



A spectral scheme for Kohn–Sham density functional theory of clusters



Amartya S. Banerjee, Ryan S. Elliott*, Richard D. James

Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN 55455, USA

ARTICLE INFO

Article history:

Received 14 April 2014

Received in revised form 24 November 2014

Accepted 3 February 2015

Available online 12 February 2015

Keywords:

Spectral scheme

Kohn–Sham density functional theory

Spherical harmonics

Spherical Bessel functions

Computational efficiency

Spectral convergence

Parallel scaling performance

Eigenvalue problem

LOBPCG

Chebyshev filtering

Nano clusters

Super atoms

ABSTRACT

Starting from the observation that one of the most successful methods for solving the Kohn–Sham equations for periodic systems – the plane-wave method – is a spectral method based on eigenfunction expansion, we formulate a spectral method designed towards solving the Kohn–Sham equations for clusters. This allows for efficient calculation of the electronic structure of clusters (and molecules) with high accuracy and systematic convergence properties without the need for any artificial periodicity. The basis functions in this method form a complete orthonormal set and are expressible in terms of spherical harmonics and spherical Bessel functions. Computation of the occupied eigenstates of the discretized Kohn–Sham Hamiltonian is carried out using a combination of preconditioned block eigensolvers and Chebyshev polynomial filter accelerated subspace iterations. Several algorithmic and computational aspects of the method, including computation of the electrostatics terms and parallelization are discussed. We have implemented these methods and algorithms into an efficient and reliable package called ClusterES (Cluster Electronic Structure). A variety of benchmark calculations employing local and non-local pseudopotentials are carried out using our package and the results are compared to the literature. Convergence properties of the basis set are discussed through numerical examples. Computations involving large systems that contain thousands of electrons are demonstrated to highlight the efficacy of our methodology. The use of our method to study clusters with arbitrary point group symmetries is briefly discussed.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Over the past few decades, quantum mechanical calculations based on Kohn–Sham Density Functional Theory (KS-DFT) have provided important insights into a variety of material systems [1–3]. One of the most widely used and successful methods for numerical solution of the equations of Kohn–Sham theory is the pseudopotential plane-wave method [4–8], currently available in a number of software packages [4,9–11].

The advantages of plane-waves include the fact that they are orthonormal and therefore result in simple discretized expressions. Also, they form a complete basis, thus allowing for systematic convergence with increasing basis set size, governed by a single parameter, the energy cutoff. The global nature of the plane wave basis also results in minimum user interference in terms of basis set choice. Being a Fourier basis, plane-waves allow for spectral convergence leading to highly accurate numerical solutions [12]. Further, independence of the basis functions on atomic positions results in the absence

* Corresponding author.

E-mail addresses: baner041@umn.edu (A.S. Banerjee), relliott@umn.edu (R.S. Elliott), james@umn.edu (R.D. James).

of (the otherwise difficult to compute) Pulay forces [10]. On the downside, while the plane-wave method is ideally suited for the study of periodic systems such as crystals, its application to non-periodic systems such as molecules and clusters is more limited due to the need for introducing artificial periodicity in the form of the supercell method [1,6,13]. In addition, while studying such systems, plane-wave method codes only take advantage of symmetry groups which are compatible with translational symmetry (such as some of the crystallographic point groups).

Alternatives to the plane-wave approach include the use of atom centered basis functions such as Gaussians and atomic orbitals [14–16], as well as real space discretization approaches such as finite differences and finite elements [17–21]. Atom centered basis functions generally require fewer basis functions per atom compared to plane-waves but these basis sets are usually incomplete and they suffer from basis set superposition errors [2]. Thus, they have issues with systematic convergence. Finite element methods, in contrast, have systematic convergence properties but the quality of the solution as well as the efficiency of the method is heavily dependent on the quality of the mesh as well as the type of element used for the calculation [21]. With the use of higher order finite elements in pseudopotential calculations, simple uniform meshes usually suffice, but mesh-coarsening is generally required for obtaining high efficiency in the vacuum region while dealing with isolated systems [21]. Readers may refer to [3] for a more general review of various basis sets and numerical methods that are in common use today for solution of the Kohn–Sham problem.

From the above discussion, it is quite clear that it would be highly desirable to have methods which are very similar to the plane-wave method but that are designed for systems which are non-periodic. Accordingly, in this work, we develop a scheme that is in many respects an exact analog of the plane-wave method but one which is designed with isolated systems such as clusters and molecules in mind. Ab initio studies of clusters, including various fullerenes and nanostructures, have received and continue to receive a lot of attention in different contexts [22–31]. The methodology developed in this work therefore is likely to be useful for carrying out first principles studies of such systems in a consistent, systematic and efficient manner.

In order to formulate the appropriate basis functions for our method, we first make the observation that plane-waves are eigenfunctions of the periodic Laplacian. Using eigenfunctions of the Laplacian as basis functions leads to numerous advantages, including the fact that the kinetic energy operator is diagonalized in such cases. Accordingly, our method uses eigenfunctions of the Dirichlet Laplacian (i.e., the Laplacian operator with Dirichlet boundary conditions) in a spherical domain as the basis set. Our basis functions are expressible as the product of spherical harmonics with spherical Bessel functions.¹ Let us remark that the use of a spherical (or near spherical) domain for the study of cluster systems has been used earlier in finite difference and finite element methods [20,24]. To the best of our knowledge however, this is the first work to make systematic use of Laplacian eigenfunction expansions in non-periodic domains for use in electronic structure calculations.

Spherical basis functions have been used in earlier work to compute electronic properties of small metallic clusters [32,33] as well as that of C_{60} [34,35]. These basis functions have the distinct advantage that for many cluster systems, the Kohn–Sham eigenstates (molecular orbitals) and their symmetry properties are relatively easy to interpret using the quantum numbers associated with the basis functions² themselves [36]. As explained in [35] the choice of spherical basis functions is usually motivated by the fact that the systems under study are nearly spherical. We show in this work however, that such a constraint on the system under study is unnecessary³ and that a wide variety of cluster systems including ones which are far from being spherical can be studied efficiently with our method. In contrast to our use of spherical Bessel functions, the radial part of the spherical basis functions used in these aforementioned studies has typically been obtained by solving a one dimensional radial eigenvalue problem.

In order to avoid computational complexity, many of the aforementioned studies use a simplified treatment of the electron–nucleus interaction in the form of simple-jellium or pseudo-jellium models [33]. The use of these simplified models however, can often lead to inaccuracies, even while studying simple metallic clusters [37]. In our view, one of the main reasons behind the computational difficulties encountered by these workers is due to the formulation of their methods in which convolution sums are carried out in reciprocal space by means of coupling coefficients (e.g. [33] and [36]). This makes certain operations such as computation of the electron density from the wavefunctions unmanageable beyond relatively small system sizes, unless approximations are used. In addition, these studies also rely on setting up of the full Hamiltonian matrix and then performing diagonalization of this matrix using direct methods, at each self-consistent field iteration cycle. This is quite unlike the approach employed by modern plane-wave codes where a dual representation of various quantities is employed for efficiency purposes and the Fast Fourier Transform (FFT) is used to switch between real and reciprocal spaces [6]. In addition, instead of direct diagonalization methods, most plane-wave codes employ matrix free iterative diagonalization methods to compute the occupied eigenspace of the Hamiltonian [4,7]. We adopt similar strategies in this work and show that this leads to a method where accurate ground state electronic structure calculations for cluster systems containing many hundreds of electrons can be done routinely using our code. In particular, employing widely used, accurate ab initio norm conserving pseudopotentials for modeling the electron–nucleus interaction, without resorting to any form of spherical averaging of the potentials [33], poses no difficulty in our method.

¹ For the purpose of clarity, we emphasize that our basis functions are centered on the origin; i.e., we are dealing with a molecule centered basis set as opposed to an atom centered basis set.

² This allows systems such as super atoms [30,31] to be studied conveniently.

³ This is owing to the fact that we have a complete orthonormal basis.

As mentioned earlier, one of the key aspects of the plane-wave method is the use of three dimensional FFTs to switch between quantities expressed in real and reciprocal spaces. Analogously, we require efficient transforms to switch between quantities expressed on an appropriate grid used to discretize our spherical domain and the expansion coefficients of that quantity when expanded using our basis set (i.e., reciprocal space). Our strategy for obtaining efficient transforms is to use separation of variables into radial and angular parts and handling each of these parts through efficient transforms individually. Specifically, the radial part is computed through Gauss–Jacobi quadrature [38] while the angular part is handled using high performance Spherical Harmonics Transforms (SHTs) [39].

Another key requirement for carrying out accurate Kohn–Sham calculations is the ability to accurately evaluate the electrostatics terms. We accomplish this task here by developing an expansion of the Green's function of the associated Poisson problem in terms of our basis functions. This is followed by computing the convolution of the Green's function with the electronic charge. This is somewhat similar in spirit to the Green's function based methods developed in the context of plane-wave codes [6,40–42] and free-boundary problems [43]. The calculation of the Green's function (in terms of its expansion) can be done ahead of time and does not have to be repeated. As explained later, this means that during the SCF (Self-Consistent Field) cycle, our method allows for the computation of the Hartree potential from the electron density at the cost of a single forward and inverse transform pair. Also, the use of the Green's function ensures that the appropriate decay of the electrostatic potentials is correctly handled, without having to use large computational domains or non-trivial boundary conditions.

Computation of the occupied eigenspace of the discretized Kohn–Sham eigenvalue problem is the most computationally demanding step in a typical self consistent field calculation. Accordingly, a number of strategies have been devised over the years for an efficient solution of this problem through iterative diagonalization methods [4,6,7,44]. We have adopted the Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) algorithm [45] for this purpose in our code. This robust method has been implemented with success in other state-of-the-art Kohn–Sham codes [46,47]. With the aid of a simple diagonal preconditioner (described later) we have found it to work well for a variety of systems. For relatively large system sizes however, especially while running under distributed memory environments, LOBPCG-like methods suffer from the need to repeatedly orthogonalize the computed eigenstates. For dealing with such situations, we have adopted a highly efficient Chebyshev polynomial filtered subspace iteration algorithm [25,48] which avoids explicit diagonalization and minimizes orthonormalizations. For these large scale calculations therefore, LOBPCG is used only in the first SCF step (to generate a good guess for the occupied eigenspace) while Chebyshev polynomial filtering is used exclusively on subsequent SCF steps.

Spectral methods like the plane-wave method and the method presented here⁴ are susceptible to suffer from scalability issues while running under distributed memory environments, since the global nature of the basis functions involved tends to induce communication between the processing elements. To ameliorate this difficulty, we have adopted a two-level parallelization scheme over electronic states as well as physical space, much in the spirit of some large scale plane-wave codes [49]. This effectively reduces global communications to an order of the square root of the number of processes (instead of the total number of processes) and it has resulted in speed-critical portions of our code scaling well up to 512 processing units.

These strategies and methods combine to give an unusually efficient and accurate method, as seen in the examples presented in Section 4. We employ norm conserving *ab initio* pseudopotentials for most of our calculations. We first study the convergence properties of our basis set and demonstrate faster than polynomial rates of convergence (i.e., spectral convergence). Then, starting from light atoms, small molecules and clusters (metallic and non-metallic), we visit various examples involving organic molecules, fullerenes and large face centered cubic (FCC) aluminum clusters. The largest example system considered here contains 1688 aluminum atoms (over 5000 electrons). Timing comparisons reveal that our method outperforms competing finite element and plane-wave method codes in benchmark calculations involving aluminum clusters. Also, comparison against well converged plane-wave method results allows us to show that extremely high accuracies in ground state energy calculations (of the order of 10^{-6} Hartrees per atom) are routinely achievable by our code. By visiting an example that involves the calculation of electrostatic multipole moments, we demonstrate that systematic convergence properties of our basis set allow for easy and accurate calculations of important physical properties.

The rest of the paper is organized as follows. Section 2 describes the formulation of our method while Section 3 describes various implementation aspects. Section 4 presents the example problems solved using our method and compares our results with the literature to assess the efficacy of our method. Section 5 concludes the paper with a summary and comments on future directions.

2. Formulation

We describe the KS-DFT energy functional and the associated Kohn–Sham equations in this section. We outline the key aspects of discretization of the Kohn–Sham equations using our spectral basis and also describe our approach to computation

⁴ More correctly, both the plane-wave method and the method presented here should be referred to as pseudospectral since they rely on interpolatory transforms.

of the various terms that appear in these equations. The atomic unit system with $m_e = 1, e = 1, \hbar = 1, \frac{1}{4\pi\epsilon_0} = 1$, is chosen for the rest of the work, unless otherwise mentioned.

2.1. The Kohn–Sham eigenvalue problem

We consider a system consisting of N_e electrons moving in the fields produced by M nuclei. We assume that the nuclei have charges (z_1, \dots, z_M) and that they are clamped to the positions $(\mathbf{x}_1, \dots, \mathbf{x}_M) \in \mathbb{R}^{3M}$. For the sake of simplicity, we consider a system in which spin polarization effects are absent and we consider N_e to be even. The extension of the present work to study spin-polarized systems is straight-forward.⁵

In line with the Born–Oppenheimer approximation [2], the Kohn–Sham model [2,50] computes the total energy of this system (denoted here as $E_{N_e,M}^{\text{KS}}$) at absolute zero, by splitting it into an electronic part (denoted here as $\mathcal{E}_{N_e,M}^{\text{KS}}$) and a nucleus–nucleus interaction part:

$$E_{N_e,M}^{\text{KS}} = \mathcal{E}_{N_e,M}^{\text{KS}} + \sum_{1 \leq k < l \leq M} \frac{z_k z_l}{|\mathbf{x}_k - \mathbf{x}_l|}. \quad (1)$$

The electronic part of the energy in the Kohn–Sham model is computed in terms of orbitals; i.e., an $N_e/2$ tuple of complex valued scalar fields $\{\phi_i\}_{i=1}^{N_e/2}$, as follows⁶:

$$\mathcal{E}_{N_e}^{\text{KS}} = \inf_{\substack{\phi_i \in H^1(\mathbb{R}^3), \\ \langle \phi_i, \phi_j \rangle_{L^2(\mathbb{R}^3)} = \delta_{ij}}} \left\{ \frac{1}{2} \sum_{i=1}^{N_e/2} \int_{\mathbb{R}^3} f_i |\nabla \phi_i|^2 + \int_{\mathbb{R}^3} \rho V_{\text{nu}} + \frac{1}{2} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \frac{\rho(\mathbf{x}) \rho(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|} d\mathbf{x} d\mathbf{y} + E_{\text{xc}}(\rho) \right\} \quad (2)$$

$$\text{with the electron density, } \rho(\mathbf{x}) = 2 \sum_{i=1}^{N_e/2} f_i |\phi_i(\mathbf{x})|^2. \quad (3)$$

The scalars f_i denote orbital occupations and have values $0 \leq f_i \leq 1$. These values need to be specified a priori or they can be computed as part of the solution (using thermalization for instance, see Section 3.4.1). The factor of two in Eq. (3) above, is due to the assumption of dealing with a spin-unpolarized system, as a consequence of which, each orbital is doubly occupied. The orthonormalization constraint on the orbitals implies that the electron density ρ satisfies the normalization condition:

$$\int_{\mathbb{R}^3} \rho = N_e. \quad (4)$$

The first term in Eq. (2) (involving the gradient of the orbitals) models the kinetic energy of the electrons. The second term models the interaction of the nuclei with the electrons. The nuclear potential appearing in that term is given as:

$$V_{\text{nu}}(\mathbf{x}) = - \sum_{k=1}^M \frac{z_k}{|\mathbf{x} - \mathbf{x}_k|}. \quad (5)$$

In practice, the Coulombic singularities present in Eq. (5) cause problems with efficient numerical solution of the equations and spectral methods (like the plane-wave method and the present one) are particularly affected due to appearance of Gibbs phenomenon [51]. The pseudopotential approximation⁷ allows these issues to be mitigated by smoothening out these singularities [1,2,52]. The bulk of the present work is devoted to pseudopotential calculations.

The third term in Eq. (2) represents the mutual electrostatic repulsion of the electrons (Hartree energy). Finally, the term $E_{\text{xc}}(\rho)$ is the exchange correlation energy. We adopt here the widely used Local Density Approximation (LDA) [50,53] of this term by using the parametrization presented in [54,55]. An extension of our method to density gradient corrected functionals [56] poses no particular difficulty.

The Euler–Lagrange equations of the minimization problem (Eq. (2)) are the celebrated Kohn–Sham equations [50], which, with the definition of the electron density introduced in Eq. (3) can be written as follows:

$$K(\rho) \phi_i = \lambda_i \phi_i; \langle \phi_i, \phi_j \rangle_{L^2(\mathbb{R}^3)} = \delta_{ij}, \quad (6)$$

$$\text{with } K(\rho) = -\frac{1}{2} \Delta + V_{\text{nu}} + \int_{\mathbb{R}^3} \frac{\rho(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|} d\mathbf{y} + V_{\text{xc}}(\rho), \quad (7)$$

$$\text{where } V_{\text{xc}}(\rho) = \frac{\partial E_{\text{xc}}(\rho)}{\partial \rho}. \quad (8)$$

⁵ Some of our example calculations presented in Section 4 do use spin-polarization.

⁶ In the equations that follow, $L^2(\mathbb{R}^3)$ is used to denote the usual space of square integrable functions on \mathbb{R}^3 while $H^1(\mathbb{R}^3)$ denotes the Sobolev space of functions in $L^2(\mathbb{R}^3)$ whose first order weak derivatives also lie in $L^2(\mathbb{R}^3)$.

