Daubechies wavelets as a basis set for density functional pseudopotential calculations

Luigi Genovese,[1] Alexey Neelov,[2] Stefan Goedecker,[2] Thierry Deutsch,[1] Seyed Alireza Ghasemi,[2] Alexander Willand,[2] Damien Caliste,[1] Oded Zilberberg,[2] Mark Rayson,[2] Anders Bergman,[1] and Reinhold Schneider[3]
[1]*CEA, INAC, SP2M, L_Sim, 38054 Grenoble Cedex 9, France*
[2]*Institut für Physik, Universität Basel, Klingelbergstr. 82, 4056 Basel, Switzerland*
[3]*Technische Universität Berlin, Sekretariat MA 8-1, Str. des 17, Juni 136, D-10623 Berlin, Germany*

(Received 16 April 2008; accepted 30 May 2008; published online 7 July 2008)

Daubechies wavelets are a powerful systematic basis set for electronic structure calculations because they are orthogonal and localized both in real and Fourier space. We describe in detail how this basis set can be used to obtain a highly efficient and accurate method for density functional electronic structure calculations. An implementation of this method is available in the ABINIT free software package. This code shows high systematic convergence properties, very good performances, and an excellent efficiency for parallel calculations. © *2008 American Institute of Physics*. [DOI: 10.1063/1.2949547]

I. INTRODUCTION

In recent years, the Kohn–Sham (KS) formalism of the density functional theory (DFT) approach has proven to be one of the most efficient and reliable first-principles methods for investigating material properties and processes that exhibit quantum mechanical behavior. The high accuracy of the results together with the relatively simple form of the exchange-correlation (XC) functionals make this method arguably the most powerful tool for *ab initio* simulations of the properties of matter. The computational machinery of DFT calculations has been widely developed in the past decade, giving rise to a plethora of DFT codes. The use of DFT calculation has thus become more and more common, and its domain of application includes solid state physics, chemistry, material science, biology, and geology.

One of the most important characteristics of a DFT code is the set of basis functions used to express the KS orbitals. The domain of applicability of a code is tightly connected to this choice. For example, a nonlocalized basis set such as plane waves is highly suitable for electronic structure calculations of periodic and homogeneous systems, while it is much less efficient in expanding localized functions, which have a wider range of components in reciprocal space. For these reasons DFT codes based on plane waves are not well suited to simulate inhomogeneous or isolated systems such as molecules due to the high memory requirements for such kind of simulations.

A strong distinction should also be made between codes that use systematic and nonsystematic basis sets. A systematic basis set allows us to calculate the solution of the KS equations with arbitrarily high precision as the number of basis functions is increased. In other terms, the numerical precision of the results is related to the number of basis functions used to expand the KS orbitals. With such a basis set it is thus possible to obtain results that are free of errors related to the choice of the basis, eliminating a source of uncertainty. This is particularly important in view of the fact that highly accurate approximations to the XC functional are now available such as the Perdew-Burke-Ernzerhof (PBE) functional.[1] Some of these functionals also contain van der Waals interactions.[2] A systematic basis set allows us to accurately calculate the solution of a particular XC functional. On the other hand, nonsystematic basis sets, for example, Gaussians, often become overcomplete and numerical instabilities arise before absolute convergence can be achieved. Such basis sets are more difficult to use since the basis set must be carefully tuned by hand by the user, which will sometimes require some preliminary knowledge of the system under investigation. This is the most important weakness of this popular basis set.

Another property that has a role in the performances of a DFT code is the orthogonality of the basis set. The use of nonorthogonal basis sets requires the calculation of the overlap matrix of the basis functions and to perform various operations with this overlap matrix such as inverting the matrix by iterative or noniterative methods. This makes methods based on nonorthogonal systematic basis functions not only more complicated but also slower.

Daubechies[3] wavelets have virtually all the properties that one might desire of a basis set being used for the simulation of isolated or inhomogeneous systems. They form a systematic orthogonal and smooth basis, localized both in real and Fourier spaces and that allows for adaptivity. A DFT approach based on such functions will meet both the requirements of precision and localization found in many applications. In this paper, we will describe in detail a DFT method based on a Daubechies wavelet basis set. This method is implemented in a DFT code, distributed under GNU-GPL license and integrated in the ABINIT (Ref. 4) software package. A separate, standalone version of this code is also available and distributed under GNU-GPL license.[5] In the next few paragraphs we will discuss the importance of the properties of Daubechies wavelets in the context of electronic structure calculations.

A wavelet basis consists of a family of functions generated from a mother function and its translations on the points of a uniform grid of spacing h. The number of basis functions is increased by decreasing the value of h. Thanks to the systematicity of the basis, this will make the numerical description more precise. The degree of smoothness determines the speed with which one converges to the exact result as h is decreased. The degree of smoothness increases as one goes to higher order Daubechies wavelets. In our method we use Daubechies wavelets of order 16. This together with the fact that our method is quasivariational gives a convergence rate of h^{14}. Obtaining such a high convergence rate is essential in the context of electronic structure calculations where one needs highly accurate results for basis sets of acceptable size. The combination of adaptivity and a high order convergence rate is typically not achieved in other electronic structure programs using systematic real space methods.[6] An adaptive finite element code, using cubic polynomial shape functions,[7] has a convergence rate of h^6 . Finite difference methods have sometimes low[8] h^3 or high convergence rates[9] but are not adaptive.

As discussed above, localization in real space is essential for molecular systems. Basis sets that are not localized in real space are wasteful in this context. For instance, with plane waves one has to fill an orthorhombic cell into which the molecule fits. Large subregions of the cell may contain no atoms and therefore no charge density, but this feature cannot be exploited with plane waves. Since Daubechies wavelets have a compact support, one can consistently define a set of localization parameters, which allows us to put the basis functions only on the points that are sufficiently close to the atoms. The computational volume in our method is thus given only by the union of spheres centered on all the atoms in the system. Real space localization is also necessary for the implementation of linear scaling algorithms.[10] This basis set is thus a promising candidate for developing such algorithms.

Localization in Fourier space is useful for preconditioning purposes. For a given system, the convergence rate of the minimization process depends on the highest eigenvalue of the Hamiltonian operator. Since the high frequency spectrum of the Hamiltonian is dominated by the kinetic energy operator, high kinetic energy basis functions are therefore also approximate eigenfunctions of the Hamiltonian. A function localized in Fourier space is an approximate eigenfunction of the kinetic energy operator. By using such functions as basis functions for the KS orbitals, the high energy spectrum of the Hamiltonian can thus easily be preconditioned.

A high degree of adaptivity is necessary for all-electron calculations since highly localized core electrons require a much higher spatial resolution than the valence wave function away from the atomic core. High adaptivity can in principle be obtained with a wavelet basis and wavelet based all-electron electronic structure programs have been developed.[11,12] In contrast to these developments we use pseudopotentials since such pseudopotentials are the easiest way to incorporate the relativistic effects that are important for heavy elements. The use of pseudopotentials drastically reduces the need for adaptivity and we have therefore only two levels of adaptivity. We have a high resolution region that contains all the chemical bonds and a low resolution region further away from the atoms where the wave functions decay exponentially to zero. In the low resolution region each grid point carries a single basis function. In the high resolution region it carries in addition seven wavelets. In terms of degrees of freedom, the high resolution region is thus eight times denser than the low resolution region. In comparison with a plane wave method our wavelet method is therefore particularly efficient for open structures with large empty spaces and a relatively small bonding region.

The outline of this paper is as follows: in the Sec. II we describe the fundamental properties of Daubechies wavelets. Then we will describe how the various operations needed in an electronic structure calculations are done in a scaling function/wavelet basis. The last part of the paper illustrates the performances of our DFT code based on wavelets.

II. ADAPTIVITY IN A WAVELET BASIS

There are two fundamental functions in wavelet theory,[3,13] the scaling function $\phi(x)$ and the wavelet $\psi(x)$.

The most important property of these functions is that they satisfy the so-called refinement equations

$$\phi(x) = \sqrt{2}\sum_{j=1-m}^{m} h_j \phi(2x-j),$$
$$\psi = \sqrt{2}\sum_{j=1-m}^{m} g_j \phi(2x-j), \tag{1}$$

which establishes a relation between the scaling functions on a grid with grid spacing h and another one with spacing $h/2$. h_j and $g_j=(-1)^j h_{-j+1}$ are the elements of a filter that characterizes the wavelet family, and m is the order of the scaling function–wavelet family. All the properties of these functions can be obtained from the relations (1). The full basis set can be obtained from all translations by a certain grid spacing h of the mother function centered at the origin. The mother function is localized, with compact support. The maximally symmetric scaling function and wavelet of order 16 that are used in this work are shown in Fig. 1.

For a three-dimensional description, the simplest basis set is obtained by a set of products of equally spaced scaling functions on a grid of grid spacing h',

$$\phi_{i,j,k}(\mathbf{r}) = \phi(x/h'-i)\phi(y/h'-j)\phi(z/h'-k). \tag{2}$$

In other terms, the three-dimensional basis functions are a tensor product of one-dimensional basis functions. Note that we are using a cubic grid, where the grid spacing is the same in all directions, but the following description can be straightforwardly applied to general orthorhombic grids.

