UiT
THE ARCTIC
UNIVERSITY
OF NORWAY

Faculty of Science and Technology

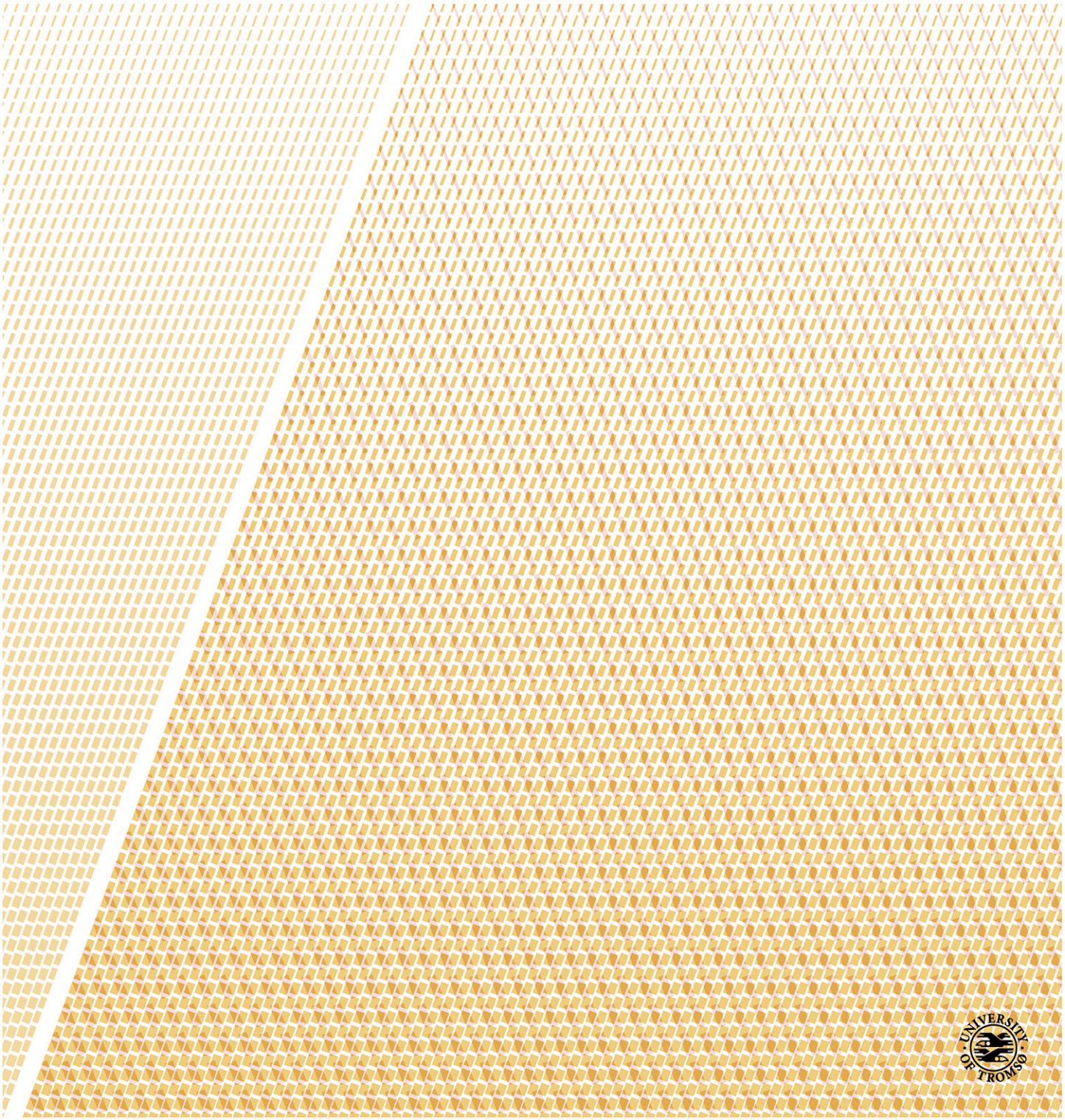
Department of Physics and Technology

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

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

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)

$$\begin{aligned}
{}^nA^{n'} &\stackrel{\text{def}}{=} \quad Q_k^n T Q_k^{n'} : \quad W_k^{n'} \to W_k^n \\
{}^nB^{n'} &\stackrel{\text{def}}{=} \quad Q_k^n T P_k^{n'} : \quad V_k^{n'} \to W_k^n \\
{}^nC^{n'} &\stackrel{\text{def}}{=} \quad P_k^n T Q_k^{n'} : \quad W_k^{n'} \to V_k^n \\
{}^nT^{n'} &\stackrel{\text{def}}{=} \quad P_k^n T P_k^{n'} : \quad V_k^{n'} \to V_k^n
\end{aligned} \tag{2.48}$$

leading to the relation

$${}^{n+1}T^{n+1} = \; {}^nT^n \; + \; {}^nC^n \; + \; {}^nB^n \; + \; {}^nA^n \tag{2.49}$$

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

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

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 $\boldsymbol{\tau}_{lm}^n$ 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

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

$$\left[\beta_{lm}^n\right]_{ij} = \int\int K(x,y)\psi_{i,l}^n(x)\phi_{j,m}^n(y)\,\mathrm{d}x\,\mathrm{d}y \tag{2.67}$$

$$\left[\gamma_{lm}^n\right]_{ij} = \int\int K(x,y)\phi_{i,l}^n(x)\psi_{j,m}^n(y)\,\mathrm{d}x\,\mathrm{d}y \tag{2.68}$$

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

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

$${}^nB^n f^n(x) = \sum_{l,m}\sum_{i,j}\left[\beta_{lm}^n\right]_{ij}\psi_{i,l}^n(x)s_{j,m}^{n,f} \tag{2.70}$$

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

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

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

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

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

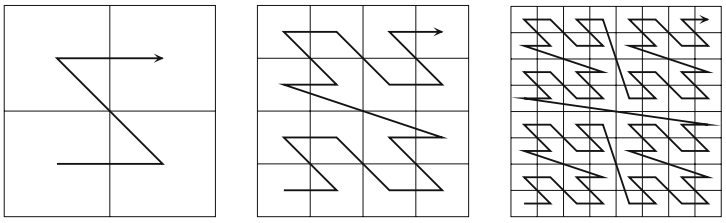


Figure 1: Three refinement levels in the construction of the Lebesgue curve in 2D.

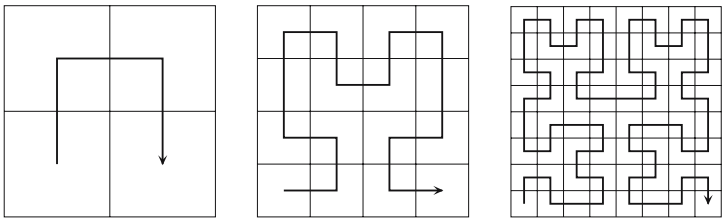


Figure 2: Three refinement levels in the construction of the Hilbert curve in 2D.