⁷ In particular, ab initio pseudopotentials provide a well defined recipe of carrying out this smoothening procedure such that the energy and length scales of the problem are dictated by the chemically relevant electronic states. See [52] for more details.

The λ_i that appear in Eq. (6) are the Lagrange multipliers of the orthonormality constraints on the orbitals. They are taken to be the lowest $N_e/2$ eigenvalues of the Kohn–Sham operator $K(\rho)$.

The usual method of solution of the Kohn–Sham equations is by a Self-Consistent Field (SCF) approach [1,2,50]. In practice, a variety of mixing schemes are employed to accelerate convergence of the SCF iterations [1,57].

2.2. Problem setup and discretization

Let \mathcal{B}_R denote the sphere of radius R centered at the origin. For the purpose of this work, we will restrict the physical domain to \mathcal{B}_R and the cluster/molecular system of interest will be embedded within this spherical region. We will apply Dirichlet boundary conditions to the electron density on the surface of the sphere in accordance with the well-known spatial exponential decay of the electron density [58,59]. The relation between the electron density and the wavefunctions (Eq. (3)) automatically implies that the Dirichlet boundary conditions apply to the wavefunctions⁸ as well. We do not enforce specific boundary conditions on the various potential terms. These are applied implicitly based on the method of computation: in case of the Hartree term for instance, our method of calculation automatically ensures that the right kind of decay of that term is obtained (Section 2.3.1).

2.2.1. Basis set

The particular choice of a spherical domain allows for the Laplacian eigenfunctions in this domain to be represented analytically in spherical coordinates.⁹ Specifically, we consider the $L^2(\mathcal{B}_R)$ orthonormal eigenfunctions of the Laplacian operator in the spherical domain and we impose Dirichlet boundary conditions on the surface of the domain. In this setup, a simple separation of variables calculation shows that the eigenfunctions of the Laplacian which are regular at the origin are expressible in terms of spherical Bessel functions of the first kind and spherical harmonics [see e.g. [61], for details of this calculation]. Letting $(l, m, n) \in \Gamma_\infty$ with:

$$\Gamma_\infty = \left\{ (l, m, n) : l \in \{0, 1, \dots\}, m \in \{-l, \dots, l\}, n \in \{0, 1, \dots\} \right\}, \quad (9)$$

the Laplacian eigenfunctions take the form:

$$F_{l,m,n}(r, \vartheta, \varphi) = \mathcal{R}_{l,n}(r) \mathcal{Y}_l^m(\vartheta, \varphi), \quad (10)$$

with the radial part being the spherical Bessel functions of the first kind:

$$\mathcal{R}_{l,n}(r) = \frac{1}{R J_{l+\frac{3}{2}}(b_{l+\frac{1}{2}}^n)} \sqrt{\frac{2}{r}} J_{l+\frac{1}{2}}\left(\frac{b_{l+\frac{1}{2}}^n}{R} r\right), \quad (11)$$

and the angular part being the spherical harmonics:

$$\mathcal{Y}_l^m(\vartheta, \varphi) = c_{l,m} \mathcal{P}_l^m(\cos \vartheta) e^{im\varphi}, \text{ with } c_{l,m} = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}}. \quad (12)$$

In Eq. (11), $J_{l+\frac{1}{2}}(\cdot)$ denotes the (ordinary) Bessel function of the first kind of order $(l + \frac{1}{2})$, while $b_{l+\frac{1}{2}}^n$ denotes its $(n+1)$ th root. Thus, $\mathcal{R}_{l,n}(r)$ attains a value of zero $(n+1)$ times in the interval $[0, R]$. In Eq. (12), $\mathcal{P}_l^m(\cdot)$ denotes the associated Legendre polynomial of degree l and order m . The eigenvalue associated with the eigenfunction $F_{l,m,n}$ is given by:

$$\Lambda_{l,m,n} = \left(\frac{b_{l+\frac{1}{2}}^n}{R} \right)^2. \quad (13)$$

Since the Laplacian is a self-adjoint operator with a compact resolvent, the infinite collection of eigenfunctions $\mathcal{E}_{\Gamma_\infty} = \{F_{l,m,n} : (l, m, n) \in \Gamma_\infty\}$ form an orthonormal basis of $L^2(\mathcal{B}_R)$ [62,63]. Further, elliptic-regularity results [62] imply that each basis function $F_{l,m,n}$ is smooth. We now choose a finite subset of $\mathcal{E}_{\Gamma_\infty}$ as our basis set.

We fix $\mathcal{L}, \mathcal{N} \in \mathbb{N}$ (henceforth referred to as the angular and radial cutoff, respectively), and form $\Gamma \subset \Gamma_\infty$ by restricting^{10,11} $l \in \{0, \dots, \mathcal{L}-1\}$ and $n \in \{0, \dots, \mathcal{N}-1\}$. Given any function $f \in L^2(\mathcal{B}_R)$, for the purpose of numerical discretization, we approximate it using the functions in $\mathcal{E}_\Gamma = \{F_{l,m,n} : (l, m, n) \in \Gamma\}$ as:

⁸ Application of Dirichlet boundary conditions to the Kohn–Sham wavefunctions has been considered earlier in the context of real-space methods [17,20,60].

⁹ In our notation for spherical coordinates, we denote $r \in [0, R]$ as the radial coordinate, $\vartheta \in [0, \pi]$ as the polar angle and $\varphi \in [0, 2\pi]$ as the azimuthal angle. The Cartesian coordinates (x, y, z) are obtained as $x = r \sin \vartheta \cos \varphi$, $y = r \sin \vartheta \sin \varphi$, $z = r \cos \vartheta$.

¹⁰ For each l , m is allowed to vary in $\{-l, \dots, l\}$ as before.

¹¹ For the purpose of this article, we have used a uniform basis set in which the maximum value of $n = (\mathcal{N} - 1)$ for every l . We are aware however, that a non-uniform basis set in which the maximum value of n is allowed to vary with l allows more flexibility and can lead to a number of desirable effects. For instance, this setting allows (as in plane-wave codes), the introduction of a kinetic energy cut-off in which the basis set only includes spherical waves

$$f = \sum_{(l,m,n) \in \Gamma} \hat{f}_{l,m,n} F_{l,m,n}. \quad (14)$$

We may observe that the span of the functions in $\mathcal{E}_{\tilde{\Gamma}}$ form a linear subspace of $L^2(\mathcal{B}_R)$ of dimension $d = \mathcal{L}^2 \mathcal{N}$. The expansion coefficients can be obtained from orthonormality of the basis functions by:

$$\hat{f}_{l,m,n} = \langle f, F_{l,m,n} \rangle_{L^2(\mathcal{B}_R)} = \int_{\mathcal{B}_R} f \overline{F_{l,m,n}}, \quad (15)$$

and the collection of expansion coefficients $\{\hat{f}_{l,m,n} : (l,m,n) \in \Gamma\}$ will often be interpreted interchangeably with vectors in \mathbb{C}^d . If the function f is real valued, as it is for example, in case of the electron density, the expansion coefficients obey the additional condition¹² $\hat{f}_{l,-m,n} = \overline{\hat{f}_{l,m,n}}$.

2.2.2. Basis transforms

In order to perform the quadratures required for evaluation of the expansion coefficients via Eq. (15) we introduce a discretization of the domain $B \subset \mathcal{B}_R$. Akin to the terminology used in the plane-wave literature, we will often refer to the representation of a given function in terms of its expansion coefficients as the *reciprocal space representation* while the representation of the same function on the grid points in B will be referred to as the *real space representation*. The operations that allow us to switch between these two representations will be referred to as *basis transforms*.

The specific choice of the grid points is made as follows. Let N_r , N_ϑ and N_φ denote the number of discretization points in the radial, polar and azimuthal directions respectively. These quantities are dependent on the radial and angular cutoffs and are chosen in accordance with constraints of the sampling theorem. We discretize the unit sphere by choosing N_ϑ Gauss quadrature points in $\cos(\vartheta)$ over the interval $[-1, 1]$ and N_φ equally spaced points in φ over the interval $[0, 2\pi]$. In the radial direction, we choose N_r Gauss–Jacobi quadrature nodes [38] associated with the quadrature weight of r^2 over the interval $[0, R]$. The set B is now taken to be a Cartesian product of the radial quadrature points and the unit sphere discretization points. This allows a separation of variables in the angular and radial directions while carrying out the basis transforms, thereby reducing computational complexity.

Given the real space representation $\tilde{f} : B \rightarrow \mathbb{C}$, we obtain the reciprocal space representation by first computing spherical harmonic transforms holding the radial variable fixed:

$$\begin{aligned} A(r; l, m) &= \int_0^{2\pi} \int_0^\pi f(r; \vartheta, \varphi) \overline{\mathcal{Y}_l^m(\vartheta, \varphi)} \sin(\vartheta) d\vartheta d\varphi, \\ &= c_{l,m} \int_{-1}^1 \mathcal{P}_l^m(t) \left[\int_0^{2\pi} f(r; \cos^{-1}(t), \varphi) e^{-im\varphi} d\varphi \right] dt, \end{aligned} \quad (16)$$

and then performing radial quadratures using the quadrature nodes $\{r_{k_r}\}_{k_r=1}^{N_r}$ and corresponding weights $\{w_{k_r}\}_{k_r=1}^{N_r}$:

$$\hat{f}_{l,m,n} = \int_0^R A(r; l, m) \mathcal{R}_{l,n}(r) r^2 dr \approx \sum_{k_r=1}^{N_r} w_{k_r} A(r_{k_r}; l, m) \mathcal{R}_{l,n}(r_{k_r}). \quad (17)$$

The spherical harmonic transform as expressed in Eq. (16) itself consists of two steps: first holding ϑ fixed, the Fast Fourier Transform (FFT) [65] is used to evaluate the inner integral involving φ . Subsequently, a quadrature in $t = \cos(\vartheta)$ is carried out on the result to evaluate the outer integral.

Similarly, given the reciprocal space representation $\hat{f} : \Gamma \rightarrow \mathbb{C}$, the inverse transform can be carried out by first computing:

$$G(l, m; r_{k_r}) = \sum_{n=0}^{\mathcal{N}-1} \hat{f}_{l,m,n} \mathcal{R}_{l,n}(r_{k_r}), \quad (18)$$

while holding l and m fixed and then, for each radial grid node r_{k_r} , performing inverse spherical harmonics transforms (using inverse FFTs and dot products as in [39]).

whose kinetic energies lie below a specified threshold. This results in a basis set which is optimal in the sense of the Sobolev H^1 norm. Also, we are aware that a uniform basis set can sometimes face a loss of approximation power for large radial distances, although we have not encountered any serious issues from this effect in this work. A non-uniform basis set however, can be made to correct for this issue automatically. More details of this methodology is the scope of future work.

¹² If the Condon–Shortley phase [64] is included, this becomes $\hat{f}_{l,-m,n} = (-1)^m \overline{\hat{f}_{l,m,n}}$. We do not make use of the Condon–Shortley phase in this work.

The basis transforms as described above, have a time complexity of $O(\mathcal{L}^3\mathcal{N} + \mathcal{L}^2\mathcal{N}^2)$ in terms of the angular and radial cutoffs.^{13,14} As far as practical implementation is concerned, the use of Gauss quadrature points as well as various numerical and implementation level optimizations [see [39], for example] can be used to ensure that the prefactor for this asymptotic estimate is rather low. This allowed us to carry out basis transforms routinely and efficiently even with basis sets containing millions of basis functions in our code.

2.2.3. Setup of matrix eigenvalue problem

Within the self-consistent field iterations, the governing equations (Eqs. (6)–(8)) posed in the spherical domain take the form of the following linearized eigenvalue problem with an effective potential V^{eff} :

$$\left(-\frac{1}{2}\Delta + V^{\text{eff}}\right)\phi_i = \lambda_i \phi_i \text{ for } i = 1, \dots, N_e/2, \quad (19)$$

$$\phi_i = 0 \text{ on } \partial\mathcal{B}_R, \quad (20)$$

and the effective potential at a point $\mathbf{x} \in \mathcal{B}_R$ is given as:

$$V^{\text{eff}}(\mathbf{x}) = V_{\text{xc}}(\rho(\mathbf{x})) + \int_{\mathcal{B}_R} \frac{\rho(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|} d\mathbf{y} + V_{\text{nu}}(\mathbf{x}). \quad (21)$$

We choose to ignore any non-local contributions to the ionic pseudopotentials at this point. The specific treatment of these non-local terms is discussed later in Section 2.3.2.

To discretize Eq. (19) we set:

$$\phi_i = \sum_{(l,m,n) \in \Gamma} \hat{\phi}_{l,m,n}^i F_{l,m,n}, \quad (22)$$

noting that this ensures that the Dirichlet boundary conditions on the wavefunctions are satisfied automatically. This gives us:

$$\frac{1}{2} \sum_{\Gamma} \hat{\phi}_{l,m,n}^i \Delta_{l,m,n} F_{l,m,n} + V^{\text{eff}} \sum_{\Gamma} \hat{\phi}_{l,m,n}^i F_{l,m,n} = \lambda_i \sum_{\Gamma} \hat{\phi}_{l,m,n}^i F_{l,m,n}. \quad (23)$$

Now, if the expansion coefficients of V^{eff} are known as $\{\hat{V}_{\tilde{l},\tilde{m},\tilde{n}}^{\text{eff}} : (\tilde{l}, \tilde{m}, \tilde{n}) \in \Gamma\}$, we may substitute the expansion of V^{eff} into Eq. (23) to get:

$$\begin{aligned} & \frac{1}{2} \sum_{\Gamma} \hat{\phi}_{l,m,n}^i \Delta_{l,m,n} F_{l,m,n} + \sum_{\Gamma} \sum_{\Gamma} \hat{\phi}_{l,m,n}^i \hat{V}_{\tilde{l},\tilde{m},\tilde{n}}^{\text{eff}} F_{\tilde{l},\tilde{m},\tilde{n}} F_{l,m,n} \\ & = \lambda_i \sum_{\Gamma} \hat{\phi}_{l,m,n}^i F_{l,m,n}. \end{aligned} \quad (24)$$

We now take the inner product of this equation with $F_{l',m',n'}$ and use orthonormality of the basis functions to obtain the following system of linear equations for $\hat{\phi}_{l',m',n'}^i$, with $(l', m', n') \in \Gamma$:

$$\frac{1}{2} \Delta_{l',m',n'} \hat{\phi}_{l',m',n'}^i + \sum_{\Gamma} \sum_{\Gamma} \mathcal{W}_{(l,m,n), (\tilde{l},\tilde{m},\tilde{n})}^{(l',m',n')} \hat{V}_{\tilde{l},\tilde{m},\tilde{n}}^{\text{eff}} \hat{\phi}_{l,m,n}^i = \lambda_i \hat{\phi}_{l',m',n'}^i, \quad (25)$$

where $\mathcal{W}_{(l,m,n), (\tilde{l},\tilde{m},\tilde{n})}^{(l',m',n')}$ denote the coupling coefficients of the basis set, i.e.,

$$\mathcal{W}_{(l,m,n), (\tilde{l},\tilde{m},\tilde{n})}^{(l',m',n')} = \left\langle F_{\tilde{l},\tilde{m},\tilde{n}} F_{l,m,n}, F_{l',m',n'} \right\rangle_{L^2(\mathcal{B}_R)}. \quad (26)$$

It is possible to express these coupling coefficients in terms of Wigner 3-j symbols [68] and the integral of the product of three spherical basis functions taken together [61]. Such an expression allows us to see that the coupling coefficients are non-zero only when $|l - \tilde{l}| \leq l' \leq l + \tilde{l}$, $m + m' + \tilde{m} = 0$ and $l + l' + \tilde{l}$ is odd.

To recognize the finite dimensional linear eigenvalue problem expressed in Eq. (25), we may introduce an indexing map $\mathcal{I} : \Gamma \rightarrow \{1, 2, \dots, d\}$ and let \mathcal{J} denote its inverse.¹⁵ We rewrite Eq. (25) using the map \mathcal{J} to obtain a matrix problem of the form:

¹³ A naive implementation of the transforms, that is, one that does not employ the separation of variables structure, would have a time complexity of $O(\mathcal{L}^4\mathcal{N}^2)$ in terms of the angular and radial cutoffs.

¹⁴ Using more sophisticated techniques for carrying out the associated Legendre polynomial transforms [66,67], this can be reduced to $O(\mathcal{L}^2(\log \mathcal{L})^2\mathcal{N} + \mathcal{L}^2\mathcal{N}^2)$.

¹⁵ A simple indexing map is, for instance, $(l, m, n) \mapsto (l^2 + l + m) * \mathcal{N} + (n + 1)$.

$$\mathbf{H}\mathbf{X} = \mathbf{X}\mathfrak{D}, \quad (27)$$

where $\mathbf{H} \in \mathbb{C}^{d \times d}$, $\mathbf{X} \in \mathbb{C}^{d \times (N_e/2)}$ and $\mathfrak{D} \in \mathbb{R}^{(N_e/2) \times (N_e/2)}$. We note that the matrix \mathbf{H} is Hermitian, while the matrix \mathbf{X} is unitary in the sense that $\mathbf{X}^\dagger \mathbf{X}$ is an identity matrix of dimension $(N_e/2) \times (N_e/2)$. Denoting $\delta_{\alpha,\beta}$ as the Kronecker delta, we see that matrices \mathbf{H} , \mathbf{X} and \mathfrak{D} have entries of the following form (in terms of the indexing map):

$$\mathbf{H}_{\alpha,\beta} = \frac{1}{2} \delta_{\alpha,\beta} \Lambda_{\mathcal{J}(\alpha)} + \sum_{(\tilde{l}, \tilde{m}, \tilde{n}) \in \Gamma} \hat{V}_{\tilde{l}, \tilde{m}, \tilde{n}}^{\text{eff}} \mathcal{W}_{\mathcal{J}(\beta), (\tilde{l}, \tilde{m}, \tilde{n})}^{\mathcal{J}(\alpha)}, \quad (28)$$

$$\mathbf{X}_{\alpha,\beta} = \hat{\phi}_{\mathcal{J}(\alpha)}^\beta \text{ and } \mathfrak{D}_{\alpha,\beta} = \delta_{\alpha,\beta} \lambda_{\mathcal{J}(\beta)}, \quad (29)$$

with α, β varying within relevant matrix dimensions.

