UiT
THE ARCTIC UNIVERSITY OF NORWAY

Faculty of Science and Technology
Department of Physics and Technology

Real-space all-electron Density Functional Theory with Multiwavelets

—

Stig Rune Jensen

A dissertation for the degree of Philosophiae Doctor – February 2014

Abstract

This thesis presents the implementation of a numerical real-space method for the calculation of the electronic structure of molecular systems within the self-consistent field approximations of quantum chemistry. The code is based on the multi-resolution multiwavelet basis which provide sparse representations of functions and operators, in particular integral operators with Green's function convolution kernels. The mathematical formalism provides efficient (linear-scaling) algorithms for operator application, e.g. for the Coulomb operator for the calculation of electrostatic potentials, as well as rigorous error control.

The Hartree-Fock and Kohn-Sham equations of quantum chemistry are reformulated in integral form and solved to self-consistency using iterative solution techniques. The code is able to attain high-accuracy for many-electron molecular systems, both restricted closed-shell and unrestricted open-shell.

Because of the inherent high demands on computational resources that comes with real-space methods, the code relies on parallel algorithms and data distribution in order to become competitive with conventional methods, and the code has been properly adapted in order to utilize modern massively parallel computing architectures.

Acknowledgments

I would like to express my gratitude to my supervisors, Luca Frediani, for teaching me what I know about quantum chemistry, and Tor Flå, for teaching me the theory of multiwavelets, and to them both for giving my a lot of freedom to persue my own interests and ideas. Luca has had a lot of things going on in his life during these years, but his door has always been open (literally, his home front door) if I ever needed advice or guidence.

I would like to give my sincere thanks to Jonas Jusélius, who I had the privilege to work with during my master's and the first years of my PhD. I might have known the basic concepts of programming when I met you, but you turned me into a programmer. I miss our sharing of WTFs over a code that doesn't work, and I hope we get the chance to work together again soon.

I would also like to mention the rest of the people at the HPC group who have been most accommodating in their support, by tweaking the hardware to suite my needs and by helping me cut in line when my jobs were to big for the queue.

I would like to give thanks to the CTCC group in Tromsø, to the people I have shared office with, Jonas, Arnfinn and Krzysztof (and various random people at our guest spot). To Peter and Antoine, who is/was also working on the multiwavelet project, and to Marco, Arnfinn, Roberto, Maarten, Luca (Oggioni) and Geir who joined me in various classes over the years (I have been the only student in several classes in the past, so your company has been appreciated), and to the rest of the lunch-eaters in the group. Finally, to the people at the CTCC in Oslo for many enjoyable joint meetings.

Contents

List of papers

This thesis is based on the following scientific papers:

I "Adaptive order polynomial algorithm in a multiwavelet representation scheme"; A. Durdek, S. R. Jensen, J. Jusèlius, P. Wind, T. Flå and L. Frediani; Submitted to *Applied Numerical Mathematics*

II "Linear scaling Coulomb iteraction in the multiwavelet basis, a parallel implementation"; S. R. Jensen, J. Jusèlius, A. Durdek, T. Flå, P. Wind and L. Frediani; Submitted to *International Journal of Modeling, Simulation and Scientific Computing*

III "Real-Space Density Functional Theory with Localized Orbitals and Multiwavelets"; S. R. Jensen, J, Jusèlius, A. Durdek, P. Wind, T. Flå and L. Frediani; Manuscript in preparation

Chapter 1

Introduction

1.1 Numerical analysis and real-world physics

The aim of the natural sciences is to model the complex processes occuring in nature as accurately as possible. It is a remarkable fact that the fundamental features of nature are so well described in terms of mathematics, by simple and elegant expressions like the wave equation, Newton's laws of motion and gravitation, and Maxwell's equations of electrodynamics. Equally remarkable is it that these simple expressions can give rise to the vast complexity that we observe in the world around us.

The underlying complexity of the these equations means that analytic solutions are available only for very simple, idealized systems, often with high symmetry, thus limiting their practial usefullness. Over the years, not few science students have been questioning the applicability of computing a cannon ball's trajecory in vacuum or the electric field around a point charge alone in the universe.

The bridge between the idealized model systems and what we observe in the real world is made through numerical analysis, which involves the translation of the physical equations into the language of the digital computer. Most modern applied sciences relies heavily upon numerical analysis and simulations, either for performing numerically intensive calculations or for analysing large amounts of data. Over the last decades computer simulation has emerged as a third way

in science besides the experimental and theoretical approach, and has become an indispesable tool for the investigation and prediction of physical and chemical processes.

However, with the breakdown of Moore's law (on a single computational device) the computational scientist cannot blindly rely on the advances of computer technology in order to push the limits of the attainable accuracy and the size of the systems, and a lot more responsibility is put back to the computational scientist in developing algorithms suitable for parallel execution. While the computational speed of a processor no longer can be said to double every second year, Moore's law continues to be valid in a more fundamental sense, as the number of transistors continues to grow, but in the form of multi-core processors. This means this in the future we might see a paradigm change where currently inferior numerical methods and algorithms will enter the stage because of favourable scaling with respect to system size *and* with the number of processors.

1.2 Chemistry without chemicals

Scientists have for centuries sought an *ab initio* theory of chemical phenomena, where molecular structure, properties and reactions can be computed with a minimal amount of empirical parameters, but without the fundamental knowledge of the building blocks of matter this was for a long time a hopeless endeavor. With the introduction of quantum mechanics almost a century ago, the complete physical theory for molecular systems became available, but although the exact problem is decievingly simple to state for an arbitrary system through the Schrödinger equation

$$\hat{H}\Psi = E\Psi \tag{1.1}$$

its solution for many-body problems is quite the opposite. In fact, whenever the system contains more than two particles the problem *cannot* be solved (at least not in the usual sense in terms of the standard elementary functions of calculus).

The most common approach in modern computational chemistry is the self-consistent field approximations that are based on the familiar chemical concept

of one-electron orbitals φ_i, each a solution of a Schrödinger-like equation

$$\hat{F}\varphi_i = \epsilon_i \varphi_i \tag{1.2}$$

While the solution of this set of N coupled, nonlinear, three-dimensional partial differential equations is still a formidable computational task, the complexity of the full $3N$-dimensional Schrödinger equation has been sufficiently reduced for the numerical solution to be feasable for systems with a remarkable number of particles.

This has been made possible by combining a great deal of chemical intuition into the development of computational methods. In particular, the introduction of the atomic orbital basis in the form of atom-centered Gaussians can be attributed most of the success of modern computational chemistry, by providing efficient and compact representations with a consistent cancellation of errors.

However, although the Gaussian basis is ideal for obtaining qualitative numbers fast, it struggles when high precision is required. Moreover, as the Gaussian functions extend throughout the entire system, it is difficult to reduce the problem into truly independent tasks that can be easily distributed among several computers and executed simultaneously.

The alternative to the elegant, compact representations using a carefully chosen, preoptimized atomic orbital basis, would be a brute force numerical solution using real-space representations in terms of numerical grids or finite elements. Such an approach would yield robust, unbiased results that do not rely on cancellation of errors (but neither would it benefit from it).

It is a well-known fact that the electronic density in molecular systems is rapidly varying in the vicinity of the atomic nuclei, and a usual problem with real-space methods is that an accurate treatment of the system requires high resolution of grid points in the nuclear regions. Keeping this high resolution uniformly througout the computaitonal domain would yield unnecessary high accuracy in the interatomic regions, thus the real-space treatment of molecular systems is demanding a *multiresolution* framework in order to achieve numerical efficiency.

1.3 Multiwavelets

As the theory of wavelets is vast and can be considered a rather advanced topic of applied mathematics, it remains unfamiliar to most chemists. However, Alpert's[1] construction of *multiwavelets* is rather simple. Starting with a small set of polynomials $\{\phi_j\}_{j=0}^{k}$ of order $\leq k$ on the unit interval, we attempt to represent a given function. If this basis turns out to be too crude to accurately describe the function, we can increase the flexibility by adding higher order polynomials (thus increasing the polynomial order k), and we approach a complete basis (and an exact representation) as $k \to \infty$.

Alpert shows that there is a second way to approach completeness in this basis. Instead of increasing the polynomial order, we split the interval and double the number of basis functions by dilating and translating the original basis to both subintervals

$$\phi_{j,l}^{1}(x) = 2^{1/2}\phi_j(2x - l), \qquad l = 0, 1 \tag{1.3}$$

The splitting procedure can be continued until we have reached a scale n where we are satisfied with the accuracy of the representation. At this level of refinement the unit interval has been split into 2^n intervals, each of size 2^{-n} containing a dilated and translated version of the original k-order basis

$$\phi_{j,l}^{n}(x) = 2^{n/2}\phi_j(2^n x - l), \qquad l = 0, \ldots, 2^n - 1 \tag{1.4}$$

This basis can be used to represent any square integrable function to any finite accuracy by adjusting the polynomial order k and/or the level of refinement n. The construction in three dimensions is similar, where at refinement level n the unit cube has been uniformly divided into 2^{3n} subcubes.

The main advantage of multiwavelets over the similar finite element bases is the possibility of constructing non-uniform grids, and thus focusing the computational efforts into the problematic nuclear region. Moreover, the grid construction can be completely automated to yield representations with guaranteed accuracy.

Although similar constructions were already familiar through the multigrid approaches within the finite element community, these methods suffered from a lack of mathematical rigour and generality, with complicated problem-specific

algorithms. Alpert's construction, on the other hand, was founded upon the well established, powerful theory of wavelets, making the basis applicable to a wide variety of physical problems and operators, yielding sparse representations and fast algorithms.

1.4 Organization of the thesis

The multiwavelet basis is described in detail within the framework of multiresolution analysis in Chap. 2, and the practical implementation of this formalism into a working computer code is presented in Chap. 3. In particular, we describe the mathematical operations necessary in order to solve the equations appearing in the self-consistent field methods of quantum chemistry. An introduction to these methods is given in Chap. 4, together with algorithms for their numerical solution. Finally, in Chap. 5, a brief discussion is given on the orbital-free methods of density functional theory, and some preliminary results are presented.

Included in this thesis are also three papers submitted for publication, that can be considered linked to each of the three main chapters. The first paper involves the construction of the multiwavelet basis and is an attempt to reduce the memory requirements of the method by decreasing the polynomial order k of the basis as the level of refinement n is increased.

The second paper describes the parallel implementation of the code with particular focus on the calculation of electrostatic potentials. The performance of the code (numerical accuracy, linear scaling of computational time with respect to system size, and parallel efficiency) is demonstrated on realistic molecular systems of up to 600 atoms.

The topic of the third paper is the solution of the self-consistent field problem in quantum chemistry. General algorithms are presented for the iterative solution of the Hartree-Fock and Kohn-Sham equations for many-electron systems in both a canonical and localized orbital framework. High accuracy energies are presented for small molecules, while robust and fast convergence is demonstrated for small and medium sized systems (less than 100 electrons).

Chapter 2

Multiresolution analysis

In this chapter a general introduction to multiwavelet theory will be given through the concept of multiresolution analysis (MRA)[1], that was developed by Mallat[2] and Daubechies[3] in the late 1980s. A detailed description of MRAs can be found in Keinert[4], from which a brief summary of the key issues are given in the following, with the difference that we limit our discussion to the unit interval instead of the real line.

2.1 Orthogonal MRA

A multiresolution analysis of $L^2([0,1])$ is an infinite nested sequence of subspaces

$$V_k^0 \subset V_k^1 \subset \cdots \subset V_k^n \subset \cdots \subset L^2([0,1]) \tag{2.1}$$

with the following properties

1. $\bigcup_{n=0}^{\infty} V_k^n$ *is dense in* $L^2([0,1])$.
2. $f(x) \in V_k^n \iff f(2x) \in V_k^{n+1}$, $\forall n \in \mathbb{N}$.
3. $f(x) \in V_k^n \iff f(x - 2^{-n}l) \in V_k^n$, $\forall n \in \mathbb{N},\ 0 \leq l \leq 2^n - 1$.
4. There exists a function vector $\boldsymbol{\phi}$ in $L^2([0,1])$ of length $k+1$ such that the vector components ϕ_i forms a basis of V_k^0.

[1] Mallat[2] uses the term multiresolution *approximation*, but in this work we will use multiresolution *analysis*, as it is more commonly used in the literature.

This means that if we can construct a basis of V_k^0, which consists of only $k+1$ functions, we can construct a basis of *any* space V_k^n, by simple compression (by a factor of 2^n, property 2), and translations (to all dyadic grid points at scale n, property 3), of the original $k+1$ functions, and by increasing the scale n, we are approaching a complete basis of $L^2([0,1])$. Since $V_k^n \subset V_k^{n+1}$ the basis functions of V_k^n can be expanded in the basis of V_k^{n+1}

$$\phi_{i,l}^n(x) \stackrel{\text{def}}{=} 2^{n/2}\phi_i(2^n x - l) = \sum_{m=0}^{2^n-1}\sum_{j=0}^{k} H_{ij}^{(m)}\phi_{j,m}^{n+1}(x) \tag{2.2}$$

where $H^{(m)}$ are the so-called filter matrices that describe the transformation between different spaces V_k^n. The MRA is called orthogonal if

$$\langle \phi_{i,l}^n, \phi_{j,m}^n \rangle = \delta_{i,j}\delta_{l,m} \tag{2.3}$$

This orthogonality condition means that the functions are orthogonal both within one function vector and through all possible translations on one scale, but *not* through the different scales.

Complementary to the nested sequence of subspaces V_k^n, we can define another series of spaces W_k^n that complements V_k^n in V_k^{n+1}

$$V_k^{n+1} = V_k^n \oplus W_k^n \tag{2.4}$$

where there exists another function vector $\boldsymbol{\psi}$ of lenght $k+1$ that, with all its translations on scale n form a basis for W_k^n. Analogously to Eq. (2.2) the function vector can be expanded in the basis of V_k^{n+1}

$$\psi_{i,l}^n(x) \stackrel{\text{def}}{=} 2^{n/2}\psi_i(2^n x - l) = \sum_{m=0}^{2^n-1}\sum_{j=0}^{k} G_{ij}^{(m)}\phi_{j,m}^{n+1}(x) \tag{2.5}$$

with filter matrices $G^{(m)}$. In orthogonal MRA the functions $\boldsymbol{\psi}$ fulfill the same othogonality condition as Eq. (2.3), and if we combine Eq. (2.1) and Eq. (2.4) we see that they must also be orthogonal with respect to different scales

$$\langle \psi_{j,l}^n, \psi_{i,m}^{n'} \rangle = \delta_{i,j}\delta_{l,m}\delta_{n,n'} \tag{2.6}$$

Recursive application of Eq. (2.4) yields the important relation

$$V_k^n = V_k^0 \oplus W_k^0 \oplus W_k^1 \oplus \cdots \oplus W_k^{n-1} \tag{2.7}$$

2.2 Multiwavelets

There are many ways to choose the basis functions $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ (which define the spanned spaces V_k^n and W_k^n), leading to different wavelet families. There is a one-to-one correspondence between the basis functions $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$, and the filter matrices $H^{(m)}$ and $G^{(m)}$ used in the two-scale relations Eq. (2.2) and Eq. (2.5), and most well known wavelet families are defined only through their filter coefficients, such as Daubechies' family of compactly supported wavelets[3].

In the following we are taking a different approach, which follows the original construction of multiwavelets by Alpert[1]. We define the *scaling space* V_k^n as the space of piecewise polynomials

$$\begin{aligned} V_k^n \stackrel{\text{def}}{=} \quad & \{f: \textit{ all polynomials of degree } \leq k \\ & \textit{on the interval } (2^{-n}l, 2^{-n}(l+1)) \\ & \textit{for } 0 \leq l < 2^n, \ f \textit{ vanishes elsewhere}\} \end{aligned} \tag{2.8}$$

This definition fulfills the conditions for a multiresolution analysis, and if the basis is chosen to be orthogonal, the V_k^n constitutes an *orthogonal* MRA.

2.2.1 The scaling basis

The construction of the scaling functions is quite straightforward; $k+1$ orthogonal polynomials are chosen to span the space of polynomials of degree $\leq k$ on the unit interval. The total scaling basis for V_k^n is then obtained by appropriate dilation and translation of these functions. One way to construct the basis is to start with the standard basis $\{1, x, x^2, \ldots, x^k\}$ and orthonormalize with respect to the L^2 inner product on the unit interval.

2.2.2 The wavelet basis

The *wavelet space* W_k^n is defined, according to Eq. (2.4), as the orthogonal complement of V_k^n in V_k^{n+1}. The wavelet basis functions of W_k^n are hence piecewise polynomials of degree $\leq k$ on *each* of the two intervals on scale $n+1$ that overlaps with *one* interval on scale n (but may be discontinous in the merging point). In the construction of the wavelet basis these piecewise polynomials should be made orthogonal both to the scaling basis of V_k^n and to each other.

One important property of the wavelet basis is its number of vanishing moments. The m-th continuous moment of a function ψ is defined as the integral

$$\mu_m \stackrel{\text{def}}{=} \int_0^1 x^m \psi(x) dx \tag{2.9}$$

and the function ψ is said to have M vanishing moments if

$$\mu_m = 0, \qquad m = 0, \ldots, M-1 \tag{2.10}$$

The vanishing moments of the wavelet functions gives information on the approximation order of the scaling functions. If the wavelet function ψ has M vanishing moments, any polynomial of order $\leq M-1$ can be exactly reproduced in the scaling space, and the error in representing an arbitrary function in the scaling basis is of M-th order. By construction, x^m is in the space V_k^0 for $0 \leq m \leq k$, and since $W_k^n \perp V_k^0$ for all $n >= 0$, the first $k+1$ moments of ψ_j^n must vanish.

2.2.3 Filter relations

With the multiwavelet basis defined, we can construct the filter matrices that fulfill the two-scale relations in Eq.(2.2) and Eq.(2.5). The exact construction will depend on the choice of scaling and wavelet polynomials, and will not be treated here, but some important properties of the filter matrices are already apparent from the definition of the scaling spaces given in Eq. (2.8).

Because of the disjoint support of the basis polynomials it is clear that a basis vector at scale n will overlap with two basis vectors at scale $n+1$, and we end up with four matrices $H^{(0)}, H^{(1)}, G^{(0)}$ and $G^{(1)}$, each of size $(k+1)\times(k+1)$. Eq. (2.2) and Eq. (2.5) thus reduces to

$$\begin{pmatrix} \boldsymbol{\psi}_l^n \\ \boldsymbol{\phi}_l^n \end{pmatrix} = \begin{pmatrix} G^{(1)} & G^{(0)} \\ H^{(1)} & H^{(0)} \end{pmatrix} \begin{pmatrix} \boldsymbol{\phi}_{2l+1}^{n+1} \\ \boldsymbol{\phi}_{2l}^{n+1} \end{pmatrix} \tag{2.11}$$

The locality of this transformation is important for numerical implementations, as it leads to efficient, linear scaling algorithms. The transformation in Eq. (2.11) is called forward wavelet transform or wavelet decomposition, while its inverse is called backward wavelet transform or wavelet reconstruction.

2.2.4 Multiwavelets in d dimensions

Multi-dimensional wavelets are usually constructed by tensor products, where the scaling space is defined as

$$V_k^{n,d} \stackrel{\text{def}}{=} \bigotimes^d V_k^n \tag{2.12}$$

The basis for this d-dimensional space is given as tensor products of the one-dimensional bases

$$\Phi_{\boldsymbol{j},\boldsymbol{l}}^n(\boldsymbol{x}) = \Phi_{j_1 j_2 \ldots j_d, l_1 l_2 \ldots l_d}^n(x_1, x_2, \ldots, x_d) \stackrel{\text{def}}{=} \prod_{p=1}^d \phi_{j_p,l_p}^n(x_p) \tag{2.13}$$

The number of basis functions on each hypercube $\boldsymbol{l} = (l_1, l_2, \ldots, l_d)$ becomes $(k+1)^d$, while the number of such hypercubes on scale n becomes 2^{dn}, which means that the total number of basis functions is growing exponentially with the number of dimensions.

The wavelet space can be defined using Eq. (2.4)

$$V_k^{n+1,d} = \bigotimes^d V_k^{n+1} = \bigotimes^d (V_k^n \oplus W_k^n) \tag{2.14}$$

where the pure scaling term obtained when expanding the product on the right hand side of Eq. (2.14) is recognized as $V_k^{n,d}$, making the wavelet space $W_k^{n,d}$ consist of all the remaining terms of the product, which are terms that contain at least one wavelet space.

To achieve a uniform notation, we can introduce a "generalized" one-dimensional wavelet function $\{\varphi_{j,l}^{\alpha,n}\}$ that, depending on the index α can be either the scaling or the wavelet function

$$\varphi_{j_p,l_p}^{\alpha_p,n} \stackrel{\text{def}}{=} \begin{cases} \phi_{j_p,l_p}^n & \text{if } \alpha_p = 0 \\ \psi_{j_p,l_p}^n & \text{if } \alpha_p = 1 \end{cases} \tag{2.15}$$

The wavelet functions for the d-dimensional space can thus be expressed as

$$\Psi_{\boldsymbol{j},\boldsymbol{l}}^{\alpha,n}(\boldsymbol{x}) = \prod_{p=1}^d \varphi_{j_p,l_p}^{\alpha_p,n}(x_p) \tag{2.16}$$

Where the total α index on Ψ separates the 2^d different possibilities of combining scaling/wavelet functions with the same index combination $\boldsymbol{j} = (j_0, j_1, \ldots, j_k)$.

α is given by the binary expansion $(\alpha_d \cdots \alpha_1 \alpha_0)$ and thus runs from 0 to $2^d - 1$. By closer inspection we see that $\alpha = 0$ recovers the pure scaling function

$$\Psi^{0,n}_{\boldsymbol{j},\boldsymbol{l}}(\boldsymbol{x}) \equiv \Phi^n_{\boldsymbol{j},\boldsymbol{l}}(\boldsymbol{x}) \tag{2.17}$$

and we will keep the notation $\Phi^n_{\boldsymbol{j},\boldsymbol{l}}$ for the scaling function, and exclude the $\alpha = 0$ term in the wavelet notation when treating multi-dimensional functions.

We can immediately see that the dimensionality of the wavelet space is higher than the scaling space on the same scale n, specifically $2^d - 1$ times higher. This must be the case in order to conserve the dimensionality through the equation

$$V_k^{n+1,d} = V_k^{n,d} \oplus W_k^{n,d} \tag{2.18}$$

since $dim(V_k^{n+1,d}) = 2^d dim(V_k^{n,d})$.

As for the mono-dimensional case we can define filter matrices that transform the scaling functions at scale $n+1$, $\{\Phi^{n+1}_{\boldsymbol{j},\boldsymbol{l}}\}$, into scaling and wavelet functions at scale n, $\{\Psi^{\alpha,n}_{\boldsymbol{j},\boldsymbol{l}}\}_{\alpha=0}^{2^d-1}$. Details of this construction can be found in the supporting information of Frediani *et al.* [5], where the corresponding matrices are shown to be tensor products of the mono-dimensional matrices. This means that the multi-dimensional wavelet transform can be done by consecutive application of d mono-dimensional filters. A detailed discussion on multi-dimensional MRAs and wavelet transforms can be found in Tymczak *et al.* [6].

2.3 Function representation

In this section we will describe how to represent functions in the multiwavelet basis, as well as how to perform simple arithmetic operations.