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

```
1: create tree skeleton of empty nodes
2: MPI: distribute leaf nodes among hosts through Hilbert path
3: MPI: create list of local nodes owned by this host
4: while number of local nodes on current iteration N_i > 0 do
5:    OpenMP: divide local nodes among available processors
6:    for each node at current iteration do
7:       compute scaling and wavelet coefficients
8:       if node needs to be refined then
9:          mark node as non-terminal
10:         allocate children nodes
11:         update list of local nodes for next iteration
12:      else
13:         mark node as terminal
14:      end if
15:   end for
16:   increment iteration
17: 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-

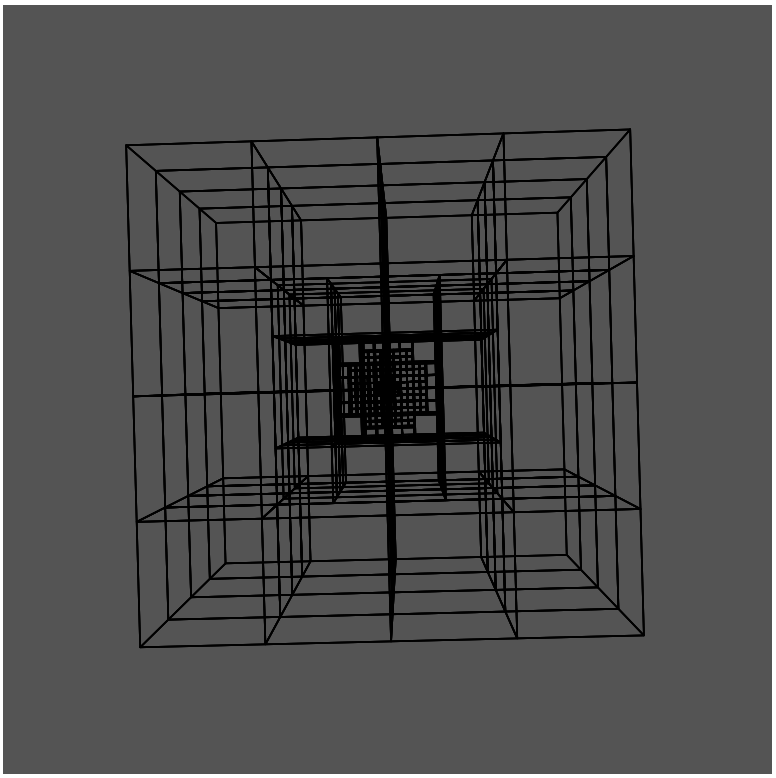


Figure 3: Adaptive grid-partitioning of the unit cube needed to reproduce the Gaussian function $f(\boldsymbol{x}) = (\beta/\pi)^{3/2} e^{-\beta(\boldsymbol{x}-\boldsymbol{x}_0)^2}$ with exponent $\beta = 500$ in position $\boldsymbol{x}_0 = (1/2, 1/2, 1/2)$ to a relative accuracy of $\epsilon = 10^{-8}$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

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

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

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

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 $\left(\|O_{\kappa,\boldsymbol{l}-m}^{\alpha\beta,n}\|_2 \cdot \|\boldsymbol{w}_{\boldsymbol{l}}^{\alpha,n,f}\|\right)$
8: **for** each contributing input `node` **do**
9: apply operator $\boldsymbol{w}_{\boldsymbol{l}}^{\alpha,n,g} += \bigotimes_{p=1}^{d} O_{\kappa,\boldsymbol{l}-m}^{\alpha_p\beta_p,n} \boldsymbol{w}_{\boldsymbol{m}}^{\beta_p,n,f}$
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

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

$$\sum_l \sum_i \tilde{w}^{n,g}_{i,l}\psi^n_{i,l}(x) = \sum_{l,m}\sum_{i,j}\Big(\big[\beta^n_{lm}\big]_{ij} s^{n,f}_{j,m} + \big[\alpha^n_{lm}\big]_{ij} w^{n,f}_{j,m}\Big)\psi^n_{i,l}(x) \tag{3.64}$$

$$\tilde{w}^{n,g}_{i,l} = \sum_m \sum_j \Big(\big[\beta^n_{lm}\big]_{ij} s^{n,f}_{j,m} + \big[\alpha^n_{lm}\big]_{ij} w^{n,f}_{j,m}\Big) \tag{3.65}$$

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)

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

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

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.

- ***E**fficiency*

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

- ***S**ystematicity*

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

- ***U**niversality*

  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 structure 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 $\varepsilon >$ threshold **do**
  Construct Helmholtz operator $\hat{H}^{\mu^n}$ using $\mu^n = \sqrt{-2E^n}$
  Multiply wave function ψ^n with potential
  Apply Helmholtz operator Eq.(4.87) and normalize
  Calculate wave function update $\Delta\psi^n = \psi^{n+1} - \psi^n$
  Calculate wave function error $\varepsilon = \|\Delta\psi^n\|$
  Calculate energy update ΔE^n from Eq.(4.96)
  Add $\left(\psi^n, E^n\right)$ and $\left(\Delta\psi^n, \Delta E^n\right)$ to KAIN history
  Calculate KAIN updates $\left(\delta\tilde{\psi}^n, \delta E^n\right)$ from Eq.(4.101)
  Update wave function $\tilde{\psi}^{n+1} = \psi^n + \delta\tilde{\psi}^n$ and normalize
  Update energy $E^{n+1} = E^n + \delta E^n$
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

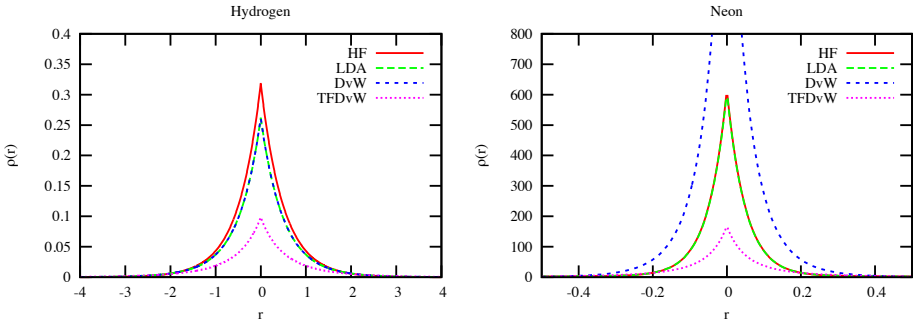


Figure 1: Density plots of hydrogen and neon atoms, calculated at different levels of theory.

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

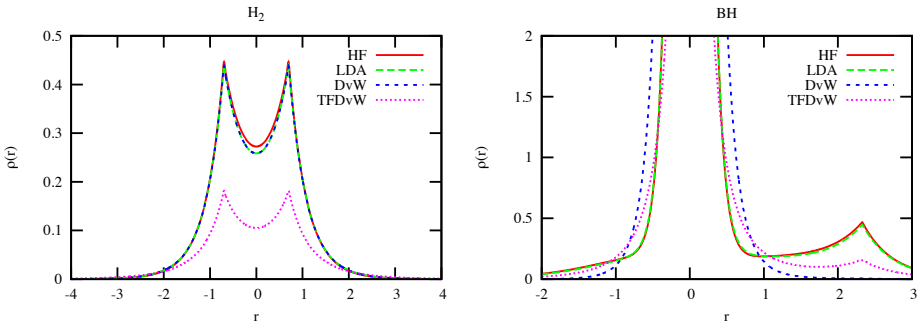


Figure 2: Density plots of H_2 and BH molecules, calculated at different levels of theory.

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

Adaptive order polynomial algorithm in a multiwavelet representation scheme

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

Abstract

We have developed a new strategy to reduce the storage requirements of a multivariate function in a multiwavelet framework. We propose that alongside the commonly used adaptivity in the grid refinement one can also vary the order of the representation k as a function of the scale n. In particular the order is decreased with increasing refinement scale. The consequences of this choice, in particular with respect to the nesting of scaling spaces, are discussed and the error of the approximation introduced is analyzed. The application of this method to some examples of mono- and multivariate functions shows that our algorithm is able to yield a storage reduction up to almost 60%. In general, values between 30 and 40% can be expected for multivariate functions. Monovariate functions are less affected but are also much less critical in view of the so called "curse of dimensionality".

1 Introduction

Kohn–Sham DFT has proven to be a computationally cost-effective approach for both the theoretical modeling of molecules and for the modeling of extended, periodic systems [1]. Recently, linear-scaling based approaches have gradually been removing the boundaries between these two extremes[2, 3]. In current computational chemistry, the Kohn–Sham orbitals are for molecules in most cases represented in terms of basis sets consisting of Gaussian functions. The molecular orbitals $\psi_i(\mathbf{r})$ are written as a linear combination of Gaussians:

$$\psi_i(\mathbf{r}) = \sum_\mu C_{i\mu}\chi_\mu(\mathbf{r}_K) = \sum_\mu C_{i\mu}P_\mu(\mathbf{r}_K)\exp\left(-\alpha_\mu r_K^2\right) \tag{1}$$

where the expansion coefficients $C_{i\mu}$ are referred to as molecular orbital coefficients, and where we have indicated that the electronic coordinates are given relatively to the nuclear center K to which the Gaussian basis function is attached. $P_\mu(\mathbf{r}_K)$ denotes a Cartesian polynomial $x_K^i y_K^j z_K^k$. In principle the atomic basis set should be complete, thus infinite, but for practical reasons it is generally restricted to a few tens of functions for each atom in the molecule.

For extended periodic systems, the most convenient approach is the representation in terms of Gaussian plane waves [1, 4] which easily exploits the periodicity of the system and allows the fast evaluation of the molecular integrals:

$$\psi_i(\mathbf{r}) = \sum_{\mathbf{k}} C_{i\mathbf{k}}\exp(i\mathbf{k}\mathbf{r}) \tag{2}$$

where $\mathbf{k}$ is a three-dimensional wave vector.

Both approaches are somewhat inadequate when facing the challenge of modeling a large system which can be partitioned into a molecular subsystem and one or more extended or periodic structures. One would therefore like a separated representation that has approximate, algorithmic size-extensivity in the sense of a local and hierarchical scale adaptivity. More generally, finer approximations could be used in subunits of crucial importance for the molecular system at hand. For large molecules we believe a modular approach is essential to reflect the importance of the different subsystems for the quantum molecular problem under scrutiny.

A step in this direction is taken by allowing different meshes in regions of space as in multigrid [5] and multiresolution[6] techniques. Multiresolution analysis may be employed to provide a sparse and efficient representation of both operators and functions in that it allows a description of the system at different scales of resolution. Wavelet bases provide important properties for designing efficient numerical solution techniques: orthogonality, vanishing moments and compact support. The latter, which is particularly important in high dimension, enables a locally adaptive representation of functions: the grid is refined only where the current representation is not sufficient to reach the required precision in the computed results, thus yielding the coarsest grid compatible with the desired numerical precision of the result.

One important candidate multiscale method is the Multiwavelet basis which has been used by Harrison *et al.* [7, 8, 9], to represent Kohn-Sham molecular orbitals.

By making use of this approach we have in our group performed extensive tests to verify the linear scaling capabilities of the approach with respect to the system size[10] and of the ability to control the error within an arbitrary and predefined value.[11, 10]. In both cases very good results have been achieved.

The main drawback of such a grid based approach compared to traditional ones based on Gaussian functions or plane waves is the large memory requirement associated with such methods: as no explicit functional form is assumed, the storage requirements for each function is very large, reaching several gigabytes if high precision is requested. The problem can be partially addressed by parallelization, thereby exploiting distributed memory architectures. A complementary strategy to address the problem is to reduce the memory footprint of each function. One such method has recently been proposed by Bischoff and coworkers[12, 13] who employed a rank-reduction based on Singular Value Decomposition.

In this paper, we will follow an alternative route to reducing the prefactor for the memory storage problem. We propose to make the order of the polynomial basis scale-dependent: $k = k(n)$. In particular, k will decrease with the grid refinement. The underlying assumption is that higher order polynomials are less important at finer scales to correctly represent cusp-like functions such as those needed to deal with molecular orbitals. It is instead more important to increase the grid refinement. Since the support of the basis is the same as for a fixed basis, the basis functions supported on different hypercubes will still be non-overlapping and therefore orthogonal. As will be shown Section 3, the main challenge posed by this approach is the lack of orthogonality between the scaling space V_k^n and the wavelet space $W_{k'}^n$ with $k' < k$. We have dealt with this problem by proposing an approximated representation. The algorithms necessary to construct it are given in Sec. 4 whereas a set of numerical tests is presented in Sec. 5 and discussed in Sec. 6.

2 Multiwavelet representation in 1D

Alpert was the first to describe the multiwavelet approach for the representation of functions and operators [14][15]. His work is based on his description of Legendre scaling functions and the corresponding wavelet functions. In order to set the notations for Section 3, we briefly review here the main ideas. Let us define the scaling spaces V_k^n as:

$$V_k^n = \mathrm{Span}\{\phi_{ij}^n | i = 0, \cdots, k,\, j = 0, \cdots, 2^n - 1\} \tag{3}$$

where $\phi_i(x)$ is the i-th compressed and translated Legendre polynomial on the interval $[0, 1]$:

$$\phi_i(x) = \begin{cases} \sqrt{2i+1} L_i(2x-1) & x \in [0,1] \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

and ϕ_{il}^n is the compressed and translated i-th scaling function on the interval $[2^{-n}l, 2^{-n}(l+1)]$ by $\phi_{il}^n(x) = 2^{n/2}\phi_i(2^n x - l)$.

Legendre polynomials are chosen as a basis as they are obtained in a recursive manner and are orthonormal with respect to the scalar product

$$\langle f, g \rangle = \int_0^1 f(x) g(x)\, \mathrm{d}x \tag{5}$$

Moreover, the Legendre polynomial $\phi_i(x)$ has degree i which implies that polynomial basis of order $k' < k$ which span $V_{k'}$ is a subset of the Legendre basis spanning V_k. We will largely exploit this in the next section: in order to change the order of the representation, one simply has to add or remove one or more basis functions keeping the other ones as they are.

By definition of the scaling spaces, one gets directly that :

$$V_k^0 \subset V_k^1 \subset \cdots \subset V_k^n \subset \cdots \tag{6}$$

and V_k^n is the space of piecewise polynomial functions of degree less or equal to k on $(2^{-n}l, 2^{-n}(l+1))$ for $0 \leq l < 2^n$. The number of basis functions at scale n is $\dim V_k^n = (k+1)2^n$.

The wavelet spaces W_k^n are defined as the orthogonal complement of V_k^n with respect to V_k^{n+1}:

$$W_k^n \oplus V_k^n = V_k^{n+1}, \forall n \tag{7}$$

which implies that $\dim W_k^n = (k+1)2^n$. If $\psi_i, i = 0, \cdots, k$ are the basis functions of W_k^0, then we have the following properties for the basis of W_k^n:

1. ψ_i is built as a piecewise polynomial function with a discontinuity in the middle of the interval since $\psi_i \in V_k^1$ and $\psi_i \notin V_k^0$.
2. $\psi_{il}^n(x) = 2^{n/2}\psi_i(2^n x - l)$
3. $\langle \phi_{il}^n | \psi_{jm}^{n'} \rangle = 0 \quad (n' \leq n)$
4. $\langle \psi_{il}^n | \psi_{jm}^{n'} \rangle = \delta_{nn'}\delta_{ij}\delta_{lm}$

The freedom in the choice of basis functions for the wavelet space can be exploited by requiring additional properties. According to[16] it is possible to construct a basis such that:

1. ψ_i has $i + k$ vanishing moments
2. ψ_i is an odd (even) function with respect to inversion through the interval center x=0.5 for even (odd) values of i.

According to Equation 7, one can describe a linear unitary transformation between the two bases via a matrix transformation, which collects the four filter matrices $G^{(0)}$, $G^{(1)}$, $H^{(0)}$ and $H^{(1)}$:

$$\begin{pmatrix} \psi_l^n \\ \phi_l^n \end{pmatrix} = \begin{pmatrix} G^{(1)} & G^{(0)} \\ H^{(1)} & H^{(0)} \end{pmatrix} \begin{pmatrix} \phi_{2l+1}^{n+1} \\ \phi_{2l}^{n+1} \end{pmatrix} \tag{8}$$

The transformation is unitary since it is a change of basis between two orthonormal bases. The inverse transformation is consequently straightforward. The transformation is also scale-independent, since $G^{(1)}, G^{(0)}, H^{(1)}, H^{(0)}$ are the same for all scales on each subdivision. Additionally we note that, thanks to the nested construction of the Legendre polynomials, the H filter matrices are also nested: they are built from the upper left corner and increasing or decreasing the degree of the polynomial basis translates into to adding or removing the last columns and rows.

As shown by Alpert *et al.* [15], the use of polynomials as scaling functions is based on the following theorem:

Theorem 1. *Let V_k^n be a scaling space described as above with polynomials as scaling functions on the interval $[0, 1]$.*
Thus we have the following result:

1. $\lim_{k\to\infty} V_k^n = L^2([0,1])$
2. $\lim_{n\to\infty} V_k^n = L^2([0,1])$

The theorem shows that completeness in the L_2 norm sense can be achieved both by increasing the polynomial order and by refinement of the dyadic subdivisions along the ladder of scales.

For any function $f \in L^2$, the projected function $\mathcal{P}_k^n f = f_k^n$ of f on V_k^n can be written as:

$$f_k^n = \sum_{j=0}^{2^n-1} \sum_{i=0}^{k} f_{ij}^n \phi_{ij}^n \tag{9}$$

$$\text{where } f_{ij}^n = \langle f | \phi_{ij}^n \rangle \tag{10}$$

which is the so called "reconstructed representation". f can also be projected on the ladder of wavelet spaces to get the "compressed representation":

$$f_k^n = f_k^0 + \sum_{m=0}^{n-1} df_k^m \tag{11}$$

$$= \sum_{i=0}^{k} f_i \phi_i + \sum_{m=0}^{n-1} \sum_{j=0}^{2^m-1} \sum_{i=0}^{k} df_{ij}^m \psi_{ij}^m \tag{12}$$

$$\text{where } f_i = \langle f | \phi_i \rangle \tag{13}$$

$$\text{and } df_{ij}^m = \langle f | \psi_{ij}^m \rangle \tag{14}$$

The two representations are equivalent and can be interconverted in one another by recursive application of the two-scale relation:

$$f_k^n + df_k^n = f_k^{n+1} \tag{15}$$

The error committed by projecting the function onto V_k^n is fully controlled and can be computed [17][18]. The accuracy is set as a parameter and the approximation can be done arbitrarily close to the true function via scale refinement and variation on the order.

It is also useful to introduce a projector notation. If we indicate $\mathcal{P}_k^n$ and $\mathcal{Q}_k^n$ the projector onto V_k^n and W_k^n respectively. It then follows that

$$\mathcal{P}_k^n + \mathcal{Q}_k^n = \mathcal{P}_k^{n+1} \tag{16}$$

For $k' < k$ we will also define a residual projector $\mathcal{P}_{k,k'}^n$as

$$\mathcal{P}_{k,k'}^n = \mathcal{P}_k^n - \mathcal{P}_{k'}^n \tag{17}$$

By definition of the wavelet projectors, and the previous relations the following relations can be easily proven:

$$\mathcal{Q}_k^n \mathcal{P}_k^n = \mathcal{P}_k^n \mathcal{Q}_k^n = \mathcal{Q}_k^n \mathcal{P}_{k'}^n = \mathcal{P}_{k'}^n \mathcal{Q}_k^n = \mathcal{Q}_k^n \mathcal{P}_{k,k'}^n = \mathcal{P}_{k,k'}^n \mathcal{Q}_k^n = 0 \tag{18}$$

$$\mathcal{Q}_{k'}^n \mathcal{P}_k^n = \mathcal{Q}_{k'}^n \mathcal{P}_{k,k'}^n \tag{19}$$

$$\mathcal{P}_k^n \mathcal{Q}_{k'}^n = \mathcal{P}_{k,k'}^n \mathcal{Q}_{k'}^n \tag{20}$$

$$\mathcal{P}_k^n \mathcal{P}_{k'}^n = \mathcal{P}_{k'}^n \mathcal{P}_k^n = \mathcal{P}_{k'}^n \tag{21}$$

As a corollary of the completeness theorem, the following relations can be written for the projection operators:

$$\lim_{k\to\infty} \mathcal{P}_k^n = \lim_{n\to\infty} \mathcal{P}_k^n = \mathbf{I} \tag{22}$$

$$\lim_{k\to\infty} \mathcal{Q}_k^n = \lim_{n\to\infty} \mathcal{Q}_k^n = 0 \tag{23}$$

3 Adaptive polynomial order representation

The representation of a multivariate function f at scale n in d dimensions with a tensorial multiwavelet basis of order k requires $2^{nd}(k+1)^d$ coefficients for the reconstructed representation at scale n. The accuracy of the representation can be increased either by augmenting the polynomial basis (larger k) or by further refinements (larger n), thus increasing drastically the data storage. In order to limit the memory requirement adaptivity is introduced, thereby refining the representation only where the predefined accuracy is not met.

We propose an additional way to reduce the data storage. Namely, instead of keeping the same polynomial order k at all scales we will assume that k can be chosen as a function of n with the limitation that $k(n) \leq k(n')$ for $n > n'$. Especially in high dimension, this could determine a reduction of the data storage requirements.

The challenging point of this approach is represented by the loss of exact inclusion of the vector space $V^n_{k(n)}$ into $V^{n+1}_{k(n+1)}$:

$$V^n_{k(n)} \subsetneq V^{n+1}_{k(n+1)} \text{ unless } k(n+1) = k(n) \tag{24}$$

Let us define $V^n_{\Delta k}$ implicitly as:

$$V^n_{k(n)} \stackrel{\text{def}}{=} V^n_{k(n+1)} \oplus V^n_{\Delta k} \tag{25}$$

$V^n_{\Delta k}$ is the subspace of $V^n_{k(n)}$ which is not strictly contained in V^{n+1}. However $V^{n+1}_{k(n+1)}$ can be employed to approximate a function belonging to $V^n_{\Delta k}$. More specifically, since

$$V^{n+1}_{k(n+1)} = V^n_{k(n+1)} \oplus W^n_{k(n+1)} \tag{26}$$

then $V^n_{\Delta k}$ can be approximated by a corresponding subspace in $W^n_{k(n+1)}$. As an example let us consider V_3 and $V_2 \oplus W_2$. The cubic function in V_3 is orthogonal to V_2 but can be approximated as a piecewise quadratic function which belongs to W_2.

We have the following theorem for any polynomial of order k:

Theorem 2. *Let V_k be the scaling space of order k, V_{k-1} the scaling space of order $k-1$. Let $\mathcal{P}^n_k$, $\mathcal{P}^n_{k-1}$, $\mathcal{Q}^n_{k-1}$ be the projectors onto V^n_k, V^n_{k-1} and W^n_{k-1} respectively. Let us define:*

$$d^n_k = \sup_f \frac{||(1-\mathcal{Q}^n_{k-1})\mathcal{P}^n_{k,k-1}f||_{L^2}}{||\mathcal{P}^n_{k,k-1}f||_{L^2}} \tag{27}$$

where $f \in L^2[0,1]$ such that $f^{(p)}$ is defined for all $p \leq k$. Then

1. $\lim_{k\to\infty} d^n_k = 0$
2. *d^n_k is decaying exponentially with k.*

To put it simply, the theorem states that the norm of the component of $\mathcal{P}^n_{k,k-1}f$ which falls outside W^n_{k-1} decays exponentially with increasing k.

Proof. We assume, without loss of generality that $n = 0$. The result comes from the fact that truncated Legendre series converges with an exponential decay for finite support functions [19, 20]. By writing the projection of f onto V^k_0 as

$$\mathcal{P}^0_k f = \sum_{i=0}^{k} c_k \phi^0_k, \qquad c_i = \left\langle f, \phi^0_i \right\rangle \tag{28}$$

and substituting into Eq. (27) one gets:

$$\begin{aligned} d^0_k =& \frac{||c_k\phi^0_k - \mathcal{Q}^0_{k-1}c_k\phi^0_k||_{L^2}}{||c_k\phi^0_k||_{L^2}} = ||\phi^0_k - \mathcal{Q}^0_{k-1}\phi^0_k||_{L^2} = \\ & ||\phi^0_k - (\mathcal{P}^1_{k-1} - \mathcal{P}^0_{k-1})\phi^0_k||_{L^2} = ||(I - \mathcal{P}^1_{k-1})\phi^0_k||_{L^2}. \end{aligned} \tag{29}$$

The first step follows from the normalization condition of the basis, and the last one one is due to the orthogonality of ϕ_k^0 with respect to V_{k-1}^0.

As shown by Alpert [14], $\mathcal{P}_k^n$ converges exponentially to the identity of $L_2[0,1]$:

$$||(I-\mathcal{P}_k^n)f|| \leq 2^{-nk}\frac{2}{4^k\cdot k!}\sup|f^{(k)}(x)|. \tag{30}$$

In particular, for $f=\phi_k^0$, we can compute explicitly $\sup|\phi_k^{0,(k-1)}(x)|$. The leading term in the Legendre polynomial $P_k(x)$ is $\binom{2k}{k}\frac{1}{2^k}x^k+O(x^k-2)$. The $(k-1)$-th derivative of $P_k(x)$ is therefore:

$$\frac{\mathrm{d}^{k-1}}{\mathrm{d}x^{k-1}}P_k(x)=\binom{2k}{k}\frac{k!}{2^k}x \tag{31}$$

Recalling that $\phi_k^0(x)=\sqrt{2k+1}P_k(2x-1)$ we obtain

$$\sup|\phi_k^{0,(k-1)}(x)|=\sqrt{2k+1}\binom{2k}{k}\frac{k!}{2} \tag{32}$$

By making use of the Stirling bounds on $n!$:

$$\sqrt{2\pi}\ n^{n+1/2}e^{-n}\leq n!\leq e\ n^{n+1/2}e^{-n}$$

one finally gets:

$$||(I-\mathcal{P}_{k-1}^1)\phi_k^0||_{L^2}\leq\frac{4e}{\pi}\sqrt{2k(2k+1)}2^{-k}<\frac{4e}{\pi}(2k+1)2^{-k} \tag{33}$$

which shows the exponential decay with increasing k and proves the theorem. □

This result can be generalized to any fixed $\Delta k=k-k'\geq 1$ In particular, for large enough k:

$$\mathcal{P}_{k,k'}^n\simeq\mathcal{P}_{k,k'}^n\mathcal{Q}_{k'}^n\simeq\mathcal{Q}_{k'}^n\mathcal{P}_{k,k'}^n \tag{34}$$

In other words, the space spanned by $V_{k,k'}^n$ becomes almost collinear with a corresponding subspace of $W_{k'}^n$.

In Figure 1 we have collected d_k^1 for $k=1,3\ldots 18$. The error decays exponentially with the polynomial order as expected. In particular, we observe that d_k^1 scales as 2^{-k} and well within the given error bound.

3.1 Projection onto $V_{k(n)}^n$ and $W_{k(n+1)}^n$

The projection step consists in the computation of the function representation in the ladder of scaling and wavelet spaces. More in detail for each scale n, the projection $f_k^n=\mathcal{P}_k^n f$ can for instance be obtained via a quadrature scheme.

The wavelet component $df_{k'}^n$is obtained by noticing that:

$$f_{k'}^{n+1}=f_{k'}^n+df_{k'}^n \tag{35}$$

For the sake of brevity we have assumed that $k=k(n)$ and $k'=k(n+1)$.

In this way we obtain at each scale a scaling part f_k^n and a wavelet part $df_{k'}^n$. We underline here that the two components are not orthogonal as $W_{k'}^n$ is only orthogonal to the first k' polynomial of V_k^n.

The projection down to the finest scale requires only the knowledge of $k(n)$ for each scale n starting from a predefined maximum value $k_{max}=k(0)$ until a minimum value $k_{min}=k(n_{min})$. Thereafter the polynomial order is kept constant at $k=k_{min}$

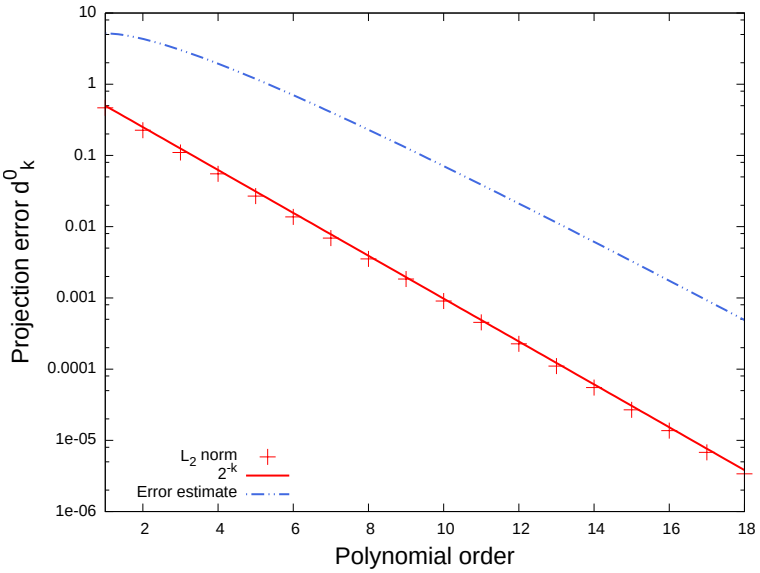


Figure 1: Representation of d_k^0 as a function of the polynomial order. The dots represent the L_2 norm of the error computed applying Eq. (29). The curve is the error estimate according to Eq. (33).

3.2 Reconstruction: $V_{k(n)}^n + W_{k(n+1)}^n \to V_{k(n+1)}^{n+1}$

The reconstruction step consists in obtaining the scaling representation at the finest scale by making use of the scaling component at the coarsest scale $f_{k(0)}^0$ and the ladder of wavelet components $df_{k(n)}^n$. Assuming again $k = k(n)$ and $k' = k(n+1)$, the reconstruction step at each scale can be achieved by the following procedure.

First the polynomial part of f_k^n from $k'+1$ to k is projected out:

$$f_{k'}^n = (1 - \mathcal{P}_{k,k'}^n) f_k^n = \mathcal{P}_{k'}^n f_k^n \tag{36}$$

then the scaling representation $f_{k'}^{n+1}$ is obtained by assembling:

$$f_{k'}^{n+1} = f_{k'}^n + df_{k'}^n \tag{37}$$

The procedure is repeated iteratively, scale by scale along the tree structure. As there is no overlap between neighboring nodes the iteration is carried on until a local finest scale, which is determined by the precision requirements.

3.3 Analysis: $V_{k(n+1)}^{n+1} \to V_{k(n)}^n + W_{k(n+1)}^n$

The analysis or compression step is the inverse transformation of the reconstruction, in the sense that it consists in obtaining the scaling component at scale $n = 0$ and the wavelet components at all scales from the reconstructed representation f_k^n at the finest scale. This is achieved iteratively, starting at the finest scale. The difference with respect to the standard algorithm is represented by the fact that, given a representation of f in $V_{k'}^{n+1}$ we want to obtain a representation in V_k^n where $k > k'$.

The first step consists in transforming $f_{k'}^{n+1}$ into the corresponding wavelet and scaling components at scale n:

$$f_{k'}^{n+1} = f_{k'}^n + df_{k'}^n \tag{38}$$

The second step consists in "transferring" the component of $df_{k'}^n$ which is collinear to V_k^n to the scaling part in an approximate way by making use of Theorem 2:

$$\begin{aligned} f_{k'}^n + df_{k'}^n &= \mathcal{P}_{k'}^n f + \mathcal{Q}_{k'}^n f \\ &= \mathcal{P}_{k'}^n f + (1 - \mathcal{P}_{k,k'}^n + \mathcal{P}_{k,k'}^n)\mathcal{Q}_{k'}^n f \\ &\simeq \mathcal{P}_{k'}^n f + \mathcal{P}_{k,k'}^n f + (1 - \mathcal{P}_{k,k'}^n)\mathcal{Q}_{k'}^n f \\ &= \mathcal{P}_k^n f + (1 - \mathcal{P}_{k,k'}^n)\mathcal{Q}_{k'}^n f = f_k^n + d\tilde{f}_{k'}^n \end{aligned} \tag{39}$$

In the last step we have implicitly defined $d\tilde{f}_k^n = (1 - \mathcal{P}_{k,k'}^n)\mathcal{Q}_{k'}^n f$.

In this way the scheme to achieve an approximate representation of f on V_k^n based on the representation in $V_{k'}^{n+1}$ is complete. Repeating this procedure iteratively from $n = n_{max}$ to $n = 0$ leads to a representation of f onto $V_{k(0)}^0 \oplus W_{k(1)}^0 \oplus \cdots \oplus W_{k(n_{max})}^{n_{max}-1}$.

3.4 Multivariate functions

For multivariate functions a tensor product representation is employed. The projector onto the scaling space at each scale is:

$$\mathcal{P}_k^n = \bigotimes_{i=1}^d \mathcal{P}_k^{n,i} \tag{40}$$

whereas the projector onto the wavelet space is obtained as the difference between two successive scales:

$$\mathcal{Q}_k^n \stackrel{\text{def}}{=} \mathcal{P}_k^{n+1} - \mathcal{P}_k^n = \bigotimes_{i=1}^d \mathcal{P}_k^{n+1,i} - \bigotimes_{i=1}^d \mathcal{P}_k^{n,i} \tag{41}$$

Similarly, we can define the residual projector as:

$$\mathcal{P}_{k,k'}^n \stackrel{\text{def}}{=} \mathcal{P}_k^n - \mathcal{P}_{k'}^n = \bigotimes_{i=1}^d \mathcal{P}_k^{n,i} - \bigotimes_{i=1}^d \mathcal{P}_{k'}^{n,i} \tag{42}$$

As for the monovariate case we can write the approximate relationship (34) which can be derived from the monovariate case by exploiting the tensor product structure:

$$\begin{aligned} \mathcal{Q}_{k'}^n \mathcal{P}_{k,k'}^n &= \left(\bigotimes_{i=1}^d \mathcal{P}_{k'}^{n+1,i} - \bigotimes_{i=1}^d \mathcal{P}_{k'}^{n,i} \right) \left(\bigotimes_{i=1}^d \mathcal{P}_k^{n,i} - \bigotimes_{i=1}^d \mathcal{P}_{k'}^{n,i} \right) \\ &= \bigotimes_{i=1}^d \mathcal{P}_{k'}^{n+1,i}\mathcal{P}_k^{n,i} - \bigotimes_{i=1}^d \mathcal{P}_{k'}^{n+1,i}\mathcal{P}_{k'}^{n,i} - \bigotimes_{i=1}^d \mathcal{P}_{k'}^{n,i}\mathcal{P}_k^{n,i} + \bigotimes_{i=1}^d \mathcal{P}_{k'}^{n,i}\mathcal{P}_{k'}^{n,i} \\ &= \bigotimes_{i=1}^d \mathcal{P}_{k'}^{n+1,i}\mathcal{P}_k^{n,i} - \bigotimes_{i=1}^d \mathcal{P}_{k'}^{n,i} = \bigotimes_{i=1}^d \left(\mathcal{P}_{k'}^{n,i} + \mathcal{Q}_{k'}^{n,i} \right) \mathcal{P}_k^{n,i} - \bigotimes_{i=1}^d \mathcal{P}_{k'}^{n,i} \\ &= \bigotimes_{i=1}^d \left(\mathcal{P}_{k'}^{n,i} + \mathcal{Q}_{k'}^{n,i}\mathcal{P}_k^{n,i} \right) - \bigotimes_{i=1}^d \mathcal{P}_{k'}^{n,i} \\ &\simeq \bigotimes_{i=1}^d \left(\mathcal{P}_{k'}^{n,i} + \mathcal{P}_{k,k'}^{n,i}\mathcal{P}_k^{n,i} \right) - \bigotimes_{i=1}^d \mathcal{P}_{k'}^{n,i} = \bigotimes_{i=1}^d \mathcal{P}_k^{n,i} - \bigotimes_{i=1}^d \mathcal{P}_{k'}^{n,i} = \mathcal{P}_{k,k'}^n \end{aligned} \tag{43}$$

We further underline that in the multivariate case, when the polynomial order is reduced from k to k' the number of components which need to be discarded as described in Sec. 3.2 or transferred from $W_{k'}^n$ to V_k^n as described in Sec. 3.3 is now $(k+1)^d - (k'+1)^d$: in other words it is the difference between the d-dimensional hypercube of length $k+1$ and the one of length $k'+1$ (e.g. for $d = 3$ and $k' = k - 1$ the number of discarded/transferred components is $3k^2 + 3k + 1$).

4 Algorithms

In this section, we present the details of our algorithm. Legendre basis functions are used for scaling functions: thanks to the construction of Legendre polynomials, only one scaling function is involved in the process. The construction of the wavelet basis [16] with additional vanishing moments is directly linked to the non-orthogonality between high order polynomial and the wavelet basis. In the simplest case where the polynomial order $k(n)$ is lowered by one at each successive scale, only the first wavelet function ψ_0 is not orthogonal to ϕ_k. All other inner products are zero by construction. E.g. for $k = 3$, this is equivalent to approximating the cubic function $\phi_k(x)$ by the piecewise polynomial $\psi_0(x)$ which is made of two adjacent parabolas, supported respectively on $[0, 1/2]$ and $[1/2, 1]$. Increasing the order will, as proved in Theorem 2, lead to better and better approximation. Additionally we stress that, in practice one only needs to "move" one projection coefficient: the coefficient representing the projection onto ψ_0 will instead be used for the projection onto ϕ_k or vice versa. This means that there is no additional loss of information or deterioration of the representation by performing successive reconstructions and compressions.

Algorithm 1 illustrates the projection of a function employing our adaptive scheme. At each scale, starting from the coarsest one the scaling part of the function $f^n_{k(n),l}$ is computed. Then the wavelet part $df^n_{k(n+1),l}$ is computed by switching to the polynomial basis $k(n+1)$. The wavelet norm is then checked against the required precision to determine whether refinement is necessary.

Algorithm 1 Adaptive projection algorithm for a function f with a given accuracy ε

```
For each scale n
    For each available node l at the current scale
        Compute f^n_{k(n),l}
        Compute df^n_{k(n+1),l}
        If(||df^n_{k(n+1),l}|| > 2^n ε)
        allocate child nodes and mw-transform coefficients
    next node
next scale
```

Algorithm 2 describes the compression of a function: it is here assumed that the function is represented at the local finest scale as $f^n_{k(n)}$ and all child nodes are present to reconstruct the parent. Starting at the next finest scale $n = n_{max} - 1$, the scaling part $f^n_{k(n+1)}$ and the wavelet part $df^n_{k(n+1)}$ of each node are obtained from its children through a standard Multiwavelet (MW) transform. If $k(n) > k(n+1)$, the scaling part is augmented to $f^n_{k(n)}$ by making use of Eq. (39) and the wavelet part is correspondingly purged. In practice thanks to the Alpert construction of the basis set, this implies that one or more coefficients are simply transferred from the wavelet to the scaling part. The sequence is repeated for all nodes at the current scale n before moving to scale $n - 1$.

Algorithm 2 Compression algorithm

```
For each scale from n = n_max − 1 to n = 0
    For each node l at the current scale
        Obtain f^n_{k(n+1)} and df^n_{k(n+1)} from f^{n+1}_{k(n+1)}
        If (k(n) > k(n+1))
            Transform f^n_{k(n+1)} + df^n_{k(n+1)} into f^n_{k(n)} + dfˆ^n_{k(n+1)}
     next node
previous scale
```

Algorithm 3 shows the reconstruction of the finest-scale representation of a function. Such a function is represented through $f^0_{k(0)}$ plus the modified wavelet part at each scale $d\tilde{f}^n_{k(n+1)}$. Starting

at the coarsest scale $n = 0$ the correct scaling and wavelet components $f^n_{k(n+1)}$ and $df^n_{k(n+1)}$ are obtained by making use of Eq. (39) if $k(n) > k(n+1)$. As for the compression algorithm this implies that one or more coefficients are simply transferred, this time from the scaling to the wavelet part. The scaling representation of the child nodes $f^n_{k(n+1)}$ is then obtained by a MW-transform. The sequence is repeated for all nodes at the current scale n before moving to scale $n+1$.

Algorithm 3 Reconstruction algorithm