As we mentioned earlier in this article, setting up of the matrix eigenvalue problem followed by direct diagonalization are both expensive operations, although this approach seems to have been adopted by earlier work involving spherical basis functions [see e.g. [36]]. From Eq. (28), for instance, we can see that the matrix \mathbf{H} is dense and therefore, the asymptotic computational complexity of the matrix setup is of cubic order in the total number of basis functions. Direct diagonalization of the Hamiltonian, even by the most efficient algorithms available today [see e.g. [69]], will have the same cubic computational complexity in the number of basis functions due to the necessity of reducing the matrix to tridiagonal form. In addition, the memory storage requirement of the full Hamiltonian matrix scales as the square of the number of basis functions and therefore this becomes an additional constraint while trying to deal with even moderate sized systems.

2.2.4. Setup of matrix–vector products

To avoid the computational difficulties associated with direct diagonalization, we choose to employ matrix-free iterative methods for computing the occupied eigenspace of the Hamiltonian matrix [see [70], for a detailed discussion of this class of methods]. As the name suggests, these methods do not need access to the individual matrix entries but only require matrix vector products to be specified. To see how the product of a given vector with the Hamiltonian matrix may be calculated efficiently, without explicit involvement of the coupling coefficients, we proceed as follows:

Let $\mathbf{Y} \in \mathbb{C}^d$ be a given vector and let $\mathbf{Z} \in \mathbb{C}^d$ be the result of the matrix vector product, that is, $\mathbf{Z} = \mathbf{H}\mathbf{Y}$. In terms of components we have:

$$\begin{aligned} \mathbf{Z}_\alpha &= \sum_{\beta=1}^d \mathbf{H}_{\alpha,\beta} \mathbf{Y}_\beta = \sum_{\beta=1}^d \left(\frac{1}{2} \delta_{\alpha,\beta} \Lambda_{\mathcal{J}(\alpha)} + \sum_{(\tilde{l}, \tilde{m}, \tilde{n}) \in \Gamma} \hat{V}_{\tilde{l}, \tilde{m}, \tilde{n}}^{\text{eff}} \mathcal{W}_{\mathcal{J}(\beta), (\tilde{l}, \tilde{m}, \tilde{n})}^{\mathcal{J}(\alpha)} \right) \mathbf{Y}_\beta \\ &= \frac{1}{2} \Lambda_{\mathcal{J}(\alpha)} \mathbf{Y}_\alpha + \sum_{\beta=1}^d \left(\sum_{(\tilde{l}, \tilde{m}, \tilde{n}) \in \Gamma} \hat{V}_{\tilde{l}, \tilde{m}, \tilde{n}}^{\text{eff}} \mathcal{W}_{\mathcal{J}(\beta), (\tilde{l}, \tilde{m}, \tilde{n})}^{\mathcal{J}(\alpha)} \right) \mathbf{Y}_\beta. \end{aligned} \quad (30)$$

The second term, by making use of Eq. (26) and the linearity of the inner product, can be written as:

$$\begin{aligned} &= \sum_{\beta=1}^d \sum_{(\tilde{l}, \tilde{m}, \tilde{n}) \in \Gamma} \hat{V}_{\tilde{l}, \tilde{m}, \tilde{n}}^{\text{eff}} \mathbf{Y}_\beta \left\langle F_{\tilde{l}, \tilde{m}, \tilde{n}}^{\mathcal{J}(\beta)}, F_{\mathcal{J}(\alpha)} \right\rangle_{L^2(\mathcal{B}_R)} \\ &= \left\langle \left(\sum_{(\tilde{l}, \tilde{m}, \tilde{n}) \in \Gamma} \hat{V}_{\tilde{l}, \tilde{m}, \tilde{n}}^{\text{eff}} F_{\tilde{l}, \tilde{m}, \tilde{n}}^{\mathcal{J}(\beta)} \right) \left(\sum_{\beta=1}^d \mathbf{Y}_\beta F_{\mathcal{J}(\beta)} \right), F_{\mathcal{J}(\alpha)} \right\rangle_{L^2(\mathcal{B}_R)}. \end{aligned}$$

We recognize the first term in parentheses as the expansion of the effective potential in our basis set, and therefore, the final expression for \mathbf{Z}_α becomes:

$$\mathbf{Z}_\alpha = \frac{1}{2} \Lambda_{\mathcal{J}(\alpha)} \mathbf{Y}_\alpha + \left\langle V^{\text{eff}} \left(\sum_{\beta=1}^d \mathbf{Y}_\beta F_{\mathcal{J}(\beta)} \right), F_{\mathcal{J}(\alpha)} \right\rangle_{L^2(\mathcal{B}_R)}. \quad (31)$$

The above equation suggests that the computation of the Hamiltonian times vector product is to be carried out in two stages and the results from these stages should be summed. First, the action of the kinetic energy operator is to be carried out in reciprocal space because of the diagonal structure of that operator in that space. In the second stage, the action of the operator expressing the action of the effective potential is to be computed. This operator however, is diagonal in real space. Thus, given the vector \mathbf{Y} , we imagine its components $\{\mathbf{Y}_\alpha\}_{\alpha=1}^d$ to represent expansion coefficients and we perform an inverse basis transform to obtain a function Y defined on the gridpoints in B . We then perform a pointwise multiplication of Y with the effective potential (also defined over B) and we finally compute a forward basis transform of the product $(V^{\text{eff}} \cdot Y)$ to obtain the result of the second stage.

The principal computational cost of the process described above arises from a pair of basis transforms and therefore the associated time and space complexities are $O(\mathcal{L}^3 \mathcal{N} + \mathcal{L}^2 \mathcal{N}^2)$ and $O(\mathcal{L}^2 \mathcal{N})$ respectively. In contrast, a direct matrix vector product, once the Hamiltonian matrix has been set up, would involve $O(\mathcal{L}^4 \mathcal{N}^2)$ complexity both in memory and speed.

Note that the discussion above does not take into account the role played by non-local pseudopotentials. When such non-local terms are present, the Hamiltonian as described above, has an additional projection operator term that acts on the given vector \mathbf{Y} . The action of this additional operator on the given vector can be directly computed as a dense linear algebra operation.

2.2.5. Computation of the electron density

Using the expansion of the wavefunctions (Eq. (22)) as well as the expression in Eq. (3), we see that the electron density admits expansion coefficients of the form:

$$\tau_{l',m',n'} = 2 \sum_{j=1}^{N_e/2} \sum_{\Gamma} \sum_{\Gamma} \mathcal{W}_{(l,m,n),(\tilde{l},\tilde{m},\tilde{n})}^{(l',m',n')} \hat{\phi}_{l,\tilde{m},\tilde{n}}^j \overline{\hat{\phi}_{l,-m,n}^j}. \quad (32)$$

Two comments are in order at this stage. First, since the coupling coefficients are non-zero only when $|l - \tilde{l}| \leq l' \leq l + \tilde{l}$ and we have $0 \leq l, \tilde{l} \leq (\mathcal{L} - 1)$, we see that $\tau_{l',m',n'}$ may have non-zero values for all l' satisfying $0 \leq l' \leq 2(\mathcal{L} - 1)$. Thus, due to the quadratic non-linearity in Eq. (3), the electron density needs to be represented using a basis set that is larger than the one used to represent the wavefunctions. A similar situation also arises in the context of the plane-wave method, where sometimes, compared to the wavefunctions, the electron density expansion employs a larger energy cutoff [71]. Often however, plane wave codes allow the so-called *dualing approximation* to be engaged, as a result of which, the electron density is expanded using the same basis set as the wavefunctions [71]. In the same vein, our implementation allows a larger basis set (as well as a correspondingly denser real space grid) for the electron density to be employed based on user choice.

The second comment is that, while Eq. (32) illustrates the basis requirements while computing the electron density, a direct application of this expression to compute the electron density expansion coefficients is prohibitively expensive. The time complexity of the operations involved in that expression, in terms of the basis set size and the number of electrons involved, is $O(N_e d^3)$. Instead, starting from the expansion coefficients of the wavefunctions, we may compute the real space representations of the wavefunctions using inverse basis transforms. We may then use Eq. (3) to compute the electron density at the grid points in B and finally apply a forward basis transform to obtain the required expansion coefficients of the density. This method results in the reduced time complexity of $O(N_e(\mathcal{L}^3 \mathcal{N} + \mathcal{L}^2 \mathcal{N}^2))$ and in practice, it turns out to be much more efficient.

The methods for computation of the electron density (as described above) and matrix vector products (described earlier in Section 2.2.4) are both based on the general idea of evaluating convolution sums through efficient transforms [72]. While this technique seems to have been used quite commonly in fluid dynamics simulations both for spherical and periodic domains [73,74], its application to Kohn–Sham density functional theory seems to have been only in the context of the plane-wave method [6] and spherical basis function based methods seem to have ignored it [see e.g. [33,36]]. However, the two different methods of evaluation of the convolution sums (i.e., direct application of Eq. (32) vs. the transform method described above) can lead to very large differences in computation times. To illustrate this point, we carried out computation of the electron density coefficients from randomly generated families of single electronic states using both methods. While using the direct method, we used hash functions [75] for expedited computation of the coupling coefficients¹⁶ that appear in Eq. (32). For both the direct method as well as the transform method, we varied the angular and radial cutoffs independently in order to directly observe the computational scalings of the density calculation routine.¹⁷ Fig. 1 shows that the transform routine has far better scaling behavior than the direct routine for both discretization parameters. In practice, the computation run times for both these routines can differ by many orders of magnitude¹⁸ as Table 1 shows. The angular and radial cutoffs that were chosen for the comparison in that table are very typical for obtaining acceptable levels of convergence in the total energies of small sized cluster systems containing a few light metallic atoms.

2.3. Computation of the potentials

We describe the computation of the various potential terms in this section with particular attention to the Hartree and pseudopotential terms. The exchange correlation terms (within the Local Density Approximation) are evaluated directly using the real space representation of the electron density and deserve no further comments.

2.3.1. Computation of the Hartree potential

The Hartree potential at a point $\mathbf{x} \in \mathcal{B}_R$ is given as:

$$V_H(\mathbf{x}) = \int_{\mathcal{B}_R} \frac{\rho(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|} d\mathbf{y}. \quad (33)$$

¹⁶ Storage of all the coupling coefficients becomes memory intensive quite quickly.

¹⁷ For both routines, we investigated the angular cutoff range $10 \leq \mathcal{L} \leq 60$ and the radial cutoff range $10 \leq \mathcal{N} \leq 60$.

¹⁸ Similar conclusions can be drawn about the matrix–vector product computation routines described earlier in Section 2.2.4.

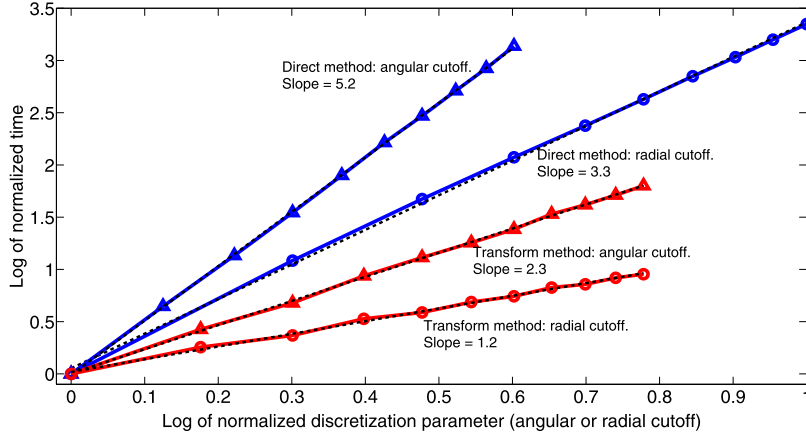


Fig. 1. Observed scaling of the density computation routine using the direct method (blue curves) and the transform method (red curves) with increasing discretization parameter. Both the angular cutoff variation (curves with triangles) as well as the radial cutoff variation (curves with circles) are shown. Slopes of the fitted lines (black) indicate scaling behavior. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1

Relative timings of the density computation routines for $\mathcal{L} = 30$, $\mathcal{N} = 30$ normalized using the transform method timing.

Transform method	Direct method: without hash functions	Direct method: with hash functions
1.00	2.18×10^6	1.71×10^6

One of the most popular approaches to solving this equation is by employing Poisson solvers [17,20,76]. Our approach to the computation of V_H is to directly deal with Eq. (33) by exploiting the so-called Laplace expansion [77] of the Green's kernel:

$$\frac{1}{|\mathbf{x} - \mathbf{y}|} = \sum_{l=0}^{\infty} \frac{4\pi}{2l+1} \sum_{m=-l}^{m=l} \frac{r_{<}^l}{r_{>}^{l+1}} \overline{\mathcal{Y}_l^m(\vartheta_{\mathbf{x}}, \varphi_{\mathbf{x}})} \mathcal{Y}_l^m(\vartheta_{\mathbf{y}}, \varphi_{\mathbf{y}}). \quad (34)$$

In the equation above, $r_{<} = \min(r_{\mathbf{x}}, r_{\mathbf{y}})$, $r_{>} = \max(r_{\mathbf{x}}, r_{\mathbf{y}})$ and $(r_{\mathbf{x}}, \vartheta_{\mathbf{x}}, \varphi_{\mathbf{x}})$ and $(r_{\mathbf{y}}, \vartheta_{\mathbf{y}}, \varphi_{\mathbf{y}})$ denote \mathbf{x} and \mathbf{y} in spherical coordinates respectively. For a typical point $\mathbf{y} \in \mathcal{B}_R$, the electron density ρ is available through a basis expansion as:

$$\rho(r_{\mathbf{y}}, \vartheta_{\mathbf{y}}, \varphi_{\mathbf{y}}) = \sum_{\hat{\Gamma}} \tau_{\hat{l}, \hat{m}, \hat{n}} \mathcal{R}_{\hat{l}, \hat{n}}(r_{\mathbf{y}}) \mathcal{Y}_{\hat{l}}^{\hat{m}}(\vartheta_{\mathbf{y}}, \varphi_{\mathbf{y}}), \quad (35)$$

with $\hat{\Gamma}$ denoting the same basis set as Γ , or a larger one, depending on whether the dualing approximation has been used or not. Now, if $d\check{\mathbf{y}}$ denotes the volume element in the sphere \mathcal{B}_R , that is, $d\check{\mathbf{y}} = r_{\mathbf{y}}^2 \sin \vartheta_{\mathbf{y}} dr_{\mathbf{y}} d\vartheta_{\mathbf{y}} d\varphi_{\mathbf{y}}$, then substituting Eqs. (34) and (35) in Eq. (33) and using orthonormality of the spherical harmonics, we get:

$$\begin{aligned} V_H(r_{\mathbf{x}}, \vartheta_{\mathbf{x}}, \varphi_{\mathbf{x}}) &= \sum_{l=0}^{\infty} \frac{4\pi}{2l+1} \sum_{m=-l}^{m=l} \sum_{\hat{\Gamma}} \tau_{\hat{l}, \hat{m}, \hat{n}} \mathcal{Y}_l^m(\vartheta_{\mathbf{x}}, \varphi_{\mathbf{x}}) \\ &\quad \times \int_{\mathcal{B}_R} \frac{r_{<}^l}{r_{>}^{l+1}} \overline{\mathcal{Y}_l^m(\vartheta_{\mathbf{y}}, \varphi_{\mathbf{y}})} \mathcal{R}_{\hat{l}, \hat{n}}(r_{\mathbf{y}}) \mathcal{Y}_{\hat{l}}^{\hat{m}}(\vartheta_{\mathbf{y}}, \varphi_{\mathbf{y}}) d\check{\mathbf{y}}, \\ &= \sum_{l=0}^{\infty} \frac{4\pi}{2l+1} \sum_{m=-l}^{m=l} \sum_{\hat{\Gamma}} \tau_{\hat{l}, \hat{m}, \hat{n}} \mathcal{Y}_l^m(\vartheta_{\mathbf{x}}, \varphi_{\mathbf{x}}) \delta_{\hat{l}, l} \delta_{\hat{m}, m} \int_{r_{\mathbf{y}}=0}^{r_{\mathbf{y}}=R} \frac{r_{<}^l}{r_{>}^{l+1}} \mathcal{R}_{\hat{l}, \hat{n}}(r_{\mathbf{y}}) r_{\mathbf{y}}^2 dr_{\mathbf{y}}, \end{aligned}$$

which we may rewrite as,

$$:= \sum_{\hat{\Gamma}} \frac{4\pi}{2\hat{l}+1} \tau_{\hat{l}, \hat{m}, \hat{n}} \mathcal{Y}_{\hat{l}}^{\hat{m}}(\vartheta_{\mathbf{x}}, \varphi_{\mathbf{x}}) \mathfrak{Z}_{\hat{l}, \hat{n}}(r_{\mathbf{x}}). \quad (36)$$

