^4 \delta_{i-j,0}\Big],
\end{aligned} \tag{A.73}$$

which again only depends on the difference $i - j$.
The other filters can be determined along the same lines, leading to

$$\begin{aligned}
b^{(4)}_{i,j;h} &= \int \psi\left(\frac{x}{h} - i\right)(x - x_0)^4 \phi\left(\frac{x}{h} - j\right) \mathrm{d}x \\
&= h\left[h^4 b^{(4)}_{i-j} + 4h^3 (hj - x_0) b^{(3)}_{i-j} + 6h^2 (hj - x_0)^2 b^{(2)}_{i-j} + 4h(hj - x_0)^3 b^{(1)}_{i-j}\right], \qquad \text{(A.74)} \\
c^{(4)}_{i,j;h} &= \int \phi\left(\frac{x}{h} - i\right)(x - x_0)^4 \psi\left(\frac{x}{h} - j\right) \mathrm{d}x \\
&= h\left[h^4 c^{(4)}_{i-j} + 4h^3 (hj - x_0) c^{(3)}_{i-j} + 6h^2 (hj - x_0)^2 c^{(2)}_{i-j} + 4h(hj - x_0)^3 c^{(1)}_{i-j}\right], \qquad \text{(A.75)} \\
e^{(4)}_{i,j;h} &= \int \psi\left(\frac{x}{h} - i\right)(x - x_0)^4 \psi\left(\frac{x}{h} - j\right) \mathrm{d}x \\
&= h\Big[h^4 e^{(4)}_{i-j} + 4h^3 (hj - x_0) e^{(3)}_{i-j} + 6h^2 (hj - x_0)^2 e^{(2)}_{i-j} \\
&\quad +4h(hj - x_0)^3 e^{(1)}_{i-j} + (hj - x_0)^4 \delta_{i-j,0}\Big]. \qquad \text{(A.76)}
\end{aligned}$$

The filters again fulfill the same symmetry relations as in Eq. (A.13).

APPENDIX B

Applying operators to quantities expanded in a wavelet basis

In appendix A it has been demonstrated how to calculate the filters – i.e. the matrix elements among scaling functions and wavelets – for various operators.

This section will now show how these results can be used to apply the same operators to a quantity which is expanded in a wavelet basis. In three dimensions, such an expansion is given by

$$\begin{aligned}|\Psi\rangle = &\sum_{j_1,j_2,j_3} sss_{j_1,j_2,j_3} |\phi_{j_1}\phi_{j_2}\phi_{j_3}\rangle + \sum_{j_1,j_2,j_3} dss_{j_1,j_2,j_3} |\psi_{j_1}\phi_{j_2}\phi_{j_3}\rangle \\ &+ \sum_{j_1,j_2,j_3} sds_{j_1,j_2,j_3} |\phi_{j_1}\psi_{j_2}\phi_{j_3}\rangle + \sum_{j_1,j_2,j_3} dds_{j_1,j_2,j_3} |\psi_{j_1}\psi_{j_2}\phi_{j_3}\rangle \\ &+ \sum_{j_1,j_2,j_3} ssd_{j_1,j_2,j_3} |\phi_{j_1}\phi_{j_2}\psi_{j_3}\rangle + \sum_{j_1,j_2,j_3} dsd_{j_1,j_2,j_3} |\psi_{j_1}\phi_{j_2}\psi_{j_3}\rangle \\ &+ \sum_{j_1,j_2,j_3} sdd_{j_1,j_2,j_3} |\phi_{j_1}\psi_{j_2}\psi_{j_3}\rangle + \sum_{j_1,j_2,j_3} ddd_{j_1,j_2,j_3} |\psi_{j_1}\psi_{j_2}\psi_{j_3}\rangle ,\end{aligned} \tag{B.1}$$

where $|\phi_{j_1}\phi_{j_2}\phi_{j_3}\rangle$ etc. is a shorthand notation for $|\phi(x-j_1)\phi(y-j_2)\phi(z-j_3)\rangle$ etc.
After the application of the operator $\hat{O}$ the quantity Ψ is still represented in the same

basis, just this time with modified coefficients:

$$\begin{aligned}\hat{\mathcal{O}}\,|\Psi\rangle = &\sum_{j_1,j_2,j_3} sss'_{j_1,j_2,j_3}\,|\phi_{j_1}\phi_{j_2}\phi_{j_3}\rangle + \sum_{j_1,j_2,j_3} dss'_{j_1,j_2,j_3}\,|\psi_{j_1}\phi_{j_2}\phi_{j_3}\rangle \\ &+ \sum_{j_1,j_2,j_3} sds'_{j_1,j_2,j_3}\,|\phi_{j_1}\psi_{j_2}\phi_{j_3}\rangle + \sum_{j_1,j_2,j_3} dds'_{j_1,j_2,j_3}\,|\psi_{j_1}\psi_{j_2}\phi_{j_3}\rangle \\ &+ \sum_{j_1,j_2,j_3} ssd'_{j_1,j_2,j_3}\,|\phi_{j_1}\phi_{j_2}\psi_{j_3}\rangle + \sum_{j_1,j_2,j_3} dsd'_{j_1,j_2,j_3}\,|\psi_{j_1}\phi_{j_2}\psi_{j_3}\rangle \\ &+ \sum_{j_1,j_2,j_3} sdd'_{j_1,j_2,j_3}\,|\phi_{j_1}\psi_{j_2}\psi_{j_3}\rangle + \sum_{j_1,j_2,j_3} ddd'_{j_1,j_2,j_3}\,|\psi_{j_1}\psi_{j_2}\psi_{j_3}\rangle\,.\end{aligned} \tag{B.2}$$

From the orthogonality relations of the scaling functions and wavelets it thus follows that the expansion coefficients after the application of the operator are given by

$$\begin{aligned} sss'_{i_1,i_2,i_3} &= \langle\phi_{i_1}\phi_{i_2}\phi_{i_3}|\hat{\mathcal{O}}|\Psi\rangle\,, & dss'_{i_1,i_2,i_3} &= \langle\psi_{i_1}\phi_{i_2}\phi_{i_3}|\hat{\mathcal{O}}|\Psi\rangle\,, \\ sds'_{i_1,i_2,i_3} &= \langle\phi_{i_1}\psi_{i_2}\phi_{i_3}|\hat{\mathcal{O}}|\Psi\rangle\,, & dds'_{i_1,i_2,i_3} &= \langle\psi_{i_1}\psi_{i_2}\phi_{i_3}|\hat{\mathcal{O}}|\Psi\rangle\,, \\ ssd'_{i_1,i_2,i_3} &= \langle\phi_{i_1}\phi_{i_2}\psi_{i_3}|\hat{\mathcal{O}}|\Psi\rangle\,, & dsd'_{i_1,i_2,i_3} &= \langle\psi_{i_1}\phi_{i_2}\psi_{i_3}|\hat{\mathcal{O}}|\Psi\rangle\,, \\ sdd'_{i_1,i_2,i_3} &= \langle\phi_{i_1}\psi_{i_2}\psi_{i_3}|\hat{\mathcal{O}}|\Psi\rangle\,, & ddd'_{i_1,i_2,i_3} &= \langle\psi_{i_1}\psi_{i_2}\psi_{i_3}|\hat{\mathcal{O}}|\Psi\rangle\,. \end{aligned} \tag{B.3}$$

B.1 Derivative operators

Applying the derivative operator of any order l in x direction, i.e. $\frac{\partial^l}{\partial x^l}$, has no effect on the other dimensions. Thus it follows from the orthonormality relations $\langle\phi_{i_2}|\phi_{j_2}\rangle = \delta_{i_2 j_2}$ and $\langle\phi_{i_2}|\psi_{j_2}\rangle = 0$ (and the analogs for i_3 and j_3) and the filters calculated in appendix A (i.e. $a_{i_1-j_1} = \langle\phi_{i_1}|\frac{\partial^l}{\partial x^l}|\phi_{j_1}\rangle$ and $c_{i_1-j_1} = \langle\phi_{i_1}|\frac{\partial}{\partial x}|\psi_{j_1}\rangle$) that

$$\begin{aligned} sss'_{i_1,i_2,i_3} &= \langle\phi_{i_1}\phi_{i_2}\phi_{i_3}|\frac{\partial^l}{\partial x^l}|\Psi\rangle \\ &= \sum_{j_1,j_2,j_3} a_{i_1-j_1} sss_{j_1,j_2,j_3}\delta_{j_2 i_2}\delta_{j_3 i_3} + \sum_{j_1,j_2,j_3} c_{i_1-j_1} dss_{j_1,j_2,j_3}\delta_{j_2 i_2}\delta_{j_3 i_3} \\ &= \sum_{j_1} a_{i_1-j_1} sss_{j_1,i_2,i_3} + \sum_{j_1} c_{i_1-j_1} dss_{j_1,i_2,i_3} \\ &= \sum_{j_1} a_{j_1-i_1} sss_{j_1,i_2,i_3} + \sum_{j_1} b_{j_1-i_1} dss_{j_1,i_2,i_3}. \end{aligned} \tag{B.4}$$