```
01 For each scale from n = 0 to n = n_max − 1
02     For each node l at the current scale
03         If (k(n) > k(n+1))
04             Transform f^n_{k(n)} + df̃^n_{k(n+1)} into f^n_{k(n+1)} + df^n_{k(n+1)}
05         Compute f^{n+1}_{k(n+1)} from f^n_{k(n+1)} and df^n_{k(n+1)}
06      next node
07 previous scale
```

5 Numerical results

In order to test the effectiveness of our approach we have selected some test functions and we have compared the amount of memory required to represent them on the one hand by making use of a regular MW-representation for a given polynomial order k and a given accuracy ϵ, and on the other hand with our decreasing order approach.

The chosen functions are Gaussian functions and so-called Slater type orbitals ($f(x) = Ae^{(-\alpha|x-x_0|)}$) which display a cusp-like singularity for $x = x_0$. Both examples are mutated from quantum chemistry as the former is the most widespread choice to build a basis set, whereas the latter is nowadays less common but has the appropriate behavior: a cusp at the atomic center and exponential asymptotic decay for large distances.

The parameterization employed for $k(n)$ is shown in Fig. 2. The polynomial order is kept fixed at k_{max} from $n = 0$ to a given n_0. It is then decreased by one at each successive scale up to n_1 and finally kept constant for all successive scales at $k_{min} = k_{max} - (n_1 - n_0)$. This strategy has been chosen to be able to adjust the range of scales where the order reduction takes place, keeping at the same time the structure as simple as possible.

Table 1 and Table 2 collect the results for two one-dimensional Gaussians with exponents $\alpha = 50$ and $\alpha = 10000$ respectively. For each of them we have reported the number of coefficients required to represent the function with the standard MW-representation and polynomial order k_{max} and with decreasing order scheme. The parameterization of $k(n)$ is also reported through the values of k_{min} (minimum allowed order) and n_0 (starting scale for order reduction). Our results show that a reduction of the size of the representation can be achieved in most cases by the appropriate choice of $k(n)$. In a few cases no reduction is possible indicating that the parameterization provided by the standard MW-representation is already optimal.

The results collected for the two three-dimensional Gaussians are reported in Table 3 and Table 4, respectively. By comparison with the results obtained in the one-dimensional case, an enhancement of the compression achieved with a decreasing-order scheme can be observed. In particular the following remarks can be made: (1) the reduction of the number of coefficients needed for the representation can be achieved in all cases tested, (2) the compression achieved is consistently larger than for the monovariate case; (3) the decreasing order scheme has a stronger impact on the narrow Gaussian (large exponent α), which is also the one requiring a larger representation.

The achieved compression expressed as percent reduction of the size of the representation for the Gaussian functions of Tables 1, 2, 3 and 4 is also reported in Fig. 3.

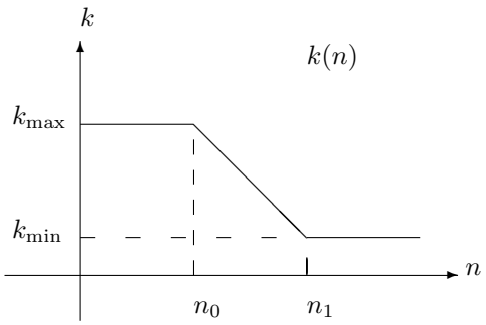


Figure 2: Generic shape of the order $k(n)$

k_{max}	SR	DOR	k_{min}	n_0	%
5	180	180	5	0	0
6	210	182	5	0	-13,33
7	176	168	5	2	-4,55
8	126	126	8	0	0
9	140	128	8	0	-8,57
10	154	120	8	0	-12,99
11	168	148	8	0	-11,90
12	182	162	8	0	-10,99
13	84	84	13	0	0
14	90	86	13	0	-4,44
15	96	92	13	0	-4,17

Table 1: Comparison of standard MW-representation (SR) with the decreasing-order representation (DOR) for a centered one-dimensional Gaussian function with $\alpha = 50$. The number of coefficients for the two representations (second and third column) is expressed as a function of the initial polynomial order k_{max}. For SR the initial order k_{max} is used throughout whereas for the DOR the function $k(n)$ is equal to k_{max} until $n = n_0$ and then decreased by one at each successive refinement until k_{min} is reached. The last column (%) is expressing the compression achieved as the percent reduction in the representation size in terms of number of coefficients.

Table 5 summarizes the same kind of information for a non-centered one-dimensional Slater-type orbital, with exponent parameter $\alpha = 100$. The function is off-centered in order to avoid the singularity to be on a discretization point and hence take artificially advantage of it. x_0 is set to 0.27. The table contains the number of coefficients required both for the standard representation with a fixed order $k = k_{max}$, and for the corresponding adaptive order representation. Our results highlight a reduction of the total number of coefficients in all cases. We have observed that in most cases the best parameterization is achieved when $k(n)$ is chosen such that k_{min} is reached at the finest scale N.

The results for the off-centered three-dimensional Slater orbital are presented in Table 6. The parameters are $\alpha = 100$ and $x_0 = (0,27; 0,27; 0,27)$. Also in this case, compared to the monodimensional one, a more consistent behavior is observed. Compression is achieved for all choices of initial order k_{max} and a more pronounced compression rate is observed compared to the monovariate case.

The achieved compression expressed as percent reduction of the size of the representation for the Slater-type functions of Tables 5 and 6 is also reported in Fig. 4.

k_{max}	SR	DOR	k_{min}	n_0	%
5	564	564	5	0	0
6	434	434	6	0	0
7	496	436	6	0	-12,10
8	414	414	8	0	0
9	460	416	8	0	-9,57
10	506	422	8	0	-16,60
11	552	436	8	0	-21,01
12	598	458	8	0	-23,41
13	532	488	8	0	-8,27
14	570	526	8	0	-7,72
15	608	572	8	0	-8,92

Table 2: Comparison of standard MW-representation (SR) with the decreasing-order representation (DOR) for a centered one-dimensional Gaussian function with $\alpha = 10000$. The number of coefficients for the two representations (second and third column) is expressed as a function of the initial polynomial order k_{max}. For SR the initial order k_{max} is used throughout whereas for the DOR the function $k(n)$ is equal to k_{max} until $n = n_0$ and then decreased by one at each successive refinement until k_{min} is reached. The last column (%) is expressing the compression achieved as the percent reduction in the representation size in terms of number of coefficients.

k_{max}	SR	DOR	k_{min}	n_0	%
5	568512	568512	5	0	0
6	375928	310904	5	2	-17,30
7	561152	323072	5	1	-42,43
8	425736	324808	5	0	-23,71
9	584000	427904	8	0	-26,73
10	777304	447896	8	0	-42,38
11	1009152	611008	9	0	-39,45
12	1283048	809640	9	0	-36,90
13	197568	197568	13	0	0
14	243000	202616	13	0	-16,62
15	294912	248768	14	0	-15,65

Table 3: Comparison of standard MW-representation (SR) with the decreasing-order representation (DOR) for a centered three-dimensional Gaussian function with $\alpha = 50$. The number of coefficients for the two representations (second and third column) is expressed as a function of the initial polynomial order k_{max}. For SR the initial order k_{max} is used throughout whereas for the DOR the function $k(n)$ is equal to k_{max} until $n = n_0$ and then decreased by one at each successive refinement until k_{min} is reached. The last column (%) is expressing the compression achieved as the percent reduction in the representation size in terms of number of coefficients.

k_{max}	SR	DOR	k_{min}	n_0	%
5	1453248	1266880	4	7	-12,8
6	1605240	1601144	4	6	-0,26
7	1609728	1523200	4	6	-5,38
8	1918728	1611464	7	0	-16,01
9	2632000	1627520	7	0	-38,16
10	3503192	1758616	7	0	-49,80
11	4548096	2032832	7	0	-55,30
12	5782504	2441384	8	0	-57,78
13	5817280	2987264	8	0	-48,65
14	7155000	3778936	8	0	-47,18
15	8683520	4856768	9	0	-44,07

Table 4: Comparison of standard MW-representation (SR) with the decreasing-order representation (DOR) for a centered three-dimensional Gaussian function with $\alpha = 100$. The number of coefficients for the two representations (second and third column) is expressed as a function of the initial polynomial order k_{max}. For SR the initial order k_{max} is used throughout whereas for the DOR the function $k(n)$ is equal to k_{max} until $n = n_0$ and then decreased by one at each successive refinement until k_{min} is reached. The last column (%) is expressing the compression achieved as the percent reduction in the representation size in terms of number of coefficients.

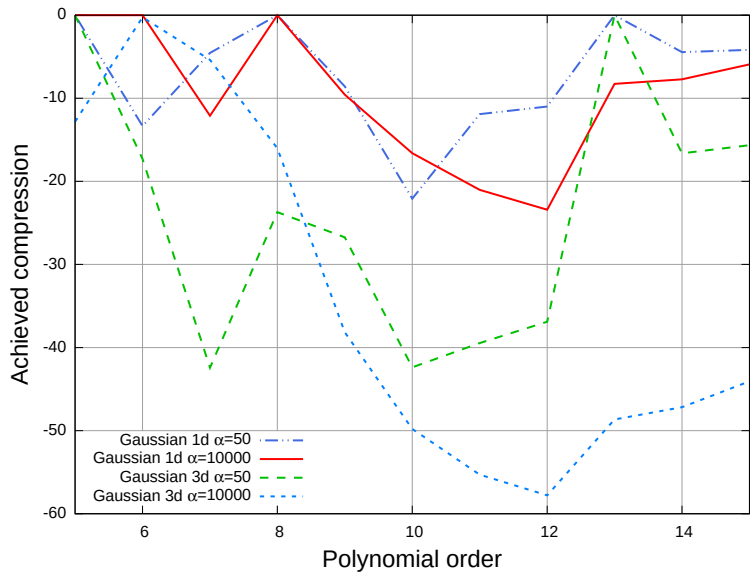


Figure 3: Percentage of coefficients gain in function of the order k_{max} for the Gaussian-type function in the one- and three-dimensional case and $\alpha = 50, 10000$. The data corresponds to the last column of the corresponding Tables.

k_{max}	SR	DOR	k_{min}	n_0	%
5	792	792	5	0	0
6	840	796	5	0	-5,24
7	832	740	5	7	-11,06
8	936	776	5	6	-17,09
9	960	752	5	6	-21,67
10	1056	780	5	5	-26,14
11	1152	800	5	4	-30,56
12	1196	816	5	3	-31,77
13	1176	824	5	1	-29,93
14	1320	828	5	0	-37,27
15	1344	840	5	0	-38,65

Table 5: Comparison of standard MW-representation (SR) with the decreasing-order representation (DOR) for a off-centered one-dimensional Slater function with $\alpha = 100$. The number of coefficients for the two representations (second and third column) is expressed as a function of the initial polynomial order k_{max}. For SR the initial order k_{max} is used throughout whereas for the DOR the function $k(n)$ is equal to k_{max} until $n = n_0$ and then decreased by one at each successive refinement until k_{min} is reached. The last column (%) is expressing the compression achieved as the percent reduction in the representation size in terms of number of coefficients.

k_{max}	SR	DOR	k_{min}	n_0	%
5	2004481	2004481	5	0	0
6	2129344	2012608	5	0	-5,48
7	2195456	2054268	5	0	-6,43
8	2472768	2091456	5	0	-13,75
9	3008000	2174464	5	0	-27,71
10	4174016	2216000	6	0	-46,91
11	4091904	2367872	6	0	-42,13
12	4921280	2640064	6	0	-46,35
13	5795328	2679168	5	0	-53,77
14	7128000	3049152	5	0	-57,22
15	8650752	3706496	5	0	-57,15

Table 6: Comparison of standard MW-representation (SR) with the decreasing-order representation (DOR) for a off-centered three-dimensional Slater function with $\alpha = 100$. The number of coefficients for the two representations (second and third column) is expressed as a function of the initial polynomial order k_{max}. For SR the initial order k_{max} is used throughout whereas for the DOR the function $k(n)$ is equal to k_{max} until $n = n_0$ and then decreased by one at each successive refinement until k_{min} is reached. The last column (%) is expressing the compression achieved as the percent reduction in the representation size in terms of number of coefficients.

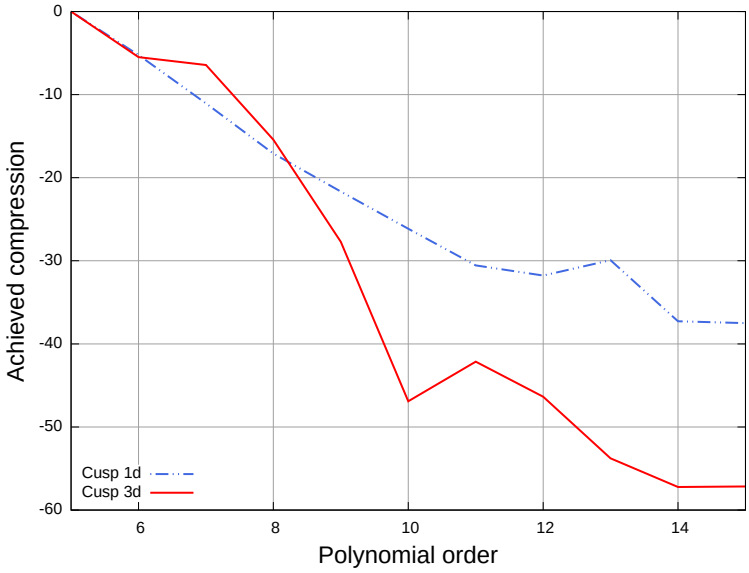


Figure 4: Percentage of coefficients gain in function of the order k_{max} for the the Slater-type function in the one- and three- dimensional case with $\alpha = 100$. The data corresponds to the last column of the corresponding Tables.

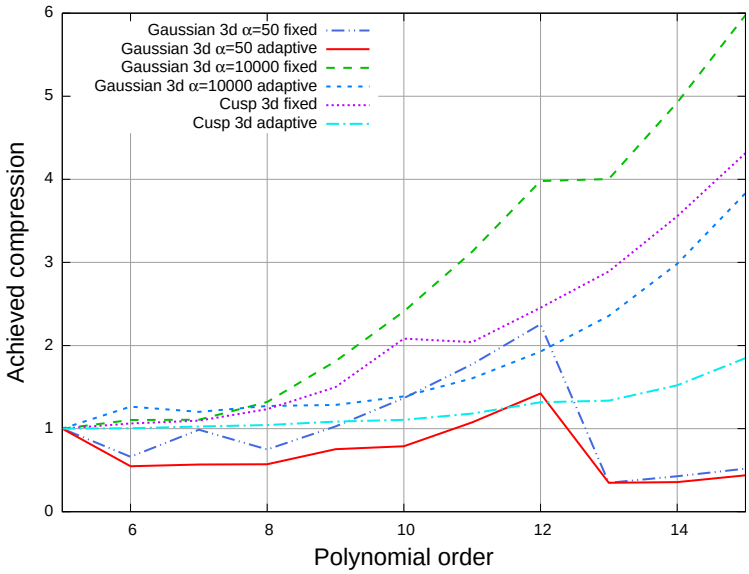


Figure 5: Relative variation on the number of coefficients for the Gaussian type function with $\alpha = 10000$ and Slater type with $\alpha = 100$ in the three-dimensional case. For the two functions, the SR and the DOR are presented. The relative variation $r(k)$ is obtained with respect to the order $k_{ref} = 5$. Writing $N(k)$, the number of coefficients needed at order k, we compute $r(k)$ as $r(k) = N(k)/N(k_{ref})$ (so that $r(5) = 1$ for any of the representation)

6 Discussion

The numerical results of the previous section, (see for a summary Fig 3 and 4) show that in most cases, a compression of the memory needed to represent a single function can be achieved. Two clear distinctions can be drawn: on the one hand the compression achieved for functions presenting short-scale variations (a Gaussian with a large exponent or a cusp) is more significant; at the same time the effect of compression is clearly more pronounced for a multivariate function than for a monovariate one. The latter consideration is motivated by the the fact that in a standard MW-representation the number of coefficients at scale n is proportional to $(k+1)^{nd}$, therefore the effect of order reduction is amplified. For the least-effective case (a monovariate Gaussian with small exponent, $\alpha = 50$) the representation is however small to start with and the lack of a significant compression is to be expected.

Concerning the parameterization of $k(n)$ (the order k employed at each scale n) we observed that within a certain range, for all the examples shown a certain degree of compression can be achieved. In practice, the parameterization $k_{max} \in [8, 12], k_{min} = 5, n_0 = 0$ leads to a moderate compression for the monovariate functions and 30% or better in the multivariate case.

It is also interesting to observe what happens to the total number of coefficients needed while increasing the order k_{max}. Such data are summarized in Fig. 5 for the multivariate functions. In the standard case, the representation size soon becomes larger with increasing k (the representation of the chosen narrow multivariate Gaussian with $k = 15$ becomes six times larger than the one with $k = 5$) both for the narrow Gaussian and the cusp. The wide Gaussian is however less sensitive to the choice of k until $k = 13$, when a significant reduction is observed. By decreasing the order one sees that the overall size of the representation stays almost constant in the beginning and becomes larger only for $k_{max} = 12$ or larger. In other words, decreasing the order helps in maintaining an optimal degree of compression: smooth and slowly varying functions (Gaussian with $\alpha = 50$) are best represented with large degree polynomials which are able to yield an accurate representation with very few refinements. For high frequency variations (Gaussian with $\alpha = 10000$) and cusps, deep refinement levels are anyway necessary; the order reduction scheme employed here is able to keep the complexity close to optimal values by gradually removing unnecessary degrees of freedom.

We also notice that for the cusp and the narrow Gaussian, when $k_{max} = 12$ or larger, also the decreasing order scheme leads to slightly larger representations, albeit not as large as the standard scheme. We argue that a more aggressive order decrease (e.g. $k(n) = k(n-1) - 2$) could help reduce the complexity in such cases but we have not pursued this route yet.

Another consideration regards the choice of n_0, namely the last scale with order $k = k_{max}$. We have often seen (*cf.* Table 5 on the Cusp-like example) that an optimal representation with the decreased-order approach is obtained when the order k_{min} is reached at the finest scale N. This requirement is however function-dependent and therefore difficult to exploit fully in practical applications one has to use the same $k(n)$ for all functions. This consideration could nevertheless guide the final choice of the order function $k(n)$.

In the future we plan to apply the decreasing order scheme $k(n)$ to the application of operators in the Non-Standard form[21]. The main challenge in this case will be the construction of the components of the operator at each scale. However, as the Non-Standard form virtually decouples scales when the operator is applied (the coupling is afterwards restored by applying the filters to the resulting functions) we believe this to be a feasible prosecution of the present work.

Acknowledgments

This work has been supported by the Research Council of Norway through a Centre of Excellence Grant (Grant No. 179568/V30). This work has received support from the Norwegian Supercomputing Program (NOTUR) through a grant of computer time (Grant No. NN4654K).

References

[1] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos. Iterative minimization techniques for *ab initio* total-energy calculations: molecular dynamics and conjugate gradients. *Rev. Mod. Phys.*, 64:1045–1097, Oct 1992.

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

[3] Konstantin N. Kudin and Gustavo E. Scuseria. Linear-scaling density-functional theory with gaussian orbitals and periodic boundary conditions: Efficient evaluation of energy and forces via the fast multipole method. *Phys. Rev. B*, 61:16440–16453, Jun 2000.

[4] R. Car and M. Parrinello. Unified approach for molecular dynamics and density-functional theory. *Phys. Rev. Lett.*, 55:2471–2474, Nov 1985.

[5] U. Trottenberg, C.W. Oosterlee, and A. Schueller. *Multigrid*. Academic Press, 2001.

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

[7] R.J. Harrison, G.I. Fann, T. Yanai, Z. Ghan, and G. Beylkin. Multiresolution quantum chemistry: Basic theory and initial applications. *Journal of Chemical Physics*, 121(23):11587–11598, 2004.

[8] T. Yanai, G.I. Fann, Z. Ghan, R.J. Harrison, and G. Beylkin. Multiresolution quantum chemistry in multiwavelet bases: Hartree-fock exchange. *Journal of chemical physics*, 121(14):6680–6688, 2004.

[9] R.J. Harrison, G.I. Fann, T. Yanai, and G. Beylkin. *Multiresolution Quantum Chemistry in Multiwavelet Bases*, pages 103–110. Springer, Heidelberg, 2003.

[10] Stig-Rune Jensen, Jonas Juselius, Antoine Durdek, Peter Wind, Tor Flå, and Luca Frediani. Linear scaling coulomb interaction in the multiwavelet basis,
a parallel implementation. Submitted, 2013.

[11] E. Fossgaard, L. Frediani, T. Flå, and K. Ruud. Fast numerical algorithms for applying integral-operators in higher dimensions. *Mol. Phys.*, 2013. accepted.

[12] Florian A Bischoff, Robert J Harrison, and Edward F Valeev. Computing many-body wave functions with guaranteed precision: The first-order Møller-Plesset wave function for the ground state of helium atom. *The Journal of Chemical Physics*, 137(10):104103, September 2012.

[13] Florian A Bischoff and Edward F Valeev. Low-order tensor approximations for electronic wave functions: Hartree-Fock method with guaranteed precision. *The Journal of Chemical Physics*, 134(10):104104–104104–10, March 2011.

[14] B. Alpert. *Sparse representation of smooth linear operators*. PhD thesis, Yale University, Department of Mathematics, 10 Hillhouse Avenue, P.O. Box 208283 New Haven, CT 06520-8283, 1990. Available online at: `http://www.math.yale.edu/pub/papers/`.

[15] B. Alpert, G. Beylkin, D. Gines, and L. Vozovoi. Adaptive solution of partial differential equations in multiwavelet bases. *Journal of computational physics*, 182:149–190, 2002.

[16] B. K. Alpert. A class of bases in L^2 for the sparse representation of integral operators. *Siam J. On Math. Analysis*, 24(1):246–262, 1993.

[17] Fritz Keinert. *Wavelets and multiwavelets*. Studies in Advanced Mathematics. Chapman & Hall/CRC, Boca Raton, FL, 2004.

[18] Gilbert Strang and T. Nguyen. *Wavelets and filter banks*. Wellesley-Cambridge Press, 1997.

[19] Narayan Kovvali. *Theory and Applications of Gaussian Quadrature Methods*. Synthesis Lectures on Algorithms and Software in Engineering. Morgan Claypool Publishers, 2011.

[20] C. Canuto, M. Y. Hussaini, and T. A. Zang. *Spectral methods in fluid dynamics*. Springer-Verlag, 1988.

[21] G. Beylkin and J.M. Keiser. On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases. *Journal of Computational Physics*, 132:233–259, 1997.

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

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

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

Abstract

We present a parallel and linear scaling implementation of the calculation of the electrostatic potential arising from an arbitrary charge distribution. Our approach is making use of the multi-resolution basis of multiwavelets. The potential is obtained as the direct solution of the Poisson equation in its Green's function integral form. In the multiwavelet basis the formally non-local integral operator decays rapidly to negligible values away from the main diagonal, yielding an effectively banded structure where the bandwidth is only dictated by the requested accuracy. This sparse operator structure has been exploited to achieve linearly scaling and parallel algorithms. Parallelization has been achieved both through the shared memory (OpenMP) and the message passing (MPI) paradigm.

Our implementation has been tested by computing the electrostatic potential of the electronic density of long-chain alkanes and diamond fragments showing (sub)linear scaling with the system size and efficent parallelization.

1 Introduction

One of the most widespread yet challenging physical problems is the calculation of the electromagnetic interactions between charge distributions.[1] Its applications range from engineering problems, to physics, chemistry and biology. The full solution is in principle described by the four Maxwell equations coupling the electric field, the magnetic field, the charge density and the charge current. Moreover, the presence of media and their interaction with the fields renders the problem even more complicated.

If we restrict ourselves to the realm of electrostatic interactions (time independent fields, and no current), the Maxwell equations can be reduced to the Poisson equation, linking the scalar electrostatic potential V to the charge density ρ:

$$\nabla \cdot \epsilon(r) \nabla V(r) = -4\pi\rho(r) \tag{1}$$

where the permittivity ϵ in general is position dependent. Most commonly, the charge distribution

ρ is known and one is interested in obtaining its effect on the surroundings by computing the electrostatic potential. The formal solution for the free boundary problem with constant ϵ can be written in a closed form by making use of a Green's kernel formalism:

$$V(r) = \int_{\mathrm{R}^3} \frac{1}{|r - r'|} \rho(r') \, \mathrm{d}r' \tag{2}$$

From a computational point of view the challenge in modeling such a problem is twofold. On the one hand, the Green's kernel in Eq. (2) is non-separable: the Cartesian coordinates are coupled and the integral cannot be decomposed into the product of mono-dimensional integrals. On the other hand, the electrostatic interaction represented by the $1/r$ Green's kernel has a singularity at short distances and is decaying slowly for long distances; the singularity makes accurate computation of electrostatic interactions challenging and the long-range interaction makes the scaling with the system size challenging for straightforward computational approaches.

Several strategies have been devised to address the problem, depending e.g. on the boundary conditions and on the type of charge distribution. When possible the electrostatic equations are recasted onto a boundary problem thus limiting the system size to a two-dimensional surface instead of the whole three-dimensional space.[2, 3, 4, 5] This is the preferred approach in case e.g. of solute-solvent interactions.[3, 6] For charge distributions that can be conveniently described in terms of distributed multipoles, a very promising approach is constituted by the Fast Multipole Methods (FMM).[7, 8, 9, 10] For more general cases real-space mesh methods must be employed.[11, 12, 13, 14, 15, 16, 17, 18] Besides the generality, the advantage of such an approach is that it is well suited for modern parallel computing architectures: the parallelization is achieved by distributing the mesh among the compute nodes, although care must be taken in order to ensure a balanced workload.

The main disadvantage of real-space mesh methods is constituted by the large storage requirements implied.[13] This problem can be alleviated by making use of an adaptive multiwavelet approach where functions are represented on adaptive multi-resolution grids, the local refinement being dictated by a preselected accuracy.[19] The other advantage of a multiwavelet basis is constituted by the efficient representation of the integral operator in Eq. (2) which can be described in terms of narrow-banded matrices in the so called Non-Standard form.[20, 21] By approximating the integral kernel in Eq. (2) as a sum of Gaussian functions, the operator is decomposed into several components, each of which is Cartesian separable and non-singular in the short-range limit. The prohibitive scaling in the long-range limit is taken care of by the multi-resolution analysis, as each operator component is further decomposed into different length scales. In this work we will present a practical implementation of such a method, with focus on computational efficiency and application to molecular systems.