The basis set of Eq. (2) is equivalent to a mixed basis set of scaling functions on a twice coarser grid of grid spacing $h=2h'$,

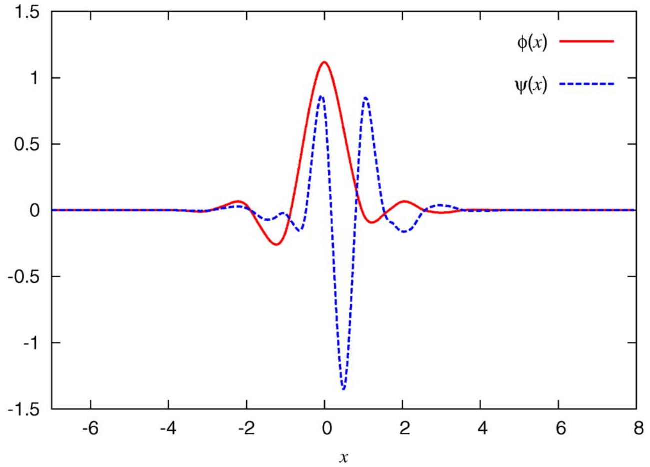

FIG. 1. (Color online) Daubechies (Ref. 3) scaling function ϕ and wavelet ψ of order 16. Both are different from zero only in the interval from −7 to 8.

$$\phi_{i,j,k}(\mathbf{r}) = \phi(x/h - i)\,\phi(y/h - j)\,\phi(z/h - k), \tag{3}$$

augmented by a set of seven wavelets

$$\begin{aligned}
\psi^1_{i,j,k}(\mathbf{r}) &= \psi(x/h - i)\,\phi(y/h - j)\,\phi(z/h - k),\\
\psi^2_{i,j,k}(\mathbf{r}) &= \phi(x/h - i)\,\psi(y/h - j)\,\phi(z/h - k),\\
\psi^3_{i,j,k}(\mathbf{r}) &= \psi(x/h - i)\,\psi(y/h - j)\,\phi(z/h - k),\\
\psi^4_{i,j,k}(\mathbf{r}) &= \phi(x/h - i)\,\phi(y/h - j)\,\psi(z/h - k), \qquad (4)\\
\psi^5_{i,j,k}(\mathbf{r}) &= \psi(x/h - i)\,\phi(y/h - j)\,\psi(z/h - k),\\
\psi^6_{i,j,k}(\mathbf{r}) &= \phi(x/h - i)\,\psi(y/h - j)\,\psi(z/h - k),\\
\psi^7_{i,j,k}(\mathbf{r}) &= \psi(x/h - i)\,\psi(y/h - j)\,\psi(z/h - k).
\end{aligned}$$

This equivalence follows from the fact that, from Eq. (1), every scaling function and wavelet on a coarse grid of spacing h can be expressed as a linear combination of scaling functions at the fine grid level h' and vice versa.

The points of the simulation grid fall into three different classes. The points that are very far from the atoms will have virtually zero charge density and thus will not carry any basis functions. The remaining grid points are either in the high resolution region that contains the chemical bonds or in the low resolution region that contains the exponentially decaying tails of the wave functions. In the low resolution region one uses only one scaling function per coarse grid point, whereas in the high resolution region one uses both the scaling function and the seven wavelets. In this region the resolution is thus doubled in each spatial dimension compared to the low resolution region. Figure 2 shows the two-level adaptive grid around a water molecule.

A wave function $\Psi(\mathbf{r})$ can thus be expanded in this basis,

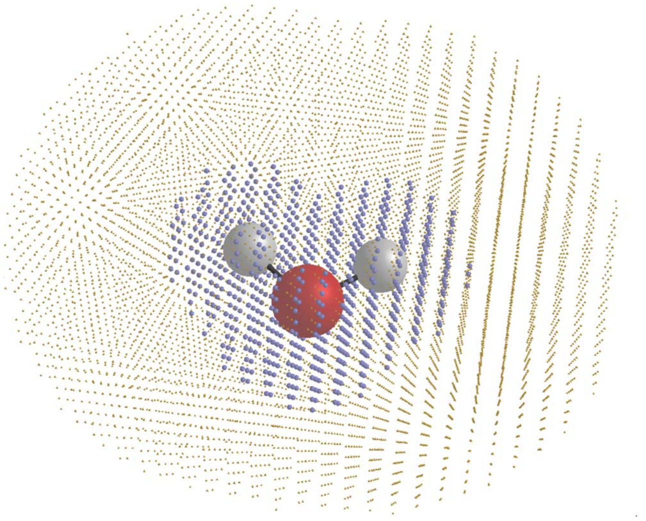

FIG. 2. (Color online) A two-level adaptive grid around a H_2O molecule. The high resolution grid points carrying both scaling functions and wavelets are shown in blue (larger points); the low resolution grid points carrying only a single scaling function are shown in yellow (smaller points).

$$\Psi(\mathbf{r}) = \sum_{i_1,i_2,i_3} s_{i_1,i_2,i_3}\phi_{i_1,i_2,i_3}(\mathbf{r}) + \sum_{j_1,j_2,j_3}\sum_{\nu=1}^{7} d^{\nu}_{j_1,j_2,j_3}\psi^{\nu}_{j_1,j_2,j_3}(\mathbf{r}). \tag{5}$$

The sum over i_1, i_2, and i_3 runs over all the grid points contained in the low resolution region and the sum over j_1, j_2, and j_3 over all the points contained in the smaller high resolution region.

The decomposition of scaling function into coarser scaling functions and wavelets can be continued recursively to obtain more than two resolution levels. We found however that a high degree of adaptivity is not of paramount importance in pseudopotential calculations. In other terms, the pseudopotentials smooth the wave functions so that two levels of resolution are enough in most cases to achieve good computational accuracy. In addition, more than two resolution levels lead to more complicated algorithms, such as the nonstandard operator form,[14] that, in turn, lead to larger prefactors.

The transformation from a pure fine scaling function representation (a basis set that contains only scaling functions centered on a finer grid of spacing h') to a mixed coarse scaling function/wavelet representation is done by the fast wavelet transformation,[13] which is a convolution and scales linearly with respect to the number of basis functions being transformed.

The wave functions are stored in a compressed form where only the nonzero scaling function and wavelet coefficients are stored. The basis set being orthogonal, several operations such as scalar products among different orbitals and between orbitals and the projectors of the nonlocal pseudopotential can directly be done in this compressed form. In the following sections we will illustrate the main operations that must be performed in the context of a DFT calculation.

III. OVERVIEW OF THE METHOD

In the KS formulation of DFT, the electronic density of a system of N electrons can be calculated from the square modulus of a set of wave functions,

$$\rho(\mathbf{r}) = \sum_{i=1}^{N/2} n_{\mathrm{occ}}^{(i)} |\Psi_i(\mathbf{r})|^2, \tag{6}$$

where the KS wave functions $|\Psi_i\rangle$ are eigenfunctions of the KS Hamiltonian, with pseudopotential V_{psp},

$$\left(-\tfrac{1}{2}\nabla^2 + V_{\mathrm{KS}}[\rho] + V_{\mathrm{psp}}\right)|\Psi_i\rangle = \epsilon_i |\Psi_i\rangle. \tag{7}$$

For the sake of simplicity we assume in this description that our electronic system is a closed-shell system of non-spin-polarized electronic orbitals. For this reason we have exactly $N/2$ KS wave functions and $\forall i\ n_{\mathrm{occ}}^{(i)}=2$.

The KS potential,

$$V_{\mathrm{KS}}[\rho] = V_H[\rho] + V_{\mathrm{xc}}[\rho] + V_{\mathrm{ext}}, \tag{8}$$

contains the Hartree potential, solution of the Poisson's equation $\nabla^2 V_H = -4\pi\rho$, the XC potential V_{xc}, and the external ionic potential V_{ext} acting on the electrons. The method we illustrate in this paper is conceived for isolated systems, namely, free boundary conditions.

In our method, we choose the pseudopotential term V_{psp} to be of the form of norm-conserving GTH-HGH pseudopotentials,[15–17] which have a local and a nonlocal term, $V_{\mathrm{psp}} = V_{\mathrm{local}} + V_{\mathrm{nonlocal}}$. For each of the ions these potentials have the form

$$V_{\mathrm{local}}(\mathbf{r}) = -\frac{Z_{\mathrm{ion}}}{r}\,\mathrm{erf}\left(\frac{r}{\sqrt{2}r_{\mathrm{loc}}}\right) + \exp\left[-\frac{1}{2}\left(\frac{r}{r_{\mathrm{loc}}}\right)^2\right] \times \left[C_1 + C_2\left(\frac{r}{r_{\mathrm{loc}}}\right)^2 + C_3\left(\frac{r}{r_{\mathrm{loc}}}\right)^4 + C_4\left(\frac{r}{r_{\mathrm{loc}}}\right)^6\right], \tag{9}$$

$$V_{\mathrm{nonlocal}} = \sum_{\ell}\sum_{i,j=1}^{3} h_{ij}^{(\ell)} |p_i^{(\ell)}\rangle\langle p_j^{(\ell)}| \tag{10}$$

$$\langle \mathbf{r}|p_i^{(\ell)}\rangle = \frac{\sqrt{2}\,r^{\ell+2(i-1)}\exp\left[-\frac{1}{2}\left(\frac{r}{r_\ell}\right)^2\right]}{r_\ell^{\ell+(4i-1)/2}\sqrt{\Gamma\left(\ell + \frac{4i-1}{2}\right)}} \sum_{m=-\ell}^{+\ell} Y_{\ell m}(\theta,\phi),$$

where $Y_{\ell m}$ are the spherical harmonics and r_{loc} and r_ℓ are, respectively, the localization radius of the local pseudopotential term and of each projector.

The analytic form of the pseudopotentials together with the fact that their expression in real space can be written in terms of a linear combination of tensor products of one-dimensional functions is of great utility in our method.

Each term in the Hamiltonian is implemented differently and will be illustrated in the following sections. After the application of the Hamiltonian, the KS wave functions are updated via a direct minimization scheme,[18] which in its actual implementation is fast and reliable for nonzero gap systems, namely, insulators. At present we have concentrated on systems with a gap; however we see no reason why the method cannot be extended to metallic systems.