In the last step the symmetry relations of Eq. (A.13) were used. Exactly the same considerations apply of course as well for the other coefficients. Consequently the

entire list for the operator $\frac{\partial^l}{\partial x^l}$ is

$$sss'_{i_1,i_2,i_3} = \langle \phi_{i_1}\phi_{i_2}\phi_{i_3}|\frac{\partial^l}{\partial x^l}|\Psi\rangle = \sum_{j_1} a_{j_1-i_1} sss_{j_1,i_2,i_3} + \sum_{j_1} b_{j_1-i_1} dss_{j_1,i_2,i_3}, \tag{B.5}$$

$$dss'_{i_1,i_2,i_3} = \langle \psi_{i_1}\phi_{i_2}\phi_{i_3}|\frac{\partial^l}{\partial x^l}|\Psi\rangle = \sum_{j_1} e_{j_1-i_1} dss_{j_1,i_2,i_3} + \sum_{j_1} c_{j_1-i_1} sss_{j_1,i_2,i_3}, \tag{B.6}$$

$$sds'_{i_1,i_2,i_3} = \langle \phi_{i_1}\psi_{i_2}\phi_{i_3}|\frac{\partial^l}{\partial x^l}|\Psi\rangle = \sum_{j_1} a_{j_1-i_1} sds_{j_1,i_2,i_3} + \sum_{j_1} b_{j_1-i_1} dds_{j_1,i_2,i_3}, \tag{B.7}$$

$$dds'_{i_1,i_2,i_3} = \langle \psi_{i_1}\psi_{i_2}\phi_{i_3}|\frac{\partial^l}{\partial x^l}|\Psi\rangle = \sum_{j_1} e_{j_1-i_1} dds_{j_1,i_2,i_3} + \sum_{j_1} c_{j_1-i_1} sds_{j_1,i_2,i_3}, \tag{B.8}$$

$$ssd'_{i_1,i_2,i_3} = \langle \phi_{i_1}\phi_{i_2}\psi_{i_3}|\frac{\partial^l}{\partial x^l}|\Psi\rangle = \sum_{j_1} a_{j_1-i_1} ssd_{j_1,i_2,i_3} + \sum_{j_1} b_{j_1-i_1} dsd_{j_1,i_2,i_3}, \tag{B.9}$$

$$dsd'_{i_1,i_2,i_3} = \langle \psi_{i_1}\phi_{i_2}\psi_{i_3}|\frac{\partial^l}{\partial x^l}|\Psi\rangle = \sum_{j_1} e_{j_1-i_1} dsd_{j_1,i_2,i_3} + \sum_{j_1} c_{j_1-i_1} ssd_{j_1,i_2,i_3}, \tag{B.10}$$

$$sdd'_{i_1,i_2,i_3} = \langle \phi_{i_1}\psi_{i_2}\psi_{i_3}|\frac{\partial^l}{\partial x^l}|\Psi\rangle = \sum_{j_1} a_{j_1-i_1} sdd_{j_1,i_2,i_3} + \sum_{j_1} b_{j_1-i_1} ddd_{j_1,i_2,i_3}, \tag{B.11}$$

$$ddd'_{i_1,i_2,i_3} = \langle \psi_{i_1}\psi_{i_2}\psi_{i_3}|\frac{\partial^l}{\partial x^l}|\Psi\rangle = \sum_{j_1} e_{j_1-i_1} ddd_{j_1,i_2,i_3} + \sum_{j_1} c_{j_1-i_1} sdd_{j_1,i_2,i_3}. \tag{B.12}$$

The coefficients for the derivative along the y dimension can be calculated along the same lines:

$$sss'_{i_1,i_2,i_3} = \langle \phi_{i_1}\phi_{i_2}\phi_{i_3}|\frac{\partial^l}{\partial y^l}|\Psi\rangle = \sum_{j_2} a_{j_2-i_2} sss_{i_1,j_2,i_3} + \sum_{j_2} b_{j_2-i_2} sds_{i_1,j_2,i_3}, \tag{B.13}$$

$$dss'_{i_1,i_2,i_3} = \langle \psi_{i_1}\phi_{i_2}\phi_{i_3}|\frac{\partial^l}{\partial y^l}|\Psi\rangle = \sum_{j_2} a_{j_2-i_2} dss_{i_1,j_2,i_3} + \sum_{j_2} b_{j_2-i_2} dds_{i_1,j_2,i_3}, \tag{B.14}$$

$$sds'_{i_1,i_2,i_3} = \langle \phi_{i_1}\psi_{i_2}\phi_{i_3}|\frac{\partial^l}{\partial y^l}|\Psi\rangle = \sum_{j_2} e_{j_2-i_2} sds_{i_1,j_2,i_3} + \sum_{j_2} c_{j_2-i_2} sss_{i_1,j_2,i_3}, \tag{B.15}$$

$$dds'_{i_1,i_2,i_3} = \langle \psi_{i_1}\psi_{i_2}\phi_{i_3}|\frac{\partial^l}{\partial y^l}|\Psi\rangle = \sum_{j_2} e_{j_2-i_2} dds_{i_1,j_2,i_3} + \sum_{j_2} c_{j_2-i_2} dss_{i_1,j_2,i_3}, \tag{B.16}$$

$$ssd'_{i_1,i_2,i_3} = \langle \phi_{i_1}\phi_{i_2}\psi_{i_3}|\frac{\partial^l}{\partial y^l}|\Psi\rangle = \sum_{j_2} a_{j_2-i_2} ssd_{i_1,j_2,i_3} + \sum_{j_2} b_{j_2-i_2} sdd_{i_1,j_2,i_3}, \tag{B.17}$$

$$dsd'_{i_1,i_2,i_3} = \langle \psi_{i_1}\phi_{i_2}\psi_{i_3}|\frac{\partial^l}{\partial y^l}|\Psi\rangle = \sum_{j_2} a_{j_2-i_2} dsd_{i_1,j_2,i_3} + \sum_{j_2} b_{j_2-i_2} ddd_{i_1,j_2,i_3}, \tag{B.18}$$

$$sdd'_{i_1,i_2,i_3} = \langle \phi_{i_1}\psi_{i_2}\psi_{i_3}|\frac{\partial^l}{\partial y^l}|\Psi\rangle = \sum_{j_2} e_{j_2-i_2} sdd_{i_1,j_2,i_3} + \sum_{j_2} c_{j_2-i_2} ssd_{i_1,j_2,i_3}, \tag{B.19}$$

$$ddd'_{i_1,i_2,i_3} = \langle \psi_{i_1}\psi_{i_2}\psi_{i_3}|\frac{\partial^l}{\partial y^l}|\Psi\rangle = \sum_{j_2} e_{j_2-i_2} ddd_{i_1,j_2,i_3} + \sum_{j_2} c_{j_2-i_2} dsd_{i_1,j_2,i_3}. \tag{B.20}$$

Finally the coefficients for derivatives along the z direction are given by

$$sss'_{i_1,i_2,i_3} = \langle \phi_{i_1}\phi_{i_2}\phi_{i_3}|\frac{\partial^l}{\partial z^l}|\Psi\rangle = \sum_{j_3} a_{j_3-i_3} sss_{i_1,i_2,j_3} + \sum_{j_3} b_{j_3-i_3} ssd_{i_1,i_2,j_3}, \tag{B.21}$$

$$dss'_{i_1,i_2,i_3} = \langle \psi_{i_1}\phi_{i_2}\phi_{i_3}|\frac{\partial^l}{\partial z^l}|\Psi\rangle = \sum_{j_3} a_{j_3-i_3} dss_{i_1,i_2,j_3} + \sum_{j_3} b_{j_3-i_3} dsd_{i_1,i_2,j_3}, \tag{B.22}$$

$$sds'_{i_1,i_2,i_3} = \langle \phi_{i_1}\psi_{i_2}\phi_{i_3}|\frac{\partial^l}{\partial z^l}|\Psi\rangle = \sum_{j_3} a_{j_3-i_3} sds_{i_1,i_2,j_3} + \sum_{j_3} b_{j_3-i_3} sdd_{i_1,i_2,j_3}, \tag{B.23}$$

$$dds'_{i_1,i_2,i_3} = \langle \psi_{i_1}\psi_{i_2}\phi_{i_3}|\frac{\partial^l}{\partial z^l}|\Psi\rangle = \sum_{j_3} a_{j_3-i_3} dds_{i_1,i_2,j_3} + \sum_{j_3} b_{j_3-i_3} ddd_{i_1,i_2,j_3}, \tag{B.24}$$