This suggests that computing the Hartree potential from the electron density expansion coefficients is very much like performing an inverse basis transform. The key difference is that the functions $\mathfrak{Z}_{\hat{l}, \hat{n}}(r)$ need to be used, instead of the usual

radial basis functions $\mathcal{R}_{l,n}(r)$, while carrying out the radial part of the calculation. If the $\mathfrak{Z}_{l,\hat{n}}(r)$ functions are pre-computed and stored, the method described here turns out to be extremely efficient: in our implementation, the entire calculation of obtaining the real space representation of V_H , starting from the real space representation of ρ , consumes less than 0.03% of the total time of a typical SCF cycle.¹⁹

The functions $\mathfrak{Z}_{l,\hat{n}}(r_{\mathbf{x}})$ may be written as follows:

$$\begin{aligned}\mathfrak{Z}_{l,\hat{n}}(r_{\mathbf{x}}) &= \frac{1}{R J_{\hat{l}+\frac{3}{2}}(b_{\hat{l}+\frac{1}{2}}^{\hat{n}})} \int_{r_{\mathbf{y}}=0}^{r_{\mathbf{y}}=R} \frac{r_{\mathbf{y}}^{\hat{l}}}{r_{\mathbf{y}}^{\hat{l}+1}} \sqrt{\frac{2}{r_{\mathbf{y}}}} J_{\hat{l}+\frac{1}{2}}\left(\frac{b_{\hat{l}+\frac{1}{2}}^{\hat{n}}}{R} r_{\mathbf{y}}\right) r_{\mathbf{y}}^2 dr_{\mathbf{y}}, \\ &:= \sqrt{2R} \tilde{\mathfrak{Z}}_{l,\hat{n}}(s), \text{ with } s = r_{\mathbf{x}}/R \text{ and } s \in [0, 1], \\ \tilde{\mathfrak{Z}}_{l,\hat{n}}(s) &= \frac{1}{J_{\hat{l}+\frac{3}{2}}(b_{\hat{l}+\frac{1}{2}}^{\hat{n}})} \left[s^{\hat{l}+1} \int_0^s r_1^{\hat{l}+\frac{3}{2}} J_{\hat{l}+\frac{1}{2}}\left(b_{\hat{l}+\frac{1}{2}}^{\hat{n}} r_1\right) dr_1 + s^{\hat{l}} \int_s^1 \frac{1}{r_1^{\hat{l}-\frac{1}{2}}} J_{\hat{l}+\frac{1}{2}}\left(b_{\hat{l}+\frac{1}{2}}^{\hat{n}} r_1\right) dr_1 \right],\end{aligned}\quad (37)$$

and r_1 simply denotes an integration variable. The integrals in Eq. (37) can be carried out numerically using Gauss quadrature. In our implementation, we have computed $\tilde{\mathfrak{Z}}_{l,\hat{n}}(s)$ accurately for a large number of values of \hat{l} and \hat{n} over a fine grid of values over $[0, 1]$ and stored the results. The values of $\tilde{\mathfrak{Z}}_{l,\hat{n}}(s)$ at other values of $s \in [0, 1]$ are computed using cubic spline interpolation as and when required. During an actual simulation, this procedure is used to quickly set up the functions $\mathfrak{Z}_{l,\hat{n}}(r_{\mathbf{x}})$ at the different radial grid points before the first SCF step.

2.3.2. Computation of the pseudopotential terms

Modern pseudopotentials usually consist of local and non-local terms [1]. We now look at how each of these terms can be evaluated within our framework.

The total local pseudopotential at a point $\mathbf{x} \in \mathcal{B}_R$ is a combination of terms of the form:

$$V_{\text{nu}}(\mathbf{x}) = \sum_{j=1}^M V_{\text{nu}}^j(|\mathbf{x} - \mathbf{x}_j|), \quad (38)$$

where each of the functions V_{nu}^j is reasonably smooth.²⁰ By observing that V_{nu} consists of radially symmetric terms which are centered at the atoms while the basis functions are centered at the origin, it is possible to make use of Löwdin transformations [78] to directly arrive at the expansion coefficients of local pseudopotential terms [35,36]. Our method for dealing with the local pseudopotential however, is to directly evaluate Eq. (38) at the gridpoints in B . This is because the local pseudopotential enters the Kohn–Sham calculation through the total effective potential and as described earlier, the total effective potential is required in real space representation during the computation of matrix vector products. The reciprocal space representation of the local pseudopotential can be evaluated by carrying out forward basis transforms, if required.

Non-local pseudopotentials are used in electronic structure methods in order to account for the effect of the inert core electrons on the chemically active valence electrons, without directly introducing these core states into the calculation [1,2]. From a computational point of view, the inclusion of a non-local pseudopotential means that a projection operator needs to be added to the Hamiltonian while performing matrix vector products or while computing the total energies/forces. In general, this projection operator can be written as the sum of atom centered rank one operators. By definition, the action of a rank one operator $\mathcal{O} = p_1 \otimes p_2$ on a function $f \in L^2(\mathcal{B}_R)$ is simply given as $\mathcal{O} f = \langle p_2, f \rangle_{L^2(\mathcal{B}_R)} p_1$. In our implementation, we first evaluate each of the projector functions p_1 and p_2 on the underlying real space grid, following which, we compute and store their expansion coefficients by means of basis transforms (ahead of the first SCF step). From these expansion coefficients, the action of the projector $p_1 \otimes p_2$ on an electronic state can be carried out in reciprocal space as the action of a rank one operator as described above. The collective action of all the atom centered projectors on all the electronic states can be conveniently described through a pair of matrix–matrix multiplications.

Instead of this reciprocal space formulation of the non-local pseudopotential terms, it is possible to carry out this calculation more efficiently in real space by making use of the fact that the functions p_1 and p_2 are usually short ranged (in real space). Some additional care is required so as to ensure that aliasing errors are avoided in this approach [79] and therefore, we intend to explore this methodology in future work.

¹⁹ This happens to be true even for the largest example systems considered later in this paper. From the discussion above, it is clear that the performance and scaling of the electrostatics terms will depend entirely on the efficiency and scalability of the basis transforms themselves. Various aspects of performance and scaling of the basis transforms are discussed throughout this paper. We do mention however, that in order to obtain better scalability of the electrostatics computation routines, we adopted a two level scheme that uses a process grid (the same grid discussed in Section 3.3) to obtain parallelization in the radial variable and in the radial basis function number (while using Eqs. (36) and (35)), thus effectively reducing communication loads.

²⁰ These functions are actually in $C^\infty(\mathbb{R})$ for the pseudopotentials considered in this work. They have a somewhat lower regularity for the popular Troullier–Martins pseudopotentials [52].

3. Implementation

We outline various implementation related issues and solution strategies in this section. In particular, we discuss methods of obtaining the occupied eigenspace of the Hamiltonian as well as parallelization aspects of some of the key routines and procedures employed in our method.

3.1. Diagonalization using LOBPCG

As remarked earlier, efficient eigensolvers for iterative diagonalization of the Hamiltonian matrix are necessary for dealing with large systems. Perhaps the most commonly used diagonalization method in ab initio calculations is the band-by-band conjugate gradient algorithm for direct minimization of the total energy [7,80], later modified to fit the iterative diagonalization framework [81]. In this work, we have adopted the Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) method [45]. The LOBPCG algorithm is much better supported theoretically [82], it has been shown to outperform the traditional preconditioned conjugate gradient method [46] and it has found applications in numerous electronic structure methods [46,47,83,84] due to its robustness.

When dealing with relatively small sized example systems (approximately a couple of hundred electrons), we have used the LOBPCG method exclusively to carry out diagonalization in all SCF steps. For some of the larger example systems described later, we used the LOBPCG method only in the first SCF step so as to generate a good guess for the Chebyshev polynomial filtered subspace iteration algorithm (described later) that was used in the subsequent SCF steps.

3.1.1. Implementation details

Our implementation of the LOBPCG method follows the algorithmic steps outlined in [85]. This allowed us to take advantage of *soft locking* whenever some eigenvectors converged faster than others,²¹ as well as *hard locking* which allowed us to carry out deflation against fixed orthonormal constraints. The latter proves to be particularly useful in calculations on large systems since the total number of electronic states may be too numerous to fit all required eigenstates into one LOBPCG block. This is primarily due to the large computational demands of the Raleigh–Ritz step used by LOBPCG.

A second detail is related to the use of Cholesky factorization for orthonormalization (of the residual vectors and conjugate directions) in the LOBPCG implementation. This technique is more computationally efficient but also known to be less reliable than the traditional approach involving QR factorization [85]. Computation of the Cholesky decomposition of matrices that are poorly scaled are often required by the LOBPCG method and so, it is crucial to either use a Cholesky decomposition that is numerically invariant with respect to matrix scaling, or to scale the columns of such matrices²² before performing the factorization [86]. In addition, our experience has been that numerical noise or round off errors (arising from the transform based matrix–vector product computations, for instance) can sometimes cause the Cholesky factorization or the Raleigh–Ritz procedure to become unstable. In these situations, we have always found it useful to restart the LOBPCG iterations (discarding the computed conjugate direction and residual vectors) by using the most recently computed block of eigenvectors as the initial guess of a fresh set of iterations. This simple strategy seems to result in a much more robust implementation and it does not introduce any computational bottlenecks.

3.1.2. Use of the Teter–Payne–Allan preconditioner

The need for a good preconditioner for use with the LOBPCG method has been emphasized in [82,85]. A majority of the generic preconditioners that have been developed over the years, are aimed towards sparse systems. These are unsuitable for our purposes because our method is matrix free, and moreover, the underlying matrix involved is dense. Fortunately, the presence of the Laplacian operator in the Kohn–Sham eigenvalue problem suggests a viable preconditioner [7,80]. These authors introduced a preconditioner within the context of the plane-wave method that has the particular advantage of being diagonal in reciprocal space (and it is therefore, inexpensive to apply). The formal similarities of our spectral method with the plane-wave method allowed us to directly adopt the diagonal preconditioner introduced by these authors.

Specifically, we used a preconditioning matrix $\mathbf{T}_{p_{\alpha,\beta}}$ of the form:

$$\mathbf{T}_{p_{\alpha,\beta}} = \frac{27 + 18g + 12g^2 + 8g^3}{27 + 18g + 12g^2 + 8g^3 + 16g^4} \delta_{\alpha,\beta}, \quad (39)$$

where g is the ratio of the Laplacian eigenvalue to the kinetic energy of the residual vector on which the preconditioner is being applied, i.e., denoting the residual vector as $\mathbf{Y} \in \mathbb{C}^d$,

$$g = \Lambda_{\mathcal{J}(\alpha)} / \left(\frac{1}{2} \sum_{\alpha=1}^d \Lambda_{\mathcal{J}(\alpha)} |\mathbf{Y}_{\alpha}|^2 \right). \quad (40)$$

²¹ This can have a considerable impact on speeding up SCF iterations – in many example systems, we found that diagonalization via LOBPCG in first SCF step is about 1.5–2 times slower than in SCF steps which are close to attaining convergence. The eigenvectors in the latter are already close to their converged values and therefore soft locking allows for faster progression of LOBPCG.

²² We are grateful to Andrew Knyazev (Mitsubishi Electric Research Laboratories) for his consistent support and suggestions during our implementation of LOBPCG, and in particular, for pointing out the stability issues related to Cholesky factorization.

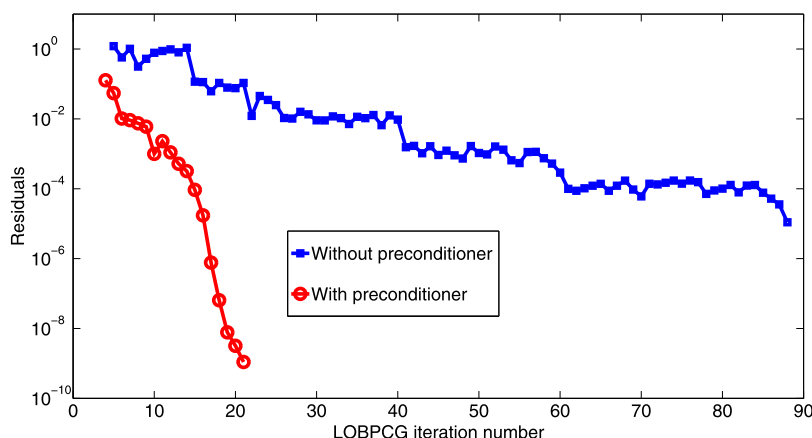


Fig. 2. Effect of the diagonal preconditioner on LOBPCG iterations. The average residual for the first few eigenvectors has been plotted against the iteration number.

As g approaches zero, the preconditioner elements approach one with zero derivatives up to third order and so, \mathbf{T}_p leaves the low energy states unchanged. On the other hand, above $g = 1$, \mathbf{T}_p asymptotically approaches the inverse Laplacian thus suitably damping out the high kinetic energy states that are responsible for ill-conditioning.

As can be seen from Fig. 2, this simple and inexpensive preconditioner makes a marked difference in the rate of convergence of the residuals in LOBPCG. The particular system used for the demonstration was an 18 atom barium cluster for which 4000 basis functions were used and only the linear part of the Kohn–Sham equations was solved. This preconditioner was therefore adopted in all further calculations wherever the LOBPCG solver was employed.

3.1.3. Parallelization scheme and scaling

Our strategy for parallelization of the LOBPCG method is to carry out relevant linear algebra operations using a distributed memory dense linear algebra library. For this purpose, we have adopted the state of the art numerical library²³ Elemental [87]. This library has been designed to be a more scalable and easier to interface successor of the ScaLAPACK [88,89] and PLAPACK [90,91] libraries that have already found widespread use in other electronic structure codes. Elemental uses an element-wise block-cyclic distribution of matrices over a two-dimensional grid of processors.²⁴ The Message Passing Interface (MPI) is used for interprocess communication while linear algebra operations that are local to each process are carried out by making calls to (serial) BLAS and LAPACK libraries.²⁵

The dimensions of the process grid that underlies the linear algebra operations in Elemental can have an impact on the resulting parallel efficiency of the LOBPCG routine. We have used square process grids in most cases. In some cases however, we found that the use of rectangular process grids, in which the height of the grid was longer than the width of the grid seemed to result in better performance.

In order to judge the parallel efficiency of our LOBPCG implementation, we studied the weak scaling of our routine in the following manner. We generated random dense hermitian matrices of various sizes and computed the first few hundred eigenstates. We used between 16 and 512 c.p.u. cores, the matrix size was increased in proportion to the number of cores used²⁶ and the number of states computed was held constant. We measured the average time per LOBPCG step and the results from this study are plotted in Fig. 3. Keeping in mind that one of the most computationally expensive steps in the setting of dense matrices is due to parallel matrix vector multiplications in which the problem size grows quadratically²⁷ with increasing matrix dimension [92], we have also plotted in Fig. 3, the computational complexity adjusted parallel efficiency. As is evident from the figure, the adjusted parallel efficiency remains above 90% up to 256 c.p.u. cores and drops to a little less than 80% at 512 c.p.u. cores.²⁸

²³ We are grateful to Jack Poulson (Georgia Tech.), the lead author of the Elemental package for his suggestions and help with the package.

²⁴ See Fig. 4 to see an example of how the data is distributed among processors.

²⁵ To ensure maximum use of hardware resources, our code was linked to machine optimized BLAS and LAPACK libraries.

²⁶ The computational platform details are described in a later section.

²⁷ This is unlike the weak scaling studies presented in [85] in which sparse matrices were used, resulting in linear growth of problem size with increasing matrix dimension.

²⁸ Here as well as in Section 3.3.1, we have focused on weak scalability results. This is because we were primarily interested in ensuring that our code is able to handle even large materials systems within a reasonably constant wall-clock time. Our underlying assumption of course, was that more computational resources would be allocated to the code, when necessary, in order to meet the wall-clock time requirements. We do mention however that the strong scaling of our LOBPCG routine is reasonably good, although it is not as encouraging as its weak scaling. In a test involving the computation of a few hundred eigenstates of a randomly generated hermitian matrix of dimension $40,960 \times 40,960$, the strong parallel efficiency was about 60% for 128 MPI processes and it dropped to about 40% for 256 MPI processes. More details may be found in [93].

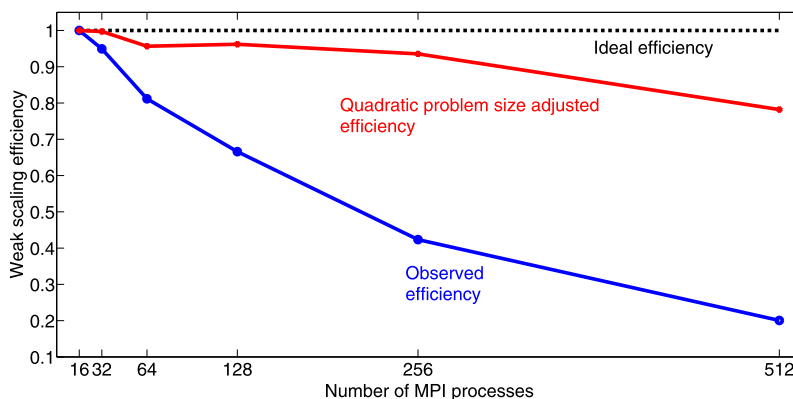


Fig. 3. Weak parallel scaling behavior of the LOBPCG implementation measured by time taken per LOBPCG step vs. the number of MPI processes employed, while keeping the problem size per process constant.

3.2. Chebyshev polynomial filter accelerated subspace iterations

Subspace iteration algorithms constitute a generalization of the classical power iterations approach to computation of eigenpairs [70,94]. These methods allow the computation of multi-dimensional invariant subspaces rather than one eigenvector at a time. Since the electron density or the total Kohn–Sham energy do not depend explicitly on the eigenvectors of the Hamiltonian, but only on the occupied subspace, subspace iterations have often been used for electronic structure calculations [95–97].

