Flexibilities of wavelets as a computational basis set for large-scale electronic structure calculations

Laura E. Ratcliff, [ID] William Dawson, [ID] Giuseppe Fisicaro, Damien Caliste, [ID] Stephan Mohr, Augustin Degomme, Brice Videau, Viviana Cristiglio, [ID] Martina Stella, Marco D'Alessandro, Stefan Goedecker, [ID] Takahito Nakajima, [ID] Thierry Deutsch, and [ID] Luigi Genovese

COLLECTIONS

Paper published as part of the special topic on Electronic Structure SoftwareESS2020

View Online   Export Citation   CrossMark

ARTICLES YOU MAY BE INTERESTED IN

CP2K: An electronic structure and molecular dynamics software package - Quickstep:
Efficient and accurate electronic structure calculations
The Journal of Chemical Physics 152, 194103 (2020); https://doi.org/10.1063/5.0007045

Quantum ESPRESSO toward the exascale
The Journal of Chemical Physics 152, 154105 (2020); https://doi.org/10.1063/5.0005082

The ONETEP linear-scaling density functional theory program
The Journal of Chemical Physics 152, 174111 (2020); https://doi.org/10.1063/5.0004445

Flexibilities of wavelets as a computational basis set for large-scale electronic structure calculations

View Online   Export Citation   CrossMark

Laura E. Ratcliff,[1] William Dawson,[2] (iD) Giuseppe Fisicaro,[3] (iD) Damien Caliste,[4] Stephan Mohr,[5,6] (iD) Augustin Degomme,[4] Brice Videau,[4] Viviana Cristiglio,[7] Martina Stella,[1] (iD) Marco D'Alessandro,[8] Stefan Goedecker,[9] Takahito Nakajima,[2] (iD) Thierry Deutsch,[4] (iD) and Luigi Genovese[4,a] (iD)

AFFILIATIONS

[1] Department of Materials, Imperial College London, London SW7 2AZ, United Kingdom
[2] RIKEN Center for Computational Science, Kobe, Japan
[3] Consiglio Nazionale delle Ricerche, Istituto per la Microelettronica e Microsistemi (CNR-IMM), Z.I. VIII Strada 5, I-95121 Catania, Italy
[4] Univ. Grenoble Alpes, CEA, IRIG-MEM-L_Sim, 38000 Grenoble, France
[5] Barcelona Supercomputing Center (BSC), Barcelona, Spain
[6] Nextmol (Bytelab Solutions SL), Barcelona, Spain
[7] Institut Laue Langevin, 38042 Grenoble, France
[8] Istituto di Struttura della Materia-CNR (ISM-CNR), Via del Fosso del Cavaliere 100, 00133 Roma, Italy
[9] Department of Physics, University of Basel, Basel, Switzerland

Note: This article is part of the JCP Special Topic on Electronic Structure Software.
[a] Author to whom correspondence should be addressed: luigi.genovese@cea.fr

ABSTRACT

The BigDFT project was started in 2005 with the aim of testing the advantages of using a Daubechies wavelet basis set for Kohn–Sham (KS) density functional theory (DFT) with pseudopotentials. This project led to the creation of the BigDFT code, which employs a computational approach with optimal features of flexibility, performance, and precision of the results. In particular, the employed formalism has enabled the implementation of an algorithm able to tackle DFT calculations of large systems, up to many thousands of atoms, with a computational effort that scales linearly with the number of atoms. In this work, we recall some of the features that have been made possible by the peculiar properties of Daubechies wavelets. In particular, we focus our attention on the usage of DFT for large-scale systems. We show how the localized description of the KS problem, emerging from the features of the basis set, is helpful in providing a simplified description of large-scale electronic structure calculations. We provide some examples on how such a simplified description can be employed, and we consider, among the case-studies, the SARS-CoV-2 main protease.

Published under license by AIP Publishing. https://doi.org/10.1063/5.0004792

I. INTRODUCTION

Since their foundation, disciplines such as computational physics and quantum chemistry have had to deal with the question of the computational reliability of results. The reliability of a given approach can be defined in terms of two key concepts, namely, "accuracy," i.e., the ability of the model to predict quantities that can be externally verified, e.g., through experiment, and "precision," i.e., the ability of the employed numerical approach to find the solution to a given physical model. A precise approach should therefore reduce the computational uncertainties of quantities extracted from a well-defined model and provide reference results that can be

compared to other computer codes employing the same model. For theoretical approaches wherein no analytic solution exists, reducing the computational uncertainty is the only way to shed light on the predictive power of the model. The accuracy of a result with respect to experimental data may therefore only be reliably quantified if the computational uncertainty is significantly lower than the observed discrepancy.

Density functional theory (DFT)[1,2] has had widespread success for simulating a range of materials, from molecules to solids, and has therefore become the most popular approach to electronic structure simulations. While the accuracy of DFT is dominated by the approximations made to the exchange–correlation (XC) functional, the precision of a given simulation depends on a number of factors, in particular the choice of basis set. Thus, two different DFT codes might use the same physical formalism (including the same XC functional) but differ in results due to the use of different numerical approaches. In order to compare the results across DFT codes, careful attention must therefore be paid to the precision of the results, as seen, for example, in the DeltaCode project in which a systematic comparison of a number of periodic DFT codes was undertaken.[3]

In this context, an important distinction should be made between codes that use systematic and non-systematic basis sets. A systematic basis set allows one to calculate the exact solution of the Kohn–Sham (KS) equations with arbitrarily high precision by increasing the number of basis functions. 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. As such, it is highly desirable to have at hand a computational formalism, which is able to provide at the same time:

- a set of reliable results, which can be systematically improved by the end-user, in view of increasing—when needed—the *precision* of the calculations;
- a flexible approach in which the desired models can be explicitly implemented without having to deal with correction terms and intrinsic approximations;
- an efficient computer program, which enables the optimal use of computational resources, especially in the context of high performance computing; and
- the ability to connect together different levels of theory, where various approaches might be linked within a given computational setup.

In 2005, the EU FP6-STREP-NEST BigDFT project funded a consortium of four European laboratories (L_Sim, CEA-Grenoble, France; Basel University, Switzerland; Louvain-la-Neuve University, Belgium; and Kiel University, Germany), with the aim of developing a novel approach for DFT calculations based on Daubechies wavelets.[4] Beyond building a DFT code from scratch, the objective of this three-year project was to test the potential benefit of a new formalism in the context of electronic structure calculations.

This project was motivated by the fact that Daubechies wavelets exhibit a set of properties that make them ideal for a precise and optimized DFT approach. In particular, their systematicity provides a reliable basis set for high-precision results, whereas their locality (both in real and reciprocal space) is highly desirable to improve the efficiency and the flexibility of the treatment. Indeed, a

localized basis set allows the optimization of the number of degrees of freedom for a required accuracy, which is highly desirable, given the complexity and inhomogeneity of systems under investigation nowadays. Moreover, an approach based on localized functions makes possible to explicitly control the nature of the boundaries of the simulation domain, allowing complex environments such as mixed boundary conditions (BCs) and/or systems with a net charge.

We organize this contribution as follows: we first present some basic illustrations of the properties of Daubechies wavelets and their peculiarities in the context of computational discretization of three-dimensional KS operators. To this aim, we present the KS formalism in BigDFT, briefly outlining the main features of the Poisson solver implemented in the code. We will then explain in more detail how the solution of the KS problem is implemented in the code and how the properties of wavelets enable the realization of a computational algorithm whose time-to-solution is linearly scaling with the number of atoms in the system. We will also explain how this formalism is useful in the context of the traditional cubic scaling (CS) KS approach, which is also available in the same computer program. Then, we will outline a few examples of how the capabilities of treating systems of many thousands of atoms in the DFT formalism enable novel investigation paradigms, with the ability to—at the same time—reduce the complexity of the computational description and gain insights into the interactions between system's constituents. The BigDFT software package also presents an innovative software approach for releasing and distributing the code. We will illustrate this approach together with some comments and examples of how the code features optimal capabilities for massively parallel supercomputers. We will conclude with some perspectives on ongoing work.

II. WAVELETS AS A COMPUTATIONAL BASIS SET

Wavelet basis sets have rarely been used for electronic structure calculations, with most efforts having been devoted to their use in all-electron calculations, e.g., in MRChem[5] and most applications of MADNESS.[6] Since such a basis is therefore rather uncommon, we explain here its use in the context of KS-DFT calculations. While referring the reader to Ref. 7 for an exhaustive presentation of how wavelet basis sets can be used for numerical simulations, we here summarize the main properties of Daubechies wavelets with a special focus on the representation of the objects (wavefunctions and operators) involved in the KS-DFT formalism. We will start by illustrating the principles of one-dimensional Daubechies wavelet basis.

A. Daubechies wavelets

Every wavelet family comprises a *scaling function* ϕ, and a second function ψ, which is properly called a *wavelet*. Figure 1 illustrates the least asymmetric Daubechies wavelet family of order $2m = 16$, the basis set which is used in the BigDFT code. These functions feature a compact support $[1 - m, m]$ and are smooth and therefore also localized in Fourier space. The use of Daubechies wavelet families is guided by different criteria. Daubechies wavelets represent the best compromise between compact support, smoothness, and orthogonality for a wavelet family. We chose the family of
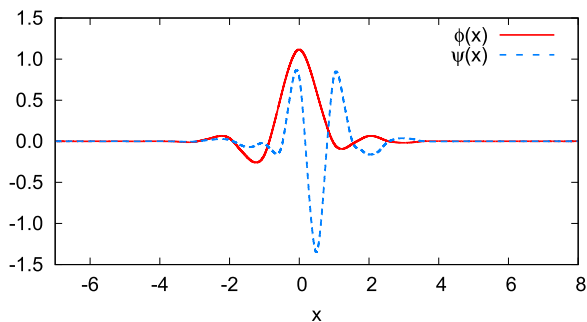
FIG. 1. Least asymmetric Daubechies wavelet family of order $2m = 16$. Note that both the scaling function $\phi(x)$ and the wavelet $\psi(x)$ are different from zero only within the interval $[1 − m, m]$.

the order 16 as it is the most compact one, which has a degree of continuity of at least 2. Nonetheless, such a family exhibits polynomial exactness of degree 8, which means that it is able to represent exactly the Taylor expansion of a Kohn–Sham orbital up to the eighth order. Such an observation, combined with the Magic Filter method (see Ref. [8]), enables an accurate and efficient approach for the evaluation of the kinetic and potential energy in KS-DFT calculations.

A basis set is simply generated by the integer translates of the scaling and wavelet functions, with arguments measured in units of the grid spacing h. For instance, a 1D domain of extension L, centered at $x = 0$, can be spanned by the following set of N scaling functions:

$$\langle x|\phi_i \rangle \equiv \phi_i(x) = \frac{1}{\sqrt{h}} \phi\left(\frac{x}{h} - i\right), \ i = -N/2, \ldots, N/2, \quad (1)$$

where $h = L/(N − 1)$ is the (uniform) grid spacing. The basis set can be completed by the addition of the translates of the wavelet functions ψ_i. These functions form an orthogonal basis set,

$$\langle \phi_i | \phi_j \rangle = \delta_{ij} = \langle \psi_i | \psi_j \rangle, \quad \langle \phi_i | \psi_j \rangle = 0. \quad (2)$$

The most important feature of any wavelet basis set is related to the concept of *multiresolution*. Such a feature builds upon the following *scaling equations* (or "refinement relations"):

$$\phi(x) = \sqrt{2} \sum_j h_j \phi(2x - j), \quad \psi(x) = \sqrt{2} \sum_j g_j \phi(2x - j), \quad (3)$$

which relate the wavelet representation at a given resolution to that at twice the resolution, and so on. According to the standard nomenclature, the sets of the h_j and $g_j = (−1)^j h_{−j}$ coefficients are called *low-* and *high-pass* filters, respectively. A wavelet family is therefore completely defined by its low-pass filter. In the case of Daubechies-$2m$ wavelets, $j \in [1 − m, m]$.

The representation $f(x)$ of a function in the above defined basis set is given by

$$f(x) = \sum_{i=-N/2}^{N/2} c_i \phi_i(x) + \sum_{i=-N/2}^{N/2} d_i \psi_i(x), \quad (4)$$

where the expansion coefficients are formally given by $c_i \equiv \langle \phi_i | f \rangle$, $d_i \equiv \langle \psi_i | f \rangle$. Using the refinement equations (3), one can map the

basis appearing in Eq. (4) to an equivalent one including only scaling functions on a finer grid of spacing $h/2$.

B. One-dimensional operators with Daubechies wavelets

The multiresolution property plays a fundamental role in the wavelet representation of differential operators. For example, it can be shown that the *exact* matrix elements of the kinetic operator can be written in the form of a circular matrix, namely,

$$T_{ij} = T_{i-j} \equiv -\frac{1}{2} \int dx \, \phi_i(x) \partial^2 \phi_j(x) \quad (5)$$

and are equal to the entries of an eigenvector of a matrix, which solely depends on the low-pass filter (see, e.g., Ref. [7]).

Daubechies-$2m$ wavelets exhibit m vanishing moments; thus, any polynomial of degree less than m can be represented exactly by an expansion over the sole scaling functions of order m. For higher order polynomials, the error is $\mathcal{O}(h^m)$, i.e., vanishingly small as soon as the grid is sufficiently fine. Hence, the difference between the representation of Eq. (4) and the exact function f decreases as h^m. The discretization error due to Daubechies-$2m$ wavelets is therefore controlled by the grid spacing. Among all the orthogonal wavelet families, Daubechies wavelets feature the minimum support length for a given number of vanishing moments.

Given a potential V known numerically on the points $\{x_k\}$ of a uniform grid, it is possible to identify an effective approximation for the potential matrix elements $V_{ij} \equiv \langle \phi_j | V | \phi_i \rangle$. It has been shown[8,9] that a quadrature filter $\{\omega_k\}$ can be defined such that the matrix elements given by

$$V_{ij} \equiv \langle \phi_j | V | \phi_i \rangle = \sum_k \omega_{k-i} V(x_k) \omega_{k-j} \quad (6)$$

yield excellent accuracy with the optimal convergence rate $\mathcal{O}(h^{2m})$ for the potential energy. The same quadrature filter can be used to express the grid point values of a (wave)function, given its expansion coefficients in terms of scaling functions,

$$f(x_k) = \sum_i c_i \omega_{k-i} + \mathcal{O}(h^m), \quad (7)$$

$$c_i = \sum_k f(x_k) \omega_{k-i} + \mathcal{O}(h^m). \quad (8)$$

As a result, the potential energy can equivalently be computed either in real space or in wavelet space, i.e., $\langle f|V|f \rangle = \sum_k f(x_k) V(x_k) f(x_k) \equiv \sum_{ij} c_i V_{ij} c_j$. The quadrature filter elements can therefore be considered as the most reliable transformation between grid point values $f(x_k)$ and scaling function coefficients c_i, as they provide exact results for polynomials of order up to $m − 1$ and do not alter the convergence properties of the basis set discretization. The filter $\{\omega_k\}$ is of length $2m$ and is defined unambiguously by the moments of the scaling functions (which, in turn, depend only on the low-pass filter).[7]

Using the above formulas, the (so far one-dimensional) Hamiltonian matrix $H_{ij} = T_{ij} + V_{ij}$ can be constructed. Note that, in contrast to other discretization schemes (finite differences, plane waves, etc.), in the wavelet basis set, *neither* the potential *nor* the kinetic terms have diagonal representations. Instead, $\hat{H}$ is represented by a *band matrix* of width $2m$.

C. Three-dimensional wavelet basis

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',

$$\phi_{i,j,k}(\mathbf{r}) = \phi(x/h' - i)\,\phi(y/h' - j)\,\phi(z/h' - k). \qquad (9)$$

In other words, 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 and non-orthorhombic grids.

The basis set of Eq. (9) is equivalent to a mixed basis set of scaling functions on a twice coarser grid of grid spacing $h = 2h'$,

$$\phi_{i,j,k}^{0}(\mathbf{r}) = \phi(x/h - i)\,\phi(y/h - j)\,\phi(z/h - k), \qquad (10)$$

augmented by a set of seven wavelets

$$
\begin{aligned}
\phi_{i,j,k}^{1}(\mathbf{r}) &= \psi(x/h - i)\,\phi(y/h - j)\,\phi(z/h - k), \\
\phi_{i,j,k}^{2}(\mathbf{r}) &= \phi(x/h - i)\,\psi(y/h - j)\,\phi(z/h - k), \\
\phi_{i,j,k}^{3}(\mathbf{r}) &= \psi(x/h - i)\,\psi(y/h - j)\,\phi(z/h - k), \\
\phi_{i,j,k}^{4}(\mathbf{r}) &= \phi(x/h - i)\,\phi(y/h - j)\,\psi(z/h - k), \qquad (11) \\
\phi_{i,j,k}^{5}(\mathbf{r}) &= \psi(x/h - i)\,\phi(y/h - j)\,\psi(z/h - k), \\
\phi_{i,j,k}^{6}(\mathbf{r}) &= \phi(x/h - i)\,\psi(y/h - j)\,\psi(z/h - k), \\
\phi_{i,j,k}^{7}(\mathbf{r}) &= \psi(x/h - i)\,\psi(y/h - j)\,\psi(z/h - k).
\end{aligned}
$$

This equivalence follows from the fact that, from Eq. (3), 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.

In a simulation domain, there are therefore three categories of grid points: those which are closest to the atoms ("fine region") carry one (three-dimensional) scaling function and seven (three-dim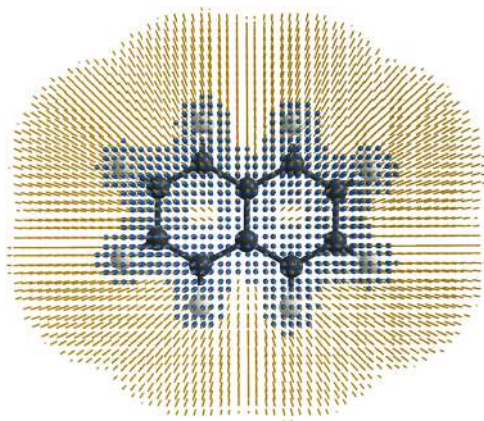ensional) wavelets; those which are further from the atoms ("coarse region") carry only one scaling function, corresponding to a resolution, which is half of that of the fine region; and those which are even further away ("empty region") carry neither scaling functions nor wavelets. To determine these regions of different resolutions, we construct two spheres around each atom a; a small one with radius $R_a^f = \lambda^f \cdot r_a^f$ and a large one with radius $R_a^c = \lambda^c \cdot r_a^c$ ($R_a^c > R_a^f$). The values of r_a^f and r_a^c are fixed for each atom type, whereas λ^f and λ^c can be specified by the user in order to control the accuracy of the calculation. The fine (coarse) region is then given by the union of all the small (large) spheres, as shown in Fig. 2. Hence, in BigDFT, the basis set is controlled by three user specified parameters; systematic convergence of the total energy is achieved by increasing the values of λ^c and λ^f while reducing the value of h.