IV. TREATMENT OF KINETIC ENERGY

The matrix elements of the kinetic energy operator among the basis functions of our mixed representation (i.e., scaling functions with scaling functions, scaling function with wavelets, and wavelets with wavelets) can be calculated analytically.[19] For simplicity, let us illustrate the application of the kinetic energy operator onto a wave function Ψ that is only expressed in terms of scaling functions.

$$\Psi(x,y,z) = \sum_{i_1,i_2,i_3} s_{i_1,i_2,i_3}\phi(x/h - i_1)\phi(y/h - i_2)\phi(z/h - i_3).$$

The result of the application of the kinetic energy operator on this wave function, projected to the original scaling function space, has the expansion coefficients

$$\hat{s}_{i_1,i_2,i_3} = -\frac{1}{2h^3}\int \phi(x/h - i_1)\phi(y/h - i_2)\phi(z/h - i_3) \times \Delta\Psi(x,y,z)dxdydz.$$

Analytically the coefficients s_{i_1,i_2,i_3} and $\hat{s}_{i_1,i_2,i_3}$ are related by a convolution

$$\hat{s}_{i_1,i_2,i_3} = \frac{1}{2}\sum_{j_1,j_2,j_3} K_{i_1-j_1,i_2-j_2,i_3-j_3} s_{j_1,j_2,j_3}, \tag{11}$$

where

$$K_{i_1,i_2,i_3} = T_{i_1}\delta_{i_2}\delta_{i_3} + \delta_{i_1}T_{i_2}\delta_{i_3} + \delta_{i_1}\delta_{i_2}T_{i_3}, \tag{12}$$

and

$$T_{i_1} = \int dx\phi(x/h - i_1)\partial_x^2\phi(x/h). \tag{13}$$

Using the refinement [Eq. (1)], the values of T_i can be calculated analytically from a suitable eigenvector of a matrix derived from the wavelet filters.[19] For this reason the expression of the kinetic energy operator is *exact* in a given Daubechies basis.

Since the three-dimensional kinetic energy filter K_{i_1,i_2,i_3} is a product of three one-dimensional filters [Eq. (12)], the convolution in Eq. (11) can be evaluated with $3N_1N_2N_3L$ operations for a three-dimensional grid of $N_1N_2N_3$ grid points. L is the length of the one-dimensional filter, which is 29 for our Daubechies family. The kinetic energy can thus be evaluated with linear scaling with respect to the number of nonvanishing expansion coefficients of the wave function. This statement remains true for a mixed scaling function–wavelet basis where we have both nonvanishing s and d coefficients and for the case where the low and high resolution regions cover only parts of the cube of $N_1N_2N_3$ grid points.

The wave functions of degree 16 have an approximation error of h^8, i.e., the difference between the exact wave function and its representation in a finite basis set [Eq. (5)] is decreasing as h^8. The error of the kinetic energy in a variational scheme decreases then as $h^{2\cdot 8-2} = h^{14}$.[20] As we will see

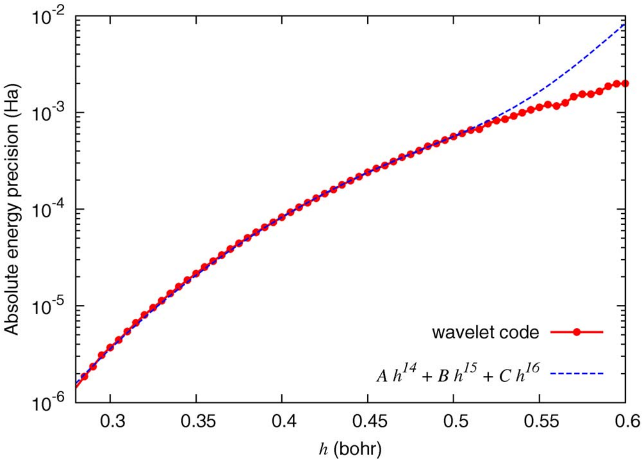

FIG. 3. (Color online) Convergence rate $\mathcal{O}(h^{14})$ of the wavelet code for a test run on a carbon atom. For this run the interpolation parameters are found to be within 2% accuracy: $A=344$, $B=-1239$, and $C=1139$. Using lower powers of h for the fit does not give accurate agreement. Other test systems gave comparable convergence rates.

the kinetic energy is limiting the convergence rate in our scheme and the overall convergence rate is thus h^{14}. Figure 3 shows this asymptotic convergence rate.

V. TREATMENT OF LOCAL POTENTIAL ENERGY

In spite of the striking advantages of wavelets, the initial exploration of this basis set[21] did not lead to any algorithm that would be useful for real electronic structure calculations. This was due to the fact that an accurate evaluation of the local potential energy is difficult in a wavelet basis.

By definition, the local potential $V(\mathbf{r})$ can be easily known on the nodes of the uniform grid of the simulation box. Approximating a potential energy matrix element $V_{i,j,k;i',j',k'}$,

$$V_{i,j,k;i',j',k'} = \int d\mathbf{r}\,\phi_{i',j',k'}(\mathbf{r})V(\mathbf{r})\phi_{i,j,k}(\mathbf{r}),$$

by

$$V_{i,j,k;i',j',k'} \approx \sum_{l,m,n} \phi_{i',j',k'}(\mathbf{r}_{l,m,n})V(\mathbf{r}_{l,m,n})\phi_{i,j,k}(\mathbf{r}_{l,m,n})$$

gives an extremely slow convergence rate with respect to the number of grid points used to approximate the integral because a single scaling function is not very smooth, i.e., it has a rather low number of continuous derivatives. Neelov and Goedecker[22] have shown that one should not try to approximate a single matrix element as accurately as possible but that one should try instead to approximate directly the expectation value of the local potential. The reason for this strategy is that the wave function expressed in the Daubechy basis is smoother than a single basis function. A single scaling function of order 16 (i.e., the corresponding wavelet has eight vanishing moments) has only two continuous derivatives. More precisely its index of Hölder continuity is about 2.7 and the Sobolev space regularity with respect to $p=2$ is about 2.91.[23] A single scaling function of order 16 has only four continuous derivatives. By suitable linear combinations of 16 one can however exactly represent polynomials up to degree 7, i.e., functions that have seven nonvanishing continuous derivatives. The discontinuities get thus canceled by taking suitable linear combinations. Since we use pseudopotentials, our exact wave functions are analytic and can locally be represented by a Taylor series. We are thus approximating functions that are approximately polynomials of order 7 and the discontinuities nearly cancel.

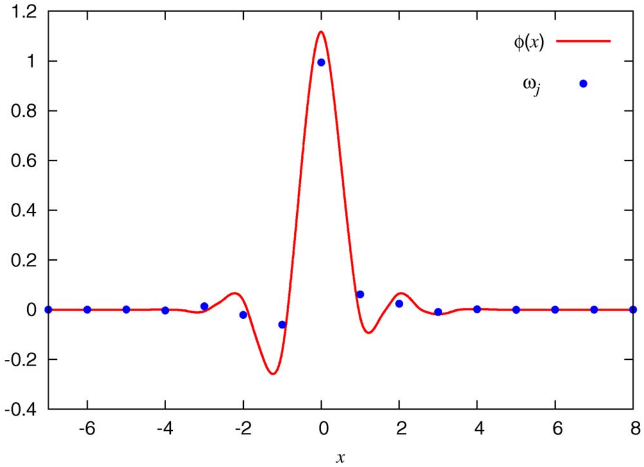

FIG. 4. (Color online) The magic filter ω_i for the least asymmetric Daubechies-16 basis.

Instead of calculating the exact matrix elements, we therefore use matrix elements with respect to a smoothed version $\tilde{\phi}$ of the scaling functions,

$$\begin{aligned} V_{i,j,k;i',j',k'} &\approx \sum_{l,m,n} \tilde{\phi}_{i',j',k'}(\mathbf{r}_{l,m,n})V(\mathbf{r}_{l,m,n})\tilde{\phi}_{i,j,k}(\mathbf{r}_{l,m,n}) \\ &= \sum_{l,m,n} \tilde{\phi}_{0,0,0}(\mathbf{r}_{l-i',m-j',n-k'})V(\mathbf{r}_{l,m,n}) \\ &\quad \times \tilde{\phi}_{0,0,0}(\mathbf{r}_{l-i,m-j,n-k}), \end{aligned} \tag{14}$$

where the smoothed wave function is defined by

$$\tilde{\phi}_{0,0,0}(\mathbf{r}_{l,m,n}) = \omega_l\omega_m\omega_n,$$

and ω_l is the "magic filter." The relation between the true functional values, i.e., the scaling function, and ω is shown in Fig. 4. Even though Eq. (14) is not a particularly good approximation for a single matrix element, it gives an excellent approximation for the expectation values of the local potential energy

$$\int dx \int dy \int dz \Psi(x,y,z)V(x,y,z)\Psi(x,y,z)$$

and also for matrix elements between different wave functions

$$\int dx \int dy \int dz \Psi_i(x,y,z)V(x,y,z)\Psi_j(x,y,z)$$

in case they are needed. Because of this remarkable achievement of the filter ω, we call it the magic filter.

In practice we do not explicitly calculate any matrix elements but we apply only filters to the wave function expan-

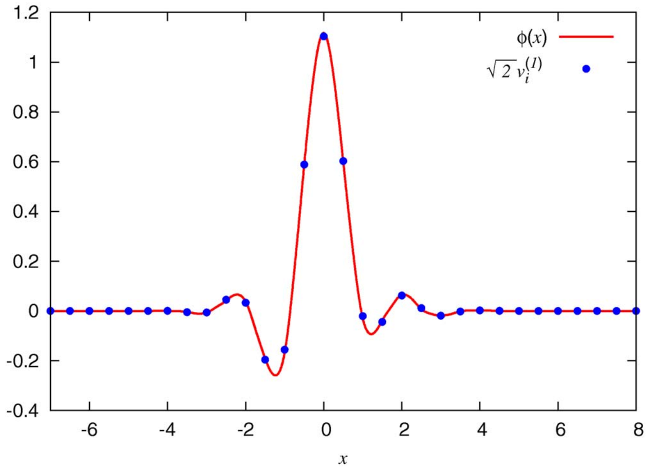

FIG. 5. (Color online) The fine scale magic filter $v_i^{(1)}$ (combination of a wavelet transform and the magic filter in Fig. 4) for the least asymmetric Daubechies-16 basis, scaled by $\sqrt{2}$ for comparison with the scaling function. The values of the filter on the graph are almost undistinguishable from the values of the scaling function. However, there is a slight difference that is important for the correct asymptotic convergence at small values of grid spacing h.

sion coefficients as will be shown in the following. This is mathematically equivalent but numerically much more efficient.