$$ssd'_{i_1,i_2,i_3} = \langle \phi_{i_1}\phi_{i_2}\psi_{i_3}|\frac{\partial^l}{\partial z^l}|\Psi\rangle = \sum_{j_3} e_{j_3-i_3} ssd_{i_1,i_2,j_3} + \sum_{j_3} c_{j_3-i_3} sss_{i_1,i_2,j_3}, \tag{B.25}$$

$$dsd'_{i_1,i_2,i_3} = \langle \psi_{i_1}\phi_{i_2}\psi_{i_3}|\frac{\partial^l}{\partial z^l}|\Psi\rangle = \sum_{j_3} e_{j_3-i_3} dsd_{i_1,i_2,j_3} + \sum_{j_3} c_{j_3-i_3} dss_{i_1,i_2,j_3}, \tag{B.26}$$

$$sdd'_{i_1,i_2,i_3} = \langle \phi_{i_1}\psi_{i_2}\psi_{i_3}|\frac{\partial^l}{\partial z^l}|\Psi\rangle = \sum_{j_3} e_{j_3-i_3} sdd_{i_1,i_2,j_3} + \sum_{j_3} c_{j_3-i_3} sds_{i_1,i_2,j_3}, \tag{B.27}$$

$$ddd'_{i_1,i_2,i_3} = \langle \psi_{i_1}\psi_{i_2}\psi_{i_3}|\frac{\partial^l}{\partial z^l}|\Psi\rangle = \sum_{j_3} e_{j_3-i_3} ddd_{i_1,i_2,j_3} + \sum_{j_3} c_{j_3-i_3} dds_{i_1,i_2,j_3}. \tag{B.28}$$

B.2 Position operators

Applying just a position operator along one dimension, for instance $x^4|\Psi\rangle$, is completely analogous to the case of the derivative, just with the difference that the filters a, b, c and e are different. Even the application of operators which are not centered around the origin, i.e. $(x-x_0)^4|\Psi\rangle$, can be done along the same lines as long as the general filters of Sec. A.2.3 are used.

However the situation is a bit more complicated for mixed expression, for instance

$x^2y^2\,|\Psi\rangle$. Such terms arise if the confining potential which is used (cf. Sec. 5.1.1) is expanded:

$$\begin{aligned}(\mathbf{r}-\mathbf{r}_0)^4 &= \Big((x-x_0)^2+(y-y_0)^2+(z-z_0)^2\Big)^2 \\ &= (x-x_0)^4+(y-y_0)^4+(z-z_0)^4 \\ &\quad +2(x-x_0)^2(y-y_0)^2+2(x-x_0)^2(z-z_0)^2+2(y-y_0)^2(z-z_0)^2.\end{aligned} \tag{B.29}$$

Again the dependency on the origin is already contained in the general filters and it is therefore sufficient to consider the cases of x^2y^2, x^2z^2 and y^2z^2.

Defining, as usual, the filter elements $a_{i_1-j_1}=\langle\phi_{i_1}|x^2|\phi_{j_1}\rangle$ and $c_{i_1-j_1}=\langle\phi_{i_1}|x^2|\psi_{j_1}\rangle$ and using the orthonormality relation $\langle\phi_{i_3}|\psi_{j_3}\rangle=0$ one can first evaluate the operator x^2:

$$\begin{aligned}sss'_{i_1,i_2,i_3} &= \langle\phi_{i_1}\phi_{i_2}\phi_{i_3}|x^2y^2|\Psi\rangle \\ &= \sum_{j_1,j_2,j_3} a_{i_1-j_1}sss_{j_1,j_2,j_3}\,\langle\phi_{i_2}\phi_{i_3}|y^2|\phi_{j_2}\phi_{j_3}\rangle + \sum_{j_1,j_2,j_3} c_{i_1-j_1}dss_{j_1,j_2,j_3}\,\langle\phi_{i_2}\phi_{i_3}|y^2|\phi_{j_2}\phi_{j_3}\rangle \\ &\quad + \sum_{j_1,j_2,j_3} a_{i_1-j_1}sds_{j_1,j_2,j_3}\,\langle\phi_{i_2}\phi_{i_3}|y^2|\psi_{j_2}\phi_{j_3}\rangle + \sum_{j_1,j_2,j_3} c_{i_1-j_1}dds_{j_1,j_2,j_3}\,\langle\phi_{i_2}\phi_{i_3}|y^2|\psi_{j_2}\phi_{j_3}\rangle\,.\end{aligned} \tag{B.30}$$

The analogous procedure can now be applied for the y^2 operator. Using the orthonormality relation $\langle\phi_{i_3}|\phi_{j_3}\rangle=\delta_{i_3j_3}$ this yields

$$\begin{aligned}sss'_{i_1,i_2,i_3} &= \sum_{j_1,j_2,j_3} a_{i_1-j_1}sss_{j_1,j_2,j_3}a_{i_2-j_2}\delta_{i_3j_3} + \sum_{j_1,j_2,j_3} c_{i_1-j_1}dss_{j_1,j_2,j_3}a_{j_2-i_2}\delta_{i_3j_3} \\ &\quad + \sum_{j_1,j_2,j_3} a_{i_1-j_1}sds_{j_1,j_2,j_3}c_{i_2-j_2}\delta_{i_3j_3} + \sum_{j_1,j_2,j_3} c_{i_1-j_1}dds_{j_1,j_2,j_3}c_{j_2-i_2}\delta_{i_3j_3} \\ &= \sum_{j_1,j_2} a_{i_1-j_1}a_{i_2-j_2}sss_{j_1,j_2,i_3} + \sum_{j_1,j_2} c_{i_1-j_1}a_{i_2-j_2}dss_{j_1,j_2,i_3} \\ &\quad + \sum_{j_1,j_2} a_{i_1-j_1}c_{i_2-j_2}sds_{j_1,j_2,i_3} + \sum_{j_1,j_2} c_{i_1-j_1}c_{i_2-j_2}dds_{j_1,j_2,i_3},\end{aligned} \tag{B.31}$$

or, again using the symmetry relations (A.13),

$$\begin{aligned}sss'_{i_1,i_2,i_3} &= \sum_{j_1,j_2} a_{j_1-i_1}a_{j_2-i_2}sss_{j_1,j_2,i_3} + \sum_{j_1,j_2} b_{j_1-i_1}a_{j_2-i_2}dss_{j_1,j_2,i_3} \\ &\quad + \sum_{j_1,j_2} a_{j_1-i_1}b_{j_2-i_2}sds_{j_1,j_2,i_3} + \sum_{j_1,j_2} b_{j_1-i_1}b_{j_2-i_2}dds_{j_1,j_2,i_3}.\end{aligned} \tag{B.32}$$

In order to avoid these cumbersome two-dimensional convolutions, a set of auxiliary coefficients can be constructed as follows:

$$\begin{aligned}\sigma\sigma\sigma^{i_1;a}_{i_1,j_2,i_3} &= \sum_{j_1} a_{j_1-i_1}sss_{j_1,j_2,i_3}, & \delta\sigma\sigma^{i_1;b}_{i_1,j_2,i_3} &= \sum_{j_1} b_{j_1-i_1}dss_{j_1,j_2,i_3}, \\ \sigma\delta\sigma^{i_1;a}_{i_1,j_2,i_3} &= \sum_{j_1} a_{j_1-i_1}sds_{j_1,j_2,i_3}, & \delta\delta\sigma^{i_1;b}_{i_1,j_2,i_3} &= \sum_{j_1} b_{j_1-i_1}dds_{j_1,j_2,i_3}.\end{aligned} \tag{B.33}$$

Now the coefficients sss'_{i_1,i_2,i_3} can be calculated as one-dimensional convolutions using these auxiliary coefficients:

$$\begin{aligned} sss'_{i_1,i_2,i_3} = &\sum_{j_2} a_{j_2-i_2}\sigma\sigma\sigma^{i_1;a}_{i_1,j_2,i_3} + \sum_{j_2} a_{j_2-i_2}\delta\sigma\sigma^{i_1;b}_{j_1,j_2,i_3} \\ &+ \sum_{j_2} b_{j_2-i_2}\sigma\delta\sigma^{i_1;a}_{i_1,j_2,i_3} + \sum_{j_2} b_{j_2-i_2}\delta\delta\sigma^{i_1;b}_{j_1,j_2,i_3}. \end{aligned} \tag{B.34}$$