III. KS-DFT FORMALISM WITH WAVELETS

In this section, we describe how a Daubechies wavelet basis may be used to solve the KS equations. We first introduce some notations relating to the operators that have to be discretized in Daubechies wavelets. The energy of the system in the KS formalism can be defined by

$$
\begin{aligned}
E[\lambda, \rho_c, F, \{\psi_i\}] &= -\frac{1}{2}\int d\mathbf{r}\,\mathrm{tr}\big(\nabla^2|\mathbf{r}\rangle\langle\mathbf{r}|F\big) + \mathrm{tr}\big(FV_{ext}[\lambda]\big) \\
&\quad + E_{xc}[\rho + \rho_c] + E_H[\rho] + \alpha_X E_X[F] \\
&= \sum_i f_i\langle\psi_i|H_{KS}[\rho,\rho_c,\lambda]|\psi_i\rangle - E_H[\rho] - \alpha_X E_X[F] \\
&\quad + E_{xc}[\rho + \rho_c] - \int d\mathbf{r}\rho(\mathbf{r})V_{xc}[\rho + \rho_c](\mathbf{r}), \quad (12)
\end{aligned}
$$

where we have indicated with f_i the occupation numbers associated with the KS wavefunctions ψ_i, which determine the density $\rho(\mathbf{r})$ = $F(\mathbf{r}, \mathbf{r})$, where $F(\mathbf{r}, \mathbf{r}') = \sum_i f_i \psi_i^*(\mathbf{r})\psi_i(\mathbf{r}')$ is the density matrix operator in real space. Such occupation numbers are functionally dependent on the KS energy $f_i = f(\epsilon_i)$, as by definition, the KS orbitals satisfy the eigenvalue problem of H_{KS}. The KS Hamiltonian is defined as

$$H_{KS}[\rho, \rho_c, \lambda] \equiv -\frac{1}{2}\nabla^2 + V_H[\rho] + V_{xc}[\rho + \rho_c] + \alpha_X \hat{D}_X + V_{ext}[\lambda]. \qquad (13)$$

The exact exchange term $E_X[F]$ and the associated Fock operator $\hat{D}_X$ are introduced in Sec. III C. An alternative definition of the operator H_{KS} can be given as the functional derivative of the total KS energy functional with respect to the density matrix operator,

$$H_{KS}(\mathbf{r}, \mathbf{r}') \equiv \frac{\delta E_{KS}}{\delta F(\mathbf{r}, \mathbf{r}')}. \qquad (14)$$

The external potential V_{ext}, which contains local and non-local pseudopotential (PSP) terms, depends on a set of electron-independent parameters λ that model the system under analysis, e.g., the atom positions. The core charge density $\rho_c[\lambda]$ also depends on these variables, but it is assumed to be independent of the KS orbitals. BigDFT efficiently treats *Gaussian pseudopotentials* of the Goedecker–Teter–Hutter (GTH) and Hartwigsen–Goedecker–Hutter (HGH)[10–13] types, since the intrinsic separability of both the

FIG. 2. Example simulation grid for a molecule with coarse (fine) grid points depicted in blue (gold).

basis set and Gaussian pseudopotentials allows for the simplification of several 3D operations into a sum of 1D products. The approximation of the all-electron KS quantities induced by the PSP terms has been shown to be much less severe than the exchange and correlation terms. Such PSP terms have proven to yield all-electron precision for most of the quantities of interest in ground-state DFT calculations, as can be seen in the DeltaTest initiative,[3] where an accuracy of 0.1 meV/atom—the best among the PSP calculations—can be obtained for the set of atoms belonging to the first three rows of the periodic table or in Ref. [13] where we show that the accuracy of the G2-1 and S22 test sets is comparable with all-electron calculations made by highly precise Gaussian basis sets.

A. Poisson solver

In Eq. (13), the Hartree potential $V_H[\rho]$ depends on the charge density from Poisson's equation, which in atomic units, in vacuum, reads $\nabla^2 V = -4\pi\rho$. Having efficient algorithms to solve the Poisson equation is therefore essential. The large variety of situations in which this equation can be found require us to face this problem with different choices of boundary conditions (BC) in mind. The long-range behavior of the inverse Laplacian operator makes this problem strongly dependent on the BC of the system.

In Refs. [14–16], a novel method for solving the screened and unscreened Poisson's equation in vacuum with free, fully periodic, surface-like, and wire-like BC was presented, including non-orthorhombic cells. Such a method is direct (rather than iterative) in that the solution along the isolated directions is found in its integral form using the Green's function method. For instance, in the case of a fully isolated system (or "cluster-like"),

$$V(\vec{r}) = 4\pi \int d^3\vec{r}' \, G(\mu_0; |\vec{r} - \vec{r}'|)\rho(\vec{r}'), \tag{15}$$

where $\vec{r} = (x, y, z)$. Homogeneous Dirichlet BCs ($V = 0$ at $|\vec{r}| \to \infty$) along the isolated directions are explicitly enforced by the selection of the Green's function.

The method has been in use for a few years in a number of *ab initio* codes (see the references cited in Ref. [16]) and has proven to be highly efficient and accurate in every application attempted to date. It is based on a representation of ρ and V in interpolating scaling functions (ISFs), which allows any sort of periodicity to be modeled in the most natural, clean, and mathematically rigorous way. ISFs—arising in wavelet theory[7]—enjoy several properties that make them superior to other basis sets. For instance, the representation in terms of mth order ISFs makes the first m moments of the continuous and discrete charge distributions coincide.[8] As a consequence, the representation is definitely faithful (more than just convenient), since the different moments of the charge distribution capture the major features of the potential. Moreover, ISFs are genuinely localized due to their compact support (the length of which is equal to $2m$) and endowed with the refinement relations, which easily allow for switching from a representation on a grid with spacing h to a doubly refined grid with spacing $h/2$.

The inclusion of such functionalities is motivated by the strong theoretical, experimental, and technological interest in the characterization of nanostructured materials, since solving Poisson's equation is only one of the many steps involved in state-of-the-art computer simulations and is repeated several times. Moreover, in the

context of KS-DFT and extensions thereof, there are quantities, which are computed via convolution integrals very similar to that in Eq. (15): for instance, the exact exchange term arising within those generalizations of KS-DFT employing orbital-dependent or hybrid functionals (see Ref. [17] and references therein) or the coupling-matrix in time-dependent DFT (TDDFT).[18] In this respect, the electrostatic problem of concern here provides the paradigm for many other computations, even well beyond the scope of electrostatics.

B. Soft-sphere implicit solvation model

This high-degree of flexibility makes the BigDFT Poisson solver library optimal for calculations of polarized systems or systems with non-Born-von Karman boundary conditions, such as material surfaces and isolated molecules. Nonetheless, the computational study of matter in various environments is a continuously growing field in solid state physics and chemistry. Systems of interest are, for instance, molecules, clusters, or surfaces in contact with solvents.[19] An alternative to the explicit inclusion of a wet environment is its implicit description while still treating the other parts of the system explicitly on an atomic quantum level.[20] Such an explicit/implicit treatment requires three main ingredients:

1. a dielectric cavity represented by a proper function $\epsilon(\mathbf{r})$ mimicking the surrounding solvent of a solute as a continuum dielectric;
2. a solver for the generalized Poisson equation,[21]

$$\nabla \cdot \epsilon(\mathbf{r})\nabla\phi(\mathbf{r}) = -4\pi\rho(\mathbf{r}), \tag{16}$$

where $\phi(\mathbf{r})$ is the potential generated by a given charge density $\rho(\mathbf{r})$; and

3. a model for the non-electrostatic terms to the total free energy of solvation.

The dielectric function $\epsilon(\mathbf{r})$ has to take the value of 1 where the solute is placed to solve a vacuum-like quantum problem and the bulk dielectric constant ϵ_0 outside.

The "soft-sphere" model developed by Fisicaro *et al.*[22] and implemented in BigDFT improves upon previous solvation approaches (see, e.g., Tomasi's method[23]). Model features are the following: accurate forces and a numerical cost comparable to standard vacuum calculations; feasible extensive potential energy surface (PES) explorations; a small number of model parameters; exact treatment of molecular or slab-like geometries; and the ability to treat neutral and charged molecules simultaneously in order to tackle complex interfaces (e.g., a double layer).

The interface between the quantum-mechanical solute and the surrounding environment is described by a fully continuous permittivity built up with atomic-centered "soft" spheres. This approach combines many of the advantages of the self-consistent continuum solvation model[24] in handling solutes and surfaces in contact with complex dielectric environments or electrolytes in electronic-structure calculations. In addition, it is able to describe accurately both neutral and charged systems.

We developed, tested, and implemented within the BigDFT suite a solver for the generalized Poisson [Eq. (16)] and the Poisson–Boltzmann equations to treat neutral and ionic solutions, respectively.[21] The solver for the solution of the generalized Poisson

equation and the linear regime of the Poisson–Boltzmann is based on a preconditioned conjugate gradient (CG) scheme. It allows for the iterative solution of the minimization problem with some ten iterations of the ordinary Poisson equation solver. In addition, a self-consistent procedure solves the non-linear Poisson–Boltzmann problem. Both solvers exhibit very high accuracy and parallel efficiency and allow for the treatment of free, slab, and wire-like boundary conditions.

The continuous function, describing the variation of the permittivity, allows for the analytic computation of the non-electrostatic contributions to the solvation free energy, which are described in terms of the quantum surface. The capability of treating arbitrary molecular or slab-like geometries as well as charged molecules is key to tackling electrolytes within mixed explicit/implicit frameworks. Within the soft-sphere model, two parameters are sufficient to give a mean absolute error of only 1.12 kcal/mol with respect to the experimental aqueous solvation energies for a set of 274 neutral solutes. For charged systems, the same set of parameters provides solvation energies for a set of 60 anions and 52 cations with an error of 2.96 kcal/mol and 2.13 kcal/mol, respectively, improving upon previous literature values.

The soft-sphere model has been already applied to the study of molecular doping of silicon,[25] the interface of fluorite terminations with water,[26] and the investigation of wet environment effects for ethanol and water adsorption on anatase TiO_2 (1 0 1) surfaces.[27] The latter example is presented as a case study in Sec. V A.

C. (Exact) exchange and correlation terms

The calculation of the exact exchange energy E_X requires a double summation over all the N occupied orbitals,

$$E_X[\hat{F}] = -\frac{1}{2} \sum_\sigma \int d\mathbf{r}\, d\mathbf{r}'\, \frac{F_\sigma(\mathbf{r},\mathbf{r}')\, F_\sigma(\mathbf{r}',\mathbf{r})}{|\mathbf{r}-\mathbf{r}'|}$$
$$= -\frac{1}{2} \sum_{i,j,\sigma} f_{i,\sigma} f_{j,\sigma} \int d\mathbf{r}\, d\mathbf{r}'\, \frac{\rho_{ij}^\sigma(\mathbf{r})\, \rho_{ji}^\sigma(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}, \quad (17)$$

where we have defined $\rho_{ij}^\sigma(\mathbf{r}) = \psi_{j,\sigma}^*(\mathbf{r})\, \psi_{i,\sigma}(\mathbf{r})$. The diagonal ($i = j$) contribution to E_X exactly cancels out the Hartree electrostatic energy $E_H[\rho]$. The action of the Fock operator $\hat{D}_X$ to be added to the KS Hamiltonian directly stems from the E_X definition,

$$\hat{D}_X|\psi_{i,\sigma}\rangle = \int d\mathbf{r}\, d\mathbf{r}'\, \frac{\delta E_X[\hat{F}]}{\delta F_\sigma(\mathbf{r},\mathbf{r}')} \psi_{i,\sigma}(\mathbf{r}')|\mathbf{r}\rangle$$
$$= -\sum_j \int d\mathbf{r}\, f_{j,\sigma} V_{ij}^\sigma(\mathbf{r})\psi_{j,\sigma}(\mathbf{r})|\mathbf{r}\rangle, \quad (18)$$

where we have defined

$$V_{ij}^\sigma(\mathbf{r}) = \int d\mathbf{r}'\, \frac{\rho_{ji}^\sigma(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}, \quad (19)$$

that is, the solution of Poisson's equation $\nabla^2 V_{ij}^\sigma = -4\pi\rho_{ij}^\sigma$. In a KS-DFT code that searches for the ground state orbitals, one has to repeatedly evaluate, during the SCF procedure, for a given set of $\psi_{i,\sigma}(\mathbf{r})$, the value of E_X as well as the action of the corresponding Fock operator $\hat{D}_X$ on the entire set of occupied orbitals.

D. Atomic forces

The atomic forces are, by definition, the opposite of the derivative of the total energy with respect to the atom position. In this notation, we should thus calculate

$$\frac{dE}{d\lambda} = \sum_i f_i \langle \psi_i | \frac{dV_{ext}}{d\lambda} | \psi_i \rangle + \sum_\sigma \int d\mathbf{r}\, \frac{d\rho_c^\sigma(\mathbf{r})}{d\lambda} V_{xc}^\sigma[\rho+\rho_c](\mathbf{r}). \quad (20)$$

Clearly, numerically, the set of $|\Psi_i\rangle$ is expressed in a finite basis set. This means that the action of $\mathcal{H}_{KS}$ can, in principle, lie *outside* the span of the $|\Psi_i\rangle$. We can define therefore a residual function,

$$|\chi_i\rangle = \mathcal{H}_{KS}|\Psi_i\rangle - \epsilon_i|\Psi_i\rangle, \quad (21)$$

which represents the deviation of the *numerical* KS orbital from being the *exact* KS orbital. By definition, $\langle \Psi_j | \chi_i \rangle = 0\ \forall i, j$. The norm of this vector, *once projected* on to the basis set used to express $|\Psi_i\rangle$, is often used as a convergence criterion for the ground state energy. However, even though the basis set is finite, the orthogonality of KS orbitals holds exactly. It is thus easy to show that the *numerical* atomic forces are defined as follows:

$$-\frac{dE_{BS}}{d\mathbf{R}_a} = -\sum_i f_i \langle \Psi_i | \frac{\partial H_{KS}}{\partial \mathbf{R}_a} | \Psi_i \rangle - 2\sum_i \mathrm{Re}\left(\left\langle \chi_i \left| \frac{\partial \Psi_i}{\partial \mathbf{R}_a} \right\rangle \right. \right), \quad (22)$$

where the first term of the right-hand side of the above equation is the Hellmann–Feynman contribution to the forces.

Let us now suppose that the KS Hamiltonian and orbitals are expressed in a basis set, which is complete enough to describe the orbitals *and their derivatives* within a targeted accuracy. For a Daubechies basis in the traditional BigDFT approach, this happens when the grid spacing h is such as to describe the PSP and orbital oscillations and the radii $\lambda^{c,f}$ are such as to contain the decreasing tails of the wavefunctions. In this case, the norm of $|\chi_i\rangle$ can be reduced within the *same* basis set such as to meet this targeted accuracy. Therefore, the projection of $\left| \frac{\partial \Psi_i}{\partial \mathbf{R}_a} \right\rangle$ onto the basis set used for the calculation can be safely neglected as it is associated with the same numerical precision. When the basis set is complete enough to also express $\left| \frac{\partial \Psi_i}{\partial \mathbf{R}_a} \right\rangle$, then the atomic forces can be evaluated by the Hellmann–Feynman term only, as the remaining part is proportional to the desired accuracy.

IV. IMPLEMENTATION IN BigDFT

We have presented so far the way in which the KS operators can be discretized in Daubechies wavelets. We will now present the computational approach implemented in BigDFT, starting from the chosen discretization of the KS orbitals.

A. Nearsightedness and support functions

In our approach, the KS orbitals are expressed as a linear combination of *intermediate*, possibly minimal, basis functions $|\phi_\alpha\rangle$, also referred to as support functions (SFs),

$$|\Psi_i(\mathbf{r})\rangle = \sum_\alpha c_i^\alpha |\phi_\alpha(\mathbf{r})\rangle. \quad (23)$$

In other terms, we assume that the density matrix of the system $\hat{F}$ can be defined from a set of localized SFs as follows:

$$\hat{F} \equiv \sum_i f(\epsilon_i)|\Psi_i\rangle\langle\Psi_i| = \sum_{\alpha,\beta}|\phi_\alpha\rangle K^{\alpha\beta}\langle\phi_\beta|, \qquad (24)$$

with a SF overlap matrix $S_{\alpha\beta} = \langle\phi_\alpha|\phi_\beta\rangle$, which can be chosen to have a unit diagonal and where $K^{\alpha\beta}$ is the so-called density kernel. This kernel is related to the density matrix formulation of Hernández and Gillan[28] and has to be thought of as functionally dependent on the KS Hamiltonian, namely, $\mathbf{K} = \mathbf{K}[H_{KS}]$. The density matrix $F(\mathbf{r}, \mathbf{r}')$ decays exponentially with respect to the distance $|\mathbf{r} - \mathbf{r}'|$ for systems with a finite gap or for metals at finite temperature;[29–35] for metals at zero temperature, it decays algebraically.[36] Therefore, in these cases, it can be represented by strictly localized basis functions. A natural and exact choice for these would be the maximally localized Wannier functions (MLWFs) that have the same exponential decay.[37] In our case, the localized functions are constructed *in situ* during the self-consistency cycle in terms of an underlying wavelet basis set.

A support function $\phi_\alpha(\mathbf{r})$ can thus be expanded in the wavelet basis as follows:

$$\phi(\mathbf{r}) = \sum_{i_1,i_2,i_3} s_{i_1,i_2,i_3}\varphi_{i_1,i_2,i_3}(\mathbf{r}) + \sum_{j_1,j_2,j_3}\sum_{l=1}^{7} d_{j_1,j_2,j_3}^{(\ell)}\psi_{i_1,i_2,i_3}^{(\ell)}(\mathbf{r}). \qquad (25)$$

We have here indicated with $\varphi_{j_1,j_2,j_3}(\mathbf{r}) = \varphi(x - j_1)\varphi(y - j_2)\varphi(z - j_3)$ the tensor product of three one-dimensional scaling functions, whereas $\psi_{j_1,j_2,j_3}^{(\ell)}(\mathbf{r})$ are the seven tensor products containing at least one one-dimensional wavelet. The sums over i_1, i_2, i_3 (j_1, j_2, j_3) run over all grid points where scaling functions (wavelets) are centered. These points are associated with regions of low and high resolution levels, as described in Sec. II C.

B. Localization regions

Thanks to the nearsightedness principle, it is possible to define an approach in which the computational cost is linear scaling (LS) with respect to the number of atoms, N, rather than the cubic scaling (CS), which arises when extended KS orbitals are used. Such approaches allow one to go beyond the treatment of a few hundred atoms as is typically seen with $\mathcal{O}(N^3)$ DFT approaches and instead treat systems containing several thousands of atoms. This has the benefit of also opening up the treatment of new types of materials and simulations using a pure quantum mechanical approach, as discussed, e.g., in Refs. 38 and 39. The $\mathcal{O}(N)$ formalism implemented in the BigDFT code exploits the possibility that, for systems with a suitable electronic structure, the support functions ϕ can be optimized while preserving their strict locality, namely, such as their support is within a pre-defined localization region. A similar approach is, for example, used in the ONETEP[40] and Conquest[41] codes.

For large systems where the nearsightedness principle guarantees that a local description of the orbitals is possible, the large number of degrees of freedom offered by the wavelet basis is a waste. It is therefore advantageous to build a minimal basis formed of localized (e.g., atom centered) functions. Of course, these functions will also be expanded in terms of the underlying wavelet basis, but to strictly impose locality, they will be expressed only in a subset of this global

basis set. To do so, we set to zero all scaling function and wavelet coefficients if they lie outside of a sphere with radius R_{loc} around the point $\mathbf{R}_\alpha$ on which the function is centered,

$$\begin{aligned} s_{i_1,i_2,i_3} = 0 &\Leftarrow |\mathbf{R}_{(i_1,i_2,i_3)} - \mathbf{R}_\alpha| > R_{loc}, \\ d_{j_1,j_2,j_3} = 0 &\Leftarrow |\mathbf{R}_{(j_1,j_2,j_3)} - \mathbf{R}_\alpha| > R_{loc}. \end{aligned} \qquad (26)$$

Here, $\mathbf{R}_{(i_1,i_2,i_3)}$ is the position of the grid point (i_1, i_2, i_3) and $\mathbf{R}_\alpha$ is that of the atom on which the minimal basis function $\phi_\alpha(\mathbf{r})$ is centered. These localization regions can still contain various resolution levels, as they are constructed on top of the global simulation domain. The index α is instead used in the following formulas to label kets that are associated with SFs.

In other terms, instead of working directly with the function $|\phi_\alpha\rangle$, we work with the *localized* function $\mathcal{L}^{(\alpha)}|\phi_\alpha\rangle$, where the definition of the localization projector operator in the Daubechies basis space is as described,

$$\mathcal{L}^{(\alpha)}_{i_1,i_2,i_3;j_1,j_2,j_3} = \delta_{i_1j_1}\delta_{i_2j_2}\delta_{i_3j_3}\theta(R_{loc} - |\mathbf{R}_{(i_1,i_2,i_3)} - \mathbf{R}_\alpha|), \qquad (27)$$

from which it becomes apparent that such a projection operator $\mathcal{L}^{(\alpha)}$ explicitly depends on the localization radius R_{loc} and the localization region center $\mathbf{R}_\alpha$. Clearly, if $|\phi_\alpha\rangle$ is localized around $\mathbf{R}_\alpha$ and R_{loc} is large enough, $\mathcal{L}^{(\alpha)}$ leaves $|\phi_\alpha\rangle$ unchanged and no approximation is introduced to the KS equations.