2 Mathematical Background

We will briefly introduce the mathematical background of our approach which is based on multi-resolution analysis and the multiwavelet basis.[22]

2.1 The multiwavelet basis

Following the original construction from Alpert,[23] a one-dimensional multiwavelet basis is obtained by defining the multi-resolution scaling space V_k^n as the space of piecewise polynomials on the unit interval

$$\begin{aligned} V_k^n \stackrel{\text{def}}{=} \{\ & f : \text{all polynomials of degree } \leq k \\ & \text{on } (2^{-n}l, 2^{-n}(l+1)) \text{ for } 0 \leq l < 2^n, \\ & f \text{ vanishes elsewhere } \} \end{aligned} \tag{3}$$

From this definition Alpert shows that each space V_k^n is fully contained in all spaces of higher resolution $V_k^m, m > n$, and we can therefore consider a so called "ladder of spaces" with increasing flexibility

$$V_k^0 \subset V_k^1 \subset \ldots \subset V_k^n \subset \ldots \tag{4}$$

It is well known that the basis obtained by taking the limit for $k \to \infty$ is dense in the L_2 norm sense, and it has also been shown that the limit $n \to \infty$ is dense. In other words, any function of L_2 can be represented within any given accuracy by making use of a polynomial of sufficiently high order at a given spatial refinement, or with sufficiently high refinement for a given polynomial order. In practical applications it is generally convenient to find a good balance by increasing both simultaneously.

The wavelet spaces W_k^n can be formally constructed by taking the orthogonal complement between two successive scaling spaces:

$$W_k^n \oplus V_k^n = V_k^{n+1} \tag{5}$$

By construction the wavelet space W_k^n is orthogonal to all scaling spaces $V_k^m,\ m \leq n$. In other words the first $k+1$ moments of a wavelet basis ψ^n of the space W_k^n are zero

$$\int_0^1 x^m \psi^n(x)\,\mathrm{d}x = 0, \qquad m = 0, \ldots, k \tag{6}$$

which means that the basis is very efficient for the representation of smooth functions.

Eq. (5) does not completely define the basis. This flexibility can be exploited in order to obtain a basis with useful properties, e.g. additional vanishing moments and symmetry (see Alpert[23] for details).

The change of basis from the scaling basis ϕ_l^{n+1} defining V_k^{n+1} to the compound scaling ϕ_l^n and wavelet ψ_l^n basis at scale n is undertaken by making use of a unitary, local transformation called filters:

$$\begin{pmatrix} \psi_l^n \\ \phi_l^n \end{pmatrix} = \begin{pmatrix} G^{(1)} & G^{(0)} \\ H^{(1)} & H^{(0)} \end{pmatrix} \begin{pmatrix} \phi_{2l+1}^{n+1} \\ \phi_{2l}^{n+1} \end{pmatrix} \tag{7}$$

The locality of the transformation ensures that it can be performed in linear complexity. In addition, it also ensures that it can easily be implemented on distributed memory architectures.

Although the construction described applies to the one-dimensional case, the multi-dimensional extension can be obtained by making use of tensor-product bases, which in three dimensions yields

$$\Phi_{ijk}^n(x,y,z) = \phi_i^n(x)\phi_j^n(y)\phi_k^n(z) \tag{8}$$

To summarize, the scaling basis is equivalent with the more commonly employed basis of the finite element method (FEM): the three-dimensional unit cube is uniformly subdivided into cubic cells, each with a polynomial basis. What separates the multiwavelet basis from FEM is the additional wavelet (or difference) basis, which allows for adaptive (non-uniform) grids.

2.1.1 Function representation

Functions are formally represented in a multiwavelet basis by projection, and we define P_k^n and Q_k^n as the projection operators onto V_k^n and W_k^n, respectively. It is useful to point out that on the unit interval

$$P_k^n + Q_k^n = P_k^{n+1} \tag{9}$$

and

$$\lim_{n\to\infty} P_k^n = 1 \qquad \lim_{n\to\infty} Q_k^n = 0 \tag{10}$$

Eq. (9) follows from the construction of the wavelet basis, whereas Eq. (10) is due to the completeness in the L_2 sense. The projection of an arbitrary function f is denoted

$$f^n \stackrel{\text{def}}{=} P_k^n f \qquad \mathrm{d}f^n \stackrel{\text{def}}{=} Q_k^n f \tag{11}$$

Any smooth function can be approximated to any finite precision by a scaling projection with sufficiently high resolution N, and by recursive application of Eq. (9), this approximation can be decomposed into its multi-resolution components

$$f \approx f^N \tag{12}$$

$$= f^0 + \sum_{n=0}^{N-1} \mathrm{d}f^n \tag{13}$$

Although mathematically equivalent, the representation in the combined (multi-resolution) scaling and wavelet basis in Eq. (13) has several advantages over the pure (high-resolution) scaling representation in Eq. (12). Because of the vanishing moments property of the wavelet basis (Eq. 6),

the sum in Eq. (13) is rapidly converging for smooth functions and can be locally truncated to a predefined precision ϵ based on the wavelet norm $\|\,\mathrm{d}f_l^n\|$ at each interval $2^{-n}(l, l+1)$

$$\|\,\mathrm{d}f_l^n\| < \frac{\epsilon}{2^{n/2}}\|f\| \tag{14}$$

which can be used to build adaptive, function specific grids that dramatically reduces the number of coefficients needed to represent the function to the given accuracy.

The vanishing moments also means that the Coulomb interaction between charges represented in the wavelet basis decays very rapidly with distance, as the leading multipole order of this interaction is $k+1$. This is what ultimately allows for linear scaling algorithms, as the full (long-ranged) interaction is decomposed into different length scales, where the interaction is "local" at each length scale separately.

2.2 Operator representation

In order to apply the Poisson operator efficiently in three dimensions it is crucial to achieve an optimal representation. Several properties need to be considered:

1. In order to exploit the tensorial representation of the basis, a representation that separates the Cartesian coordinates needs to be employed.

2. Coupling between different length scales should be avoided in order to limit communication and to exploit adaptivity for function representations.

3. The operator representation should ideally be banded in order to limit the coupling to the minimum necessary.

The first point is achieved by constructing a separated representation of the Poisson kernel in terms of Gaussian functions:

$$\frac{1}{r} \simeq K_M(r) = \sum_{i=1}^{M} a_i e^{-\alpha_i r^2} \tag{15}$$

where the components of the expansion are determined by an efficient quadrature scheme.[24] Although the Poisson kernel is not separable, each term in Eq. (15) is, and the expansion can be made arbitrarily accurate since for any given ϵ it is possible to find a rank M such that

$$\left|\frac{K_M(r) - 1/r}{1/r}\right| < \epsilon \tag{16}$$

is fulfilled within any given interval $r \in [r_0, r_1]$, where r_0 is chosen so that the contribution due to the integration at the singularity can be neglected,[25] and r_1 is the longest possible distance in the domain ($\sqrt{3}$ for the unit cube).

The second and third points are achieved through the so called Non-Standard form of the operator.[20, 21] The application of an operator T can be written as

$$g(x) = [Tf](x) \tag{17}$$

which can be discretized in a Galerkin scheme by projecting both the functions and the operator on a given scaling space V_k^n

$$T_k^n \stackrel{\text{def}}{=} P_k^n T P_k^n \tag{18}$$

By recursive application of Eq. (9), T_k^N can be rewritten in a telescopic series:

$$\begin{aligned} T_k^N &= P_k^N T P_k^N && (19)\\ &= P_k^0 T P_k^0 + \sum_{n=0}^{N-1} (P_k^{n+1} T P_k^{n+1} - P_k^n T P_k^n) && (20)\\ &= P_k^0 T P_k^0 + \sum_{n=0}^{N-1} (P_k^n + Q_k^n) T (P_k^n + Q_k^n) - P_k^n T P_k^n && (21)\\ &= P_k^0 T P_k^0 + \sum_{n=0}^{N-1} Q_k^n T Q_k^n + Q_k^n T P_k^n + P_k^n T Q_k^n && (22)\\ &= T_k^0 + \sum_{n=0}^{N-1} (A_k^n + B_k^n + C_k^n) && (23) \end{aligned}$$

where we have implicitly defined the following components:

$$A_k^n \stackrel{\text{def}}{=} Q_k^n T Q_k^n \qquad B_k^n \stackrel{\text{def}}{=} Q_k^n T P_k^n \qquad C_k^n \stackrel{\text{def}}{=} P_k^n T Q_k^n \tag{24}$$

The full operator can in theory be recovered by taking the limit to infinite refinement and by making use of Eq. (10):

$$T = \lim_{N\to\infty} T_k^N = T_k^0 + \sum_{n=0}^{\infty} (A_k^n + B_k^n + C_k^n) \tag{25}$$

and by truncating the infinite sum we arrive at a multi-resolution operator with finite precision. If we introduce the following auxiliary functions:

$$\begin{aligned} \hat{g}^n &\stackrel{\text{def}}{=} T_k^n f^n && (26)\\ \tilde{g}^n &\stackrel{\text{def}}{=} C_k^n \, \mathrm{d}f^n && (27)\\ \mathrm{d}\tilde{g}^n &\stackrel{\text{def}}{=} (A_k^n + B_k^n)(f^n + \mathrm{d}f^n) && (28) \end{aligned}$$

the operator application can be written

$$\begin{aligned} \hat{g}^N = T_k^N f^N &= T_k^0 f^0 + \sum_{n=0}^{N-1} (A_k^n + B_k^n + C_k^n)(f^n + \mathrm{d}f^n) && (29)\\ &= \hat{g}^0 + \sum_{n=0}^{N-1} \tilde{g}^n + \mathrm{d}\tilde{g}^n && (30) \end{aligned}$$

In the Non-Standard form the operator is applied one scale at the time, starting from scale zero. In this way the function g can be built adaptively in the same way as for the projection of functions, refining locally based on the wavelet norm at scale n and translation l

$$\| \mathrm{d}\tilde{g}_l^n \| \leq \frac{\epsilon}{2^{n/2}} \|g\| \tag{31}$$

At each scale $\tilde{g}^n$ and $\mathrm{d}\tilde{g}^n$ are computed whereas $\hat{g}$ is only computed at the coarsest scale and reconstructed by recursion at all scales $n > 0$:

$$\hat{g}^n = \hat{g}^{n-1} + \mathrm{d}\tilde{g}^{n-1} + \tilde{g}^{n-1} \qquad n \leq N \tag{32}$$

Eq. (32) shows how the coupling between scales is then achieved by propagating the result from the coarsest to the finest scale. This is done using the filter operations of Eq. (7) in linear complexity.

The main advantage of such a construction is connected to the vanishing moments property of the multiwavelet basis. Apart from the coarsest scale where the full operator is applied, the pure scaling component T_k^n of the operator is never used. The other components contain at least one projection onto the wavelet basis (B_k^n and C_k^n) or two (A_k^n) and it can be shown that these terms decay rapidly with the spatial separation between two nodes[20] and are hence diagonally dominated, opening the way for linear scaling algorithms with reduced communication requirements for parallel implementations.

2.3 Extension to several dimensions

The extension to several dimensions can formally be achieved by a standard tensor-product structure (Eq. 8). The main challenge arises from the so called "curse of dimensionality": the storage and computing costs scales exponentially with the dimension d. The tensor product structure of functions and operators makes the computational cost per grid cell scale as Mdk^{d+1} instead of k^{2d}, where M is the separation rank and k is the order of the polynomial basis, which reduces the exponent and complexity significantly, making it feasible to use the described approach up to $d = 3$. For dimensionality higher than 3, it would however be unavoidable to make use of rank reduction techniques as described originally by Beyklin *et al.* [24, 26, 27] and recently employed by Bischoff and Valeev.[28] It is beyond the scope of the present paper to describe the details of the multi-dimensional implementation which have been presented in detail elsewhere.[25] Suffice to say that the number of components of the Non-Standard form becomes 2^{2d} and all but the pure scaling component are employed in the operator application. What makes it feasible to use such an approach is again the tensor product structure coupled with the vanishing moments which significantly reduce the cost of applying the different components.

3 Implementation

Implementing the mathematical formalism outlined in section 2 in an efficient computer code is a complex task. In grid based, real-space methods the amount of grid data required grows rapidly with the complexity of the function. The computational requirements increase similarly with the number of grid points. For the method to be feasible, it is important to utilize algorithms which reduce the memory footprint, lower the computational demands and have favorable parallel scaling properties. Finding a good balance between these requirements can be challenging. We have addressed these issues by exploiting three major algorithmic ideas: automatic grid adaptation, use of sparsity and parallelization. Our computer code has been implemented in C++, due to the demands for high performance, as well as the complex data structures involved. The code has been written using a fully object-oriented approach, using generic programming and polymorphic classes.

3.1 Grid adaptation

In order to drastically reduce the memory footprint we exploit the refinement properties of multiwavelets to automatically generate fully problem adapted grids. One of the major problems encountered in FEM that employs a uniform (Eq. 12) distribution of grid points is that the overall precision of the representation is limited by the regions where the representation is poor, typically in regions where the function is changing rapidly. With uniform grids large parts of the function will be excessively overrepresented, if good precision is to be achieved. For molecular systems it is well known that the problematic regions are in the vicinity of the nuclear positions, and the problem has previously been addressed by projecting out the spherically symmetric part around each nucleus and treating this separately.[29, 30, 31] The remaining part of the solution is then treated much more efficiently by a uniform FEM grid. The main issues with this approach are the lack of generality and the complication that arises from the coupling between the different parts. Instead, by expanding the functions in the multi-resolution wavelet basis (Eq. 13), and truncating this expansion locally according to Eq. (14), we achieve grid adaptation that largely solves the problem by locally varying the grid density according to the actual needs. However, much of the code complexity arises from the grid adaptation in a tensorial basis, since the grids cannot be easily stored in large blocks or arrays. Instead the grids are stored in sub-blocks, or "nodes", whose spatial extensions are dependent on the level of refinement. Each node has a fixed number of grid points, typically $10 \times 10 \times 10$ points, and is uniquely addressable by four integers (scale and translation in three dimensions). From an implementation point of view, it is practical to store the grid blocks in a tree structure.

3.2 Sparsity

In the Non-Standard multiwavelet representation of integral operators the operator matrices become sparse and diagonally dominant. The sparsity implies that the operator has a limited bandwidth at each scale, and is negligible outside the bandwidth. By exploiting the sparsity one avoids calculating trivial zero contributions, in addition to reducing the data access. We will show that by properly exploiting the sparsity, the amount of computation scales linearly with the size of the problem.

Fully exploiting the operator sparsity is rather complex in a general, n-dimensional tensorial basis. In tensorial form the operator has 4^d components where d is the dimensionality of the problem (A, B, C and T of Eq. (25) in each dimension). The structure of the operator is further complicated by the fact that different operator components have different bandwidths (that also changes depending on the length scale), and in the current implementation sparsity is fully utilized by treating all 64 (for $d = 3$) operator components separately.

3.3 Parallelization

As pointed out above, any grid based, real-space method is faced with two major problems: The data storage requirements that quickly exceed the available memory, and the computational time that increases accordingly. Thus, calculations rapidly become intractable. Both problems can be addressed simultaneously by parallel processing. By having more processors working on the same problem the computation time can be reduced without additional memory requirements on a given computational device. Furthermore, on cluster type machines with distributed memory, the memory requirements on the individual compute nodes (hosts) can be reduced by only storing parts of the function representation.

In the current version of the code we use two parallelization schemes: Utilizing symmetric multi-processing (SMP) and shared memory using the standard OpenMP paradigm, as well as distributed processing using the Message Passing Interface (MPI) for communication between computational hosts.

3.3.1 Symmetric multi-processing and OpenMP

Modern computers have multiple, independent computational cores sharing a common memory space, and using OpenMP technology it does not require much in terms of implementation to efficiently utilize all available processors on a given host. Using multiwavelets, there is by construction no overlap between basis functions located on different grid cells, so each node is formally independent of the other nodes in the tree. Efficient parallelization is achieved by traversing the tree structures and defining tasks for each node, letting the OpenMP runtime system handle the

scheduling. For function representation (projection of a function onto the multiwavelet basis), the tasks are truly independent and the parallelization is simple and highly efficient. Since every OpenMP thread has a fair amount of computation to perform, the parallelization overhead is negligible.

Due to the grid adaptation, parallelization of the operator application is somewhat more complicated. Applying an operator with a non-zero bandwidth, implies accessing neighboring nodes that may not be available at the given scale, since the grid refinement might have stopped at a coarser scale for the neighboring part of the function. By applying the wavelet transform (Eq. 7), information about the function at the correct scale can be generated. However, generating the corresponding coefficients is costly, and they often get reused many times. It is wasteful to generate the coefficients every time, and instead they are generated once when needed, and kept until explicitly released. This complicates the parallelization of the operator application, since a generated node might be within the bandwidth of two resultant nodes being processed simultaneously, causing a data race. By carefully using explicit data locking at the deepest possible location in the code, and by tuning task scheduling, we have been able to achieve very low impact from locking, achieving good parallel performance.

3.3.2 Distributed processing and MPI

While SMP reduces the total computational time, it does not reduce the local memory requirements. By using homogeneous clusters of computers, we can reduce both the computational time and the local memory footprint. However, contrary to SMP, distributed processing adds significant complexity to the code.

When dealing with distributed data structures, one must consider not only how the data should be distributed, but also how it should be accessed. In the current version of the code we employ a series of techniques to distribute data in an efficient manner. One of the key elements is a redundant representation of the tree structure. Each host has a complete and identical, skeleton tree structure. Each node in the tree has a label identifying the host it belongs to, and only the owner of a node has allocated memory for it, and has coefficients, although some redundancy in storage is allowed.

When applying an operator on a distributed tree (both input and output functions are distributed), one finds that some of the required data is located on different MPI hosts and needs to be transferred. Because of a non-zero bandwidth of the operator, each node of the output function (the potential) will get contributions from several of the surrounding nodes of the input function (the charge density), and at the same time each node of the input function will give contribution to several surrounding nodes of the output function.

In such situations there are two possible communication strategies: one can focus on the data

distributions of either the output or the input function. Each MPI host will be allocated a number of nodes of the potential to calculate, and one can then choose to fetch all the data needed of the charge distribution to calculate these nodes. This requires MPI communication only prior to operator application, but will result in a redundant distribution of the charge density nodes (several MPI hosts will have the same data).

On the other hand, all MPI hosts also have a number of allocated nodes of the charge density, and one can instead choose to calculate the potential arising from these nodes directly. This requires no initial communication step, but will result in an incomplete potential for all MPI hosts, and the result needs to be communicated and added up by the appropriate MPI host afterwards. We find that the latter strategy generally requires less data communication, so this will be our method of choice, although both strategies are implemented.

In both schemes it is equally important to ensure data localization to minimize communication between hosts. When the real-space domains of all MPI hosts are well localized, communication is reduced drastically, and when the number of MPI hosts gets large the limited bandwidth of the operator will ensure that the communication will be limited to only near neighbors. In the current code, good localization is achieved by first building a skeleton tree structure without coefficients, which tries to estimate the final tree structure as closely as possible. Then the tree is traversed through a so-called space filling curve: a path constructed recursively such that its fractal limit coincide with the whole multi-dimensional space. In this way, an ordering of the nodes in a tree structure is introduced and the data can easily be distributed by partitioning the curve into contiguous chunks. Moreover, the lengths of such portions should ideally reflect the computational work load in subsequent computations.

There are several ways to construct such a space filling curve, but to ensure maximum locality Griebel[32] suggests Hilbert curves. The construction of the Hilbert curve starts at scale $n = 1$, connecting all 2^d nodes in a specific sequence. In particular, addressing each node using a bit notation (one bit for each direction), the fundamental Hilbert curve will start at a given node, and end at one of its adjacent nodes (only one bit different from the starting node). Each step of the curve can be defined by a bit switch: only one bit of the sequence changes every time. The procedure is continued recursively through all nodes of the tree, but the sequence through which the children nodes is connected is changing in such a way that the curve remains continuous. Figures 1 and 2 show the difference between the Lebesgue curve, which is induced by the natural ordering of the bit sequences (00, 01, 10, 11) and the Hilbert curve (00, 10, 11, 01) for three refinement levels in two dimensions. The natural bit ordering of the Lebesgue curve implies "jumps" in the sequence (shown by the Z shape) which will ultimately make the curve converge to its space-filling limit point-wise to a discontinuous limiting curve. The Hilbert curve converges uniformly to a continuous space filling curve. As a consequence, employing a Hilbert curve to

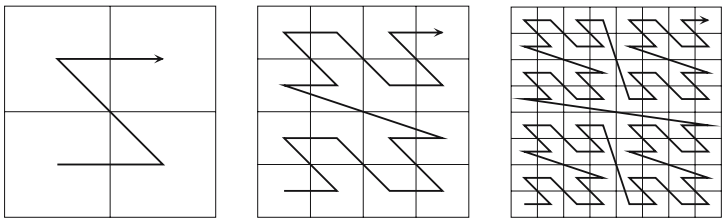


Figure 1: Three refinement levels in the construction of the Lebesgue curve in 2D.

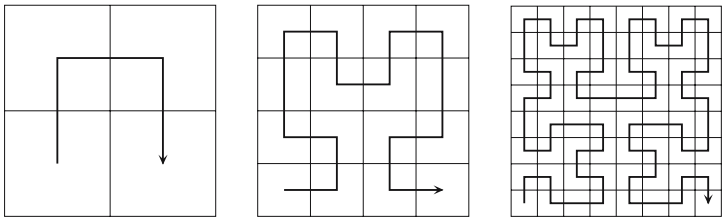


Figure 2: Three refinement levels in the construction of the Hilbert curve in 2D.

partition the domain will result in connected domains with good locality properties which will eventually minimize the communication overhead.