The Chebyshev polynomial filtered SCF iteration technique for computing the occupied eigenspace of the Kohn–Sham operator was introduced in [25,48], and was originally presented in the context of the finite difference method. However, this method has enjoyed success in conjunction with finite elements as well [21]. The method can be thought of as a form of non-linear subspace iteration which takes advantage of the fact that eigenvectors of the Hamiltonian do not need to be computed accurately at every SCF step since the Hamiltonians involved are approximate as well. This allows one to exploit the non-linear nature of the problem in the sense that the technique removes emphasis on the accurate solution of the intermediate linearized Kohn–Sham eigenvalue problems. By means of spectral mapping, the method employs the exponential growth property of the Chebyshev polynomials outside the region $[-1, 1]$ to damp out the unwanted part of the spectrum of the Hamiltonian thus accelerating the subspace iterations.

Although the Chebyshev polynomial filtered SCF iteration technique does not seem to have been adopted in the plane-wave method literature so far, it was apparent to us that as long as one has access to efficient matrix–vector product routines, the technique is likely to yield large savings compared to traditional diagonalization methods. This is primarily due to the fact that orthonormalization and other linear algebra operation costs that accompany traditional diagonalization methods are minimal in this method.

3.2.1. Implementation details

The Chebyshev polynomial filtered SCF iteration technique is presently the work horse of most medium to large sized computations carried out using the ClusterES package. In our implementation of this method, we first obtain a guess for the initial electron density by linearly superposing precomputed atomic electron densities. Next, having computed the potentials, we use the LOBPCG method (on a collection of randomly generated vectors used as an initial guess²⁹) to obtain a good eigenbasis of the Hamiltonian for the first SCF step. This is used to serve as a good guess for the occupied subspace of the Hamiltonian at self-consistency.³⁰ The Chebyshev polynomial filtered subspace iterations begin after this first SCF step and they attempt to adaptively improve the initial guess subspace by polynomial filtering.³¹

In the original presentation of the Chebyshev filtering method, the authors used the DGKS algorithm [99] for orthonormalization of the basis vectors of the occupied subspace. In the spirit of the LOBPCG method as well as the RMM-DIIS method [4], we have used the faster (but less stable) Cholesky factorization method instead. This helped speed up the orthonormalization calculation (by a factor of 2–3 in most cases) and we have not witnessed any problematic side effects.

As described in [25,48], the bulk of the Chebyshev filtering algorithm consists of evaluating the polynomial filter using matrix vector products. The additional linear algebra operations involved are in the form of scaling and shifting, orthonor-

²⁹ This appears to be a fairly common practice in the literature – see for example [80,98].

³⁰ Typically, a few extra states (about 10–20) are included from the LOBPCG calculation so that the Raleigh–Ritz step (used in the Chebyshev filtering method) and finite-temperature Fermi–Dirac smearing (used for metallic systems) can be employed.

³¹ We recently became aware of techniques which can omit the first diagonalization step in lieu of filtering [98]. We intend to adopt this methodology into our code in the near future.

Table 2

Performance of Chebyshev filtered SCF iterations compared against LOBPCG based SCF iterations.

Material system	No. of basis functions used	No. of electronic states used	No. of LOBPCG SCF steps	No. of Chebyshev SCF steps	Ratio of LOBPCG step time to Chebyshev step time
172 atom aluminum FCC cluster	512,000	280	22	23	20.2
C ₆₀ buckyball	343,000	136	11	13	12.3

0	2
1	3

(a) Numbering of processes in a 2×2 process grid.

$$\begin{pmatrix} 0 & 2 & 0 & 2 \\ 1 & 3 & 1 & 3 \\ 0 & 2 & 0 & 2 \\ 1 & 3 & 1 & 3 \\ 0 & 2 & 0 & 2 \\ 1 & 3 & 1 & 3 \end{pmatrix} \xrightleftharpoons[\text{Inv. Trans.}]{\text{Fwd. Trans.}} \begin{pmatrix} \{0,1\} & \{2,3\} & \{0,1\} & \{2,3\} \\ \{0,1\} & \{2,3\} & \{0,1\} & \{2,3\} \\ \{0,1\} & \{2,3\} & \{0,1\} & \{2,3\} \\ \{0,1\} & \{2,3\} & \{0,1\} & \{2,3\} \\ \{0,1\} & \{2,3\} & \{0,1\} & \{2,3\} \\ \{0,1\} & \{2,3\} & \{0,1\} & \{2,3\} \end{pmatrix}$$

(b) Individual matrix entry ownership (by processes) during basis transforms.

Fig. 4. Example of data redistribution during forward and inverse transforms. A process grid of dimension 2×2 has been used for storing 4 electronic states and the number of expansion coefficients for each state (i.e., basis set size) is 6. Inter process communication occurs only along process grid columns during the transforms.

malization and the Raleigh–Ritz step. Therefore, as in our implementation of the LOBPCG method, we used the Elemental package and its underlying process grid structure for carrying out these dense linear algebra operations in parallel.

Table 2 shows the performance gains obtained by our use of the Chebyshev filtered subspace iteration method when compared to LOBPCG. For the two examples presented in that table, each typical Chebyshev filtered SCF step is about 10–20 times faster than each typical LOBPCG based SCF step while the total number of SCF steps used by both methods to reach similar levels of convergence is about the same. Thus, there is an enormous amount of savings in the total computation time for the examples presented. It seems likely that for larger material systems, the savings are even greater.

3.3. Parallelization of matrix vector products and electron density computation: two level scheme

For systems containing up to a few thousand electronic states, the Hamiltonian matrix times vector computation routine is one of the main computationally intensive steps in the LOBPCG method and it is the principle one in the Chebyshev filtering method. These methods typically require the product of the Hamiltonian with a block of vectors to be computed. Due to our use of the two dimensional process grid for carrying out dense linear algebra operations (see Sections 3.1.3, 3.2.1), the block of vectors that needs to be multiplied with the Hamiltonian already appears distributed over the process grid. Specifically, the states are distributed over the process grid columns and each state is further distributed over process grid rows. In this situation, it is natural to parallelize the matrix vector product over the different Kohn–Sham states (i.e., band/state parallelization, as it is often called in the plane-wave literature) since this involves no communication between the processors that store the different states.

However, we observe that the inverse basis transform requires access to all the expansion coefficients that constitute an entire state (Eq. (18)) while the forward basis transform requires access to function values at all the grid points (Eqs. (16), (17)). Since the process grid for the linear algebra operations distributes each state over the process grid rows, this requires the basis transforms to induce communication within process grid columns. The data redistribution over the process grid that is required for these basis transforms is shown schematically in Fig. 4.

A crucial detail is that the forward and inverse spherical harmonics transforms (which constitute the bulk of the operations involved within the basis transforms) can be performed independently over the various radial grid points. Thus, we may adopt a second level of parallelization by distributing real space quantities over different values of the radial gridpoints since this will ensure that the basis transform routines are mostly communication free. Fig. 5 shows a schematic outline of the individual steps of the matrix vector computation procedure over the process grid.

The two level parallelization scheme described above is in the spirit of similar schemes adopted by some large scale plane-wave codes [49,71]. Due to this scheme, the only communication involved during the matrix vector product calculations is over individual process grid columns: one time during the broadcast of the different portions of a particular state while the inverse basis transform occurs and a second time during computation of the radial quadratures while the forward basis transform occurs. The important point however, is that the communication load gets reduced from the total number of processors involved, to roughly the square root of the total number of processors (if a square process grid is in use). Also, due to the distribution of various real space quantities over the radial grid points, the memory overhead is reduced as well.

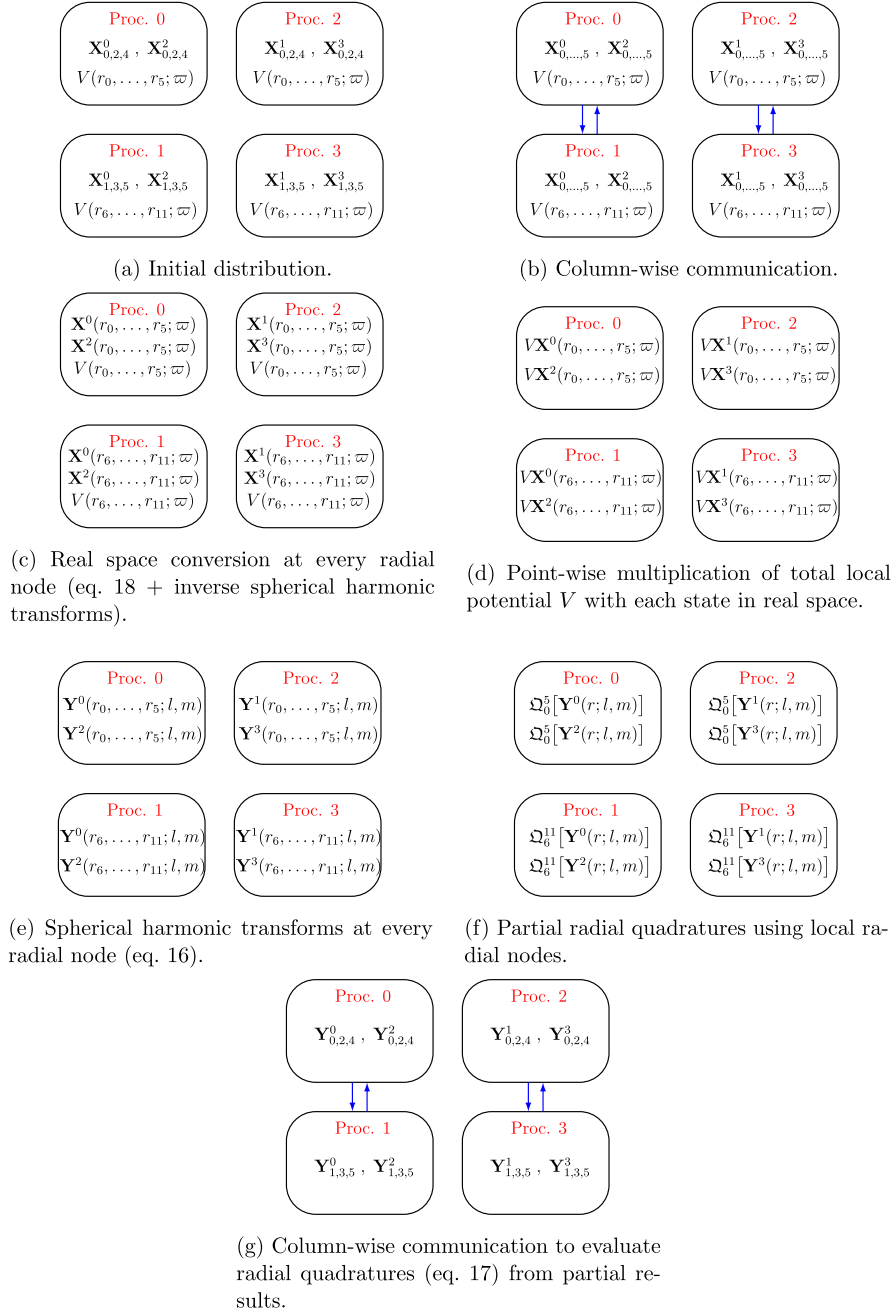


Fig. 5. (a)–(g): Schematic of the steps involved in computing the Hamiltonian times block of vectors product using the 2×2 process grid. Only the local part of the total potential is considered here. The block of vectors \mathbf{X} has 4 states (denoted with superscripts) with 6 expansion coefficients (labeled using subscripts) used for each state. The real space grid has 12 points r_0, \dots, r_{11} in the radial direction. The angular grid is left unspecified here and denoted as $\varpi = (\vartheta, \varphi)$. Real space quantities are shared along process grid columns by distributing the radial nodes. The product of the Hamiltonian with the block of vectors is denoted as \mathbf{Y} here. The symbol $\Omega_l^j[t(r)]$ is used to denote a partial radial quadrature (i.e., evaluation of Eq. (17)) over the radial nodes r_i, \dots, r_j .

The computation of the electron density from the Kohn–Sham states can also be made to follow this two level parallelization scheme. In this case, while computing the electron density in real space (from the Kohn–Sham states in reciprocal space), the communication involved is once along individual process grid columns during computation of the inverse basis transform and a second time along individual process grid rows while summing the results from the different Kohn–Sham states according to Eq. (3). Once again, this means that the communication load scales roughly as the square root of the number of processors.

3.3.1. Scaling performance and process grid geometry choice

We now discuss the scaling performance of the matrix vector product routine. In order to properly assess and interpret the scaling performance, we need to be mindful of the two dimensional nature of the underlying parallelization scheme. In particular, the choice of an optimal process grid geometry for a fixed basis set size, may be done as follows. First, with the given basis set size, we observe the computation time for the matrix vector product routine using only one electronic state. We increase the process grid height (keeping the process grid width fixed at 1) till the optimum performance is reached. Due to increasing communication costs during basis transforms, the performance saturates after a sufficiently large process grid height. In Fig. 6(a), we used one million basis functions for our study and for this basis set size, saturation occurs³² after a grid height of 16.

Now, keeping the process grid height at 16, the process grid width may be varied in proportion to the number of Kohn–Sham states that are required for the calculation. The reason for this strategy is clear from Fig. 6(b) – the two level parallelization scheme is able to make use of the embarrassingly parallel nature of the problem with the number of Kohn–Sham states in use and this is reflected from the nearly perfect weak scaling performance of the code.³³ For this same reason, we have also been able to verify that the strong scaling behavior of our code remains highly favorable³⁴ at even 512 total MPI processes. We anticipate that the scaling performance is likely to remain at such favorable levels for even larger numbers of processors.

Although this straight-forward method for obtaining optimum process grid geometries (based on insights into the scaling performance) is useful, in practice it is sometimes possible to get even better performance from individual process grid configurations, depending on the specific number of basis functions and states in use. Some examples of such cases are also shown³⁵ in Fig. 6(b). The overall conclusion that we were able to draw from these studies is that out of the two levels of parallelism available in our implementation, state parallelism is more effective and scalable than the physical domain parallelism.

The scaling performance of the electron density computation routine can be understood on similar lines. We choose to skip further details of this since, unlike the time spent on matrix vector products, the time spent on computing the electron density is typically a relatively minor fraction of the total time spent in every SCF step.

3.4. Miscellaneous details

We briefly outline miscellaneous implementation related details in this section.

3.4.1. Mixing and smearing schemes

As mentioned earlier, SCF iterations typically employ mixing schemes in order to accelerate convergence towards the fixed point of the Kohn–Sham map [1]. The importance of mixing schemes in SCF iterations has been recognized both empirically and theoretically [57], leading to the development of various methods over the years [100–105]. We employed the multiple stage Anderson mixing scheme [100,106] in this work. Our implementation allows for mixing of the total effective potentials or of the electron density. We have found that potential mixing tends to result in faster convergence of the total energies in most systems. A complete mixing history was used in all the examples and the associated linear mixing parameter used was between 0.1 and 0.3.

Regardless of the mixing procedure, materials systems which have small or no band gaps (metallic systems, for instance) tend to experience convergence issues in the SCF iterations. This occurs due to degenerate energy levels near the Fermi surface in these systems, and it usually manifests itself as *charge sloshing* [107]. A common solution to this problem is to introduce *smearing* of the Fermi surface by prescribing a distribution of occupation numbers for the Kohn–Sham states [106]. We implemented the widely used Fermi–Dirac distribution for this purpose. This scheme introduces electronic temperature dependent orbital occupations as:

$$f_i = \frac{1}{1 + \exp\left(\frac{\lambda_i - \epsilon_F}{k_B \Theta}\right)}, \quad (41)$$

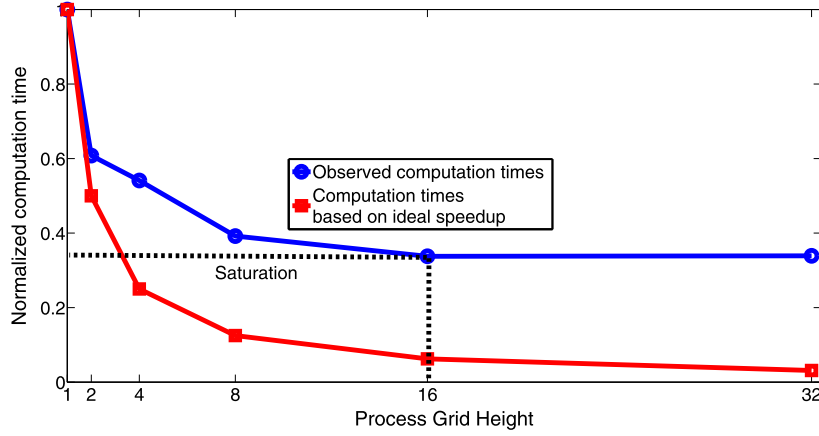
in which, the Fermi level ϵ_F can be determined by solving the constraint:

³² Note that while the number of basis functions used was one million, the number of radial points used was only 200. This helps us understand why the parallelization based on decomposition of the real space grid (which is based on the radial variable), has a relatively quick saturation at a grid height of 16.