The procedure to calculate the other coefficients is completely analogous; thus the entire list is

$$\begin{aligned} sss'_{i_1,i_2,i_3} = \langle\phi_{i_1}\phi_{i_2}\phi_{i_3}|x^2y^2|\Psi\rangle = &\sum_{j_2} a_{j_2-i_2}\sigma\sigma\sigma^{i_1;a}_{i_1,j_2,i_3} + \sum_{j_2} a_{j_2-i_2}\delta\sigma\sigma^{i_1;b}_{i_1,j_2,i_3} \\ &+ \sum_{j_2} b_{j_2-i_2}\sigma\delta\sigma^{i_1;a}_{i_1,j_2,i_3} + \sum_{j_2} b_{j_2-i_2}\delta\delta\sigma^{i_1;b}_{i_1,j_2,i_3}, \end{aligned} \tag{B.35}$$

$$\begin{aligned} dss'_{i_1,i_2,i_3} = \langle\psi_{i_1}\phi_{i_2}\phi_{i_3}|x^2y^2|\Psi\rangle = &\sum_{j_2} a_{j_2-i_2}\sigma\sigma\sigma^{i_1;c}_{i_1,j_2,i_3} + \sum_{j_2} a_{j_2-i_2}\delta\sigma\sigma^{i_1;e}_{i_1,j_2,i_3} \\ &+ \sum_{j_2} b_{j_2-i_2}\sigma\delta\sigma^{i_1;c}_{i_1,j_2,i_3} + \sum_{j_2} b_{j_2-i_2}\delta\delta\sigma^{i_1;e}_{i_1,j_2,i_3}, \end{aligned} \tag{B.36}$$

$$\begin{aligned} sds'_{i_1,i_2,i_3} = \langle\phi_{i_1}\psi_{i_2}\phi_{i_3}|x^2y^2|\Psi\rangle = &\sum_{j_2} c_{j_2-i_2}\sigma\sigma\sigma^{i_1;a}_{i_1,j_2,i_3} + \sum_{j_2} c_{j_2-i_2}\delta\sigma\sigma^{i_1;b}_{i_1,j_2,i_3} \\ &+ \sum_{j_2} e_{j_2-i_2}\sigma\delta\sigma^{i_1;a}_{i_1,j_2,i_3} + \sum_{j_2} e_{j_2-i_2}\delta\delta\sigma^{i_1;b}_{i_1,j_2,i_3}, \end{aligned} \tag{B.37}$$

$$\begin{aligned} dds'_{i_1,i_2,i_3} = \langle\phi_{i_1}\psi_{i_2}\psi_{i_3}|x^2y^2|\Psi\rangle = &\sum_{j_2} c_{j_2-i_2}\sigma\sigma\sigma^{i_1;c}_{i_1,j_2,i_3} + \sum_{j_2} c_{j_2-i_2}\delta\sigma\sigma^{i_1;e}_{i_1,j_2,i_3} \\ &+ \sum_{j_2} e_{j_2-i_2}\sigma\delta\sigma^{i_1;c}_{i_1,j_2,i_3} + \sum_{j_2} e_{j_2-i_2}\delta\delta\sigma^{i_1;e}_{i_1,j_2,i_3}, \end{aligned} \tag{B.38}$$

$$\begin{aligned} ssd'_{i_1,i_2,i_3} = \langle\phi_{i_1}\phi_{i_2}\psi_{i_3}|x^2y^2|\Psi\rangle = &\sum_{j_2} a_{j_2-i_2}\sigma\sigma\delta^{i_1;a}_{i_1,j_2,i_3} + \sum_{j_2} a_{j_2-i_2}\delta\sigma\delta^{i_1;b}_{i_1,j_2,i_3} \\ &+ \sum_{j_2} b_{j_2-i_2}\sigma\delta\delta^{i_1;a}_{i_1,j_2,i_3} + \sum_{j_2} b_{j_2-i_2}\delta\delta\delta^{i_1;b}_{i_1,j_2,i_3}, \end{aligned} \tag{B.39}$$

$$\begin{aligned} dsd'_{i_1,i_2,i_3} = \langle\psi_{i_1}\phi_{i_2}\psi_{i_3}|x^2y^2|\Psi\rangle = &\sum_{j_2} a_{j_2-i_2}\sigma\sigma\delta^{i_1;c}_{i_1,j_2,i_3} + \sum_{j_2} a_{j_2-i_2}\delta\sigma\delta^{i_1;e}_{i_1,j_2,i_3} \\ &+ \sum_{j_2} b_{j_2-i_2}\sigma\delta\delta^{i_1;c}_{i_1,j_2,i_3} + \sum_{j_2} b_{j_2-i_2}\delta\delta\delta^{i_1;e}_{i_1,j_2,i_3}, \end{aligned} \tag{B.40}$$

$$\begin{aligned} sdd'_{i_1,i_2,i_3} = \langle\phi_{i_1}\psi_{i_2}\psi_{i_3}|x^2y^2|\Psi\rangle = &\sum_{j_2} c_{j_2-i_2}\sigma\sigma\delta^{i_1;a}_{i_1,j_2,i_3} + \sum_{j_2} c_{j_2-i_2}\delta\sigma\delta^{i_1;b}_{i_1,j_2,i_3} \\ &+ \sum_{j_2} e_{j_2-i_2}\sigma\delta\delta^{i_1;a}_{i_1,j_2,i_3} + \sum_{j_2} e_{j_2-i_2}\delta\delta\delta^{i_1;b}_{i_1,j_2,i_3}, \end{aligned} \tag{B.41}$$

$$ddd'_{i_1,i_2,i_3} = \langle \psi_{i_1}\psi_{i_2}\psi_{i_3}|x^2y^2|\Psi\rangle = \sum_{j_2} c_{j_2-i_2}\sigma\sigma\delta^{i_1;c}_{i_1,j_2,i_3} + \sum_{j_2} c_{j_2-i_2}\delta\sigma\delta^{i_1;e}_{i_1,j_2,i_3} + \sum_{j_2} e_{j_2-i_2}\sigma\delta\delta^{i_1;c}_{i_1,j_2,i_3} + \sum_{j_2} e_{j_2-i_2}\delta\delta\delta^{i_1;e}_{i_1,j_2,i_3}, \quad \text{(B.42)}$$

where the various auxiliary filters are obvious generalizations of (B.33).

The coefficients for the other two operators, that is x^2z^2 and y^2z^2, can be derived along the same lines. The result for x^2z^2 is

$$sss'_{i_1,i_2,i_3} = \langle \phi_{i_1}\phi_{i_2}\phi_{i_3}|x^2z^2|\Psi\rangle = \sum_{j_3} a_{j_3-i_3}\sigma\sigma\sigma^{i_1;a}_{i_1,i_2,j_3} + \sum_{j_3} a_{j_3-i_3}\delta\sigma\sigma^{i_1;b}_{i_1,i_2,j_3} + \sum_{j_3} b_{j_3-i_3}\sigma\sigma\delta^{i_1;a}_{i_1,i_2,j_3} + \sum_{j_3} b_{j_3-i_3}\delta\sigma\delta^{i_1;b}_{i_1,i_2,j_3}, \quad \text{(B.43)}$$

$$dss'_{i_1,i_2,i_3} = \langle \psi_{i_1}\phi_{i_2}\phi_{i_3}|x^2z^2|\Psi\rangle = \sum_{j_3} a_{j_3-i_3}\sigma\sigma\sigma^{i_1;c}_{i_1,i_2,j_3} + \sum_{j_3} a_{j_3-i_3}\delta\sigma\sigma^{i_1;e}_{i_1,i_2,j_3} + \sum_{j_3} b_{j_3-i_3}\sigma\sigma\delta^{i_1;c}_{i_1,i_2,j_3} + \sum_{j_3} b_{j_3-i_3}\delta\sigma\delta^{i_1;e}_{i_1,i_2,j_3}, \quad \text{(B.44)}$$

$$sds'_{i_1,i_2,i_3} = \langle \phi_{i_1}\psi_{i_2}\phi_{i_3}|x^2z^2|\Psi\rangle = \sum_{j_3} a_{j_3-i_3}\sigma\delta\sigma^{i_1;a}_{i_1,i_2,j_3} + \sum_{j_3} a_{j_3-i_3}\delta\delta\sigma^{i_1;b}_{i_1,i_2,j_3} + \sum_{j_3} b_{j_3-i_3}\sigma\delta\delta^{i_1;a}_{i_1,i_2,j_3} + \sum_{j_3} b_{j_3-i_3}\delta\delta\delta^{i_1;b}_{i_1,i_2,j_3}, \quad \text{(B.45)}$$

$$dds'_{i_1,i_2,i_3} = \langle \phi_{i_1}\psi_{i_2}\psi_{i_3}|x^2z^2|\Psi\rangle = \sum_{j_3} a_{j_3-i_3}\sigma\delta\sigma^{i_1;c}_{i_1,i_2,j_3} + \sum_{j_3} a_{j_3-i_3}\delta\delta\sigma^{i_1;e}_{i_1,i_2,j_3} + \sum_{j_3} b_{j_3-i_3}\sigma\delta\delta^{i_1;c}_{i_1,i_2,j_3} + \sum_{j_3} b_{j_3-i_3}\delta\delta\delta^{i_1;e}_{i_1,i_2,j_3}, \quad \text{(B.46)}$$