It is important to emphasize that since the Daubechies basis set is independent of $\mathbf{R}_\alpha$, $|\phi_\alpha\rangle$ depends on the center of the localization region by the introduction of the projector $\mathcal{L}^{(\alpha)}$,

$$|\phi_\alpha\rangle = \mathcal{L}^{(\alpha)}|\phi_\alpha\rangle. \qquad (28)$$

By taking the derivative of this equation with respect to $\mathbf{R}_\beta$, it is easy to find

$$\left(1 - \mathcal{L}^{(\alpha)}\right)\left|\frac{\partial\phi_\alpha}{\partial\mathbf{R}_\beta}\right\rangle = \delta_{\alpha\beta}\frac{\partial\mathcal{L}^{(\alpha)}}{\partial\mathbf{R}_\alpha}|\phi_\alpha\rangle. \qquad (29)$$

Let us now employ this result in the calculation of the atomic forces. When the KS orbitals are expressed in terms of the SFs, the non-Hellmann–Feynman term can be written as follows:

$$\mathbf{F}_a - \mathbf{F}_a^{(HF)} = -2\sum_{\alpha\beta}\mathrm{Re}\left(K^{\alpha\beta}\right)\left\langle\chi_\beta\left|\frac{\partial\phi_\alpha}{\partial\mathbf{R}_a}\right\rangle, \qquad (30)$$

where the SF residue is

$$|\chi_\alpha\rangle = H_{KS}|\phi_\alpha\rangle - \sum_{j\rho\sigma}c_j^\rho\epsilon_j c_j^\sigma S_{\sigma\alpha}|\phi_\rho\rangle. \qquad (31)$$

This result would be completely identical to Eq. (22) when no localization projectors are applied on the SF. Therefore, the only term of the forces, which cannot be captured by the localization regions, is the part that is projected outside the localization regions (but still inside the computational domain of the CS approach). The extra Pulay term due to the localization constraint is, therefore,

$$\mathbf{F}^{(P)} = -2\sum_{\alpha\beta}\mathrm{Re}\left(K^{\alpha\beta}\right)\left\langle\chi_\beta\left|(1 - \mathcal{L}^{(\alpha)})\right|\frac{\partial\phi_\alpha}{\partial\mathbf{R}_a}\right\rangle. \qquad (32)$$

From Eq. (29), we obtain

$$\mathbf{F}^{(P)} = -2 \sum_{\alpha\beta} \mathrm{Re}\left(K^{\alpha\beta}\right)\left\langle \chi_\beta \left| \frac{\partial \mathcal{L}^{(\alpha)}}{\partial \mathbf{R}_a} \right| \phi_\alpha \right\rangle, \qquad (33)$$

and from Eq. (27), we derive

$$\frac{\partial \mathcal{L}^{(\alpha)}}{\partial \mathbf{R}_\beta}\bigg|_{i_1,i_2,i_3;j_1,j_2,j_3} = \delta_{\alpha\beta}\delta_{i_1 j_1}\delta_{i_2 j_2}\delta_{i_3 j_3}\frac{\mathbf{R}_{(i_1,i_2,i_3)} - \mathbf{R}_\alpha}{R_{loc}}$$
$$\times \delta\left(R_{loc} - |\mathbf{R}_{(i_1,i_2,i_3)} - \mathbf{R}_\alpha|\right). \qquad (34)$$

Therefore, if the support functions are zero at the border of the localization region, there is no Pulay term in the atomic forces.

The Hellmann–Feynman force, given by the expression

$$\mathbf{F}_a^{HF} = -\sum_i f_i \left\langle \Psi_i \left| \frac{\partial \mathcal{H}}{\partial \mathbf{R}_a} \right| \Psi_i \right\rangle$$
$$= -\sum_{\alpha,\beta} K^{\alpha\beta}\left\langle \phi_\alpha \left| \frac{\partial \mathcal{H}}{\partial \mathbf{R}_a} \right| \phi_\beta \right\rangle, \qquad (35)$$

involves only the functional derivative of the Hamiltonian operator, which is independent of the localization regions. The CS and the LS implementations of the atomic forces are therefore identical.

C. Overview of linear scaling algorithm

The BigDFT code may therefore express the solution of the KS problem in two ways. The traditional approach, which has a computational overhead that scales cubically with the number of atoms in the system and, therefore, called the cubic scaling algorithm, expresses directly the KS orbitals Ψ_i in the wavelet basis. In this case, only the KS optimization loop is needed, and no localization projection operator $\mathcal{L}_\alpha$ is considered. The orbitals are directly labeled by their index i.

The linear scaling approach in BigDFT instead consists of two optimization loops, as depicted by the flowchart in Fig. 3. The SF and kernel optimization loops are independent of each other, with the number of iterations, convergence criteria, etc., specified independently.

Although additional approximations are introduced in the LS approach compared to the extended KS orbitals used in the CS approach, excellent agreement between total energies and forces calculated with the LS and CS approaches has been demonstrated for a range of materials and system sizes.[42,43] A number of examples are given in Table III. Furthermore, systematic convergence remains possible—as the value of R_{loc} is increased, both the total energy and forces converge toward the CS result. For example, for a fullerene molecule, it was shown that for SF radii of 7.4 Å, the total energies agree to within 0.1 meV, while the forces show better than 1 meV/Å agreement.[42] Thus, in addition to the wavelet basis parameters, the user should take care to ensure the localization radii (which may be varied independently for different atomic species) that are large enough for the required accuracy. In some cases, it may also be desirable to increase the number of SFs per atom, although, in the majority of cases, a minimal basis is sufficient, e.g., 4 SFs per C/N/O atom and 1 SF per H atom. Indeed, all the systems in Table III use a minimal SF basis apart from bulk Si.
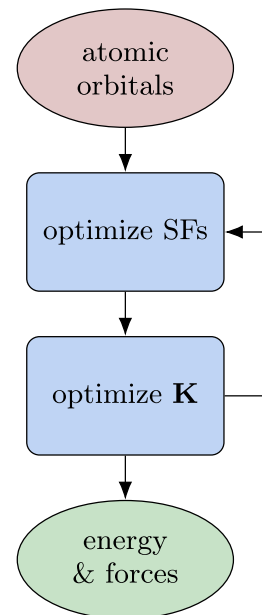

FIG. 3. Flowchart summarizing the high-level algorithm used in LS-BigDFT.

Aside from the basic SF parameters, a number of additional options are available in LS-BigDFT to allow for additional flexibility, such as whether or not to impose orthogonality on the SFs. Given the number of additional parameter choices compared to the CS approach, it is worth asking to what extent the choice of computational parameters depends on the system in question. Where a very high accuracy or optimal parallel performance is required, it is important to carefully converge with respect to parameters such as the SF radii. Nonetheless, we have demonstrated that a common set of parameters may be used to achieve consistent accuracy and robust performance across a wide range of systems.[43] Such parameters might easily be accessed by using the appropriate input profile, which have been defined to give a suitable set of parameters for a number of common use-cases.

D. Direct minimization approach

We have seen that the support functions *or* the KS orbitals are represented in a Daubechies wavelet basis and are therefore susceptible to be optimized in the LS and CS algorithm, respectively. We present in the following the main algorithm that is employed for their optimization.

The KS Hamiltonian $H_{KS}[\rho, \rho_c, \lambda]$ is the operator that defines the band-structure energy functional,

$$E_{BS}[\lambda, \rho_c, \rho, \{\psi_i\}] = \sum_i f_i \langle \psi_i | H_{KS}[\rho, \rho_c, \lambda]|\psi_i\rangle = \mathrm{tr}\left(\hat{F}\hat{H}_{KS}\right). \qquad (36)$$

Let us now impose that the wavefunctions have to be orthogonal with a certain Hermitian metric operator $\hat{\mathbb{S}}[\lambda]$ (not to be confused with the SF overlap matrix), that is, $\langle \psi_i|\hat{\mathbb{S}}|\psi_j\rangle = \delta_{ij}$. For norm-conserving PSP, frozen core, and/or all-electron calculations, S is the

identity operator, whereas it is a non-trivial quantity for projector augmented wave (PAW)[44] and ultrasoft PSPs.

Orthogonality is imposed via the following Lagrangian:

$$L[\{\psi_i\}, \Lambda] = E - \sum_{ij} \Lambda_{ij} \left(\langle \psi_i | \hat{\mathbb{S}}[\lambda] | \psi_j \rangle - \delta_{ij} \right). \quad (37)$$

As always for Lagrange multiplier techniques, minimization with respect to Λ leads to the orthogonality constraint. Only the Hermitian part of the matrix of the Lagrange multiplier coefficients Λ_{ij} should contribute due to the Hermiticity of $\hat{\mathbb{S}}$. Minimization of this Lagrangian under variation of the KS orbitals $\langle \psi_i |$ leads to the equation

$$0 = \frac{\delta L}{\delta \langle \psi_i |} = f_i | H_{KS} \psi_j \rangle - \sum_j \Lambda_{ij} \hat{\mathbb{S}}[\lambda] | \psi_j \rangle. \quad (38)$$

Multiplying Eq. (38) by $\langle \psi_k |$ leads to the condition on the Hermitian part of $\Lambda_{ik} = \frac{1}{2}(f_i + f_k)(\langle \psi_k | H_{KS} \psi_i \rangle + \langle \psi_i | H_{KS} \psi_k \rangle)$. Convergence is achieved when the average norm of the Lagrangian derivative is below a user-defined numerical tolerance.

E. Support function optimization

In principle, the optimization of KS orbitals in the basis of the support functions should minimize the total energy. This is, in fact, equivalent to minimizing the band structure energy, i.e.,

$$\min_{\phi_\alpha} \sum_{\alpha,\beta} K^{\alpha\beta} \langle \phi_\alpha | \mathcal{H}_{KS} | \phi_\beta \rangle. \quad (39)$$

The KS Hamiltonian H_{KS} does not commute with the $\mathcal{L}^{(\alpha)}$ operators. Therefore, when calculating $\mathcal{L}^{(\alpha)} | \mathcal{H}_{KS} \phi_\beta \rangle$, the localization constraint has to be relaxed before applying the Hamiltonian operator. Practically, this is done as follows. When applying the Hamiltonian, the value of the cutoff radius R_{loc} must be increased by half of the convolution filter length times the grid spacing, corresponding to a buffer region of eight grid points around the localization region. These buffers are initialized to zero, but the convolution will result in non-zero values in those regions. When the scalar product with another basis function is evaluated, it is therefore important to keep this buffer zone. Therefore, given a set of truncation radii R_{loc}, the KS Hamiltonian can be explicitly evaluated within the applied truncation scheme, preserving the variationality of the result.

As discussed, the SFs used in LS-BigDFT are strictly localized (numerical) functions that are expressed in a Daubechies wavelet basis. Starting from an atomic orbital (AO) input guess, they are optimized by minimizing the target function $\Omega = \text{tr}(\hat{F}\hat{H}_c)$ subject to the orthonormality condition of the KS orbitals,

$$\langle \Psi_i | \Psi_j \rangle = \sum_{\alpha,\beta} c_i^\alpha S_{\alpha\beta} c_j^\beta = \delta_{ij}. \quad (40)$$

The operator $\hat{H}_c = \hat{H}_{KS}[\rho] + \hat{V}_c$ is the sum of the density-dependent KS Hamiltonian plus a confining operator $\hat{V}_c$ such that

$$\langle \phi_\alpha | \hat{V}_c | \phi_\beta \rangle = \delta_{\alpha\beta} \langle \phi_\alpha | \hat{V}_c^\alpha | \phi_\alpha \rangle, \quad (41)$$

$$V_c^\alpha(\mathbf{r}) = c_\alpha | \mathbf{r} - \mathbf{R}_\alpha |^4. \quad (42)$$

We therefore have to minimize the following functional:

$$\Omega = \text{tr}(\hat{F}\hat{H}_c) - \sum_{\alpha,\beta,i,j} \Lambda_{ij} \left(c_i^\alpha c_j^\beta \langle \phi_\alpha | \phi_\beta \rangle - \delta_{ij} \right), \quad (43)$$

where the coefficients Λ_{ij} are determined by the relation

$$\sum_{i,j} c_i^\alpha c_j^\beta \Lambda_{ij} = \sum_{\rho\sigma} K^{\alpha\rho} \langle \phi_\sigma | \mathcal{H}_{KS} | \phi_\rho \rangle S_{\sigma\beta}^{-1}. \quad (44)$$

Together with this constraint, we impose the localization condition $|\phi_\alpha\rangle = \mathcal{L}^{(\alpha)} |\phi_\alpha\rangle$ on the SFs. The functional to be minimized therefore becomes

$$\Omega - \sum_\alpha \left\langle \phi_\alpha \left| 1 - \mathcal{L}^{(\alpha)} \right| \ell_\alpha \right\rangle, \quad (45)$$

where the components of the vector $|\ell_\alpha\rangle = \left(1 - \mathcal{L}^{(\alpha)}\right) |\ell_\alpha\rangle$ are the Lagrange multipliers of the constraints. The stationary condition on the functional $0 = \left| \frac{\delta\Omega}{\delta\langle \phi_\alpha |} \right|$ provides the following gradient:

$$
\begin{aligned}
|g_\alpha\rangle &= \sum_\beta K^{\alpha\beta} \mathcal{L}^{(\alpha)} \mathcal{H}_{KS} |\phi_\beta\rangle - \sum_{\beta\rho\sigma} K^{\alpha\rho} \langle \phi_\sigma | \mathcal{H}_{KS} | \phi_\rho \rangle S_{\sigma\beta}^{-1} \mathcal{L}^{(\alpha)} |\phi_\beta\rangle \\
&= \sum_{\beta\rho} K^{\alpha\rho} S_{\rho\beta}^{1/2} \left[\mathcal{L}^{(\alpha)} \mathcal{H}_{KS} |\widetilde{\phi}_\beta\rangle - \sum_\sigma \langle \widetilde{\phi}_\sigma | \mathcal{H}_{KS} | \widetilde{\phi}_\rho \rangle \mathcal{L}^{(\alpha)} |\widetilde{\phi}_\sigma\rangle \right], \quad (46)
\end{aligned}
$$

which is explicitly localized ($|g_\alpha\rangle = \mathcal{L}^{(\alpha)} |g_\alpha\rangle$). The gradient is here expressed in terms of the *orthogonalized* support functions $|\widetilde{\phi}_\alpha\rangle = S_{\alpha\beta}^{-1/2} |\phi_\beta\rangle$. The localization condition can therefore be imposed more easily by applying the constraint on (quasi-)*orthogonal* support functions, i.e., $S_{\alpha\beta} = \delta_{\alpha\beta}$. This further simplifies the evaluation of the gradient. To ensure good compromise between locality and flexibility, in general, the orthogonality is not ensured strictly for the support functions, but it is inserted in the gradient to provide a search direction, which optimizes the diagonality of the overlap matrix.

Such a minimization proceeds applying the same guidelines of the direct minimization approach of Sec. IV D by assuming unit values for the occupation numbers. The coefficient c_α is dynamically adjusted during the basis set optimization procedure. This approach has the effect of keeping the SFs confined in their localization regions, centered on the position $\mathbf{R}_\alpha$, while reducing the KS band structure energy. Usually, the position $\mathbf{R}_\alpha$ of support function α coincides with the position $\mathbf{R}_a$ of the atom a, where ϕ_α is initially centered at the beginning of the SCF optimization procedure. To some extent, this enables one to associate ϕ_α with a particular atom a.

As illustrated in Fig. 4, this procedure results in a set of SFs, which have adapted to their local chemical environments. For a molecular calculation, we therefore obtain a *minimal* set of molecular orbitals that, by construction, exactly represent the occupied KS orbitals. The SF basis also has a non-zero projection to the unoccupied orbitals subspace, although, in general, the unoccupied KS orbitals are not expected to be well represented. Although the SFs resulting from LS-BigDFT are entirely numerical and are therefore not constrained to any particular form, nonetheless, even in extended systems, they generally retain some resemblance to AOs and are thus referred to as, e.g., *s*-like SFs.
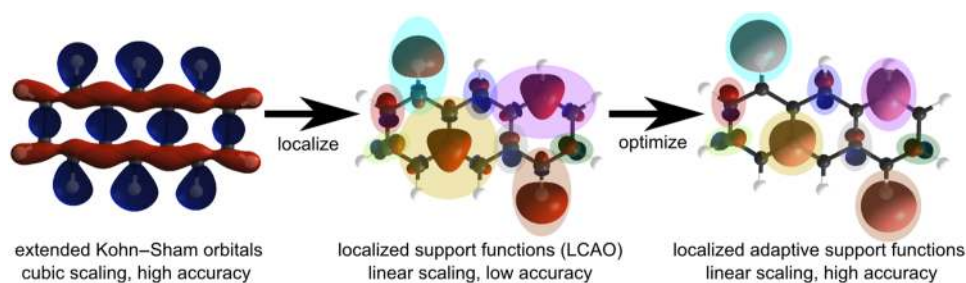
FIG. 4. Schematic illustrating different approaches in BigDFT for the example of anthracene. Shown as examples are an extended KS orbital from the CS approach (left), select unoptimized AOs used as a starting guess in the LS approach (center), and select resulting optimized SFs that constitute an accurate minimal basis in the LS approach (right). Note that the colors are used only to highlight different SFs, while the localization regions are spherical.

F. Preconditioning

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

$$|\tilde{g}_i\rangle = \mathcal{P}\left(\frac{1}{f_i}\frac{\delta L}{\delta\langle\psi_i|}\right), \tag{47}$$

where we indicate with $\mathcal{P}$ a preconditioning operator that is employed to facilitate the convergence of the wavefunction. Given the gradient direction at each step, several algorithms can be used to improve convergence. In our method, we use either a preconditioned steepest-descent algorithm or a preconditioned Direct Inversion of Iterative Subspace (DIIS) method.[45] These methods work very well to improve the convergence for non-zero 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

$$\left(-\frac{1}{2}\nabla^2 + a_\alpha(\mathbf{r} - \mathbf{R}_\alpha)^4 - \varepsilon_\alpha\right)|g_\alpha^{prec}\rangle = -|g_\alpha\rangle, \tag{48}$$

adding an extra term to account for the confining potential if present, where ε_α is an appropriate eigenvalue. The values ε_α are approximate eigenvalues obtained by a subspace diagonalization in a minimal basis of atomic pseudopotential orbitals during the generation of the input guess. For the isolated systems, the values of the ϵ_α for the occupied states are always negative; therefore, the operator of Eq. (48) is positive definite.

Equation (48) is solved by a preconditioned conjugate gradient (CG) method. The preconditioning is done by using the diagonal elements of the matrix representing the operator $\frac{1}{2}\nabla^2 - \epsilon$ in a scaling function-wavelet basis. The inclusion of the confining potential adds only a small overhead to the preconditioning equation as it can be evaluated via wavelet convolutions. The typical number of CG iterations necessary to obtain a meaningful preconditioned gradient is 5.

In a direct minimization scheme, the convergence criterion should then be based on the constrained gradient,

$$|g_i\rangle = \frac{1}{f_i}\frac{\delta L}{\delta\langle\psi_i|} = |H_{KS}\psi_j\rangle - \frac{1}{2}\sum_k\left(1 + \eta\frac{f_k}{f_i}\right)(\langle\psi_k|H_{KS}\psi_i\rangle + \langle\psi_i|H_{KS}\psi_k\rangle), \tag{49}$$

which should be zero at the stationary state. Such a definition enables one to define the direct minimization scheme also for finite temperature occupations or even KS orbitals that do not contribute to the total energy, as their converged occupation number would be 0. When both ψ_i and ψ_k correspond to occupied states, clearly $\frac{f_k}{f_i} = 1$ and we set $\eta = 1$. We assume the same limit for the case in which both states are unoccupied. We also note that, close to convergence, the Hamiltonian matrix elements are diagonally dominant, i.e., we expect that $\langle\psi_i|H_{KS}\psi_k\rangle \simeq \delta_{ik}\epsilon_i$. For this reason, in order to precondition the gradient, we set $\eta = 0$ when the value f_i is smaller than one half, such as to exclude the contribution of the occupied states from the gradient of the empty states. The value of η is set to -1 when f_i is occupied and $f_k \neq f_i$, such as to remove the off-diagonal contribution from the wavefunction update. As already explained, $\eta = 1$ in all the other cases.

G. Density kernel optimization

Given a recipe for optimizing the SFs, the question remains of how to find the density kernel for a given set of SFs. Three options are available in BigDFT: diagonalization, direct minimization, and the Fermi Operator Expansion (FOE). The first two approaches retain explicit reference to the KS wavefunctions, while FOE works directly with the density kernel. As such, FOE is the preferred approach when strict linear scaling behavior is required.

The first approach is straightforward and uses standard linear algebra routines in LAPACK, or optionally SCALAPACK, to solve the generalized eigenproblem defined by the SF Hamiltonian and overlap matrices. Although such an approach is, of course, not linear scaling, it can be useful as a benchmark approach, while the minimal size of the SF basis means that the computational cost is low compared to diagonalizing in the full wavelet basis.

The direct minimization approach works directly with the KS wavefunctions but avoids explicit diagonalization by instead minimizing the band structure energy, subject to appropriate orthogonality constraints, as described in Ref. 42. The direct minimization approach does not scale as well with respect to the system size (see Ref. 42); however, it may nonetheless be preferred to FOE in certain cases. Notably, a few unoccupied states may be straightforwardly included in both the SF and kernel optimization steps, and so direct minimization is typically used for cases where it is important to have a SF basis, which is capable of accurately representing the lowest unoccupied molecular orbital (LUMO).