2.3.1 Function projection

We introduce the projection operator P_k^n onto the basis $\{\phi^n_{j,l}\}$ that span the scaling space V_k^n

$$f(x) \approx P_k^n f(x) \stackrel{\text{def}}{=} f^n(x) = \sum_{l=0}^{2^n-1} \sum_{j=0}^{k} s^{n,f}_{j,l} \phi^n_{j,l}(x) \tag{2.19}$$

where the expansion coefficients $s_{j,l}^{n,f}$, the so-called *scaling* coefficients, are obtained by the projection integral

$$s_{j,l}^{n,f} \stackrel{\text{def}}{=} \int_0^1 f(x)\phi_{j,l}^n(x)\,\mathrm{d}x \tag{2.20}$$

The accuracy of this approximation is determined by the scale n at which the projection is performed, and the order k of the polynomial basis.

2.3.2 Multiresolution functions

We can also introduce the projection operator Q_k^n that projects onto the wavelet basis $\{\psi_{j,l}^n\}$ of the space W_k^n

$$Q_k^n f(x) \stackrel{\text{def}}{=} \mathrm{d}f^n(x) = \sum_{l=0}^{2^n-1}\sum_{j=0}^{k} w_{j,l}^{n,f}\psi_{j,l}^n(x) \tag{2.21}$$

where the *wavelet* coefficients are given as

$$w_{j,l}^{n,f} \stackrel{\text{def}}{=} \int_0^1 f(x)\psi_{j,l}^n(x)\,\mathrm{d}x \tag{2.22}$$

According to Eq. (2.4) we have the following relationship between the projection operators

$$P_k^{n+1} = P_k^n + Q_k^n \tag{2.23}$$

which means that the wavelet projection should not be regarded as an approximation of the function f, but rather the difference between two approximations

$$\mathrm{d}f^n = Q_k^n f = (P_k^{n+1} - P_k^n)f = f^{n+1} - f^n \tag{2.24}$$

This means that the wavelet projection $\mathrm{d}f^n$ can be used as a measure of the accuracy of the scaling projection f^n, provided that the projection sequence is converging, $\lim_{n\to\infty} f^n = f$, which will be the case for square integrable functions[1]. By recursive application of Eq. (2.24) a given approximation f^N can be expressed as the much coarser approximation f^0 with a number of wavelet corrections

$$f(x) \approx f^N(x) \tag{2.25}$$

$$= f^0(x) + \sum_{n=0}^{N-1} \mathrm{d}f^n(x) \tag{2.26}$$

These equivalent representations are the high-resolution and multi-resolution approximations, respectively, of the function f. The forward and backward wavelet transforms of Eq. (2.11) allow us to change between the representations of Eqs. (2.25) and (2.26).

In principle it is possible to perform wavelet reconstructions *beyond* the finest scale N in the function representation f^N. In this case the wavelet contributions $\boldsymbol{\psi}_l^n$ in the inverse of Eq. (2.11) are zero, and no additional information is given to the scaling representation. However, the size of the scaling basis is doubled when the scale is increased by one, and the effect of such a wavelet reconstruction is that we get an *oversampled* representation of the function. This upsampling, usually denoted by the operator $\uparrow (f^N)$, is often necessary in practical implementations, as it is usually convenient to relate different function representations at a *common* scale that might be beyond the finest scale of one of the individual representations.

We also have the downsampling operator $\downarrow (f^N)$ that reduces the size of the basis, which means that information is thrown away in the process. In particular, a downsampling correspond to a projection onto the next coarser scaling space, and we have $\downarrow (f^N) \equiv f^{N-1}$. Note that the upsampling and downsampling operators do not commute, as

$$\downarrow (\uparrow (f^N)) = f^N \tag{2.27}$$

$$\uparrow (\downarrow (f^N)) = \uparrow (f^{N-1}) \neq f^N \tag{2.28}$$

2.3.3 Multiresolution functions in d dimensions

The multi-dimensional function representation is obtained similarly to Eq. (2.19) by projection onto the multi-dimensional basis Eq. (2.13)

$$f(\boldsymbol{x}) \approx f^n(\boldsymbol{x}) = \sum_{\boldsymbol{l}} \sum_{\boldsymbol{j}} s_{\boldsymbol{j},\boldsymbol{l}}^{n,f} \Phi_{\boldsymbol{j},\boldsymbol{l}}^n(\boldsymbol{x}) \tag{2.29}$$

where the sums are over all possible translation vectors $\boldsymbol{l} = (l_1, \ldots, l_d)$ for $0 \leq l_p \leq 2^n - 1$, and all possible scaling function combinations $\boldsymbol{j} = (j_1, \ldots, j_d)$ for $0 \leq j_p \leq k$. The scaling coefficients are obtained by the multi-dimensional integral

$$s_{\boldsymbol{j},\boldsymbol{l}}^{n,f} \stackrel{\text{def}}{=} \int_{[0,1]^d} f(\boldsymbol{x}) \Phi_{\boldsymbol{j},\boldsymbol{l}}^n(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \tag{2.30}$$

The wavelet components are given as

$$\mathrm{d}f^n(\boldsymbol{x}) = \sum_{\boldsymbol{l}} \sum_{\boldsymbol{j}} \sum_{\alpha=1}^{2^d-1} w_{\boldsymbol{j},\boldsymbol{l}}^{\alpha,n,f} \Psi_{\boldsymbol{j},\boldsymbol{l}}^{\alpha,n}(\boldsymbol{x}) \tag{2.31}$$

where the $\boldsymbol{l}$ and $\boldsymbol{j}$ summations are the same as in Eq. (2.29), and the α sum is over all combinations of scaling/wavelet functions (excluding the pure scaling $\alpha = 0$). The expansion coefficients are obtained by the multi-dimensional projection

$$w_{\boldsymbol{j},\boldsymbol{l}}^{\alpha,n,f} \stackrel{\text{def}}{=} \int_{[0,1]^d} f(\boldsymbol{x}) \Psi_{\boldsymbol{j},\boldsymbol{l}}^{\alpha,n}(\boldsymbol{x}) d\boldsymbol{x} \tag{2.32}$$

We can again approximate the function $f(\boldsymbol{x})$ at scale N and decompose it into its multiresolution components

$$f(\boldsymbol{x}) \approx f^N(\boldsymbol{x}) = f^0(\boldsymbol{x}) + \sum_{n=0}^{N-1} \mathrm{d}f^n(\boldsymbol{x}) \tag{2.33}$$

2.3.4 Addition of functions

The addition of functions in the multiwavelet basis is quite straightforward, as it is represented by the mappings

$$\begin{aligned} V_k^n + V_k^n &\to V_k^n \\ W_k^n + W_k^n &\to W_k^n \end{aligned} \tag{2.34}$$

This basically means that the projection of the sum equals the sum of the projections. In the polynomial basis this is simply the fact that the sum of two k-order polynomials is still a k-order polynomial.

2.3.5 Multiplication of functions

Multiplication of functions in the multiwavelet basis is somewhat more involved than addition. The reason for this is that, in contrast to Eq. (2.34), the product is represented by the mapping

$$V_k^n \times V_k^n \to V_{2k}^n \tag{2.35}$$

This means that the product of two functions falls outside of the MRA and needs to be projected back onto the scaling space sequence. Following Beylkin [7] we

can say that the product of two functions on a given scale "spills over" into the finer scales

$$V_k^n \times V_k^n \to V_k^n \oplus \bigoplus_{n'=n}^{\infty} W_k^{n'} \tag{2.36}$$

Working with a finite precision it is desirable to make the product as accurate as each of the multiplicands. This is done by terminating the sum in Eq. (2.36) at some sufficiently large scale $N > n$

$$V_k^n \times V_k^n \to V_k^n \oplus \bigoplus_{n'=n}^{N-1} W_k^{n'} = V_k^N \tag{2.37}$$

As the finest scale N required in the product in general will be higher than the finest scale n in each of the multiplicands, it is convenient to perform the multiplication on oversampled representations of the multiplicands obtained by $N - n$ upsamplings.

2.4 Operator representation

In this section we discuss the multiresolution analysis of a general operator T

$$g(x) = [Tf](x) \tag{2.38}$$

and we describe two different multiresolution representation of the operator: the so-called standard and non-standard representations. The difference between the two is largely a matter of implementation, as they are mathematically equivalent, but as we will see below, the non-standard form leads to considerably simpler algorithms, especially in the multi-dimensional implementation. In the standard representation the operator couples all length scales in all dimensions, leading to a very complicated operator structure, while in the non-standard representation the different scales are decoupled in the operator application, while the interaction between scales are handled by a post-processing step.

An essential feature in the discussion of operators in the multiresolution framework is the number of vanishing moments of the chosen basis. This property leads to effectively sparse representations of certain operators (in the sense that sparse representations can be obtained to a given accuracy by *a priori* thresholding of small coefficients), and fast (linear-scaling) algorithms can be obtained for the operator application.

A necessary assumption for an efficient implementation of a multi-dimensional operator is that it is separable in the Cartesian coordinates. This, combined with the tensor structure of the multiwavelet basis, ensures that the multi-dimensional operator application can be performed using mono-dimensional algorithms, and that the exponential scaling in the dimension is significantly reduced. This assumption does not limit the applicability of the method on real-world problems, as many important non-separable operators in physics can be made separable to a finite, but arbitrary precision.

2.4.1 Operator projection

Working in the multiresolution analysis, the operator is applied to the projection of f at a given scaling space V_k^n

$$\hat{g}(x) = [TP_k^n f](x) \tag{2.39}$$

and we are looking for the projected solution

$$P_k^n \hat{g}(x) = [P_k^n T P_k^n f](x) \tag{2.40}$$

Using the fundamental property of projection operators $P_k^n P_k^n = P_k^n$ we get

$$P_k^n \hat{g}(x) = [P_k^n T P_k^n P_k^n f](x) \tag{2.41}$$

and we can represent the full operator application on scale n

$$\hat{g}^n(x) = {}^nT^n f^n(x) \tag{2.42}$$

where the projection of the operator T at the scaling space V_k^n is defined as

$${}^nT^n \stackrel{\text{def}}{=} P_k^n T P_k^n \tag{2.43}$$

This operation should be performed at a scale N where the overall accuracy of the representations are satisfactory, and we can assume that

$$\hat{g}^N \approx g^N \stackrel{\text{def}}{=} (Tf)^N \approx g \tag{2.44}$$

Algorithms for how to achieve this accuracy is presented in Chap. 3.

2.4.2 Multiresolution operators

Making use of Eqs. (2.43) and (2.23) we can decompose the scaling representation of the operator at scale $n+1$ into scaling and wavelet contributions at the next coarser scale

$$T \approx P_k^{n+1} T P_k^{n+1} \tag{2.45}$$

$$= \big(P_k^n + Q_k^n\big) T \big(P_k^n + Q_k^n\big) \tag{2.46}$$

$$= P_k^n T P_k^n + \; P_k^n T Q_k^n + \; Q_k^n T P_k^n + \; Q_k^n T Q_k^n \tag{2.47}$$

and we simplify the notation with the following definitions, including a generalization of the definition in Eq. (2.43)

$$\begin{aligned}
{}^n A^{n'} &\stackrel{\text{def}}{=} \quad Q_k^n T Q_k^{n'} : \quad W_k^{n'} \to W_k^n \\
{}^n B^{n'} &\stackrel{\text{def}}{=} \quad Q_k^n T P_k^{n'} : \quad V_k^{n'} \to W_k^n \\
{}^n C^{n'} &\stackrel{\text{def}}{=} \quad P_k^n T Q_k^{n'} : \quad W_k^{n'} \to V_k^n \\
{}^n T^{n'} &\stackrel{\text{def}}{=} \quad P_k^n T P_k^{n'} : \quad V_k^{n'} \to V_k^n
\end{aligned} \tag{2.48}$$

leading to the relation

$${}^{n+1}T^{n+1} = \; {}^n T^n \; + \; {}^n C^n \; + \; {}^n B^n \; + \; {}^n A^n \tag{2.49}$$

The motivation for such a decomposition of the operator lies in the vanishing moments of the basis. The A, B and C parts of the operator involves projections into the wavelet basis, which has the property of vanishing moments, and we will see later that this leads to sparse representations of certain operators.

The decomposition in Eq. (2.49) can be continued recursively, and by this introduce more sparsity into the operator, and there are two ways to proceed in order to achieve this. In the following both the standard and the non-standard form of the multiresolution operator will be presented.

2.4.3 Standard representation

The standard representation is the straightforward matrix realization of the operator in the multiresolution basis. In order to obtain this representation we

start with the matrix representation in the scaling basis at scale N

$$\begin{pmatrix} & & \\ & {}^{N}T^{N} & \\ & & \end{pmatrix} \begin{pmatrix} \\ f^{N} \\ \\ \end{pmatrix} = \begin{pmatrix} \\ g^{N} \\ \\ \end{pmatrix} \tag{2.50}$$

This matrix can be decomposed into four submatrices according to Eq. (2.49) while the functions are decomposed into scaling and wavelet contributions at scale $N-1$

$$f^{N} = f^{N-1} + \mathrm{d}f^{N-1} \tag{2.51}$$

$$g^{N} = g^{N-1} + \mathrm{d}g^{N-1} \tag{2.52}$$

According to Eq. (2.48) ${}^{n}T^{n}$ and ${}^{n}C^{n}$ produce the scaling part of g, acting on the scaling and wavelet parts of f, respectively. Similarly, ${}^{n}A^{n}$ and ${}^{n}B^{n}$ produce the wavelet part of g, by acting on the wavelet and scaling parts of f,

respectively. The matrix equation Eq. (2.50) can thus be decomposed as

$$\left(\begin{array}{c|c} {}^{N-1}T^{N-1} & {}^{N-1}C^{N-1} \\ \hline {}^{N-1}B^{N-1} & {}^{N-1}A^{N-1} \end{array}\right)\left(\begin{array}{c} f^{N-1} \\ \hline \mathrm{d}f^{N-1} \end{array}\right) = \left(\begin{array}{c} g^{N-1} \\ \hline \mathrm{d}g^{N-1} \end{array}\right) \tag{2.53}$$

where the size of the total matrix is unchanged. We can now do the same decomposition of ${}^{N-1}T^{N-1}$ into submatrices at scale $N-2$. The function components f^{N-1} and g^{N-1} need to be decomposed as well, so to keep everything consistent, the ${}^{N-1}B^{N-1}$ and ${}^{N-1}C^{N-1}$ parts of the operator will have to be transformed accoringly. To proceed from here we need the following relations

$$\begin{aligned} {}^{n}B^{n} &= Q_k^n T P_k^n \\ &= Q_k^n T (P_k^{n-1} + Q_k^{n-1}) \\ &= Q_k^n T P_k^{n-1} + Q_k^n T Q_k^{n-1} \\ &= {}^{n}B^{n-1} + {}^{n}A^{n-1} \end{aligned} \tag{2.54}$$

and similarly for the C block

$${}^{n}C^{n} = {}^{n-1}C^{n} + {}^{n-1}A^{n} \tag{2.55}$$

which is the change in the operator that is taking place when we decompose f^n into $f^{n-1} + \mathrm{d}f^{n-1}$ and g^n into $g^{n-1} + \mathrm{d}g^{n-1}$. The matrix equation now turns

into

$$\left(\begin{array}{c|c|c} {}^{N-2}T^{N-2} & {}^{N-2}C^{N-2} & {}^{N-2}C^{N-1} \\ \hline {}^{N-2}B^{N-2} & {}^{N-2}A^{N-2} & {}^{N-2}A^{N-1} \\ \hline {}^{N-1}B^{N-2} & {}^{N-1}A^{N-2} & {}^{N-1}A^{N-1} \end{array}\right) \left(\begin{array}{c} f^{N-2} \\ \hline \mathrm{d}f^{N-2} \\ \hline \mathrm{d}f^{N-1} \end{array}\right) = \left(\begin{array}{c} g^{N-2} \\ \hline \mathrm{d}g^{N-2} \\ \hline \mathrm{d}g^{N-1} \end{array}\right) \tag{2.56}$$

and we can continue this transformation recursively until we reach the coarsest scale.

Symbolically, we can do the decomposition of Eq. (2.49) by recursive application of itself as well as Eqs. (2.54) and (2.55), where we gradually introduce more A-character into the operator

$$\begin{aligned} {}^{N}T^{N} &= {}^{0}T^{0} + \sum_{n=0}^{N-1} {}^{n}C^{n} + \sum_{n=0}^{N-1} {}^{n}B^{n} + \sum_{n=0}^{N-1} {}^{n}A^{n} \\ &= {}^{0}T^{0} + \sum_{n=0}^{N-1} \Big({}^{0}C^{n} + \sum_{n'<n} {}^{n'}A^{n} \Big) + \\ &\qquad \sum_{n=0}^{N-1} \Big({}^{n}B^{0} + \sum_{n'>n} {}^{n'}A^{n} \Big) + \sum_{n=0}^{N-1} {}^{n}A^{n} \\ &= {}^{0}T^{0} + \sum_{n=0}^{N-1} \Big({}^{0}C^{n} + {}^{n}B^{0} + \sum_{n'=0}^{N-1} {}^{n}A^{n'} \Big) \end{aligned} \tag{2.57}$$

This multiresolution matrix representation of the operator is called the standard representation.

2.4.4 Non-Standard representation

While the standard form of the operator given in Eq. (2.57) does lead to sparse representations, it gives rise to rather complicated algorithms, especially in sev-

eral dimensions, as it couples all scales in the problem. Beylkin *et al.* [8] introduced a different approach, which they called the non-standard representation, where the scales are explicitly separated, by organizing the operator as a collection of triples

$$ {}^{N}T^{N} = {}^{0}T^{0} + \sum_{n=0}^{N-1} \left({}^{n}A^{n} + {}^{n}B^{n} + {}^{n}C^{n} \right) \tag{2.58} $$

where each triple (${}^{n}A^{n}$, ${}^{n}B^{n}$, ${}^{n}C^{n}$) corresponds to the interaction at a particular scale n. The interaction *between* different length scales are not explicitly treated in this representation, and needs to be accounted for in a post-prosessing step. In order to achieve this separation of scales some redundancy is necessary in the function representations for f and g, as we need to keep the scaling projections at *all* scales. The operator matrix that is applied to the function will in this case be

$$ \left(\begin{array}{cc|cc} {}^{N-1}T^{N-1} & {}^{N-1}C^{N-1} & & \\ {}^{N-1}B^{N-1} & {}^{N-1}A^{N-1} & & \\ \hline & & & {}^{N-1}C^{N-1} \\ & & {}^{N-1}B^{N-1} & {}^{N-1}A^{N-1} \end{array} \right) \left(\begin{array}{c} f^{N-2} \\ \mathrm{d}f^{N-2} \\ \hline f^{N-1} \\ \mathrm{d}f^{N-1} \end{array} \right) \tag{2.59} $$

and although the total matrix has grown in size, this representation leads to straightforward adaptive algorithms, as the operator can be applied one scale at the time, starting from the coarsest (usually $n = 0$). As pointed out above, this does not directly account for the interaction between scales, but this can be included by a series of wavelet transforms on parts of the result. This is described fully in the implementation part in Chap. 3. The post-processing wavelet transforms require $O(N)$ operations, and provided sparse A, B and C parts of the operator, the complete non-standard application scales as $O(N)$, in contrast to the standard form, where scale-to-scale interactions are treated explicitly, which has a formal $O(NlogN)$ scaling [8].

2.4.5 Integral operator

Multiwavelets were originally introduced for their effectively sparse representation of certain integral operators, in particular operators with non-oscillatory kernels that are analytic except along a finite set of curves [1]. To be more specific, we consider one-dimensional operators on the form

$$[Tf](x) = \int K(x,y)f(y)\,\mathrm{d}y \tag{2.60}$$

The sparsity of the operator representation follows under certain conditions on the integral kernel K, which is discussed below. We start, however, by expanding the kernel in the multiwavelet basis

$$K^n(x,y) = \sum_{l,m}\sum_{i,j}\left[\tau^n_{lm}\right]_{ij}\phi^n_{i,l}(x)\phi^n_{j,m}(y) \tag{2.61}$$

where the expansion coefficients are given by the integrals

$$\left[\tau^n_{lm}\right]_{ij} = \int\int K(x,y)\phi^n_{i,l}(x)\phi^n_{j,m}(y)\,\mathrm{d}x\,\mathrm{d}y \tag{2.62}$$

Inserting Eq. (2.61) into Eq. (2.60) yields

$$^nT^n f^n(x) = \int\left(\sum_{l,m}\sum_{i,j}\left[\tau^n_{lm}\right]_{ij}\phi^n_{i,l}(x)\phi^n_{j,m}(y)\right)f(y)\,\mathrm{d}y \tag{2.63}$$

$$= \sum_{l,m}\sum_{i,j}\left[\tau^n_{lm}\right]_{ij}\phi^n_{i,l}(x)\int f(y)\phi^n_{j,m}(y)\,\mathrm{d}y \tag{2.64}$$

where the last integral is recognized as the vector of scaling coefficients of f from Eq. (2.20)

$$^nT^n f^n(x) = \sum_{l,m}\sum_{i,j}\left[\tau^n_{lm}\right]_{ij}\phi^n_{i,l}(x)s^{n,f}_{j,m} \tag{2.65}$$

We can now identify $\boldsymbol{\tau}^n_{lm}$ as the matrix elements of ${}^nT^n$ and Eq. (2.65) is Eq. (2.50) written explicitly. Similarly, we define $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ as the matrix elements of A, B and C, respectively

$$\left[\alpha^n_{lm}\right]_{ij} = \int\int K(x,y)\psi^n_{i,l}(x)\psi^n_{j,m}(y)\,\mathrm{d}x\,\mathrm{d}y \tag{2.66}$$

$$\left[\beta^n_{lm}\right]_{ij} = \int\int K(x,y)\psi^n_{i,l}(x)\phi^n_{j,m}(y)\,\mathrm{d}x\,\mathrm{d}y \tag{2.67}$$

$$\left[\gamma^n_{lm}\right]_{ij} = \int\int K(x,y)\phi^n_{i,l}(x)\psi^n_{j,m}(y)\,\mathrm{d}x\,\mathrm{d}y \tag{2.68}$$

which act on the function representations of f in the following way

$${}^nA^n\,\mathrm{d}f^n(x) = \sum_{l,m}\sum_{i,j}\left[\alpha^n_{lm}\right]_{ij}\psi^n_{i,l}(x)w^{n,f}_{j,m} \tag{2.69}$$

$${}^nB^n f^n(x) = \sum_{l,m}\sum_{i,j}\left[\beta^n_{lm}\right]_{ij}\psi^n_{i,l}(x)s^{n,f}_{j,m} \tag{2.70}$$

$${}^nC^n\,\mathrm{d}f^n(x) = \sum_{l,m}\sum_{i,j}\left[\gamma^n_{lm}\right]_{ij}\phi^n_{i,l}(x)w^{n,f}_{j,m} \tag{2.71}$$

As was mentioned above, the motivation for decomposing the operator into A, B and C terms is that these matrices will be sparse for certain operators. Suppose that the integral kernel in Eq. (2.60) satisfy the estimates

$$|K(x,y)| \leq \frac{1}{|x-y|} \tag{2.72}$$

$$|\partial^M_x K(x,y)| + |\partial^M_y K(x,y)| \leq \frac{C_M}{|x-y|^{M+1}} \tag{2.73}$$

for some $M \geq 1$. Such operators are called Calderon-Zygmund operators, and include both the Poisson and bound-state Helmholtz operators which are discussed in detail in Chap. 3. Beylkin *et al.* [8] shows that in a basis with M vanishing moments, the wavelet components $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ will be bounded as

$$\|\boldsymbol{\alpha}_{lm}\|_2 + \|\boldsymbol{\beta}_{lm}\|_2 + \|\boldsymbol{\gamma}_{lm}\|_2 \leq \frac{C_M}{1+|l-m|^{M+1}} \tag{2.74}$$

where the expression has been adapted to a multiwavelet setting using the matrix 2-norm. This means that within a given accuracy, all contributions beyond a certain spatial separation $|l-m|$ can be set to zero, leading to operators that are banded along the diagonal.