$$ssd'_{i_1,i_2,i_3} = \langle \phi_{i_1}\phi_{i_2}\psi_{i_3}|x^2z^2|\Psi\rangle = \sum_{j_3} c_{j_3-i_3}\sigma\sigma\sigma^{i_1;a}_{i_1,i_2,j_3} + \sum_{j_3} c_{j_3-i_3}\delta\sigma\sigma^{i_1;b}_{i_1,i_2,j_3} + \sum_{j_3} e_{j_3-i_3}\sigma\sigma\delta^{i_1;a}_{i_1,i_2,j_3} + \sum_{j_3} e_{j_3-i_3}\delta\sigma\delta^{i_1;b}_{i_1,i_2,j_3}, \quad \text{(B.47)}$$

$$dsd'_{i_1,i_2,i_3} = \langle \psi_{i_1}\phi_{i_2}\psi_{i_3}|x^2z^2|\Psi\rangle = \sum_{j_3} c_{j_3-i_3}\sigma\sigma\sigma^{i_1;c}_{i_1,i_2,j_3} + \sum_{j_3} c_{j_3-i_3}\delta\sigma\sigma^{i_1;e}_{i_1,i_2,j_3} + \sum_{j_3} e_{j_3-i_3}\sigma\sigma\delta^{i_1;c}_{i_1,i_2,j_3} + \sum_{j_3} e_{j_3-i_3}\delta\sigma\delta^{i_1;e}_{i_1,i_2,j_3}, \quad \text{(B.48)}$$

$$sdd'_{i_1,i_2,i_3} = \langle \phi_{i_1}\psi_{i_2}\psi_{i_3}|x^2z^2|\Psi\rangle = \sum_{j_3} c_{j_3-i_3}\sigma\delta\sigma^{i_1;a}_{i_1,i_2,j_3} + \sum_{j_3} c_{j_3-i_3}\delta\delta\sigma^{i_1;b}_{i_1,i_2,j_3} + \sum_{j_3} e_{j_3-i_3}\sigma\delta\delta^{i_1;a}_{i_1,i_2,j_3} + \sum_{j_3} e_{j_3-i_3}\delta\delta\delta^{i_1;b}_{i_1,i_2,j_3}, \quad \text{(B.49)}$$

$$ddd'_{i_1,i_2,i_3} = \langle \psi_{i_1}\psi_{i_2}\psi_{i_3}|x^2z^2|\Psi\rangle = \sum_{j_3} c_{j_3-i_3}\sigma\delta\sigma^{i_1;c}_{i_1,i_2,j_3} + \sum_{j_3} c_{j_3-i_3}\delta\delta\sigma^{i_1;e}_{i_1,i_2,j_3}$$

$$+\sum_{j_3} e_{j_3-i_3}\sigma\delta\delta^{i_1;c}_{i_1,i_2,j_3}+\sum_{j_3} e_{j_3-i_3}\delta\delta\delta^{i_1;e}_{i_1,i_2,j_3}, \tag{B.50}$$

and finally the one for y^2z^2

$$sss'_{i_1,i_2,i_3}=\langle\phi_{i_1}\phi_{i_2}\phi_{i_3}|y^2z^2|\Psi\rangle=\sum_{j_3} a_{j_3-i_3}\sigma\sigma\sigma^{i_2;a}_{i_1,i_2,j_3}+\sum_{j_3} a_{j_3-i_3}\sigma\delta\sigma^{i_2;b}_{i_1,i_2,j_3}$$
$$+\sum_{j_3} b_{j_3-i_3}\sigma\sigma\delta^{i_2;a}_{i_1,i_2,j_3}+\sum_{j_3} b_{j_3-i_3}\sigma\delta\delta^{i_2;b}_{i_1,i_2,j_3}, \tag{B.51}$$

$$dss'_{i_1,i_2,i_3}=\langle\psi_{i_1}\phi_{i_2}\phi_{i_3}|y^2z^2|\Psi\rangle=\sum_{j_3} a_{j_3-i_3}\delta\sigma\sigma^{i_2;a}_{i_1,i_2,j_3}+\sum_{j_3} a_{j_3-i_3}\delta\delta\sigma^{i_2;b}_{i_1,i_2,j_3}$$
$$+\sum_{j_3} b_{j_3-i_3}\delta\sigma\delta^{i_2;a}_{i_1,i_2,j_3}+\sum_{j_3} b_{j_3-i_3}\delta\delta\delta^{i_2;b}_{i_1,i_2,j_3}, \tag{B.52}$$

$$sds'_{i_1,i_2,i_3}=\langle\phi_{i_1}\psi_{i_2}\phi_{i_3}|y^2z^2|\Psi\rangle=\sum_{j_3} a_{j_3-i_3}\sigma\sigma\sigma^{i_2;c}_{i_1,i_2,j_3}+\sum_{j_3} a_{j_3-i_3}\sigma\delta\sigma^{i_2;e}_{i_1,i_2,j_3}$$
$$+\sum_{j_3} b_{j_3-i_3}\sigma\sigma\delta^{i_2;c}_{i_1,i_2,j_3}+\sum_{j_3} b_{j_3-i_3}\sigma\delta\delta^{i_2;e}_{i_1,i_2,j_3}, \tag{B.53}$$

$$dds'_{i_1,i_2,i_3}=\langle\phi_{i_1}\psi_{i_2}\psi_{i_3}|y^2z^2|\Psi\rangle=\sum_{j_3} a_{j_3-i_3}\delta\sigma\sigma^{i_2;c}_{i_1,i_2,j_3}+\sum_{j_3} a_{j_3-i_3}\delta\delta\sigma^{i_2;e}_{i_1,i_2,j_3}$$
$$+\sum_{j_3} b_{j_3-i_3}\delta\sigma\delta^{i_2;c}_{i_1,i_2,j_3}+\sum_{j_3} b_{j_3-i_3}\delta\delta\delta^{i_2;e}_{i_1,i_2,j_3}, \tag{B.54}$$

$$ssd'_{i_1,i_2,i_3}=\langle\phi_{i_1}\phi_{i_2}\psi_{i_3}|y^2z^2|\Psi\rangle=\sum_{j_3} c_{j_3-i_3}\sigma\sigma\sigma^{i_2;a}_{i_1,i_2,j_3}+\sum_{j_3} c_{j_3-i_3}\sigma\delta\sigma^{i_2;b}_{i_1,i_2,j_3}$$
$$+\sum_{j_3} e_{j_3-i_3}\sigma\sigma\delta^{i_2;a}_{i_1,i_2,j_3}+\sum_{j_3} e_{j_3-i_3}\sigma\delta\delta^{i_2;b}_{i_1,i_2,j_3}, \tag{B.55}$$

$$dsd'_{i_1,i_2,i_3}=\langle\psi_{i_1}\phi_{i_2}\psi_{i_3}|y^2z^2|\Psi\rangle=\sum_{j_3} c_{j_3-i_3}\delta\sigma\sigma^{i_2;a}_{i_1,i_2,j_3}+\sum_{j_3} c_{j_3-i_3}\delta\delta\sigma^{i_2;b}_{i_1,i_2,j_3}$$
$$+\sum_{j_3} e_{j_3-i_3}\delta\sigma\delta^{i_2;a}_{i_1,i_2,j_3}+\sum_{j_3} e_{j_3-i_3}\delta\delta\delta^{i_2;b}_{i_1,i_2,j_3}, \tag{B.56}$$

$$sdd'_{i_1,i_2,i_3}=\langle\phi_{i_1}\psi_{i_2}\psi_{i_3}|y^2z^2|\Psi\rangle=\sum_{j_3} c_{j_3-i_3}\sigma\sigma\sigma^{i_2;c}_{i_1,i_2,j_3}+\sum_{j_3} c_{j_3-i_3}\sigma\delta\sigma^{i_2;e}_{i_1,i_2,j_3}$$
$$+\sum_{j_3} e_{j_3-i_3}\sigma\sigma\delta^{i_2;c}_{i_1,i_2,j_3}+\sum_{j_3} e_{j_3-i_3}\sigma\delta\delta^{i_2;e}_{i_1,i_2,j_3}, \tag{B.57}$$

$$ddd'_{i_1,i_2,i_3}=\langle\psi_{i_1}\psi_{i_2}\psi_{i_3}|y^2z^2|\Psi\rangle=\sum_{j_3} c_{j_3-i_3}\delta\sigma\sigma^{i_2;c}_{i_1,i_2,j_3}+\sum_{j_3} c_{j_3-i_3}\delta\delta\sigma^{i_2;e}_{i_1,i_2,j_3}$$
$$+\sum_{j_3} e_{j_3-i_3}\delta\sigma\delta^{i_2;c}_{i_1,i_2,j_3}+\sum_{j_3} e_{j_3-i_3}\delta\delta\delta^{i_2;e}_{i_1,i_2,j_3}. \tag{B.58}$$

Bibliography

[1] R.M. Martin. *Electronic Structure: Basic Theory and Practical Methods*. Cambridge University Press, 2004.