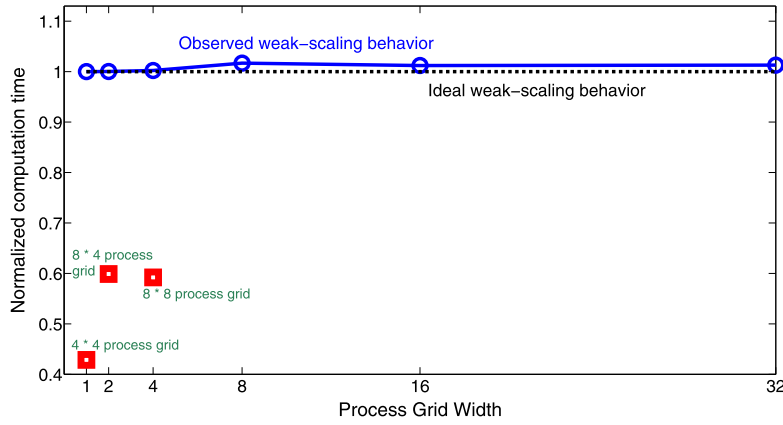
³³ A weak scaling parallel efficiency of over 98% is attained with a process grid width of 32, i.e., a total of 512 processes. We changed the process grid width (from 1 to 32, in multiples of 2) in proportion to the number of states for the weak scaling study, thus varying the total number of MPI processes between 16 and 512.

³⁴ The strong parallel efficiency remains well above 97% at process grid width of 32 (i.e. 512 total MPI processes). For testing the strong scalability, we kept the basis set size as well as the number of Kohn–Sham states constant (at 10^6 and 256 respectively). The process grid height was kept constant, while the process grid width was allowed to vary with increase in the number of MPI processes employed. More details may be found in [93].

³⁵ The problem sizes used for these individual examples were commensurate with the total number of processes in use, i.e., the 8×8 and 16×4 process grids were made to use the same number of processes, for example.



(a) Normalized matrix vector product computation time variation with increasing process grid height. A single Kohn-Sham state and 10^6 basis functions was used for the computation. Saturation occurs at a grid height of 16.



(b) Normalized matrix vector product computation time variation with increasing process grid width and proportionate increase in the number of Kohn-Sham states (10^6 basis functions for each state). The grid height was kept fixed at 16 for the blue curve and so, a total of 16 to 512 MPI processes were used.

Fig. 6. Parallel scaling efficiency of the matrix–vector product routine and its dependence on process grid geometry. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\int_{\mathbb{R}^3} \rho = N_e \implies \sum_{i=1}^{N_e/2} f_i = N_e/2. \quad (42)$$

We solved Eq. (42) using Brent's method [108] and we set the electronic temperature Θ to 100–200 Kelvin for all simulations where Fermi–Dirac smearing was used.

3.4.2. The ClusterES package

We have incorporated all the methods and algorithms discussed so far into an efficient and reliable package called ClusterES (Cluster Electronic Structure). Since our package makes heavy use of Spherical Harmonics Transforms, access to optimized and efficient routines for carrying out these transforms is essential for good performance of our code. We adopted the state of the art SHTns³⁶ library [39] for this purpose. In spite of using a traditional cubic order algorithm for computation (as opposed to algorithms which are asymptotically faster, e.g. [66]) this library has been shown to far outperform other Spherical Harmonics Transform routines because of its use of various hardware level optimizations [39].

³⁶ We are grateful to Nathanaël Schaeffer (CNRS, France) for his help and support with the SHTns library.

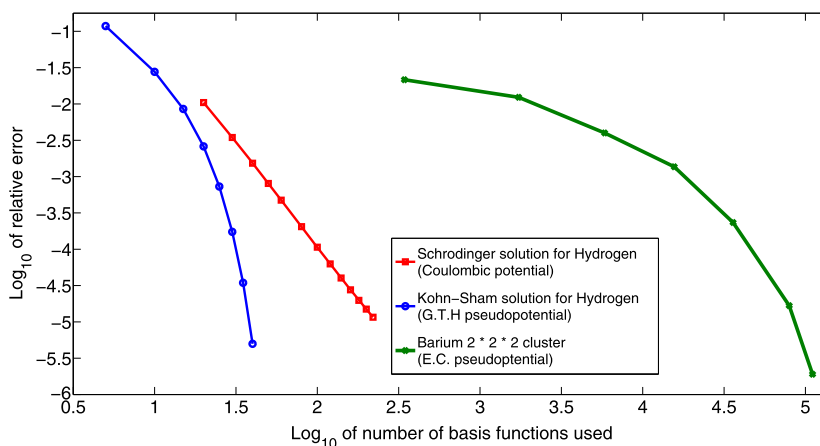


Fig. 7. Convergence of numerical solutions with increasing number of basis functions. The Coulombic nuclear potential shows a polynomial rate of convergence while smooth pseudopotential solutions show a faster than polynomial rate of convergence. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The spherical Bessel functions and the associated Legendre polynomials required for various computations in our code were generated using routines from the GNU Scientific Library [109]. Evaluation of the Gauss quadrature weights and nodes were carried out using the algorithm presented in [110]. Computation of the roots of the Bessel functions was carried out by Halley's method [64].

3.4.3. Computational platform

All computations were carried out on the Itasca cluster of the Minnesota Supercomputing Institute. Itasca is an HP Linux cluster with 1091 HP ProLiant BL280c G6 blade servers, each of which has two-socket, quad-core 2.8 GHz Intel Xeon X5560 “Nehalem EP” processors sharing 24 GB of system memory, with a 40 gigabit QDR InfiniBand (IB) interconnect. The GNU g++ compiler (ver. 4.8.1) along with Open MPI (ver. 1.7.1) was used and all serial linear algebra and FFT operations were carried out using the hardware optimized Intel Math Kernel Library (ver. 11.0).

4. Numerical results, example systems and applications

We finally describe various numerical results obtained using our method in this section. For all the computations described here, the radius of the spherical domain was chosen by following the procedure suggested in [17]: We first center the cluster/molecular system under study at the origin and then ensure that the atom(s) in the system that are furthest from the origin, are at least 8–16 atomic units away from the boundary of the sphere.³⁷

4.1. Convergence properties

We begin by studying the convergence properties of our method using numerical examples. First, we computed the ground state of the hydrogen atom based on the Schrödinger equation (i.e., Kohn–Sham self-consistent iterations were not used). This system has the particular advantage that the ground state energy is known analytically to be -0.5 Hartrees and so, it serves as an accurate reference for convergence studies. Since the ground state wave function is radially symmetric, we used only the $l=0$, $m=0$ spherical harmonic in the angular direction. Fig. 7 shows the convergence of the numerical solution to the analytical one with increasing number of basis functions. Due to the Coulombic singularity in the nuclear potential, the plot is a straight line indicating that the convergence rate is polynomial.

Next, we replaced the Coulombic potential for hydrogen with a smooth pseudopotential as parametrized in [111]. We computed the Kohn–Sham ground state of the hydrogen (pseudo) atom with this pseudopotential for increasing values of N , while using only the $l=0$, $m=0$ spherical harmonic in the angular direction in every case. We used the $N=50$ case as a reference³⁸ and plotted the (logarithmic) relative errors with increasing basis set size (with respect to the reference) in Fig. 7. In this case, due to the smoothness of the potential used, the plot has an overall curvature indicating a faster than polynomial rate of convergence (i.e., spectral convergence).

Finally, in order to assess the convergence properties for a full scale problem, we computed the ground state of a $2 \times 2 \times 2$ body centered cubic (BCC) cluster of barium. This cluster system has 35 atoms. We employed the smooth ‘Evanescent Core’

³⁷ The specific choice is dictated by computing single atom solutions and observing the decay rates of the electron density in such solutions. If multiple atom species are present, the atom with the slowest decay rate of the electron density is used to set the radius for the spherical domain in case of the cluster system.

³⁸ We verified that this reference case agrees with results from a standard plane-wave code up to at least 10^{-5} Hartrees.

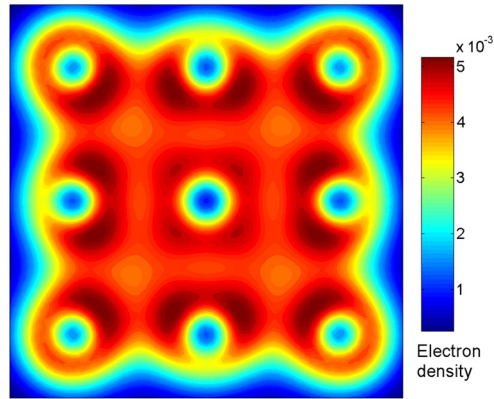


Fig. 8. Electron density contour plot of the $2 \times 2 \times 2$ BCC barium cluster computed using ClusterES.

local pseudopotential [112] for simulation of the barium atoms. In order to be able to obtain results which can be compared to the literature, we followed [21] to use a lattice constant of 9.5 a.u. and an electronic temperature of 200 K for our calculations. To make apparent the convergence rate of our method, we computed the ground state energy of this system using $\mathcal{L} = 100$, $\mathcal{N} = 100$ (i.e., one million basis functions) and used it as a reference value. Thereafter, starting from $\mathcal{L} = 7$, $\mathcal{N} = 7$ we computed the ground state energy for increasing values of \mathcal{L} and \mathcal{N} and plotted the (logarithmic) relative errors (compared to the reference value) as a function of the (logarithmic) basis set size as shown in Fig. 7. We see from this figure that because of our use of smooth pseudopotentials, once again, the convergence is rapid: even on a logarithmic scale, there is an overall curvature in the plot, thus indicating faster than polynomial rate of convergence (i.e., spectral convergence). Fig. 8 shows contour plots of the electron density for the barium cluster obtained using our code.

For the barium cluster, the ground state energy per atom obtained using our code comes out to be -0.6386253 Ha which compares well with the value of -0.6386277 Ha obtained using a plane-wave code³⁹ in [21]. This indicates not only rapid convergence of our code but also convergence to the correct value. As a matter of further comparison, we mention that in order to reach these aforementioned numbers, the plane-wave code needed to use over two million plane-waves (mainly arising due to a large vacuum region that had to be used around the cluster) whereas, our code used only 216,000 basis functions. Even with approximately 55,000 basis functions (a calculation that took only about 15 c.p.u. minutes on a laptop), we were able to reach convergence levels of about 2×10^{-4} eV/atom, which is an order of magnitude smaller than the usual levels of convergence demanded in accurate ground state energy calculations.

Aside from the numerical observations presented here, we should point out that an application of the analysis presented in [113] rigorously establishes that our basis set correctly approximates the Kohn–Sham ground states. A full scale mathematical investigation of the convergence rates of our basis set (similar to the results presented in [12] in the context of the plane-wave method) is the scope of future work.⁴⁰

4.2. Example calculations of various materials systems

Having ascertained the convergence properties of our method, we now compute the ground state properties of various metallic and non-metallic materials systems using our code and compare our results with the literature. Many of our examples related to metallic clusters (computed using local pseudopotentials) are motivated by recent work in finite element methods for the Kohn–Sham equations [20,21]. These examples gave us a way of verifying and benchmarking our calculations, as well as helping establish the relative ease and effectiveness by which such computations can be done routinely using our method.

4.2.1. All-electron calculations of light atoms

We begin by computing the ground state electronic structures of the first few elements of the periodic table. This serves as a simple test of our implementation. No pseudopotential was used. That is, these are all electron calculations. We used the parametrization of the Local Density Approximation as presented in [54,55]. The results of our computations are shown in Table 3 and compared with values from the literature. While the results are largely positive, they also illustrate the difficulty that our code faces when dealing with all-electron calculations. In spite of the spherical symmetry of the ground state, the Coulombic singularity at the origin makes it necessary to use a large number of radial basis functions to converge towards expected results. As the atomic numbers increase, so does the strength of the singularity and hence the increased difficulty of the computation. Therefore, we did not pursue the lithium and beryllium atom calculations after we ascertained

³⁹ The relative difference is of the order of 10^{-6} .

⁴⁰ We are grateful to Eric Cancès (Ecole des Ponts ParisTech, France) for providing us with useful suggestions in this direction.

Table 3

Ground state energies of a few light atoms (Hartree units used). Items marked with * indicate results where basis set convergence was not pursued due to the requirement of a large number of radial basis functions.

Element	ClusterES	Ref. [114]	Ref. [115]
Hydrogen	−0.445	−0.445	−0.445
Helium	−2.833	−2.834	−2.830
Lithium	−7.327*	−7.335	−7.338
Beryllium	−14.265*	−14.447	−14.434

Table 4

Ground state energies of a few light atoms (electron volt units used).

Element	ClusterES	Ref. [116]	Ref. [20]
Lithium	−5.97	−5.97	−5.97
Sodium	−5.21	−5.21	−5.21
Magnesium	−23.06	−23.06	−23.05

Table 5

Binding energy in electron volts per atom and bond length in atomic units for sodium and lithium dimers and octahedral clusters.

Cluster	Parameters	ClusterES	Plane-wave	Ref. [115]	Ref. [116]
Li ₂	Binding energy	−0.48	−0.48	−0.49	−0.52
	Bond length	4.75	4.75	4.86	4.92
Na ₂	Binding energy	−0.35	−0.35	−0.36	−0.46
	Bond length	5.71	5.72	5.72	5.77
Li ₆	Binding energy	−0.54	−0.54	−0.50	−0.72
	Bond length	5.72	5.72	5.69	5.79
Na ₆	Binding energy	−0.43	−0.43	−0.42	−0.53
	Bond length	6.79	6.79	6.80	6.87

that our results were within about 1% of the values from the literature. All subsequent calculations reported here employ pseudopotentials to mitigate this issue.

4.2.2. Local pseudopotential calculations

Having validated the basic correctness of our methodology and implementation using the all electron atomic calculations, we now move to pseudopotential calculations. We first work with the smooth local ‘Evanescent Core’ pseudopotential [112]. This bulk-fitted pseudopotential has been designed to deal with various simple metallic systems and because of the lack of non-local projectors, it is relatively computationally inexpensive. Due to the smoothness of the pseudopotential, we witnessed rapid convergence of our code with increasing basis set size in all the examples that follow.

We first compute the ground state energies of various pseudo-atoms using the pseudopotential and compare with the values from the literature. The results displayed in Table 4 show perfect agreement. Next, we computed the ground state properties of lithium and sodium dimers and octahedral clusters. We computed the binding energy (in electron volts per atom units) and the bond length (in atomic units) of these systems. For the octahedral clusters, as in [115,116], we did not perform any geometry optimization but only sought minima in terms of the nearest neighbor bond length. Also, following these authors, the cluster system ground states were computed without spin polarization while the individual atomic data used spin polarization. For reference purposes, we also carried out well converged plane-wave calculations⁴¹ on these cluster systems using the ABINIT [9] code. The results are shown in Table 5. We see that our results match with the plane-wave results almost exactly. Additionally, the overall agreement with the values in the literature is also good. The observable discrepancies with the results of [116] is probably because of the use of the LCAO method by those authors. We are not completely sure of the reasons behind the minor discrepancies with the results of [115]. However, there seems to be some confusion in the literature about the correct values of the parameters used by the evanescent core pseudopotential [117] and this might have caused a slightly different set of parameters to have been used in [115]. Also, as noted in more recent work [21], higher order finite elements are often necessary for well converged reliable calculations and these were not employed in [115]. We believe however, that the precise agreement between our results and the plane-wave code lend support to the credibility of our results.

Next, we study the properties of a few larger clusters of sodium consisting of $2 \times 2 \times 2$, and $3 \times 3 \times 3$ body centered cubic unit cells. We calculated the binding energy per atom and lattice constant for these clusters by computing the total energy for various values of the lattice parameter and then fitting this data to a cubic polynomial. Our results, compare essentially exactly to the results from well converged plane-wave calculations as Table 6 shows, assuring us of the efficacy of our method. As a matter of further illustration, let us mention that for the $3 \times 3 \times 3$ sodium cluster, at the minimum energy

⁴¹ We employed an energy cutoff of 30 Hartrees and a cell length of 30 atomic units or more in ABINIT.

Table 6

Binding energy per atom and lattice constant of sodium BCC unit cells.

Sodium cluster	Properties	ClusterES	Plane-wave	Ref. [20]
$2 \times 2 \times 2$	Binding energy (eV/atom)	−0.71	−0.71	−0.71
	Bond length (a.u.)	7.61	7.61	7.55
$3 \times 3 \times 3$	Binding energy (eV/atom)	−0.78	−0.78	−0.80
	Bond length (a.u.)	7.78	7.78	7.75

Table 7

Bond lengths of a few small molecules computed using the Goedecker–Teter–Hutter pseudopotentials.

Molecule	Bond length (a.u.): ClusterES	Bond length (a.u.): Ref. [111]
CO	2.128	2.127
CH ₄	2.074	2.072
SiH ₄	2.810	2.810
NH ₃	1.928	1.931
H ₂ O	1.833	1.835

bond length, the total ground state free energies from the plane-wave code and our code are −20.010982 Hartrees and −20.011008 Hartrees respectively. This corresponds to a difference of less than 0.5 micro-Hartrees per atom, demonstrating the extremely high accuracies that are easily accessible with our code.

Other values from the literature are also shown in that table. The overall agreement with these values is also very good (the bond lengths agree to within 1%, for example). The minor discrepancies from [20] are likely to be explained by the factors mentioned above.⁴²

4.2.3. Non-local pseudopotential calculations

In order to deal with a wider variety of materials systems, we now turn to calculations involving *ab initio* norm-conserving non-local pseudopotentials. This class of pseudopotentials is attractive because the pseudopotentials are accurate and transferable and at the same time, they are available for all elements in the periodic table (including ones which require relativistic treatment of the core electrons). Here, we look at the results obtained using the separable dual space Gaussian pseudopotentials introduced in [111,118]. This pseudopotential is available in analytical form with a small set of parameters for every element (thus allowing for easy implementation) and it satisfies an optimality criterion for the real-space integration of the nonlocal part. While this pseudopotential is known to be harder than other norm conserving pseudopotentials (i.e., it requires many more basis functions per atom for converged results), it is also known to be more accurate and transferable than other pseudopotentials [111].

We computed the bond lengths of a few small molecules using our spectral code and compared our results with values from literature, as presented in Table 7. Our results all agree to within 0.2% of values obtained by the authors of [111].