Since the operations with the local potential V are performed in the computational box on the double resolution grid with grid spacing $h' = h/2$, we must perform a wavelet transformation before applying the magic filters. These two operations can be combined in one, giving rise to modified magic filters both for scaling functions and wavelets on the original grid of spacing h. These modified magic filters can be obtained from the original ones using the refinement relations and they are shown in Figs. 5 and 6. Following the same guidelines as the kinetic energy filters, the smoothed real space values $\tilde{\Psi}_{i,j,k}$ of a wave function Ψ are calculated by performing a product of three one-dimensional convolutions with the magic filters along the x, y, and z directions. For the scaling function part of the wave function, the corresponding formula is

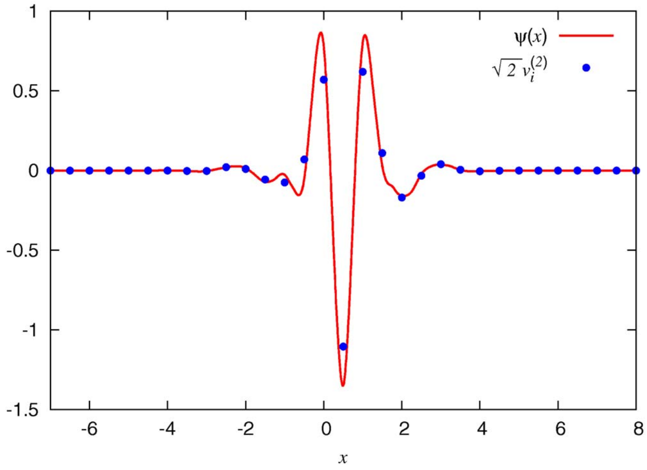

FIG. 6. (Color online) The fine scale magic filter $v_i^{(2)}$ (combination of a wavelet transform and the magic filter in Fig. 4) for the least asymmetric Daubechies-16 *wavelet*, scaled by $\sqrt{2}$ for comparison with the wavelet itself.

$$\tilde{\Psi}_{i_1,i_2,i_3} = \sum_{j_1,j_2,j_3} s_{j_1,j_2,j_3} v^{(1)}_{i_1-2j_1} v^{(1)}_{i_2-2j_2} v^{(1)}_{i_3-2j_3},$$

where $v_i^{(1)}$ is the filter that maps a scaling function on a double resolution grid. Similar convolutions are needed for the wavelet part. The calculation is thus similar to the treatment of the Laplacian in the kinetic energy.

Once we have calculated $\tilde{\Psi}_{i,j,k}$ the approximate expectation value ϵ_V of the local potential V for a wave function Ψ is obtained by simple summation on the double resolution real space grid,

$$\epsilon_V = \sum_{j_1,j_2,j_3} \tilde{\Psi}_{j_1,j_2,j_3} V_{j_1,j_2,j_3} \tilde{\Psi}_{j_1,j_2,j_3}.$$

The evaluation of the local potential energy ϵ_V converges with a convergence rate of h^{16} to the exact value where h is the grid spacing. Therefore, the potential energy has a convergence rate two powers of h faster than the rate for the kinetic energy.

VI. CALCULATION OF HARTREE POTENTIAL

We saw in the section on the treatment of the local potential energy how to express efficiently the point values of the smoothed wave function $\tilde{\Psi}$ on the fine grid mesh. From these values the charge density on a grid point j_1, j_2, j_3 of the double resolution grid is given by

$$\rho_{j_1,j_2,j_3} = \sum_i n_{\text{occ}}^{(i)} \tilde{\Psi}^2_{i;j_1,j_2,j_3}, \tag{15}$$

where $n_{\text{occ}}^{(i)}$ are the occupation numbers. For a closed shell system they equal 2 for the occupied orbitals and zero for all other orbitals. The discrete charge density ρ_{j_1,j_2,j_3} is a very good approximation to the charge distribution of the continuous wave functions $|\Psi\rangle$ in the sense that the first multipoles of the discrete charge distribution converge rapidly to the values of the continuous charge distribution. The monopole converges with a rate of h^{16}. For each higher multipole moment the convergence rate is reduced by one power of h, i.e., dipoles converge with a rate of h^{15}, quadrupoles with h^{14}, etc. The discrete charge density ρ on the double resolution grid is then the input to various Poisson solvers that are available for different boundary conditions. In the case of free boundary conditions, appropriate for isolated molecules, the values ρ_{j_1,j_2,j_3} form the coefficients for an expansion in interpolating scaling functions of order 16. This expansion strictly conserves all the multipoles up to the angular moment $\ell = 15$ and allows us to solve the integral equation for the potential explicitly with the correct boundary conditions.[24] In addition to free boundary conditions, we have also implemented surface boundary conditions,[25] i.e., periodicity in two directions and free boundary conditions in the third direction. In this case the charge density is represented in a mixed plane-wave-scaling function representation.

These Poisson solvers have a convergence rate of h'^m, where m is the order of the interpolating scaling functions used to express the Poisson kernel. Since we use interpolating scaling functions of order 16, the convergence rate of the electrostatic potential is faster than the rate for the kinetic energy. All these Poisson Solvers have one thing in common: they perform explicitly the convolution of the density with Green's functions of Poisson's equation. The necessary convolutions are done by a traditional zero-padded fast Fourier transform (FFT) procedure that leads to an $\mathcal{O}(N \log N)$ operation count with respect to the number of grid points N. The accuracy of the potential is uniform over the whole volume and one can thus use the smallest possible volume compatible with the requirement that the tails of the wave functions have decayed to very small values at the surface of this volume. The fraction of the computational time needed for the solution of Poisson's equation decreases with increasing system size and is roughly 1% for large systems, see Sec. XVI. Moreover, the explicit Green's function treatment of the Poisson's solver allows us to treat isolated systems with a net charge directly without the insertion of compensating charges.

VII. XC FUNCTIONALS AND IMPLEMENTATION OF GENERALIZED GRADIENT APPROXIMATION (GGA)

The charge density expression used to calculate the Hartree potential,

$$\rho(\mathbf{r}) = \sum_i n_{\mathrm{occ}}^{(i)} |\tilde{\Psi}_i(\mathbf{r})|^2, \tag{16}$$

is also used for the calculation of the XC energy E_{xc} and the corresponding potential V_{xc}. Any real space based implementation of the XC functionals fits well with this density representation. In our program we use the XC functionals as implemented in ABINIT code. To this aim, we use the same ABINIT XC routines to calculate the XC energy

$$E_{\mathrm{xc}} = \int \rho(\mathbf{r})\epsilon_{\mathrm{xc}}(\mathbf{r}) d\mathbf{r}, \tag{17}$$

together with the XC potential

$$V_{\mathrm{xc}}(\mathbf{r}) = \frac{\delta E_{\mathrm{xc}}}{\delta\rho(\mathbf{r})}. \tag{18}$$

Also spin-polarized (collinear) version of the ABINIT XC functionals can be used with our method.

In the case of GGA XC functionals, the XC energy density depends both on the local values of the charge density ρ and on the modulus of its gradient,

$$\epsilon_{\mathrm{xc}}(\mathbf{r}) = \epsilon_{\mathrm{xc}}(\rho(\mathbf{r}), |\nabla\rho|(\mathbf{r})). \tag{19}$$

A traditional finite difference scheme of fourth order is used on the double resolution grid to calculate the gradient of the charge density

$$\partial_w \rho(\mathbf{r}_{i_1,i_2,i_3}) = \sum_{j_1,j_2,j_3} c^{(t)}_{i_1,i_2,i_3;j_1,j_2,j_3} \rho_{j_1,j_2,j_3}, \tag{20}$$

where $w = x, y, z$. For grid points close to the boundary of the computational volume, the above formula requires grid points outside the volume. For free boundary conditions the values of the charge density outside the computational volume in a given direction are taken to be equal to the value at the border of the grid.

The relation between the gradient and the density must be taken into account when calculating V_{xc} in the standard White–Bird[26] approach, where the density gradient is considered as an explicit functional of the density. There the XC potential can be split in two terms,

$$V_{\mathrm{xc}}(\mathbf{r}_{i_1,i_2,i_3}) = V^o_{\mathrm{xc}}(\mathbf{r}) + V^c_{\mathrm{xc}}(\mathbf{r}), \tag{21}$$

where

$$V^o_{\mathrm{xc}}(\mathbf{r}_{i_1,i_2,i_3}) = \epsilon_{\mathrm{xc}}(\mathbf{r}) + \rho(\mathbf{r}) \frac{\partial \epsilon_{\mathrm{xc}}}{\partial \rho}(\mathbf{r}),$$

$$\begin{aligned} V^c_{\mathrm{xc}}(\mathbf{r}_{i_1,i_2,i_3}) = &\sum_{j_1,j_2,j_3} \frac{\rho}{|\nabla\rho|} \frac{\partial \epsilon_{\mathrm{xc}}}{\partial |\nabla\rho|}(\mathbf{r}_{j_1,j_2,j_3}) \\ &\times \sum_{w=x,y,z} \partial_w \rho(\mathbf{r}_{j_1,j_2,j_3}) c^{(w)}_{j_1,j_2,j_3;i_1,i_2,i_3}, \end{aligned} \tag{22}$$

where the "ordinary" part V^o_{xc} is present in the same form of LDA functionals, while the White–Bird[26] "correction" term V^c_{xc} appears only when the XC energy depends explicitly on $|\nabla\rho|$. The $c^{(w)}$ are the coefficients of the finite difference formula used to calculate the gradient of the charge density (20).

The evaluation of the XC terms and also, when needed, the calculation of the gradient of the charge density may easily be performed together with the Poisson solver used to evaluate the Hartree potential. This allows us to save computational time.