[2] M. Born and R. Oppenheimer. Zur Quantentheorie der Molekeln. *Ann. Phys-Berlin*, 389(20):457–484, 1927.

[3] C.J. Cramer. *Essentials of Computational Chemistry: Theories and Models*. Wiley, 2005.

[4] W. Pauli. Über den Zusammenhang des Abschlusses der Elektronengruppen im Atom mit der Komplexstruktur der Spektren. *Z. Phys.*, 31:765–783, 1925.

[5] A. J. Coleman. Structure of Fermion Density Matrices. *Rev. Mod. Phys.*, 35:668–686, 1963.

[6] Per-Olov Löwdin. Quantum Theory of Many-Particle Systems. I. Physical Interpretations by Means of Density Matrices, Natural Spin-Orbitals, and Convergence Problems in the Method of Configurational Interaction. *Phys. Rev.*, 97:1474–1489, 1955.

[7] Martin Schütz, Georg Hetzer, and Hans-Joachim Werner. Low-order scaling local electron correlation methods. I. Linear scaling local MP2. *J. Chem. Phys.*, 111(13):5691–5705, 1999.

[8] Martin Schütz and Hans-Joachim Werner. Low-order scaling local electron correlation methods. IV. Linear scaling local coupled-cluster (LCCSD). *J. Chem. Phys.*, 114(2):661–681, 2001.

[9] Christian Ochsenfeld, Christopher A. White, and Martin Head-Gordon. Linear and sublinear scaling formation of Hartree–Fock-type exchange matrices. *J. Chem. Phys.*, 109(5):1663–1669, 1998.

[10] Kieron Burke. The ABC of DFT, 2007.

[11] P. Hohenberg and W. Kohn. Inhomogeneous Electron Gas. *Phys. Rev.*, 136:B864–B871, 1964.

[12] Mel Levy. Universal variational functionals of electron densities, first-order density matrices, and natural spin-orbitals and solution of the v-representability problem. *Proc. Natl. Acad. Sci. U. S. A.*, 76(12):6062–6065, 1979.

[13] W. Kohn and L. J. Sham. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.*, 140:A1133–A1138, 1965.

[14] J. C. Slater. The Theory of Complex Spectra. *Phys. Rev.*, 34:1293–1322, 1929.

[15] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos. Iterative minimization techniques for *ab initio* total-energy calculations: molecular dynamics and conjugate gradients. *Rev. Mod. Phys.*, 64:1045–1097, 1992.

[16] C. L. Fu and K. M. Ho. First-principles calculation of the equilibrium ground-state properties of transition metals: Applications to Nb and Mo. *Phys. Rev. B*, 28:5480–5486, 1983.

[17] M J Gillan. Calculation of the vacancy formation energy in aluminium. *J. Phys.: Condens. Matter*, 1(4):689, 1989.

[18] Murray Gell-Mann and Keith A. Brueckner. Correlation Energy of an Electron Gas at High Density. *Phys. Rev.*, 106:364–368, 1957.

[19] W. J. Carr and A. A. Maradudin. Ground-State Energy of a High-Density Electron Gas. *Phys. Rev.*, 133:A371–A374, 1964.

[20] P. Nozieres and D. Pines. *Theory Of Quantum Liquids*. Advanced Books Classics. Westview Press, 1999.

[21] W. J. Carr. Energy, Specific Heat, and Magnetic Properties of the Low-Density Electron Gas. *Phys. Rev.*, 122:1437–1446, 1961.

[22] A. D. Becke. Density-functional exchange-energy approximation with correct asymptotic behavior. *Phys. Rev. A*, 38:3098–3100, 1988.

[23] John P. Perdew and Yue Wang. Accurate and simple analytic representation of the electron-gas correlation energy. *Phys. Rev. B*, 45:13244–13249, 1992.

[24] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.*, 77:3865–3868, 1996.

[25] J. P. Perdew and Alex Zunger. Self-interaction correction to density-functional approximations for many-electron systems. *Phys. Rev. B*, 23:5048–5079, 1981.

[26] A. Svane and O. Gunnarsson. Localization in the self-interaction-corrected density-functional formalism. *Phys. Rev. B*, 37:9919–9922, 1988.

[27] Jianmin Tao, John P. Perdew, Viktor N. Staroverov, and Gustavo E. Scuseria. Climbing the Density Functional Ladder: Nonempirical Meta–Generalized Gradient Approximation Designed for Molecules and Solids. *Phys. Rev. Lett.*, 91:146401, 2003.

[28] Roberto Peverati and Donald G. Truhlar. Screened-exchange density functionals with broad accuracy for chemistry and solid-state physics. *Phys. Chem. Chem. Phys.*, 14:16187–16191, 2012.

[29] Axel D. Becke. A new mixing of Hartree–Fock and local density-functional theories. *J. Chem. Phys.*, 98(2):1372–1377, 1993.

[30] Axel D. Becke. Density-functional thermochemistry. III. The role of exact exchange. *J. Chem. Phys.*, 98(7):5648–5652, 1993.

[31] John P. Perdew, Matthias Ernzerhof, and Kieron Burke. Rationale for mixing exact exchange with density functional approximations. *J. Chem. Phys.*, 105(22):9982–9985, 1996.

[32] Giovanni B. Bachelet and M. Schlüter. Relativistic norm-conserving pseudopotentials. *Phys. Rev. B*, 25:2103–2108, 1982.

[33] S. Goedecker, M. Teter, and J. Hutter. Separable dual-space Gaussian pseudopotentials. *Phys. Rev. B*, 54:1703–1710, 1996.

[34] C. Hartwigsen, S. Goedecker, and J. Hutter. Relativistic separable dual-space Gaussian pseudopotentials from H to Rn. *Phys. Rev. B*, 58:3641–3662, 1998.

[35] M. Krack. Pseudopotentials for H to Kr optimized for gradient-corrected exchange-correlation functionals. *Theor. Chem. Acc.*, 114:145–152, 2005.

[36] W. Kohn. Density Functional and Density Matrix Method Scaling Linearly with the Number of Atoms. *Phys. Rev. Lett.*, 76:3168–3171, 1996.

[37] Nicola Marzari and David Vanderbilt. Maximally localized generalized Wannier functions for composite energy bands. *Phys. Rev. B*, 56:12847–12865, 1997.

[38] Jacques Des Cloizeaux. Energy Bands and Projection Operators in a Crystal: Analytic and Asymptotic Properties. *Phys. Rev.*, 135:A685–A697, 1964.

[39] Jacques Des Cloizeaux. Analytical Properties of n-Dimensional Energy Bands and Wannier Functions. *Phys. Rev.*, 135:A698–A707, 1964.

[40] W. Kohn. Analytic Properties of Bloch Waves and Wannier Functions. *Phys. Rev.*, 115:809–821, 1959.

[41] Roi Baer and Martin Head-Gordon. Sparsity of the Density Matrix in Kohn-Sham Density Functional Theory and an Assessment of Linear System-Size Scaling Methods. *Phys. Rev. Lett.*, 79:3962–3965, 1997.

[42] Sohrab Ismail-Beigi and T. A. Arias. Locality of the Density Matrix in Metals, Semiconductors, and Insulators. *Phys. Rev. Lett.*, 82:2127–2130, 1999.

[43] S. Goedecker. Decay properties of the finite-temperature density matrix in metals. *Phys. Rev. B*, 58:3501–3502, 1998.

[44] Lixin He and David Vanderbilt. Exponential Decay Properties of Wannier Functions and Related Quantities. *Phys. Rev. Lett.*, 86:5341–5344, 2001.

[45] N.H. March, W.H. Young, and S. Sampanthar. *The Many-body Problem in Quantum Mechanics*. Dover books on physics. Dover Publications, Incorporated, 1967.

[46] Stefan Goedecker. Linear scaling electronic structure methods. *Rev. Mod. Phys.*, 71:1085–1123, 1999.

[47] S. Goedecker and L. Colombo. Efficient Linear Scaling Algorithm for Tight-Binding Molecular Dynamics. *Phys. Rev. Lett.*, 73:122–125, 1994.

[48] S. Goedecker and M. Teter. Tight-binding electronic-structure calculations and tight-binding molecular dynamics with localized orbitals. *Phys. Rev. B*, 51:9455–9464, 1995.