2.4.6 Derivative operator

Alpert *et al.* [9] described how to construct derivative operators in the multiwavelet basis. Since the basis is discontinuous, there does not exist a unique

representation of the derivative operator. This non-uniqueness appears as two adjustable parameters that handles boundary conditions at the discontinuous merging point between basis functions. The representation can be viewed as the straightforward differentiation of the basis functions at the *interior* of each interval, combined with a finite difference representation *across* intervals.

The matrix representation of the operator $T = d/dx$ is formally given as

$$\left[\tau^n_{lm}\right]_{ij} = \int_{2^{-n}l}^{2^{-n}(l+1)} \phi^n_{i,l}(x) T \phi^n_{j,m}(x)\,\mathrm{d}x \tag{2.75}$$

$$= 2^n \int_0^1 \phi_i(x) T \phi_j(x-(l-m))\,\mathrm{d}x \tag{2.76}$$

However, for derivative operators, this integral is not absolutely convergent. Because of the disjoint support of the basis functions, it is immidiately clear that there will be no interaction beyond the neighboring interval, and $\boldsymbol{\tau}_{lm} = 0$ for $|l-m| > 1$. The case $|l-m| = 1$ needs to be treated with care, since there are boundary effects to consider even if the basis functions are non-overlapping. This becomes apparent if we look at the scaling coefficients of the derivative f'^n of a function f^n represented in the scaling basis at scale n

$$s^{n,f'}_{i,l} = \int_{2^{-n}l}^{2^{-n}(l+1)} \phi^n_{i,l}(x) \frac{\mathrm{d}}{\mathrm{d}x} f^n(x)\,\mathrm{d}x \tag{2.77}$$

Integration by parts now introduces a boundary term

$$s^{n,f'}_{i,l} = \phi^n_{i,l}(x) f^n(x)\Big|_{2^{-n}l}^{2^{-n}(l+1)} - \int_{2^{-n}l}^{2^{-n}(l+1)} f^n(x) \frac{\mathrm{d}}{\mathrm{d}x} \phi^n_{i,l}(x)\,\mathrm{d}x \tag{2.78}$$

$$= 2^{n/2}\Big[f^n(2^{-n}(l+1))\phi_i(1) - f^n(2^{-n}l)\phi_i(0)\Big] - 2^n \sum_{j=0}^{k} K_{ij} s^{n,f}_{j,l} \tag{2.79}$$

where the matrix K is defined

$$K_{ij} = \int_0^1 \phi_j(x) \frac{\mathrm{d}}{\mathrm{d}x} \phi_i(x)\,\mathrm{d}x \tag{2.80}$$

We see in Eq. (2.79) that the function representation f^n needs to be evaluated precisely at the discontinuities of the basis where the function value is not well defined. This problem is circumvented by interpolating between the function values obtained at both sides of the boundary

$$f^n = a f^n_- + b f^n_+ \tag{2.81}$$

where a and b are adjustable parameters. In the Haar basis (piecewise constants) this reduces to a finite difference definition of the derivative, with the choice $a = b = 1/2$ corresponding to central difference, and $a = 1, b = 0$ and $a = 0, b = 1$ corresponding to forward and backward differences, respectively. With the choice $a = b = 0$ no boundary effects are treated, and the derivative is obtained by a straightforward piecewise derivative of the polynomial basis.

2.4.7 Multiresolution operators in d dimensions

We assume that we have a separable representation of a d-dimensional operator $\mathcal{T}$ such that

$$\mathcal{T} = \bigotimes_{p=1}^{d} T_p \tag{2.82}$$

where T_p correspond to a one-dimensional operator as described above. As for the one-dimensional case we have the equation

$$g^{n+1} = \bigotimes^{d} \; {}^{n+1}T^{n+1} f^{n+1} \tag{2.83}$$

which we can decompose to

$$g^n + \mathrm{d}g^n = \bigotimes^{d} \left({}^{n}A^n + {}^{n}B^n + {}^{n}C^n + {}^{n}T^n \right) \left(f^n + \mathrm{d}f^n \right) \tag{2.84}$$

and we can simplify the notation in the following way

$$\begin{aligned} {}^{n}A^n &= O^{11,n} & {}^{n}B^n &= O^{10,n} \\ {}^{n}C^n &= O^{01,n} & {}^{n}T^n &= O^{00,n} \end{aligned} \tag{2.85}$$

and the tensor product of the operator can be written

$$\bigotimes^{d} \left({}^{n}A^n + {}^{n}B^n + {}^{n}C^n + {}^{n}T^n \right) = \sum_{\alpha=0}^{2^d-1} \sum_{\beta=0}^{2^d-1} O^{\alpha,\beta,n} \tag{2.86}$$

where we define

$$O^{\alpha\beta,n} \stackrel{\text{def}}{=} \bigotimes_{p}^{d} O^{\alpha_p \beta_p, n} \tag{2.87}$$

with $0 \leq \alpha < 2^d$ and $0 \leq \beta < 2^d$ and α_p and β_p are defined by the binary expansion of α and β in d dimensions. We can now obtain a completely equivalent structure as for the mono-dimensional case

$$g^n + \mathrm{d}g^n = \left(\mathcal{A}^n + \mathcal{B}^n + \mathcal{C}^n + \mathcal{T}^n \right) \left(f^n + \mathrm{d}f^n \right) \tag{2.88}$$

with the following definitions

$$\begin{aligned} \mathcal{A}^n &\stackrel{\text{def}}{=} \sum_{\alpha=1}^{2^d-1} \sum_{\beta=1}^{2^d-1} O^{\alpha\beta,n} & \mathcal{B}^n &\stackrel{\text{def}}{=} \sum_{\alpha=1}^{2^d-1} O^{\alpha 0,n} \\ \mathcal{C}^n &\stackrel{\text{def}}{=} \sum_{\beta=1}^{2^d-1} O^{0\beta,n} & \mathcal{T}^n &\stackrel{\text{def}}{=} O^{00,n} \end{aligned} \tag{2.89}$$

We could now proceed with a further decomposition of the scaling parts of the operator and functions to the next coarser scale, obtaining the standard representation of the operator in multiple dimension. It is quite clear that the notation (as well as implementation) becomes very complicated in this case, and this is one of the main motivations for using the non-standard representation of operators, as the scales are decoupled, and Eq. (2.88) applies to each scale separately.

Chapter 3

Implementation

The multiresolution formalism presented in Chap. 2 gives prospects of efficient (sparse) representations of functions and operators, and in this chapter we describe how this is achieved in practice. By local thresholding of small wavelet coefficients, functions can be represented on adaptive, multiresolution grids, where each grid is specifically constructed to the function it holds. For operators, we use the concept of separation of variables [10, 11] in order to reduce the complexity of application i three dimensions, together with *a priori* thresholding of long-range wavelet terms according to the estimates of Eq. (2.74).

In the following we describe the important data structures and algorithms that are used in the MultiResolution Computational Program Package (MRCPP). The code is written in C++, utilizing the concepts of object-orientation and generic programming, where for instance the dimension appears as a template parameter, which means that the code is immediately applicable to any dimension, although some algorithms are specialized and optimized for $d = 3$.

Due to the inherent high demands on memory and computational resources that comes with all real-space numerical methods, the code relies heavily upon parallel algorithms and data distribution. In the current code data distribution is handled by the Message-Passing Interface (MPI), and further work load distribution is provided by an additional shared memory (OpenMP) parallelization on top. The parallel implementation and the performance of the code is discussed fully in publication II.

3.1 Data structures

3.1.1 `Node`

The `node` is the multidimensional box on which the set of scaling and wavelet functions that share the same support are defined. The `node` is specified by its scale n, which gives its size ($[0, 2^{-n}]^d$) and translation vector $\boldsymbol{l} = (l_1, l_2, \ldots, l_d)$, which gives its position. The `node` holds the $(k+1)^d$ scaling coefficients and $(2^d - 1)(k+1)^d$ wavelet coefficients that share the same scale and translation. It will also keep track of its parent and all 2^d children `nodes`, giving the `nodes` a tree-like structure.

3.1.2 `Tree`

The `tree` data structure is a collection of `nodes` that makes up a function. In order to minimize the memory requirements, all variables that are common to all `nodes` (like polynomial order, number of coefficients, type of scaling functions, etc) are stored in the `tree` structure. The `tree` keeps the entire set of `nodes`, from root to leaf, and each `node` keeps both the scaling and wavelet coefficients. This means that there is a redundancy in the function representation as the multiresolution representation in Eq. (2.26) requires scaling coefficients at the coarsest scale only. However, it proves more efficient to keep all scaling coefficients in memory rather than obtaining them by the filter operations of Eq. (2.11), as they are needed e.g. in the non-standard operator application.

3.1.3 Parallel data distribution

As the data storage requirements of real-space methods quickly exceeds the available memory on a single computational device, it eventually becomes necessary to distribute the data that is contained in the full `tree` representation of a function among the memory of several computers (hosts). In the multiwavelet basis the function representations are conveniently partitioned into equally sized portions (equal in terms of memory, not spatial extension), and data distribution is achieved by dividing these `nodes` among the available hosts.

There are several possible strategies for how the `nodes` could be distributed and we have chosen one that leads to strictly connected domains, in the sense

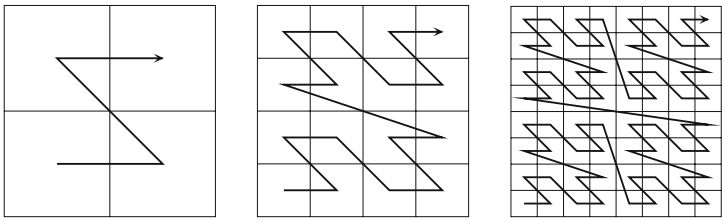

Figure 1: Three refinement levels in the construction of the Lebesgue curve in 2D.

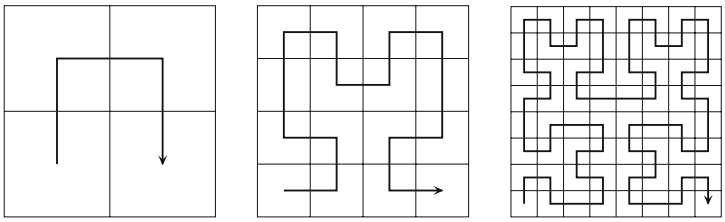

Figure 2: Three refinement levels in the construction of the Hilbert curve in 2D.

that all `nodes` belonging to a given host is connected (share a common vertice, not necessarily on the same scale) to at least one other `node` owned by the same host. This ensures that the real-space domain of a given host will be localized in space, with the motivation that the interaction between hosts could be limited to involve only near neighbors, and thus hopefully reduce the need for communication between hosts.

In order to achieve this localization we traverse the `tree` following a space-filling path, assigning `nodes` to hosts as we go. By following a so-called Hilbert path [12], we obtain a continuous curve with good locality properties, that can be partitioned among the hosts. The construction of the curve is done recursively, going through the 2^d children of each `node` in a specific order. Using bit notation (one bit for each dimension), the natural ordering (Lebesgue) will lead to a discontinuous path. For $d = 2$ this is shown as the Z shape of the bit sequence (00, 01, 10, 11) in Fig. 1. A corresponding (there are several possibilities) Hilbert path through the four children in two dimensions could be the bit sequence (00, 10, 11, 01) shown in the first panel in Fig. 2. In order to keep the continuity as the path is recursively refined, the order in which the children are traversed needs to be adapted, and will depend on the position of the parent among its siblings, as shown in Fig. 2.

3.2 Adaptive algorithm

Algorithm 1 Generation of adaptive multiwavelet representation of a function

```
create tree skeleton of empty nodes
MPI: distribute leaf nodes among hosts through Hilbert path
MPI: create list of local nodes owned by this host
while number of local nodes on current iteration N_i > 0 do
    OpenMP: divide local nodes among available processors
    for each node at current iteration do
        compute scaling and wavelet coefficients
        if node needs to be refined then
            mark node as non-terminal
            allocate children nodes
            update list of local nodes for next iteration
        else
            mark node as terminal
        end if
    end for
    increment iteration
end while
```

Alg. 1 used to obtain adaptive representations of functions was originally presented in [5], but is here extended to include parallelization. The first lines in this algorithm are very important in order to ensure a good load balancing among MPI hosts. By utilizing some *a priori* knowledge of the function that is about to be buildt, we try to estimate the final `tree` structure as closely as possible before calculating any coefficients. In this way we have a lot more flexibility when it comes to parallel distribution of data and work load in all iterations. Without this preprocessing step, the first three iterations would contain one, eight and 64 `nodes`, respectively, allowing little freedom in parallel computations. It is important in this step to capture the global structure of the function (where in space is high level of refinement needed), as this initial `tree` skeleton is used in the data distribution among MPI hosts and all subsequent additional refinent is done locally on each host (although some load balancing can be preformed by redistribution of data if needed). How to construct this

skeleton depends on the function, and will be discussed in the following sections.

The algorithm consists of two loops, the first iteration will add levels of refinement on top of the initial skeleton wherever necessary in order to guarantee the overall accuracy of the representation. This loop terminates when no further refinement is needed. The second loop runs over the `nodes` present at the current iteration (only local `nodes` that belong to the given MPI host), and these are distributed among the available processors (OpenMP) at the given host. Once the scaling/wavelet coefficients of a given `node` are known, a split check is performed based on the desired precision. If the `node` does not satisfy the accuracy criterion, it is marked as non-terminal and its children `nodes` are allocated and added to the list of `nodes` needed in the next iteration. If the `node` does not need to be split, it is marked as terminal and no children `nodes` are allocated. In this way, once the loop over `nodes` on one iteration is terminated, the complete list of `nodes` needed in the next iteration has been obtained. The `tree` is grown until no `nodes` are needed at the next iteration.

There are two points in the algorithm that need to be elaborated further, the first being the actual computation of the coefficients (line 7). This can be done in many ways, e.g. projection or by operator application, and will be treated in the subsequent sections.

The second point is how to perform the split check (line 8), which is used to decide whether or not the function is represented accurately enough on the current `node`, based on a predefined relative precision ϵ. Formally, this relative precision requires that

$$\|f - f^n\| < \epsilon \|f\| \tag{3.1}$$

However, this check cannot be performed since the *true* function f is generally not known. Instead we will use the norm of the wavelet projections as a measure of the accuracy of the representation. Specifically, the norm of the wavelet coefficients on one `node` is used as a measure for the accuracy of the part of the function represented by this `node`, and we require that

$$\|\boldsymbol{w}_{\boldsymbol{l}}^n\| < \frac{\epsilon}{2^{n/2}} \|f^n\| \tag{3.2}$$

The local, disjoint support of the wavelet basis ensures that the global error of the representation can be controlled by locally truncating the wavelet expan-

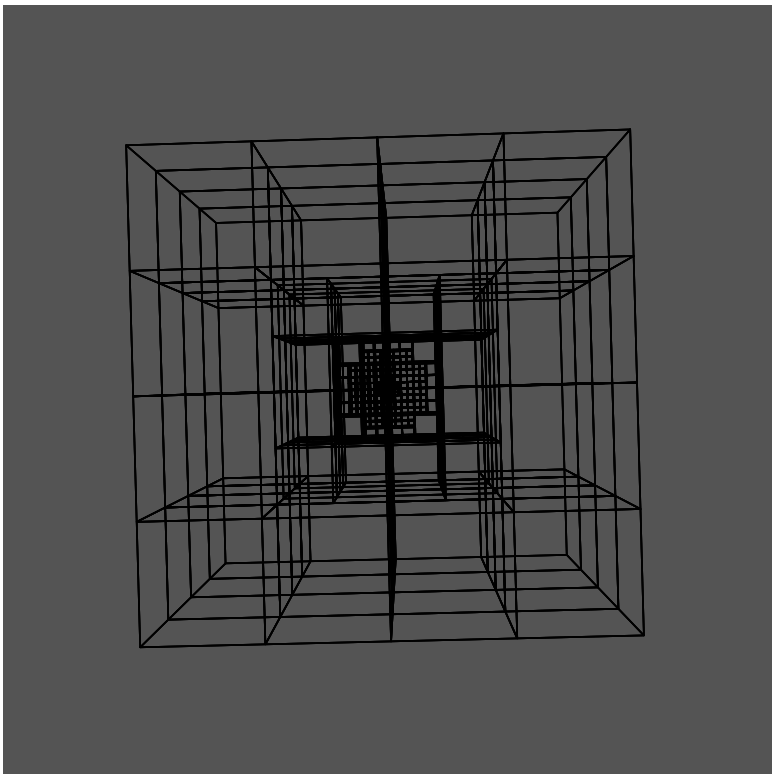

Figure 3: Adaptive grid-partitioning of the unit cube needed to reproduce the Gaussian function $f(\boldsymbol{x}) = (\beta/\pi)^{3/2} e^{-\beta(\boldsymbol{x}-\boldsymbol{x}_0)^2}$ with exponent $\beta = 500$ in position $\boldsymbol{x}_0 = (1/2, 1/2, 1/2)$ to a relative accuracy of $\epsilon = 10^{-8}$ using multiwavelets of order $k = 9$.

sion, allowing a fully on-the-fly adaptive algorithm. This reduces the number of expansion coefficients needed to represent the function to the given accuracy dramatically compared to the uniform high-resolution representation in Eq. (2.25). In practical calculations one can easily get significant contribution over a range of ten length scales, and a uniform grid in three dimensions at scale $n = 10$ would require $(2^d)^n = 8^{10} \sim 10^9$ `nodes`, while a typical multi-resolution representation requires in the order of $10^2 - 10^4$ `nodes` per scale. Fig. 3 shows an adaptive grid used for representing a spherical Gaussian positioned at the center of the unit cube.

The presented algorithm is very general, and is used to build adaptive representations of functions regardless of how the expansion coefficients are obtained, and later in the chapter we will look at different ways of doing this.

3.3 Choice of basis functions

Before we can describe how to calculate expansion coefficients we need to specify the type of scaling and wavelet functions that is used in the multiresolution analysis. In principal, any polynomial basis that span the appropriate scaling V_k^n and wavelet W_k^n spaces can be used, and in the original construction Alpert [1] used Legendre polynomials as scaling basis, but in a later work, Alpert *et al.* [9] introduced an alternative basis with interpolating properties. Both scaling bases have been implemented, but in practice only the latter is used, because of its superior numerical efficiency. The choice of wavelet basis follows that of Alpert [1].

3.3.1 Legendre scaling functions

The Legendre polynomials $\{L_j(x)\}_{j\in\mathbb{N}}$ are a family of functions, defined on the interval $[-1,1]$. The functions are orthogonal with respect to the $L^2([-1,1])$ inner product

$$\int_{-1}^{1} L_i(x)L_j(x)\,\mathrm{d}x = 0, \qquad i \neq j \tag{3.3}$$

but they are usually normalized such that $L_j(1) = 1$. The polynomials can be constructed by induction

$$L_0(x) = 1 \tag{3.4}$$

$$L_1(x) = x \tag{3.5}$$

$$L_{j+1}(x) = \frac{2j+1}{j+1}xL_j(x) - \frac{j}{j+1}L_{j-1} \tag{3.6}$$

and the *Legendre scaling functions* ϕ_j^L are obtained by dilation and translation to the unit interval, followed by L^2 normalization

$$\phi_j^L(x) = \sqrt{2j+1}L_j(2x-1), \qquad x \in [0,1] \tag{3.7}$$

This is the original construction of scaling functions by Alpert [1].

3.3.2 Interpolating scaling functions

Alpert *et al.* [9] presented an alternative set of scaling functions with interpolating properties. These *Interpolating scaling functions* ϕ_j^I are based on the

Legendre scaling functions $\{\phi_j^L\}_{j=0}^k$, and the roots $\{x_j\}_{j=0}^k$ and weights $\{\omega_j\}_{j=0}^k$ of the Gauss-Legendre quadrature of order $k+1$, and are constructed as the linear combinations

$$\phi_j^I(x) = \sqrt{\omega_j} \sum_{i=0}^{k} \phi_i^L(x_j)\phi_i^L(x), \qquad x \in [0,1] \tag{3.8}$$

This construction leads to orthogonality on the unit interval, as well as the interpolating property

$$\phi_j^I(x_i) = \frac{\delta_{j,i}}{\sqrt{\omega_i}} \tag{3.9}$$

which will prove important for numerical efficiency. A detailed discussion on the properties of interpolating wavelets can be found in Donoho [13].

3.3.3 Wavelet basis

There are two necessary constraints in the construction of the wavelet functions ψ_j: they must be orthogonal to the scaling functions and orthogonal among themselves. It turns out that this is not sufficient in order to determine the wavelet functions uniquely, so Alpert [1] posed additional conditions in terms of vanishing moments. The exact construction is done iteratively, starting with the following set of functions $\{f_j(x)\}_{j=0}^k$ defined on the interval $(-1,1)$

$$f_j(x) = \begin{cases} x^j, & x \in (0,1) \\ -x^j, & x \in (-1,0) \\ 0, & otherwise \end{cases} \tag{3.10}$$

followed by a Gram-Schmidt orthogonalization with respect to the low-order polynomials $1, x, x^2, \ldots, x^k$ that span the corresponding scaling space. Furthermore, we require that the function f_j has $j+1$ additional vanishing moments by orthogonalization with respect to the polynomials $x^{k+1}, \ldots, x^{j+k+1}$, and finally, the functions f_j are orthogonalized among themselves in order of increasing j. The wavelet basis ψ_j of the space W_k^0 is then constructed by dilation and translation to the unit interval, followed by L^2 normalization.

3.4 Function projection

In order to obtain the expansion coefficients of a general function f in the scaling basis we need to evaluate the projection integral in Eq. (2.19). This is done numerically using Gauss-Legendre quadrature

$$s_{j,l}^{n,f} = \int_{2^{-n}l}^{2^{-n}(l+1)} f(x)\phi_{j,l}^{n}(x)\,\mathrm{d}x \tag{3.11}$$

$$= 2^{-n/2}\int_{0}^{1} f\big(2^{-n}(x+l)\big)\phi_j(x)\,\mathrm{d}x \tag{3.12}$$

$$\approx 2^{-n/2}\sum_{q=0}^{k_q-1}\omega_q f\big(2^{-n}(x_q+l)\big)\phi_j(x_q) \tag{3.13}$$

where $\{\omega_q\}_{q=0}^{k_q-1}$ are the weights and $\{x_q\}_{q=0}^{k_q-1}$ the roots of the Legendre polynomial L_{k_q} used in k_q-th order quadrature. The Legendre quadrature holds a $(2k-1)$-rule which states that the k-order quadrature is exact whenever the integrand is a polynomial of order $2k-1$. By choosing $k_q = k+1$ order quadrature, where k is the order of the polynomial basis, we will obtain the exact coefficient whenever $f(x)$ is a polynomial of degree $\leq (k+1)$, and we will use quadrature order $k+1$ throughout.