Finally, in the FOE approach,[46,47] the density kernel is expressed as a function of the SF Hamiltonian matrix, i.e., $K = f(H)$, where f is the Fermi function. The Fermi function is written as an expansion of Chebyshev polynomials in such a way as to allow the K to be constructed using only matrix vector multiplications. This is combined with sparse matrix algebra, as implemented in the CheSS library,[48] resulting in LS behavior, while it can also be used to treat metallic systems at a (small) finite temperature.[49] The finite temperature can also be used to ensure robust convergence even when the gap of a system closes due to a poor initial guess, bond breaking, or when computing charged systems.

The LS behavior of BigDFT when using the FOE approach has been demonstrated for a number of materials, for systems containing up to tens of thousands of atoms.[42,43,49,50] An example is given in Fig. 17, with further discussion concerning performance considerations such as the crossover point with respect to cubic scaling BigDFT in Sec. VIII C.

H. Suitability of linear scaling approach

The SF basis of BigDFT offers numerous benefits for linear algebra based code bottlenecks. The *in situ* optimized approach allows for the accuracy of a large basis while keeping the number of basis functions similar to the size of a minimal basis, leading to small matrices even for large systems. The use of strictly localized, quasi-orthogonal basis functions further ensures that the matrices used are sparse and well-conditioned. In Table I, we report the matrix dimensions and sparsities for four different systems: a 1CRN protein[51] in the gas phase, a pentacene cluster, a 1L2Y protein[52] in solution, and a cluster of water molecules.

The benefits of this basis set are further reflected in the spectral quantities of these matrices, as shown in Table II. We see that the spectral width of the overlap matrix is quite low, reflecting how well conditioned the basis is. The ratio of the bandgap to the spectral width of the Hamiltonian is also relatively high, which leads to huge

efficiency gains for diagonalization-free methods. The lower the ratio, the more the polynomials that are required in order to approximate the Fermi function of the Hamiltonian. BigDFT's matrices require few polynomials, similar to what would be needed for minimal basis calculations with Gaussians or tight binding calculations, as shown in a head to head comparison when using density matrix purification techniques.[53]

Sparse matrices are stored in a custom segment storage format, which groups together consecutive non-zero values in a matrix row. This format not only reduces the storage overhead of a matrix but also can improve the performance of matrix-vector multiplication by using calls to dense operations. The Hamiltonian matrix is replicated across processes, and columns of the density matrix are distributed. This data distribution allows each column to be computed independently to improve parallel performance.

As mentioned in Sec. IV G, the diagonalization free method of choice in BigDFT is the Fermi operator expansion based on matrix vector multiplication. This approach is usually far more expensive than those based on recursive polynomial expansions such as density matrix purification.[54] However, the high sparsity, relatively low dimension, and good conditioning of BigDFT's matrices enable a more tailored choice of algorithm and parallelization scheme. This novel approach is made possible by the unique properties of the Daubechies wavelet basis set employed in BigDFT.

V. CUBIC SCALING APPROACH IN BigDFT

We have already introduced the CS approach of BigDFT as the explicit expression of the KS orbitals in the Daubechies wavelet basis without localization constraint. Such an approach is very useful as it provides the complete basis set limit of the (PSP) KS problem in the absence of localization constraint and with explicit boundary conditions. This approach presents optimal features of precision vs computational performance, which may prove useful in many cases. We here present as an illustration the study of the formation energy of a defect in a slab-like system with hybrid functionals and with continuum polarizable models.

A. Case study: Oxygen vacancy formation energies on the anatase TiO$_2$ (1 0 1) surface

Titanium dioxide (TiO$_2$) has gained interest in recent years for its applications in clean energy and in carbon-free production of hydrogen.[55] Various technological areas include photoassisted water splitting,[56] photocatalysis,[57] solar energy conversion,[58] and photoelectrocatalytic environmental cleanup.[59] In particular, its anatase phase has superior oxidation and photocatalytic properties. The anatase (1 0 1) surface (A$_{101}$) is the most stable and, as a consequence, the majority surface for nanocrystals. In order to better understand the photocatalytic events on anatase surfaces, it is of utmost importance to comprehend at an atomistic level the relative energetics of various surface defect configurations as well as the interactions of ambient molecules with ideal or defective surface sites.[60–64] Defects play a key role in photocatalytic processes,[65,66] whereas both shallow and deep localized trap states can be introduced by crystal defects into the wide bandgap (3.2 eV) of the anatase polymorph.[67]

TABLE I. Matrix properties of four example systems. NNZ refers to the percentage of non-zero elements in the Hamiltonian (**H**), overlap (**S**), and density kernel (**K**).

System	N_{atoms}	N_{SFs}	**H** NNZ	**S** NNZ	**K** NNZ
1CRN	642	1 623	22.09	9.40	37.20
Pentacene	6876	19 482	2.89	1.04	5.70
1L2Y	1942	4 045	5.90	2.13	11.57
Water	1719	3 438	9.43	3.43	18.30

TABLE II. Spectral properties of four example systems. The "Gap-to-width" is the ratio of the bandgap to the spectral width of the Hamiltonian.

System	S width	H width (eV)	Gap (eV)	Gap-to-width
1CRN	0.9557	47.0783	1.9977	0.0424
Pentacene	0.9852	42.2971	1.0323	0.0244
1L2Y	0.8936	47.9860	1.3682	0.0285
Water	0.4496	40.8961	7.7297	0.1890

From this side, it is important to properly model electronic states lying at or within the semiconductor bandgap. DFT calculations with the semilocal functional Perdew–Burke–Ernzerhof (PBE)[68] tend to overdelocalize electrons of strongly correlated systems that feature localized d- or f-orbitals.[69] As a consequence, the bandgap is underestimated and localized states do not fall in between the valence and conduction states. Electron localization becomes even more important in nonstoichiometric systems, such as those containing oxygen vacancies. The additional two electrons of a surface oxygen vacancy may localize on nearby Ti atoms, reducing Ti^{4+} to Ti^{3+}.[70,71] The use of hybrid functionals or the DFT + U approach has proven to be effective to correct such electron delocalization and to correctly describe the material properties.[72–74]

Oxygen vacancies are the main defects at the anatase (1 0 1) surface, playing a fundamental role in the TiO_2 surface reactivity. Among all possible vacancy configurations at surface sites, the most stable are represented by an oxygen vacancy at the outermost twofold coordinated oxygen surface site belonging to the first trilayer (named surface vacancy or V_O^{surf}) and an oxygen vacancy at the bulk lattice site belonging to the second trilayer (named subsurface vacancy or V_O^{sub}). Both oxygen vacancies are depicted in Fig. 5. In order to evaluate their relative stability, we extracted
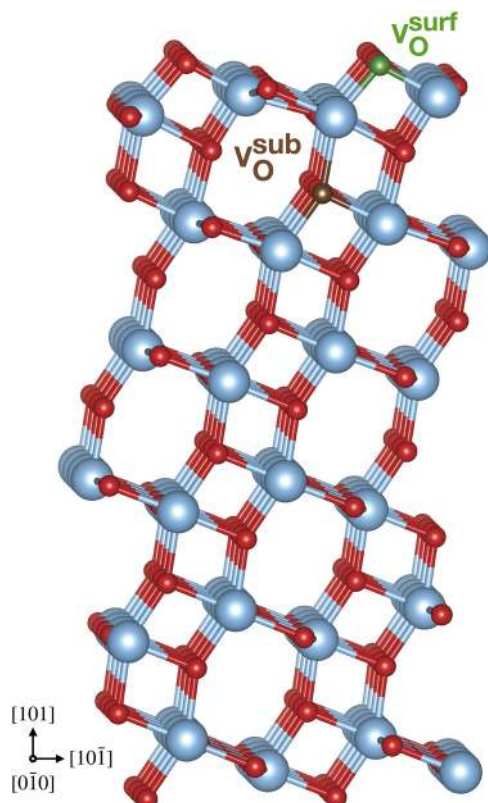

FIG. 5. Depiction of the six trilayer 4 × 1 supercells for the anatase TiO_2 (1 0 1) surface. O: red; Ti: cyan; surface O vacancy V_O^{surf}: green; and subsurface O vacancy V_O^{sub}: brown.

formation energies for the oxygen vacancy from the total energy of the defective $E_{tot}(def)$ and the stoichiometric $E_{tot}(no\ def)$ anatase A_{101} surface,

$$E_{form}(V_O) = E_{tot}(def) - E_{tot}(no\ def) + \frac{1}{2}\mu(O_2). \quad (50)$$

The chemical potential of $\mu(O_2)$ is taken to be the total energy of an isolated O_2 molecule.

The anatase TiO_2 (1 0 1) surface was modeled with six trilayers of a 4 × 1 supercell (288 atoms in total) with periodic dimensions of 15.18 Å and 10.43 Å. Figure 5 shows the whole A_{101} surface. A six trilayer slab and a 4 × 1 supercell with a fixed bottom layer have provided a fully reliable balance between computational efficiency and accuracy.[66,75,76] The 4 × 1 periodicity prevents spurious interactions with periodic images of the defects or the adsorbed organic molecule in the periodic directions x and y (z is considered orthogonal to the surface plane, i.e., along the[100] crystal direction). Surface boundary conditions, where periodic replicas for the atomistic system are imposed only on the x and y surface directions, have been set for all surface calculations. The wavelet basis functions were distributed on an adaptive uniform mesh with a resolution of $h := h_x = h_y = h_z = 0.40$ bohr for all calculations. Due to the large sizes of the system, only the Γ point has been used for the k-space integration in all geometry optimizations. We employed soft norm-conserving PSPs including non-linear core corrections[10,13] to describe the core electrons along with a PBE0[77] functional hybrid XC functional, as implemented in the LibXC[78] library. BigDFT takes advantage of the highly efficient graphics processing unit (GPU) implementation of a real-space based algorithm for the evaluation of the exact exchange, which reduces the cost of hybrid functional calculations in systematic basis sets, without any approximation, by nearly one order of magnitude.[17] To make energetic comparisons, we performed all runs with equivalent parameter settings and convergence criteria. All energetics refer to the final relaxed structures.

Formation energies are 3.17 eV for V_O^{surf}, and 2.94 eV for V_O^{sub}, with a difference of 0.23 eV. As a consequence, a subsurface vacancy is energetically favorable with respect to surface vacancies. A previous study with a hybrid screened exchange functional reports similar results, obtaining a difference in stability of 0.30 eV.[73] This result also agrees with previous DFT studies using the PBE functional where the formation energy of a subsurface vacancy on the anatase TiO_2 (1 0 1) surface was found to be 0.40 eV lower than a defect on surface sites.[27,65,66,73]

When the anatase TiO_2 (1 0 1) surface interfaces with a wet environment, the stability of vacancy defects or adsorbed species lying at the solid/liquid interface can be modified.[27] In a recent study of the interface of water and ethanol with the A_{101} surface,[27] we show that the proper inclusion of the wet environment in the methodological scheme is fundamental for obtaining reliable results. Calculations are based on structure predictions[25,26,79] at a DFT level for molecules interacting with the perfect and defective anatase (1 0 1) surfaces under both vacuum and wet conditions. The soft-sphere implicit solvation model is used to describe the polar character of the two solvents.[20–22] As a result, we find that surface oxygen vacancies become energetically favorable with respect to subsurface vacancies at the solid/liquid interface. This aspect is confirmed by *ab initio* molecular dynamics simulations with explicit water molecules. Ethanol molecules are able to strongly passivate these

vacancies, whereas water molecules only weakly interact with the (1 0 1) surface, allowing the coexistence of surface vacancy defects and adsorbed species. The infrared and photoluminescence spectra of anatase nanoparticles exposing predominantly (1 0 1) surfaces dispersed in water and ethanol support the predicted molecular–surface interactions, validating the whole computational paradigm.

VI. OPPORTUNITIES ARISING FROM LINEAR SCALING IN WAVELETS

Beyond a means of achieving linear scaling, the localized SF basis also offers other advantages, both as a tool for analysis and as a means of further reducing the complexity and thereby the computational cost of treating large systems. Specifically, the SF basis permits two separate, but related approaches:

1. the reduction in computational cost by exploiting similarity between fragments in both molecular and periodic systems and
2. the identification and exploitation of independent fragments for building Quantum Mechanics/Molecular Modelling (QM/MM) models.

Both approaches center around a fragment-like description wherein a system is divided into subsystems, which might or might not considered to be quasi-independent. Importantly, the SFs can also be used to define indicators, which enable one to assess the similarity and/or separability of fragments and thereby determine to what extent such a description is valid. In the following, we describe the two approaches and their associated indicators.

A. Fragment approach

As discussed above, the SFs facilitate a LS approach such that simulations of 1000s of atoms are possible. Nonetheless, one might ask if it is possible to also reduce the prefactor? A significant portion of the computational cost is due to the SF optimization, which cannot easily be avoided since it is essential for achieving high accuracy. However, since the SFs adapt to their chemical environment, for systems with many SFs in similar environments, one might imagine taking advantage of this similarity. To this end, we have developed a fragment-based approach, which is applicable to both molecular[80] and extended systems.[50] In both cases, the system is divided into a number of fragments, which, in the molecular case, are quasi-independent and, in the extended case, are interacting pseudo-fragments. If a system has repeated fragments, then one can reduce the computational cost.

Before further discussing the fragment approach, it is useful to consider under which circumstances SFs are similar. In other words, how should one define the (pseudo-)fragments? For weakly interacting supramolecular systems, it is reasonable to assume that the SFs will be similar between molecules. For example, the SFs associated with the O atom in one water molecule within a droplet are similar to those associated with O atoms in other molecules so that a water molecule represents a good choice of fragment. For a periodic solid, on the other hand, one might assume that SFs associated with equivalent atoms are in similar environments, e.g., all Si atoms in bulk Si so that a single Si atom might be treated as a pseudo-fragment. However, other systems are less intuitive, e.g., where there is a defect or an edge. It is therefore useful to have a

quantitative measure of SF similarity. To this end, we use the onsite overlap matrix,[50] defined as

$$S_{\alpha\beta}^{\text{onsite}} \equiv \langle \mathcal{RT}^{\mathbf{R}_\alpha \to \mathbf{R}_0} \phi_\alpha | \mathcal{RT}^{\mathbf{R}_\beta \to \mathbf{R}_0} \phi_\beta \rangle, \tag{51}$$

and in other words, the overlap between two SFs, originally centered on R_α and R_β, has been rototranslated by the matrices $\mathcal{R}$ and $\mathcal{T}$ to be on a site of center $\mathbf{R}_0$. Thus, $S_{\alpha\beta}^{\text{onsite}} = 1$ implies that ϕ_α and ϕ_β are identical, having both the same character (s, p_x, etc.) and the local chemical environment.

In Fig. 6(a), we show the optimized SFs for a pentacene molecule, which might be used to generate template SFs for longer (finite or periodic) acenes. If we consider pairs of C atoms as pseudo-fragments (with their associated H atoms), then we can define two inequivalent bulk-like pseudo-fragment types, i.e., which are unaffected by the edges. One can see that SFs near the edges differ from the equivalent bulk-like fragments, as confirmed by S^{onsite} [Fig. 6(b)]. We have previously shown that S^{onsite} is correlated with the error induced by a given pseudo-fragment setup in quantities such as the total energy for the examples of SiC nanotubes and defective graphene.[50,81] In other words, the smaller the threshold for $1 - S^{\text{onsite}}$, the smaller the error compared to using fully optimized SFs. In this system, such an approach may be used to determine which fragments may be treated as bulk-like and which correspond to distinct edge regions. A smaller threshold for $(1 - S^{\text{onsite}})_{\text{max}}$ results in larger edge regions and vice versa, as demonstrated in Fig. 6(d). Finally, the error induced by a given fragment setup may be tested by retaining correctly optimized SFs for the edge fragments and replacing the remainder with the bulk-like fragments, as shown in Fig. 6(c). As can be seen, there is indeed a correlation between the selected threshold and the induced error. Thus, if one chooses the threshold according to the desired accuracy, S^{onsite} can be used to define a satisfactory pseudo-fragment setup. For example, in this case, a threshold of 10^{-3} results in an error of 12 meV/atom. For molecular fragment calculations, an alternative indicator is more appropriate for informing the fragmentation of a system. This is presented in Sec. VI B.

Having chosen an appropriate (pseudo-)fragment setup, the fragment approach consists of three steps:

1. **Template calculation**: optimize the SFs for each template fragment T via a full LS calculation.
2. **SF replication**: replicate the SFs from the template fragments for the full system, rototranslating as needed to account for differing orientations between template and system fragments.
3. **Full calculation**: perform a LS calculation of the full system using the replicated SFs keeping the SFs fixed and only optimizing the density kernel.

The procedure is illustrated in Fig. 7 for both molecular and extended (pseudo-)fragments. Since the calculation on the full system involves the use of a fixed SF basis, significant computational savings can be made by avoiding the need for SF optimization. Steps one and three rely only on the standard machinery of LS-BigDFT. The second step can be automated, given both a means of detecting and performing the rototranslation. The appropriate rototranslation is found by defining a cost function,

$$J\left(\mathcal{R}^{T \to S}\right) = \frac{1}{2N} \sum_{a=1}^{N} \| \mathbf{R}_a^S - \sum_{b=1}^{N} \mathcal{R}_{ab}^{T \to S} \mathbf{R}_a^T \|^2, \tag{52}$$
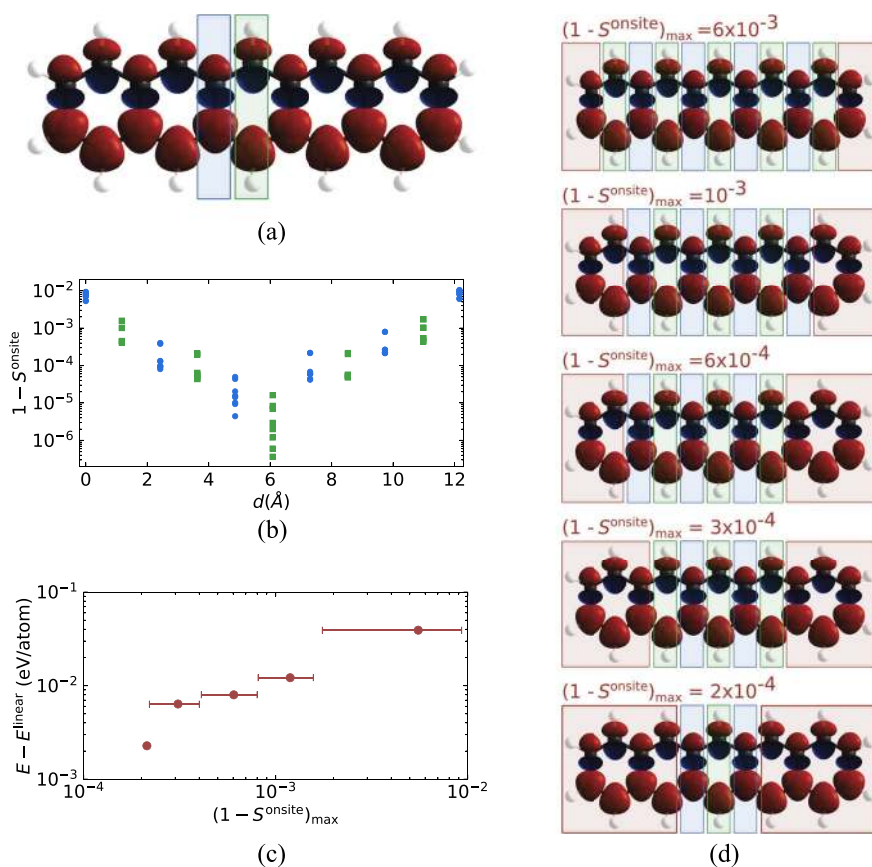
FIG. 6. Schematic illustrating the use of the onsite overlap matrix in setting up a pseudo-fragment calculation starting from a pentacene template calculation, following which the SFs could be reused for a longer acene. (a) Select optimized SFs and definition of two distinct bulk-like pseudo-fragment types (colored boxes). (b) $1 - S_{ij}^{\text{onsite}}$ vs distance d along the x-axis, where i runs over SFs associated with atoms in the highlighted fragments of (a) and j is over all SFs. Matrix elements are only shown between SFs belonging to C atoms with the same y-coordinates and between the same SF type, e.g., s-like with s-like. (c) Error in energy relative to the full linear scaling result for the fragment setups of (d), where the SFs associated with the edge-like fragments in each setup are the correct optimized SFs and the SFs associated with all other atoms are those coming from the bulk-like fragments defined in (a). The error bars denote the range of values of $(1 - S^{\text{onsite}})_{\text{max}}$, which would give the same fragment setup. (d) Different pseudo-fragment setups, where the threshold determines the extent of the edge fragments: if any values of $1 - S^{\text{onsite}}$ between a given fragment instance and the equivalent bulk-like fragment are above the threshold, $(1 - S^{\text{onsite}})_{\text{max}}$, then that fragment is considered to be distinct and therefore treated as edge-like, otherwise it is considered to be bulk-like.

where N is the number of atoms in the fragment and $\mathbf{R}_a^{T(S)}$ are the coordinates of the template (system) fragment. The optimal transformation may be found by minimizing the cost function.[82–84] The rototranslation itself is achieved via an accurate and efficient wavelet-based interpolation, described in Ref. 80.