Besides the communication overhead, the main concern in distributed processing is to ensure a balanced work load between MPI hosts. This is not a straightforward task as the work needed to calculate a given node can vary a lot, depending on the bandwidth (in each dimension, at a specific length scale) of the operator as well as the local norms of both the input and output functions. This means that simply dividing the Hilbert curve into equally sized portions (which is the way the data is initially distributed) does usually not give the best work distribution, and some tuning is needed prior to operator application. This is done by assigning a work load to each node in the tree and shifting the domain boundary along the Hilbert curve in such a way that the total work gets evenly distributed. In this way the communication is done only between neighboring MPI hosts. However, our experience show that this load balancing algorithm is not always flexible enough, and it requires that the work load is quite well distributed to begin with, and it will probably be insufficient when the number of MPI hosts gets large.

4 Results

In this section we will demonstrate the performance of our code with various test calculations on realistic molecular systems, specifically linear alkane chains C_nH_{2n+2} for $n = \{2, \ldots, 70\}$ and pyramidal diamond fragments $C_{(2n+3)(n+2)(n+1)/6}H_{2(n+2)(n+1)}$ for $n = \{1, \ldots, 9\}$. All systems were constructed with a constant C-C bond length of 0.154 nm, and terminal H atoms were

attached. The electron densities were precomputed at Density Functional Theory (DFT) level (BLYP[33, 34, 35] functional, Dunning's DZ[36, 37] Gaussian basis) using the LSDalton[38] program and then projected onto the multiwavelet basis. Similar calculations were performed by Watson and Hirao[39] using a spectral-element method with a high-order Chebyshev polynomial basis.

All computations have been performed on a cluster consisting of 2×8 cores Intel Xeon E5-2670 processors with 16 GB memory, connected by an infiniband network.

4.1 Accuracy

One attractive property of the multiwavelet basis that separates it from other bases commonly used in quantum chemistry calculations is the strict error control in both function projections and operator applications. By truncating the wavelet expansions *locally* according to Eq. (14) we can control the *global* error of the calculations. We demonstrate this by calculating the Coulomb self-repulsion energy of a charge distribution $\rho(r)$:

$$E = \int \rho(r) V(r) dr \tag{33}$$

where the potential $V(r)$ is obtained by solving the Poisson equation (Eq. 2). There are several critical issues in order to guarantee the precision of the calculated energy: The charge distribution as well as the Poisson operator (e.i. the kernel expansion in Eq. (15)) must be represented with sufficient accuracy, and the operator must be applied in such a way that no significant contributions are omitted, while at the same time thresholding as much as possible for efficiency. A detailed discussion of the accuracy parameters that appear in the operator construction and application is presented in a separate study.[25]

Table 1: Accuracy of Coulomb energy and charge integral of small alkane molecules. Densities are precomputed at DFT (BLYP) level and analytic energies are calculated in the Gaussian basis (Dunning DZ) using the LSDalton program. Computation times are for a single processor.

Requested precision	Time (sec)	Coulomb energy (hartree)	Relative error Energy	Charge
		C_2H_6		
ϵ = 1.0e-05	12	80.085849034	3.2e-06	5.9e-07
ϵ = 1.0e-07	41	80.086102499	1.4e-08	2.9e-09
ϵ = 1.0e-09	607	80.086103631	1.3e-10	2.2e-11
Analytic	-	80.086103641	-	-
		C_4H_{10}		
ϵ = 1.0e-05	13	205.548292326	2.0e-06	4.4e-07
ϵ = 1.0e-07	52	205.548692277	1.4e-08	2.4e-09
ϵ = 1.0e-09	774	205.548695185	1.3e-10	1.6e-11
Analytic	-	205.548695213	-	-
		C_6H_{14}		
ϵ = 1.0e-05	16	355.995462175	3.4e-06	4.2e-07
ϵ = 1.0e-07	63	355.996666233	1.4e-08	1.8e-09
ϵ = 1.0e-09	792	355.996671114	1.4e-10	1.3e-11
Analytic	-	355.996671163	-	-
		C_8H_{18}		
ϵ = 1.0e-05	18	523.397878651	4.1e-06	3.8e-07
ϵ = 1.0e-07	73	523.400041990	1.4e-08	1.2e-09
ϵ = 1.0e-09	1035	523.400049203	1.4e-10	9.7e-12
Analytic	-	523.400049277	-	-
		$C_{10}H_{22}$		
ϵ = 1.0e-05	20	703.6228144331	7.1e-06	3.4e-07
ϵ = 1.0e-07	85	703.6277668016	1.5e-08	1.3e-09
ϵ = 1.0e-09	1148	703.6277770551	1.5e-10	8.4e-12
Analytic	-	703.6277771588	-	-

Test calculations were performed on the smaller alkane systems, $n = \{2, 4, 6, 8, 10\}$. The Coulomb energy was calculated analytically in the Gaussian basis by LSDalton, and the accuracy of our Poisson solver is tested against the analytical result. For each system we calculate the energy to three different target accuracies $\epsilon = \{10^{-5}, 10^{-7}, 10^{-9}\}$ and the numbers are presented in table 1. The error in the charge integral gives an indication of the accuracy of the projection of the Gaussian basis onto the multiwavelet basis, and we see that the errors are well within the given threshold. The errors in the Coulomb energy are also consistently below the requested

Table 2: Absolute and relative errors in Coulomb energies calculated to requested relative accuracy 10^{-6}, and requested absolute accuracy 10^{-3}. Densities are precomputed at DFT (BLYP) level and analytic energies are calculated in the Gaussian basis (Dunning DZ) using the LSDalton program. Computation times are for a single processor.

Alkane system n in C_nH_{2n+2}

		Req. rel. precision 1.0e-6			Req. abs. precision 1.0e-3		
n	Coulomb energy	Time	Error		Time	Error	
	(hartree)	(sec)	Relative	Absolute	(sec)	Relative	Absolute
2	80.08610364	24.9	1.0e-07	8.3e-06	7.7	2.7e-06	2.1e-04
10	703.62777716	42.8	1.1e-07	8.1e-05	30.9	2.4e-07	1.7e-04
20	1752.56975975	62.6	1.7e-07	3.0e-04	72.0	1.0e-07	1.8e-04
30	2936.25648588	84.0	4.4e-07	1.3e-03	110.8	2.0e-07	6.0e-04
40	4211.56647408	101.6	4.9e-07	2.1e-03	151.6	1.7e-07	7.3e-04
50	5555.20166293	116.3	5.5e-07	3.0e-03	190.4	1.6e-07	8.9e-04
60	6953.41087253	130.9	6.9e-07	4.8e-03	225.9	6.6e-08	4.6e-04
70	8396.93995214	143.0	8.8e-07	7.5e-03	259.0	6.4e-08	5.3e-04

precision and are more or less independent of the size of the system (at least for the more accurate calculations).

The calculations were performed using a polynomial order $k = -\log(\epsilon) + 2$ that increases with increasing accuracy. It should be noted that the increasing accuracy is *not* a result of the increasing polynomial order, and a fixed, low polynomial order should give similar precision, but it requires much deeper local refinement of the grid. This relation between target accuracy and polynomial order is empirically chosen, as it seems to be a good compromise between the total number of cells vs. the order of the polynomial in each cell.[25]

4.2 Linear scaling

A direct consequence of the banded structure of the operator matrix is that the computation time of applying the operator should scale linearly with system size. We test this property by calculating the Coulomb energy on the full range of alkane systems, $n = \{2, \ldots, 70\}$. By making use of the Fast Multipole Method and adaptive resolution (adaptive in polynomial order, not cell size) Watson and Hirao[39] were able to achieve linear scaling of the calculation of the Coulomb energy for both the alkane and the diamond systems, with a relative error of around 1 ppm.

In trying to reproduce these calculations we choose a 9th order polynomial basis, with a target relative accuracy of $\epsilon = 10^{-6}$, and the numbers are presented in table 2. We see that although the relative accuracy is somewhat degrading as the system size increase, all numbers are within

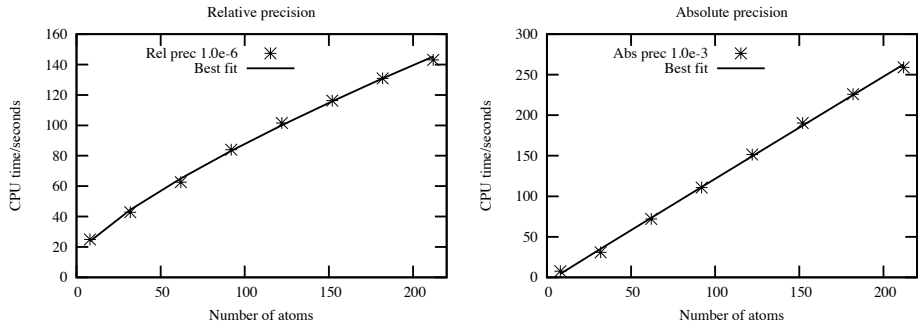


Figure 3: Scaling of computation time on a single processor for linear alkane systems when requested accuracy is relative (left) and absolute (right). Best fit curves are Eqs. (34) and (35), respectively.

the requested precision. The computation times are plotted to the left in figure 3 and we see that the scaling is in fact sublinear. The solid line in the plot is obtained using the Levenberg-Marquardt[40, 41] algorithm to fit the timings t with respect to the number of atoms n to the function

$$t(n) = 12.5 + 2.34n^{0.754} \tag{34}$$

which shows an exponent significantly lower than one. This behavior can be understood by the use of relative rather than absolute precision in combination with the automatic adaptivity of the grids in our calculations. As the system size increases the global norm of the functions involved increase accordingly, and the local truncation criterion in Eq. (14) is gradually relaxed. This means that the local resolution of the grid around each CH_2 fragment is gradually getting coarser, while the overall *relative* accuracy of the function is maintained. As the computation time is expected to be directly related to the overall number of grid cells, one can expect the scaling to less than linear as the system size increases.

If we on the other hand are interested in an absolute accuracy, we need to maintain the local high resolution around each fragment as the system size is increased. In this case the number of grid cells should grow linearly with the size of the system, and one can expect the computation time to do the same. This would correspond to the calculations done by Watson and Hirao,[39] as their truncation is based on the local rather than global norm of the function and the grid of the full system is more or less the union of the grids around its constituent atoms.

Even though our code is based on relative precision, we can simulate absolute precision by gradually increasing the relative accuracy as the system size grows in such a way that the absolute error remains approximately constant. We present in table 2 also the numbers for such calculations, where the absolute error is kept below 10^{-3}, and the computation times are shown to the right of figure 3. We see that the slope of this curve is steeper than the corresponding curve with relative

Table 3: Wall clock computation time in seconds for parallel calculation of electronic potential of diamond fragments using pure OpenMP, pure MPI and hybrid MPI/OpenMP strategies. Densities are precomputed at DFT (BLYP) level using the LSDalton program.

Number of CPUs			Diamond system n in $C_{(2n+3)(n+2)(n+1)/6}H_{2(n+2)(n+1)}$								
MPI	OMP	TOT	1	2	3	4	5	6	7	8	9
1	1	1	29.4	46.2	64.7	97.8	133.9	166.8	213.4	269.4	332.0
1	2	2	15.5	25.5	33.0	51.3	69.5	87.2	110.0	138.9	163.4
1	4	4	8.0	12.8	17.4	27.0	36.1	47.3	57.8	73.7	90.1
1	8	8	4.2	6.5	8.8	13.9	18.6	23.5	29.4	36.7	45.0
1	16	16	2.4	3.5	4.7	7.5	9.6	12.2	15.4	19.1	23.3
2	1	2	17.9	28.4	35.8	59.2	81.6	93.0	121.7	147.5	176.5
4	1	4	10.1	16.3	21.2	32.5	51.6	53.8	60.8	85.3	100.7
8	1	8	6.1	9.4	11.8	19.9	25.6	28.0	35.4	43.7	50.7
16	1	16	5.0	6.2	8.2	12.4	15.0	17.3	23.0	26.1	31.3
32	1	32	3.6	4.2	5.4	9.5	9.2	10.9	14.1	16.8	20.0
64	1	64	3.1	3.9	4.3	6.5	6.6	7.4	10.3	11.4	13.4
128	1	128	5.2	5.7	6.4	7.6	7.8	8.8	10.9	10.9	11.9
2	16	32	1.5	2.2	2.9	4.8	6.0	7.0	9.6	11.4	13.7
4	16	64	1.1	1.6	2.0	3.2	4.3	4.7	6.3	7.3	7.9
8	16	128	1.0	1.4	1.8	2.9	3.3	3.8	4.8	5.9	6.4
16	16	256	0.9	1.3	1.6	2.2	2.9	3.3	4.2	4.9	6.0
32	16	512	1.1	1.3	1.5	2.0	2.4	2.8	3.4	4.0	4.8

precision, as the absolute error criterion progressively becomes more demanding than the relative one. The Levenberg-Marquardt algorithm gives a best fit

$$t(n) = -6.0 + 1.33n^{0.991} \tag{35}$$

where the exponent now is very close to one. Comparing with Watson and Hirao, we see a factor of 8-10 reduction of the computation time while the accuracy is about an order of magnitude higher. However, it should be pointed out that their objective was only to demonstrate the linear scaling behavior of their method, and their parameters were not optimized for computational speed.

If one is satisfied with sticking only to a relative accuracy, the calculations quickly becomes much more favorable, and for the biggest alkane system $C_{70}H_{142}$ (562 electrons) the calculation is more than 15 times faster than Watson and Hirao with comparable accuracy.

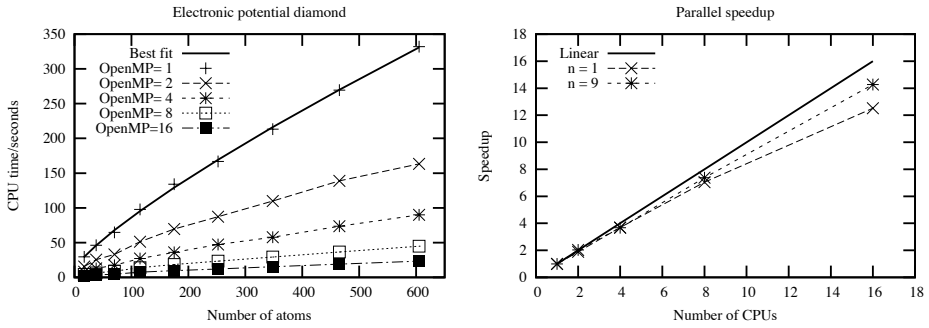


Figure 4: Left: Scaling of computation time for the calculation of the electronic potential of diamond fragments with number of processors $1, \ldots, 16$ using OpenMP. Numbers are taken from table 3. Best fit curve is Eq. (36). Right: Parallel efficiency of the operator application for diamond systems 1 and 9.

4.3 Parallel performance

We test the parallel performance of our code by calculating the electrostatic potential of the pyramid shaped diamond fragments $C_{(2n+3)(n+2)(n+1)/6}H_{2(n+2)(n+1)}$ for $n = \{1, \ldots, 9\}$, where the biggest system contains more than 600 atoms. The calculations were done using a 9th order polynomial basis with a requested relative accuracy of $\epsilon = 10^{-6}$.

Figure 4 shows to the left the walltime of the diamond calculations with 1, 2, 4, 8 and 16 shared memory processors. There are several things to note about this figure. Firstly, we see that the scaling with respect to system size of the single processor calculation is again sublinear. The Levenberg-Marquardt algorithm gives

$$t(n) = 11.6 + 1.84n^{0.805} \tag{36}$$

which is quite close to what we had for the linear alkane systems (up to the biggest alkane system of 200 atoms the difference is less than 2%). This indicates that the computation time depends solely on the size of the system, not its geometry.

Secondly, we can see a reduction of computation time by roughly a factor of two when the number of processors is doubled, while the scaling properties with respect to the number of atoms is kept also for the parallel computations. To the right of figure 4 we see the parallel speedup for the smallest and the biggest system, which shows an efficiency of 80 and 90 %, respectively, on 16 processors. It is worth noticing that for the biggest system the performance does not fall off significantly up to 16 processors, and it would be interesting to test the code on computers with even more shared memory processors available.

Even though the shared memory parallelization shows very good performance there are still reasons for utilizing distributed memory (MPI) parallelization techniques. Firstly, the memory requirements for representing three-dimensional functions in the multiwavelet basis is still rather

big, even with automatic grid adaptation, and the biggest diamond calculation presented in figure 4 more or less exhausts the available resources (16 GB). By adding another layer of parallelization the functions can be distributed over the memory of several MPI hosts, and we can reach even bigger systems. It is expected that the parallelization overhead is more substantial for the MPI implementation, but with an efficient OpenMP implementation there are prospects of efficiently utilizing thousands of processors in a single calculation using a hybrid MPI/OpenMP strategy.

Table 3 shows the computation time for the diamond fragments using different numbers of processors, both in pure OpenMP and pure MPI, as well as using a hybrid MPI/OpenMP strategy, and we see that the wall clock computation time for the biggest diamond system $C_{385}H_{220}$ (2530 electrons) can be pushed down to 5 seconds (which is almost 1000 times faster than Watson and Hirao's single processor calculations) if one throws enough processors on the problem. However, the parallel efficiency in this case is less than 20%, and we would need even bigger systems to fully utilize the capacity of modern parallel computer clusters.

The effect of the Hilbert curve partition in the calculations presented above is not substantial, but it is expected to become more important as the system size increases. We do observe a slight decrease of post operator communication using the Hilbert curve for up to 128 MPI hosts, but the work load distribution of the Hilbert vs. Lebesgue curve is very different, and the overall effect is quite ambiguous. It seems difficult to obtain a strong scaling and reduce the wall clock computation time to less than a few seconds, which means that bigger systems are needed in order to efficiently utilize thousands of MPI processes, where the Hilbert curve is expected to really make a difference, but for such calculations there are other bottlenecks like work load estimation and balancing, and the current implementation cannot efficiently balance more than a few hundred MPI processes. However, the hybrid implementation is rather efficient and a thousand processors is then still within reach, which is satisfactory for the current applications of the code.

5 Conclusions

We have shown that by making use of the properties of multi-resolution analysis and the multiwavelet basis the electrostatic potential arising from arbitrary charge distributions can be calculated efficiently and with guaranteed precision. Test calculations on linear alkane chains demonstrates the inherent linear scaling of the application of integral operators in the multiwavelet formalism. In fact, by virtue of the automatic grid adaptation in our implementation, the scaling becomes sublinear if only a relative accuracy is maintained, as the local accuracy criterion gradually becomes more relaxed as the norm of the function increases. This was demonstrated both for linear alkanes and pyramidal diamond fragments, which showed similar scaling behavior, indicating that this property is independent of the geometry of the system.

The code has been successfully parallelized using both a shared memory (OpenMP) and a distributed memory (MPI) strategy, as well as a combination of these, and it is shown that a hybrid MPI/OpenMP strategy is preferable for a given number of processors. We show a significant improvement in computation time compared to previously reported numbers: an order of magnitude for single-processor calculations, and three orders of magnitude reduction in wall time if parallelization is added.

6 Acknowledgments

This work has been supported by the Research Council of Norway through a Centre of Excellence Grant (Grant No. 179568/V30) and from the Norwegian Supercomputing Program (NOTUR) through a grant of computer time (Grant No. NN4654K).

References

[1] Jackson J. D. *Classical Electrodynamics*. Wiley, 1998.

[2] Benighaus T. and Thiel W. Efficiency and accuracy of the generalized solvent boundary potential for hybrid qm/mm simulations: Implementation for semiempirical hamiltonians. *J. Chem. Theory Comput.*, 4(10):1600–1609, Oct 2008.

[3] Tomasi J., Mennucci B., and Cammi R. Quantum mechanical continuum solvation models. *Chem. Rev.*, 105(8):2999–3093, Jan 2005.

[4] Zhou H. X. Boundary element solution of macromolecular electrostatics: interaction energy between two proteins. *Biophysical Journal*, 65(2):955–963, Aug 2005.

[5] Liang J. and Subramaniam S. Computation of molecular electrostatics with boundary element methods. *Biophysical Journal*, 73(4):1830–1841, Jul 2005.

[6] Tomasi J. and Persico M. Molecular interactions in solution: An overview of methods based on continuous distributions of the solvent. *Chem. Rev.*, 94(7):2027–2094, 1994.

[7] Rokhlin V. Rapid solution of integral equations of classical potential theory. *J. Comput. Physics*, 60(2):187–207, 1985.

[8] Greengard L. and Rokhlin V. A fast algorithm for particle simulations. *J. Comput. Physics*, 73(2):325–348, 1987.

[9] White C. A. and Head-Gordon M. Derivation and efficient implementation of the fast multipole method. *J. Chem. Phys.*, 101(8):6593–6605, 1994.

[10] Choi C., Ruedenberg K., and Gordon M. S. New parallel optimal-parameter fast multipole method (opfmm). *J. Comput. Chemistry*, 22(13):1484–1501, 2001.

[11] Losilla S. A., D Sundholm D., and Jusélius J. The direct approach to gravitation and electrostatics method for periodic systems. *J. Chem. Phys.*, 132(2):024102, Jan 2010.

[12] Jusélius J. and Sundholm D. Parallel implementation of a direct method for calculating electrostatic potentials. *J. Chem. Phys.*, 126(9):094101, Jan 2007.

[13] Berger R. J. F. and Sundholm D. A non-iterative numerical solver of poisson and helmholtz equations using high-order finite-element functions. *Advances in Quantum Chemistry*, 50:235–247, Jan 2005.

[14] Beck T. L. Real-space mesh techniques in density-functional theory. *Reviews of Modern Physics*, 72(4):1041–1080, 2000.

[15] Bylaska E. J., Holst M., and Weare J. H. Adaptive finite element method for solving the exact kohn-sham equation of density functional theory. *J. Chem. Theory Comput.*, 5(4):937–948, Jan 2009.

[16] Chen L., Holst M., and Xu J. The finite element approximation of the nonlinear poisson-boltzmann equation. *Siam J. Numer. Anal*, 45(6):2298–2320, 2007.

[17] Holst M., Baker N., and Wang F. Adaptive multilevel finite element solution of the poisson-boltzmann equation i. algorithms and examples. *J. Comput. Chemistry*, 21(15):1319–1342, 2000.

[18] Genovese L., Deutsch T., Neelov A., Goedecker S., and Beylkin G. Efficient solution of poisson's equation with free boundary conditions. *J. Chem. Phys.*, 125:074105, 2006.

[19] 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.

[20] 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.

[21] Gines D., Beylkin G., and Dunn J. Lu factorization of non-standard forms and direct multiresolution solvers* 1. *Appl. and Comput. Harmonic Analysis*, 5(2):156–201, 1998.

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

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

[24] Beylkin G. and Monzón L. On approximation of functions by exponential sums. *Appl. and Comput. Harmonic Analysis*, 19(1):17–48, 2005.

[25] 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.

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

[27] Beylkin G. and Mohlenkamp M. J. Numerical operator calculus in higher dimensions. *Proceedings of the National Academy of Sciences*, 99(16):10246, 2002.

[28] Bischoff F. A. and Valeev E. F. Low-order tensor approximations for electronic wave functions: Hartree–fock method with guaranteed precision. *J. Chem. Phys.*, 134:104104, 2011.

[29] 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.

[30] 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.

[31] 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.*, 136:214104, 2012.

[32] 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.

[33] Becke A. D. Density-functional exchange-energy approximation with correct asymptotic behavior. *Phys. Rev. A*, 38:3098–3100, Sep 1988.

[34] Lee C., Yang W., and Parr R. G. Development of the colle-salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B*, 37:785–789, Jan 1988.

[35] Johnson B. G., Gill P. M. W., and Pople J. A. The performance of a family of density functional methods. *J. Chem. Phys.*, 98(7):5612–5626, 1993.

[36] Dunning Jr. T. H. Gaussian basis functions for use in molecular calculations. i. contraction of (9s5p) atomic basis sets for the first-row atoms. *J. Chem. Phys.*, 53(7):2823–2833, 1970.

[37] Dunning Jr. T. H. Gaussian basis sets for the atoms gallium through krypton. *J. Chem. Phys.*, 66(3):1382–1383, 1977.

[38] Lsdalton, a linear scaling molecular electronic structure program, release dalton 2011, see http://daltonprogram.org/.

[39] Watson M. A. and Hirao K. A linear-scaling spectral-element method for computing electrostatic potentials. *J. Chem. Phys.*, 129(18):184107, Jan 2008.

[40] Levenberg K. A method for the solution of certain nonlinear problems in least squares. *Q. Appl. Math*, 2(164), 1944.

[41] Marquardt D. W. An algorithm for least squares estimation of nonlinear parameters. *J. Soc. Ind. Appl. Math*, 11(431), 1963.

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

Real-Space Density Functional Theory with Localized Orbitals and Multiwavelets

S. R. Jensen, J. Jusélius, A. Durdek, P. Wind, T. Flåand L. Frediani

Abstract