[49] S. Goedecker. Low Complexity Algorithms for Electronic Structure Calculations. *J. Comput. Phys.*, 118(2):261 – 268, 1995.

[50] Otto F. Sankey, David A. Drabold, and Andrew Gibson. Projected random vectors and the recursion method in the electronic-structure problem. *Phys. Rev. B*, 50:1376–1381, 1994.

[51] Uwe Stephan and David A. Drabold. Order-N projection method for first-principles computations of electronic quantities and Wannier functions. *Phys. Rev. B*, 57:6391–6407, 1998.

[52] Weitao Yang. Direct calculation of electron density in density-functional theory. *Phys. Rev. Lett.*, 66:1438–1441, 1991.

[53] Weitao Yang. A local projection method for the linear combination of atomic orbital implementation of density-functional theory. *J. Chem. Phys.*, 94(2):1208–1214, 1991.

[54] Weitao Yang and Tai-Sung Lee. A density-matrix divide-and-conquer approach for electronic structure calculations of large molecules. *J. Chem. Phys.*, 103(13):5674–5678, 1995.

[55] X.-P. Li, R. W. Nunes, and David Vanderbilt. Density-matrix electronic-structure method with linear system-size scaling. *Phys. Rev. B*, 47:10891–10894, 1993.

[56] R. McWeeny. Some Recent Advances in Density Matrix Theory. *Rev. Mod. Phys.*, 32:335–369, 1960.

[57] Francesco Mauri, Giulia Galli, and Roberto Car. Orbital formulation for electronic-structure calculations with linear system-size scaling. *Phys. Rev. B*, 47:9973–9976, 1993.

[58] Pablo Ordejón, David A. Drabold, Matthew P. Grumbach, and Richard M. Martin. Unconstrained minimization approach for electronic computations that scales linearly with system size. *Phys. Rev. B*, 48:14646–14649, 1993.

[59] Pablo Ordejón, David A. Drabold, Richard M. Martin, and Matthew P. Grumbach. Linear system-size scaling methods for electronic-structure calculations. *Phys. Rev. B*, 51:1456–1476, 1995.

[60] Francesco Mauri and Giulia Galli. Electronic-structure calculations and molecular-dynamics simulations with linear system-size scaling. *Phys. Rev. B*, 50:4316–4326, 1994.

[61] Jeongnim Kim, Francesco Mauri, and Giulia Galli. Total-energy global optimizations using nonorthogonal localized orbitals. *Phys. Rev. B*, 52:1640–1648, 1995.

[62] W. Hierse and E. B. Stechel. Order-N methods in self-consistent density-functional calculations. *Phys. Rev. B*, 50:17811–17819, 1994.

[63] E. Hernández and M. J. Gillan. Self-consistent first-principles technique with linear scaling. *Phys. Rev. B*, 51:10157–10160, 1995.

[64] Chris-Kriton Skylaris, Peter D. Haynes, Arash A. Mostofi, and Mike C. Payne. Introducing [small-caps ONETEP]: Linear-scaling density functional simulations on parallel computers. *J. Chem. Phys.*, 122(8):084119, 2005.

[65] D. R. Bowler, R. Choudhury, M. J. Gillan, and T. Miyazaki. Recent progress with large-scale ab initio calculations: the CONQUEST code. *Phys. Status Solidi B*, 243(5):989–1000, 2006.

[66] Peter David Haynes. *Linear-scaling methods in ab initio quantum-mechanical calculations*. PhD thesis, 1998.

[67] I. Daubechies et al. *Ten lectures on wavelets*, volume 61. SIAM, 1992.

[68] Alfred Haar. Zur Theorie der orthogonalen Funktionensysteme. *Math. Ann.*, 69(3):331–371, 1910.

[69] S. Goedecker. *Wavelets and Their Application: For the Solution of Partial Differential Equations in Physics*. Presses polytechniques et universitaires romandes, 1998.

[70] A.I. Neelov and S. Goedecker. An efficient numerical quadrature for the calculation of the potential energy of wavefunctions expressed in the Daubechies wavelet basis. *J. Comput. Phys.*, 217(2):312 – 339, 2006.

[71] Luigi Genovese, Alexey Neelov, Stefan Goedecker, Thierry Deutsch, Seyed Alireza Ghasemi, Alexander Willand, Damien Caliste, Oded Zilberberg, Mark Rayson, Anders Bergman, and Reinhold Schneider. Daubechies wavelets as a basis set for density functional pseudopotential calculations. *J. Chem. Phys.*, 129(1):014109, 2008.

[72] Péter Pulay. Convergence acceleration of iterative sequences. the case of scf iteration. *Chem. Phys. Lett.*, 73(2):393 – 398, 1980.

[73] E.K.P. Chong and S.H. Zak. *An Introduction to Optimization*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2011.

[74] Per-Olov Lowdin. On the Non-Orthogonality Problem Connected with the Use of Atomic Wave Functions in the Theory of Molecules and Crystals. *J. Chem. Phys.*, 18(3):365–375, 1950.

[75] S. Goedecker and C. J. Umrigar. Critical assessment of the self-interaction-corrected–local-density-functional method and its algorithmic implementation. *Phys. Rev. A*, 55:1765–1771, 1997.

[76] C. G. Baker, U. L. Hetmaniuk, R. B. Lehoucq, and H. K. Thornquist. Anasazi software for the numerical solution of large-scale eigenvalue problems. *ACM T. Math. Software*, 36(3):13:1–13:23, 2009.

[77] Vicente Hernandez, Jose E. Roman, and Vicente Vidal. SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM T. Math. Software*, 31(3):351–362, 2005.

[78] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007.

[79] H. Hellmann. *Einführung in die Quantenchemie*. Franz Deuticke, 1937.

[80] R. P. Feynman. Forces in molecules. *Phys. Rev.*, 56:340–343, 1939.

[81] P. Pulay. Ab initio calculation of force constants and equilibrium geometries in polyatomic molecules. *Mol. Phys.*, 17(2):197–204, 1969.

[82] Erik Bitzek, Pekka Koskinen, Franz Gähler, Michael Moseler, and Peter Gumbsch. Structural Relaxation Made Simple. *Phys. Rev. Lett.*, 97:170201, 2006.

[83] Message Passing Interface Forum. MPI: A Message-Passing Interface Standard Version 3.0, 2012.

[84] OpenMP Archtitecture Review Board. OpenMP Application Program Interface. Version 3.1, 2011.

[85] Luigi Genovese, Matthieu Ospici, Thierry Deutsch, Jean-Francois Mehaut, Alexey Neelov, and Stefan Goedecker. Density functional theory calculation on many-cores hybrid central processing unit-graphic processing unit architectures. *J. Chem. Phys.*, 131(3):034103, 2009.

[86] Luigi Genovese, Thierry Deutsch, Alexey Neelov, Stefan Goedecker, and Gregory Beylkin. Efficient solution of Poisson's equation with free boundary conditions. *J. Chem. Phys.*, 125(7):074105, 2006.

[87] Luigi Genovese, Thierry Deutsch, and Stefan Goedecker. Efficient and accurate three-dimensional Poisson solver for surface problems. *J. Chem. Phys.*, 127(5):054704, 2007.

[88] Gene M Amdahl. Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities, Reprinted from the AFIPS Conference Proceedings, Vol. 30 (Atlantic City, N.J., Apr. 18–20), AFIPS Press, Reston, Va., 1967, pp. 483–485, when Dr. Amdahl was at Inte. *IEEE Solid State Circuits Mag.*, 12(3):19–20, 2007.

[89] H. W. Kroto, J. R. Heath, S. C. O'Brien, R. F. Curl, and R. E. Smalley. C60: Buckminsterfullerene. *Nature*, 318(6042):162–163, 1985.

[90] W. Krätschmer, Lowell D. Lamb, K. Fostiropoulos, and Donald R. Huffman. Solid C60: a new form of carbon. *Nature*, 347(6291):354–358, 1990.

[91] Yufeng Zhao, Yong-Hyun Kim, A. C. Dillon, M. J. Heben, and S. B. Zhang. Hydrogen Storage in Novel Organometallic Buckyballs. *Phys. Rev. Lett.*, 94:155504, 2005.

[92] Ting Guo, Changming Jin, and R. E. Smalley. Doping bucky: formation and properties of boron-doped buckminsterfullerene. *J. Phys. Chem.*, 95(13):4948–4950, 1991.

[93] Paul W. Dunk, Antonio Rodríguez-Fortea, Nathan K. Kaiser, Hisanori Shinohara, Josep M. Poblet, and Harold W. Kroto. Formation of Heterofullerenes by Direct Exposure of C60 to Boron Vapor. *Angew. Chem.-Ger. Edit.*, 125(1):333–337, 2013.