3.4.1 Projection in d dimensions

In the multi-dimensional case the expansion coefficients are given by multi-dimensional quadrature

$$s_{\boldsymbol{j},\boldsymbol{l}}^{n,f} = 2^{-nd/2}\sum_{q_1=0}^{k}\sum_{q_2=0}^{k}\cdots\sum_{q_d=0}^{k} f\big(2^{-n}(\boldsymbol{x}_q+\boldsymbol{l})\big)\prod_{i=1}^{d}\omega_{q_i}\phi_{j_p}(x_{q_i}) \tag{3.14}$$

using the following notation for the vector of quadrature roots

$$\boldsymbol{x}_q \stackrel{\text{def}}{=} (x_{q_1}, x_{q_2}, \ldots, x_{q_d}) \tag{3.15}$$

This multi-dimensional quadrature is not very efficient in a general polynomial basis, as the number of terms scales as $(k+1)^d$. This can be avoided if the function f is separable and can be written $f(x_1, x_2, \ldots, x_d) = f_1(x_1)f_2(x_2)\cdots f_d(x_d)$, in which Eq. (3.14) can be reduced to a product of mono-dimensional summations with a scaling of $d(k+1)$.

However, working in the Interpolating basis, no assumption needs to be made on the function to obtain numerical efficiency. By choosing a quadrature

order of $k_q = k + 1$, a very important property of the Interpolating scaling functions emerges, that follows from the specific construction of these functions in Eq. (3.8). The interpolating property in Eq. (3.9) inserts a Kronecker delta whenever the scaling function is evaluated in a quadrature root, which is exactly the case in the quadrature sum. This reduces Eq. (3.14) to

$$s_{\boldsymbol{j},\boldsymbol{l}}^{n,f} = 2^{-nd/2} f\big(2^{-n}(\boldsymbol{x}_j + \boldsymbol{l})\big) \prod_{i=1}^{d} \sqrt{\omega_{j_i}} \tag{3.16}$$

which means that the scaling coefficients are related to the function values on the quadrature grid by simple constant factors, leading to very efficient evaluation.

3.4.2 Obtaining the wavelet coefficients

The wavelet coefficients are formally obtained by the projection of the function onto the wavelet basis, and we could derive expressions similar to the scaling expressions based on quadrature. There are however some accuracy issues connected to this wavelet quadrature, so we will take another approach that utilizes the wavelet transform. We know that we can obtain the scaling and wavelet coefficients on scale n by doing a wavelet decomposition of the scaling coefficients on scale $n+1$ according to Eq. (2.11). Line 7 of Alg. 1 is thus performed by computing the scaling coefficients of the 2^d children of the current `node` by the appropriate expression (Legendre or Interpolating) followed by a wavelet decomposition.

3.4.3 Estimating the `tree` structure

In projection of analytic functions it is quite straightforward to predict the final adaptive `tree` structure of the representation without any actual calculation of coefficients. E.g. in the case of Gaussian ($e^{-\beta(\boldsymbol{x}-\boldsymbol{x}_0)^2}$) and Slater ($e^{-\beta|\boldsymbol{x}-\boldsymbol{x}_0|}$) type functions, the position $\boldsymbol{x}_0$ and exponent β tells you where and approximately how much the grid needs to be refined. Furthermore, in the case of very narrow, high-exponent functions this "forced" refinement is essential, as the quadrature at the coarsest scale would probably not pick up any signal at all, giving a zero-representation of the function.

3.5 Arithmetic operations

3.5.1 Addition

The recipe for the addition of two function `trees` follows straightforwardly from the mappings in Eq. (2.34). Consider the equation $h(x) = f(x) + g(x)$. Projecting h onto the scaling space V_k^n yields

$$h^n(x) = P_k^n \left(f(x) + g(x) \right) \tag{3.17}$$

$$= P_k^n f(x) + P_k^n g(x) \tag{3.18}$$

$$= f^n(x) + g^n(x) \tag{3.19}$$

and similarly for the wavelet projections. At a deeper level it simply means adding scaling and wavelet coefficients on corresponding `nodes`

$$s_{j,l}^{n,h} = s_{j,l}^{n,f} + s_{j,l}^{n,g} \tag{3.20}$$

$$w_{j,l}^{n,h} = w_{j,l}^{n,f} + w_{j,l}^{n,g} \tag{3.21}$$

If the given `node` does not exist in the representation of either f or g, it is obtained by oversampling using the wavelet transform Eq. (2.11). No absolute accuracy will be lost during an addition, but *relative* accuracy might be lost if the additon reduces the norm of the function.

3.5.2 Multiplication

Consider the equation $h(x) = f(x) \times g(x)$. In practice this means to multiply the representations f^n and g^n

$$h(x) \approx \hat{h}(x) \stackrel{\text{def}}{=} f^n(x) \times g^n(x) \tag{3.22}$$

However, as we have seen in Sec. 2.3.5, the product of the scaling representations at scale n will give wavelet contributions at higher scales, and Beylkin [7] suggests to perform the multiplication of *oversampled* function representations

$$\hat{h}^{n+1} \;=\; P_k^{n+1}\Big(\uparrow (f^n) \times \uparrow (g^n) \Big) \tag{3.23}$$

to allow enough flexibility in the basis to represent the product. In our implementation the adaptive algorithm will take care of the extra refinement in the

product only if and where it is necessary. We will thus perform the multiplication in Eq. (3.22) purely on the given scale n, which means that we project the product of the representations back onto the scaling space V_k^n

$$\hat{h}^n = P_k^n\big(f^n \times g^n\big) \tag{3.24}$$

and the coefficients of the product are approximated by the projection integral

$$\begin{aligned} s_{j^h,l}^{n,h} &\approx \int_{2^{-n}l}^{2^{-n}(l+1)} \hat{h}(x)\phi_{j^h,l}^n(x)\,\mathrm{d}x && (3.25)\\ &= \int_{2^{-n}l}^{2^{-n}(l+1)} f^n(x)g^n(x)\phi_{j^h,l}^n(x)\,\mathrm{d}x && (3.26)\\ &= 2^{-n/2}\int_0^1 f^n\big(2^{-n}(x+l)\big)g^n\big(2^{-n}(x+l)\big)\phi_{j^h}(x)\,\mathrm{d}x && (3.27)\end{aligned}$$

The projection integral is again done by Gauss-Legendre quadrature and all the information we need from the multiplicands are their pointvalues in the quadrature roots $\{x_q\}_{q=0}^k$ at scale n, which can be obtained from their respective scaling coefficients

$$\begin{aligned} s_{j^h,l}^{n,h} &\approx 2^{-n/2}\sum_{q=0}^{k}\omega_q f^n\big(2^{-n}(x_q+l)\big)g^n\big(2^{-n}(x_q+l)\big)\phi_{j^h}(x_q) && (3.28)\\ &= 2^{n/2}\sum_{q=0}^{k}\omega_q\left(\sum_{j^f=0}^{k} s_{j^f,l}^{n,f}\phi_{j^f}(x_q)\right)\left(\sum_{j^g=0}^{k} s_{j^g,l}^{n,g}\phi_{j^g}(x_q)\right)\phi_{j^h}(x_q) && (3.29)\end{aligned}$$

3.5.3 Multiplication in d dimensions

Generalizing the above expression for multiple dimensions reveals that multiplication will become a time consuming process in a general polynomial basis

$$\begin{aligned} s_{\boldsymbol{j}^h,\boldsymbol{l}}^{n,h} \approx 2^{nd/2}\sum_{q_1=0}^{k}\sum_{q_2=0}^{k}\cdots\sum_{q_d=0}^{k}\Bigg(&\left(\prod_{i=1}^{d}\omega_{q_i}\right)\\ &\times\left(\sum_{j_1^f=0}^{k}\sum_{j_2^f=0}^{k}\cdots\sum_{j_d^f=0}^{k} s_{\boldsymbol{j}^f\boldsymbol{l}}^{n,f}\left(\prod_{i=1}^{d}\phi_{j_i^f}(x_{q_i})\right)\right)\\ &\times\left(\sum_{j_1^g=0}^{k}\sum_{j_2^g=0}^{k}\cdots\sum_{j_d^g=0}^{k} s_{\boldsymbol{j}^g\boldsymbol{l}}^{n,g}\left(\prod_{i=1}^{d}\phi_{j_i^g}(x_{q_i})\right)\right)\\ &\times\left(\prod_{i=1}^{d}\phi_{j_i^h}(x_{q_i})\right)\Bigg)\end{aligned} \tag{3.30}$$

The scaling behavior of this expression is $(k+1)^{2d}$, however, the only function evaluations that are actually taking place are again the $k+1$ different scaling

functions evaluated in the $k+1$ different quadrature roots. These $(k+1)^2$ function values need to be evaluated only once, and fetched from memory whenever needed in the expression Eq. (3.30), which will speed up the process.

Working in the Interpolating basis, the multiplication complexity is significantly reduced, as the basis is specifically designed to return Kronecker deltas when evaluated in the quadrature roots. Inserting this property into Eq. (3.30) will remove all nested summations are left with a single term in the evaluation of the coefficient of the product

$$s^{n,h}_{\boldsymbol{j}^h\boldsymbol{l}} = 2^{nd/2} s^{n,f}_{\boldsymbol{j}^h\boldsymbol{l}} s^{n,g}_{\boldsymbol{j}^h\boldsymbol{l}} \prod_{i=1}^{d} \frac{1}{\sqrt{\omega_{j_i^h}}} \tag{3.31}$$

3.5.4 Obtaining the wavelet coefficients

In the case of multiplication, the calculation of the wavelet coefficients on a given scale n is done in the same way as for the projection, by wavelet transform of the scaling coefficients at scale $n+1$. Line 7 of Alg. 1 is again obtained by calculation of the scaling coefficients of the 2^d children of the current `node` by the appropriate expression (Legendre or Interpolating), followed by a wavelet decomposition.

3.5.5 Estimating the `tree` structure

In both addition and multiplication we use the union of the `tree` structures of the input functions as the starting guess for the `tree` structure of the result. In the case of addition, there is no need for further refinement, as there will be no wavelet contribution beyond this level of refinement in the result. In multiplications, however, it might be necessary to refine a scale or two locally, and this is taken care of by the adaptive algorithm.

3.6 Operator construction

It was shown in Chap. 2 that the matrix elements of a general one-dimensional integral operator is obtained by projection of the two-dimensional integral kernel onto the multiwavelet basis. This corresponds to a regular function projection, as described in Sec. 3.4, and at the end of the day the construction of such

operators will follow the algorithms presented above for projection. However, for d-dimensional problems, the integral kernel will in general have $2d$ dimensions, and this complexity needs to be reduced in order to obtain efficient algorithms both in the construction and application of operators in multiple dimensions. This can be achieved by the technique of separation of variables.

3.6.1 Separated representation of operators

In the discussion of multi-dimensional operators in Chap. 2 it was assumed that the kernel is separable in the Cartesian coordinates. This assumption is necessary in order to make calculations feasable in higher dimensions, as the straightforward generalization of a one-dimensional approach leads to a prohibitive exponential scaling in the dimension. It is, however, not necessary that the operator separates exactly, and Beylkin and Mohlenkamp [10, 11] shows that the integral kernel of many physically interresting operators can be approximated as a linear combination of products of one-dimensional kernels

$$K(\boldsymbol{x}, \boldsymbol{y}) \approx \hat{K}(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \sum_{\kappa=1}^{M} \alpha_\kappa \prod_{p=1}^{d} K_p^\kappa(x_p, y_p) \tag{3.32}$$

The accuracy of this separated representation can be controlled by adapting the functions K_p^κ, the expansion coefficients α_κ and the separation rank M, and any precision can in principle be achieved. Such a representation allows the multi-dimensional operator to be applied one dimension at the time, reducing the computational complexity from k^{2d} per `node` of the full non-separable operator, to Mdk^{d+1} per `node` of the separated representation in Eq. (3.32), where k is the order of the polynomial basis. While the scaling is still exponential in the dimension, the exponent is sufficiently reduced for the approach to be applicable for $d = 2, 3$.

3.6.2 Poisson kernel

The Poisson equation is usually written in its differential form

$$\nabla^2 g(\boldsymbol{x}) = -f(\boldsymbol{x}) \tag{3.33}$$

and the solution of can be expressed in terms of the convolution integral

$$g(\boldsymbol{x}) = \int P(\boldsymbol{x} - \boldsymbol{y}) f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} \tag{3.34}$$

where $P(\boldsymbol{x}-\boldsymbol{y})$ is the Green's function satisfying the fundamental equation with free boundary conditions (zero at infinity)

$$\nabla^2 P(\boldsymbol{x}-\boldsymbol{y}) = -\delta(\boldsymbol{x}-\boldsymbol{y}) \tag{3.35}$$

This equation can be solved analytically and the Green's function for the Poisson equation for $d = 3$ is given as

$$P(\boldsymbol{x}-\boldsymbol{y}) = \frac{1}{4\pi\|\boldsymbol{x}-\boldsymbol{y}\|} \tag{3.36}$$

3.6.3 Helmholtz kernel

The inhomogeneous Helmholtz equation (also called screened Poisson equation) is a generalization of the Poisson equation, and is given in differential form

$$\left(\nabla^2 - \mu^2\right) g(\boldsymbol{x}) = -f(\boldsymbol{x}) \tag{3.37}$$

The solution can again be expressed as an integral

$$g(\boldsymbol{x}) = \int H^{\mu}(\boldsymbol{x}-\boldsymbol{y}) f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \tag{3.38}$$

using the Helmholtz kernel $H^{\mu}(\boldsymbol{x}-\boldsymbol{y})$, which is the Greeen's function satisfying the fundamental equation

$$\left(\nabla^2 - \mu^2\right) H^{\mu}(\boldsymbol{x}-\boldsymbol{y}) = -\delta(\boldsymbol{x}-\boldsymbol{y}) \tag{3.39}$$

with zero boundary conditions at infinity. The Green's function for the Helmholtz equation for $d = 3$ is known analytically as

$$H^{\mu}(\boldsymbol{x}-\boldsymbol{y}) = \frac{e^{-\mu\|\boldsymbol{x}-\boldsymbol{y}\|}}{4\pi\|\boldsymbol{x}-\boldsymbol{y}\|} \tag{3.40}$$

3.6.4 Separation using Gaussians

Neither the Poisson nor the Helmholtz kernel is separable in the Cartesian coordinates, but it is possible to obtain a separated representation as in Eq. (3.32) of low rank using Gaussian functions

$$K(\boldsymbol{x}-\boldsymbol{y}) \approx \hat{K}(\boldsymbol{x}-\boldsymbol{y}) = \sum_{\kappa=1}^{M} \alpha_{\kappa} e^{-\beta_{\kappa}\|\boldsymbol{x}-\boldsymbol{y}\|^2} \tag{3.41}$$

This representation is motivated by the well known integral representation of the Poisson kernel[14]

$$P(r) = \frac{1}{r} = \frac{4}{\sqrt{\pi}} \int_0^\infty e^{-4r^2t^2}\,\mathrm{d}t, \quad r \stackrel{\text{def}}{=} \|\boldsymbol{x} - \boldsymbol{y}\| \tag{3.42}$$

and the parameters in Eq. (3.41) are obtained in the case of the Poisson kernel by transforming Eq. (3.42) into an integral of super-exponential decay, and discretizing using the trapezoidal rule [15, 5]. In this way, and similarly in the case of the Helmholtz kernel, it is possible to obtain a separated representation $\hat{K}$ of Eq.(3.41) with accuracy ϵ_s over a finite interval

$$sup_{r>0}\left|\frac{K(r) - \hat{K}(r)}{K(r)}\right| < \epsilon_s, \quad r \in [r_0, r_1] \tag{3.43}$$

where the upper bound r_1 should be chosen as the longest possible distance in the computational domain ($r_1 = \sqrt{3}$ for the unit cube), and the lower bound r_0 should be chosen so that the contribution due to the integration at the singularity can be neglected[5].

3.6.5 Derivative kernel

As a final note we show how we can obtain approximate representations of the derivative operator using the framework of integral operators presented above. The derivative operator can be expressed as

$$\frac{\mathrm{d}}{\mathrm{d}x} f(x) = \int \frac{\mathrm{d}}{\mathrm{d}x} \delta(x-y) f(y)\,\mathrm{d}y \tag{3.44}$$

where the delta function can be approximated by a high-exponent Gaussian

$$\delta(x-y) \approx \sqrt{\frac{\beta}{\pi}} e^{-\beta(x-y)^2} \tag{3.45}$$

which is normalized so that it integrates to unity. This approximation can be differentiated, and the derivative operator can be expressed as the integral

$$\frac{\mathrm{d}}{\mathrm{d}x} f(x) = \int D(x-y) f(y)\,\mathrm{d}y \tag{3.46}$$

using the derivative kernel

$$D(x-y) = \frac{\mathrm{d}}{\mathrm{d}x} \sqrt{\frac{\beta}{\pi}} e^{-\beta(x-y)^2} = -2\beta \sqrt{\frac{\beta}{\pi}} (x-y) e^{-\beta(x-y)^2} \tag{3.47}$$

This representation approaches the exact derivative as defined by Alpert *et al.* [9] and presented in Sec. 2.4.6 as the Gaussian in Eq. (3.45) approaches the delta function ($\beta \to \infty$).

3.6.6 Cross-Correlation functions

All operators presented above involve integrals with convolution kernels $K(x, y) = K(x - y)$, and the matrix elements can be expressed in terms of the cross-correlation of the scaling functions [16]

$$\Phi_{ij}(z) = \int_0^1 \phi_i(z + y)\phi_j(y)\,\mathrm{d}y \tag{3.48}$$

and the two-dimensional projection integral is reduced to one dimension

$$\left[\tau_{lm}^n\right]_{ij} = \int_0^1 \int_0^1 K(x - y)\phi_i^n(x - l)\phi_j^n(y - m)\,\mathrm{d}x\,\mathrm{d}y \tag{3.49}$$

$$= \int_{-1}^1 K(z)\Phi_{ij}(2^n z + m - l)\,\mathrm{d}z \tag{3.50}$$

For d-dimensional operators the kernel is $2d$-dimensional, and the cross-correlation functions will reduce the integral to d dimensions. Moreover, if the kernel is separable, the matrix element can be computed as products of one-dimensional integrals

$$\left[\tau_{\boldsymbol{lm}}^n\right]_{ij} = \prod_{p=1}^d \int_{-1}^1 K_p(z_p)\Phi_{ij}\left(2^n z_p + m_p - l_p\right)\mathrm{d}z_p \tag{3.51}$$

which significantly reduces the cost of constructing multi-dimensional operators.

3.7 Operator application

In the non-standard operator application given in the matrix equation (2.59), the length scales of the problem have been explicitly separated. In this way it is possible to use Alg. 1 to adaptively build the resulting function `tree`, also in several dimensions. For a `node` at a given scale n we need to calculate the scaling and wavelet representations of the resulting function g

$$g^n + \mathrm{d}g^n = \left({}^nA^n + {}^nB^n + {}^nC^n + {}^nT^n \right)\left(f^n + \mathrm{d}f^n\right) \tag{3.52}$$

but as was pointed out in the theory part in Sec. 2.4, the T part of the operator is only applied at the coarsest scale, and thus, no interaction with the coarser scales are taken into account for $n > 0$. However, when the operator is applied scale by scale, the effect of the missing T part at scale n has already been calculated at scale $n-1$, and this information can be retrieved by making use of

the wavelet transform in Eq. (2.11). We define the following auxiliary functions

$$\hat{g}^n \stackrel{\text{def}}{=} {}^nT^n f^n \tag{3.53}$$

$$\tilde{g}^n \stackrel{\text{def}}{=} {}^nC^n \, \mathrm{d}f^n \tag{3.54}$$

$$\mathrm{d}\tilde{g}^n \stackrel{\text{def}}{=} \left({}^nA^n + {}^nB^n \right) \left(f^n + \mathrm{d}f^n \right) \tag{3.55}$$

where all three contributions are calculated at the coarsest scale. At all scales $n > 0$, however, we only need to calculate $\tilde{g}^n$ and $\mathrm{d}\tilde{g}^n$, as $\hat{g}^n$ can be obtained from the next coarser scale

$$\hat{g}^n = \hat{g}^{n-1} + \tilde{g}^{n-1} + \mathrm{d}\tilde{g}^{n-1}, \qquad n > 0 \tag{3.56}$$

and this is continued locally `node` by `node` until we reach a representation of sufficient accuracy, following the same algorithm as before.

3.7.1 Obtaining the coefficients

Algorithm 2 Operator application. Inserted in line 7 of Alg. 1

1: **for** each separated component ($\kappa = 1, \ldots, M$) of the operator **do**
2: **for** each ($\alpha = 0, \ldots, 2^d - 1$) of output function **do**
3: **for** each ($\beta = 0, \ldots, 2^d - 1$) of input function **do**
4: get operator component $O_\kappa^{\alpha\beta,n} = \bigotimes_{p=1}^d O_\kappa^{\alpha_p\beta_p,n}$
5: construct bandwidth
6: fetch input and operator `nodes` within bandwidth
7: prune list of input `nodes` based on norm product $\left(\|O_{\kappa,\boldsymbol{l}-\boldsymbol{m}}^{\alpha\beta,n}\|_2 \cdot \|\boldsymbol{w}_{\boldsymbol{l}}^{\alpha,n,f}\|\right)$
8: **for** each contributing input `node` **do**
9: apply operator $\boldsymbol{w}_{\boldsymbol{l}}^{\alpha,n,g} += \bigotimes_{p=1}^d O_{\kappa,\boldsymbol{l}-\boldsymbol{m}}^{\alpha_p\beta_p,n} \boldsymbol{w}_{\boldsymbol{m}}^{\beta_p,n,f}$
10: **end for**
11: **end for**
12: **end for**
13: **end for**

The calculation of scaling/wavelet coefficients (line 7 of Alg. 1) in the operator application is somewhat involved in multiple dimensions, and is presented in Alg. 2. Each component of the separated representation of the operator needs to be applied separately in order to exploit the tensorial structure of the operator. Also, the different separated components will have very different bandwidths at a given scale (the higher the Gaussian exponent of the operator, the deeper in

scale its main contribution will be). A more detailed discussion of the algorithm can be found in Frediani *et al.* [5].

The bandwith of each specific operator component can be calculated according to Eq. (2.74), and by explicitly treating and thresholding all 2^{2d} (A, B, C and T in each dimension) operator components the number of contributing terms is reduced significantly, with prospects of algorithms that scale linearly with system size.

In parallel computations where the data of the functions involved are distributed among the memory of several computational hosts, the presented algorithm inevitably requires some communication, as the calculation of a given `node` of the result requires all `nodes` of the input function within the bandwidth, and these input `nodes` are not necessarily located on the same host. There are different strategies for how this data transfer can be performed, and this is discussed in publication II, where the performance (linear scaling and parallelization) of the code is presented.