Real-space methods for quantum chemistry have been gaining popularity in recent years. Despite the still significant overhead with respect to more common Gaussian-Type Orbitals and plane waves, they possess very attractive properties, especially when combined with a rigorous mathematical framework such as Multiresolution Analysis and Multiwavelets. Among the distinctive features are a localized orthonormal basis with disjoint support which is ideal in combination with massively parallel computing architectures, full adaptivity for automatic local refinement around nuclei, vanishing moments for sparse representations of functions and operators, rigorous error control with respect to basis-set limit results (a "golden standard" of Quantum Chemistry). We have implemented a Self Consistent Field solver within the Multiwavelet framework for restricted and unrestricted Hartree-Fock and Density Functional Theory. Our solver is based on a preconditioned steepest descent step combined with a Krylov accelerator. Among the distinctive features are the use of localized orbitals throughout and the construction of the full Fock matrix without any reference to the kinetic energy. In our results we have shown that we are able to attain high accuracy for Density Functional Theory as well as Hartree-Fock, both for restricted closed shell cases as well as unrestricted open shell ones. Moreover we have found that the use of localized orbitals is highly beneficial for the SCF convergence of larger species, making the acceleration scheme less critical compared to canonical orbitals.

1 Introduction

Atom-centered Gaussians have traditionally been the most common and widespread choice of basis set for molecules[1]. Several strong arguments are in favor of such a choice: the compactness of the representation which is defined by a handful of coefficients, the ability to represent atomic orbitals well (Slater functions are in theory superior due to the cusp at the nuclear position and the correct asymptotic), the simplification in the computation of molecular integrals which are often obtained analytically (this is the weak point of Slater orbitals which require expensive

numerical evaluations). Their main disadvantage is the non-orthogonality of the basis which can become a severe problem especially for large bases leading to a computational bottleneck when orthonormalization is required or worse numerical instabilities due to near linear-dependency in the basis[2].

On the opposite side of the spectrum, plane waves are ideally suited for periodic systems and are orthonormal by construction. However a very large number of them needs to be employed in order to achieve good accuracy, especially if one is interested in high resolution in the nuclear-core regions[3]. A popular choice to circumvent the problem is to use pseudopotentials[4] in the core region, thereby reducing the number of electrons to be treated and at the same time removing the need for very high-frequency components. Another challenge is constituted by non-periodic systems, which can only be dealt with by using a supercell approach[5].

Quantum chemical modeling is constantly expanding its horizons: cutting edge research is focused on achieving good accuracy (either in energetics or molecular properties) on large non-periodic systems such as large biomolecules or molecular nanosystems. This progression is constantly exposing the weaknesses of the traditional approach thus rendering the use of unconventional methods, which are free from the above mentioned limitations ever more attractive. One such choice is constituted by numerical, real-space grid-based methods which are gaining popularity in quantum chemistry as a promising strategy to deal with the Self Consistent Field (SCF) problem of Hartree-Fock and Density Functional Theory.

Among real-space approaches three strategies have been commonly employed: Finite Differences, Finite Elements and Wavelets/Multiwavelets[6]. Among these methods, Multiwavelets are particularly well suited for all-electron calculations[7, 8]. The basis functions are localized (as Gaussian-type orbitals) yet orthonormal (as plane waves). One crucial property of Multiwavelets is the disjoint support (zero overlap) between basis functions in adjacent nodes[9], paving the way for adaptive refinement of the mesh, tailored to each given function. This is essential for an all-electron description where varying resolution is a prerequisite for efficiency. The price to pay is that, in order to provide a representation with a given number of vanishing moments, a corresponding number of basis functions must be employed. The most common choice of basis functions in the Multiwavelet framework is a generic orthonormal polynomial basis of order k, providing a second possibility to increase the resolution of the representation alongside the adaptive grid refinement[10]. Currently, the main drawbacks of this approach are a large memory footprint (a numerical representation of a molecular orbital is much larger in terms of number of coefficients), and a significant computational overhead[11, 12]. On the other hand, a localized orthonormal basis is an ideal match for modern massively-parallel architectures[13] and we are confident that it is only a matter of time before real-space grid methods in general and Multiwavelets in particular will become competitive with or even superior to traditional ones.

Adaptivity is an excellent strategy to achieve good accuracy, as expensive high-resolution functions on a fine grid are present locally only where necessary[14]. The challenge with such an approach is constituted by the consequent lack of a fixed basis set which instead is only present "on demand". This detail has a profound impact on the minimization strategies that can be adopted in order to solve SCF problems such as the Roothaan-Hall equations of Hartree-Fock (HF) or the Kohn-Sham (KS) equations of Density Functional Theory (DFT). In other words all such strategies which strictly depend on having a fixed basis, such as the most common atomic orbital based methods[15] are excluded. On the other hand, only the occupied molecular orbitals are strictly needed both in HF and DFT to describe the wavefunction/electronic density. Several strategies, where only the knowledge of the occupied molecular orbitals and the electronic density are needed to minimize the energy, can be employed. The most straightforward albeit not efficient choices are the Steepest Descent (SD) and the Conjugate Gradient (CG) methods [16]. A better approach is constituted by the computation of the lowest eigenpairs (eigenvalue/eigenfunction) of the Fock operator, such as in the Lanczos method[17] and the algorithm proposed by Davidson[18]. The drawback of such an approach is the focus on the canonical, delocalized orbitals, which prevent linear scaling of any algorithm.

Currently the most popular minimization scheme is constituted by the Direct Inversion of the Iterative Subspace (DIIS), originally proposed by Pulay[19] and later revised by Wood and Zunger[20] in their Residual Minimization Method. Other strategies have recently been employed such a preconditioned conjugate gradient method[21] and a Krylov method proposed by Harrison[22]. An analysis of DIIS and Krylov methods has been presented by Rohwedder and Schneider[23]. A general formalization of the problem has been laid out by Schneider *et al.* [24], who considered the HF and KS problems as a minimization problem subject to (1) orthonormality constraint and (2) invariance with respect to rotation among the occupied orbitals. By properly defining the admissible Grassmann manifold based on the two constraints above, they were able to make use of standard minimization techniques such as a preconditioned steepest descent which could then later be combined with any acceleration method such as the DIIS technique[19] or the Krylov method proposed by Harrison[22].

In the present work we have considered a preconditioned steepest descent approach, in combination with the Krylov acceleration and localized molecular orbitals. In Sec. 2 we briefly summarize the mathematical framework of Multiwavelets, whereas in Sec. 3 we introduce the formalism of the SCF method, which is then applied in Sec. 4 to develop the equations for the real-space SCF optimizer both with canonical and localized orbitals. In particular, in Sec. 4.1 we describe how the Fock matrix is computed circumventing the need to apply the kinetic energy operator at any stage and in Sec. 1 we present the details of our algorithm. In Sec. 5 we present some benchmark results for DFT, RHF and UHF, emphasizing the convergence of the algorithm, the accuracy of

the results and the scaling of the code. The final conclusions and future perspectives are presented in Sec. 6.

2 Functions and operators in the Multiwavelet framework

Multiwavelets are kind of wavelets where disjoint support between adjacent nodes is enforced[25], by allowing more than one function to be present in each interval. A common choice for the initial scaling basis is a set of Legendre polynomials as originally proposed by Alpert[10]. Successive refinements are obtained by dilation and translation:

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

which yields the usual ladder of scaling spaces:

$$V^0_k \subset V^1_k \subset \ldots \subset V^n_k \subset \ldots \tag{2}$$

The wavelet functions $\psi^n_{j,l}$ constitute the basis spanning two consecutive scaling spaces:

$$W^n_k \oplus V^n_k = V^{n+1}_k \tag{3}$$

Representation of functions is done by projection either on the scaling space at the finest scale $n+1$ (reconstructed representation) or on V^0 and the full ladder of wavelet spaces from W^0 to W^n (compressed representation). The two representations are equivalent and can be interconverted through the so-called Multiwavelet *filters*.

If we indicate with P^n and Q^n the projectors onto the scaling and wavelet spaces at scale n, respectively, then, it follows from the definition of the wavelet space that $P^{n+1} = P^n + Q^n$.

$$f^{n+1} \stackrel{\text{def}}{=} P^{n+1}f = (P^n + Q^n)f = f^n + \mathrm{d}f^n \tag{4}$$

where $\mathrm{d}f^n \stackrel{\text{def}}{=} Q^n f$.

The application of operators using a Multiwavelet basis can be performed using either the Standard form or the Non-Standard form. We have chosen the Non-Standard form [26, 27] as it virtually decouples scales from each other rendering the implementation of adaptive algorithms much simpler.

As shown in the following section, for SCF algorithms within a preconditioned steepest descent framework the necessary operators are the Poisson operator for the electrostatic potential, the Coulomb electronic energy and the quantistic exact exchange and the bound-state Helmholtz operator for the SCF iteration. Their Green's kernel can be written as

$$H^\mu(r - r') = \frac{e^{-\mu\|r-r'\|}}{4\pi\|r-r'\|} \tag{5}$$

, where $\mu > 0$ yields the bound-state Helmholtz kernel, whereas $\mu = 0$ is the Poisson kernel. Their application is achieved by convolution of a function with the corresponding Green's kernel

$$g(r) = [Tf](r) = \int G(r - r')f(r')\,\mathrm{d}r' \tag{6}$$

once an approximate separated form in terms of Gaussian functions has been computed[28, 26, 29]:

$$G(r - r') \approx \sum_{i=1}^{M} a_i e^{-\alpha_i (r-r')^2} \tag{7}$$

.

The Non-Standard form of the operator T is built as a telescopic expansion of the finest scale projection $T^N = P^N T P^N$

$$T^N = T^0 + \sum_{n=0}^{N-1} (A^n + B^n + C^n) \tag{8}$$

where $A^n = Q^n T Q^n$, $B^n = Q^n T P^n$, $C^n = P^n T Q^n$. Thanks to the vanishing moments of the Multiwavelet basis, the matrix representations of A^n, B^n and C^n are sparse and diagonally dominant for the Poisson and bound-state Helmholtz kernels. Therefore all terms beyond a predetermined bandwidth can be omitted in the operator application, without a loss of accuracy. In particular, we have shown that the application of the Poisson operator for the calculation of the electrostatic potential scales linearly with the size of the system [30].

3 Self Consistent Field methods

In the SCF approximations of an N-electron molecular system, the $3N$-dimensional wavefunction is expressed in terms of N three-dimensional one-electron spinorbitals $\{\phi_i\}_{i=1}^N$, in the form of a Slater determinant $\Phi(\langle\phi_1 \dots \phi_N\rangle)$. The spinorbitals are a solution of the HF or KS equations, if they a minimize the HF or KS energy functional $\mathcal{I}^{\mathrm{HF/KS}}(\Phi)$. Since Slater determinants are invariant with respect to unitary transformations among the spinorbitals, this redundancy is traditionally exploited to write the problem in the so-called canonical form

$$\hat{F}|\phi_i\rangle = \epsilon_i|\phi_i\rangle \tag{9}$$

in which the Fock matrix is diagonal

$$F_{ij} = \langle\phi_i|\hat{F}|\phi_j\rangle = \delta_{ij}\epsilon_i \tag{10}$$

where $\hat{F}$ is the Fock operator. Solving these equations will give the natural, delocalized molecular orbitals that are eigenfunctions of the given Fock operator.

As the SCF energy is invariant among rotations in the occupied orbital space, the more general set of explicitly coupled equations can be written as:

$$\hat{F}|\phi_i\rangle = \left[\sum_j |\phi_j\rangle\langle\phi_j|\right] \hat{F}\,|\phi_i\rangle = \sum_j F_{ji}\,|\phi_j\rangle \tag{11}$$

This invariance can be exploited in several ways, e.g. to achieve orbital localization in space, thus leading to more compact real-space representations, faster convergence for larger systems and prospects of low-scaling algorithms.

3.1 Kohn-Sham equations

In Kohn-Sham DFT, the electron density is given in terms of the occupied orbitals, assuming a closed-shell system

$$\rho(r) = 2\sum_{i=1}^{N/2} |\phi_i(r)|^2 . \tag{12}$$

Minimizing the DFT energy functional with respect to orbital variations (under the constraints that the orbitals remain orthonormal and the density integrates to the number of electrons) is equivalent to finding the $N/2$ lowest energy eigenfunctions of the Kohn-Sham operator

$$\hat{F} = -\frac{1}{2}\nabla^2 + V_{eff}(r), \tag{13}$$

where the effective potential have three contributions $V_{eff} = V_{nuc} + V_{el} + V_{xc}$. The electron-nuclear attraction is described by the nuclear potential

$$V_{nuc}(r) = -\sum_I \frac{Z_I}{|r - R_I|}, \tag{14}$$

the electron-electron repulsion is given through the electronic potential

$$V_{el}(r) = \int \frac{\rho(r')}{|r - r'|}\,\mathrm{d}r', \tag{15}$$

while all non-classical effects are included in the exchange-correlation potential V_{xc}, which is commonly approximated as a scalar function that depends only upon the density (LDA) or upon the density and its gradient (GGA) and higher order derivatives (meta-GGA).

The non-canonical Kohn-Sham equations are thus given as

$$\Big[-\frac{1}{2}\nabla^2 + V_{nuc}(r) + V_{el}(r) + V_{xc}(r)\Big]|\phi_i\rangle = \sum_j F_{ji}|\phi_j\rangle \tag{16}$$

and as both the electronic and exchange-correlation potentials depend upon the orbitals through the electron density, we have a set of coupled, non-linear equations.

3.2 Hartree-Fock equations

In Hartree-Fock theory the multiplicative exchange-correlation potential of Kohn-Sham theory is replaced by the non-multiplicative exchange operator $\hat{K}$, and the Fock operator is given as

$$\hat{F} = -\frac{1}{2}\nabla^2 + V_{nuc}(r) + V_{el}(r) - \hat{K}, \tag{17}$$

where the nuclear and electronic potentials are the same as in Eqs. (14) and (15), respectively, for the Kohn-Sham case. The exchange operator is defined through its effect on an orbital and can be written (for a closed-shell system) as[7]

$$\hat{K}\phi_i(r) = \sum_{j=1}^{N/2} \phi_j(r) \int \frac{\phi_i(r')\phi_j(r')}{|r-r'|}\,\mathrm{d}r' \tag{18}$$

and we get the non-canonical Hartree-Fock equations

$$\Big[-\frac{1}{2}\nabla^2 + V_{nuc}(r) + V_{el}(r) - \hat{K}\Big]|\phi_i\rangle = \sum_j F_{ji}|\phi_j\rangle \tag{19}$$

which again are non-linear through the electronic potential and exchange operator.

3.3 Integral formulation

The similarity between the Hartree-Fock and Kohn-Sham equations allows for the same solution algorithms to be used in both cases, and to get a unified description of we introduce the potential operator $\hat{V}$, defined as

$$\hat{V}^{HF} = V_{nuc}(r) + V_{el}(r) - \hat{K} \tag{20}$$

$$\hat{V}^{KS} = V_{nuc}(r) + V_{el}(r) + V_{xc}(r) \tag{21}$$

in Hartree-Fock and Kohn-Sham theory, respectively. Following Kalos[31] and Harrison *et al.* [7], we can rewrite the non-canonical equations in terms of the Helmholtz integral operator of Eq. (5) in the following way

$$\Big[-\frac{1}{2}\nabla^2 + \hat{V}\Big]\phi_i(\boldsymbol{r}) = \sum_j F_{ji}\phi_j(\boldsymbol{r}) \tag{22}$$

$$\big[-\nabla^2 - 2\lambda\big]\phi_i(\boldsymbol{r}) = -2\Big[\hat{V}\phi_i(\boldsymbol{r}) + \sum_j \big(\lambda\delta_{ij} - F_{ji}\big)\phi_j(\boldsymbol{r})\Big]. \tag{23}$$

By imposing that $\mu = \sqrt{-2\lambda}$ and by making use of the Green's function definition $(-\nabla^2 - 2\lambda)H^\mu(r-r') = \delta(r-r')$:

$$\phi_i(r) = -2\int H^\mu(r-r')\Big[\hat{V}\phi_i(r') + \sum_j \big(\lambda\delta_{ij} - F_{ji}\big)\phi_j(r')\Big]\,\mathrm{d}r' \tag{24}$$

$$\phi_i = -2\hat{H}^\mu\Big[\hat{V}\phi_i + \sum_j \big(\lambda\delta_{ij} - F_{ji}\big)\phi_j\Big] \tag{25}$$

This general expression can be simplified. In the canonical orbital basis, the Fock matrix becomes diagonal $F_{ji} = \epsilon_i\delta_{ij}$ and the orbital equations become formally decoupled:

$$\phi_i = -2\hat{H}^\mu\Big[\hat{V}\phi_i + \big(\lambda - \epsilon_i\big)\phi_i\Big] \tag{26}$$

By choosing $\lambda = \epsilon_i$, each orbital equation is further simplified:

$$\phi_i = -2\hat{H}^{\mu_i}\Big[\hat{V}\phi_i\Big] \tag{27}$$

with $\mu_i = \sqrt{-2\epsilon_i}$. We underline that, as for the standard differential formulations, the equations are still implicitly coupled through the potential operator $\hat{V}$, which is defined in terms of the orbitals: the solution must therefore be sought self-consistently by iterative methods. Finally, it is important to notice that both the orbitals ϕ_i and their corresponding energy ϵ_i are unknowns in the equations, and must be determined simultaneously.

The choice of orbital basis (canonical or not) as well as the λ parameter that appears in the Helmholtz operator will affect the numerical stability and the rate of convergence of the iterative solution algorithms that are presented in the following section.

4 Iterative solution algorithm

For simple one-orbital systems, the integral equation (27) can easily be brought to convergence up to arbitrary accuracy by straightforward iteration

$$\tilde{\phi}^{n+1} = -2\hat{H}^{\mu^n}\left[\hat{V}\phi^n\right] \tag{28}$$

$$\phi^{n+1} = \frac{\tilde{\phi}^{n+1}}{\|\tilde{\phi}^{n+1}\|} \tag{29}$$

where the Helmholtz operator is reconstructed in each iteration using the latest approximation of the orbital energy $\mu^n = \sqrt{-2\epsilon^n}$. The energy update can be approximated to first order at each iteration[7]

$$\epsilon^{n+1} = \epsilon^n - \frac{\langle \hat{V}\phi^n | \phi^n - \tilde{\phi}^{n+1} \rangle}{\|\tilde{\phi}^{n+1}\|^2} \tag{30}$$

This energy update is calculated very efficiently since the product $\hat{V}\phi^n$ is a byproduct of the iteration in Eq. (28), and inner products are efficiently evaluated in the orthonormal Multiwavelet basis.

For many-electron systems it is necessary to enforce orthonormality between the occupied orbitals in order to arrive at a true Aufbau solution of the HF/KS equations, as a straightforward iteration of Eq. (27) would bring all orbitals to the lowest eigenfunction.

The simplest approach to keep orthonormality would be to apply Eq. (27) for each orbital followed by a Gram-Schmidt orthogonalization in order of increasing energy. This would however lead to very slow convergence, especially for valence orbitals, as the convergence of each orbital is restrained by the level of convergence of lower-lying orbitals, whose convergence is again dependent on the accuracy of *all* orbitals through the potential energy operator $\hat{V}$.

Harrison *et al.* [7] described how to use deflation to extract multiple eigenpairs from the Fock operator by recasting the equation for each orbital as a ground state problem. Another approach, which was also suggested by Harrison *et al.* , is to diagonalize the Fock matrix at each iteration. By comparing a simple iteration of Eq. (28) with the general expression in Eq. (25), the off-diagonal

coupling terms are missing unless the Fock matrix is diagonal, and neglecting these will severely hamper the convergence for systems with many orbitals.

To keep a satisfactory convergence one can either include the off-diagonal elements and iterate Eq. (25) instead (using $\lambda_i^n = F_{ii}^n$)

$$\tilde{\phi}_i^{n+1} = -2\hat{H}^{\mu_i^n}\Big[\hat{V}^n\phi_i^n - \sum_{j\neq i} F_{ji}^n\phi_j^n\Big], \tag{31}$$

or one can diagonalize the Fock matrix in each iteration. In the latter case Eq. (28) becomes exact. In the following we will describe both approaches, where specifically we have the choice in each iteration to either diagonalize the Fock matrix [1] and obtain the canonical molecular orbitals (CMO), or to obtain localized molecular orbitals (LMO), in which case the Fock matrix is not diagonal. In either case we need to calculate the Fock matrix, and in the following we show how this can be done with no approximations without applying the kinetic energy operator.

4.1 Calculation of Fock matrix

The starting point is a set of orthonormal orbitals $\{\phi_i^n\}$ and an initial guess for the corresponding Fock matrix $F_{ij}^n \approx \langle\phi_i^n|\hat{F}|\phi_j^n\rangle$. We emphasize that such a guess need not to be the exact Fock matrix for the given orbital set. The new and now exact Fock matrix $\tilde{F}_{ij}^{n+1} = \langle\tilde{\phi}_i^{n+1}|\hat{F}|\tilde{\phi}_j^{n+1}\rangle$ in the non-orthonormal basis obtained by applying Eq. (31) (λ can formally be chosen arbitrarily) is computed without any reference to the kinetic energy operator.

As will be shown shortly, for the definition of the new Fock matrix to be consistent, the potential operator needs to be constructed *after* orthonormalization of the new orbitals. This requires a temporary set of orbitals $\{\bar{\phi}_i\}$, constructed e.g. through a Gram-Schmidt process, so that $\langle\bar{\phi}_i|\bar{\phi}_j\rangle = \delta_{ij}$. In this basis we calculate the potentials V_{el} and V_{xc} (in the case of DFT) through the density

$$\rho^{n+1}(r) = 2\sum_i^{N/2} |\bar{\phi}_i^{n+1}(r)|^2, \tag{32}$$

and exchange (in the case of Hartree-Fock) as

$$\hat{K}^{n+1}\tilde{\phi}_i^{n+1} = \sum_j^{N/2} \bar{\phi}_j^{n+1}(r) \int \frac{\tilde{\phi}_i^{n+1}(r')\bar{\phi}_j^{n+1}(r')}{|r-r'|}\,\mathrm{d}r'. \tag{33}$$

In order to avoid the application of the kinetic energy operator we exploit the formal definition of the bound-state Helmholtz operator as the inverse of the level-shifted Laplacian:

$$\left(\hat{T} - \lambda_i^n\right)^{-1} = 2\hat{H}^{\mu_i^n}. \tag{34}$$

[1] Since we are working only with occupied orbitals, the matrix diagonalization is considerably cheaper than the corresponding diagonalization using high-quality Gaussian basis sets, where the number of virtual orbitals can be much larger than the number of occupied orbitals.

The application of $(\hat{T} - \lambda_i^n)$ to the new orbital will return the argument from the Helmholtz operator (provided that $\mu_i^n = \sqrt{-2\lambda_i^n}$ was used in this operator):

$$\big(\hat{T} - \lambda_i^n\big)\tilde{\phi}_i^{n+1} = -\Big[\hat{V}\phi_i^n + \sum_j \big(\Lambda_{ji}^n - F_{ji}^n\big)\phi_j^n\Big], \tag{35}$$

where we have defined the diagonal matrix $\Lambda_{ji}^n = \lambda_i^n \delta_{ji}$. Before we proceed we also define the updates in the orbitals and in the potential

$$\Delta\tilde{\phi}_i^n = \tilde{\phi}_i^{n+1} - \phi_i^n, \tag{36}$$

$$\Delta\hat{V}^n = \hat{V}^{n+1} - \hat{V}^n. \tag{37}$$

We can now use the above observations to eliminate the kinetic operator from the calculation of the Fock matrix.

$$\begin{aligned}
\tilde{F}_{ij}^{n+1} &= \langle\tilde{\phi}_i^{n+1}|\hat{T} + \hat{V}^{n+1}|\tilde{\phi}_j^{n+1}\rangle \\
&= \langle\tilde{\phi}_i^{n+1}|\hat{T} - \lambda_j^n|\tilde{\phi}_j^{n+1}\rangle + \langle\tilde{\phi}_i^{n+1}|\hat{V}^{n+1} + \lambda_j^n|\tilde{\phi}_j^{n+1}\rangle \\
&= \langle\tilde{\phi}_i^{n+1}| - \Big[\hat{V}^n\phi_j^n + \sum_k \big(\Lambda_{kj}^n - F_{kj}^n\big)\phi_k^n\Big]\rangle + \langle\tilde{\phi}_i^{n+1}|\hat{V}^{n+1} + \lambda_j^n|\tilde{\phi}_j^{n+1}\rangle \\