VIII. TREATMENT OF THE NONLOCAL PSEUDOPOTENTIAL

The energy contributions from the nonlocal pseudopotential have for each angular moment l the form

$$\sum_{i,j} \langle \Psi | p_i \rangle h_{ij} \langle p_j | \Psi \rangle,$$

where $|p_i\rangle$ is a pseudopotential projector. Once applying the Hamiltonian operator, the application of one projector on the wave functions requires the calculation of

$$|\Psi\rangle \rightarrow |\Psi\rangle + \sum_{i,j} |p_i\rangle h_{ij} \langle p_j | \Psi \rangle.$$

If we use for the projectors the representation of Eq. (5) (i.e., the same as for the wave functions), both operations are trivial to perform. Because of the orthogonality of the basis set, we just have to calculate scalar products among the coefficient vectors and to update the wave functions. The scaling function and wavelet expansion coefficients for the projectors are given by[13]

$$\int p(\mathbf{r}) \phi_{i_1,i_2,i_3}(\mathbf{r}) d\mathbf{r}, \quad \int p(\mathbf{r}) \psi^{\nu}_{i_1,i_2,i_3}(\mathbf{r}) d\mathbf{r}, \tag{23}$$

where we used the notation.[3,4]

The GTH-HGH pseudopotentials[15,16] have projectors that are written in terms of Gaussians times polynomials.

This form of projectors is particularly convenient to be expanded in the basis. In other terms, since the general form of the projector is

$$\langle \mathbf{r}|p\rangle = e^{-cr^2}x^{\ell_x}y^{\ell_y}z^{\ell_z},$$

the three-dimensional integrals can be calculated easily since they can be factorized into a product of three one-dimensional integrals,

$$\int \langle \mathbf{r}|p\rangle \phi_{i_1,i_2,i_3}(\mathbf{r})d\mathbf{r} = W_{i_1}(c,l_x)W_{i_2}(c,l_y)W_{i_3}(c,l_x), \tag{24}$$

$$W_j(c,\ell) = \int_{-\infty}^{+\infty} e^{-ct^2}t^{\ell}\phi(t/h-j)dt. \tag{25}$$

The one-dimensional integrals are calculated in the following way. We first calculate the scaling function expansion coefficients for scaling functions on a one-dimensional grid that is 16 times denser. The integration on this dense grid is done by the well-known quadrature introduced in Ref. 27 that coincides with the magic filter.[22] This integration scheme based on the magic filter has a convergence rate of h^{16} and we therefore gain a factor of 16^{16} in accuracy by going to a denser grid. This means that the expansion coefficients are for reasonable grid spacings h accurate to machine precision. After having obtained the expansion coefficients with respect to the fine scaling functions, we obtain the expansion coefficients with respect to the scaling functions and wavelets on the required resolution level by one-dimensional fast wavelet transformations. No accuracy is lost in the wavelet transforms and our representation of the projectors is therefore typically accurate to nearly machine precision. In order to treat with the same advantages other pseudopotentials that are not given under the form of Gaussians, it would be necessary to approximate them by a small number of Gaussians.

IX. PRECONDITIONING

As already mentioned, direct minimization of the total energy is used to find the converged wave functions. The gradient g_i of the total energy with respect to the ith wave function $|\Psi_i\rangle$ is given by

$$|g_i\rangle = H|\Psi_i\rangle - \sum_j \Lambda_{ij}|\Psi_j\rangle, \tag{26}$$

where $\Lambda_{ij}=\langle\psi_j|H|\psi_i\rangle$ are the Lagrange multipliers enforcing the orthogonality constraints. Convergence is achieved when the average norm of the residue $\langle g_i|g_i\rangle^{1/2}$ is below a user-defined numerical tolerance.

Given the gradient direction at each step, several algorithms can be used to improve convergence. In our method we use either preconditioned steepest-descent algorithm or preconditioned direct inversion in the iterative subspace (DIIS) method.[28,29] These methods work very well to improve the convergence for nonzero gap systems if a good preconditioner is available.

The preconditioning gradient $|\tilde{g}_i\rangle$ that approximately points in the direction of the minimum is obtained by solving the linear system of equations obtained by discretizing the equation

$$\left(\tfrac{1}{2}\nabla^2 - \epsilon_i\right)\tilde{g}_i(\mathbf{r}) = g_i(\mathbf{r}). \tag{27}$$

The values ϵ_i are approximate eigenvalues obtained by a subspace diagonalization in a minimal basis of atomic pseudopotential orbitals during the generation of the input guess. For isolated systems, the values of ϵ_i for the occupied states are always negative; therefore the operator of Eq. (27) is positive definite.

Equation (27) is solved by a preconditioned conjugate gradient (CG) method. The preconditioning is done by using the diagonal elements of the matrix representing the operator $\frac{1}{2}\nabla^2-\epsilon_i$ in a scaling function–wavelet basis. In the initial step we use ℓ resolution levels of wavelets, where ℓ is typically 4. To do this we have to enlarge the domain where the scaling function part of the gradient is defined to a grid that is a multiple of 2^{ℓ}. This means that the preconditioned gradient $\tilde{g}_i$ will also exist in a domain that is larger than the domain of the wave function Ψ_i. Nevertheless this approach is useful since it allows us to obtain rapidly a preconditioned gradient that has the correct overall shape. In the following iterations of the CG, we use only one wavelet level in addition to the scaling functions for preconditioning. In this way we can do the preconditioning exactly in the domain of basis functions that are used to represent the wave functions [Eq. (5)]. The typical number of CG iterations necessary to obtain a meaningful preconditioned gradient is 5.

X. ORTHOGONALIZATION

We saw the need of keeping the wave functions Ψ_i orthonormal at each step of the minimization loop. This means that the overlap matrix S, with matrix elements

$$S_{ij} = \langle\Psi_j|\Psi_i\rangle, \tag{28}$$

must be equal to the identity matrix.

All the orthogonalization algorithms have a cubic complexity causing this part of the program to dominate for large systems (see Fig. 11). We therefore optimized this part carefully and found that a pseudo-Gram–Schmidt algorithm that uses a Cholesky factorization of the overlap matrix S is the most efficient method on parallel computers. In the following, we discuss the reasons for this choice by comparing it to two other orthogonalization algorithms: classical Gram–Schmidt and Loewdin orthogonalizations.

A. Gram–Schmidt orthogonalization

The classical Gram–Schmidt orthonormalization algorithm generates an orthogonal set of orbital $\{|\bar{\Psi}_i\rangle\}$ out of a nonorthogonal set $\{|\Psi_i\rangle\}$ by processing separately each orbital. The overlap of the currently processed orbital $|\Psi_i\rangle$ with the set of the already processed orbitals $\{|\bar{\Psi}_j\rangle\}_{j=1\cdots i-1}$ is calculated and is removed from $|\Psi_i\rangle$. Thereafter, the transformed orbital $|\bar{\Psi}_i\rangle$ is normalized,

$$|\bar{\Psi}_i\rangle = |\Psi_i\rangle - \sum_{j=1}^{i-1}\langle\bar{\Psi}_j|\Psi_i\rangle|\bar{\Psi}_j\rangle, \tag{29}$$

$$|\bar{\Psi}_j\rangle \rightarrow \frac{|\bar{\Psi}_j\rangle}{\sqrt{\langle\bar{\Psi}_j|\bar{\Psi}_j\rangle}}. \tag{30}$$

The algorithm consists of the calculation of $n(n+1)/2$ scalar products and wave function updates. If the coefficients of each orbital are distributed among several processors, $n(n+1)/2$ communication steps are needed to sum up the various contributions from each processor to each scalar product. Such a large number of communication steps leads to a large latency overhead on a parallel computer and therefore to poor performances.

B. Loewdin orthogonalization

The Loewdin orthonormalization algorithm is based on the following equation:

$$|\bar{\Psi}_i\rangle = \sum_j S_{ij}^{-1/2}|\Psi_j\rangle, \tag{31}$$

where a new set of orthonormal orbitals $|\bar{\Psi}_i\rangle$ is obtained by multiplying the inverse square root of the overlap matrix S with the original orbital set.

The implementation of this algorithm requires that the symmetric overlap matrix S is calculated. In contrast to the classical Gram–Schmidt algorithm, the matrix elements S_{ij} depend on the original set of orbitals and can be calculated in parallel in the case where each processor holds a certain subset of the coefficients of each wave function. At the end of this calculation a single communication step is needed to sum up the entire overlap matrix out of the contributions to each matrix element calculated by the different processors. Since S is a Hermitian positive definite matrix, there exists a unitary matrix U that diagonalizes $S=U^*\Lambda U$, where Λ is a diagonal matrix with positive eigenvalues. The inverse square root of S is then given by $S^{-1/2}=U^\dagger\Lambda^{-1/2}U$. Hence, an eigenvalue problem must be solved in order to find U and Λ.

C. Pseudo-Gram–Schmidt using Cholesky factorization

In this scheme a Cholesky factorization of the overlap matrix $S=LL^T$ is calculated. The new orthonormal orbitals are obtained by

$$|\bar{\Psi}_i\rangle = \sum_j (L_{ij}^{-1})|\Psi_j\rangle \tag{32}$$

and are equivalent to the orbitals obtained by the classical Gram–Schmidt. The procedure for calculating the overlap matrix out of the contributions calculated by each processor is identical to the Loewdin case. Instead of solving an eigenvalue problem, we have however to calculate the decomposition of the overlap matrix. This can be done much faster. This algorithm also requires only one communication step on a parallel computer but has a lower prefactor than the Loewdin scheme.

XI. CALCULATION OF FORCES

Atomic forces can be calculated with the same method used for the application of the Hamiltonian onto a wave function. Since the scaling function/wavelet basis is not moving together with atoms, we have no Pulay[30] forces and atomic forces can be evaluated directly through the Feynman–Hellmann theorem. Except for the force arising from the trivial ion-ion interaction, which for the ith atom is

$$\mathbf{F}_i^{(\text{ionic})} = \sum_{j\neq i}\frac{Z_iZ_j}{R_{ij}^3}(\mathbf{R}_i-\mathbf{R}_j), \tag{33}$$

the energy terms that depend explicitly on the atom positions are related to the pseudopotentials. As shown in the previous sections, the GTH-HGH pseudopotentials we are using are based on separable functions[15,16] and can be splitted into a local and a nonlocal contribution.