Next, we computed the ground state properties of a few larger systems consisting of organic molecules and fullerenes.⁴³ We compared our results with the literature as well as with plane-wave code⁴⁴ calculations (using ABINIT [9]) and finite difference method calculations (using the Octopus code [18]). The results are presented in Table 8. The agreement with the plane-wave and finite difference method results is excellent, thereby confirming the efficacy of our method. The overall agreement with other independent sources from the literature is also very good. The relatively minor differences with the results presented in [119] are most likely because of the use of a different pseudopotential by the authors of that work, while the difference from [120] occurs probably because of the use of an LCAO basis with a gradient corrected functional by those authors. Fig. 9(a) shows the electron density iso-surfaces of the Buckyball cluster while Fig. 9(b) shows the electron density contour plots for the azobenzene molecule.

Some of the examples presented above (for both local and non-local pseudopotentials) highlight the fact that our code is easily able to handle arbitrary cluster/molecule shapes and geometries. Indeed, thanks to the convergence properties of our basis set and our efficient implementation, even linear or planar molecules, which are quite far from spherical shapes, present no issues.⁴⁵

⁴² Also, it was not completely clear to us if the authors of [20] used spin-polarization for these particular sets of calculations. As before, we computed the cluster system ground states without spin polarization while the individual atomic data used spin polarization.

⁴³ We are grateful to Qing-Bo Yan (UCAS, China) for making the coordinates of the Boron fullerene available to us.

⁴⁴ The hardness of the pseudopotentials used often required energy cutoffs as large as 200 Hartrees to be employed for the plane-wave code.

⁴⁵ In order to further verify that our code does not face any difficulties in dealing with asymmetric or non-spherical systems, we carried out the following test (suggested to us by an anonymous reviewer): We computed the ground state energy of single silicon atom placed at the origin of a domain of radius 20 a.u. and then observed the change in energy of the system as this atom was moved outward radially. We observed that even at a radial distance of 8 a.u. from the origin, the energy change in the system remained less than 0.1 milli eV. The basis set size used in these examples was not particularly large – in fact even when we used a basis set which was 40% smaller in size, the change in energy of the system changed only by 0.25 milli eV when placed at a distance of 8 a.u. from the origin. Thus, the origin does not have a special status in our method and our code does not seem to require the use of very

Table 8
Ground state energy (eV/atom) and HOMO-LUMO gap (eV) of some organic molecules and fullerenes computed using our code and compared with results obtained from other sources. F.D.M. denotes finite difference method calculations done using the Octopus [18] code. Plane-wave calculations were carried out using the ABINIT code [9].

System	Properties	ClusterES	Plane-wave	F.D.M.	Other sources
Benzene	Ground state energy	−85.47	−85.47	−85.48	−85.65 (Ref. [119])
C ₆ H ₆	HOMO-LUMO gap	5.15	5.15	5.15	5.22 (Ref. [119])
Buckyball	Ground state energy	−155.09	−155.09	−155.09	−155.02 (Ref. [119])
C ₆₀	HOMO-LUMO gap	1.64	1.64	1.64	1.64 (Ref. [121])
Azobenzene	Ground state energy	−106.68	−106.68	−106.68	–
C ₁₂ H ₁₀ N ₂	HOMO-LUMO gap	1.39	1.39	1.39	–
Boron fullerene	Ground state energy	−76.94	−76.94	−76.95	–
B ₉₆	HOMO-LUMO gap	0.79	0.79	0.79	0.78 (Ref. [120])

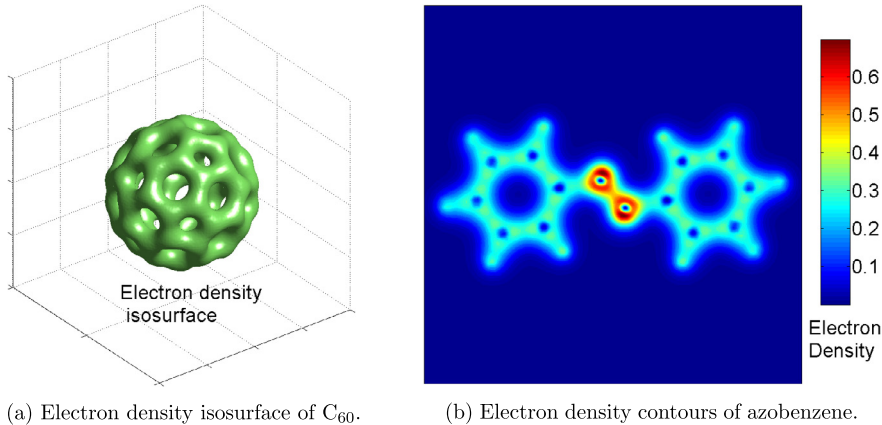


Fig. 9. Ground state electron density plots for the C₆₀ buckyball and the azobenzene molecule.

Table 9
Dipole moment of the carbon monoxide molecule at equilibrium bond length. F.D.M. stands for a calculation using the finite difference method carried out using the Octopus code [18].

System	Property	ClusterES	Ref. [124]	F.D.M.
CO	Dipole moment (Debye)	−0.23	−0.22	−0.23

The use of spherical harmonics in our basis set makes it very convenient to systematically compute several other quantities of interest, such as electrostatic multipole moments. Indeed, following the expressions presented in [122, p. 108] we see that quadruple or dipole moments can be easily obtained in our method by carrying out computations similar to the computation of forward basis transforms. We carried out this exercise for obtaining the dipole moment of the carbon monoxide molecule. We chose this particular example since it appears to us that various authors seem to have obtained a wide range of values of this quantity due to systematic errors in their computations – probably, either through incomplete basis sets (see [123] and [122] where values of −0.01 D and −0.60 D are respectively mentioned) or possibly through the use of unconverged grids or inaccurate pseudopotentials (as apparently obtained in [17]). The currently accepted value of this quantity at equilibrium bond length (using LDA calculations)⁴⁶ appears to be about −0.22 D [122,124,125] which agrees well with our results as Table 9 shows.

4.3. Benchmark calculations on large systems

Finally, in order to demonstrate the capabilities of our method in dealing with large systems efficiently, we carry out computations of the ground states of large aluminum clusters. We looked at 3 × 3 × 3, 5 × 5 × 5 and 7 × 7 × 7 face centered cubic (FCC) clusters for this study. The lattice spacing was fixed at 7.45 a.u. for all the clusters and we used the ‘Evanescent Core’ pseudopotential [112] for these calculations. A thermalization temperature of 100 Kelvin was used. For the 3 × 3 × 3 and 5 × 5 × 5 clusters, in order to assess the efficacy of our method, we aimed to converge our ground state energies (per

large basis sets in dealing with highly asymmetric systems. Of course, in most practical situations, the system under study is placed such that its center of mass coincides with the origin, at least approximately.
⁴⁶ This differs from the experimental value by about a factor of two [see e.g. [124]]. This discrepancy is usually ascribed to correlation effects being insufficiently modeled in Kohn–Sham LDA calculations [122].

Table 10

Ground state energy per atom of large aluminum clusters. Electron-volt units used. The plane wave and FEM results were obtained from Ref. [21].

System	No. atoms	No. electrons	ClusterES	Plane-wave	FEM
$3 \times 3 \times 3$	172	516	−56.01809	−56.01814	−56.01776
$5 \times 5 \times 5$	666	1998	−56.05057	−56.05068	−56.04906
$7 \times 7 \times 7$	1688	5064	−56.05812	–	−56.06826

Table 11

Computational run times of ClusterES compared against existing plane-wave and FEM codes. All run times are presented in c.p.u. hours. The plane wave and FEM results were obtained from Ref. [21].

System	ClusterES	Plane-wave	FEM
$3 \times 3 \times 3$ FCC aluminum cluster	18	646	371
$5 \times 5 \times 5$ FCC aluminum cluster	1948	7307	6619

atom) to within one–two milli electron volts of the plane-wave and higher order finite element method (FEM) results^{47,48} presented in [21]. For the $7 \times 7 \times 7$ cluster, due to computational resource constraints, we used a somewhat smaller basis set than what would be required to achieve this same level of convergence. So we present here results in which the total energy was within 0.01 electron volts per atom of the higher order finite element method (FEM) results. The results are shown in Table 10. To show that our methodology and its implementation is highly competitive with existing methods, we display in Table 11 timing results⁴⁹ of the $3 \times 3 \times 3$ and $5 \times 5 \times 5$ systems and compare them with the results presented in [21]. The computational platform used by the authors in [21] was quite similar to our own, if not by some measures, superior to ours. Nevertheless, due to the fast convergence of our spectral basis set, the efficient basis transforms, and various other algorithmic methodologies adopted here, our code was able to well outperform the plane-wave and finite element codes. In particular, in spite of having access to highly efficient FFTs, the plane-wave code performance seems to have suffered due to the requirement of having large supercells (with large vacuum regions) for obtaining converged results with these clusters.

4.4. Brief comments on symmetry adaptation

Most plane-wave codes allow for some method of symmetry adaptation, usually in the form of special point sampling methods for the Brillouin zone [126,127]. Due to the formal similarities of our method with the plane-wave method, it is natural to investigate if symmetry adaptation can be carried out in a straight-forward way in our setting.

There indeed seems to be a relatively simple way of carrying out this enterprise. The key point is that (like in the case of plane-waves), the basis functions in use arise as eigenfunctions of the Laplacian operator. This operator commutes with all relevant symmetry operations – it commutes with translational symmetry in a periodic setting and similarly it commutes with all point group operations in our setting. In our case, this results in the fact that point group actions on the basis set can be computed easily: the spherical harmonics ensure that symmetry group action on the basis set can be written down analytically in terms of Wigner D-matrices [128]. Therefore, this gives us an efficient method of constructing Peter–Weyl projectors [129,130] onto the symmetry invariant irreducible subspaces of the problem at hand. These projected subspaces can then be employed, in conjunction with subspace iteration methods (such as Chebyshev filtered SCF iterations) to obtain a symmetry adapted reduction of a given problem.

A full scale report on symmetry adaptation within our basis set highlighting these points is currently under preparation.

5. Conclusions and future directions

In summary, we have proposed and implemented a method for efficient and accurate solution of the Kohn–Sham equations for clusters. This method serves as an analog of the plane-wave method for periodic systems and similar to that

⁴⁷ This corresponds to relative errors of the order of 10^{-5} .

⁴⁸ It was pointed out to us by an anonymous reviewer that these levels are convergence are more demanding than the standards typically adhered to in many electronic structure calculations. Often, it is sufficient for the ground state energy to reach convergence levels of about 1 milli-Hartree per atom. This does not change the conclusions about the benchmark calculations laid out in this section, in the sense that the performance of our code remains very favorable when compared with a standard plane-wave code (ABINIT, [9]), if both codes are made to use the minimum number of basis functions that would allow them to reach convergence levels of 1 milli-Hartree per atom in the ground state energies. In a cluster system involving 62 aluminum atoms (FCC structure with a single mono-vacancy), our code had a wall clock time which was about 1.8 times smaller than the wall clock time registered by the plane-wave code when both codes were made to attain this level of convergence. Both codes were executed with the same computational resources – specifically, each code was run on a single node of the Itasca cluster (hardware described at the beginning of Section 4) and a single computational thread on a single core of this node was used. For reaching this convergence level, the plane-wave code used over 370,000 basis functions, while our code used only about 43,000. We systematically ensured that the minimum energy cutoff and cell size that would be required by the plane-wave code to reach the desired accuracy was used. We intend to present more details of these kinds of benchmarks in future work.

⁴⁹ In order to enable comparison with the results in [21], we report here timings in total c.p.u. hours. This was estimated by multiplying the wall clock timings with the number of MPI processes used. The $3 \times 3 \times 3$ and $5 \times 5 \times 5$ aluminum clusters used 16 MPI and 256 MPI processes respectively.

method, it shows rapid and systematic convergence properties. We have demonstrated that with the adoption of various algorithmic strategies, our method produces reliable results for a vast array of materials systems. In terms of performance metrics, benchmark calculations on various cluster systems show that our method is highly competitive when compared with other established basis sets and methods, both in terms of accuracy and speed. The formal analogies of our method with the plane-wave method allow us to adopt, mutatis mutandis, a multitude of numerical and algorithmic strategies commonly employed by the plane-wave method, which eventually lead to the efficient and reliable performance of our implementation.

An additional outcome is that our method forms a systematic generalization of approximate spherical basis function based methods introduced earlier in the literature in a variety of contexts (most commonly for the purpose of jellium calculations). Our method retains the basic simplicity of those methods (since the basis functions employed are of a similar nature), but it has far superior performance and applicability than any of those approximate methods. Our basis functions allow arbitrary point group symmetries to be exploited systematically, and obtaining leverage out of this fact constitutes the subject of ongoing and future work. A promising area of research is to use the ClusterES package for first principles materials discovery based on the ideas broadly outlined in [131].

In a separate contribution, we are currently following up on this work by demonstrating the use of our method in the accurate computation of quantum mechanical forces. It appears to us that the application of the Hellman–Feynman force formula is straight-forward in our method: the global nature of the basis results in the absence of Pulay forces and further, the spectral convergence properties that are inherent to our basis also carry over to the forces. In the near future, we aim to carry out ab initio molecular dynamics simulations of various cluster systems of interest using our method.

Currently, one of the main computational bottlenecks in carrying out basis transforms is due to the transform in the radial direction, which scales quadratically in the number of radial basis functions (see the discussion following Eq. (18)). So far, our use of Gauss quadrature and that of machine optimized libraries has enabled us to keep the constant in front of this asymptotic expression small, thus leading to the competitive run times of our code in practice. In the long run, however, an asymptotically faster algorithm should be employed and we intend to explore various possibilities in this direction since it has a direct bearing on our ability to successfully tackle even larger systems of interest.

Finally, due to its use of Dirichlet boundary conditions, our proposed method allows for charged systems to be easily studied without the need for introducing an artificial background charge (as currently used in plane-wave codes). Thus, the study of charged cluster systems⁵⁰ using our approach is likely to be a fruitful avenue of research in the near future.

Acknowledgements

This work was primarily supported by Russell Penrose. It also benefited from the support of NSF-PIRE Grant No. OISE-0967140, ONR N00014-14-1-0714 and the MURI project FA9550-12-1-0458 (administered by AFOSR). We would like to thank the Minnesota Supercomputing Institute for making the parallel computing resources used in this work available. We would like to thank Phanish Suryanarayana (Georgia Tech.) for his many insightful comments and suggestions at various stages of this work. We would like to thank Vikram Gavini and Phani Motamarri (U. Michigan) for stimulating discussions as well as for making available some of their Finite Element Method results which helped us in carrying out validation studies. We would also like to thank Gero Friesecke (TU Munich, Germany) and Michael Ortiz (Caltech) for informative discussions. We gratefully acknowledge comments from the anonymous reviewers which helped us in improving the presentation of our work. ASB and RDJ would like to acknowledge the hospitality of the Hausdorff Research Institute for Mathematics, Bonn, Germany where this work was partially carried out.

References

- [1] R.M. Martin, *Electronic Structure: Basic Theory and Practical Methods*, 1st edition, Cambridge University Press, 2004.
- [2] C. Le Bris (Ed.), *Computational Chemistry, Handb. Numer. Anal.*, vol. X, North-Holland, 2003.
- [3] Y. Saad, J.R. Chelikowsky, S.M. Shontz, Numerical methods for electronic structure calculations of materials, *SIAM Rev.* 52 (1) (2010) 3–54.
- [4] G. Kresse, J. Furthmüller, Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set, *Phys. Rev. B* 54 (1996) 11169–11186.
- [5] G. Kresse, J. Furthmüller, Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set, *Comput. Mater. Sci.* 6 (1) (1996) 15–50.
- [6] D. Marx, J. Hutter, *Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods*, 1st edition, Cambridge University Press, 2009.
- [7] M.C. Payne, M.P. Teter, D.C. Allan, T.A. Arias, J.D. Joannopoulos, Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients, *Rev. Mod. Phys.* 64 (1992) 1045–1097.
- [8] R.N. Barnett, U. Landman, Born–Oppenheimer molecular-dynamics simulations of finite systems: structure and dynamics of (H₂O)₂, *Phys. Rev. B* 48 (4) (1993) 2081.
- [9] X. Gonze, J.-M. Beuken, R. Caracas, F. Detraux, M. Fuchs, G.-M. Rignanese, L. Sindic, M. Verstraete, G. Zerah, F. Jollet, M. Torrent, A. Roy, M. Mikami, P. Ghosez, J.-Y. Raty, D. Allan, First-principles computation of material properties: the ABINIT software project, *Comput. Mater. Sci.* 25 (3) (2002) 478–492.
- [10] M.D. Segall, P.J.D. Lindan, M.J. Probert, C.J. Pickard, P.J. Hasnip, S.J. Clark, M.C. Payne, First-principles simulation: ideas, illustrations and the CASTEP code, *J. Phys. Condens. Matter* 14 (11) (2002) 2717.

⁵⁰ This possible avenue of research was suggested to us by an anonymous reviewer.