&= -\langle\tilde{\phi}_i^{n+1}|\hat{V}^n|\phi_j^n\rangle - \sum_k \langle\tilde{\phi}_i^{n+1}|\phi_k^n\rangle\big(\Lambda_{kj}^n - F_{kj}^n\big) + \langle\tilde{\phi}_i^{n+1}|\hat{V}^{n+1}|\tilde{\phi}_j^{n+1}\rangle + \langle\tilde{\phi}_i^{n+1}|\tilde{\phi}_j^{n+1}\rangle\lambda_j^n \\
&= \langle\tilde{\phi}_i^{n+1}|\Delta\hat{V}^n|\phi_j^n\rangle + \langle\tilde{\phi}_i^{n+1}|\hat{V}^n|\Delta\tilde{\phi}_j^n\rangle + \sum_k \langle\tilde{\phi}_i^{n+1}|\phi_k^n\rangle\big(F_{kj}^n - \Lambda_{kj}^n\big) + \langle\tilde{\phi}_i^{n+1}|\tilde{\phi}_j^{n+1}\rangle\lambda_j^n \\
&= \langle\tilde{\phi}_i^{n+1}|\Delta\hat{V}^n|\phi_j^n\rangle + \langle\tilde{\phi}_i^{n+1}|\hat{V}^n|\Delta\tilde{\phi}_j^n\rangle + \sum_k \langle\phi_i^n + \Delta\tilde{\phi}_i^n|\phi_k^n\rangle F_{kj}^n + \langle\tilde{\phi}_i^{n+1}|\Delta\tilde{\phi}_j^n\rangle\lambda_j^n \\
&= \langle\tilde{\phi}_i^{n+1}|\Delta\hat{V}^n|\phi_j^n\rangle + \langle\tilde{\phi}_i^{n+1}|\hat{V}^n|\Delta\tilde{\phi}_j^n\rangle + \big(S^n F^n\big)_{ij} + \big(\Delta\tilde{S}_1^n F^n\big)_{ij} + \big(\Delta\tilde{S}_2^n \Lambda^n\big)_{ij}
\end{aligned}$$

where S^n is the overlap matrix (assumed identity). Finally, we see that we can calculate the new Fock matrix with no approximations by evaluating three update terms

$$\tilde{F}^{n+1} = F^n + \Delta\tilde{S}_1^n F^n + \Delta\tilde{S}_2^n \Lambda^n + \Delta\tilde{F}_{pot}^n \tag{38}$$

where we have defined two updates involving the overlap matrix

$$\big(\Delta\tilde{S}_1^n\big)_{ij} = \langle\Delta\tilde{\phi}_i^n|\phi_j^n\rangle \tag{39}$$

$$\big(\Delta\tilde{S}_2^n\big)_{ij} = \langle\tilde{\phi}_i^{n+1}|\Delta\tilde{\phi}_j^n\rangle \tag{40}$$

and one update involving the potential operator

$$\big(\tilde{F}_{pot}^n\big)_{ij} = \langle\tilde{\phi}_i^{n+1}|\hat{V}^n|\Delta\tilde{\phi}_j^n\rangle + \langle\tilde{\phi}_i^{n+1}|\Delta\hat{V}^n|\phi_j^n\rangle \tag{41}$$

The above expression is exact, but can be approximated and simplified while keeping track of the approximation order (e.g. the two overlap updates are equal to first order). As already mentioned,

the new potential should ideally be evaluated in a temporary orthonormal set of orbitals, but to avoid a costly intermediate orthogonalization one can choose to only normalize the orbitals and calculate the new potential operator in a non-orthogonal basis. While this is usually a fair approximation, we have observed a significant loss of convergence for certain problematic cases (in particular for some unrestricted open-shell Hartree-Fock calculations presented in Sec. 5).

We want to emphasize that these expressions *require* that the orbitals of the new set $\{\tilde{\phi}_i^{n+1}\}$ are related to the orbitals of the old set $\{\phi_i^n\}$ exactly through the application of the Helmholtz operator in Eq. (31). Otherwise the application of the kinetic energy operator cannot be avoided to obtain the Fock matrix.

4.2 Orbital orthogonalization

As already mentioned, the basic iteration of the integral operators in Eq. (31) to all orbitals $\{\phi_i\}$ does not preserve the orthonormality of the orbitals. This could be achieved by making use of an exponential parameterization[1, 24], but we have not pursued this strategy yet. At each iteration, orthonormalization needs to be restored explicitly. This is done in combination with a unitary transformation that either brings the equations to canonical form, or that localizes the orbitals in space.

Canonical orbitals

The orbital orthonormalization and Fock matrix diagonalization can be collected into a single orbital transformation. With the usual definition of the overlap matrix

$$\tilde{S}_{ij} = \langle \tilde{\phi}_i | \tilde{\phi}_j \rangle \tag{42}$$

the orbitals can be orthonormalized through the transformation

$$\bar{\phi}_i = \sum_{j=1}^{N} \tilde{S}_{ij}^{-1/2} \tilde{\phi}_j, \qquad \langle \bar{\phi}_i | \bar{\phi}_j \rangle = \delta_{ij}. \tag{43}$$

The Fock matrix in the non-orthonormal basis reads: $\tilde{F}_{ij} = \langle \tilde{\phi}_i | T + \hat{V} | \tilde{\phi}_j \rangle$. Its transformation to the orthonormal basis is achieved as $\bar{F} = \tilde{S}^{-1/2} \tilde{F} \tilde{S}^{1/2}$. By calling M_X the unitary transformation to the requested basis ($X = C, L$ for canonical or localized basis respectively) the overall transformation matrix becomes $U = M^T \tilde{S}^{-1/2}$.

Localized orbitals

Instead of diagonalizing the Fock matrix, it is beneficial for bigger systems to work with localized molecular orbitals (LMOs), both in terms of the familiar prospects of low-scaling algorithms of

conventional SCF implementations[32], but also to reduce the large storage requirements of real-space methods.

Following Yanai *et al.* [8], we use the Foster-Boys[33, 34] algorithm for orbital localization, where the unitary matrix is calculated based on the one-electron dipole integrals $\langle\phi_i|\boldsymbol{r}|\phi_j\rangle$. More specifically, the dipole integrals are calculated in the non-orthogonal orbitals $\{\tilde{\phi}_i\}$ and then orthonormalized using the $\tilde{S}^{-1/2}$ matrix before the unitary matrix M_L that localize the orbitals is calculated using an iterative non-linear optimizer[35].

With the formulation of the HF/KS equations given in Eq. (31) it is possible to work exclusively in the LMO basis throughout the SCF optimization. This is in contrast to the algorithm proposed by Yanai *et al.* [8], which relies on CMOs for the application of the Helmholtz operator, and low-scaling calculation of HF exchange is obtained by first transforming the CMOs into LMOs, then apply the exchange operator, before transforming the result back to the CMO basis.

It is in general not necessary to localize the orbitals in every iteration, as the new orbitals will only be slightly perturbed from the old ones, and a simple orthonormalization using the $\tilde{S}^{-1/2}$ matrix will to a large extent keep the localization of the old orbitals. However, in this case the calculation of the M_L matrix is very efficient requiring only a handful of iterations, keeping the computational overhead of the extra localization low, compared to the actual orthonormalization and rotation of the orbitals, which is indeed rather inefficient in real-space bases (although it should become increasingly diagonally dominant as the SCF iteration proceeds).

4.3 Krylov subspace accelerated inexact Newton method

By collecting the orbital vector $\Phi = (\phi_0, \ldots, \phi_N)^T$ and the Fock matrix F into a new vector $\boldsymbol{x} = (\Phi, F)$, the SCF problem in Eq. (25) can be viewed as finding the roots of the following residual function

$$f(\boldsymbol{x}) = -2\hat{H}^{\mu}\big[\hat{V}\Phi + \big(\Lambda - F\big)\Phi\big] - \Phi. \tag{44}$$

At a given iteration n, we have the current approximation $\boldsymbol{x}^n = (\Phi^n, F^n)$ and the corresponding residual $f(\boldsymbol{x}^n) = (\Delta\Phi^n, \Delta F^n)$. In the Krylov subspace accelerated inexact Newton (KAIN)[22] method the new update $\delta\boldsymbol{x}^n$ is calculated in terms of the m latest iterations

$$\delta\boldsymbol{x}^n = f(\boldsymbol{x}^n) + \sum_{j=(n-m)}^{n-1} c_j\big[\,\big(\boldsymbol{x}^j - \boldsymbol{x}^n\big) + \big(f(\boldsymbol{x}^j) - f(\boldsymbol{x}^n)\big)\,\big], \tag{45}$$

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

$$A_{ij} = \langle \boldsymbol{x}^n - \boldsymbol{x}^i | f(\boldsymbol{x}^n) - f(\boldsymbol{x}^j)\rangle, \tag{46}$$

$$b_i = \langle \boldsymbol{x}^n - \boldsymbol{x}^i | f(\boldsymbol{x}^n)\rangle. \tag{47}$$

The Frobenius inner product is employed for the Fock matrix. The size m of the Krylov subspace is without constraints.

We conclude this section with a few important aspects to consider with the KAIN accelerator with more than one orbital:

1. As the orbital energies are updated in each iteration, it might occur that orbitals switch place during the optimization process, if ordered by increasing energy. This means that one must make sure that each orbital is linked to the correct orbital in the iterative history.

2. In the case of degeneracies, the matrix diagonalization will not result in a unique set of orbitals, as any basis for the degenerate subspace can be obtained. It is thus important to keep a consistent definition of all orbitals as the iteration proceeds. This and the previous issue can be treated simultaneously by always applying the same orbital rotation M_D that diagonalized the Fock matrix to the entire iterative history.

3. With localization there is a consistency in the definition of the orbital history unless there is an infinite rotational symmetry axis in the molecule, which makes the localization minimum non-unique. Otherwise the algorithm will always fall back to the same minimum, as long as the Helmholtz step does not bring the orbitals too far from equilibrium.

4. The KAIN updates do not conserve the orthogonality between the orbitals, and a second orthogonalization process can be added at the end of the cycle, although this is not strictly necessary to achieve convergence.

4.4 Algorithm

Finally, we put all the pieces together to a general algorithm for the SCF optimization for many-electron systems. Starting from an arbitrary initial guess for the orbitals and the Fock matrix, we calculate the electron density and corresponding potentials. The Helmholtz operator is applied once to each orbital, and we judge the convergence by the norm of the orbital update at this point. The potential update is calculated using an intermediate orthonormalized orbital set, and the new Fock matrix is calculated as described in Sec. 4.1. The appropriate transformation matrix U for Fock matrix diagonalization or orbital localization is obtained and applied to the orbitals and Fock matrix. The orbital and Fock matrix residuals are calculated *after* the orbital orthonormalization and rotation described in Sec. 4.2

$$\Delta\Phi^n = \Phi^{n+1} - \Phi^n = U\tilde{\Phi}^{n+1} - \Phi^n, \tag{48}$$

$$\Delta F^n = F^{n+1} - F^n = U\tilde{F}^{n+1}U^{-1} - F^n. \tag{49}$$

and then added to the KAIN history

$$\boldsymbol{x}^n = (\Phi^n, F^n) \qquad f(\boldsymbol{x}^n) = (\Delta\Phi^n, \Delta F^n)\,. \tag{50}$$

If the length of the history exceeds some modest number the oldest vector is discarded. New updates are then calculated based on Eq. (45)

$$\delta \boldsymbol{x}^n = (\delta\Phi^n, \delta F^n) \tag{51}$$

and subsequently added to the previous guess. The orbitals are then (optionally) orthonormalized. This sequence is iterated until the maximum norm among the orbital updates (after the Helmholtz operator application) is below some predefined threshold. The whole procedure is summarized in Algorithm 1.

Algorithm 1 Iterative SCF optimization of many-electron systems.

Given initial orbitals ϕ_i^0 and Fock matrix F^0
while $\varepsilon > \varepsilon_r$ **do**
  Calculate electron density through Eq.(12)
  Calculate potential operator $\hat{V}^n$ through Eq.(20) or Eq. (21)
  for each orbital i **do**
    if F_{ii}^n differs significantly from λ_i^{n-1} **then**
      Set $\lambda_i^n = F_{ii}^n$ and reconstruct Helmholtz operator
    else
      Set $\lambda_i^n = \lambda_i^{n-1}$ and reuse Helmholtz operator
    end if
    Calculate Helmholtz argument $\varphi_i^n = \hat{V}^n\phi_i^n + \sum_j \left(\lambda_i^n\delta_{ji} - F_{ji}^n\right)\phi_j^n$
    Apply Helmholtz operator $\tilde{\phi}_i^{n+1} = -2\hat{H}^{\mu_i^n}\left[\varphi_i^n\right]$
    Calculate orbital update $\Delta\tilde{\phi}_i^n = \tilde{\phi}_i^{n+1} - \phi_i^n$
  end for
  Determine maximum orbital error $\varepsilon = \max_i \|\Delta\tilde{\phi}_i^n\|$
  Calculate potential operator update $\Delta\hat{V}^n = \hat{V}^{n+1} - \hat{V}^n$
  Calculate Fock matrix through Eq. (38)
  Calculate rotation matrix U to diagonalize or localize
  Calculate KAIN residual $f(\boldsymbol{x}^n)$ through Eqs. (48) and (49)
  Calculate KAIN updates $\delta\boldsymbol{x}^n$ through Eq. (45)
  Update orbitals $\phi_i^{n+1} = \phi_i^n + \delta\phi_i^n$
  Update Fock matrix $F^{n+1} = F^n + \delta F^n$
  Orthonormalize orbitals
end while

5 Results

The performance of the code has been tested focusing in particular on convergence rate and accuracy for small- and medium-sized molecular systems. Two accuracy parameters ε_r and ε_{tot} have been considered: the former denotes the convergence threshold in the iterative algorithm, and the latter is the overall accuracy of the underlying function representations and mathematical operations.

All DFT calculations are performed using the standard Local Density Approximation (LDA), consisting of the Slater-Dirac[36, 37] exchange and the VWN5[38] correlation potentials. The exchange-correlation potentials are provided by the XCFun library[39], and we use the same smoothed nuclear potential as described by Yanai *et al.* [7].

For all calculations, the initial guess is obtained from conventional Gaussian basis calculations using the LSDalton[32, 40] program with a 3-21G[41, 42] basis set. All calculations have been performed on a single compute node containing 2×8 cores Intel Xeon E5-2670 processors with 16 GB memory (128 GB for the most demanding calculations).

5.1 Convergence

The convergence rate of the algorithms presented in Sec. 4 for one- and many-electron systems has been investigated with respect to three parameters: the size of the iterative subspace in the KAIN method, the choice of orbital transformation (canonical vs. localized), and the choice of λ in Eq. (31).

5.1.1 Hydrogen atom

The Hydrogen atom is a simple one-electron system, where the Schrödinger equation can be solved by straightforward power iteration of Eqs. (28)-(30). The potential operator $\hat{V}$ contains only the nuclear potential and does not depend on the wavefunction.

As can be seen from Fig.1, the convergence of the wavefunction and the corresponding energy of the power iteration is remarkably uniform: the norm of the wavefunction update is almost exactly halved between each iteration, while the energy update is divided by 4, showing that the error in the energy is quadratic in the error of the wavefunction. The overall accuracy is kept to $\varepsilon_{tot} = 10^{-10}$ throughout the iterations.

The convergence of the Hydrogen atom is significantly improved when Krylov subspace acceleration ($m = 4$) is included, with a threefold reduction in the number of iterations required to achieve a given accuracy in the wavefunction. The error in the energy is however no longer quadratic with respect to the wavefunction error, since both the wavefunction and the energy are included in the KAIN subspace vector, leading to a comparable accuracy for both.

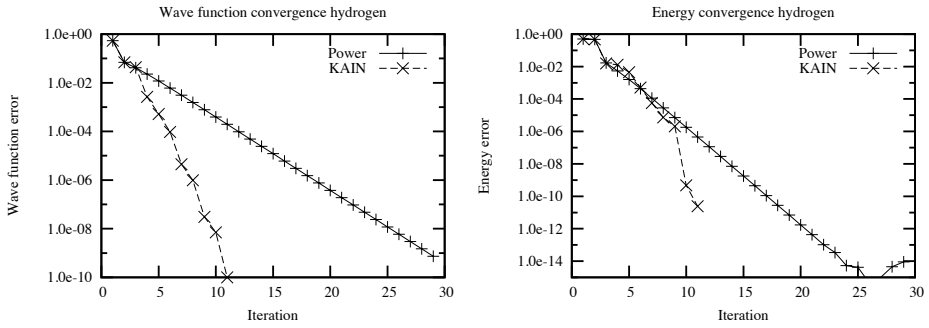


Figure 1: Convergence (relative error) of the Hydrogen wavefunction and total energy for a simple power iteration and using Krylov subspace acceleration with $m = 4$. The overall accuracy is kept at $\varepsilon_{tot} = 10^{-10}$.

5.1.2 Many-electron systems

Using canonical orbitals, we iterate the separated orbital equations

$$\tilde{\phi}_i^{n+1} = -2\hat{H}^{\mu_i^n}\left[\hat{V}^n\phi_i^n\right]. \tag{52}$$

provided that the Fock matrix is diagonalized in each iteration and the Helmholtz operator is updated using the latest orbital energy $\lambda_i^n = \epsilon_i^n$. This corresponds to the algorithm proposed by Harrison *et al.* [7].

The convergence of the canonical orbitals and the total energy of methane is presented in Fig. 2, where the overall accuracy is kept at $\varepsilon_{tot} = 10^{-8}$ throughout the iterations. Again we observe linear convergence, albeit somewhat slower than the Hydrogen atom: the error is each time roughly reduced by a factor of 1.6 instead of 2. The total energy shows similar convergence, and the error in the energy lies consistently below the corresponding error in the orbitals by about two orders of magnitude. With the KAIN method ($m = 4$) a significant improvement in the orbital convergence is achieved and the error in the energy is consistent with the error in the orbitals.

5.1.3 Convergence acceleration and orbital localization

The size of the iterative subspace m used in the KAIN algorithm affects the convergence for many-electron systems. In Tab. 1 the number of iterations required to reach convergence ($\varepsilon_r \leq 10^{-4}$ for all orbitals) for some of the smallest linear alkanes (C_nH_{2n+2}, $n = 1, 24, 6, 8, 10$) is reported as a function of m ($m = 0$ reduces to the power method) at the LDA level of theory. The first series of data refers to calculations performed with canonical orbitals and the second one to localized orbitals as described in Sec. 4.2.

For canonical orbitals, the straightforward iteration of Eq. (52) followed by a Fock matrix diagonalization displays poor convergence as the size of the system increases. However, a modest

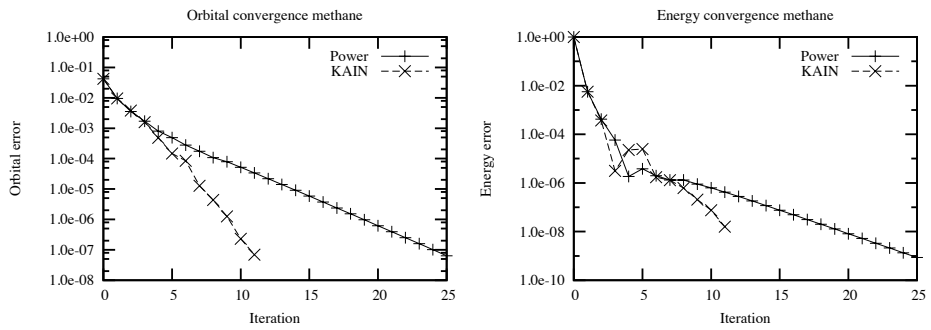


Figure 2: Convergence (relative error) of the maximum orbital residual and total energy of the methane molecule by power iteration of the canonical orbitals, and using Krylov subspace acceleration with $m = 4$. The overall accuracy is kept at $\varepsilon_{tot} = 10^{-8}$.

number ($m = 2 - 4$) of the iterative history improves convergence significantly. A larger iterative subspace $m > 4$ has practically no effect on the convergence.

With localized orbitals, the more general Eq. (31) is employed, followed by an orbital localization. The final energies are observed to be consistent with the canonical energies within the truncation threshold. Without acceleration ($m = 0$) the smaller molecules display a rate of convergence similar to the canonical orbitals. However, the number of iterations does not increase significantly for larger systems. When acceleration is included we observe an improvement until $m = 2$. Overall, using localized orbitals with $m = 2$ is slightly better than using canonical orbitals with $m = 4$.

In conclusion, for atoms and small molecules using canonical or localized orbitals is almost equivalent when KAIN is switched on: canonical orbitals converge in fewer iterations but require a larger subspace ($m = 4$) whereas localized orbitals require 2-3 additional iterations but a smaller subspace $m = 2$. For larger systems, orbital localization yields faster convergence even with a smaller subspace $m = 2$.

5.1.4 Fixed λ

As Eq. (52) can be viewed as a preconditioned steepest descent iteration, it is interesting to investigate how the convergence is affected by the choice of the parameter λ entering the preconditioner $\hat{H}^{\mu}$.

We have studied the convergence when a *fixed* λ value is employed in the preconditioner throughout the iterations. The canonical orbital equations then take the form:

$$\tilde{\phi}_i^{n+1} = -2\hat{H}^{\mu_i^n}\Big[\hat{V}^n\phi_i^n + \big(\lambda - \epsilon_i^n\big)\phi_i^n\Big]. \tag{53}$$

For the Hydrogen atom the exact energy is known ($E = 1/2$ Hartrees). This value will then be used as a reference, expressing λ relative to it. The number of iterations necessary to converge

Table 1: Number of iterations required to bring the orbital residual to $\epsilon_r \leq 10^{-4}$ for different lengths of KAIN history ($m = 0$ corresponding to regular power iteration) using canonical and localized orbitals.

		Size m of KAIN history							
Molecule	N orbitals	0	1	2	3	4	5	6	7
		Canonical orbitals							
C_1H_4	5	8	8	7	6	6	6	6	6
C_2H_6	9	9	8	7	7	7	7	7	7
C_4H_{10}	17	17	15	10	9	10	10	10	10
C_6H_{14}	25	36	23	13	12	11	11	11	11
		Localized orbitals							
C_1H_4	5	9	9	7	7	7	7	7	7
C_2H_6	9	10	10	8	8	8	8	8	8
C_4H_{10}	17	13	12	9	9	9	9	9	9
C_6H_{14}	25	14	11	9	10	9	9	9	9
C_8H_{18}	33	11	10	8	9	8	8	8	8
$C_{10}H_{22}$	41	11	11	9	9	8	8	8	8

Eq. (53) up to a residual norm of $\varepsilon_r \leq 10^{-7}$ without any subspace acceleration, is presented in Fig. 3 as a function of λ/E. We underline that $\lambda = E$ does not mean that the last term in Eq. (53) vanishes in all iterations; however this term approaches zero as the eigenvalue ϵ_0^n approaches the true energy. The value corresponding to the "dynamic" algorithm (the Helmholtz operator is updated in each iteration as in Eq. (52)) is denoted in the figure by a dashed line.

Surprisingly, the fastest convergence is achieved by choosing λ slightly smaller (in absolute value) than the true energy. The dynamic update is indeed equally fast as setting $\lambda = E$. The value of λ cannot however be chosen arbitrarily, as the rate of convergence deteriorates rapidly away from $\lambda = E$, ultimately leading to divergence. It is therefore important to choose λ close to the final energy, but updating the operator in each iteration can safely be avoided, especially close to convergence.

The same analysis has been repeated for methane. Since the exact orbital energies ϵ_i are not available, the converged results from a previous calculation have been employed: for each occupied orbital i λ_i is expressed relative to the converged ϵ_i. The fixed-λ equations (53) are then iterated to a maximum residual orbital norm of $\varepsilon_r \leq 10^{-5}$ without subspace acceleration, and the Fock matrix is diagonalized in each iteration. The number of iterations required with different choices of λ_i/ϵ_i is presented in figure 3.

In contrast to the Hydrogen atom, the best choice is now $\lambda_i/\epsilon_i = 1$ (exact eigenvalues). The dynamic strategy (dashed line) is in this case equally good. The conclusion is again that λ should always be chosen close to the current orbital energy, but not necessarily equal to ϵ_i^n in each

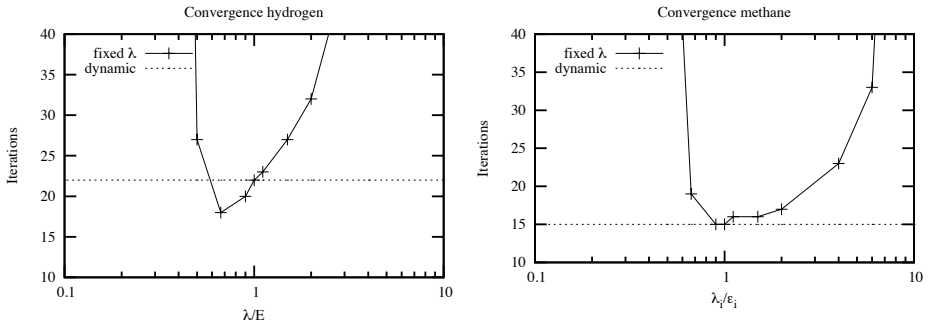


Figure 3: The effect of using a fixed λ in the power iteration. The number of iterations required to bring the residual below $\varepsilon_r \leq 10^{-7}$ for Hydrogen and $\varepsilon_r \leq 10^{-5}$ for methane for different choices of λ. The number of iterations when using a dynamic λ is given for comparison.

iteration.

Based on these observations the Helmholtz operator is recomputed with $\lambda_i = \epsilon_i^n$ whenever the new energy differs from the old λ_i by more than 1%. Our tests show that this choice is close to optimal and has minimal effect on the overall convergence: at most one extra iteration was required on a wide variety of molecules and to different choices of final precision. The significant advantage is that the Helmholtz operator associated with a given orbital is in practice fixed once the orbital is converged within $\varepsilon_r \leq 10^{-2}$. This strategy leads to considerable savings in computing time, especially for large systems and high accuracy.

5.2 Accuracy

As the approach followed is unconventional (real-space minimization with Multiwavelets) and presents also novel aspects compared to previous work (localized orbitals and full Fock matrix computation) it is important to investigate the overall accuracy that can be attained.

The accuracy in the computation of the total energy is related to the overall accuracy of the calculation ε_{tot}. It is however important to assess how accurate the orbital representation has to be in order to guarantee the demanded accuracy on the energy. Fig. 4 shows the convergence of the maximum orbital residual and the total LDA energy for small closed-shell atoms. The overall accuracy of the calculations was $\varepsilon_{tot} = 10^{-6}$, and the canonical orbital equations were iterated without Krylov acceleration in order to determine the maximum accuracy attainable. The total energies were compared to the basis set limit taken from the National Institute of Standards and Technology (NIST)[43].

The error in the total energy is brought below the threshold within ten iterations for all atoms, and it is afterwards stabilized at about an order of magnitude below ε_{tot}. A similar behavior is

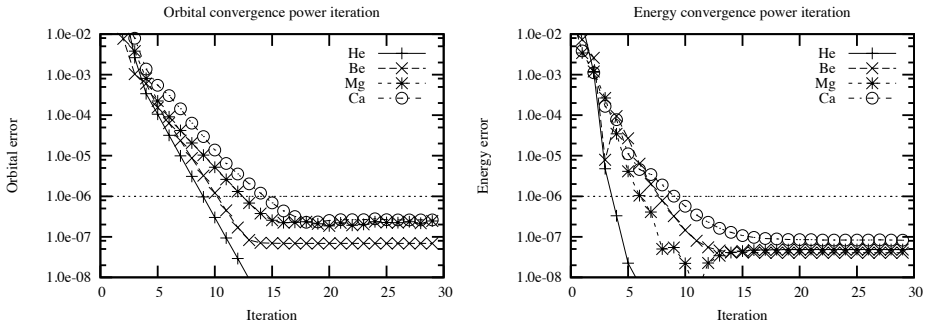


Figure 4: Orbital and energy convergence for power iteration without subspace acceleration. Overall accuracy is kept at $\varepsilon_{tot} = 10^{-6}$, and the equations are iterated to maximum accuracy. The error in the total energy is taken with respect to the NIST standard as a reference, whereas the error in the orbital is the maximum orbital residual defined as the norm of the largest orbital update at each iteration.