The actual calculation of the coefficients is performed in the following way, for simplicity presented for a single operator component in one dimension. At the coarsest scale, in this case $n = 0$, the T part of the operator is applied, where we according to Eq. (2.65) have

$$\hat{g}^0(x) = {}^0T^0 f^0(x) \tag{3.57}$$

$$\sum_i \hat{s}^{0,g}_{i,0}\phi^0_{i,0}(x) = \sum_{i,j} \left[\tau^0_{00}\right]_{ij} s^{0,f}_{j,0}\phi^0_{i,0}(x) \tag{3.58}$$

$$\hat{s}^{0,g}_{i,0} = \sum_j \left[\tau^0_{00}\right]_{ij} s^{0,f}_{j,0} \tag{3.59}$$

The A, B and C parts are applied at all scales $n \geq 0$, and from Eqs. (2.69)-(2.71), we see that we get a contribution to the scaling coefficient from C

$$\tilde{g}^n(x) = {}^nC^n\,\mathrm{d}f^n(x) \tag{3.60}$$

$$\sum_l \sum_i \tilde{s}^{n,g}_{i,l}\phi^n_{i,l}(x) = \sum_{l,m}\sum_{i,j} \left[\gamma^n_{lm}\right]_{ij} w^{n,f}_{j,m}\phi^n_{i,l}(x) \tag{3.61}$$

$$\tilde{s}^{n,g}_{i,l} = \sum_m \sum_j \left[\gamma^n_{lm}\right]_{ij} w^{n,f}_{j,m} \tag{3.62}$$

while the wavelet coefficients are obtained from parts B and A

$$\mathrm{d}\tilde{g}^n(x) = {}^nB^n f^n + {}^nA^n \,\mathrm{d}f^n \tag{3.63}$$

$$\sum_l \sum_i \tilde{w}_{i,l}^{n,g} \psi_{i,l}^n(x) = \sum_{l,m} \sum_{i,j} \Big(\big[\beta_{lm}^n\big]_{ij} s_{j,m}^{n,f} + \big[\alpha_{lm}^n\big]_{ij} w_{j,m}^{n,f} \Big) \psi_{i,l}^n(x) \tag{3.64}$$

$$\tilde{w}_{i,l}^{n,g} = \sum_m \sum_j \Big(\big[\beta_{lm}^n\big]_{ij} s_{j,m}^{n,f} + \big[\alpha_{lm}^n\big]_{ij} w_{j,m}^{n,f} \Big) \tag{3.65}$$

For all scales $n > 0$ the T part is obtained by wavelet reconstruction of the result at the next coarser scale according to Eq. (3.56)

$$\hat{s}_{i,(l=even)}^{n,g} = \sum_j \Big(H_{ji}^{(0)} \big(\hat{s}_{j,l/2}^{n-1,g} + \tilde{s}_{j,l/2}^{n-1,g}\big) + G_{ji}^{(0)} \tilde{w}_{j,l/2}^{n-1,g} \Big) \tag{3.66}$$

$$\hat{s}_{i,(l=odd)}^{n,g} = \sum_j \Big(H_{ji}^{(1)} \big(\hat{s}_{j,(l-1)/2}^{n-1,g} + \tilde{s}_{j,(l-1)/2}^{n-1,g}\big) + G_{ji}^{(1)} \tilde{w}_{j,(l-1)/2}^{n-1,g} \Big) \tag{3.67}$$

3.7.2 Estimating the `tree` structure

One way of estimating the `tree` structure in the case of operator application is simply to copy the grid of the input function, which is done by Beylkin *et al.* [17]. However, as the integral operators treated in this work are known for their smoothing properties, the output function will in general require a coarser (but possibly wider) grid than the input, and such a construction will lead to an overestimation of the grid refinement.

Instead we will set up a much simplified operator whose purpose is only to build the initial grid. We have found that by only applying the purly diagonal part ($l = m$) of the original operator, we capture more than 95% of the norm of the result, but at a fraction of the computational cost, and by building an adaptive grid using this operator, we end up with a `tree` structure that is quite close to the final grid of the full operator. Only when this estimated grid is complete we apply the full operator, and the grid is further refined if needed. Moreover, if the operator expansion has M terms, it is in general not necessary to include all of them in the simplified operator, and typically $M/10$ should be sufficient, if the terms are chosen among the full range of exponents.

Chapter 4

Electronic structure theory

In this chapter we present the equations that govern chemical systems, in particular the electronic stucture of atoms and molecules. At the molecular length scale, nature is most accurately described by the theory of quantum mechanics, where the central problem is the solution of the non-relativistic Schrödinger equation.

Being that this problem cannot be solved exactly by any analytical method whenever the system contains more than two particles, much of the work in the field of quantum chemistry has been concerned with developing accurate and efficient approximations, a work that has been given invaluable support by the developments in computer technology over the last half-century.

This chapter will give an introduction to the self-consistent field (SCF) approximations that are commonly employed in computational chemistry. We will start with a traditional presentation of the orbital based methods of Hartree-Fock and Kohn-Sham density functional theory, where the aim of the chapter is to rewrite the equations into their less familiar integral form. An optimization algorithm using the mathematical tools as implemented in Chap. 3 is demonstrated for simple one-electron systems, while the treatment of general many-electron systems is the topic of publication III.

Most of the exposition follows that of the standard textbooks of computational chemistry, like Szabo and Ostlund[18], Parr and Yang[19] and Jensen[20], as well as the thesis of Losilla[21].

4.1 The electronic Schrödinger equation

The physical state of a quantum system influenced by potentials that do not change with time is described by the time-independent Schrödinger equation

$$\hat{H}\Psi = E\Psi \tag{4.1}$$

where the Hamiltonian $\hat{H}$ is the operator for the total energy E of the system. The wave function Ψ is an eigenfunction of the Hamiltonian operator, and is a multi-dimensional (in general complex-valued) function that depends on the degrees of freedom of the system, e.i. the position $\boldsymbol{r}$ and spin s of all N particles, and we have $\Psi = \Psi(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$, where $\boldsymbol{x}_i = (\boldsymbol{r}_i, s_i)$ denotes the position and spin of the i-th particle. There are in general infinitely many eigenfunctions for a given Hamiltonian operator, each corresponding to a possible state.

The wave function contains all the information that can possibly be extracted from the physical system. For each physical observable Ω there is an associated mathematical operator $\hat{\Omega}$, such that the expectation value of an experimental measurement is given by

$$\langle\hat{\Omega}\rangle = \frac{\langle\Psi|\hat{\Omega}|\Psi\rangle}{\langle\Psi|\Psi\rangle} \tag{4.2}$$

This means that the fundamental problem in quantum chemistry is to obtain the molecular wave function by solving the Schrödinger equation (4.1). For a molecule, the Hamiltonian contains kinetic $\hat{T}$ and potential $\hat{V}$ energy of the electrons and nuclei that make up the system

$$\hat{H} = \hat{T}_{nuc} + \hat{T}_{el} + \hat{V}_{nn} + \hat{V}_{ee} + \hat{V}_{ne} \tag{4.3}$$

Analytic solutions exists only for the one- and two-particle problems, and approximations are inevitable if we want to be able to treat more interresting chemical systems.

The first approximation for molecular systems is almost exclusively the Born-Oppenheimer approximation[22], in which we consider the nuclei to be fixed in space, so that the electrons move in a static nuclear potential. The motivation behind this approximation is that the nuclei are much heavier than the electrons, and hence move much slower, so that at the electronic time scale, the nuclei are percieved as classical particles frozen in space. This means that we can disregard

the instantaneous correlation between the electrons and the nuclei, and we can separate the nuclear kinetic energy from an electronic Hamiltonian

$$\hat{H} = \hat{T}_{nuc} + \hat{H}_{el} \tag{4.4}$$

$$\hat{H}_{el} = \hat{T}_{el} + \hat{V}_{ne} + \hat{V}_{ee} + \hat{V}_{nn} \tag{4.5}$$

In atomic units[1], using uppercase indices for the nuclei and lowercase indices for the electrons, we have the electron kinetic energy

$$\hat{T}_{el} = -\sum_i \frac{1}{2}\nabla_i^2 \tag{4.6}$$

the electron-nuclear attraction

$$\hat{V}_{ne} = -\sum_{i,I} \frac{Z_I}{\|\boldsymbol{r}_i - \boldsymbol{R}_I\|} \tag{4.7}$$

the electron-electron repulsion

$$\hat{V}_{ee} = \sum_{i>j} \frac{1}{\|\boldsymbol{r}_j - \boldsymbol{r}_i\|} \tag{4.8}$$

and finally the nuclear-nuclear repulsion

$$\hat{V}_{nn} = \sum_{I>J} \frac{Z_I Z_J}{\|\boldsymbol{R}_I - \boldsymbol{R}_J\|} \tag{4.9}$$

Within the Born-Oppenheimer approximation, the last term is a simple additive constant and is usually left out when solving the electronic problem

$$\hat{H}_{el}\psi_{el} = E_{el}\psi_{el} \tag{4.10}$$

At the nuclear time scale, the electrons are percieved as a diffuse charge density that is able to respond instantaneously to the movement of the nuclei, and molecular rotations and vibrations are described by the nuclear wave function which is influenced by this dynamic electron density. In the following, however, we are concerned exclusively with the calculation of the electronic wave function through Eq. (4.10), where the *el* subscript henceforth will be dropped.

The particular state ψ_0 with the lowest energy E_0 is called the electronic ground state of the system and serves special attention in quantum chemistry. The reason for this is that for most chemical systems the ground state is the only

[1] $e = m_e = \hbar = 4\pi\epsilon_0 = 1$

state significantly populated under normal laboratory conditions, and hence, most chemical phenomena can be explained in terms of properties of the electronic ground state. The way to calculate the ground state is usually to exploit the variational principle, which states that for a given Hamiltonian $\hat{H}$ with true ground state ψ_0, we have for an arbitrary trial wave function $\tilde{\psi}$

$$\frac{\langle\tilde{\psi}|\hat{H}|\tilde{\psi}\rangle}{\langle\tilde{\psi}|\tilde{\psi}\rangle} \geq \frac{\langle\psi_0|\hat{H}|\psi_0\rangle}{\langle\psi_0|\psi_0\rangle} \tag{4.11}$$

which means that finding the ground state can be regarded as a minimization problem, where the trial wave function is varied to the point where the corresponding energy is minimized.

4.2 Hartree-Fock Theory

The most apparent complication in developing approximate methods for the solution of the electronic Schödinger equation is perhaps the high dimensionality of the problem. For a system containing N electrons, the wave function is a $3N$-dimensional scalar function (disregarding spin). The common way to approach such high-dimensional problems is by approximating the full d-dimensional function in terms of products of functions of lower dimensionality. In chemistry it is convenient to use one-particle functions ϕ_i, called spin-orbitals, which depend on the coordinates of a single electron

$$\psi(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N) = \sum_m c_m \phi_1^m(\boldsymbol{x}_1)\phi_2^m(\boldsymbol{x}_2)\cdots\phi_N^m(\boldsymbol{x}_N) \tag{4.12}$$

Unfortunately, the convergence of such expansions is not very good, and a large number of terms is usually required in order to obtain high accuracy (chemical accuracy is usually defined as 1 kcal/mol). One way of improving the convergence is to include two-particle functions in the expansion. Such approaches, known as *explicitly correlated methods*[23, 24], will not be discussed in this thesis, and in the following we use wave functions constructed using one-particle functions in the form of a Slater determinant.

4.2.1 Slater determinant

Being fermionic, the electronic wave function needs to be anti-symmetric with respect to the exchange of two particles

$$\psi(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_N) = -\psi(\boldsymbol{x}_2, \boldsymbol{x}_1, \boldsymbol{x}_3 \ldots, \boldsymbol{x}_N) \tag{4.13}$$

This condition is known as the Pauli exclusion principle[25], which has the consequence that each fermionic state can only be occupied by one particle. The simplest way of constructing a wave function that fulfills the anti-symmetry requirement using one-particle spin-orbitals is the Slater determinant[26]

$$\psi = |\phi_1 \phi_2 \cdots \phi_N\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{N!}} \begin{vmatrix} \phi_1(\boldsymbol{x}_1) & \phi_1(\boldsymbol{x}_2) & \cdots & \phi_1(\boldsymbol{x}_N) \\ \phi_2(\boldsymbol{x}_1) & \phi_2(\boldsymbol{x}_2) & \cdots & \phi_2(\boldsymbol{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_N(\boldsymbol{x}_1) & \phi_N(\boldsymbol{x}_2) & \cdots & \phi_N(\boldsymbol{x}_N) \end{vmatrix} \tag{4.14}$$

where the spin-orbitals $\phi_i(\boldsymbol{x})$ are orthonormal and can be expressed as a product of a three-dimensional spatial part and a spin part. The energy of such a wave function is evaluated as the expectation value of the Hamiltonian

$$E[\psi] = \langle \phi_1 \phi_2 \cdots \phi_N | \hat{H} | \phi_1 \phi_2 \cdots \phi_N \rangle \tag{4.15}$$

$$= \sum_{i=1}^{N} \langle \phi_i | \hat{h} | \phi_i \rangle + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \langle \phi_i | \hat{J}_j - \hat{K}_j | \phi_i \rangle \tag{4.16}$$

where we have defined the one-electron operator

$$\hat{h}\phi_i(\boldsymbol{x}) = \left(-\frac{1}{2}\nabla^2 - \sum_I \frac{Z_I}{\|\boldsymbol{r} - \boldsymbol{R}_I\|} \right) \phi_i(\boldsymbol{x}) \tag{4.17}$$

as well as the Coulomb $\hat{J}_j$ and exchange $\hat{K}_j$ operators

$$\hat{J}_j \phi_i(\boldsymbol{x}) = \left(\int \frac{\phi_j^*(\boldsymbol{x}')\phi_j(\boldsymbol{x}')}{\|\boldsymbol{r} - \boldsymbol{r}'\|} \, \mathrm{d}\boldsymbol{x}' \right) \phi_i(\boldsymbol{x}) \tag{4.18}$$

$$\hat{K}_j \phi_i(\boldsymbol{x}) = \left(\int \frac{\phi_j^*(\boldsymbol{x}')\phi_i(\boldsymbol{x}')}{\|\boldsymbol{r} - \boldsymbol{r}'\|} \, \mathrm{d}\boldsymbol{x}' \right) \phi_j(\boldsymbol{x}) \tag{4.19}$$

where it is important to note that the integration is over space *and* spin coordinates, which means that the exchange operator is zero if the spin of orbitals i and j differ. The Coulomb operator, on the other hand, in non-vanishing for all pairs of spin-orbitals.

4.2.2 The Hartree-Fock equations

The best approximation to the ground state in terms of a *single* Slater determinant is called the Hartree-Fock wave function, and is obtained by minimizing the energy with respect to orbital variations

$$E_0 = \min_{\psi} E[\psi] \tag{4.20}$$

following the variational principle of Eq. (4.11). In the following we will assume that we have a closed-shell system, so that the N electrons are grouped into $N/2$ pairs sharing the same spatial function, but with opposite spins

$$\phi_i^\sigma(\boldsymbol{x}) = \phi_i(\boldsymbol{r})\sigma(s), \qquad \sigma = \alpha, \beta \tag{4.21}$$

By imposing the constraint that the spatial orbitals remain orthonormal $\langle \phi_i | \phi_j \rangle = \delta_{i,j}$ by means of Lagrange multipliers, the energy minimization yields the Hartree-Fock equations

$$\hat{F}\phi_i(\boldsymbol{r}) = \epsilon_i \phi_i(\boldsymbol{r}) \tag{4.22}$$

where the Fock operator is given as

$$\hat{F} = \hat{h} + \sum_j^{N/2} \left(2\hat{J}_j - \hat{K}_j \right) \tag{4.23}$$

The (restricted) Hartree-Fock wave function is then obtained as the Slater determinant constructed by the $N/2$ lowest energy eigenfunctions ϕ_i of the Fock operator, each appearing twice with paired spins

$$\psi = |\phi_1^\alpha \phi_1^\beta \cdots \phi_{N/2}^\alpha \phi_{N/2}^\beta\rangle \tag{4.24}$$

Some of the terms included in the Fock operator can be expressed as multiplicative potentials instead of operators. The core Hamiltonian $\hat{h}$ includes the scalar electrostatic potential arising from the nuclear charges

$$v_{nuc}(\boldsymbol{r}) = \sum_I \frac{Z_I}{\|\boldsymbol{r} - \boldsymbol{R}_I\|} \tag{4.25}$$

and the sum of the Coulomb operators is the potential arising from all electrons of the system

$$v_{el}(\boldsymbol{r}) = \sum_j^{N/2} 2\hat{J}_j = 2\sum_j^{N/2} \int \frac{|\phi_j(\boldsymbol{r}')|^2}{\|\boldsymbol{r} - \boldsymbol{r}'\|} \, \mathrm{d}\boldsymbol{r}' \tag{4.26}$$

If we further collect the exchange operators into a single operator

$$\hat{K}\phi_i(\boldsymbol{r}) = \sum_j^{N/2} \hat{K}_j \phi_i(\boldsymbol{r}) = \sum_j^{N/2} \phi_j(\boldsymbol{r}) \int \frac{\phi_j^*(\boldsymbol{r}')\phi_i(\boldsymbol{r}')}{\|\boldsymbol{r}-\boldsymbol{r}'\|} \, \mathrm{d}\boldsymbol{r}' \tag{4.27}$$

we can write the Hartree-Fock equations as

$$\Big[-\frac{1}{2}\nabla^2 + v_{nuc}(\boldsymbol{r}) + v_{el}(\boldsymbol{r}) - \hat{K} \Big] \phi_i(\boldsymbol{r}) = \epsilon_i \phi_i(\boldsymbol{r}) \tag{4.28}$$

As both the electronic potential v_{el} and the exchange operator $\hat{K}$ depend on the set of occupied orbitals, we have a set of coupled non-linear differential equations that need to be solved iteratively until we reach a self-consistent solution.

The main deficiancy of such a self-consistent field (SCF) approximation is that each electron only interacts with the average field created by the other electrons. While this is a good approximation for the electron's interaction with the slow moving nuclei, the instantaneous correlation is more important between two electrons. The Hartree-Fock method still provides a reasonable qualitative description of molecules near their equilibrium geometry, capturing 95-99% of the total energy. This, however, is generally not sufficient in order to reach chemical accuracy, and there exist several post-Hartree-Fock methods that model the missing *correlation* energy, including configuration interaction (CI) and coupled-cluster (CC) theory, but these will not be discussed (see e.g.[18, 20, 27]).

4.3 Density Functional Theory

We have seen that the main computational challenge in solving the Schrödinger equation is its high dimensionality, and that by introducing one-particle orbitals the $3N$-dimensional differential equation can be separated into N ($N/2$ for a closed-shell system) coupled three-dimensional equations. Hohenberg and Kohn[28] showed that the complexity can be reduced even further by proving that the only quantity that is really needed in order to determine the system uniquely is the three-dimensional electron density

$$\rho(\boldsymbol{r}_1) = N \int |\psi(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)|^2 \, \mathrm{d}s_1 \, \mathrm{d}\boldsymbol{x}_2 \cdots \mathrm{d}\boldsymbol{x}_N \tag{4.29}$$

and that the true energy of the system can be expressed in terms of a universal energy functional

$$E[\rho] = T[\rho] + V_{ne}[\rho] + V_{ee}[\rho] \tag{4.30}$$

where the ground state density can be obtained by minimizing the energy

$$E_0 = \min_{\rho} E[\rho] \tag{4.31}$$

with the constraints that the density is everywhere positive and integrates to the number of electrons. Within the Born-Oppenheimer approximation the electron-nuclear interaction energy is known as the classical electrostatic energy between charge densities

$$V_{ne}[\rho] = \int \rho(\boldsymbol{r}) v_{nuc}(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r} \tag{4.32}$$

with the nuclear potential defined through Eq. (4.25), but the functional form of the kinetic and electron-electron energies are not known for quantum mechanical densities (as we have seen in the previous section, the quantum mechanical interaction between electrons includes both exchange and correlation energy, in addition to the classical electrostatic interaction), and the fundamental problem in density functional theory (DFT) is to find good approximations for these energy functionals, either based on theoretical considerations, or semi-empirically by fitting parameters to experimental data.

4.3.1 The Kohn-Sham equations

The general idea of DFT appears very appealing, as we only need to solve one three-dimensional equation for the electron density. However, it turns out to be very difficult to find good approximations for the kinetic energy functional, and according to the virial theorem this energy is of the order of the total energy of the system, and thus needs to be accurately represented. To circumvent this problem, Kohn and Sham[29] proposed to express the density in terms of one-particle functions, which for a closed shell system with double occupancy yields

$$\rho(\boldsymbol{r}) = 2 \sum_{i}^{N/2} |\phi_i(\boldsymbol{r})|^2 \tag{4.33}$$

thus reintroducing the orbital notion of Hartree-Fock theory. The motivation behind this is that the kinetic energy is known for a set of (non-interacting) orbitals as

$$T_s[\rho] = 2\sum_i^{N/2} \langle \phi_i | -\frac{1}{2}\nabla^2 | \phi_i \rangle \tag{4.34}$$

However, this is not equal to the real kinetic energy of the (interacting) system, and we are missing a small part of the total energy $T[\rho] - T_s[\rho]$. We can similarly extract the known classical part from the density's interaction with itself

$$J[\rho] = \frac{1}{2}\int\int \frac{\rho(\boldsymbol{r})\rho(\boldsymbol{r}')}{\|\boldsymbol{r}-\boldsymbol{r}'\|}\,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}' = \frac{1}{2}\int \rho(\boldsymbol{r}) v_{el}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \tag{4.35}$$

where again we are missing a small part of the total energy $V_{ee}[\rho] - J[\rho]$. The custom in Kohn-Sham theory is then to collect the missing parts into a single exchange-correlation functional

$$E_{xc}[\rho] = T[\rho] - T_s[\rho] + V_{ee}[\rho] - J[\rho] \tag{4.36}$$

and we get the total Kohn-Sham energy expressed as

$$E[\rho] = T_s[\rho] + V_{en}[\rho] + J[\rho] + E_{xc}[\rho] \tag{4.37}$$

Minimizing the energy with respect to the density leads to the Euler equation

$$\mu = \frac{\delta T_s[\rho]}{\delta \rho(\boldsymbol{r})} + v_{eff}(\boldsymbol{r}) \tag{4.38}$$

where the chemical potential μ is a Lagrange multiplier that fixes the number of electrons, and the effective potential is given in terms of functional derivatives

$$v_{eff}(\boldsymbol{r}) = \frac{\delta V_{en}[\rho]}{\delta \rho(\boldsymbol{r})} + \frac{\delta J[\rho]}{\delta \rho(\boldsymbol{r})} + \frac{\delta E_{xc}[\rho]}{\delta \rho(\boldsymbol{r})} \tag{4.39}$$

$$= v_{nuc}(\boldsymbol{r}) + v_{el}(\boldsymbol{r}) + v_{xc}(\boldsymbol{r}) \tag{4.40}$$

The Euler equation (4.38) describes a system of non-interacting electrons moving in an effective potential v_{eff}, and the Hamiltonian for such a system is given trivially as

$$\hat{H} = -\sum_i^{N/2} \frac{1}{2}\nabla_i^2 + \sum_i^{N/2} v_{eff}(\boldsymbol{r}_i) \tag{4.41}$$

This operator is separable and the exact wave function is a single determinant constructed by the $N/2$ lowest energy eigenfunctions of the Fock (or Kohn-Sham) operator

$$\hat{F} = -\frac{1}{2}\nabla^2 + v_{eff}(\boldsymbol{r}) \tag{4.42}$$

each appearing twice with paired spins, and the minimization problem of the DFT Euler equation now entails solving the Kohn-Sham equations

$$\Big[-\frac{1}{2}\nabla^2 + v_{nuc}(\boldsymbol{r}) + v_{el}(\boldsymbol{r}) + v_{xc}(\boldsymbol{r})\Big]\phi_i(\boldsymbol{r}) = \epsilon_i\phi_i(\boldsymbol{r}) \tag{4.43}$$

We see that by reintroducing orbitals we abandon the hope of expressing the problem in terms of a single three-dimensional equation, and again we get a set of $N/2$ coupled non-linear equations for the orbitals. As the effective potential in the Kohn-Sham operator depends on the density, and thus on the orbitals, Kohn-Sham DFT is also referred to as an SCF method, and given the similarity with the Hartree-Fock equations (4.28), the same techniques can be used to solve both problems.