So far, we have considered rigid fragments. If there are significant deviations between the template and system fragments' geometries, then the fragments should no longer be treated as the same, since the associated SFs are not similar. However, we often want to allow small deformations. Thankfully, the cost function J not only already takes into account deformations but also provides an indication of the similarity between fragment geometries. Like S^{onsite}, J is correlated with the error induced by a given fragment setup[50,81] and can therefore be used as an indicator of whether a given setup is appropriate. Unlike S^{onsite}, J requires only atomic information and so can be calculated without performing a LS calculation. When the

differences between fragments are dominated by the atomic structure, e.g., in the case of a defect introducing local distortions in a periodic material, J is therefore particularly useful. As an example, in Fig. 8, we show the correlation between J and the induced error for noisy SiC nanotubes, where the template SFs were generated in a pristine nanotube. As can be seen, despite being based purely on geometric information, the value of J is indeed a good indicator for the induced error since it is dominated by the random distortions in the atomic structure.

The fragment approach has thus far been applied to a range of molecular and extended systems[50,80,81,85] and, provided that an appropriate fragment setup is chosen, has been shown to give good accuracy for quantities including densities of states (DoS) [see, e.g., Fig. 17(c)], transfer integrals, and total energies. In Table III, we show the total energies for a range of systems, calculated using cubic, linear, and fragment approaches. While the induced error relative to
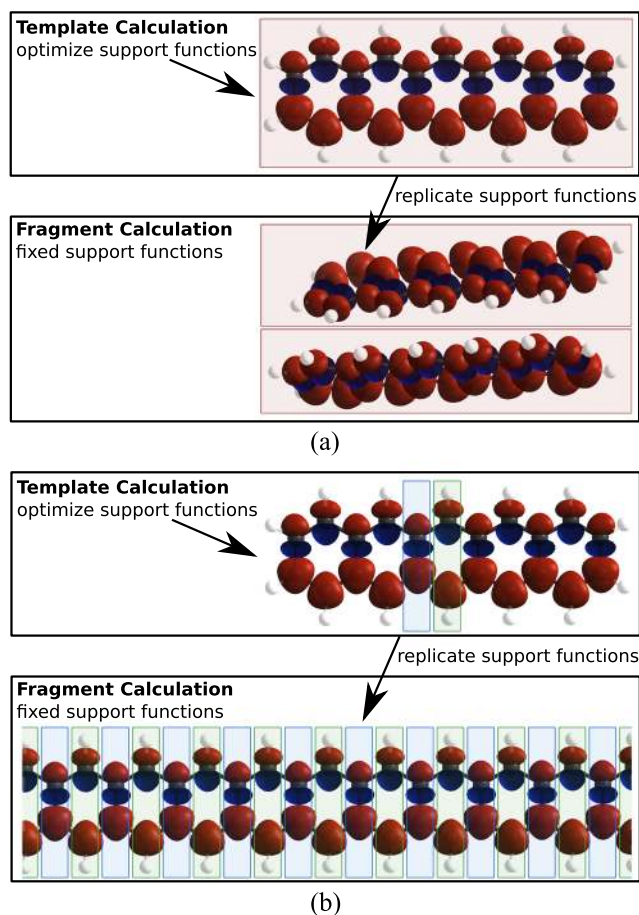
FIG. 7. Schematic illustrating the fragment approach for molecular and extended systems. The boxes indicate a (pseudo-)fragment, while the color refers to the fragment type. (a) Molecular fragment approach: a pentacene molecule used as a template for a pentacene dimer. (b) Pseudo-fragment approach: a pentacene molecule used as a template for polyacene. Since the polymer is in periodic boundary conditions, no edge pseudo-fragments are needed.
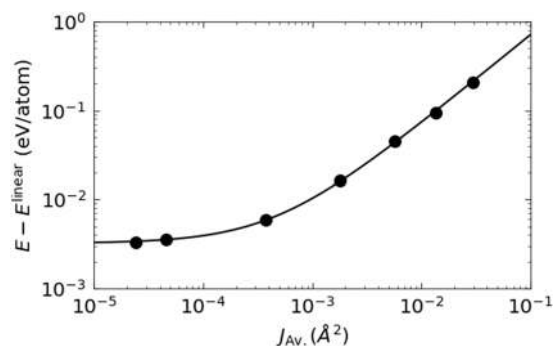

FIG. 8. Correlation between the average value of the cost function J and the induced error for pseudo-fragment calculations of noisy periodic SiC nanotubes, where the SFs from a pristine nanotube are used as a basis for nanotubes with varying amounts of random noise. The line represents a linear fit to the data. Additional details are given in Ref. 50.

fully optimized SFs depends on the system, fragment setup, etc. in the majority of cases, it is of the order of 10 meV/atom or lower. The computational cost of the fragment approach also depends strongly on the system and setup in question, however, as shown in, e.g., Figs. 15(d) and 17(a); the computational cost is typically at least five times lower than for a full linear scaling calculation, and in some cases, the savings are considerably higher. Finally, we note that the forces are more sensitive to the calculation setup than the total energies. In many cases, they are therefore not reliable when using the fragment approach, and so we have thus far focused on energies rather than forces.

B. Complexity reduction

In this paper, we have so far shown how BigDFT's linear scaling capabilities enable calculations on large systems. Nonetheless, these calculations remain computationally intense, and it is unrealistic (if not unnecessary) to expect DFT calculations to replace commonly used forcefield methods, as a full statistical sampling of a system's configuration space (free energy calculations, molecular dynamics, etc.) remains expensive. It is thus crucial for us to develop analysis tools that use the results of large scale DFT calculations to gain new kinds of insights into the emergent properties of systems.

One such tool we have developed in this spirit is a complexity reduction framework,[86,87] which takes large scale calculations and uses them to decompose systems into coarse grained fragments. This fragment generation procedure is based on two metrics:

- the purity indicator that measures the quality of a fragment and
- the fragment bond order that quantifies inter-fragment interactions.

Both of these measures can be cheaply computed directly from the single particle density matrix and can be used to automatically partition a system into fragments, design embedding environments for QM/MM type approaches, and produce graph-like views of a system.

Here, we describe two example applications of this approach. For the first, we consider the problem of describing the interaction of a molecule with its surrounding solvent environment. While BigDFT's implicit solvation model can effectively mimic solvation effects (Sec. III B) without the need for sampling, it can nonetheless be desirable to perform calculations, including explicit solvent molecules, when solvent properties differ significantly near the surface of the solute (e.g., the role of structural waters in proteins). However, including a large number of solvent molecules in a system can greatly increase the computational cost. Using BigDFT's complexity reduction framework, we can efficiently quantify solute–solvent interactions and use this information to design minimal solvation shells. Figure 9 shows an example of such a calculation using an RNA molecule in solution. When the RNA molecule is computed without the solvent, the dipole is significantly different from when computed in the solution. To define a suitable environment, we compute the fragment bond order between each solvent molecule and the RNA. We then build an environment by including all fragments until the sum of the fragment bond order of the remaining fragments falls below some cutoff. By including enough solvent molecules (values around 10^{-3}), we can

TABLE III. Comparison of total energies, E, and root mean squared forces, F, calculated using the cubic, linear, and fragment ("frag") approaches for a range of systems with different boundary conditions ("bcs"). The forces are sometimes unreliable in the fragment approach, and so only energies are given. The number of atoms, N_{atom}, is given for the full system ("sys"), the template ("temp") calculation, and an individual fragment, while the number of distinct fragment types, N_{type}, is also given. The errors in energy and forces are defined as the error introduced by the *additional* approximation, i.e., $\Delta E^{linear} = E^{linear} - E^{cubic}$ and $\Delta E^{fragment} = E^{fragment} - E^{linear}$.

| | | N_{atom} | | | | E (eV/atom) | | | ΔE (meV/atom) | | F (eV/Å) | | ΔF (meV/Å) |
	bcs	sys	temp	frag	N_{type}	cubic	linear	frag	linear	frag	cubic	linear	linear
Molecular fragments													
Pentacene dimer[a]	Free	72	36	36	1	−100.97	−100.96	−100.96	7.9	6.3	0.25	0.25	20.8
Anthracene cluster[b]	Free	216	24	24	1	−96.98	−96.96	−96.95	18.7	6.3	1.00	1.00	24.7
CBP[c]	Free	2480	62	62	1	−119.38	−119.36	−119.35	20.1	10.2	0.09	0.07	23.3
Pseudo-fragments													
Polyacene[a]	Periodic	60	36	2/4	2	−108.73	−108.72	−108.70	9.1	12.6	0.04	0.03	9.2
Bulk Si[d]	Periodic	512	216	1	1	−191.49	−191.46	−191.46	25.4	5.4	0.00	0.00	2.8
Graphene[e]	Surface	576	112	4	1	−179.13	−179.11	179.11	11.4	3.8	0.01	0.00	13.3
Nanoribbon[e]	Surface	576	108	18	1	−178.64	−178.63	−178.63	10.2	0.1	0.06	0.10	35.4
SiC nanotube[f]	Periodic	392	168	28	1	−130.80	−130.79	−130.79	9.6	3.3	0.01	0.01	9.3
			168		6			−130.67		44.0			
SiC nanotube[f]	Free	392	224	28	8	−130.73	−130.72	−130.71	9.9	7.0	0.00	0.01	12.7
			280		10			−130.71		4.8			

[a]Calculations were performed with LDA, $h = 0.4$ bohr, 4(1) SFs per C(H) atom, and $R_{loc} = 7$ bohrs with fragment setups, as depicted in Fig. 7.
[b]Calculations were performed with PBE, $h = 0.4$ bohr, 4(1) SFs per C(H) atom, and $R_{loc} = 6$ bohrs.
[c]Calculation details as described in Fig. 17.
[d]Calculation details as described for PBE calculations in Fig. 15.
[e]Computational details are given in Ref. 81.
[f]Computational details are given in Ref. 50.

automatically generate the appropriate environment to reproduce the dipole.

For our second example, we consider the problem of describing the active site of a protein. Here, we consider a Laccase enzyme, which is known to have an active site involving four copper atoms.[88] One copper sits alone and is used to oxidize substrates that dock with the protein. This electron is then transferred to a three copper ring, where it is finally used to reduce O_2 to H_2O. A coarse grained view
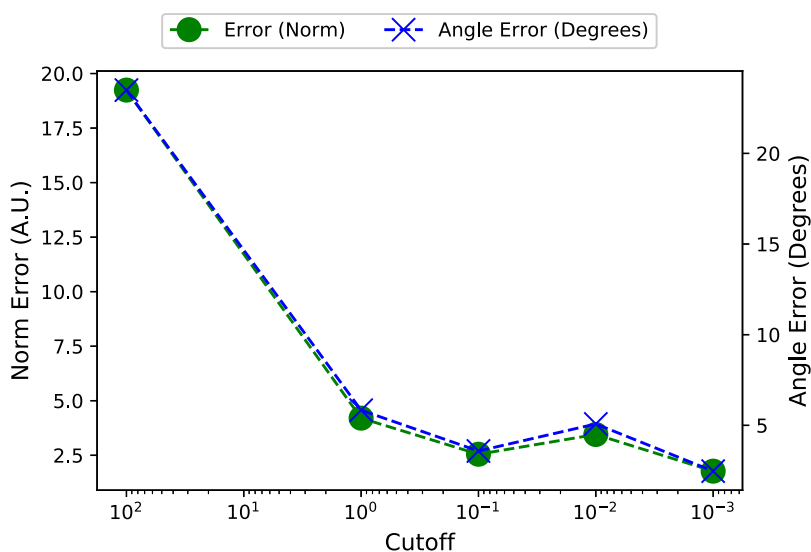

FIG. 9. The error in the dipole of an RNA molecule when computed in different solvation environments defined by the fragment bond order. The value computed using the full system was 38.8 a.u.
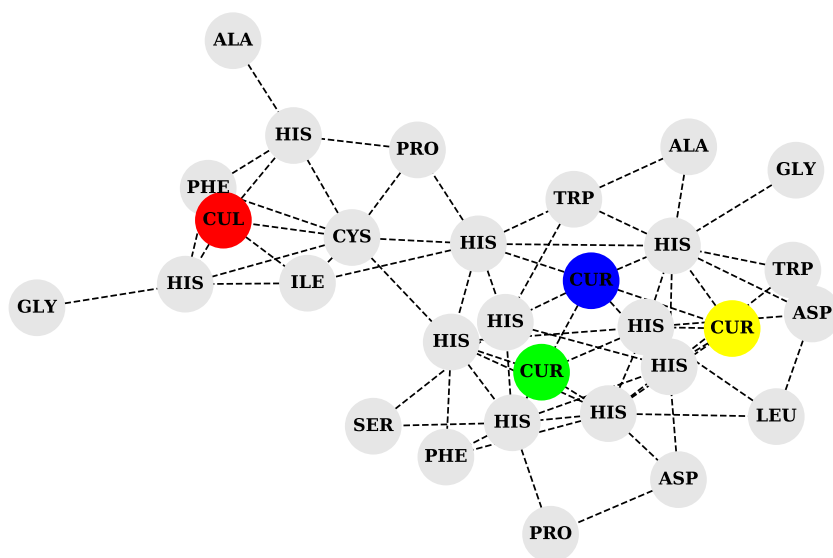
FIG. 10. A graphic view of the active site of a Laccase enzyme. Nodes are defined as the amino acids and copper atoms, and edges are drawn between fragments, which share an embedding environment, as defined by the fragment bond order measure. Nodes are labeled by their amino acid type, with the lone copper labeled CUL and the ring copper CUR.

of this electron transfer, and knowledge of the participating amino acids, might gain insight into this mechanism and how this enzyme might be tuned for applications. To generate such a view, we compute the fragment bond order between those copper atoms and each of the amino acids of the system. We then construct a graph, where amino acids and copper atoms are nodes, and edges are defined using the bond order as a guide, as drawn in Fig. 10.

VII. BigDFT FORMALISM FOR THE STUDY OF BIOLOGICAL SYSTEMS: APPLICATION TO SARS-CoV-2 MAIN CONSTITUENTS

Given the critical situation facing the world at the moment, several researchers belonging to different scientific communities and groups worldwide are working on SARS-CoV-2, in particular on the protease and spike protein. The protease is, in fact, found within the virus core along with the nucleocapsid protein and RNA. It is an essential enzyme for the life-cycle of the virus since it produces structural and functional proteins that are required for the maturation and replication of the virus. As such it is an important antiviral drug target since if its function is inhibited, the virus remains immature and non-infectious. Using fragment based screening, researchers have identified a number of small compounds that bind in the active-site of the protease, which can be used as a starting point for the development of protease inhibitors (see, e.g., Refs. 89 and 90).

Thanks to the development of the BigDFT code and its LS approach, we have the possibility to model the electronic structure of the protease in contact with a potential docked inhibitor and provide new insights into the interactions of the potential PIs with the protease by selecting specific amino acids that are involved in the interaction and characterizing their polarities. This new approach we propose is complementary to the docking methods used up to now and based on *in silico* research of the inhibitor. We have started a series of calculations, taking advantage of the Protein Data Bank

(PDB) structures available, which will be presented elsewhere. In this section, our main objective is to show a simplified demonstration of a computational approach based on the fragmentation approach described in Sec. VI B, which would be accessible to other scientific communities, such as biologists or medicinal chemists, who may be able to extract new ideas from data presented as follows.

A. Fragmentation of biological systems: The main protease of SARS-CoV-2

Biological systems are naturally composed of fragments such as amino acids in proteins or nitrogenous bases in DNA. We show in this example the SARS-CoV-2 main protease (PDB ID 6LU7) in complex with an N3 inhibitor. Such a structure is made of a dimer of two identical sub-units, each one with a docked inhibitor. We made our calculation by presenting only one monomer, made of 4732 atoms, depicted in Fig. 11. Such a biological system is made of two chains: one associated with the amino acids that belong to the enzyme and the second associated with the inhibitor.

As already stated, with our approach, we are able to evaluate whether the amino acid-based fragmentation is consistent with the QM computational setup. Such a system has already been analyzed at a QM level of theory with a fragment molecular orbital technique,[91] where the fragmentation of the system has to be imposed beforehand following chemical intuition. Here, to evaluate the reliability of the model, we have at our disposal the purity indicator that gives us, for each fragment, the level of confidence with which such a fragment can be considered as an independent unit of the system. This is an important indicator for the end-user, as it enables to evaluate the quality of the information associated with a given fragment. Usually, a cutoff of 0.05 is employed, which has proven to provide meaningful physico-chemical results in most circumstances (see Refs. 86 and 87). The representation of Fig. 11 is instead based on a re-fragmentation with a much tighter cutoff (0.025), which results in a coarse-grained view of the system.
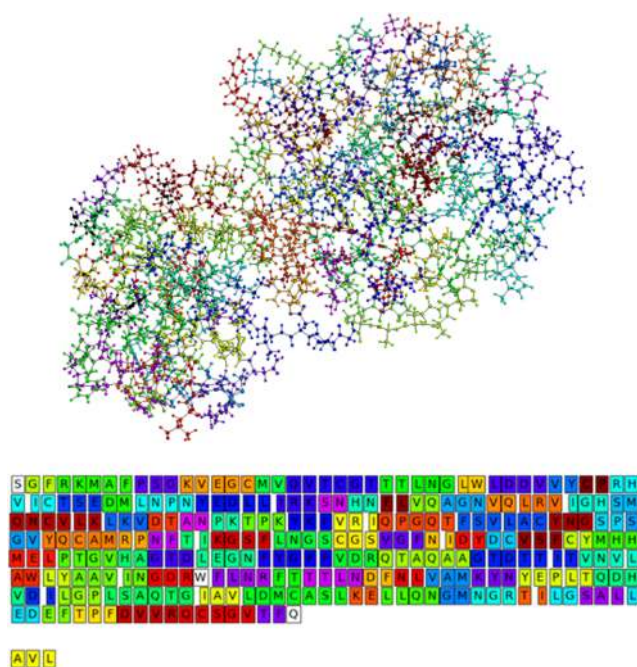
FIG. 11. The atomistic representation of the SARS-CoV-2 main protease monomer (top) employed in the present case-study, together with the amino acid sequences (bottom). Here, we see two chains: the main enzyme chain and the amino acid chain that are associated with the inhibitor (the second AVL sequence). Amino acids and respective atoms, which belong to the same QM fragment, are colored with the same color.

B. Population analysis of the system after the QM calculation: Role of implicit solvation

After a given fragmentation is identified, we know that we may decompose the system's observables in terms of the fragments. The fragment population analysis that has been introduced in Refs. 86 and 87 can then be employed. This would enable the association of the system constituent values such as the charge state and the polarity with a simple and straightforward interpretation. For instance, we may extract the charge population on each of the amino acid of the systems, as depicted in Fig. 12. Thanks to our efficient implementation of continuum solvents, we are able to perform the same calculation in the electrostatic cavity to understand the role that the (implicit) solvent may play in the protonation states of the system's residues. Results are shown in Fig. 12, where we see that the amino acid charge is actually influenced by the presence of the employed environment.