observed for the orbital error. However, at any given iteration the error in the energy is at least an order of magnitude below the error in the orbitals. Therefore, in order to achieve an accuracy ε in the total energy, it is sufficient to set $\varepsilon_{tot} = \varepsilon$, and $\varepsilon_r \leq 10\varepsilon$.

The corresponding results using Krylov subspace acceleration are reported in Fig. 5. Although the convergence is less regular, the same trend is observed as for the straightforward power iteration. The accuracy obtained is the same, but it is reached with 30-40% fewer iterations. The relationship between the error in the orbitals and in the total energy remains at one order of magnitude.

A further confirmation of the correctness of this choice can be found by inspecting the results obtained for the electronic structure of selected atoms (see Tab. 2 for closed shell systems and Tab. 3 for open shell ones) using the Local Spin-Density Approximation (LSDA). All calculations were iterated to a maximum orbital residual of $\varepsilon_r \leq 10\varepsilon_{tot}$.

In Tables 4 to 6 results for atoms and small molecules at the HF level are reported. The test molecules are taken from the HEAT[44, 45, 46] project which is aiming at high-accuracy *ab initio* thermochemistry. The molecular geometries as well as the presented energies for estimated Hartree-Fock limit and using high-quality Gaussian basis sets can be found in the references above. Robust and rapid convergence is achieved for all investigated systems with accuacy comparable to the LDA calculations. An important point to achieve fast and accurate convergence is the calculation of the electronic density in Eq. (32) and the Hartree-Fock exchange in Eq. (33): in an attempt to improve performance the orbitals used therein were only normalized (orthogonalization is naturally achieved close to convergence) but such an approximation affected convergence severely, especially for some open-shell systems (e.g. NO, CH, CCH, HO_2).

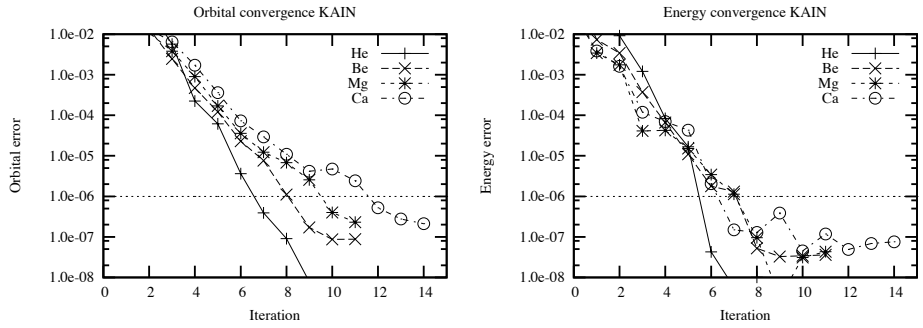


Figure 5: Orbital and energy convergence for power iteration including subspace acceleration $m = 4$. Overall accuracy is kept at $\varepsilon_{tot} = 10^{-6}$, and the equations are iterated to maximum accuracy. The error in the total energy is taken with respect to the NIST standard as a reference, whereas the error in the orbital is the maximum orbital residual defined as the norm of the largest orbital update at each iteration.

Table 2: LDA energies for closed-shell atoms. NIST represents basis set limit[43]. Orbitals are converged to a maximum residual of $\varepsilon_r \leq 10\varepsilon_{tot}$.

Requested	Energy (Hartree)				
precision	E_{kin}	E_{ee}	E_{en}	E_{xc}	E_{tot}
			Helium		
$\varepsilon_{tot} \leq 10^{-3}$	2.77205228	1.99665754	-6.63046074	-0.97373563	-2.83548654
$\varepsilon_{tot} \leq 10^{-5}$	2.76793649	1.99612381	-6.62558036	-0.97331623	-2.83483629
$\varepsilon_{tot} \leq 10^{-7}$	2.76792245	1.99611976	-6.62556386	-0.97331398	-2.83483563
NIST	2.767922	1.996120	-6.625564	-0.973314	-2.834836
			Beryllium		
$\varepsilon_{tot} \leq 10^{-3}$	14.32209505	7.11589404	-33.37086518	-2.51552236	-14.44839844
$\varepsilon_{tot} \leq 10^{-5}$	14.30973819	7.11528422	-33.35734165	-2.51487298	-14.44719222
$\varepsilon_{tot} \leq 10^{-7}$	14.30942392	7.11525737	-33.35703462	-2.51485614	-14.44720948
NIST	14.309424	7.115257	-33.357034	-2.514856	-14.447209
			Magnesium		
$\varepsilon_{tot} \leq 10^{-3}$	197.83334926	95.71507014	-477.14806651	-15.44282502	-199.04247213
$\varepsilon_{tot} \leq 10^{-5}$	198.53996893	95.67281328	-477.89751361	-15.45500303	-199.13973443
$\varepsilon_{tot} \leq 10^{-7}$	198.54150414	95.67328823	-477.89914697	-15.45505141	-199.13940601
NIST	198.541505	95.673290	-477.899149	-15.455051	-199.139406
			Calcium		
$\varepsilon_{tot} \leq 10^{-3}$	674.19516760	285.15968984	-1600.92618633	-34.13913904	-675.71046793
$\varepsilon_{tot} \leq 10^{-5}$	674.66137975	285.20533903	-1601.46680475	-34.14256738	-675.74265335
$\varepsilon_{tot} \leq 10^{-7}$	674.65729500	285.20613876	-1601.46319769	-34.14253866	-675.74230258
NIST	674.657334	285.206130	-1601.463209	-34.142538	-675.742283

Table 3: LSDA energies for open-shell atoms. NIST represents basis set limit[43]. Orbitals are converged to a maximum residual of $\varepsilon_r \leq 10\varepsilon_{tot}$.

Requested	Energy (Hartree)				
precision	E_{kin}	E_{ee}	E_{en}	E_{xc}	E_{tot}
			Hydrogen		
$\varepsilon_{tot} \leq 10^{-3}$	0.46408449	0.29795528	-0.96306035	-0.27764807	-0.47866865
$\varepsilon_{tot} \leq 10^{-5}$	0.46664520	0.29837713	-0.96562129	-0.27807194	-0.47867090
$\varepsilon_{tot} \leq 10^{-7}$	0.46664311	0.29837681	-0.96561918	-0.27807151	-0.47867076
NIST	0.466643	0.298377	-0.965619	-0.278072	-0.478671
			Lithium		
$\varepsilon_{tot} \leq 10^{-3}$	7.25182279	4.00976028	-16.93974456	-1.66542991	-7.34359140
$\varepsilon_{tot} \leq 10^{-5}$	7.24980921	4.00923212	-16.93781276	-1.66517957	-7.34395099
$\varepsilon_{tot} \leq 10^{-7}$	7.24989420	4.00925979	-16.93792111	-1.66518948	-7.34395660
NIST	7.249892	4.009258	-16.937918	-1.665189	-7.343957
			Sodium		
$\varepsilon_{tot} \leq 10^{-3}$	160.14833493	79.70030511	-387.75498135	-13.51431838	-161.42065969
$\varepsilon_{tot} \leq 10^{-5}$	160.90590302	79.78423947	-388.60422402	-13.53369303	-161.44777456
$\varepsilon_{tot} \leq 10^{-7}$	160.90724741	79.78490710	-388.60601781	-13.53376279	-161.44762609
NIST	160.907255	79.784904	-388.606022	-13.533763	-161.447625
			Potassium		
$\varepsilon_{tot} \leq 10^{-3}$	597.01717215	257.40676800	-1421.00811117	-31.65933886	-598.24350988
$\varepsilon_{tot} \leq 10^{-5}$	597.17434653	257.43558611	-1421.15953204	-31.65796306	-598.20756246
$\varepsilon_{tot} \leq 10^{-7}$	597.17998931	257.43704851	-1421.16500554	-31.65806931	-598.20603703
NIST	597.179968	257.437000	-1421.164934	-31.658067	-598.206032

Table 4: RHF total energies for closed-shell molecules. Estimated HF limit, Gaussian basis energies and geometries are taken from Refs.[44, 45, 46].

Requested precision	Total energy (Hartree)			
	H_2O	H_2O_2	CO	CO_2
$\varepsilon_{tot} = 10^{-5}$	-76.067611455	-150.85253297	-112.79087294	-187.72538886
$\varepsilon_{tot} = 10^{-6}$	-76.067556696	-150.85249254	-112.79069389	-187.72541991
$\varepsilon_{tot} = 10^{-7}$	-76.067535613	-150.85246986	-112.79081263	-187.72538522
$\varepsilon_{tot} = 10^{-8}$	-76.067535431	-150.85247037	-112.79081269	-187.72538560
Est. HF limit	-76.0675	-150.8525	-112.7908	-187.7254
aug-cc-pCV5Z	-76.067379371	-150.85218780	-112.79063514	-187.72508317
aug-cc-pCVQZ	-76.066140457	-150.84985235	-112.78919290	-187.72260431
	H_2	C_2H_2	HF	F_2
$\varepsilon_{tot} = 10^{-5}$	-1.1336231555	-76.855426687	-100.07104518	-198.77361614
$\varepsilon_{tot} = 10^{-6}$	-1.1336185308	-76.855583643	-100.07090391	-198.77356693
$\varepsilon_{tot} = 10^{-7}$	-1.1336185408	-76.855589419	-100.07088606	-198.77354557
$\varepsilon_{tot} = 10^{-8}$	-1.1336185528	-76.855588322	-100.07088501	-198.77354427
Est. HF limit	-1.1336	-76.8556	-100.0709	-198.7735
aug-cc-pCV5Z	-1.1335996653	-76.855469571	-100.07066449	-198.77312544
aug-cc-pCVQZ	-1.1334622625	-76.854638249	-100.06873826	-198.76928779
	N_2	NH_3	HCN	HNO
$\varepsilon_{tot} = 10^{-5}$	-108.99285921	-56.225111832	-92.915931357	-129.84997375
$\varepsilon_{tot} = 10^{-6}$	-108.99312380	-56.225050995	-92.915781860	-129.84996270
$\varepsilon_{tot} = 10^{-7}$	-108.99309384	-56.225052223	-92.915802570	-129.84998798
$\varepsilon_{tot} = 10^{-8}$	-108.99309024	-56.225051873	-92.915801013	-129.84998863
Est. HF limit	-108.9931	-56.2250	-92.9158	-129.8500
aug-cc-pCV5Z	-108.99293047	-56.224938182	-92.915668251	-129.84977143
aug-cc-pCVQZ	-108.99167619	-56.224119853	-92.914667092	-129.84803908

Table 5: UHF total energies for open-shell molecules. Estimated HF limit, Gaussian basis energies and geometries are taken from Refs.[44, 45, 46].

Requested precision	Total energy (Hartree)			
	H	OH	HO_2	O_2
$\varepsilon_{tot} = 10^{-5}$	-0.5000005418	-75.428181578	-150.25211601	-149.69178275
$\varepsilon_{tot} = 10^{-6}$	-0.5000000246	-75.428114257	-150.25269142	-149.69155112
$\varepsilon_{tot} = 10^{-7}$	-0.5000000111	-75.428109595	-150.25267450	-149.69154144
$\varepsilon_{tot} = 10^{-8}$	-0.5000000001	-75.428109170	-150.25267364	-149.69154036
Est. HF limit	-0.5000	-75.4281	-150.2527	-149.6915
aug-cc-pCV5Z	-0.4999947846	-75.427965654	-150.25239658	-149.69126719
aug-cc-pCVQZ	-0.4999483215	-75.426782123	-150.25009561	-149.68900724
	CH	CH_2	CH_3	CCH
$\varepsilon_{tot} = 10^{-5}$	-38.284525066	-38.941000192	-39.581248206	-76.183662196
$\varepsilon_{tot} = 10^{-6}$	-38.284539890	-38.940974280	-39.581213883	-76.183555283
$\varepsilon_{tot} = 10^{-7}$	-38.284505692	-38.940972985	-39.581211226	-76.183559436
$\varepsilon_{tot} = 10^{-8}$	-38.284505511	-38.940972753	-39.581211023	-76.183559250
Est. HF limit	-38.2845	-38.9410	-39.5812	-76.1835
aug-cc-pCV5Z	-38.284449323	-38.940902183	-39.581129872	-76.183449583
aug-cc-pCVQZ	-38.284067801	-38.940408627	-39.580563543	-76.182657470
	NH	NH_2	NO	NO_2
$\varepsilon_{tot} = 10^{-5}$	-54.986447671	-55.592360153	-129.30950551	-204.13091462
$\varepsilon_{tot} = 10^{-6}$	-54.986416818	-55.592337083	-129.30951408	-204.13070528
$\varepsilon_{tot} = 10^{-7}$	-54.986414045	-55.592333120	-129.30953328	-204.13072391
$\varepsilon_{tot} = 10^{-8}$	-54.986413719	-55.592332706	-129.30953318	-204.13072160
Est. HF limit	-54.9864	-55.5923	-129.3095	
aug-cc-pCV5Z	-54.986325573	-55.592232278	-129.30932180	-204.13038130
aug-cc-pCVQZ	-54.985662610	-55.591500695	-129.30760011	-204.12761715

Table 6: UHF total energies for open-shell molecules. Estimated HF limit, Gaussian basis energies and geometries are taken from Refs.[44, 45, 46].

Requested precision	Total energy (Hartree)			
	C	N	O	F
$\varepsilon_{tot} = 10^{-5}$	-37.693753180	-54.404705282	-74.819062105	-99.416366868
$\varepsilon_{tot} = 10^{-6}$	-37.693741232	-54.404546399	-74.818985544	-99.416319097
$\varepsilon_{tot} = 10^{-7}$	-37.693740484	-54.404548263	-74.818980697	-99.416306995
$\varepsilon_{tot} = 10^{-8}$	-37.693740393	-54.404548319	-74.818980196	-99.416306085
Est. HF limit	-37.6937	-54.4045	-74.8190	-99.4163
aug-cc-pCV5Z	-37.693694337	-54.404470064	-74.818844591	-99.416094067
aug-cc-pCVQZ	-37.693383688	-54.403857000	-74.817689844	-99.414170791
	CN	CF	HCO	OF
$\varepsilon_{tot} = 10^{-5}$	-92.242768032	-137.23900869	-113.30415122	-174.21099973
$\varepsilon_{tot} = 10^{-6}$	-92.242824581	-137.23895560	-113.30393123	-174.21091105
$\varepsilon_{tot} = 10^{-7}$	-92.242825135	-137.23898474	-113.30400749	-174.21091140
$\varepsilon_{tot} = 10^{-8}$	-92.242824460	-137.23898575	-113.30401610	-174.21091074
Est. HF limit	-92.2428	-137.2390	-113.3040	-174.2109
aug-cc-pCV5Z	-92.242699028	-137.23872998	-113.30382984	-174.21056660
aug-cc-pCVQZ	-92.241728449	-137.23646629	-113.30232555	-174.20751193

Table 7: Computation time for LDA calculation on small alkane systems. The residual norm is converged to $\varepsilon_r \leq 10^{-4}$ using localized orbitals. N_e is the number of electrons and n_{it} is the number of iterations. The overall accuracy of the calculations were $\varepsilon_{tot} = 10^{-5}$, which should reflect the relative accuracy of the presented energies.

Molecule	N_e	n_{it}	Time (sec)	Wall time 16 CPUs	Energy (Hartree)
C_1H_4	10	7	2300	$\sim$ 2 min	-40.12193
C_2H_6	18	8	5800	$\sim$ 5 min	-79.07702
C_4H_{10}	34	9	15500	$\sim$ 15 min	-156.99457
C_6H_{14}	50	9	30000	$\sim$ 30 min	-234.90188
C_8H_{18}	66	8	52000	$\sim$ 60 min	-312.82352
$C_{10}H_{22}$	82	8	82000	$\sim$ 90 min	-390.72429

5.3 Computing time and scaling

A brief illustration of the performance and scaling of the code is presented in Tab. 7 for LDA calculations of small alkane molecules using localized orbitals. As the code is still a prototype in a development phase, no explicit attempt has been made exploiting the locality of the molecular orbitals to reduce the scaling with respect to system size. Therefore the code displays a formal quadratic scaling due to the sum appearing in the non-canonical orbital equations (31), in the calculation of the Fock matrix and in the orbital orthogonalizations. The calculation of the Coulomb potential has been demonstrated to scale linearly in a previous study[30]. In the case of Hartree-Fock there is also quadratic scaling in the calculation of the exchange potential, assuming that the application of the Coulomb operator scales linearly with the system size.

In the future, it should be possible to approach an asymptotic linear scaling of both Kohn-Sham and Hartree-Fock calculations, by taking full advantage of the exponential fall-off of the localized orbitals. Similar screening techniques to the ones already adopted in traditional Gaussian based SCF implementations [32, 47] could be employed in our Multiwavelet implementation.

6 Conclusions

We have presented a new implementation of a Multiwavelet-based SCF solver for HF and DFT, which is able to deal both with closed-shell as well as open-shell systems. Our solver is based on a preconditioned steepest descent step[24], accelerated through the KAIN method[22]. Our implementation is parallel (both OMP and MPI parallelizations are present) and able to deal exclusively with localized orbitals, without any reference to the delocalized canonical orbitals. In order to do that we have shown how it is possible to compute the Fock matrix in the localized orbital basis exploiting the formal relation between the level-shifted Laplacian and the bound-state

Helmholtz kernel, thus avoiding any reference to the kinetic energy operator. We have shown that we are able to obtain high accuracy results (basis-set limit within an arbitrary, predefined threshold) both with the canonical basis and the localized one. We have empirically found that keeping the parameter λ which defines the Helmholtz operator close to the current orbital energy value for each orbital is a near-optimal choice, thus limiting the need to update H^μ to the initial steps of the SCF cycle. Another important finding of our study is the significant improvement of the preconditioned steepest descent convergence without acceleration when a localized basis is employed. This result seems to indicate that localization improves significantly the quality of the preconditioner, thus making the non-accelerated convergence simpler. In fact when Krylov acceleration is added on top only a few iterations are gained indicating that each preconditioned step is already a very good guess. This observation deserves further attention and we will assess whether this is achieved also for other molecular systems, with more complicated electronic structure. The use of localized orbitals is also important for practical but important reasons: (1) the development of low-scaling (asymptotically linear) algorithms; (2) the reduction of the memory footprint of the orbital representation; (3) the exploitation of modern massive parallel architectures.

7 Acknowledgments

This work has been supported by the Research Council of Norway through a Centre of Excellence Grant (Grant No. 179568/V30) and from the Norwegian Supercomputing Program (NOTUR) through a grant of computer time (Grant No. NN4654K).

References

[1] Trygve Helgaker, Poul Jorgensen, and Jeppe Olsen. *Molecular Electronic-Structure Theory*. Wiley-Blackwell, 2008.

[2] David Moncrieff and S Wilson. Computational linear dependence in molecular electronic structure calculations using universal basis sets. *Int J Quantum Chem*, 101(4):363–371, 2005.

[3] G Kresse and J Furthmuller. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys Rev B*, 54(16):11169–11186, 1996.

[4] D Vanderbilt. Soft Self-Consistent Pseudopotentials in a Generalized Eigenvalue Formalism. *Phys. Rev., B Condens. Matter*, 41(11):7892–7895, 1990.

[5] G Y Sun, J Kurti, P Rajczy, M Kertesz, J Hafner, and G Kresse. Performance of the Vienna ab initio simulation package (VASP) in chemical applications. *J Mol Struc-Theochem*, 624:37–45, 2003.

[6] T Torsti, T Eirola, J Enkovaara, T Hakala, P Havu, V Havu, T Hoynalanmaa, J Ignatius, M Lyly, I Makkonen, T T Rantala, J Ruokolainen, K Ruotsalainen, E Rasanen, H Saarikoski, and M J Puska. Three real-space discretization techniques in electronic structure calculations. *Physica Status Solidi B-Basic Solid State Physics*, 243(5):1016–1053, April 2006.

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

[8] T Yanai, GI Fann, Z Gan, RJ Harrison, and G Beylkin. Multiresolution quantum chemistry in multiwavelet bases: Hartree–Fock exchange. *J Chem Phys*, 121:6680, 2004.

[9] B Alpert, G Beylkin, D Gines, and L Vozovoi. Adaptive solution of partial differential equations in multiwavelet bases. *J Comput Phys*, 1999.

[10] Bradley K Alpert. A Class of Bases in L^2 for the Sparse Representation of Integral Operators. *SIAM Journal on Mathematical Analysis*, 24(1):246–262, January 1999.

[11] Florian A Bischoff, Robert J Harrison, and Edward F Valeev. Computing many-body wave functions with guaranteed precision: The first-order Moller-Plesset wave function for the ground state of helium atom. *J Chem Phys*, 137(10):104103, September 2012.

[12] A. Durdek, S. R. Jensen, J. Jusélius, P. Wind, T. Flå, and L. Frediani. Adaptive order polynomial algorithm in a multi-wavelet representation scheme. Submitted, 2013.

[13] G Beylkin, R J Harrison, and K E Jordan. Singular operators in multiwavelet bases. *IBM Journal of Research and Development*, 48(2):161–171, 2004.

[14] G Beylkin. Fast adaptive algorithms in the non-standard form for multidimensional problems. *Appl Comput Harmon A*, 24(3):354–377, 2008.

[15] Stinne Høst, Jeppe Olsen, Branislav Jansik, Lea Thøgersen, Poul Jorgensen, and Trygve Helgaker. The augmented Roothaan-Hall method for optimizing Hartree-Fock and Kohn-Sham density matrices. *J Chem Phys*, 129(12):124106, September 2008.

[16] Yousef Saad. *Numerical Methods for Large Eigenvalue Problems*. Revised Edition. SIAM, 2011.

[17] Lin-Wang Wang and Alex Zunger. Large scale electronic structure calculations using the Lanczos method. *Computational Materials Science*, 2(2):326–340, 1994.

[18] Ernest R Davidson. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *J Comput Phys*, 17(1):87–94, 1975.

[19] Peter Pulay. Convergence acceleration of iterative sequences. The case of SCF iteration. *Chem Phys Lett*, 73(2):393–398, 1980.

[20] D M Wood and Alex Zunger. A new method for diagonalising large matrices. *J. Phys. A: Math. Gen.*, 18(9):1343–1359, 1985.

[21] Mike C Payne, Michael P Teter, Douglas C Allan, T A Arias, and J D Joannopoulos. Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients. *Rev. Mod. Phys.*, 64(4):1045–1097, 1992.

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

[23] Thorsten Rohwedder and Reinhold Schneider. An analysis for the DIIS acceleration method used in quantum chemistry calculations. *J Math Chem*, 49(9):1889–1914, October 2011.

[24] Reinhold Schneider, Thorsten Rohwedder, Alexej Neelov, and Johannes Blauert. Direct minimization for calculating invariant subspaces in density functional computations of the electronic structure. *J Comput Math*, 27(2-3):360–387, 2008.

[25] Fritz Keinert. *Wavelets and Multiwavelets*, volume 42 of *Studies in advanced mathematics*. Chapman and Hall, CRC Press, Boca Raton, FL, USA, 2003.

[26] Gregory Beylkin, Vani Cheruvu, and Fernando Perez. Fast adaptive algorithms in the non-standard form for multidimensional problems. *Appl Comput Harmon A*, 24(3):354–377, 2008.

[27] D Gines, G Beylkin, and J Dunn. LU Factorization of Non-standard Forms and Direct Multiresolution Solvers* 1. *Appl Comput Harmon A*, 5(2):156–201, 1998.

[28] Luca Frediani, Eirik Fossgaard, Tor Flå, and Kenneth Ruud. 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. *Mol Phys*, 111(9-11):1143–1160, July 2013.

[29] G Beylkin and MJ Mohlenkamp. Numerical operator calculus in higher dimensions. *P Natl Acad Sci Usa*, 99(16):10246, 2002.

[30] S. R. Jensen, J. Jusélius, A. Durdek, T. Flå, P. Wind, and L. Frediani. Linear scaling coulomb interaction in the multiwavelet basis, a parallel implementation. Submitted, 2013.

[31] M H Kalos. Monte Carlo calculations of the ground state of three-and four-body nuclei. *Physical Review*, 128(4):1791, 1962.

[32] Lsdalton, a linear scaling molecular electronic structure program, release dalton2013 (2013), see http://daltonprogram.org.

[33] S. F. Boys. Construction of some molecular orbitals to be approximately invariant for changes from one molecule to another. *Rev. Mod. Phys.*, 32:296–299, 1960.

[34] J. M. Foster and S. F. Boys. Canonical configurational interaction procedure. *Rev. Mod. Phys.*, 32:300–302, 1960.

[35] Helgaker T. Optimization of minima and saddle points. In BjrnO. Roos, editor, *Lecture Notes in Quantum Chemistry*, volume 58 of *Lecture Notes in Chemistry*, pages 295–324. Springer Berlin Heidelberg, 1992.

[36] John C Slater. A simplification of the Hartree-Fock method. *Physical Review*, 81(3):385, 1951.

[37] Paul Adrien Maurice Dirac. Quantum mechanics of many-electron systems. *Proceedings of the Royal Society of London A*, 123(729):714–733, 1929.

[38] S. H. Vosko, L. Wilk, and M. Nusair. Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis. *Canadian Journal of Physics*, 58(8):1200–1211, 1980.

[39] Ulf Ekström, Lucas Visscher, Radovan Bast, Andreas J Thorvaldsen, and Kenneth Ruud. Arbitrary-Order Density Functional Response Theory from Automatic Differentiation. *J Chem Theory Comput*, 6(7):1971–1980, July 2010.

[40] Kestutis Aidas, Celestino Angeli, Keld L Bak, Vebjørn Bakken, Radovan Bast, Linus Boman, Ove Christiansen, Renzo Cimiraglia, Sonia Coriani, Pål Dahle, Erik K Dalskov, Ulf Ekström, Thomas Enevoldsen, Janus J Eriksen, Patrick Ettenhuber, Berta Fernández, Lara Ferrighi, Heike Fliegl, Luca Frediani, Kasper Hald, Asger Halkier, Christof Hättig, Hanne Heiberg, Trygve Helgaker, Alf Christian Hennum, Hinne Hettema, Eirik Hjertenaes, Stinne Høst, Ida-Marie Høyvik, Maria Francesca Iozzi, Branislav Jansik, Hans Jørgen Aa Jensen, Dan Jonsson, Poul Jorgensen, Joanna Kauczor, Sheela Kirpekar, Thomas Kjaergaard, Wim Klopper, Stefan Knecht, Rika Kobayashi, Henrik Koch, Jacob Kongsted, Andreas Krapp, Kasper Kristensen, Andrea Ligabue, Ola B Lutnaes, Juan I Melo, Kurt V Mikkelsen, Rolf H Myhre, Christian Neiss, Christian B Nielsen, Patrick Norman, Jeppe Olsen, Jógvan Magnus H Olsen, Anders Osted, Martin J Packer, Filip Pawlowski, Thomas B Pedersen, Patricio F Provasi, Simen Reine, Zilvinas Rinkevicius, Torgeir A Ruden, Kenneth Ruud, Vladimir V Rybkin, Pawel Sałek, Claire C M Samson, Alfredo Sánchez de Merás, Trond Saue, Stephan P A Sauer, Bernd Schimmelpfennig, Kristian Sneskov, Arnfinn H Steindal, Kristian O Sylvester-Hvid,

Peter R Taylor, Andrew M Teale, Erik I Tellgren, David P Tew, Andreas J Thorvaldsen, Lea Thøgersen, Olav Vahtras, Mark A Watson, David J D Wilson, Marcin Ziolkowski, and Hans Agren. The Dalton quantum chemistry program system. *WIREs Comput Mol Sci*, pages n/a–n/a, September 2013.

[41] J Stephen Binkley, John A Pople, and Warren J Hehre. Self-consistent molecular orbital methods. 21. Small split-valence basis sets for first-row elements. *J Am Chem Soc*, 102(3):939–947, 1980.

[42] Mark S Gordon, J Stephen Binkley, John A Pople, William J Pietro, and Warren J Hehre. Self-consistent molecular-orbital methods. 22. Small split-valence basis sets for second-row elements. *J Am Chem Soc*, 104(10):2797–2803, May 1982.

[43] http://physics.nist.gov/PhysRefData/DFTdata/Tables/ptable.html.

[44] A. Tajti, P. G. Szalay, A. G. Csszr, M. Kllay, J. Gauss, E. F. Valeev, B. A. Flowers, J. Vzquez, and J. F. Stanton. Heat: High accuracy extrapolated ab initio thermochemistry. *J. Chem. Phys.*, 121(23):11599–11613, 2004.

[45] Y. J. Bomble, J. Vzquez, M. Kllay, C. Michauk, P. G. Szalay, A. G. Csszr, J. Gauss, and J. F. Stanton. High-accuracy extrapolated ab initio thermochemistry. ii. minor improvements to the protocol and a vital simplification. *J. Chem. Phys.*, 125(6), 2006.

[46] M. E. Harding, J. Vzquez, B. Ruscic, A. K. Wilson, J. Gauss, and J. F. Stanton. High-accuracy extrapolated ab initio thermochemistry. iii. additional improvements and overview. *J. Chem. Phys.*, 128(11), 2008.

[47] S Goedecker. Linear scaling electronic structure methods. *Rev. Mod. Phys.*, 71(4):1085–1123, July 1999.