[94] Lars Hultman, Sven Stafström, Zsolt Czigány, Jörg Neidhardt, Niklas Hellgren, Ian F. Brunell, Kazu Suenaga, and Christian Colliex. Cross-Linked Nano-onions of Carbon Nitride in the Solid Phase: Existence of a Novel $C_{48}N_{12}$ Aza-Fullerene. *Phys. Rev. Lett.*, 87:225503, 2001.

[95] Haibo Wang, Thandavarayan Maiyalagan, and Xin Wang. Review on Recent Progress in Nitrogen-Doped Graphene: Synthesis, Characterization, and Its Potential Applications. *ACS Catal.*, 2(5):781–794, 2012.

[96] Rui-Hua Xie, Garnett W. Bryant, Jijun Zhao, Vedene H. Smith, Aldo Di Carlo, and Alessandro Pecchia. Tailorable Acceptor $C_{60-n}B_n$ and Donor $C_{60-m}N_m$ Pairs for Molecular Electronics. *Phys. Rev. Lett.*, 90:206602, 2003.

[97] Chris J Pickard and R J Needs. Ab initio random structure searching. *J. Phys.: Condens. Matter*, 23(5):053201, 2011.

[98] Colin W. Glass, Artem R. Oganov, and Nikolaus Hansen. USPEX—Evolutionary crystal structure prediction . *Comput. Phys. Commun.*, 175(11-12):713 – 720, 2006.

[99] Yanchao Wang, Jian Lv, Li Zhu, and Yanming Ma. CALYPSO: A method for crystal structure prediction . *Comput. Phys. Commun.*, 183(10):2063 – 2070, 2012.

[100] Stefan Goedecker. Minima hopping: An efficient search method for the global minimum of the potential energy surface of complex molecular systems. *J. Chem. Phys.*, 120(21):9911–9917, 2004.

[101] Isha Garg, Hitesh Sharma, Keya Dharamvir, and V.K. Jindal. Substitutional Patterns in Boron Doped Heterofullerenes $C_{60\text{-}n}B_n$ (n = 1-12). *J. Comput. Theor. Nanosci.*, 8(4):642–655, 2011.

[102] L. Viani and M.C. Dos Santos. Comparative study of lower fullerenes doped with boron and nitrogen. *Solid State Commun.*, 138(10-11):498–501, 2006.

[103] M. Riad Manaa, Heather a. Ichord, and David W. Sprehn. Predicted molecular structure of novel $C_{48}B_{12}$. *Chem. Phys. Lett.*, 378(3-4):449–455, 2003.

[104] M Riad Manaa, David W Sprehn, and Heather a Ichord. Prediction of extended aromaticity for a novel C(48)N(12) azafullerene structure. *J. Am. Chem. Soc.*, 124(47):13990–1, 2002.

[105] D. Wales. *Energy Landscapes: Applications to Clusters, Biomolecules and Glasses*. Cambridge Molecular Science. Cambridge University Press, 2003.

[106] Stephan Mohr, Pascal Pochet, Maximilian Amsler, Bastian Schaefer, Ali Sadeghi, Luigi Genovese, and Stefan Goedecker. Boron aggregation in the ground states of boron-carbon fullerenes, 2013.

[107] Pascal Pochet, Luigi Genovese, Sandip De, Stefan Goedecker, Damien Caliste, S. Alireza Ghasemi, Kuo Bao, and Thierry Deutsch. Low-energy boron fullerenes: Role of disorder and potential synthesis pathways. *Phys. Rev. B*, 83:081403, 2011.

[108] Paul Boulanger, Maxime Moriniere, Luigi Genovese, and Pascal Pochet. Selecting boron fullerenes by cage-doping mechanisms. *J. Chem. Phys.*, 138(18):184302, 2013.

[109] Nevill Gonzalez Szwacki, Arta Sadrzadeh, and Boris I. Yakobson. B_{80} Fullerene: An *Ab Initio* Prediction of Geometry, Stability, and Electronic Structure. *Phys. Rev. Lett.*, 98:166804, 2007.

[110] A. Ayuela, P. W. Fowler, D. Mitchell, R. Schmidt, G. Seifert, and F. Zerbetto. C62: Theoretical Evidence for a Nonclassical Fullerene with a Heptagonal Ring. *J. Phys. Chem.*, 100(39):15634–15636, 1996.

[111] Wenyuan Qian, Shih-Ching Chuang, Roberto B. Amador, Thibaut Jarrosson, Michael Sander, Susan Pieniazek, Saeed I. Khan, and Yves Rubin. Synthesis of Stable Derivatives of C62: The First Nonclassical Fullerene Incorporating a Four-Membered Ring. *J. Am. Chem. Soc.*, 125(8):2066–2067, 2003.

[112] Yan-Hong Cui, De-Li Chen, Wei Quan Tian, and Ji-Kang Feng. Structures, Stabilities, and Electronic and Optical Properties of C62 Fullerene Isomers. *J. Phys. Chem. A*, 111(32):7933–7939, 2007.

[113] C. R. Hsing, C. M. Wei, N. D. Drummond, and R. J. Needs. Quantum Monte Carlo studies of covalent and metallic clusters: Accuracy of density functional approximations. *Phys. Rev. B*, 79:245401, 2009.

[114] Miguel A.L. Marques, Micael J.T. Oliveira, and Tobias Burnus. Libxc: A library of exchange and correlation functionals for density functional theory . *Comput. Phys. Commun.*, 183(10):2272 – 2281, 2012.

[115] M. Riad Manaa. C48N12 and C48B12 as a donor–acceptor pair for molecular electronics. *Chem. Phys. Lett.*, 382(1–2):194 – 197, 2003.

[116] Zhongfang Chen, Haijun Jiao, Damian Moran, Andreas Hirsch, Walter Thiel, and Paul von Ragué Schleyer. Aromatic stabilization in heterofullerenes $C_{48}X_{12}$ (X = N, P, B, Si). *J. Phys. Org. Chem.*, 16(10):726–730, 2003.

[117] S. Stafström, L Hultman, and N Hellgren. Predicted stability of a new aza[60]fullerene molecule, C48N12. *Chem. Phys. Lett.*, 340(3-4):227–231, 2001.

[118] F.a. Shakib and M.R. Momeni. Isolation: A strategy for obtaining highly doped heterofullerenes. *Chem. Phys. Lett.*, 514(4-6):321–324, 2011.

[119] H. W. Kroto. The stability of the fullerenes C_n, with n = 24, 28, 32, 36, 50, 60 and 70. *Nature*, 329(6139):529–531, 1987.

[120] Ali Sadeghi, S. Alireza Ghasemi, Markus A. Lill, and Stefan Goedecker. Metrics for measuring distances in configuration spaces, 2013.

[121] Sandip De, Alexander Willand, Maximilian Amsler, Pascal Pochet, Luigi Genovese, and Stefan Goedecker. Energy Landscape of Fullerene Materials: A Comparison of Boron to Boron Nitride and Carbon. *Phys. Rev. Lett.*, 106:225502, 2011.

[122] Hua-Jin Zhai, Boggavarapu Kiran, Jun Li, and Lai-Sheng Wang. Hydrocarbon analogues of boron clusters–planarity, aromaticity and antiaromaticity. *Nat. Mater.*, 2(12):827–33, 2003.

[123] G. Beylkin. On the Representation of Operators in Bases of Compactly Supported Wavelets. *SIAM J. Numer. Anal.*, 29(6):1716–1740, 1992.

Stephan Mohr

Benkenstrasse 26
CH-4054 Basel
☎ *+41 61 281 11 60*
+41 76 421 88 53
✉ *stephan.mohr@unibas.ch*

Personal Information

date of birth	10.10.1985
place of birth	Basel, Switzerland
citizenship	Swiss

Education

April 2010 – June 2013	**Ph.D. in Physics**, *University of Basel*, Switzerland.
August 2008 – March 2010	**M.Sc. in Physics**, *University of Basel*, Switzerland.
October 2005 – July 2008	**B.Sc. in Physics**, *University of Basel*, Switzerland.

PhD Thesis

title	*Fast and accurate electronic structure methods: large systems and applications to boron-carbon heterofullerenes*
supervisor	Prof. Dr. Stefan Goedecker
co-referee	Dr. Thierry Deutsch

Master Thesis

title	*Using saddle points in the context of minima hopping*
supervisor	Prof. Dr. Stefan Goedecker

Publications

2013 Accurate minimal basis set for electronic structure calculations of large systems, to be submitted

2013 Stephan Mohr, Pascal Pochet, Maximilian Amsler, Bastian Schaefer, Ali Sadeghi, Luigi Genovese, Stefan Goedecker. Boron aggregation in the ground states of boron-carbon fullerenes. arXiv:1305.2302

2011 Michael Sicher, Stephan Mohr, and Stefan Goedecker. Efficient moves for global geometry optimization methods and their application to binary systems. J. Chem. Phys. **134**, 044106