For an atom i placed at position $\mathbf{R}_i$, the contribution to the energy that comes from the local part of the pseudopotential is

$$E_{\text{local}}(\mathbf{R}_i) = \int d\mathbf{r}V_{\text{local}}(|\mathbf{r}-\mathbf{R}_i|)\rho(\mathbf{r}), \tag{34}$$

where the local pseudopotential can be split into long- and a short-ranged terms $V_{\text{local}}(\lambda)=V_L(\lambda)+V_S(\lambda)$, and

$$V_L(\lambda) = -\frac{Z_i}{\lambda}\,\text{erf}\left(\frac{\lambda}{\sqrt{2}r_\ell}\right),$$

$$V_S(\lambda) = \exp\left(-\frac{\lambda^2}{2r_\ell^2}\right)\left[C_1 + C_2\left(\frac{\lambda}{r_\ell}\right)^2 + C_3\left(\frac{\lambda}{r_\ell}\right)^4 + C_4\left(\frac{\lambda}{r_\ell}\right)^6\right], \tag{35}$$

where the C_i and r_ℓ are the pseudopotential parameters, depending on the atom of atomic number Z_i under consideration. The energy contribution $E_{\text{local}}(\mathbf{R}_i)$ can be rewritten in an equivalent form. It is straightforward to verify that

$$E_{\text{local}}(\mathbf{R}_i) = \int d\mathbf{r}\rho_{\text{L}}(|\mathbf{r}-\mathbf{R}_i|)V_H(\mathbf{r}) + \int d\mathbf{r}V_S(|\mathbf{r}-\mathbf{R}_i|)\rho(\mathbf{r}), \tag{36}$$

where V_H is the Hartree potential and ρ_L is such that $\nabla_{\mathbf{r}}^2V_L(|\mathbf{r}-\mathbf{R}_i|)=-4\pi\rho_L(|\mathbf{r}-\mathbf{R}_i|)$. This analytical transformation remains also valid in our procedure for solving the discretized Poisson's equation. From Eq. (36) we can calculate

$$\rho_L(\lambda) = -\frac{1}{(2\pi)^{3/2}}\frac{Z_i}{r_\ell^3}e^{-\lambda^2/2r_\ell^2}, \tag{37}$$

which is a localized (thus short-ranged) function. The forces coming from the local pseudopotential are thus

$$\mathbf{F}_i^{(\text{local})} = -\frac{\partial E_\ell(\mathbf{R}_i)}{\partial\mathbf{R}_i} = \frac{1}{r_\ell}\int d\mathbf{r}\frac{\mathbf{r}-\mathbf{R}_i}{|\mathbf{r}-\mathbf{R}_i|}[\rho_L'(|\mathbf{r}-\mathbf{R}_i|)V_H(\mathbf{r}) + V_S'(|\mathbf{r}-\mathbf{R}_i|)\rho(\mathbf{r})], \tag{38}$$

where

$$\rho_L'(\lambda) = \frac{1}{(2\pi)^{3/2}}\frac{Z_{\text{ion}}}{r_{\text{loc}}^4}\lambda e^{-\lambda^2/2r_\ell^2},$$

$$V_S'(\lambda) = \frac{\lambda}{r_\ell} e^{-\lambda^2/2r_\ell^2}\Bigg[(2C_2 - C_1) + (4C_3 - C_2)\left(\frac{\lambda}{r_\ell}\right)^2 + (6C_4 - C_3)\left(\frac{\lambda}{r_\ell}\right)^4 - C_4\left(\frac{\lambda}{r_\ell}\right)^6\Bigg]. \qquad (39)$$

Within this formulation, the contribution to the forces from the local part of pseudopotential is written in terms of integrals with localized functions (Gaussians times polynomials) times the charge density and the Hartree potential. This allows us to perform the integrals only in a relatively small region around the atom position and to assign different integrations to different processors. Moreover, the calculation is performed with almost linear $[\mathcal{O}(N \log N)]$ scaling.

The contribution to the energy that comes from the nonlocal part of the pseudopotential is, as we saw in Sec. VIII,

$$E_{\text{nonlocal}}(\mathbf{R}_i) = \sum_l \sum_{mn} \langle \Psi | p_m^l(\mathbf{R}_i)\rangle h_{mn}^l \langle p_n^l(\mathbf{R}_i)|\Psi\rangle, \qquad (40)$$

where we wrote explicitly the dependence of the projector on the atom position $\mathbf{R}_i$. The contribution of this term to the atomic forces is thus

$$\mathbf{F}_i^{(\text{nonlocal})} = -\sum_l \sum_{m,n} \left\langle \Psi \Big| \frac{\partial p(\mathbf{R}_i)}{\partial \mathbf{R}_i} \right\rangle h_{mn} \langle p(\mathbf{R}_i)|\Psi\rangle - \sum \langle \Psi | p(\mathbf{R}_i)\rangle h_{mn} \left\langle \frac{\partial p(\mathbf{R}_i)}{\partial \mathbf{R}_i} \Big| \Psi \right\rangle. \qquad (41)$$

Expressing the derivatives of the projectors in the basis, the evaluation of the scalar products is straightforward. The scaling function–wavelet expansion coefficients of the projector derivatives can be calculated with machine precision accuracy in the same way as the projectors themselves were calculated. This is due to the fact that the derivative of the projectors is like the projectors themselves products of Gaussians and polynomials.

XII. LOCALIZATION PROPERTIES AND SMOOTHNESS OF THE BASIS FUNCTIONS

As discussed above, basis functions are suitable for expanding localized functions. There is no need to put basis functions on grid points that are far from the atoms. For this reason, we choose to associate the basis functions to points lying inside the union of atom-centered spheres defined by their radii. This operation must be performed both for the high and low resolution grid points (see Fig. 2). In our method, we measure these radii in two different units. For the high resolution region the radius is expressed in terms of the shortest localization radius of the atom pseudopotential. For the low resolution region, the distance is expressed in units of the asymptotic decaying length of the atomic wave function $1/\sqrt{2\epsilon_{\text{HOMO}}}$, calculated from the energy ϵ_{HOMO} of the highest occupied atomic orbital, obtained from Ref. 31. In this way we can easily determine nearly optimal sizes for the high and low resolution regions and minimize the number of degrees of freedom to achieve a target accuracy (Sec. XVI).

We saw that wavelets have the property that linear combinations of them can be smoother than a single scaling function or wavelet. The wave function of Eq. (5) is thus typically smoother than the scaling functions and wavelets used to represent it. The reduced smoothness of scaling function of order 16 in the tail region can be seen from Fig. 7. The cancellation of discontinuities in the basis set by suitable linear combinations is only possible in an infinite interval where several basis functions are present between any two grid points. Since we use a finite grid of scaling functions in the tail region, the number of scaling functions that contribute to the value of the wave function at a certain point is dropping as we are going out of the computational volume. The outermost intervals of the wave function are actually only described by the tail of a single scaling function. Hence the wave function is getting less smooth toward its end. This reduced smoothness affects principally the kinetic energy. For systems without a net charge, far from the atoms the potential is very small and for this reason errors in the potential energy are decreasing exponentially with respect to the size of the computational volume.

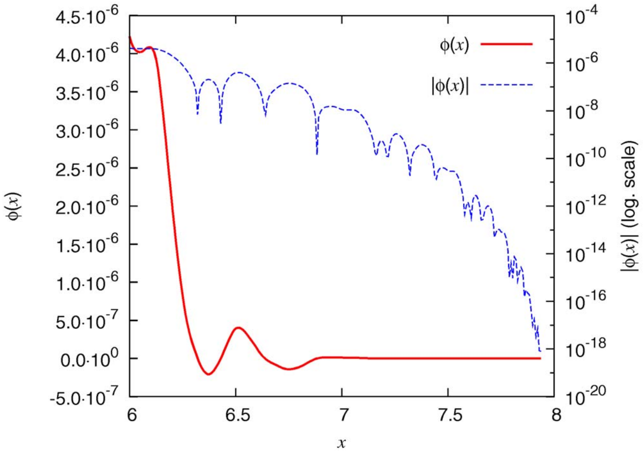

FIG. 7. (Color online) Zoom of the Daubechies (Ref. 3) scaling function near the border of its support. Both the function and its absolute value are plotted.

XIII. PERTURBATIVE CALCULATION OF THE FINITE SIZE CORRECTIONS

Far from the atoms, each wave function decays exponentially with a decay rate that depends on its KS eigenvalue.[32] If A is the amplitude of the tail of the wave function, the kinetic energy contribution of the nonsmooth wave function in its tail region is of the order of A/h^2, whereas the exact wave function has a kinetic energy of the order of A. As a consequence the kinetic energy error increases as one decreases h and the total energy increases as well if the computational volume is too small. We know, however, that the contribution to the kinetic energy in this region will depend uniquely on the asymptotic behavior of the wave function, which is governed by its KS eigenvalue. In other terms, the magnitude of the kinetic energy error due to the localization of the system in a finite volume can, in principle, be estimated by knowing the KS eigenvalue of the wave function.

If, on the other hand, the computational volume is large enough such that amplitude A is very small, our method shows a strict variational behavior with a convergence rate of h^{14} over a large range of grid spacings h. This is illustrated in Fig. 3.