4.3.2 Density functional approximations

As already mentioned, the exact form of the universal exchange-correlation functional is not known, so the quality of any Kohn-Sham calculation is only as good as the quality of the density functional approximation (DFA) being used. The exchange-correlation energy is expressed as an integral over an energy density

$$E_{xc}[\rho] = \int F_{xc}\,\mathrm{d}\boldsymbol{r} \tag{4.44}$$

In the local density approximation (LDA) the energy density is a function of the density alone $F_{xc}(\rho)$, in the generalized gradient approximation (GGA) it is a function of the density and its gradient $F_{xc}(\rho, |\nabla\rho|)$, while in meta-GGA's, higher order derivatives are introduced $F_{xc}(\rho, |\nabla\rho|, \nabla^2\rho, \cdots)$. Hybrid functionals are GGA's with a certain amount of exact Hartree-Fock exchange, evaluated as in Eq. (4.27) using Kohn-Sham orbitals. This increasing complexity in the DFA will in general yield increasingly accurate results.

The exchange-correlation potential was implicitly defined in Eq. (4.40) as the functional derivative of the exchange-correlation energy with respect to the density

$$v_{xc} = \frac{\delta E_{xc}[\rho]}{\delta\rho} = \frac{\delta}{\delta\rho}\int F_{xc}\,\mathrm{d}\boldsymbol{r} \tag{4.45}$$

which can be calculated for LDAs and GGAs through

$$v_{xc}^{LDA} = \frac{\partial F_{xc}}{\partial \rho} \tag{4.46}$$

$$v_{xc}^{GGA} = \frac{\partial F_{xc}}{\partial \rho} - \nabla \cdot \frac{\partial F_{xc}}{\partial \nabla \rho} \tag{4.47}$$

A wide range of DFAs are available in the literature, with different costs, accuracies and ranges of applicability[30].

4.4 Basis sets in computational chemistry

Even with the approximations presented in the previous sections, the SCF equations are still to complicated to be solved analytically for many-electron systems, and we rely on numerical solution algorithms in order to make the theoretical methods useful. As computers work in finite arithmetic using floating point numbers of finite accuracy, we need to discretize the problem in one way or another. This can be done either by representing functions as a collection of point values on a grid with some kind of regularity, where for instance differential operators can be defined through finite differences, or by expanding the solution in terms of a set of basis functions χ_p

$$f(\boldsymbol{r}) = \sum_{p}^{\infty} c_p \chi_p(\boldsymbol{r}) \approx \sum_{p}^{N} c_p \chi_p(\boldsymbol{r}) \tag{4.48}$$

The equality in Eq. (4.48) holds for any function f if the basis set is complete, but this usually requires an infinite expansion. In practice, the expansion is truncated at some point, yielding an approximation of the given function, and the problem has been discretized to a finite number of expansion coefficients c_p.

In principle any set of linearly independent functions can be used as a basis, but there are certain properties that we want from the basis for it to be computationally attractive[21]

- ***A**ccuracy*

  The basis set must be able to represent the target functions faithfully, and provide results that are sufficiently accurate for a given purpose.

- ***C**ompactness*

  For a given accuracy, the size of the basis set should be as small as possible.

- ***Efficiency***

  The mathematical operations that involve the basis functions should be performed as fast as possible.

- ***Systematicity***

  The basis set should depend on a set of parameters that can be modified such that the accuracy of a given calculation will improve.

- ***Universality***

  The performance, in terms of accuracy and efficiency, should be adequate to model a large variety of properties and systems.

It turns out that no basis can give you all these properties at once, so we have to make some kind of compromise when choosing a basis set for a certain problem, and the choice will often depend on known analytical properties of the solution.

For instance, it is known that the ground state wave function is continuous, but not differentiable at the nuclear positions[31]. Similar *cusps* appear in the wave function when the coordinate of two electrons coincide, as well as for the molecular orbitals and the electron density at the nuclear positions. Specifically, the behavior of the density close to a nucleus is known to be

$$\rho(\boldsymbol{r}) \sim e^{-2Z_J|\boldsymbol{r}-\boldsymbol{R}_J|}, \qquad |\boldsymbol{r}-\boldsymbol{R}_J| \ll 1 \tag{4.49}$$

while it decays exponentially at long distances

$$\rho(\boldsymbol{r}) \sim e^{-2\sqrt{2E_I}|\boldsymbol{r}-\boldsymbol{R}_J|}, \qquad |\boldsymbol{r}-\boldsymbol{R}_J| \gg 1 \tag{4.50}$$

where E_I is the ionization potential. Similar conditions apply for the molecular orbitals.

4.4.1 Atom-centered basis functions

It is desireable to use basis functions with the same asymptotic behavior as the density in order to get efficient representations, e.i. localized functions centered at the nuclear positions, with a short range cusp and an exponential tail. Furthermore, the chemical notion of a molecule being a collection of atoms suggests that a reasonable approach would be to express the molecular orbitals

(MOs) as linear combinations of atomic orbitals (LCAO)

$$\phi_i(\boldsymbol{r}) = \sum_I \sum_p^{M_I} c_{ip}\chi_p(\boldsymbol{r} - \boldsymbol{R}_I) \tag{4.51}$$

where the atomic orbitals (AOs) are atom-centered functions similar to the eigenfunctions of the hydrogen atom. Even if the presence of several nuclei in a molecule breaks the angular symmetry around each atom, the nuclear potential is so steep that the symmetry is to a large extent retained in the vicinity of the nucleus. The AOs are thus chosen to be spherically symmetric functions that can be separated into an angular part, in the form of spherical harmonics $Y_{lm}(\theta, \varphi)$, and a radial part $R(r)$

$$\chi_p(\boldsymbol{r}) = R_p(r)Y_{l_p,m_p}(\theta, \varphi) \tag{4.52}$$

This basis can approach completeness both in the angular part, by increasing the maximum angular momentum L in the spherical harmonics, and in the radial part by adding more linearly independent radial functions. It is well established that the convergence in the angular part is exponential ($\sim e^{-\sqrt{L}}$) for Hartree-Fock energies (for post-Hartree-Fock methods the convergence is slower $\sim L^{-3}$), which means that very large L is typically not needed for SCF calculations.

By choosing exponential radial functions

$$R_p^{STO}(r) = N_p r^{n_p} e^{-\xi_p r} \tag{4.53}$$

we get the so-called Slater type orbitals (STO)[32], which have the correct asymptotic behavior. This means that the basis is rather efficient for describing molecular orbitals and densities, leading to compact representations and fairly rapid basis set convergence also for the radial part. The main problem, however, with STOs is numerical efficiency. In Hartree-Fock calculations the main bottleneck is the evaluation of three- and four-center two-electron integrals in the form

$$g_{pqrs} = \int\int \chi_p(\boldsymbol{r}_1)\chi_q(\boldsymbol{r}_1)\frac{1}{\|\boldsymbol{r}_1 - \boldsymbol{r}_2\|}\chi_r(\boldsymbol{r}_2)\chi_s(\boldsymbol{r}_2)\,\mathrm{d}\boldsymbol{r}_1\,\mathrm{d}\boldsymbol{r}_2 \tag{4.54}$$

for which there exist no analytic formula in the case of STOs. For this reason, the main applications for the STO basis is for small systems (atoms and diatomics) where high accuracy is required, or for density functional methods that do not

include exact exchange, and where the Coulomb energy is calculated using an auxiliary basis.

The computational efficiency of the evaluation of two-electron integrals can be dramatically improved by choosing Gaussian type orbitals (GTOs)[33], where the radial functions have the form

$$R_p^{GTO}(r) = N_p r^{n_p} e^{-\xi_p r^2} \tag{4.55}$$

In this case the integrals can be calculated analytically, however, the r^2 dependence in the exponential makes the GTOs inferior to the STOs in describing molecular orbitals and densities, as they do not have the correct asymptotic behavior: at the nucleus the GTO has zero slope instead of a cusp, and it falls off too rapidly at long distances. This means that much larger basis sets are required for a given accuacy, but this is more than compensated for in terms of computational efficiency by the ease of which the required integrals can be calculated. Furthermore, by using contracted GTOs, where each basis function can contain several primitive Gaussians

$$R_p^{cGTO}(r) = r^{n_p} \sum_j a_{pj} e^{-\xi_{pj} r^2} \tag{4.56}$$

where the coefficients a_{pj} are kept fixed, we can to a large degree compensate for the incorrect asymptotic behavior, while keeping the number of variational parameters that need to be optimized as low as possible. The computational efficiency of the cGTO bases have made them by far the most popular choice in computational chemistry. The parameters (contraction coefficients and exponents) of the basis are preoptimized, usually based on atomic calculations, and there are several basis set families that are systematized in sequences of increasing accuracy (and consequently increasing computational cost).

A rigorous systematicity, however, holds only for smaller systems in the lower-quality end of the basis set ladder. When the number of basis functions grows, the basis sets become overcomplete, and linear dependencies appear, leading to numerical instabilities, poorly conditioned equations and poor convergence of iterative methods. This also affects the minimum error attainable, making it difficult to approach the basis set limit for a given level of theory.

Another problem of atom-centered basis sets is their lack of universality. The preoptimization of the parameters biases the results towards a particular

property, making it difficult to judge the quality of the calculation of other properties.

4.4.2 Plane wave basis functions

Rather than using localized AO-like basis functions that are trying to model each atom separately, and forming molecular orbitals through LCAOs, one can start with basis functions that are aimed directly at the full system. This approach is most appropriate for modelling infinite systems represented by a unit cell with periodic boundary conditions, such as metals where the valence electrons are delocalized and thus well represented by solutions of the free electron Schrödinger equation. The three-dimensional plane wave basis is usually written in terms of complex exponentials

$$\chi_p(\boldsymbol{r}) = e^{i\boldsymbol{k}_p \cdot \boldsymbol{r}} \tag{4.57}$$

where the wave vector $\boldsymbol{k}$ gives the oscillation frequency and is related to the energy of the basis function. The size of the basis is determined by the sampling resolution in k-space (spacing between k-vectors) and the highest energy $\boldsymbol{k}$-vector included, which depend on the size of the unit cell, and is usually significantly larger than the size of typical Gaussian basis sets.

Plane waves can in principle be used for non-periodic systems as well, by placing the molecule in a sufficiently large unit cell where its interaction with its own image in the neighboring cells can be neglected. However, placing a small molecule in a large unit cell requires disproportionally many basis functions, and the molecule is represented much more efficiently using localized atomic orbitals.

The plane wave basis is also ill-suited to represent the core region of atoms, where many rapidly oscillating functions are required, and especially the singularity in the nuclear potential, which is almost impossible to describe in this basis. On the other hand, plane waves are ideal for representing the smooth density of delocalized valence electrons, and are usually used in connection with pseudopotentials[**?**], where the effect of the core electrons are combined with the nuclear charges to give an *effective core potential*, and only the valence electrons are treated explicitly. This, in combination with the fast Fourier transform

(FFT), have made plane wave methods the preferred choice for the treatment of many-particle problems of condensed phases.

4.4.3 Real-space representations

Most of the problems connected with atom-centered basis sets are related to their global support, and these issues can be adressed using numerical real-space methods. In these methods each expansion coefficient is usually directly related to the function value at a certain grid point in space, and a systematic improvement of the accuracy is readily obtained by decreasing the spacing between the grid points. The finite element (FE) basis is considered a real-space method even if the representations are given through basis set expansions. The reason for this is that the basis is grouped into a small number of n functions sharing the same compact support, disjoint from the support of all other basis functions, making them responsible for the function representation in a certain region of real space. The expansion coefficients are usually obtained through numerical quadrature, which means that the n functions are related to n point values. Moreover, using interpolating polynomials each basis function is directly connected to a single grid point.

While the FE bases can solve the problems of the AO basis concerning systematicity, universality and attainable accuracy, they suffer from a lack of compactness of the representation. Originally, the FE bases required a uniform grid, making them highly inefficient for the treatment of multiscale problems like the electronic structure of molecules, where high precision requires high resolution in the nuclear region. A uniform grid will in this case result in an excessive overrepresentation of the much smoother interatomic region, making accurate calculations very computationally demanding, even if the fundamental mathematical operations involving the polynomial basis are very efficient.

Due to the high cost of real-space methods, applications in electronic structrure calculations are uncommon, and for a long time they were limited to benchmarking calculations on small systems of high symmetry[34, 35, 36, 37]. Some attempts have been made to overcome the problem, either by removing the high frequency core region by means of pseudopotentials, or by combining the FE basis with another basis of AO type with complementary properties that

is able to treat the nuclear region more efficiently[38, 39, 40, 41]. Another approach, which is the one persued in this work, that is applicable to all-electron calculations of systems of arbitrary geometries, is based on multiresolution analysis and the multiwavelet basis. This approach, that was pioneered by Harrison and coworkers[42, 43, 44, 45] ten years ago, allows for strict error control using adaptive non-uniform grids, thus reducing the computational cost significantly.

4.5 Integral formulation

The discretization of the Hartree-Fock (4.28) and Kohn-Sham (4.43) equations using the atom-centered basis leads to the Roothaan-Hall[46, 47] matrix equations that are solved iteratively using standard convergence acceleration techniques like the direct inversion of the iterative subspace (DIIS)[48]. This approach is not appropriate for the FE and multiwavelet bases due to the high number of basis functions involved, as well as the requirement of a fixed basis set. Moreover, in a discontinuous basis, differential operators (especially higher order operators like the kinetic energy) should be avoided in order to maintain high accuracy[42].

Following Harrison *et al.* [42], we use Kalos'[49] integral formulation of the Schrödinger equation, and in the following we rewrite the Hartree-Fock (4.28) and Kohn-Sham (4.43) equations into their integral form, using the integral convolution operators

$$g(\boldsymbol{r}) = \hat{G}\big[f\big](\boldsymbol{r}) \stackrel{\text{def}}{=} \int G(\boldsymbol{r}-\boldsymbol{r}')f(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}' \tag{4.58}$$

that were presented in Chap. 3, where we specifically described the implementation of the Poisson, the bound-state Helmholtz and the first order derivative operators, with respective integral kernels

$$P(\boldsymbol{r}-\boldsymbol{r}') = \frac{1}{4\pi\|\boldsymbol{r}-\boldsymbol{r}'\|} \tag{4.59}$$

$$H^{\mu}(\boldsymbol{r}-\boldsymbol{r}') = \frac{e^{-\mu\|\boldsymbol{r}-\boldsymbol{r}'\|}}{4\pi\|\boldsymbol{r}-\boldsymbol{r}'\|} \tag{4.60}$$

$$D(x-x') = -2\beta\sqrt{\frac{\beta}{\pi}}(x-x')e^{-\beta(x-x')^2} \tag{4.61}$$

The Poisson operator $\hat{P} = \big[-\nabla^2\big]^{-1}$ will be used in the calculation of electrostatic potentials as well as the Hartree-Fock exchange operator, the Helmholtz

operator $\hat{H}^{\mu} = \big[-\nabla^2+\mu^2\big]^{-1}$ appears in the integral formulation of the Hartree-Fock and Kohn-Sham equations, and the derivative operator $\hat{D}^x$ is needed for the calculation of exchange-correlation potentials using GGA functionals through Eq. (4.47).

4.5.1 Hartree-Fock

In the closed-shell restricted Hartree-Fock model, the electron density is given from $N/2$ doubly occupied orbitals

$$\rho(\boldsymbol{r}) = \sum_{i}^{N/2} 2|\phi_i(\boldsymbol{r})|^2 \tag{4.62}$$

The electronic potential is calculated from the electron density by application of the Poisson operator

$$v_{el}(\boldsymbol{r}) = \hat{P}\big[\rho\big](\boldsymbol{r}) \tag{4.63}$$

and we denote the total Coulomb potential experienced by the electrons as

$$v_{coul}(\boldsymbol{r}) = v_{nuc}(\boldsymbol{r}) + v_{el}(\boldsymbol{r}) \tag{4.64}$$

The exchange operator can also be expressed in terms of the Poisson operator

$$\hat{K}\phi_i(\boldsymbol{r}) = \sum_{j}^{N/2} \phi_j(\boldsymbol{r})\hat{P}\big[\phi_i\phi_j\big](\boldsymbol{r}) \tag{4.65}$$

Furthermore, we can rearrange the Hartree-Fock equations so that they can be expressed in terms of the Helmholtz operator

$$\Big[-\frac{1}{2}\nabla^2 + v_{coul}(\boldsymbol{r}) + \hat{K}\Big]\phi_i(\boldsymbol{r}) = \epsilon_i\phi_i(\boldsymbol{r}) \tag{4.66}$$

$$\big[-\nabla^2 - 2\epsilon_i\big]\phi_i(\boldsymbol{r}) = -2\Big[\big(v_{coul}(\boldsymbol{r}) - \hat{K}\big)\phi_i(\boldsymbol{r}) \tag{4.67}$$

$$\phi_i = -2\hat{H}^{\mu_i}\Big[\big(v_{coul} - \hat{K}\big)\phi_i\Big] \tag{4.68}$$

with $\mu_i = \sqrt{-2\epsilon_i}$. The equations are still implicitly coupled through the electronic potential and the exchange operator, and need to be solved self-consistently by iterative methods. Note that both the orbitals ϕ_i and their corresponding energy ϵ_i are unknowns in the equations, and must be determined simultaneously.

4.5.2 Kohn-Sham DFT

In the Kohn-Sham equations the exchange operator is replaced by the exchange-correlation potential, which for a given functional can be calculated from Eqs. (4.46) and (4.47) for LDAs and GGAs, respectively, using the gradient operator $\nabla = \left(\hat{D}^x, \hat{D}^y, \hat{D}^z\right)$ in case of the latter. Following the same procedure as for the Hartree-Fock equations we get $N/2$ separated equations

$$\Big[-\frac{1}{2}\nabla^2 + v_{eff}(\boldsymbol{r})\Big]\phi_i(\boldsymbol{r}) = \epsilon_i\phi_i(\boldsymbol{r}) \tag{4.69}$$

$$\phi_i = -2\hat{H}^{\mu_i}\Big[v_{eff}\phi_i\Big] \tag{4.70}$$

where $\mu_i = \sqrt{-2\epsilon_i}$. Again, the equations are coupled through the effective potential, and are solved self-consistently with respect to the orbitals and energies.

4.5.3 Calculation of energy

We will now assume that the Hartree-Fock or Kohn-Sham equations have been solved to obtain the orbitals ϕ_i that make up ground state wave function, as well as their energies ϵ_i, and use these to calculate the electronic energy of the molecular system. Numerical algorithms for how to solve these equations are presented in Sec. 4.6 in the simple case of a one-electron system, and more generally in publication III for many-electron systems. In addition to the electronic energy we have the constant nuclear repulsion energy

$$\hat{V}_{nn} = \sum_{I>J} \frac{Z_I Z_J}{\|\boldsymbol{R}_I - \boldsymbol{R}_J\|} \tag{4.71}$$

The goal of this section is to rewrite the expressions given above into something better suited for evaluation in the multiwavelet framework. In particular this means to avoid the application of the kinetic energy operator.

Hartree-Fock

The energy of a Slater determinant wave function was given in Eq. (4.16), which can be expressed in the following way, assuming a closed-shell system and doubly

occupied orbitals

$$E = \sum_i^{N/2} 2\langle\phi_i|\hat{h}|\phi_i\rangle + \frac{1}{2}\sum_i^{N/2} 2\langle\phi_i|2\hat{J} - \hat{K}|\phi_i\rangle \tag{4.72}$$

$$= \sum_i^{N/2} 2\langle\phi_i|\hat{T}|\phi_i\rangle + \sum_i^{N/2} 2\langle\phi_i|v_{nuc}|\phi_i\rangle + \sum_i^{N/2}\langle\phi_i|v_{el} - \hat{K}|\phi_i\rangle \tag{4.73}$$

$$= \sum_i^{N/2}\langle\phi_i|2\hat{T} - \hat{K}|\phi_i\rangle + \int \rho(\boldsymbol{r})v_{nuc}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} + \frac{1}{2}\int \rho(\boldsymbol{r})v_{el}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \tag{4.74}$$

The kinetic energy operator can be avoided by making the following observation

$$\sum_i^{N/2} 2\epsilon_i = \sum_i^{N/2} 2\langle\phi_i|\hat{T} + v_{nuc} + v_{el} - \hat{K}|\phi_i\rangle \tag{4.75}$$

$$= \sum_i^{N/2} 2\langle\phi_i|\hat{T} - \hat{K}|\phi_i\rangle + \int \rho(\boldsymbol{r})v_{nuc}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} + \int \rho(\boldsymbol{r})v_{el}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \tag{4.76}$$

Comparing the expressions in Eqs. (4.74) and (4.76) we see that the total electronic energy can be calculated as

$$E = 2\sum_i^{N/2}\epsilon_i - \frac{1}{2}\int \rho(\boldsymbol{r})v_{el}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} - \sum_i^{N/2}\langle\phi_i|\hat{K}|\phi_i\rangle \tag{4.77}$$

without the need of applying the kinetic energy operator, given the orbitals and orbital energies that solves the Hartree-Fock equations.

Kohn-Sham DFT

The energy in Kohn-Sham DFT was given through the energy functionals

$$E[\rho] = T_s[\rho] + V_{en}[\rho] + J[\rho] + E_{xc}[\rho] \tag{4.78}$$

which for a closed-shell system with double occupancy gives

$$E = \sum_i^{N/2} 2\langle\phi_i|\hat{T}|\phi_i\rangle + \int \rho(\boldsymbol{r})v_{nuc}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} + \frac{1}{2}\int \rho(\boldsymbol{r})v_{el}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} + \int F_{xc}\,\mathrm{d}\boldsymbol{r} \tag{4.79}$$

The sum of orbital energies can be expressed as

$$\sum_i^{N/2} 2\epsilon_i = \sum_i^{N/2} 2\langle\phi_i|\hat{T} + v_{eff}|\phi_i\rangle \tag{4.80}$$

$$= \sum_i^{N/2} 2\langle\phi_i|\hat{T}|\phi_i\rangle + \int \rho(\boldsymbol{r})\Big[v_{nuc}(\boldsymbol{r}) + v_{el}(\boldsymbol{r}) + v_{xc}(\boldsymbol{r})\Big]\,\mathrm{d}\boldsymbol{r} \tag{4.81}$$

Combining Eqs. (4.79) and (4.81) gives an expression without kinetic energy

$$E = 2\sum_{i}^{N/2} \epsilon_i - \frac{1}{2}\int \rho(\boldsymbol{r}) v_{el}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} + \int F_{xc}\,\mathrm{d}\boldsymbol{r} - \int \rho(\boldsymbol{r}) v_{xc}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \tag{4.82}$$

where it should be noted that

$$E_{xc}[\rho] = \int F_{xc}\,\mathrm{d}\boldsymbol{r} \neq \int \rho(\boldsymbol{r}) v_{xc}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \tag{4.83}$$

4.6 Iterative solution algorithms

We will illustrate the iterative algorithms by looking at a simple one-electron system in which the electron is influenced only by a fixed nuclear potential $\hat{V} = v_{nuc}(\boldsymbol{r})$, which include the H atom, the He^+ and H_2^+ ions or any other one-electron molecular ion within the Born-Oppenheimer approximation. Just as the Hartree-Fock and Kohn-Sham equations presented above, the electronic Scrödinger equation is rewritten in integral form

$$\Big[-\frac{1}{2}\nabla^2 + \hat{V}\Big]\psi(\boldsymbol{r}) = E\psi(\boldsymbol{r}) \tag{4.84}$$

$$\psi(\boldsymbol{r}) = -2\int H^{\mu}(\boldsymbol{r}-\boldsymbol{r}')\hat{V}(\boldsymbol{r}')\psi(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}' \tag{4.85}$$

$$\psi = -2\hat{H}^{\mu}\Big[\hat{V}\psi\Big] \tag{4.86}$$

with $\mu = \sqrt{-2E}$. This equation needs to be solved with respect to both the wave function ψ and the energy E.