C. Evaluating the section of the system that interacts with the inhibitor

We have seen in the Laccase example how a coarse grained view of the system is important. We have now presented QM observables on the system's fragments, which are based on a population analysis of the electronic density of the system, projected on the amino acid. A novel quantity that our approach enables to address is the possibility of quantifying the strength of the chemical interaction between the different fragments. It is possible to select a target region
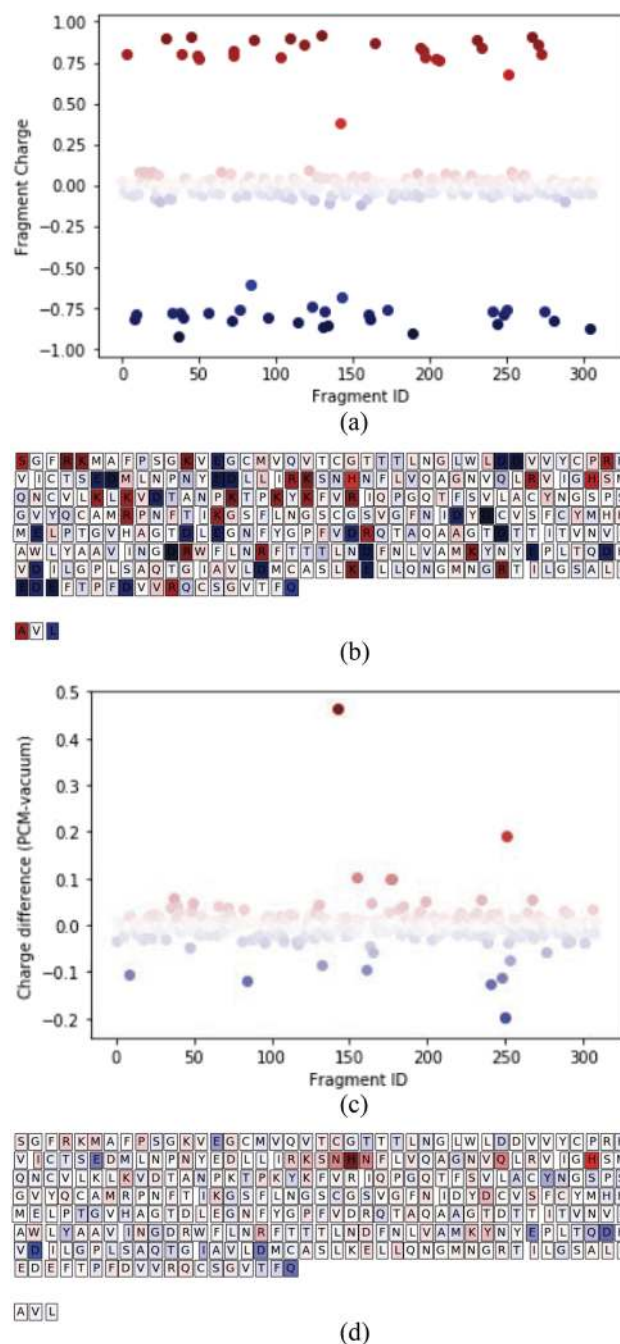

FIG. 12. Charges on the amino acids calculated by the fragment population analysis [(a) and (b)] and their difference with respect to implicit solvent calculation, which is performed with the same technique of the study in Sec. V A in (c) and (d). For simplicity, we have employed the amino acid-based fragmentation to present the data.

and identify which fragments of the systems interact by sharing some electrons with this region. We can reconstruct the fragmentation of the system in a way such as to focus on the active site in a specific portion of the protein. In this example, we will focus around the
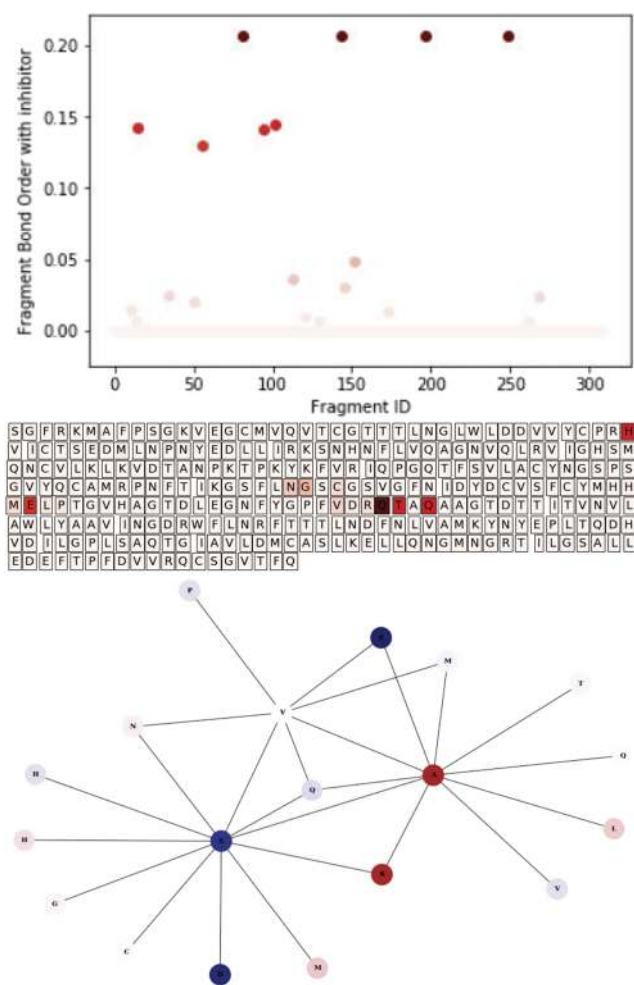
FIG. 13. (Top) Value of the fragment bond order between the inhibitor and the amino acids of the enzymatic sequence. (Bottom) A graphic representation of the fragments that interact most with the inhibitor, where the amino acids are also colored by the following charge color code as per Fig. 12.

inhibitor. We show in Fig. 13 which are the sections of the amino acid sequence that has a non-negligible interaction with the fragments that belong to the chain of the inhibitor. Such representation can be transformed into a graph-like view like in the case of Fig. 10, where the interacting fragments may also be characterized by their QM charge.

VIII. PARALLEL PERFORMANCE AND COMPUTING ARCHITECTURES

A. CPU performance

BigDFT incorporates a hybrid MPI (Message Passing Interface)/OpenMP parallelization scheme, with parts of the code also having been ported to GPUs, as described in Sec. VIII B. For both CS and LS BigDFT, the MPI parallelization is at the highest level with

the KS orbitals or, in the latter case SFs, divided between MPI tasks. There is therefore a fundamental limit on the number of MPI tasks that should be used for a given system (proportional to the number of atoms and **k**-points), as there should always be at least one orbital (SF) per MPI task for the cubic (linear) scaling approach. In order to reduce the MPI communication, two different data distributions are used to divide the orbitals/SFs among the tasks depending on the operation involved such that each task is either responsible for a fixed number of orbitals or for all orbitals, which are defined over a given set of grid points.

OpenMP is used to parallelize operations at a finer grained level. This allows the code to scale to a higher number of central processing unit (CPU) cores and is also important on architectures where memory is a limiting factor, since the use of OpenMP and a threaded BLAS library allows all processors on a given node to be exploited. Further details on the parallelization scheme, including a discussion of how good load balancing is optimized in the LS case, are given elsewhere.[42,43,92,93]

Thanks to the hybrid parallelization approach and the efficient properties of wavelets, it is possible to simulate up to a few hundred atoms even on a workstation. For example, Fig. 14(a) shows the timing breakdown for a PBE calculation on 428 atoms of a periodic disordered Ga_2O_3 structure on a 16 core workstation. Such a calculation is memory intensive but takes only of the order of an hour for a single point calculation, with the time dominated by the linear algebra and convolutions. Therefore, even without supercomputer access, CS BigDFT might offer some advantages in terms of performance. Nonetheless, BigDFT is primarily designed with massively parallel architectures in mind, and as also shown in Fig. 14(a), the same calculation takes only a few minutes on 16 nodes of Archer.

In order to explore the parallel performance in more detail, in Fig. 15, we show the performance of different modes of BigDFT for bulk silicon on Archer. Aside from the choice of method, the parallel performance also depends on the system in question, including the boundary conditions and simulation parameters. Nonetheless, similar trends will hold. In particular, it can be seen that the hybrid MPI/OpenMP approach results in good scaling up to a number of cores, which is significantly higher than would be possible for a pure MPI approach. Specifically, for Γ-point calculations of large supercells, a speedup is achieved up to around 256 nodes. For larger systems, it is possible to achieve speedups for much larger numbers of cores (within a pure CPU approach), for example, speedups have been demonstrated for more than 20 000 cores for around 14 000 atoms of DNA[43] and on more than 65 000 cores for hybrid functional calculations of 768 atoms of UO_2.[50]

It is also worth noting the absolute walltime. For bulk Si, 512 atoms are below the crossover point between cubic and linear scaling BigDFT, i.e., the cubic scaling approach is still faster than the linear scaling approach. This is because fully periodic systems generally have a higher crossover point, while a large number of SFs per atom are required to achieve a reasonable accuracy. Further discussion on the crossover between cubic, linear, and fragment approaches is presented in Sec. VIII C. The cubic approach takes less than two minutes of walltime for a complete single point calculation with PBE. In the case of a hybrid functional, this increases significantly; however, it is nonetheless possible to complete a single point calculation on such a large supercell in a little more than an hour even within a
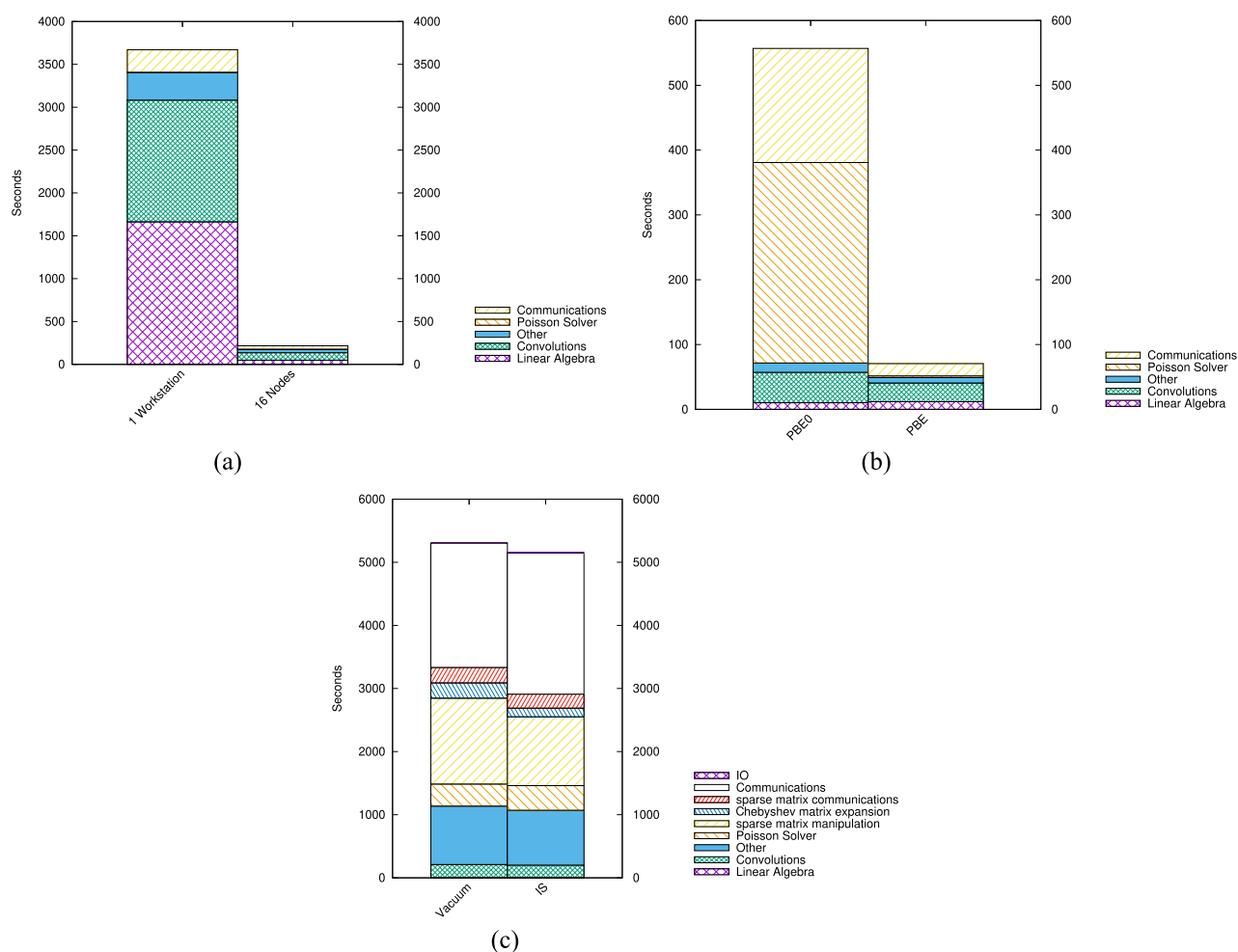
FIG. 14. Performance of cubic scaling BigDFT for different systems, XC functionals, parallelization modes, and architectures. Calculations are at the Γ point only, and walltimes are for single-point ground state calculations. (a) CPU only PBE timings for 428 atoms of a periodic disordered Ga_2O_3 structure on a 16 core Intel Xeon 6130 workstation using 16 MPI tasks and 2 OpenMP threads, compared to 16 nodes on Archer using 6 MPI tasks and 4 OpenMP threads per node. Calculations took 21 self-consistent iterations to converge. (b) CPU-GPU PBE and PBE0 timings for the TiO_2 slab presented in Sec. V A using 144 nodes on Daint (with GPU acceleration for PBE0). The slab contains 287 atoms (1149 KS orbitals), and the calculations took 12 self-consistent iterations to converge. For the PBE0 calculation, there are 660 675 different solutions of Poisson's equation *per SCF iteration*, but thanks to GPU acceleration, it can nonetheless be executed in less than 10 min. (c) Comparison of the walltime needed to perform a full SCF loop *and* to write the support function matrices on the disk for the SARS-CoV-2 main protease calculations that are presented in Sec. VII, made on 8 nodes of the Rome partition of Irene supercomputer Computing times are similar between vacuum and implcit solvent (IS), even slightly lower for IS, but mostly due to fluctuation in the network performance.

CPU-only approach. Further savings might be expected in a GPU-based approach (see Sec. VIII B). The LS walltime is reduced when the pseudo-fragment approach is used to generate an input guess from a small molecular cluster, and even further when the SFs from a smaller supercell are used without further optimization, although the walltime remains higher than the cubic approach.

B. GPU acceleration

In the past few years, the possibility of using graphics processing units for scientific calculations has increased a lot of interest. A technology initially developed for home personal computer (PC)

hardware has rapidly evolved in the direction of programmable parallel streaming processors. The features of these devices, in particular the very low price to performance ratio, together with the relatively low energy consumption, make them attractive platforms for intensive scientific computations.

The operations of BigDFT are well suited for GPU acceleration. For example, the computational nature of 3D separable convolutions allows efficient routines, which may benefit from GPU computational power. The parallelization scheme of BigDFT is also optimal in this sense: GPUs can be used without affecting the nature of the communications between the different MPI processes. This is in the same spirit as the multi-level MPI/OpenMP parallelization. Porting
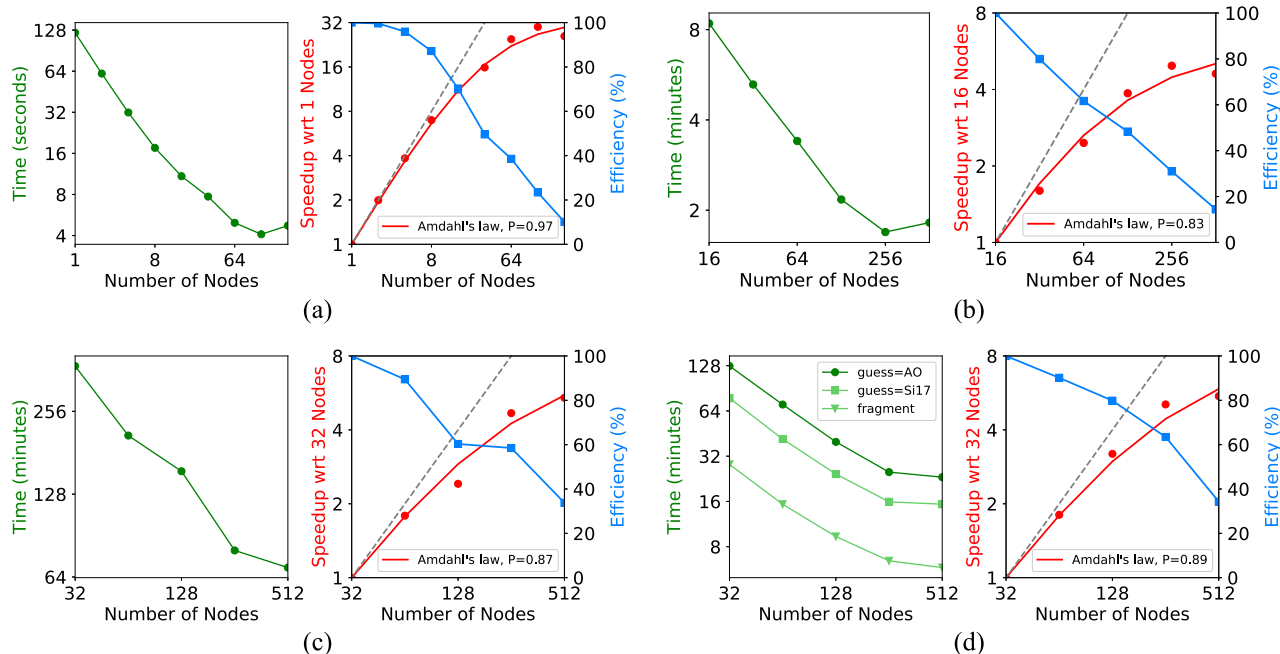
FIG. 15. Parallel performance for bulk Si on Archer using 6 MPI tasks and 4 OpenMP threads per node, including walltime, speedup, efficiency, and fit to Amdahl's law. There are three LS scenarios: a standard atomic orbital ("AO") SF input guess, a guess whereby the SFs are generated for a Si atom in the center of an isolated cluster of 17 Si atoms ("Si17") and for the pseudo-fragment approach with optimized SFs from a 216 atom supercell reused as a fixed basis ("fragment"). h = 0.43 bohr with 9 SFs per atom and R_{loc} = 8 bohrs for LS calculations. (a) Cubic scaling PBE calculation in an 8 atom unit cell with 4 × 4 × 4 k-points, (b) cubic scaling PBE calculation in a 512 atom supercell at the Γ-point, (c) cubic scaling PBE0 calculation in a 512 atom supercell at the Γ-point, and (d) linear scaling PBE calculation in a 512 atom supercell at the Γ-point. The time is shown for three scenarios, while speedup and efficiency are shown only for the AO case.

has been done within the Khronos' OpenCL standard, which allows for multi-architecture acceleration. We have therefore at hand a multilevel parallelized code, combining MPI, OpenMP, OpenCL, and CUDA (for the FFT and linear algebra), which can work on state-of-the-art hybrid supercomputers. Thanks to OpenCL porting, even heterogeneous architectures are at hand. Further details on the implementation and performance of BigDFT with GPUs are given in Refs. 17, 93, and 94.

The operations that have to be explicitly ported to GPUs are a set of separable three-dimensional convolutions. For a code with the complexity of BigDFT, the evaluation of the benefits of using a GPU-accelerated code must be performed at three different levels.

Initially, one has to evaluate the effective speedup provided by the GPU kernels with respect to the corresponding CPU routines that perform the same operations. This is the "bare" speedup, which, of course, for a given implementation, depends on the computational power that the device can provide. It has to be kept in mind that vendors, who do not know about the details of the full code, *are only able to provide bare speedups.*

At the second level, the "complete" speedup has to be evaluated; the performances of the whole hybrid CPU/GPU code should be analyzed with respect to the pure CPU execution. Clearly, this result depends on the actual importance of the ported routines in the context of the whole code (i.e., following Amdahl's law). This is the first reliable result of the actual performance enhancement of

porting the code to GPUs. For a hybrid code that originates from a single-core CPU program, this is the last level of evaluation.

However, for a parallel code, there is still another step that has to be evaluated. This is the behavior of the hybrid code in a parallel
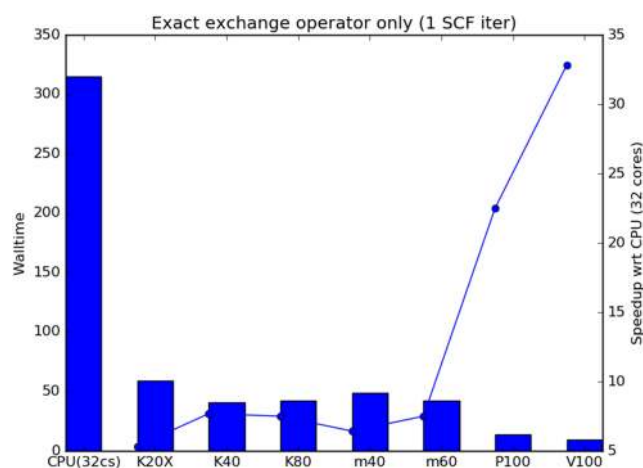

FIG. 16. Increase in the code performance with various GPU architectures for the calculation of the exact exchange operator on a system of 32 water molecules.

(or massively parallel) environment. Indeed, for parallel runs, the picture is complicated by two things. First, the management of the extra level of communication is introduced by the PCI (Peripheral Component Interconnect)-express bus, which may interact negatively with the underlying code communication scheduling (MPI or OpenMP, for example). The second is the behavior of the code for a number of GPU devices, which is lower than the number of CPU processes that are running. In this case, the GPU resource is not homogeneously distributed—managing this fact adds an extra level

of complexity. The evaluation of the code at this stage contributes at the "user level" speedup, which is the actual time-to-solution speedup. Such considerations are not specific to BigDFT and, thus, may be useful for any developer of complex codes like those typically used in scientific computing.