The above described facts prompted us to develop a method that cuts off the wave function tail at a very large radius but which is computationally much less expensive than a fully self-consistent calculation in a very large computational volume. We do first a fully self-consistent calculation in a medium size box and we then add afterward the missing far tail to the wave function. Let us denote the wave function that we have calculated in the medium size box by $|\Psi\rangle$ and the wave function in the very large box by $|\Psi\rangle + |\Delta\Psi\rangle$. As we will see $|\Delta\Psi\rangle$ is negligible inside the medium size box. It is essentially the tail outside the original medium size box plus a part that cancels the nonsmooth behavior in the surface region of the medium size box. Evidently $|\Psi\rangle + |\Delta\Psi\rangle$ has to satisfy the Schrödinger equation

$$\left(\tfrac{1}{2}\nabla^2 + V(\mathbf{r})\right)(|\Psi\rangle + |\Delta\Psi\rangle) = \epsilon(|\Psi\rangle + |\Delta\Psi\rangle).$$

Rearranging the term one obtains

$$\left(\tfrac{1}{2}\nabla^2 + V(\mathbf{r}) - \epsilon\right)|\Delta\Psi\rangle = -\left(\tfrac{1}{2}\nabla^2 + V(\mathbf{r}) - \epsilon\right)|\Psi\rangle. \qquad (42)$$

The term on the right hand side of the above equation is the gradient $|g\rangle$ that is needed in any minimization scheme. When the calculation of the wave functions is converged, the gradient is zero (actually less than a small numerical tolerance) when projected onto the subspace of the basis functions spanning the medium size volume. The gradient is, however, not anymore zero when it is projected onto the basis set of the larger volume. In this case the projection onto the basis function just outside the medium size volume gives a nonzero contribution. Remember that the fact that these basis functions are missing in the basis set of the medium size volume is causing the nonsmooth behavior. Projection on basis functions that are far outside the surface region of the medium size volume is again zero since $|\Psi\rangle$ is identically zero. So, in this context, the gradient is a quantity that is nonzero only in a small shell outside the original medium size volume. The width of this shell is given by the length of the kinetic energy filter. Since the potential is very small in the tail region, Eq. (43) can be approximated by

$$\left(\tfrac{1}{2}\nabla^2 - \epsilon\right)|\Delta\Psi\rangle = |g\rangle.$$

As usual in a perturbative treatment we rely on the fact that the eigenvalues ϵ converge faster than the wave function and the zeroth order eigenvalues can therefore be used for the first order correction to the wave function. The above equation is identical to the preconditioning equation [Eq. (27)] and can be solved with the same method, just within a larger volume. In this way we can eliminate in a single preconditioning step at the end of the fully self-consistent calculation in the medium size volume a large fraction of the error arising from cutting off the wave functions at the surface of our computational volume. We can thus have a reliable estimation of the approximation resulting from the restriction of the system to a finite computational volume. Figure 8 shows an example of the convergence rate of the total energy with respect to the size of the computational volume both with and without tail corrections for two different grid spacings.

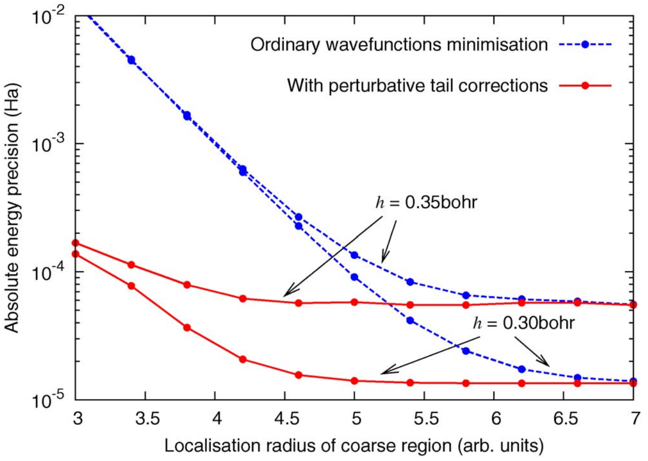

FIG. 8. (Color online) Absolute convergence of the total energy of a methane molecule as a function of the low resolution localization radius with and without the tail corrections. The curves for two different values of the grid spacing are plotted, showing the h convergence for the localization parameter sufficiently extended.

XIV. PARALLELIZATION

Two data distribution schemes are used in the parallel version of our program. In the orbital distribution scheme, each processor works on one or a few orbitals for which it holds all its scaling function and wavelet coefficients. In the coefficient distribution scheme each processor holds a certain subset of the coefficients of all the orbitals. Most of the operations such as applying the Hamiltonian on the orbitals and the preconditioning are done in the orbital distribution scheme. This has the advantage that we do not have to parallelize these routines and we therefore achieve almost perfect parallel speedup. The calculation of the Lagrange multipliers that enforce the orthogonality constraints onto the gradient as well as the orthogonalization of the orbitals is done in the coefficient distribution scheme. For the orthogonalization we have to calculate the matrix $\langle\Psi_j|\Psi_i\rangle$ and for the Lagrange multipliers the matrix $\langle\Psi_j|H|\Psi_i\rangle$. So each matrix element is a scalar product and each processor is calculating the contribution to this scalar product from the coefficients it is holding. A global reduction sum is then used to sum the contributions to obtain the correct matrix. Such sums can easily be performed with the very well optimized BLAS-LAPACK libraries. Switching back and forth between the orbital distribution scheme and the coefficient distribution scheme is done by the MPI global transposition routine MPI_ALLTOALL. For parallel computers where the cross sectional bandwidth[33] scales well with the number of processors, this global transposition does not require a lot of CPU time. The most time consuming communication is the global reduction sum required to obtain the total charge distribution from the partial charge distribution of the individual orbital [sum in Eq. (15)].

XV. CALCULATION OF UNOCCUPIED ORBITALS

In order to calculate the unoccupied KS orbitals, we use the Davidson[34] method after having found the self-consistent occupied KS orbitals. An initial guess for the N_{virt} unoccu-

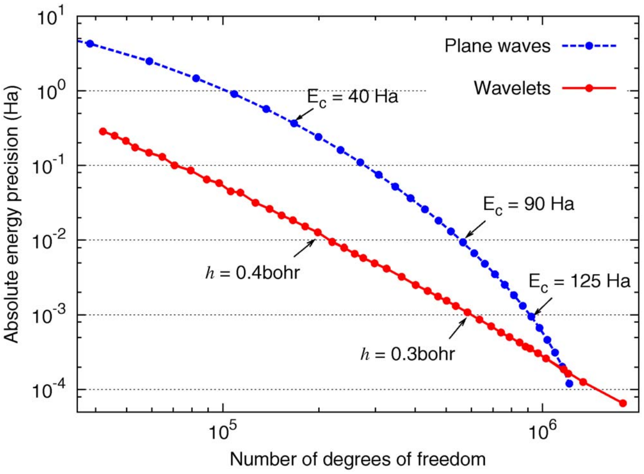

FIG. 9. (Color online) Absolute precision (not precision per atom) as a function of the number of degrees of freedom for a cinchonidine molecule (44 atoms). Our method is compared with a plane wave code. In the case of the plane wave code, the plane wave cutoff and the volume of the computational box were chosen such as to obtain the required precision with the smallest number of degrees of freedom. In the case of our wavelet program, the grid spacing h and the localization radii were optimized. For very high accuracies the exponential convergence rate of the plane waves beats the algebraic convergence rate of the wavelets. Such high accuracies are however not required in practice. Since convolutions can be executed at very high speed, the wavelet code is faster than the plane wave code at any accuracy even if the number of degrees of freedom are similar (see Table I).

pied eigenvectors Ψ_j and eigenvalues ϵ_j of the KS Hamiltonian H_{KS} is obtained from the subspace diagonalization in a minimal atomic basis set that is also used to generate the input guess for the occupied orbitals. For any given set of virtual orbitals, we calculate then the gradients [Eq. (26) where the Lagrange multipliers ensure only orthogonality to the occupied orbitals] and then precondition these gradients according to Eq. (27). A subspace diagonalization is then done in the space spanned by the present set of approximate eigenvectors and their preconditioned gradients. In the original Davidson method the dimension of the subspace is increased in each iteration since one keeps all the previous preconditioned gradients in the subspace. To save memory we have limited the dimension of the subspace in each iteration to $2N_{\text{virt}}$ using only the present set of approximate eigenvectors together with their preconditioned gradients. Even though the number of requested unoccupied orbitals is typically small (frequently only the lowest unoccupied molecular orbital), a larger set of vectors N_{virt} is considered in our method (in a parallel calculation at least one per processor), but only the gradients of the desired number of orbitals are taken into account for the convergence criterion for the norm of the gradients. This, together with the fact that our preconditioner is rather good, allows us to achieve fast convergence rates comparable to the ones achieved in the calculation of the occupied orbitals. Some 20 iterations are typically needed.

XVI. PERFORMANCE RESULTS

TABLE I. Computational time in seconds for a single minimization iteration for different runs of the cinchonidine molecule used for the plot in Fig. 9. The timings for different cutoff energies E_c for the plane wave runs are shown. The input parameters for the wavelet runs are chosen such as to obtain the same absolute precision of the plane wave calculations. The plane wave runs are performed with the ABINIT code, which uses iterative diagonalization and with CPMD code (Ref. 35) in direct minimization. These timings are taken from a serial run on a 2.4 GHz AMD Opteron CPU.

E_c (hartree)	ABINIT (s)	CPMD (s)	Abs. Precision	Wavelets (s)
40	403	173	3.7×10^{-1}	30
50	570	207	1.6×10^{-1}	45
75	1123	422	2.5×10^{-2}	94
90	1659	538	9.3×10^{-3}	129
145	4109		2×10^{-4}	474

We have applied our method on different molecular systems in order to test its performances. As expected, the localization of the basis set allows us to reduce considerably the number of degrees of freedom (i.e., the number of basis functions that must be used) to attain a given absolute precision with respect to a plane wave code. This fact reduces the memory requirements and the number of floating point operations. Figure 9 shows the comparison of the absolute precision in a calculation of a 44 atom molecule as a function of the number of degrees of freedom used for the calculation. In Table I the comparison of the timings of a single SCF cycle with respect to two other plane wave based codes is shown. Since the system is relatively small the cubic terms do not dominate. For large systems of several hundred atoms, the gain in CPU time compared to a plane wave program is proportional to the reduction in the number of degrees of freedom [compare Eq. (44)] and can thus be very significant as one can conclude from Fig. 9.

The parallellization scheme of the code has been tested and has given the efficiency detailed in Fig. 10. The overall efficiency is always higher than 88%, also for large systems with a big number of processors.