4.6.1 The power method

Eq. (4.86) defines a fixed-point problem, and perhaps the simplest procedure to solve such a problem is the power method, where the operator is applied iteratively

$$\tilde{\psi}^{n+1} = -2\hat{H}^{\mu^n}\Big[\hat{V}\psi^n\Big] \tag{4.87}$$

$$\psi^{n+1} = \frac{\tilde{\psi}^{n+1}}{\|\tilde{\psi}^{n+1}\|} \tag{4.88}$$

The tilde on the new wave function denotes that it is no longer normalized, as the operator $\hat{H}^{\mu}$ does not conserve the norm when the eigenvalue is not exact[49]. The iteration label on the operator reflects the fact that the operator

depends on the energy through $\mu^n = \sqrt{-2E^n}$ which needs to be updated in each iteration.

Such an iteration sequence $\boldsymbol{x}^{n+1} = \hat{O}(\boldsymbol{x}^n)$ will converge to the lowest energy eigenfunction of $\hat{O}$, provided that $\hat{O}$ defines a so-called contraction map. Schneider *et al.* [50] proves linear convergence of the wave function and quadratic convergence of the energy for a simplified *fixed* operator $\hat{O}$ (a general proof of the convergence of the Hartree-Fock and Kohn-Sham equations is yet to be found).

4.6.2 Energy calculation

The energy of the wave function is formally calculated as the expectation value

$$E = \frac{\langle \psi | \hat{T} + \hat{V} | \psi \rangle}{\langle \psi | \psi \rangle} \tag{4.89}$$

where $\hat{T} = -\nabla^2/2$ is the kinetic energy operator, and the potential energy operator in this case is the fixed nuclear potential $\hat{V} = v_{nuc}(\boldsymbol{r})$. As pointed out above, it is desirable to avoid the application of the kinetic operator, so following Harrison *et al.* [42] we exploit the fact that the Helmholtz operator is basically the inverse of the kinetic operator $2\hat{H}^\mu = (\hat{T} - E)^{-1}$, and extract the energy through the application of this operator. Given a wave function ψ^n and energy E^n (this does not have to be the exact energy of ψ^n, but it must be the energy used in $\mu^n = \sqrt{-2E^n}$ in the construction of the operator $\hat{H}^{\mu^n}$) at one iteration, we can calculate the (exact) energy E^{n+1} of the wave function ψ^{n+1} at the next iteration as follows

$$\begin{aligned}
\tilde{E}^{n+1} &= \langle \tilde{\psi}^{n+1} | \hat{T} + \hat{V} | \tilde{\psi}^{n+1} \rangle && (4.90)\\
&= \langle \tilde{\psi}^{n+1} | \hat{T} - E^n | \tilde{\psi}^{n+1} \rangle + \langle \tilde{\psi}^{n+1} | E^n + \hat{V} | \tilde{\psi}^{n+1} \rangle && (4.91)\\
&= \langle \tilde{\psi}^{n+1} | \hat{T} - E^n | - 2\hat{H}^{\mu^n} \left[\hat{V} \psi^n \right] \rangle + \langle \tilde{\psi}^{n+1} | E^n + \hat{V} | \tilde{\psi}^{n+1} \rangle && (4.92)\\
&= -\langle \tilde{\psi}^{n+1} | \hat{V} | \psi^n \rangle + \langle \tilde{\psi}^{n+1} | E^n + \hat{V} | \tilde{\psi}^{n+1} \rangle && (4.93)\\
&= E^n \langle \tilde{\psi}^{n+1} | \tilde{\psi}^{n+1} \rangle + \langle \tilde{\psi}^{n+1} | \hat{V} | \Delta \tilde{\psi}^n \rangle && (4.94)
\end{aligned}$$

where $\Delta\tilde{\psi}^n \stackrel{\text{def}}{=} \tilde{\psi}^{n+1} - \psi^n$. Normalizing this expression gives the energy of ψ^{n+1}, calculated directly from the wave function update

$$E^{n+1} = E^n + \Delta E^n \tag{4.95}$$

$$\Delta E^n = \frac{\langle \tilde{\psi}^{n+1} | \hat{V} | \Delta\tilde{\psi}^n \rangle}{\langle \tilde{\psi}^{n+1} | \tilde{\psi}^{n+1} \rangle} \tag{4.96}$$

without having to apply the kinetic energy operator, provided that the update comes directly from the application of the Helmholtz operator. For future reference, we also define the "normalized" wave function update

$$\Delta\psi^n = \psi^{n+1} - \psi^n = \frac{-2\hat{H}^{\mu^n}\left[\hat{V}\psi^n\right]}{\|\tilde{\psi}^{n+1}\|} - \psi^n \tag{4.97}$$

4.6.3 Krylov subspace accelerated inexact Newton method

The fixed-point problem in Eq. (4.86) can be viewed as finding the roots of the the following residual function

$$f(\psi) = -2\hat{H}^{\mu}\left[\hat{V}\psi\right] - \psi \tag{4.98}$$

which can be done using Newton's method

$$\psi^{n+1} = \psi^n - \left[J(\psi^n)\right]^{-1} f(\psi^n) \tag{4.99}$$

$$= \psi^n - \left[J(\psi^n)\right]^{-1}\left(-2\hat{H}^{\mu^n}\left[\hat{V}\psi^n\right] - \psi^n\right) \tag{4.100}$$

where $J(\psi^n)$ is the Jacobian. Comparing Eq. (4.100) with Eq. (4.87), we can identify the power method as an *inexact* Newton method where the Jacobian is approximated by $J(\psi) \approx -1$. Harrison[51] describes how to make use of the information in the iterative history (Krylov subspace) to improve the approximation of the Jacobian in the Krylov subspace accelerated inexact Newton (KAIN) method. The method is similar to the more commonly used direct inversion of iterative subspace (DIIS) method of Pulay[48], but while DIIS is looking for the best step within the iterative subspace, KAIN is using the same information to extrapolate to a step outside the iterative subspace and is thus considered superior to DIIS[51].

Collecting the wave function and the energy into a vector $\boldsymbol{x} = (\psi, E)$ we get the non-linear equation $f(\boldsymbol{x}) = \mathbf{0}$. At a given iteration n, we have the current approximation $\boldsymbol{x}^n = (\psi^n, E^n)$ and the corresponding residual $f(\boldsymbol{x}^n) =$

$(\Delta\psi^n, \Delta E^n)$ defined through Eqs. (4.96) and (4.97). In the KAIN method the new update $\delta\boldsymbol{x}^n$ is calculated in terms of the m latest iterations

$$\delta\boldsymbol{x}^n = f(\boldsymbol{x}^n) + \sum_{j=1}^{m} c_j \left[\left(\boldsymbol{x}^j - \boldsymbol{x}^n\right) + \left(f(\boldsymbol{x}^j) - f(\boldsymbol{x}^n)\right) \right] \tag{4.101}$$

where the coefficients c_j are obtained by solving the linear system $Ac = b$

$$A_{ij} = \langle \boldsymbol{x}^n - \boldsymbol{x}^i | f(\boldsymbol{x}^n) - f(\boldsymbol{x}^j) \rangle \tag{4.102}$$

$$b_i = \langle \boldsymbol{x}^n - \boldsymbol{x}^i | f(\boldsymbol{x}^n) \rangle \tag{4.103}$$

The size m of the Krylov subspace is without constraints. The larger it is, the better is the Krylov update, but also the larger is the linear system. In general, the Krylov update will not conserve the norm of the wave function, so an additional normalization step should be added at this point.

4.6.4 Algorithm for one-electron systems

The single-orbital algorithm is quite straightforward. Starting from an arbitrary initial guess for the wave function and the energy, the Helmholtz operator is applied once, the resulting wave function is normalized, and the correction $\Delta\psi^n$ and the corresponding energy update ΔE^n is calculated as described above. Then the wave function and energy are added to the KAIN history

$$\boldsymbol{x}^n = (\psi^n, E^n) \qquad\qquad f(\boldsymbol{x}^n) = (\Delta\psi^n, \Delta E^n) \tag{4.104}$$

If the length of the history exceeds some modest number the oldest vector is discarded. New updates are then calculated based on Eq. (4.101)

$$\delta\boldsymbol{x}^n = (\delta\psi^n, \delta E^n) \tag{4.105}$$

which are added to the previous guess, and the iteration is continued until the norm of the wave function update (after the Helmholtz operator application) is below some threshold.

4.6.5 Extension to many-electron systems

There are a few important complications when the algorithm is extended to many-electron systems. In the self-consistent field approximations we get systems of equations involving one-electron orbitals, like the canonical Kohn-Sham

Algorithm 3 Iterative algorithm for the solution of the one-election Schrödinger equation in its integral formulation.

```
Given initial wave function ψ^0 and energy E^0
while ε > threshold do
    Construct Helmholtz operator Ĥ^{μ^n} using μ^n = √(−2E^n)
    Multiply wave function ψ^n with potential
    Apply Helmholtz operator Eq.(4.87) and normalize
    Calculate wave function update Δψ^n = ψ^{n+1} − ψ^n
    Calculate wave function error ε = ‖Δψ^n‖
    Calculate energy update ΔE^n from Eq.(4.96)
    Add (ψ^n, E^n) and (Δψ^n, ΔE^n) to KAIN history
    Calculate KAIN updates (δψ̃^n, δE^n) from Eq.(4.101)
    Update wave function ψ̃^{n+1} = ψ^n + δψ̃^n and normalize
    Update energy E^{n+1} = E^n + δE^n
end while
```

equations

$$\phi_i = -2\hat{H}^{\mu_i}\left[v_{eff}\phi_i\right] \tag{4.106}$$

These equations can be solved in the same way as the one-electron Schrödinger equation presented above, by iterating each equation separately. However, to avoid a collapse of all orbitals into the lowest energy eigenfunction, orthogonality between the orbitals must be explicitly enforced[42]. There are many ways in which this can be achieved, but it is convenient to keep the canonical character of the orbitals throughout the optimization, by calculating and diagonalizing the Fock matrix in each iteration. The calculation of the Fock matrix

$$F_{ij} = \langle\phi_i|\hat{T} + \hat{V}|\phi_j\rangle \tag{4.107}$$

can be done without the need to apply the kinetic energy operator by the same arguments as for the energy calculation of the one-electron wave function, but now the orbital dependence of the effective potential must be accounted for as well. Further complication arises in the KAIN solver, where all orbitals and

energies are included in the Krylov vector

$$\boldsymbol{x}^n = (\phi_0^n, \cdots, \phi_N^n, \epsilon_0^n, \cdots, \epsilon_N^n) \tag{4.108}$$

$$f(\boldsymbol{x}^n) = (\Delta\phi_0^n, \cdots, \Delta\phi_N^n, \Delta\epsilon_0^n, \cdots, \Delta\epsilon_N^n) \tag{4.109}$$

where it is important to keep track of the ordering of the orbitals throughout the iteration, especially in the case of degeneracies, where the orbitals are not uniquely defined. This is discussed further in publication III.

Chapter 5

Orbital-Free DFT

The orbital-based formulation of density functional theory that was introduced by Kohn and Sham[29] fifty years ago has been the most widely used method for determining the electronic structure of molecules during the last few decades. Even without the systematic improvability of the wave function based, post-Hartree-Fock methods, modern density functional approximations are capable of reaching accuracies far surpassing the Hartree-Fock method, but at similar computational cost, although some experience is required for judging the applicability of each functional for a particular problem.

Despite the tremendous success of the method, Kohn-Sham density functional theory (KS-DFT) still runs into trouble when applied to very large systems due to its relience on one-electron orbitals. For an N-electron system, this leads to N coupled, non-linear equations, for which a general solution scales approximately N^3, although several order-N methods have been proposed[52, 53, 54, 55]. Furthermore, in the limit of macroscopic systems, the notion of one-electron orbitals appears utterly impractical, and in fact, the Hohenberg-Kohn[28] theorems suggests that the key quantity should be the three-dimensional electron density, where the energy is given through the universal functional

$$E[\rho] = T_s[\rho] + V_{en}[\rho] + J[\rho] + E_{xc}[\rho] \tag{5.1}$$

In this expression we have kept the notion of non-interacting electrons that was introduced in Kohn-Sham theory, and separated the energy into non-interacting

kinetic energy T_s, classical electrostatic interaction between electrons and nuclei V_{en} and between electrons J, and the quantum mechanical remainder E_{xc}, that accounts for electron exchange and correlation as well as the remaining "interacting" part of the kinetic energy.

5.1 Density functionals

In the early years of quantum mechanics, some attempts was made to model the kinetic and exchange energies as pure density functionals. These models, by the work of Thomas[56], Fermi[57] and Dirac[58], are based on theoretical considerations of the three-dimensional particle-in-a-box problem, and are exact for a non-interacting uniform electron gas. The Thomas-Fermi kinetic energy is given by

$$T_{TF}[\rho] = \frac{3}{10}(3\pi^2)^{2/3} \int \rho^{5/3}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \tag{5.2}$$

whereas the Dirac exchange energy has the form

$$E_x[\rho] = -\frac{3}{4}\left(\frac{3}{\pi}\right)^{1/3} \int \rho^{4/3}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \tag{5.3}$$

Needless to say, the uniform electron gas description does not apply to molecular densities, and the above approximations (especially for the kinetic energy) fail to give even a qualitative description of real chemical systems (Teller[59] even proved that chemical binding is impossible within these models), and for this reason DFT was more or less discarded as a method for chemistry and solid-state physics. At that time, there was also no proof that the energy *could* in fact be expressed as a functional of the electron density, and there was no theory of density functionals.

This, of course, was going to change in the 1960's when a rigorous theory was founded upon the Hohenberg-Kohn theorems, and practical (and accurate) calculations became available through the Kohn-Sham formulation. Even so, the original orbital-free (OF-DFT) formulation was still regarded as unsuited for treating molecular systems, mainly because of the many unsuccessful attempts of improving the accuracy of the kinetic energy functional.

However, some progress have been made over the years. The introduction

of a gradient correction to the Thomas-Fermi energy by von Weizäcker[60]

$$T_W[\rho] = \frac{1}{8}\int \frac{|\nabla\rho(\boldsymbol{r})|^2}{\rho(\boldsymbol{r})}\,\mathrm{d}\boldsymbol{r} \tag{5.4}$$

which gives the exact energy for one- and two-electron (singlet) systems, made chemical binding possible. The more recent approaches are commonly separated into two distinct classes, one-point functionals

$$T_s[\rho] = \int t_s(\rho;\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \tag{5.5}$$

and two-point functionals, which are able to reproduce the shell structure of atomic densities[61]

$$T_s[\rho] = \int\int f_1(\rho;\boldsymbol{r})\chi(\boldsymbol{r},\boldsymbol{r}')f_2(\rho;\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}' \tag{5.6}$$

and a lot of work has gone into the development of new functionals based on purely theoretical considerations, see e.g. Karasiev *et al.* [62]. For instance, the exponents of the density appearing in the Thomas-Fermi and Dirac models are not arbitrary, but satisfy the known coordinate scaling of the exact functional. A functional is said to be homogeneous of degree m under coordinate scaling if it satisfies

$$F[\lambda^3\rho(\lambda\boldsymbol{r})] = \lambda^m F[\rho(\boldsymbol{r})] \tag{5.7}$$

and the exact exchange and non-interacting kinetic energies are homogeneus of degrees 1 and 2, respectively, leading to their respective exponents $\rho^{4/3}$ and $\rho^{5/3}$.

In a recent work, Borgoo and Tozer[63] have looked into the less familiar *density scaling*, where a functional homogeneous of order k satisfies

$$F[\lambda\rho(\boldsymbol{r})] = \lambda^k F[\rho(\boldsymbol{r})] \tag{5.8}$$

and the exact functional is believed to be inhomogeneous. However, the sufficiently accurate approximation that would make OF-DFT useful for the description of molecular systems remains to be found[64], although some applications are found for large, periodic systems in condensed-phase physics in combination with pseudo-potentials, where the valence electrons are better approximated as a uniform electron gas[65, 66].

Nevertheless, with the highly appealing prospect of fully realizing the Hohenberg-Kohn theorems by expressing the energy purely as a functional of the density, work continues in finding better approximations.

5.2 Solution of the Euler equation

In OF-DFT, the ground state density is obtained by solving a single three-dimensional Euler equation

$$\frac{\delta T_s[\rho]}{\delta \rho(\boldsymbol{r})} + v_{KS}(\boldsymbol{r}) = \mu \tag{5.9}$$

where v_{KS} is the effective potential of Kohn-Sham theory, as defined in Eq. (4.40), and μ is the chemical potential. As the problem now involves the treatment of just a few global functions (density and potentials), instead of N (possibly localized) one-electron orbitals appearing in KS-DFT, the lack of compactness of real-space representations becomes less of a problem[67, 68]. In particular, properties such as grid adaptivity and guaranteed accuracy should make the multiwavelet basis well suited to tackle the problem, if the equations can be formulated in such a way that an efficient optimization is possible.

It is common to separate the non-interacting kinetic energy into the von Weizäcker contribution given in Eq. (5.4) plus a non-negative remainder, known as the Pauli term

$$T_s[\rho] = T_W[\rho] + T_\theta[\rho], \qquad T_\theta[\rho] \geq 0 \tag{5.10}$$

The functional derivative of the von Weizäcker energy is

$$\frac{\delta T_W[\rho]}{\delta \rho(\boldsymbol{r})} = \frac{1}{\sqrt{\rho(\boldsymbol{r})}}\Big(-\frac{1}{2}\nabla^2\Big)\sqrt{\rho(\boldsymbol{r})} \tag{5.11}$$

which brings the Euler equation over to the form

$$\Big[-\frac{1}{2}\nabla^2 + v_\theta(\boldsymbol{r}) + v_{KS}(\boldsymbol{r})\Big]\sqrt{\rho(\boldsymbol{r})} = \mu\sqrt{\rho(\boldsymbol{r})} \tag{5.12}$$

which is identical to the Kohn-Sham equations for one "orbital" $\phi(\boldsymbol{r}) = \sqrt{\rho(\boldsymbol{r})}$ and effective potential $v_{eff} = v_\theta + v_{nuc} + v_{el} + v_{xc}$

$$\Big[-\frac{1}{2}\nabla^2 + v_{eff}(\boldsymbol{r})\Big]\phi(\boldsymbol{r}) = \mu\phi(\boldsymbol{r}) \tag{5.13}$$

The similarity with the KS equations have lead to the misconception that the problem can be easily solved to self-consistency by any Kohn-Sham solver by only minor modifications[69]. More recent studies, however, have shown the opposite, both in the context of the usual atomic GTOs[70] and in a real-space numerical basis[71]. The claim is that the kinetic energy is to non-quadratic for a straightforward iterative optimization, and that more robust techniques are required, like the one presented by Jiang *et al.* [72].

5.3 Preliminary results

In the following we will attempt to solve the OF-DFT Euler equation (5.9) in the multiwavelet framework using a modified form of the KS-DFT solver that is presented in publication III. The iterative procedure is based on the one-orbital formulation given in Eq. (5.13), and thus relies on the von Weizäcker kinetic energy functional. The single orbital is normalized to the number of electrons $\langle\phi|\phi\rangle = N$, so that the density is given as

$$\rho(\boldsymbol{r}) = |\phi(\boldsymbol{r})|^2 \tag{5.14}$$

Also appearing in the equation is the usual nuclear and electronic potentials

$$v_{nuc}(\boldsymbol{r}) = \sum_I \frac{Z_I}{\|\boldsymbol{r} - \boldsymbol{R}_I\|} \tag{5.15}$$

$$v_{el}(\boldsymbol{r}) = \int \frac{\rho(\boldsymbol{r}')}{\|\boldsymbol{r} - \boldsymbol{r}'\|} \, \mathrm{d}\boldsymbol{r}' \tag{5.16}$$

where the singularities in the nuclear potential have been smoothed out as described in publication III, originally introduced by Harrison *et al.* [42]. As v_{xc} we choose the simple Dirac exchange functional presented above in Eq. (5.3) with no correlation treatment, which gives the potential

$$v_{xc}(\boldsymbol{r}) = \frac{\delta E_x[\rho]}{\delta\rho} = -\left(\frac{3}{\pi}\right)^{1/3} \rho^{1/3}(\boldsymbol{r}) \tag{5.17}$$

and we perform calculations both in the Dirac-vonWeizäcker (DvW) model, where the Pauli term is zero $T_\theta = 0$, and in the Thomas-Fermi-Dirac-vonWeizäcker (TFDvW) model, where the Pauli term is chosen as the Thomas-Fermi kinetic functional given in Eq. (5.2), giving a purely repulsive potential

$$v_\theta(\boldsymbol{r}) = \frac{\delta T_\theta[\rho]}{\delta\rho} = \frac{1}{2}\left(3\pi^2\right)^{2/3} \rho^{2/3}(\boldsymbol{r}) \tag{5.18}$$

The results (chemical potential and total energy) of such calculations are presented in Tab. 5.1, where the total energies are compared to conventional (spin-restricted) KS-DFT calculations, using the same Dirac exchange, as well as (spin-restricted) Hartree-Fock energies, taken from Karasiev and Trickey[71] and Chan *et al.* [70], respectively (The Hartree-Fock energies presented in Ref.[70] are actually calculations taken from an old reference, Clementi and Roetti[73]).

As can be seen from Tab. 5.1, we are able to reach self-consistent solutions that agree with previously reported numbers for small systems. All calculations

Table 5.1: Chemical potentials and total energies of atoms and small molecules using the Dirac-von-Weizäcker (DvW), Thomas-Fermi-Dirac-von-Weizäcker (TFDvW) OF-DFT models, and in spin-restricted KS-DFT using the Dirac exchange functional (LDA) as well as spin-restricted Hartree-Fock (RHF).

