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 $k=k_{min}$

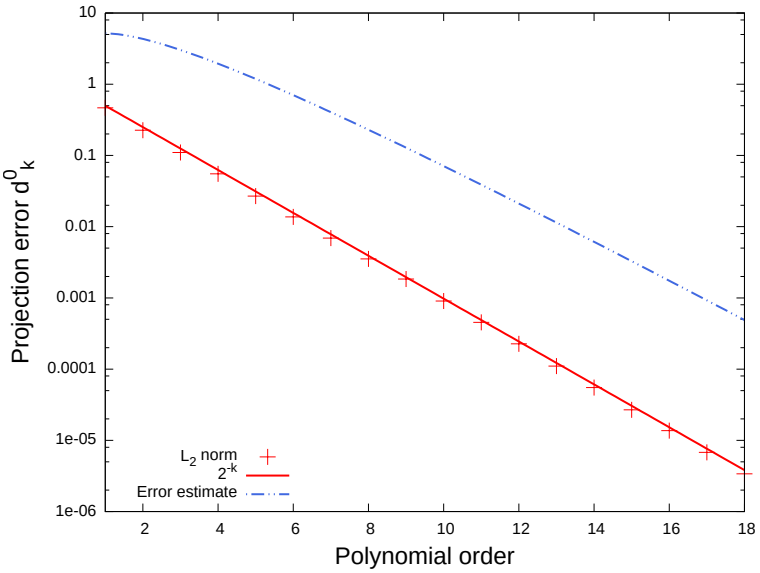

Figure 1: Representation of d_k^0 as a function of the polynomial order. The dots represent the L_2 norm of the error computed applying Eq. (29). The curve is the error estimate according to Eq. (33).

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.

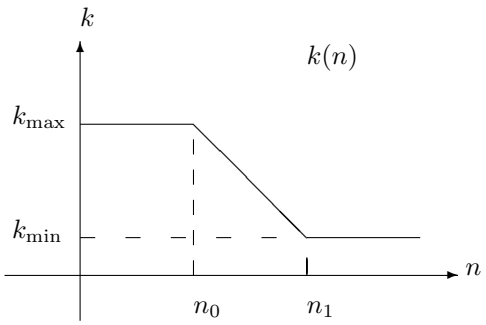

Figure 2: Generic shape of the order $k(n)$

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.

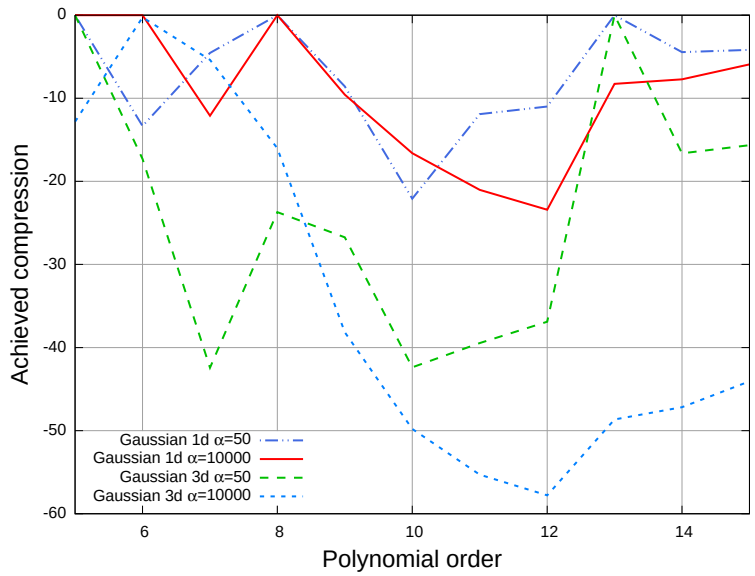

Figure 3: Percentage of coefficients gain in function of the order k_{max} for the Gaussian-type function in the one- and three-dimensional case and $\alpha = 50, 10000$. The data corresponds to the last column of the corresponding Tables.

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.

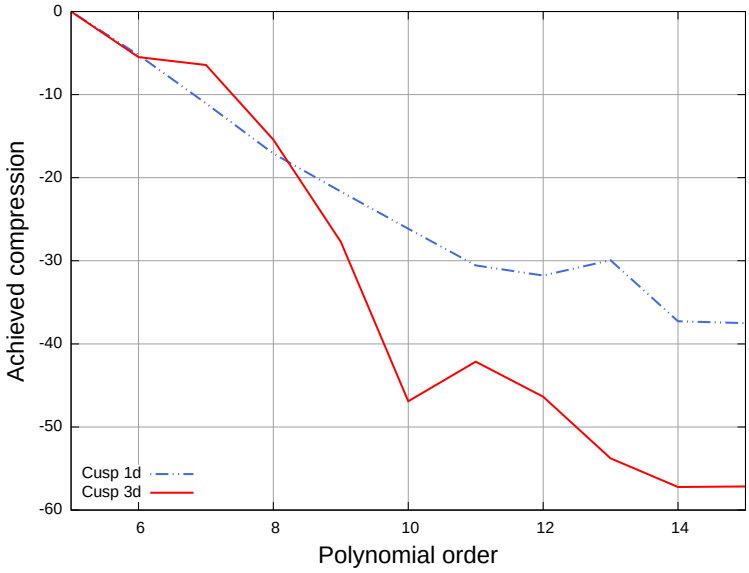

Figure 4: Percentage of coefficients gain in function of the order k_{max} for the the Slater-type function in the one- and three- dimensional case with $\alpha = 100$. The data corresponds to the last column of the corresponding Tables.

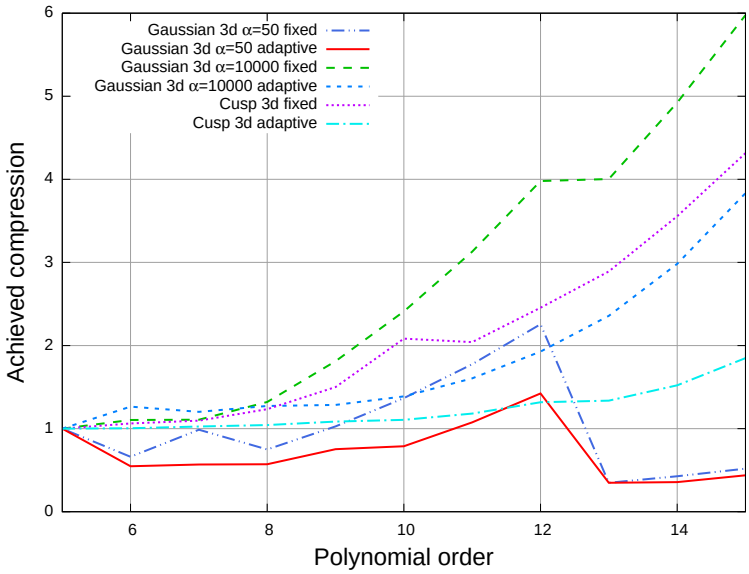

Figure 5: Relative variation on the number of coefficients for the Gaussian type function with $\alpha = 10000$ and Slater type with $\alpha = 100$ in the three-dimensional case. For the two functions, the SR and the DOR are presented. The relative variation $r(k)$ is obtained with respect to the order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