One area where the use of GPUs within BigDFT is particularly advantageous is for hybrid XC functional calculations. Hybrid functional calculations of periodic solids in systematic basis sets typically incur a very high computational cost relative to semi-local
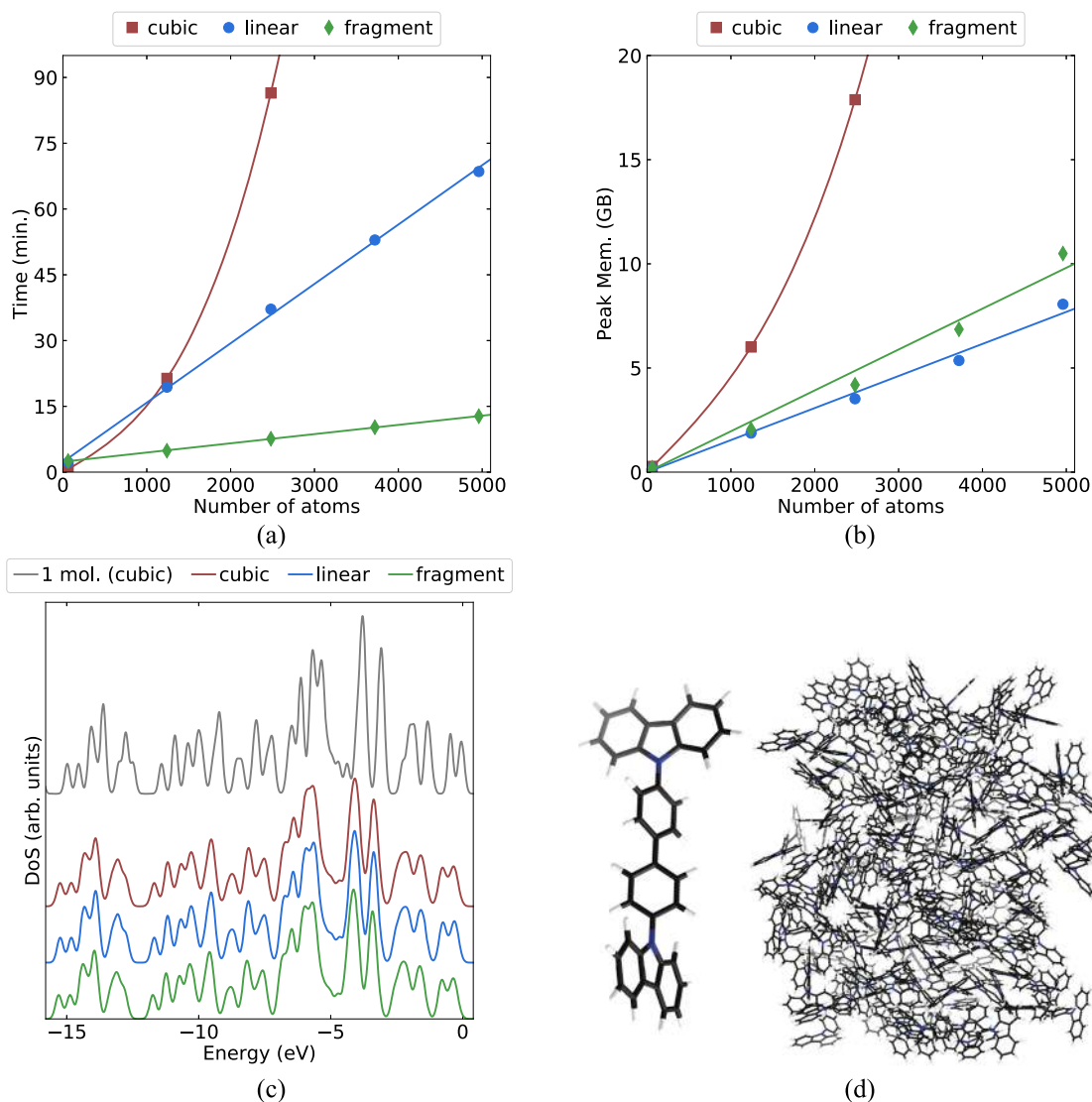

FIG. 17. Time and memory scaling for PBE single point calculations of disordered clusters of CBP.[85] The DoS is also shown for the different methods; the corresponding total energies are given in Table III. h = 0.45 bohr with 4 (1) SFs per C/N (H) atom and R_{loc} = 6 bohrs. Simulations used 48 nodes on Archer with 6 MPI tasks and 4 OpenMP threads per node. (a) Total walltime. The cost of the template calculation is included in the fragment timings. The crossovers with respect to the cubic approach are at 1068 and 256 atoms for the linear and fragment approaches, respectively. (b) Peak memory usage per MPI. Both linear and fragment approaches have lower memory usage than the cubic approach even for the single molecule (62 atoms). (c) Occupied DoS for 2480 atoms, alongside the (scaled) single molecule DoS. Gaussian smearing of 0.1 eV has been applied, and the spectra have been aligned to the HOMO. (d) Depiction of 1 molecule (left) and 4960 atom structure (right) of CBP with C/N/H atoms depicted in gray/blue/white.

functionals such that calculations on large systems are prohibitively expensive in a well converged basis set. The ratio γ of walltimes between a hybrid functional (e.g., PBE0) and equivalent semi-local functional (e.g., PBE) thus gives an indicator of the affordability of a hybrid functional calculation. Using a recently implemented GPU approach in BigDFT, γ has been shown to be significantly lower when using GPUs compared to a CPU-only implementation. As an example, for 324 atoms of UO_2 $\gamma = 14.4$, while $\gamma = 4.1$ in the equivalent GPU calculation.[17] Such an approach enables production calculations on large systems with hybrid functionals, for example, the case study on TiO_2 presented in Sec. V A. In Fig. 14(b), we show an example of the timings for this system using both PBE and PBE0. While the PBE0 calculation incurs a significantly higher cost for both the Poisson solver and communications, the ratio remains modest.

Another advantage that emerges from the usage of GPU acceleration is that the code can benefit from architectural evolutions, which are still very effective in the domain of GPGPU computing. As an illustration, we show in Fig. 16 the capability of the code to be adapted to the different generations of GPU cards by showing the speedup and walltime (in seconds) for the calculation of the Fock exchange operator for a system of 32 water molecules (128 KS orbitals) with respect to a single-node calculation with 32 CPU cores. These runs were accelerated by using 4 GPU cards per node.

C. Scaling with system size

Thanks to both the favorable properties of the wavelet basis set and the efficient parallelization described above, it is already possible to treat several hundred atoms using the cubic scaling approach in BigDFT. It is therefore worth asking the question as to when it is useful to switch to the LS approach. In other words, where is the crossover point between the two approaches? As an example, we compare the performance of the cubic, linear, and fragment approaches for 4,4′-N,N′-dicarbazole-biphenyl (CBP), which is a typical host material for organic LEDs. We consider a number of disordered clusters of CBP with increasing size. Since the molecules interact relatively weakly within the cluster and there is only a single template repeated many times, this provides an excellent test case for the fragment approach. Indeed, the fragment approach was originally motivated by the goal of simulating such systems.[85]

As shown in Fig. 17, the (occupied) DoS is in remarkable agreement between the three approaches, confirming the suitability of the fragment approach for treating such types of systems. In Fig. 17, we also show both the walltime and memory scaling, demonstrating the expected scaling behavior for each method. Here, we limit the simulations to around 5000 atoms; however, LS-BigDFT has been tested for system sizes of around 30 000 atoms. Thanks to both wavelet properties and efficient parallelization, the cubic approach is already able to treat more than 2000 atoms. The crossover point between cubic and linear scaling BigDFT is around 1000 atoms for the walltime and significantly lower for memory usage, although these depend strongly on the properties of the system, including the dimensionality[81] and the simulation parameters. As can be seen, the prefactor for the walltime is significantly lower for the fragment approach compared to the linear, reducing both the crossover and the computational cost of treating large systems. There is a small increase in the memory requirements of a fragment calculation, which is due to the SF load balancing being less well optimized in fragment calculations.

IX. SOFTWARE APPROACH

From version 1.8.0 of BigDFT, which can be downloaded using the command `git clone` https://gitlab.com/l_sim/bigdft-suite, the build system of BigDFT has been modified. Instead of building the code with one single configure line, the code is now built as a suite of different packages.

In Fig. 18, we see how the BigDFT code is separated into packages. This figure describes the interdependencies among these packages. The packages might be separated into *upstream* contributions (i.e., not associated with the developers of BigDFT) and *native* contributions. We have used a build suite tool based on the `JHBUILD` project, which is regularly used by developers of the "gnome" project. We have re-adapted/added some of the functionality of the `JHBUILD` package to meet the needs of our package.

The most important upstream packages that are employed in the BigDFT code are as follows:

libyaml: This library is used to parse the "yaml <http://yaml.org/>" Markup language, which is used in the BigDFT input files.

PyYaml: "<https://pyyaml.org/>": This is a Python module, which makes it possible to convert Yaml into python objects. This part is mainly used for postprocessing purposes as the BigDFT logfile also comes in the Yaml format.

libXC: This library[78] handles most of the XC functionals, which can be invoked from BigDFT runs.

GaIn: This library handles analytic integrals of common operators between Gaussian functions. It does not perform low-level operations and can be linked separately.

The native libraries arising from the BigDFT project are the following:

futile: A library handling most common Fortran low-level operations, such as memory management, profiling routines, and I/O operations. It also supports Yaml output and parsing for Fortran programs. It additionally provides wrapper routines to MPI and linear algebra operations. This library is used extensively in BigDFT packages.

CheSS: A module for performing the Fermi operator expansion via Chebyshev polynomials, released as a separate project on "Launchpad <https://launchpad.net/chess>".

psolver: A flexible real-space Poisson solver based on interpolating scaling functions. It constitutes a fundamental building block of the BigDFT code, and it can also be used separately and linked to other codes. It also internally uses the "futile" library for I/O.

libabinit: This is a subsection of files coming from the ABINIT[95] software package to which BigDFT has been coupled since the early days. It handles different parts such as symmetries, ewald corrections, PAW routines, density and potential mixing routines, and some molecular dynamics minimizers. Some XC functionals, initially natively implemented in the "ABINIT" code, have also been coded in this library. This library also uses
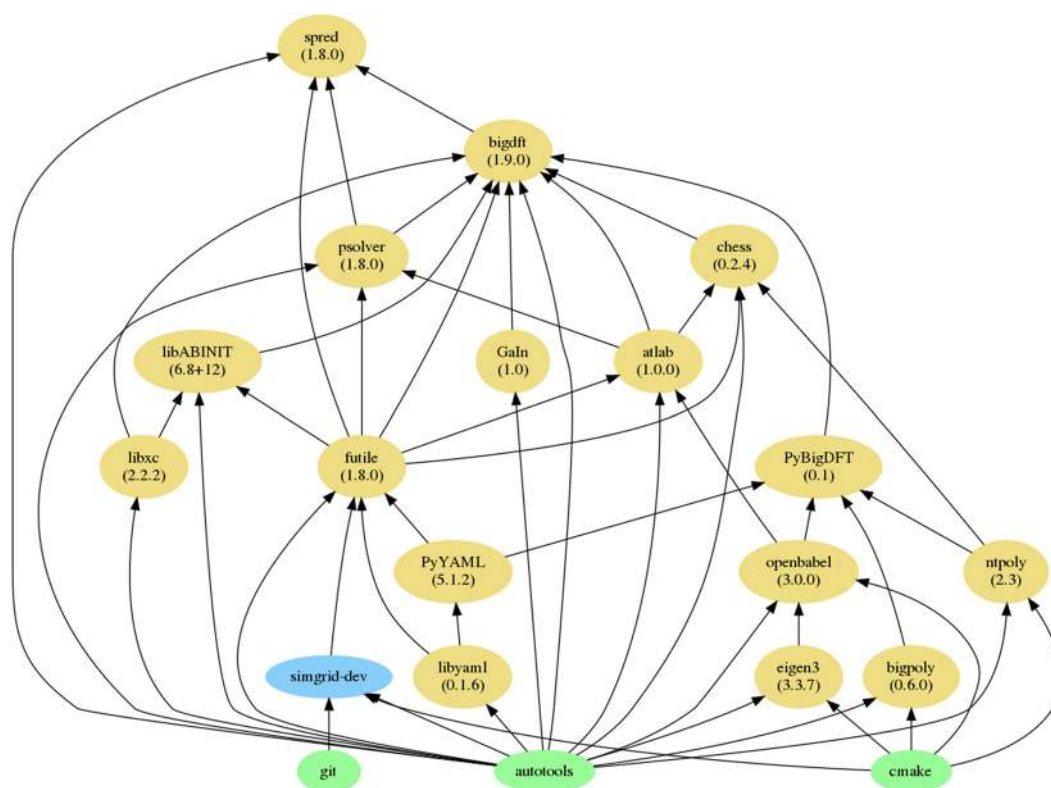
FIG. 18. Interdependencies between the various packages of the BigDFT suite.

the "futile" code through the (experimental) PAW section.

BigDFT: The core routines of the software suite.

spred: A library for structure prediction tools, which is compiled on top of BigDFT routines.

In previous versions of BigDFT, all these different packages were compiled with the same configuration instructions. With the present version, each of the code sections described above can be considered as a separate package (some more are upcoming), which improves modularity between code sections and reduces side-effects. In addition, each package can now be compiled with different build and installation instructions, and even using different build systems.

A. PyBigDFT computational approach

The diverse capabilities of BigDFT can lead to elaborate calculation workflows as users attempt to study increasingly complex systems. In particular, when studying large systems, we anticipate users programmatically designing and refining their systems of interest over many calculation cycles. To facilitate such complex workflows, we have introduced high level Python bindings for BigDFT in the form of PyBigDFT. These bindings focus on input preparation, running calculations, and post-processing the results.

The fragment capabilities of BigDFT, introduced in Sec. VI, lead to a three-level view of a system:

- Atoms: the lowest level view of a system. Atoms are dictionary-like objects. This makes it possible to store any type of observable computed over the course of a calculation.
- Fragments: a list of atoms. Fragments are decorated with accessors to fragment level observables, as well as methods for rototranslation.
- System: a named collection of fragments.

Helper routines are provided for these data structures such as file I/O using common formats, interaction with visualization programs, and conversion to data structures used in other Python libraries commonly employed in computational chemistry (OpenBabel,[96] ASE,[97] etc.).

When a system has been built, calculations can then be performed using the following classes:

- Inputfile: while input files for BigDFT are simply Yaml files and could be stored as a dictionary, the Inputfile class provides additional helper routines for common calculation parameters.
- Logfile: similar to Inputfile, a Logfile is also just a Yaml file, and this class exposes that interface along with helper routines to access commonly used information.

- SystemCalculator: given an input file and a system, this class invokes the BigDFT code, and places the results into a Logfile.

Once a calculation has been completed, quantities of interest can be extracted using the post-processing facilities provided in PyBigDFT. These post-processing features are a mix of pure Python routines and wrappers for Fortran utility programs, which can perform more computationally demanding analysis. We encourage users to perform all calculations through these high level bindings, particularly in the form of Python notebooks. This results in complete, reproducible, and self-documented artifacts to accompany a calculation result, ensuring an increased reliability of any calculation. To this end, we aim to provide alongside publications using BigDFT notebooks that reproduce the workflow used, particularly in the case where new functionalities are introduced (see, for example, Refs. 50, 87, and 98). The benefits of such an approach go far beyond reproducibility. Indeed, we do not anticipate that many readers will simply take a provided notebook, run it, and be satisfied. Rather, we believe the main benefit of this approach lies in computational *continuity*. By combining an all in-one-workflow with PyBigDFT's built-in features for lazy evaluation, it becomes possible to quickly build on top of a previous result and to check its reliability by performing additional predictions and analysis.

X. OVERVIEW OF FUNCTIONALITIES IN BigDFT

A number of functionalities are available within BigDFT, although some are currently only available within, e.g., CS BigDFT. The current status of core functionalities is summarized in Table IV, which also highlights developments that are currently in progress.

Beyond these functionalities, BigDFT has also been widely used within the context of structure prediction approaches such as minima hopping,[79,99] enabled by the programs of the spred library (see Refs. 26, 27, and 100–114). In such cases, a precise calculation of the total energies is required for which the CS BigDFT is well suited, while the parameters used in a typical LS calculation are not suitable for obtaining accurate energetic orderings where different structures are very close in energy.[43]

One active area of development, which is worth highlighting, is the treatment of excited states. Time-dependent DFT is available within CS BigDFT,[98] although currently only local density approximation (LDA)[115] calculations are possible. Aside from TD-DFT, constrained DFT (CDFT)[116] is also available for the treatment of charge-transfer (CT) excitations.[80] The CDFT implementation has also been used to calculate on-site energies in a large disordered host–guest supramolecular material typically used for organic LEDs.[85] Work extending the CDFT approach for the treatment of local excitations (LE) is also currently underway, with the aim of providing a computationally efficient approach to simulating both CT and LE states in large systems. Since CDFT has been implemented within the framework of the fragment approach, this requires the existence of a fixed SF basis, which has enough degrees of freedom to also represent excited states. As discussed in Sec. IV G, the direct minimization approach can be used to generate a SF basis, which is capable of representing a few unoccupied states. We are also exploring other approaches to generating SFs for excited states.

TABLE IV. Current status of the various core functionalities that are available in BigDFT.

	Cubic	Linear	Fragment
Parallelization			
MPI	✓	✓	✓
OpenMP	✓	✓	✓
GPU	✓	×[a]	×[a]
Pseudopotentials			
GTH/HGH	✓	✓	✓
Trouiller Martin norm-conserving		In progress	
PAW	✓	In progress	
Boundary conditions			
Free/wire/surface/periodic	✓	✓	✓
Non-orthorhombic cells		In progress	
Electronic structure			
k-points	✓	×	×
Metals	✓	✓	✓
Spin polarization	✓	✓	✓
Functionals			
Semi-local functionals	✓	✓	✓
Hybrid functionals[b]	✓	In progress	
Environments			
Empirical van der Waals	✓	✓	✓
Explicit charges[c]	✓	✓	✓
External electric field[d]	✓	✓	✓
Electrostatic embedding	✓	✓	✓
Dynamics			
Geometry optimizations	✓	✓	×
Molecular dynamics	✓	✓	×
Post-processing			
Projected densities of states	✓	✓	✓
Charge analysis	✓	✓	✓
Excitations			
Time-dependent DFT[e]	✓	×	×
Constrained DFT		In progress	✓

[a] Possible but not advantageous (no hot spot operations).
[b] Γ-point only.
[c] Meaningful for free BC only.
[d] Meaningful in the isolated directions only.
[e] LDA only.

XI. PERSPECTIVE

Daubechies wavelets have a number of favorable properties that make them an ideal basis set for electronic structure calculations. In this work, we have outlined the use of such a basis set for density functional theory calculations, as implemented in the open source BigDFT code. Through examples presented here and referenced within the text, we have shown how the combination of a wavelet basis set with an implementation designed for massively parallel machines allows for efficient and accurate calculations of hundreds of atoms, even within a traditional cubic scaling approach. Such a treatment also allows for the simulation of relatively large systems using hybrid functionals, particularly where GPUs are available, while the availability of different boundary conditions allows for the straightforward treatment of molecules, surfaces, and solids. A number of functionalities are available in BigDFT, including dynamics, explicit charges and electric fields, and implicit environments, while there are ongoing developments in, e.g., the treatment of excited states.

Going beyond the cubic scaling approach, the localized nature of wavelets is also highly suitable for a linear scaling approach wherein the nearsightedness of matter is exploited by imposing localization on the system via the use of a minimal set of localized support functions. Such an approach further expands the applicability of BigDFT to systems containing several thousand atoms and has been shown to converge reliably for a range of materials. The localized support function based approach may also be further exploited to define a fragment approach in which the computational cost may be significantly reduced by exploiting repetition in molecular or periodic systems.

The treatment of such large systems brings new types of problems within the reach of first principles simulations. However, such simulations also bring new challenges, for example, the increased size of the configuration space associated with complex materials containing large numbers of atoms. Furthermore, as well as treating large length scales, it is desirable to treat also long timescales, which remains unfeasible within a purely quantum mechanical approach so that QM/MM approaches are required. To this end, the support function approach allows not only the treatment of large enough systems to test and validate QM/MM approaches but can also be used to analyze and fragment a system without requiring any *a priori* knowledge of the system. This offers a route to reduce the complexity of QM calculations of large systems and thereby inform the setup of multiscale simulations.