		Chemical potential		Total energy (Hartree)			
		DvW	TFDvW	DvW	TFDvW	LDA	RHF
H	MRChem	-0.194320	-0.071640	-0.406534	-0.261826	-0.406534	-0.500000
	Ref.[70]		-0.071		-0.2618		-0.5000
	Ref.[71]	-0.1943	-0.0715	-0.406534	-0.261827	-0.4065	
He	MRChem	-0.516991	-0.108327	-2.723640	-1.477451		
	Ref.[70]		-0.108		-1.4775		-2.8617
	Ref.[71]					-2.7236	
Li	MRChem	-0.957510	-0.130656	-8.525825	-4.105425		
	Ref.[70]		-0.131		-4.1054		-7.4327
	Ref.[71]	-0.9575	-0.1306	-8.525825	-4.105425	-7.1749	
Be	MRChem	-1.510360	-0.145379	-19.352891	-8.492186		
	Ref.[70]		-0.145		-8.4922		-14.5730
	Ref.[71]					-14.2233	
B	MRChem	-2.172342	-0.155706	-36.729140	-14.925883		
	Ref.[70]		-0.156		-14.9258		-24.5291
	Ref.[71]					-24.5275	
C	MRChem	-2.941311	-0.163319	-62.169552	-23.656875		
	Ref.[70]		-0.163		-23.6568		-37.6886
	Ref.[71]					-37.6863	
N	MRChem	-3.815709	-0.169164	-97.182735	-34.908435		
	Ref.[70]		-0.169		-34.9084		-54.4009
	Ref.[71]					-54.3977	
O	MRChem	-4.794343	-0.173804	-143.272616	-48.883228		
	Ref.[70]		-0.174		-48.8831		-74.8094
	Ref.[71]					-74.8076	
F	MRChem	-5.876263	-0.177591	-201.939506	-65.767584		
	Ref.[70]		-0.178		-65.7674		-99.4094
	Ref.[71]					-99.4072	
Ne	MRChem	-7.060692	-0.180760	-274.680827	-85.734479	-127.490748	-128.547101
	Ref.[70]		-0.181		-85.7343		-128.5471
	Ref.[71]	-7.0607	-0.1807	-274.68080	-85.734451	-127.4907	
H_2	MRChem	-0.331330	-0.100168	-1.043736	-0.430723	-1.043736	-1.133619
BH	MRChem	-1.170066	-0.146852	-38.589138	-15.301851	-24.629804	-25.131640

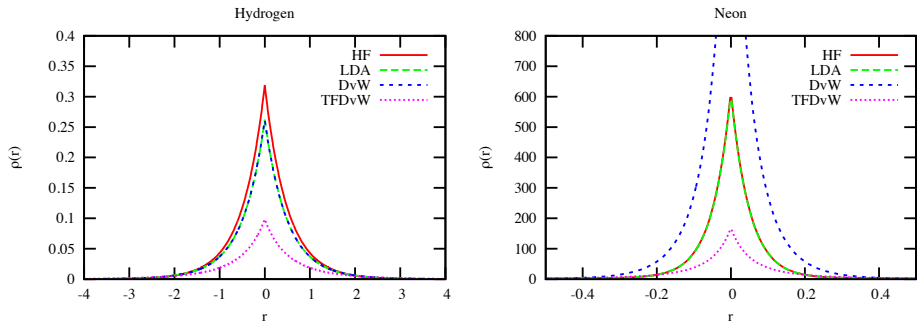

Figure 1: Density plots of hydrogen and neon atoms, calculated at different levels of theory.

were performed using a 9th order multiwavelet basis with an relative accuracy threshold of $\epsilon = 10^{-6}$, and converged to a residual norm of $\|\phi^{n+1} - \phi^n\| < 10^{-6}$, which means that the presented numbers should be correct to six significant digits. We observe, in accord with the claims of Chan and Karasiev, that the optimization is non-trivial, in particular when the Thomas-Fermi (TF) potential is included, and we were unable to reach convergence for bigger systems than the ones presented within a reasonable number of iterations.

Without the TF potential, however, we observe similar convergence as for a single-orbital KS-DFT calculation, and all the presented calculations reached the desired accuracy in about 10 iterations, starting from a random Gaussian density, but it seems that things get more complicated when more nuclear sites are introduced, as for instance the benzene molecule did not converge from a similar poor starting point. As already mentioned, the inclusion of the purely repulsive TF term makes convergence much more problematic, and only the hydrogen atom converged straightforwardly. In all other calculations the TF term had to be introduced gradually. By introducing a TF parameter α and writing the effective potential as

$$v_{eff} = \alpha v_\theta + v_{nuc} + v_{el} + v_{xc} \tag{5.19}$$

we were able to converge the many-electron systems in many intermediate steps, where for instance one could start with $\alpha = 0.20$ and converge to 10^{-2}, and then add five per cent TF ($\Delta\alpha = 0.05$), converge again to 10^{-2}, add another five per cent, and so on until the full TFDvW is reached. This, of course, requires a

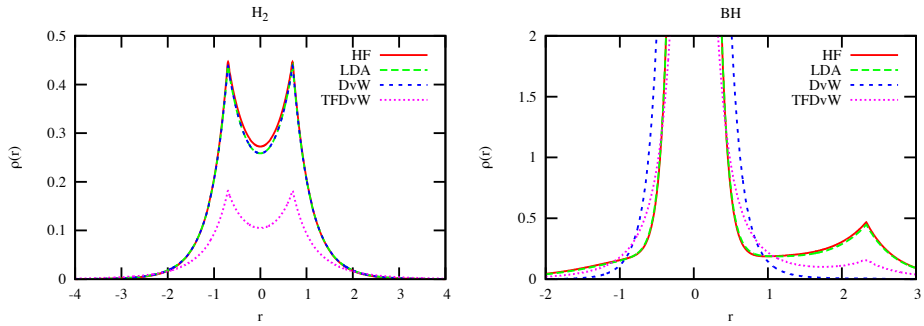

Figure 2: Density plots of H_2 and BH molecules, calculated at different levels of theory.

lot of iterations, and the bigger the system, the more sensitive it is to the TF potential, and consequently, a smaller $\Delta\alpha$ is required. Introduce the TF too fast, and the solution blows up and diverges. For instance, the neon energy was obtained using $\Delta\alpha = 0.005$, and required more than 600 iterations. However, no attempt was made to optimize the parameters on this respect.

If we examine the physics of these models we see that both DvW and TFDvW fail to reproduce the Hartree-Fock energies, even qualitatively. As mentioned above, the von Weizäcker functional is exact for one-orbital systems, which means that DvW model is identical to LDA for the hydrogen and helium atoms, as well as the hydrogen molecule, as can be seen from the numbers. The same is observed in the density plots in Figs. 1 and 2, where we can see the radial density of the hydrogen and neon atoms in Fig. 1, and the density along the internuclear axis of the H_2 and BH molecules in Fig. 2.

From this we can conclude, as is already well established, that the Thomas-Fermi-Dirac-von-Weizäcker models do not perform well for atomic and molecular systems. It is common to introduce a parameter λ for the von Weizäcker term in order to correct for a known over-estimation for molecular systems

$$T_s = \lambda T_W + T_\theta \tag{5.20}$$

and by adjusting this parameter one can get within a few per cent of the Hartree-Fock energy for the given atomic systems, as is shown by Chan *et al.* [70] using $\lambda = 1/5$. However, this parameter is not universal, and the densities that are obtained are not equally accurate (see Ref.[70] for details).

5.4 Outlook

The purpose of this study was not to examine the performance of the given kinetic energy functionals on molecular systems, as their inadequacies in this respect are well known, but rather to see whether the multiresolution framework is appropriate for the solution of the OF-DFT Euler equation. It seems quite clear that its formulation as a one-orbital Kohn-Sham problem is not appropriate, as the convergence of the iterative solution for many-electron systems is problematic at best, as the Thomas-Fermi contribution had to be introduced very carefully to avoid divergence. However, this is related to the mathematical formulation of the problem, and is not specific to the multiwavelet basis. Also, once the full TF potential had been included, high order convergence was not difficult to obtain, and accuracies of 10^{-9} was easily achieved.

Given the properties of the multiwavelet basis, which is easily parallelizable for the few global functions that are involved, and with representations that are free of basis set error, this could still be the ideal framework for the development of better kinetic energy functionals, but this will require much more robust optimization algorithms. This will be subject for further investigation.

Chapter 6

Summary of papers

6.1 Paper I: Adaptive order polynomial algorithm in a multiwavelet representation scheme

In this work, a new strategy is presented for the reduction of the storage requirements of functions in a multiwavelet framework. The work is based on Alpert's[1] definition of the multiwavelet basis which leads to considerable data compression by allowing adaptive refinement of the grid for a given order k of the polynomial basis. We propose an additional adaptivity in the polynomial order, where the order $k(n)$ depends on the refinement level n. We have found that decreasing the order with increasing refinement can lead to considerable reduction in storage requirements for the representations of multivariate functions to a given accuracy.

Stig Rune Jensen wrote the computer implementation of the mathematical formalism presented in the paper, and assisted in running the test calculations. The theory was developed by Antoine Durdek.

6.2 Paper II: Linear scaling Coulomb interaction in the multiwavelet basis, a parallel implementation

The paper describes the implementation of a general Poisson solver in a multiwavelet framework, using the non-standard form of operators. By exploiting the sparsity in the representation of the involved functions and operators, we were able to achieve linear scaling complexity with respect to system size. The performance of the code was demonstrated for molecular systems with up to 600 atoms.

The presented code is based on an implementation of the application of operators in the multiwavelet basis using the non-standard form, written in the C language by Frediani and Fossgaard[5]. The code was completely rewritten in C++ by Stig Rune Jensen and Jonas Jusèlus using a hybrid MPI/OpenMP parallelization strategy. The code, which is called MultiResolution Computational Program Package (MRCPP) is organized as a mathematical library with general features such as function representation and non-standard operator application in multiple dimensions. Jensen also planned and ran all test calculations and wrote parts of the manuscript.

6.3 Paper III: Real-Space Density Functional Theory with Localized Orbitals and Multiwavelets

We present algorithms for the minimization of the Hartree-Fock and Kohn-Sham energies for many-electron molecular systems. The general non-canonical HF/KS equations are rewritten in integral form and solved in the multiwavelet framework using localized orbitals. Robust and fast convergence is demonstrated for small and medium sized systems, and high accuracy energies are presented for a variety of small molecules.

Stig Rune Jensen wrote (with contributions from Peter Wind) the computational chemistry program MultiResolution Chemistry (MRChem) based on the MRCPP library, and together with Antoine Durdek, developed the algorithms for the SCF optimization. Jensen also planned and ran all test calculations and wrote parts of the manuscript.

Bibliography

[1] Alpert B. K. A class of bases in l2 for the sparse representation of integral operators. *Siam J. Math. Anal.*, 24:246, 1993.

[2] Mallat S. G. Multiresolution approximations and wavelet orthonormal bases of l2(r). *Transactions of the American Mathematical Society*, 315(1):69–87, 1989.

[3] Daubechies I. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41(7):909–996, 1988.

[4] Keinert F. *Wavelets and Multiwavelets*. Studies in advanced mathematics. Chapman and Hall/CRC, 2004.

[5] Frediani L., Fossgaard E., Flå, and T. Ruud K. Fully adaptive algorithms for multivariate integral equations using the non-standard form and multiwavelets with applications to the poisson and bound-state helmholtz kernels in three dimensions. *Molecular Physics*, 111(9-11):1143–1160, 2013.

[6] Tymczak C. J., Niklasson A. M. N., and Röder H. Separable and nonseparable multiwavelets in multiple dimensions. *J. Comput. Phys.*, 175(2):363–397, 2002.

[7] Beylkin G. On the fast algorithm for multiplication of functions in the wavelet bases. In *Proc. Int. Conf., Wavelets and Applications*, pages 259–273. Editions Frontiers, 1992.

[8] Beylkin G., Coifman R., and Rokhlin V. Fast wavelet transforms and numerical algorithms i. *Comm. Pure Appl. Math.*, 44(2):141–183, 1991.

[9] Alpert B. K., Beylkin G., Gines D., and Vozovoi L. Adaptive solution of partial differential equations in multiwavelet bases. *J. Comput. Physics*, 182(1):149–190, 2002.

[10] Beylkin G. and Mohlenkamp M. J. Numerical operator calculus in higher dimensions. *Proc. Nat. Acad. Sci.*, 99(16):10246, 2002.

[11] Beylkin G. and Mohlenkamp M. J. Algorithms for numerical analysis in high dimensions. *SIAM Journal on Scientific Computing*, 26(6):2133–2159, 2005.

[12] Griebel M., Zumbusch G., and Knapek S. *Tree algorithms for long-range potentials*, volume 5 of *Texts in Computational Science and Engineering*. Springer Berlin Heidelberg, 2007.

[13] Donoho D. L. Interpolating wavelet transform, 1992.

[14] Singer K. The use of gaussian (exponential quadratic) wave functions in molecular problems. i. general formulae for the evaluation of integrals. *Proc. R. Soc. A*, 258(1294):pp. 412–420, 1960.

[15] Harrison R.J., Fann G.I., Yanai T., and Beylkin G. Multiresolution quantum chemistry in multiwavelet bases. In *Lecture Notes in Computer Science*, volume 2660, pages 103–110. Springer, Heidelberg, 2003.

[16] Fann G., Beylkin G., Harrison R. J., and Jordan K. E. Singular operators in multiwavelet bases. *IBM J. Res. Dev.*, 48(2):161–171, 2004.

[17] Beylkin G., Cheruvu V., and Pérez F. Fast adaptive algorithms in the non-standard form for multidimensional problems. *Appl. and Comput. Harmonic Analysis*, 24(3):354–377, 2008.

[18] Szabo A. and Ostlund N. S. *Modern Quantum Chemistry*. Dover Publications, Inc., 1982.

[19] Parr R. G. and Yang W. *Density-Functional Theory of Atoms and Molecules*. Oxford University Press, 1989.

[20] Jensen F. *Introduction to Computational Chemistry, 2nd edition*. Wiley, 2007.

[21] Losilla S. Numerical methods for electronic structure calculations. *Ph.D. thesis, Uni. of Helsinki*, 2013.

[22] Born M. and Oppenheimer R. Zur quantentheorie der molekeln. *Ann. Phys.*, 389(20):457–484, 1927.

[23] Rychlewski J. *Explicitly correlated wave functions in chemistry and physics: Theory and applications*, volume 13. Springer, 2003.

[24] Kong L., Bischoff F. A., and Valeev E. F. Explicitly correlated r12/f12 methods for electronic structure. *Chem. Rev.*, 112(1):75–107, 2012.

[25] Pauli W. ber den zusammenhang des abschlusses der elektronengruppen im atom mit der komplexstruktur der spektren. *Z. Physik*, 31(1):765–783, 1925.

[26] Slater J. C. The theory of complex spectra. *Phys. Rev.*, 34:1293–1322, 1929.

[27] Helgaker T., Jørgensen P., and Olsen J. *Molecular Electronic-Structure Theory*. Wiley, 2000.

[28] Hohenberg P. and Kohn W. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, 1964.

[29] Kohn W. and Sham L. J. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133–A1138, 1965.

[30] Burke K. Perspective on density functional theory. *J. Chem. Phys.*, 136:150901, 2012.

[31] Kato T. On the eigenfunctions of many-particle systems in quantum mechanics. *Comm. Pure Appl. Math.*, 10(2):151–177, 1957.

[32] Slater J. C. Atomic shielding constants. *Phys. Rev.*, 36(1):57, 1930.

[33] S Francis Boys. Electronic wave functions. i. a general method of calculation for the stationary states of any molecular system. *Proc. R. Soc. A.*, 200(1063):542–554, 1950.

[34] Laaksonen L., Pyykkö P., and Sundholm D. Two-dimensional fully numerical solutions of molecular schrödinger equations. i. one-electron molecules. *Int. J. Q. Chem.*, 23(1):309–317, 1983.

[35] Laaksonen L., Pyykkö P., and Sundholm D. Two-dimensional fully numerical solutions of molecular schrödinger equations. ii. solution of the poisson equation and results for singlet states of h2 and heh+. *Int. J. Q. Chem.*, 23(1):319–323, 1983.

[36] Laaksonen L., Pyykkö P., and Sundholm D. Two-dimensional fully numerical solutions of molecular hartree-fock equations: Lih and bh. *Chem. Phys. Lett.*, 96(1):1–3, 1983.

[37] Kobus J., Laaksonen L., and Sundholm D. A numerical hartree-fock program for diatomic molecules. *Comp. Phys. Comm.*, 98(3):346–358, 1996.

[38] Kurashige Y., Nakajima T., and Hirao K. Gaussian and finite-element coulomb method for the fast evaluation of coulomb integrals. *J. Chem. Phys.*, 126:144106, 2007.

[39] Watson M. A., Kurashige Y., Nakajima T., and Hirao K. Linear-scaling multipole-accelerated gaussian and finite-element coulomb method. *J. Chem. Phys.*, 128:054105, 2008.

[40] Kurashige Y., Nakajima T., Sato T., and Hirao K. Efficient evaluation of the coulomb force in the gaussian and finite-element coulomb method. *J. Chem. Phys.*, 132:244107, 2010.

[41] Losilla S. A. and Sundholm D. A divide and conquer real-space approach for all-electron molecular electrostatic potentials and interaction energies. *J. Chem. Phys.*, 136(21), 2012.

[42] Harrison R. J., Fann G. I, Yanai T., Gan Z., and Beylkin G. Multiresolution quantum chemistry: Basic theory and initial applications. *J. Chem. Phys.*, 121:11587, 2004.

[43] Yanai T., Fann G. I., Gan Z., Harrison R. J., and Beylkin G. Multiresolution quantum chemistry in multiwavelet bases: Hartree–fock exchange. *J. Chem. Phys.*, 121(14):6680–6688, 2004.

[44] Yanai T., Fann G. I., Gan Z., Harrison R. J., and Beylkin G. Multiresolution quantum chemistry in multiwavelet bases: Analytic derivatives for hartree–fock and density functional theory. *J. Chem. Phys.*, 121(7):2866–2876, 2004.

[45] Yanai T., Harrison R. J., and Handy N. C. Multiresolution quantum chemistry in multiwavelet bases: time-dependent density functional theory with asymptotically corrected potentials in local density and generalized gradient approximations. *Mol. Phys.*, 103(2-3):413–424, 2005.

[46] Roothaan C. C. J. New developments in molecular orbital theory. *Rev. Mod. Phys.*, 23:69–89, 1951.

[47] Hall G. G. The molecular orbital theory of chemical valency. viii. a method of calculating ionization potentials. *Proc. R. Soc. A.*, 205(1083):541–552, 1951.

[48] Pulay P. Convergence acceleration of iterative sequences. the case of scf iteration. *Chem. Phys. Lett.*, 73(2):393–398, 1980.

[49] Kalos M.H. Monte carlo calculations of the ground state of three-and four-body nuclei. *Phys. Rev.*, 128(4):1791, 1962.

[50] Schneider R., Rohwedder T., Neelov A., and Blauert J. Direct minimization for calculating invariant subspaces in density functional computations of the electronic structure. *arXiv preprint arXiv:0805.1190*, 2008.

[51] Harrison R. J. Krylov subspace accelerated inexact newton method for linear and nonlinear equations. *J. Comput. Chem.*, 25(3):328–334, 2004.

[52] Goedecker S. Linear scaling electronic structure methods. *Rev. Mod. Phys.*, 71:1085–1123, 1999.

[53] Goedecker S. and Scuserza G. E. Linear scaling electronic structure methods in chemistry and physics. *Comp. Sci. En.*, 5(4):14–21, 2003.

[54] Watson M. A., Saek P., Macak P., and Helgaker T. Linear-scaling formation of kohn-sham hamiltonian: Application to the calculation of excitation energies and polarizabilities of large molecular systems. *J. Chem. Phys.*, 121(7), 2004.

[55] Saek P., Hst S., Thgersen L., Jrgensen P., Manninen P., Olsen J., Jansk B., Reine S., Pawowski F., Tellgren E., Helgaker T., and Coriani S. Linear-scaling implementation of molecular electronic self-consistent field theory. *J. Chem. Phys.*, 126(11), 2007.

[56] Thomas L.H. The calculation of atomic fields. *Proc. Camb. Phil. Soc.*, 23:542, 1927.

[57] Fermi E. Un metodo statistico per la determinazione di alcune priopriet dell'atomo. *Rend. Accad. Naz. Lincei*, 6:602–607, 1927.

[58] Dirac P.A.M. Quantum mechanics of many-electron systems. *R. Soc. London Proc. A*, 123:714–733, 1929.

[59] Teller E. On the stability of molecules in the thomas-fermi theory. *Rev. Mod. Phys.*, 34:627–631, 1962.

[60] von Weizscker C.F. Zur theorie der kernmassen. *Z. fr Physik*, 96(7-8):431–458, 1935.

[61] Wang L. W. and Teter M. P. Kinetic-energy functional of the electron density. *Phys. Rev. B*, 45:13196–13220, 1992.

[62] Karasiev V. V., Jones R. S., Trickey S. B., and Harris F. E. Recent advances in developing orbital-free kinetic energy functionals. *New Dev. Q. Chem.*, pages 25–54, 2009.

[63] Borgoo A. and Tozer D. J. Density scaling of noninteracting kinetic energy functionals. *J. Chem. Theory and Comp.*, 9(5):2250–2255, 2013.

[64] Xia J., Huang C., Shin I., and Carter E. A. Can orbital-free density functional theory simulate molecules? *The Journal of Chemical Physics*, 136(8), 2012.

[65] Hung L. and Carter E. A. Accurate simulations of metals at the mesoscale: Explicit treatment of 1 million atoms with quantum mechanics. *Chem. Phys. Lett.*, 475:163–170, 2009.

[66] Huang C. and Carter E. A. Nonlocal orbital-free kinetic energy density functional for semiconductors. *Phys. Rev. B*, 81:045206, 2010.

[67] Garcia-Cervera C. J. An efficient real space method for orbital-free density-functional theory. *Commun. Comput. Phys*, 2:334–357, 2007.

[68] Gavini V., Knap J., Bhattacharya K., and Ortiz M. Non-periodic finite-element formulation of orbital-free density functional theory. *J. Mech. Phys. Sol.*, 55(4):669 – 696, 2007.

[69] Levy M., Perdew J. P., and Sahni V. Exact differential equation for the density and ionization energy of a many-particle system. *Phys. Rev. A*, 30:2745–2748, 1984.

[70] Chan G. K. L., Cohen A. J., and Handy N. C. Thomasfermidiracvon weizscker models in finite systems. *J. Chem. Phys.*, 114(2):631–638, 2001.

[71] Karasiev V. V. and Trickey S. B. Issues and challenges in orbital-free density functional calculations. *Comp. Phys. Comm.*, 2012.

[72] Jiang H. and Yang W. Conjugate-gradient optimization method for orbital-free density functional calculations. *J. Chem. Phys.*, 121(5):2030–2036, 2004.

[73] Clementi E. and Roetti C. Roothaan-hartree-fock atomic wavefunctions: Basis functions and their coefficients for ground and certain excited states of neutral and ionized atoms. *At. Data Nuc. Data Tables*, 14(34):177 – 478, 1974.

Paper I

Adaptive order polynomial algorithm in a multiwavelet representation scheme

A. Durdek, S. R. Jensen, J. Jusèlius, P. Wind, T. Flå and L. Frediani

Submitted to Applied Numerical Mathematics

Paper II

Linear scaling Coulomb interaction in the multiwavelet basis, a parallel implementation

S. R. Jensen, J. Jusèlius, A. Durdek P. Wind, T. Flå and L. Frediani
Submitted to Int. Journal of Modeling, Simulation and Scientific Computing

Paper III

Real-Space Density Functional Theory with Localized Orbitals and Multiwavelets

S. R. Jensen, J. Jusèlius, A. Durdek P. Wind, T. Flå and L. Frediani

Manuscript in preparation