It is also interesting to see which is the computational share of the different sections of the code with respect to the total execution time. Figure 11 shows the percentage of the computational time for the different sections of the code as a

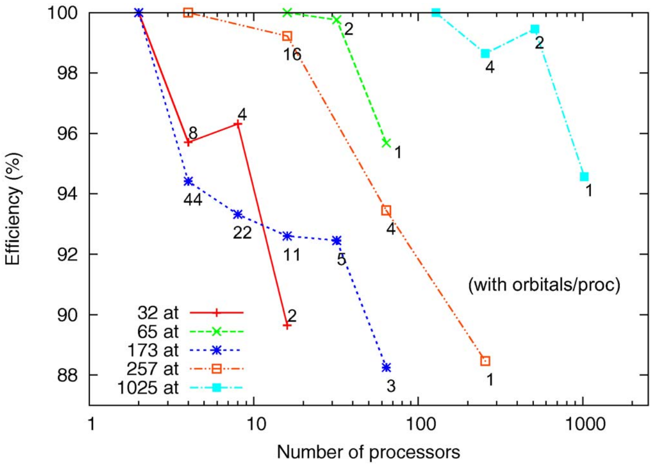

FIG. 10. (Color online) Efficiency of the parallel implementation of the code for several runs with different numbers of atoms. The number close to each point indicates the number of orbitals treated by each processor in the orbital distribution scheme.

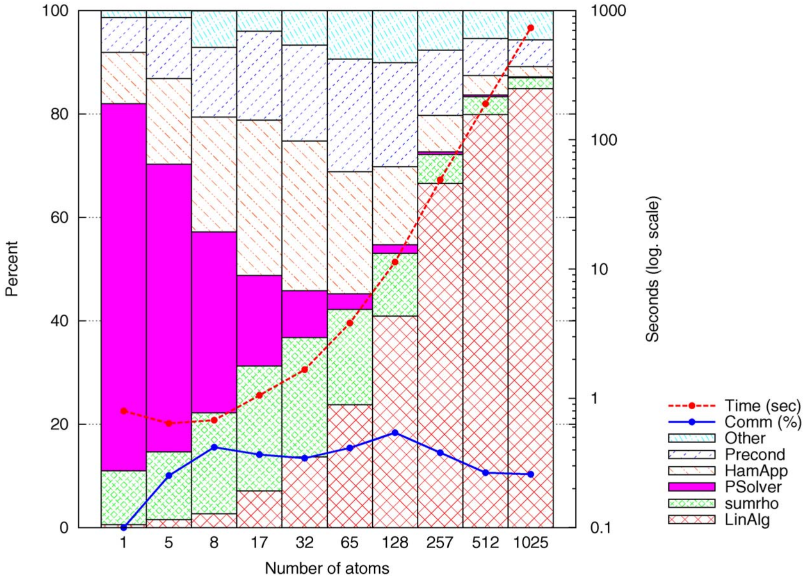

FIG. 11. (Color online) Relative importance of different code sections as a function of the number of atoms of a simple alkane chain, starting from single carbon atom. The calculation is performed in parallel such that each processor holds the same number of orbitals (two in this figure). Also the time in seconds for a single minimization iteration is indicated, showing the asymptotic cubic scaling of present implementation.

function of the number of orbitals while keeping constant the number of orbitals per processor. The different sections considered are the application of the Hamiltonian (kinetic, local plus nonlocal potential), the construction of the density [Eq. (15)], the Poisson solver for creating the Hartree potential, the preconditioning DIIS, and the operations needed for the orthogonality constraint as well as the orthogonalization, which are mainly matrix-matrix products or matrix decompositions. These operations are all performed by linear algebra subroutines provided by the LAPACK libraries.[36] Also, the percentage of the communication time is shown. While for relatively small systems the most time-dominating part of the code is related to the Poisson solver, for large systems the most expensive section is by far the calculation of the linear algebra operations. The operations performed in this section scale cubically with respect to the number of atoms. Apart from the Cholesky factorization, which has a scaling of $\mathcal{O}(n_{\text{orb}}^3)$, where n_{orb} is the number of orbitals, the cubic terms are of the form

$$\mathcal{O}(n \cdot n_{\text{orb}}^2), \tag{43}$$

where n is the number of degrees of freedom, i.e., the number of scaling function and wavelet expansion coefficients. Both the calculation of the overlap matrix in Eq. (28) and the orthogonality transformation of the orbitals in Eq. (32) lead to this scaling. The number of the coefficients n is typically much larger than the number of orbitals.

XVII. CONCLUSIONS

In this paper we have shown the principal features of an electronic structure pseudopotential method based on wavelets. Their properties make this basis set a powerful and promising tool for electronic structure calculations. The matrix elements, the kinetic energy, and nonlocal pseudopotential operators can be calculated analytically in this basis. The other operations are mainly based on convolutions with short-range filters, which can be highly optimized in order to obtain good computational performances. Our code shows high systematic convergence properties, very good performances, and an excellent efficiency for parallel calculations. This code is integrated in the ABINIT software package and is freely available under GNU-GPL license. At present, several developments are in progress to improve the features of this code. Mainly, they concern the extension of this formalism to fully periodic systems and surfaces, as well as the inclusion of noncollinear spin-polarized XC functionals. A linear scaling version of this wavelet code is also under preparation and will be presented in a forthcoming paper.

ACKNOWLEDGMENTS

We acknowledge support from the European Commission within the Sixth Framework Program through NEST-BigDFT (Contract No. BigDFT-511815), the French ANR Project LN3M (Project No. ANR-05-CIGC-003), and the Swiss National Science Foundation. Computer calculations were also performed at the Centre de Calcul Recherche et Technologie (CCRT) at CEA-Bruyères le Châtel, France, and at the Swiss national Scientific Computing Center (CSCS) in Manno.

[1] J. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).
[2] M. Dion, H. Rydberg, E. Schr, D. C. Langreth, and B. I. Lundqvist, Phys. Rev. Lett. **92**, 246401 (2004).
[3] I. Daubechies, *Ten Lectures on Wavelets* (SIAM, Philadelphia, 1992).
[4] X. Gonze, J.-M. Beuken, R. Caracas, F. Detraux, M. Fuchs, G.-M. Rignanese, L. Sindic, M. Verstraete, G. Zerah, F. Jollet, M. Torrent, A. Roy, M. Mikami, Ph. Ghosez, J.-Y. Raty, and D. C. Allan, Comput. Mater. Sci. **25**, 478 (2002); see website: http://www.abinit.org
[5] See http://inac.cea.fr/sp2m/L_Sim/BigDFT project, see http://www.unibas.ch/comphys/comphys/SOFTWARE
[6] T. L. Beck, Rev. Mod. Phys. **72**, 1041 (2000).
[7] J. E. Pask, B. M. Klein, C. Y. Fong, and P. A. Sterne, Phys. Rev. B **59**, 12352 (1999).
[8] J. J. Mortensen, L. B. Hansen, and K. W. Jacobsen, Phys. Rev. B **71**, 035109 (2005).
[9] J. R. Chelikowsky, N. Troullier, and Y. Saad, Phys. Rev. Lett. **72**, 1240 (1994).
[10] S. Goedecker, Rev. Mod. Phys. **71**, 1085 (1999).
[11] T. A. Arias, Rev. Mod. Phys. **71**, 267 (1999).
[12] T. Yanai, G. I. Fann, Z. Gan, R. J. Harrison, and G. Beylkin, J. Chem. Phys. **121**, 6680 (2004).

[13] S. Goedecker, *Wavelets and their Application for the Solution of Partial Differential Equations* (Polytechniques Universitaires Romandes, Lausanne, 1998).
[14] G. Beylkin, R. Coifman, and V. Rokhlin, Commun. Pure Appl. Math. **44**, 141 (1991).
[15] S. Goedecker, M. Teter, and J. Hutter, Phys. Rev. B **54**, 1703 (1996).
[16] C. Hartwigsen, S. Goedecker, and J. Hutter, Phys. Rev. B **58**, 3641 (1998).
[17] M. Krack, Theor. Chem. Acc. **114**, 145 (2005).
[18] M. Payne, M. Teter, D. Allan, T. Arias, and J. Joannopoulos, Rev. Mod. Phys. **64**, 1045 (1992).
[19] G. Beylkin, SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal. **29**, 1716 (1992).
[20] J. Strang and G. J. Fix, *An analysis of the Finite Element Method* (Cambridge, Wellesley, 1988).
[21] C. J. Tymczak and X.-Q. Wang, Phys. Rev. Lett. **78**, 3654 (1997).
[22] A. I. Neelov and S. Goedecker J. Comput. Phys. **217**, 312 (2006).
[23] K.-S. Lau and Q. Sun, Proc. Am. Math. Soc. **128**, 1087 (2000).
[24] L. Genovese, T. Deutsch, A. Neelov, S. Goedecker, and G. Beylkin, J. Chem. Phys. **125**, 074105 (2006).
[25] L. Genovese, T. Deutsch, and S. Goedecker, J. Chem. Phys. **127**, 054704 (2007).
[26] J. A. White and D. M. Bird, Phys. Rev. B **50**, 4954 (1994).
[27] B. R. Johnson, J. P. Modisette, P. J. Nordlander, and J. L. Kinsey, J. Chem. Phys. **110**, 8309 (1999).
[28] P. Pulay, Chem. Phys. Lett. **73**, 393 (1980).
[29] J. Hutter, H. P. Lüthi, and M. Parrinello, Comput. Mater. Sci. **2**, 244 (1994).
[30] P. Pulay, in *Modern Theoretical Chemistry*, edited by H. F. Schaefer (Plenum, New York, 1977).
[31] See NIST website: http://physics.nist.gov/PhysRefData/DFTdata/Tables/ptable.html
[32] M. M. Morrell, R. G. Parr, and M. Levy, J. Chem. Phys. **62**, 549 (1975).
[33] S. Goedecker and A. Hoisie, *Performance Optimization of Numerically Intensive Codes* (SIAM, Philadelphia, 2001).
[34] E. R. Davidson, J. Comput. Phys. **17**, 87 (1975).
[35] J. Hutter, A. Alavi, T. Deutsch, M. Bernasconi, S. Goedecker, D. Marx, M. Tuckerman, and M. Parrinello, CPMD, Version 3.8, Max-Planck-Institut für Festkörperforschung and IBM Zürich Research Laboratory, 1995–1999.
[36] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide* (SIAM, Philadelphia, 1999).