In the context of large scale electronic structure simulations, a wavelet-based approach therefore offers another significant advantage in that it facilitates the implementation of a range of approaches designed to treat different system sizes, as illustrated in Fig. 19. The existence of such a comprehensive framework with a single underlying formalism means that each successive approximation is applied in a systematic and controlled manner. Therefore, one can easily test and validate the approximations between different levels of theory. For example, QM/MM simulations may be benchmarked with respect to fragment calculations, which may be compared with full linear scaling calculations and so on. Furthermore, we have also introduced a number of indicators, which can be used to predict whether or not a particular approximation is appropriate for a given system.
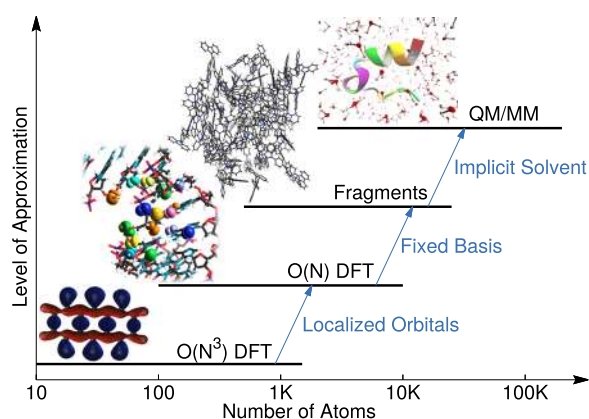

FIG. 19. Illustration of the different approaches available in BigDFT, the approximations introduced, and approximate applicable length scales.

The development and improvement of a comprehensive multiscale framework is an ongoing priority for BigDFT. As with other developments in BigDFT, a key aspect of this is a module-based approach that aims toward sustainable software development. In tandem with the implementation of new approaches, we also aim to prioritize both accessibility and reproducibility of new functionalities, by providing jupyter-notebooks demonstrating how such functionalities may be used via the PyBigDFT interface. Further information on BigDFT may be found on the website[117] and in the documentation,[118] while the code may be downloaded from GitLab.[119]

DATA AVAILABILITY

The data that support the findings of this study are available within the article.

REFERENCES

[1] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).

[2] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).

[3] K. Lejaeghere, G. Bihlmayer, T. Björkman, P. Blaha, S. Blügel, V. Blum, D. Caliste, I. E. Castelli, S. J. Clark, A. Dal Corso, S. de Gironcoli, T. Deutsch, J. K. Dewhurst, I. Di Marco, C. Draxl, M. Dułak, O. Eriksson, J. A. Flores-Livas, K. F. Garrity, L. Genovese, P. Giannozzi, M. Giantomassi, S. Goedecker, X. Gonze, O. Grånäs, E. K. U. Gross, A. Gulans, F. Gygi, D. R. Hamann, P. J. Hasnip, N. A. W. Holzwarth, D. Iuşan, D. B. Jochym, F. Jollet, D. Jones, G. Kresse, K. Koepernik, E. Küçükbenli, Y. O. Kvashnin, I. L. M. Locht, S. Lubeck, M. Marsman, N. Marzari, U. Nitzsche, L. Nordström, T. Ozaki, L. Paulatto, C. J. Pickard, W. Poelmans, M. I. J. Probert, K. Refson, M. Richter, G.-M. Rignanese, S. Saha, M. Scheffler, M. Schlipf, K. Schwarz, S. Sharma, F. Tavazza, P. Thunström, A. Tkatchenko, M. Torrent, D. Vanderbilt, M. J. van Setten, V. Van Speybroeck, J. M. Wills, J. R. Yates, G.-X. Zhang, and S. Cottenier, Science **351**, aad3000 (2016).

[4] I. Daubechies, *Ten Lectures on Wavelets* (SIAM, 1992).

[5] See https://mrchem.readthedocs.io/en/latest/ for MultiResolution Chemistry (MRChem) program package.

[6] R. J. Harrison, G. Beylkin, F. A. Bischoff, J. A. Calvin, G. I. Fann, J. Fosso-Tande, D. Galindo, J. R. Hammond, R. Hartman-Baker, J. C. Hill, J. Jia, J. S. Kottmann, M.-J. Yvonne Ou, J. Pei, L. E. Ratcliff, M. G. Reuter, A. C. Richie-Halford, N. A. Romero, H. Sekino, W. A. Shelton, B. E. Sundahl, W. S. Thornton, E. F. Valeev, Á. Vázquez-Mayagoitia, N. Vence, T. Yanai, and Y. Yokoi, SIAM J. Sci. Comput. **38**, S123 (2016).

[7] S. Goedecker, *Wavelets and Their Application: For The Solution of Partial Differential Equations in Physics* (Presses Polytechniques et Universitaires Romandes, 1998).

[8] L. Genovese and T. Deutsch, Phys. Chem. Chem. Phys. **17**, 31582 (2015).

[9] A. I. Neelov and S. Goedecker, J. Comput. Phys. **217**, 312 (2006).

[10] S. Goedecker, M. Teter, and J. Hutter, Phys. Rev. B **54**, 1703 (1996).

[11] C. Hartwigsen, S. Goedecker, and J. Hutter, Phys. Rev. B **58**, 3641 (1998).

[12] M. Krack, Theor. Chem. Acc. **114**, 145 (2005).

[13] A. Willand, Y. O. Kvashnin, L. Genovese, Á. Vázquez-Mayagoitia, A. K. Deb, A. Sadeghi, T. Deutsch, and S. Goedecker, J. Chem. Phys. **138**, 104109 (2013).

[14] L. Genovese, T. Deutsch, A. Neelov, S. Goedecker, and G. Beylkin, J. Chem. Phys. **125**, 074105 (2006).

[15] L. Genovese, T. Deutsch, and S. Goedecker, J. Chem. Phys. **127**, 054704 (2007).

[16] A. Cerioni, L. Genovese, A. Mirone, and V. A. Sole, J. Chem. Phys. **137**, 134108 (2012).

[17] L. E. Ratcliff, A. Degomme, J. A. Flores-Livas, S. Goedecker, and L. Genovese, J. Phys.: Condens. Matter **30**, 095901 (2018).

[18] B. Natarajan, L. Genovese, M. E. Casida, T. Deutsch, O. N. Burchak, C. Philouze, and M. Y. Balakirev, Chem. Phys. **402**, 29 (2012).

[19] I. Dabo, Y. Li, N. Bonnet, and N. Marzari, "*Ab initio* electrochemical properties of electrode surfaces," in *Fuel Cell Science: Theory, Fundamentals and Bio-Catalysis*, edited by A. Wieckowski and J. Nørskov (John Wiley & Sons, Inc., 2010), pp. 415–431.

[20] O. Andreussi and G. Fisicaro, Int. J. Quantum Chem. **119**, e25725 (2019).

[21] G. Fisicaro, L. Genovese, O. Andreussi, N. Marzari, and S. Goedecker, J. Chem. Phys. **144**, 014103 (2016).

[22] G. Fisicaro, L. Genovese, O. Andreussi, S. Mandal, N. N. Nair, N. Marzari, and S. Goedecker, J. Chem. Theory Comput. **13**, 3829 (2017).

[23] J. Tomasi, B. Mennucci, and R. Cammi, Chem. Rev. **105**, 2999 (2005).

[24] O. Andreussi, I. Dabo, and N. Marzari, J. Chem. Phys. **136**, 064102 (2012).

[25] R. A. Puglisi, S. Caccamo, C. Bongiorno, G. Fisicaro, L. Genovese, S. Goedecker, G. Mannino, and A. La Magna, Sci. Rep. **9**, 5647 (2019).

[26] G. Fisicaro, M. Sicher, M. Amsler, S. Saha, L. Genovese, and S. Goedecker, Phys. Rev. Mater. **1**, 033609 (2017).

[27] G. Fisicaro, S. Filice, S. Scalese, G. Compagnini, R. Reitano, L. Genovese, S. Goedecker, I. Deretzis, and A. La Magna, J. Phys. Chem. C **124**, 2406 (2020).

[28] E. Hernández and M. J. Gillan, Phys. Rev. B **51**, 10157 (1995).

[29] J. D. Cloizeaux, Phys. Rev. **135**, A685 (1964).

[30] J. D. Cloizeaux, Phys. Rev. **135**, A698 (1964).

[31] W. Kohn, Phys. Rev. **115**, 809 (1959).

[32] R. Baer and M. Head-Gordon, Phys. Rev. Lett. **79**, 3962 (1997).

[33] S. Ismail-Beigi and T. A. Arias, Phys. Rev. Lett. **82**, 2127 (1999).

[34] S. Goedecker, Phys. Rev. B **58**, 3501 (1998).

[35] L. He and D. Vanderbilt, Phys. Rev. Lett. **86**, 5341 (2001).

[36] N. March, W. Young, and S. Sampanthar, *The Many-Body Problem in Quantum Mechanics*, Dover Books on Physics (Dover Publications, Incorporated, 1967).

[37] N. Marzari and D. Vanderbilt, Phys. Rev. B **56**, 12847 (1997).

[38] L. E. Ratcliff, S. Mohr, G. Huhs, T. Deutsch, M. Masella, and L. Genovese, Wiley Interdiscip. Rev.: Comput. Mol. Sci. **7**, e1290 (2017).

[39] M. Zaccaria, W. Dawson, V. Cristiglio, M. Reverberi, L. E. Ratcliff, T. Nakajima, L. Genovese, and B. Momeni, Curr. Opin. Biotech. **62**, 98 (2020).

[40] C.-K. Skylaris, P. D. Haynes, A. A. Mostofi, and M. C. Payne, J. Chem. Phys. **122**, 084119 (2005).

[41] D. R. Bowler and T. Miyazaki, J. Phys.: Condens. Matter **22**, 074207 (2010).

[42] S. Mohr, L. E. Ratcliff, P. Boulanger, L. Genovese, D. Caliste, T. Deutsch, and S. Goedecker, J. Chem. Phys. **140**, 204110 (2014).

[43] S. Mohr, L. E. Ratcliff, L. Genovese, D. Caliste, P. Boulanger, S. Goedecker, and T. Deutsch, Phys. Chem. Chem. Phys. **17**, 31360 (2015).

[44] P. E. Blöchl, Phys. Rev. B **50**, 17953 (1994).

[45] P. Pulay, Chem. Phys. Lett. **73**, 393 (1980).

[46] S. Goedecker and L. Colombo, Phys. Rev. Lett. **73**, 122 (1994).

[47] S. Goedecker and M. Teter, Phys. Rev. B **51**, 9455 (1995).

[48] S. Mohr, W. Dawson, M. Wagner, D. Caliste, T. Nakajima, and L. Genovese, J. Chem. Theory Comput. **13**, 4684 (2017).

[49] S. Mohr, M. Eixarch, M. Amsler, M. J. Mantsinen, and L. Genovese, J. Nucl. Mater. **15**, 64 (2018).

[50] L. E. Ratcliff and L. Genovese, J. Phys.: Condens. Matter **31**, 285901 (2019).

[51] M. Teeter, Proc. Natl. Acad. Sci. U. S. A. **81**, 6014 (1984).

[52] J. W. Neidigh, R. M. Fesinmeyer, and N. H. Andersen, Nat. Struct. Mol. Biol. **9**, 425 (2002).

[53] W. Dawson and T. Nakajima, Comput. Phys. Commun. **225**, 154 (2018).

[54] A. D. Daniels and G. E. Scuseria, J. Chem. Phys. **110**, 1321 (1999).

[55] X. Chen and S. S. Mao, Chem. Rev. **107**, 2891 (2007).

[56] A. Fujishima and K. Honda, Nature **238**, 37 (1972).

[57] R. Asahi, T. Morikawa, T. Ohwaki, K. Aoki, and Y. Taga, Science **293**, 269 (2001).

[58] M. Grätzel, Nature **414**, 338 (2001).

[59] R. Daghrir, P. Drogui, and D. Robert, Ind. Eng. Chem. Res. **52**, 3581 (2013).

[60] L. E. Walle, A. Borg, E. M. J. Johansson, H. Rensmo, P. Uvdal, and A. Sandell, J. Phys. Chem. C **115**, 9545 (2011).

[61] C. E. Patrick and F. Giustino, Phys. Rev. Appl. **2**, 014001 (2014).

[62] I. M. Nadeem, J. P. W. Treacy, S. Selcuk, X. Torrelles, H. Hussain, A. Wilson, D. C. Grinter, G. Cabailh, O. Bikondoa, C. Nicklin *et al.*, J. Phys. Chem. Lett. **9**, 3131 (2018).

[63] R. Martinez-Casado, G. Mallia, N. M. Harrison, and R. Pérez, J. Phys. Chem. C **122**, 20736 (2018).

[64] M. F. Calegari Andrade, H.-Y. Ko, R. Car, and A. Selloni, J. Phys. Chem. Lett. **9**, 6716 (2018).

[65] Y. He, O. Dulub, H. Cheng, A. Selloni, and U. Diebold, Phys. Rev. Lett. **102**, 106105 (2009).

[66] H. Cheng and A. Selloni, Phys. Rev. B **79**, 092101 (2009).

[67] C. C. Mercado, F. J. Knorr, J. L. McHale, S. M. Usmani, A. S. Ichimura, and L. V. Saraf, J. Phys. Chem. C **116**, 10796 (2012).

[68] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).

[69] B. Himmetoglu, A. Floris, S. de Gironcoli, and M. Cococcioni, Int. J. Quantum Chem. **114**, 14 (2014).

[70] B. J. Morgan and G. W. Watson, J. Phys. Chem. C **114**, 2321 (2010).

[71] J. P. Allen and G. W. Watson, Phys. Chem. Chem. Phys. **16**, 21016 (2014).

[72] E. Finazzi, C. Di Valentin, G. Pacchioni, and A. Selloni, J. Chem. Phys. **129**, 154113 (2008).

[73] H. Li, Y. Guo, and J. Robertson, J. Phys. Chem. C **119**, 18160 (2015).

[74] M.-A. Ha and A. N. Alexandrova, J. Chem. Theory Comput. **12**, 2889 (2016).

[75] F. Labat, P. Baranek, and C. Adamo, J. Chem. Theory Comput. **4**, 341 (2008).

[76] J. Fan, J. Z. Zhao, H. Xu, and S. Y. Tong, Phys. Rev. Lett. **115**, 149601 (2015).

[77] C. Adamo and V. Barone, J. Chem. Phys. **110**, 6158 (1999).

[78] M. A. L. Marques, M. J. T. Oliveira, and T. Burnus, Comput. Phys. Commun. **183**, 2272 (2012).

[79] S. Goedecker, J. Chem. Phys. **120**, 9911 (2004).

[80] L. E. Ratcliff, L. Genovese, S. Mohr, and T. Deutsch, J. Chem. Phys. **142**, 234105 (2015).

[81] L. E. Ratcliff and L. Genovese, in *Theory and Simulation in Physics for Materials Applications*, edited by E. Levchenko, Y. J. Dappe, and G. Ori (Springer International Publishing, 2020) (in press).

[82] G. Wahba, SIAM Rev. **7**, 409 (1965).

[83] W. Kabsch, Acta Crystallogr. A **34**, 827 (1978).

[84] F. L. Markley, J. Astronaut. Sci. **36**, 245 (1988).

[85] L. E. Ratcliff, L. Grisanti, L. Genovese, T. Deutsch, T. Neumann, D. Danilov, W. Wenzel, D. Beljonne, and J. Cornil, J. Chem. Theory Comput. **11**, 2077 (2015).

[86] S. Mohr, M. Masella, L. E. Ratcliff, and L. Genovese, J. Chem. Theory Comput. **13**, 4079 (2017).

[87] W. Dawson, S. Mohr, L. E. Ratcliff, T. Nakajima, and L. Genovese, J. Chem. Theory Comput. **16**(5), 2952–2964 (2020).

[88] V. Madhavi and S. Lele, BioResources **4**, 1694 (2009).

[89] See https://www.diamond.ac.uk/covid-19/for-scientists/Main-protease-structure-and-XChem.html for online resources about the outcome of the screening.

[90] See https://covid.postera.ai/covid for the website of the initiative.

[91] R. Hatada, K. Okuwaki, Y. Mochizuki, K. Fukuzawa, Y. Komeiji, Y. Okiyama, and S. Tanaka, chemRxiv:11988120.v1 (2020).

[92] L. Genovese, A. Neelov, S. Goedecker, T. Deutsch, S. A. Ghasemi, A. Willand, D. Caliste, O. Zilberberg, M. Rayson, A. Bergman, and R. Schneider, J. Chem. Phys. **129**, 014109 (2008).

[93] L. Genovese, B. Videau, M. Ospici, T. Deutsch, S. Goedecker, and J.-F. Méhaut, C. R. Méc. **339**, 149 (2011).

[94] L. Genovese, M. Ospici, T. Deutsch, J.-F. Méhaut, A. Neelov, and S. Goedecker, J. Chem. Phys. **131**, 034103 (2009).

[95] X. Gonze, B. Amadon, P.-M. Anglade, J.-M. Beuken, F. Bottin, P. Boulanger, F. Bruneval, D. Caliste, R. Caracas, M. Côté, T. Deutsch, L. Genovese, P. Ghosez, M. Giantomassi, S. Goedecker, D. R. Hamann, P. Hermet, F. Jollet, G. Jomard, S. Leroux, M. Mancini, S. Mazevet, M. J. T. Oliveira, G. Onida, Y. Pouillon, T. Rangel, G.-M. Rignanese, D. Sangalli, R. Shaltaf, M. Torrent, M. J. Verstraete, G. Zerah, and J. W. Zwanziger, Comput. Phys. Commun. **180**, 2582 (2009).

[96] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison, J. Cheminf. **3**, 33 (2011).

[97] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiřtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, J. Phys.: Condens. Matter **29**, 273002 (2017).

[98] M. D'Alessandro and L. Genovese, Phys. Rev. Materials **3**, 023805 (2019).

[99] B. Schaefer, S. Mohr, M. Amsler, and S. Goedecker, J. Chem. Phys. **140**, 214102 (2014).

[100] A. Willand, M. Gramzow, S. A. Ghasemi, L. Genovese, T. Deutsch, K. Reuter, and S. Goedecker, Phys. Rev. B **81**, 201405 (2010).

[101] P. Pochet, L. Genovese, D. Caliste, I. Rousseau, S. Goedecker, and T. Deutsch, Phys. Rev. B **82**, 035431 (2010).

[102] E. Machado-Charry, L. K. Béland, D. Caliste, L. Genovese, T. Deutsch, N. Mousseau, and P. Pochet, J. Chem. Phys. **135**, 034102 (2011).

[103] S. A. Ghasemi, M. Amsler, R. G. Hennig, S. Roy, S. Goedecker, T. J. Lenosky, C. Umrigar, L. Genovese, T. Morishita, and K. Nishio, Phys. Rev. B **81**, 214107 (2010).

[104] M. Amsler, S. Alireza Ghasemi, S. Goedecker, A. Neelov, and L. Genovese, Nanotechnology **20**, 445301 (2009).

[105] S. De, S. A. Ghasemi, A. Willand, L. Genovese, D. Kanhere, and S. Goedecker, J. Chem. Phys. **134**, 124302 (2011).

[106] P. Pochet, L. Genovese, S. De, S. Goedecker, D. Caliste, S. A. Ghasemi, K. Bao, and T. Deutsch, Phys. Rev. B **83**, 081403 (2011).

[107] E. Machado-Charry, P. Boulanger, L. Genovese, N. Mousseau, and P. Pochet, Appl. Phys. Lett. **101**, 132405 (2012).

[108] P. Boulanger, M. Morinière, L. Genovese, and P. Pochet, J. Chem. Phys. **138**, 184302 (2013).

[109] S. Mohr, P. Pochet, M. Amsler, B. Schaefer, A. Sadeghi, L. Genovese, and S. Goedecker, Phys. Rev. B **89**, 041404 (2014).

[110] S. De, A. Willand, M. Amsler, P. Pochet, and L. Genovese, Phys. Rev. Lett. **106**, 225502 (2011).

[111] S. Krishnan, G. Brenet, E. Machado-Charry, D. Caliste, L. Genovese, T. Deutsch, and P. Pochet, Appl. Phys. Lett. **103**, 251904 (2013).

[112] S. Saha, L. Genovese, and S. Goedecker, Sci. Rep. **7**, 7618 (2017).

[113] D. S. De, J. A. Flores-Livas, S. Saha, L. Genovese, and S. Goedecker, Carbon **129**, 847 (2018).

[114] F. Lançon, L. Genovese, and J. Eymery, Phys. Rev. B **98**, 165306 (2018).

[115] D. M. Ceperley and B. J. Alder, Phys. Rev. Lett. **45**, 566 (1980).

[116] B. Kaduk, T. Kowalczyk, and T. Van Voorhis, Chem. Rev. **112**, 321 (2012).

[117] See http://www.bigdft.org for the BigDFT official website.

[118] See http://bigdft-suite.readthedocs.io/en/latest for the developer's documentation.

[119] See http://gitlab.com/l_sim/bigdft-suite for the production branch repository.