- [11] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G.L. Chiarotti, M. Cococcioni, I. Dabo, A.D. Corso, S. de Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougousis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A.P. Seitsonen, A. Smogunov, P. Umari, R.M. Wentzcovitch, QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials, *J. Phys. Condens. Matter* 21 (39) (2009).
- [12] E. Cancès, R. Chakir, Y. Maday, Numerical analysis of the planewave discretization of some orbital-free and Kohn–Sham models, *ESAIM: Math. Model. Numer. Anal.* 46 (2012) 341–388.
- [13] A.M. Rappe, J. Joannopoulos, P. Bash, A test of the utility of plane-waves for the study of molecules from first principles, *J. Am. Chem. Soc.* 114 (16) (1992) 6466–6469.
- [14] W. Hehre, R. Stewart, J. Pople, Self-consistent molecular-orbital methods. I. Use of Gaussian expansions of Slater-type atomic orbitals, *J. Chem. Phys.* 51 (6) (1969) 2657–2664.
- [15] J.C. Slater, G.F. Koster, Simplified LCAO method for the periodic potential problem, *Phys. Rev.* 94 (1954) 1498–1524.
- [16] J. Soler, E. Artacho, J. Gale, A. García, J. Junquera, P. Ordejón, D. Sánchez-Portal, The SIESTA method for ab initio order- n materials simulation, *J. Phys. Condens. Matter* 14 (11) (2002) 2745–2779.
- [17] J.R. Chelikowsky, N. Troullier, K. Wu, Y. Saad, Higher order finite difference pseudopotential method: an application to diatomic molecules, *Phys. Rev. B* 50 (1994) 11355–11364.
- [18] A. Castro, H. Appel, M. Oliveira, C. Rozzi, X. Andrade, F. Lorenzen, M. Marques, E. Gross, A. Rubio, Octopus: a tool for the application of time-dependent density functional theory, *Phys. Status Solidi, B Basic Res.* 243 (11) (2006) 2465–2488.
- [19] J. Pask, P. Sterne, Finite element methods in ab initio electronic structure calculations, *Model. Simul. Mater. Sci. Eng.* 13 (3) (2005) R71.
- [20] P. Suryanarayana, V. Gavini, T. Blesgen, K. Bhattacharya, M. Ortiz, Non-periodic finite-element formulation of Kohn–Sham density functional theory, *J. Mech. Phys. Solids* 58 (2) (2010) 256–280.
- [21] P. Motamarri, M. Nowak, K. Leiter, J. Knap, V. Gavini, Higher-order adaptive finite-element methods for Kohn–Sham density functional theory, *J. Comput. Phys.* 253 (2013) 308–343.
- [22] S. Botti, A. Castro, N.N. Lathiotakis, X. Andrade, M.A.L. Marques, Optical and magnetic properties of boron fullerenes, *Phys. Chem. Chem. Phys.* 11 (2009) 4523–4527.
- [23] X. Jing, N. Troullier, J.R. Chelikowsky, K. Wu, Y. Saad, Vibrational modes of silicon nanostructures, *Solid State Commun.* 96 (4) (1995) 231–235.
- [24] L. Kronik, A. Makmal, M.L. Tiago, M.M.G. Alemany, M. Jain, X. Huang, Y. Saad, J.R. Chelikowsky, Parsec the pseudopotential algorithm for real-space electronic structure calculations: recent advances and novel applications to nano-structures, *Phys. Status Solidi B* 243 (5) (2006) 1063–1079.
- [25] Y. Zhou, Y. Saad, M.L. Tiago, J.R. Chelikowsky, Parallel self-consistent-field calculations via Chebyshev-filtered subspace acceleration, *Phys. Rev. E* 74 (2006) 066704.
- [26] G.E. Scuseria, Ab initio calculations of fullerenes, *Science* 271 (5251) (1996) 942–945.
- [27] V. Gurin, Small metal clusters: ab initio calculated bare clusters and models within fullerene cages, in: E. Buzaneva, P. Scharff (Eds.), *Frontiers of Multifunctional Integrated Nanosystems*, in: NATO Sci. Ser. II Math. Phys. Chem., vol. 152, Springer, Netherlands, 2005, pp. 31–38.
- [28] N. Gonzalez-Szwacki, A. Sadrzadeh, B.I. Yakobson, B_{80} fullerene: an ab initio prediction of geometry, stability, and electronic structure, *Phys. Rev. Lett.* 98 (2007) 166804.
- [29] H. Hakkinen, Atomic and electronic structure of gold clusters: understanding flakes, cages and superatoms from simple concepts, *Chem. Soc. Rev.* 37 (2008) 1847–1859.
- [30] A.W. Castleman, S.N. Khanna, Clusters, superatoms, and building blocks of new materials, *J. Phys. Chem. C* 113 (7) (2009) 2664–2675.
- [31] A.W. Castleman, From elements to clusters: the periodic table revisited, *J. Phys. Chem. Lett.* 2 (9) (2011) 1062–1069.
- [32] M. Iñiguez, M. Lopez, J. Alonso, J. Soler, Electronic and atomic structure of Na, Mg, Al and Pb clusters, *Z. Phys., D, At. Mol. Clust.* 11 (2) (1988) 163–174.
- [33] G. Mattei, F. Toigo, Spherical averaged jellium model with norm-conserving pseudopotentials, *Eur. Phys. J., D, At. Mol. Opt. Phys.* 3 (3) (1998) 245–256.
- [34] K. Yabana, G.F. Bertsch, Electronic structure of C_{60} in a spherical basis, *Phys. Scr.* 48 (5) (1993) 633.
- [35] F. Alasia, R.A. Broglia, H.E. Roman, L. Serra, G. Colo, J.M. Pacheco, Single-particle and collective degrees of freedom in C_{60} , *J. Phys. B, At. Mol. Opt. Phys.* 27 (18) (1994) L643.
- [36] R. Broglia, G. Coló, G. Onida, H. Roman, *Solid State Physics of Finite Systems: Metallic Clusters, Fullerenes, Atomic Wires*, 1st edition, Adv. Texts Phys., Springer, 2004.
- [37] M. Brack, The physics of simple metal clusters: self-consistent jellium model and semiclassical approaches, *Rev. Mod. Phys.* 65 (1993) 677–732.
- [38] P.P. Teodorescu, N.-D. Stănescu, N. Pandrea, *Numerical Analysis with Applications in Mechanics and Engineering*, 1st edition, John Wiley & Sons, 2013.
- [39] N. Schaeffer, Efficient spherical harmonic transforms aimed at pseudospectral numerical simulations, *Geochem. Geophys. Geosyst.* 14 (3) (2013) 751–758.
- [40] R. Hockney, Potential calculation and some applications, *Methods Comput. Phys.* 9 (1970) 135–211.
- [41] J. Eastwood, D. Brownrigg, Remarks on the solution of Poisson's equation for isolated systems, *J. Comput. Phys.* 32 (1) (1979) 24–38.
- [42] G.J. Martyna, M.E. Tuckerman, A reciprocal space based method for treating long range interactions in ab initio and force-field-based calculations in clusters, *J. Chem. Phys.* 110 (6) (1999) 2810–2821.
- [43] L. Genovese, T. Deutsch, A. Neelov, S. Goedecker, G. Beylkin, Efficient solution of Poisson's equation with free boundary conditions, *J. Chem. Phys.* 125 (7) (2006) 074105.
- [44] C. Vömel, S.Z. Tomov, O.A. Marques, A. Canning, L.-W. Wang, J.J. Dongarra, State-of-the-art eigensolvers for electronic structure calculations of large scale nano-systems, *J. Comput. Phys.* 227 (15) (2008) 7113–7124.
- [45] A. Knyazev, Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method, *SIAM J. Sci. Comput.* 23 (2) (2001) 517–541.
- [46] F. Bottin, S. Leroux, A. Knyazev, G. Zerah, Large-scale ab initio calculations based on three levels of parallelization, *Comput. Mater. Sci.* 42 (2) (2008) 329–336.
- [47] F. Lorenzen, Massively-parallel eigensolver for the Octopus code, online article, 2006.
- [48] Y. Zhou, Y. Saad, M.L. Tiago, J.R. Chelikowsky, Self-consistent-field calculations using Chebyshev-filtered subspace iteration, *J. Comput. Phys.* 219 (2006) 172–184.
- [49] F. Gygi, E.W. Draeger, M. Schulz, B.R. de Supinski, J.A. Gunnels, V. Austel, J.C. Sexton, F. Franchetti, S. Kral, C.W. Ueberhuber, J. Lorenz, Large-scale electronic structure calculations of high- z metals on the bluegene/l platform, in: *Proceedings of the 2006 ACM/IEEE conference on Supercomputing, SC'06*, ACM, New York, NY, USA, 2006.
- [50] W. Kohn, L.J. Sham, Self-consistent equations including exchange and correlation effects, *Phys. Rev. A* 140 (4) (1965) 1133–1138.
- [51] G.B. Folland, *Real Analysis: Modern Techniques and Their Applications*, 2nd edition, Wiley, 1999.
- [52] N. Troullier, J.L. Martins, Efficient pseudopotentials for plane-wave calculations, *Phys. Rev. B* 43 (3) (1991) 1993.
- [53] R.G. Parr, W. Yang, *Density-Functional Theory of Atoms and Molecules*, Int. Ser. Monogr. Chem., vol. 16, Oxford University Press, USA, 1994.
- [54] J.P. Perdew, A. Zunger, Self interaction correction to density functional approximations for many electron systems, *Phys. Rev. B* 23 (10) (1981) 5048–5079.

- [55] D.M. Ceperley, B.J. Alder, Ground state of the electron gas by a stochastic method, *Phys. Rev. Lett.* 45 (7) (1980) 566–569.
- [56] J.P. Perdew, K. Burke, M. Ernzerhof, Generalized gradient approximation made simple, *Phys. Rev. Lett.* 77 (1996) 3865–3868.
- [57] P.H. Dederichs, R. Zeller, Self-consistency iterations in electronic-structure calculations, *Phys. Rev. B* 28 (1983) 5462–5472.
- [58] M. Hoffmann-Ostenhof, T. Hoffmann-Ostenhof, R. Ahlrichs, J. Morgan, On the exponential fall off of wavefunctions and electron densities, in: K. Osterwalder (Ed.), *Mathematical Problems in Theoretical Physics*, in: *Lect. Notes Phys.*, vol. 116, Springer, Berlin/Heidelberg, 1980, pp. 62–67.
- [59] R. Ahlrichs, M. Hoffmann-Ostenhof, T. Hoffmann-Ostenhof, J.D. Morgan, Bounds on the decay of electron densities with screening, *Phys. Rev. A* 23 (5) (1981) 2106–2117.
- [60] X. Jing, N. Troullier, D. Dean, N. Binggeli, J.R. Chelikowsky, K. Wu, Y. Saad, Ab initio molecular-dynamics simulations of Si clusters using the higher-order finite-difference-pseudopotential method, *Phys. Rev. B* 50 (16) (1994) 12234–12237.
- [61] A.S. Banerjee, Harmonic analysis on isometry groups of objective structures and its applications to objective density functional theory, Master's thesis, University of Minnesota, Minneapolis, 2011.
- [62] L.C. Evans, *Partial Differential Equations*, *Grad. Stud. Math.*, vol. 19, American Mathematical Society, 1998.
- [63] T. Kato, *Perturbation Theory for Linear Operators*, *Class. Math.*, Springer, 1995.
- [64] E.W. Weisstein, *CRC Concise Encyclopedia of Mathematics*, 2nd edition, Chapman & Hall/CRC, 2010.
- [65] J.W. Cooley, J.W. Tukey, An algorithm for the machine calculation of complex Fourier series, *Math. Comput.* 19 (90) (1965) 297–301.
- [66] M. Mohlenkamp, A fast transform for spherical harmonics, *J. Fourier Anal. Appl.* 5 (2–3) (1999) 159–184.
- [67] J.R. Driscoll, D. Healy, Computing Fourier transforms and convolutions on the 2-sphere, *Adv. Appl. Math.* 15 (2) (1994) 202–250.
- [68] A. Messiah, Clebsch–Gordan coefficients and 3j symbols, in: *Quantum Mechanics*, vol. 2, North-Holland, 1962, pp. 1054–1060, Appendix C.
- [69] I.S. Dhillon, B.N. Parlett, C. Vömel, The design and implementation of the MRRR algorithm, *ACM Trans. Math. Softw.* 32 (4) (2006) 533–560.
- [70] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, revised edition, SIAM, 2011.
- [71] E. Bylaska, K. Tsemekhman, N. Govind, M. Valiev, Large-scale plane-wave-based density functional theory: formalism, parallelization and applications, in: J.R. Reimers (Ed.), *Computational Methods for Large Systems: Electronic Structure Approaches for Biotechnology and Nanotechnology*, John Wiley & Sons, Inc., 2011, pp. 77–116.
- [72] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, *Spectral Methods: Fundamentals in Single Domains*, 1st edition, *Sci. Comput.*, Springer, 2006.
- [73] S.A. Orszag, Transform method for the calculation of vector-coupled sums: application to the spectral form of the vorticity equation, *J. Atmos. Sci.* 27 (6) (1970) 890–895.
- [74] S.A. Orszag, Numerical methods for the simulation of turbulence, *Phys. Fluids* 12 (12) (1969), II-250.
- [75] J. Rasch, A. Yu, Efficient storage scheme for precalculated Wigner 3 j, 6 j and Gaunt coefficients, *SIAM J. Sci. Comput.* 25 (4) (2004) 1416–1428.
- [76] L. Genovese, A. Neelov, S. Goedecker, T. Deutsch, S.A. Ghasemi, A. Willand, D. Caliste, O. Zilberberg, M. Rayson, A. Bergman, et al., Daubechies wavelets as a basis set for density functional pseudopotential calculations, *J. Chem. Phys.* 129 (2008) 014109.
- [77] J.D. Jackson, *Classical Electrodynamics*, John Wiley & Sons, New York, 1975.
- [78] P.O. Löwdin, Quantum theory of cohesive properties of solids, *Adv. Phys.* 5 (17) (1956) 1–171.
- [79] R. King-Smith, M. Payne, J. Lin, Real-space implementation of nonlocal pseudopotentials for first-principles total-energy calculations, *Phys. Rev. B* 44 (23) (1991) 13063.
- [80] M.P. Teter, M.C. Payne, D.C. Allan, Solution of Schrödinger's equation for large systems, *Phys. Rev. B* 40 (1989) 12255–12263.
- [81] D. Bylander, L. Kleinman, S. Lee, Self-consistent calculations of the energy bands and bonding properties of $B_{12}C_3$, *Phys. Rev. B* 42 (2) (1990) 1394.
- [82] A.V. Knyazev, K. Neymeyr, A geometric theory for preconditioned inverse iteration, III: A short and sharp convergence estimate for generalized eigenvalue problems, *Linear Algebra Appl.* 358 (1) (2003) 95–114.
- [83] C. Yang, J.C. Meza, L.-W. Wang, A constrained optimization algorithm for total energy minimization in electronic structure calculations, *J. Comput. Phys.* 217 (2) (2006) 709–721.
- [84] L. Lin, S. Shao, W. E, Efficient iterative method for solving the Dirac–Kohn–Sham density functional theory, *J. Comput. Phys.* 245 (15) (2013) 205–217.
- [85] A.V. Knyazev, M.E. Argentati, I. Lashuk, E. Ovtchinnikov, Block locally optimal preconditioned eigenvalue solvers (BLOPEX) in HYPRE and PETSc, *SIAM J. Sci. Comput.* 29 (5) (2007) 2224–2239.
- [86] A. Knyazev, private communication, 2013.
- [87] J. Poulson, B. Marker, R.A. van de Geijn, J.R. Hammond, N.A. Romero, Elemental: a new framework for distributed memory dense matrix computations, *ACM Trans. Math. Softw.* 39 (2) (2013) 13, 24 pp.
- [88] L.S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, R.C. Whaley, *ScalAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.
- [89] J. Choi, J.J. Dongarra, R. Pozo, D.W. Walker, Scalapack: a scalable linear algebra library for distributed memory concurrent computers, in: *Fourth Symposium on the Frontiers of Massively Parallel Computation*, IEEE, 1992, pp. 120–127.
- [90] R.A. van de Geijn, *Using LAPACK: Parallel Linear Algebra Package*, The MIT Press, 1997.
- [91] P. Alpatov, G. Baker, C. Edwards, J. Gunness, J. Overfelt, R. van de Geijn, Y.-J. Wu, PLAPACK: parallel linear algebra package design overview, in: *Proceedings of the 1997 ACM/IEEE Conference on Supercomputing*, CDROM, ACM, 1997, pp. 1–16.
- [92] M.D. Schatz, J. Poulson, R.A. van de Geijn, Scalable universal matrix multiplication algorithms: 2d and 3d variations on a theme, *ACM Trans. Math. Softw.* (2015), submitted for publication.
- [93] A.S. Banerjee, Density functional methods for objective structures: theory and simulation schemes, PhD thesis, University of Minnesota, Minneapolis, 2013.
- [94] M. Gu, Single- and multiple-vector iterations, in: Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst (Eds.), *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.
- [95] U. Stephan, D.A. Drabold, R.M. Martin, Improved accuracy and acceleration of variational order- n electronic-structure computations by projection techniques, *Phys. Rev. B* 58 (20) (1998) 13472.
- [96] C. Bekas, Y. Saad, M.L. Tiago, J.R. Chelikowsky, Computing charge densities with partially reorthogonalized Lanczos, *Comput. Phys. Commun.* 171 (3) (2005) 175–186.
- [97] S. Baroni, P. Giannozzi, Towards very large-scale electronic-structure calculations, *Europhys. Lett.* 17 (6) (1992) 547.
- [98] Y. Zhou, J.R. Chelikowsky, Y. Saad, Chebyshev-filtered subspace iteration method free of sparse diagonalization for solving the Kohn–Sham equation, *J. Comput. Phys.* 274 (2014) 770–782.
- [99] J.W. Daniel, W.B. Gragg, L. Kaufman, G. Stewart, Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization, *Math. Comput.* 30 (136) (1976) 772–795.
- [100] D.G. Anderson, Iterative procedures for nonlinear integral equations, *J. ACM* 12 (4) (1965) 547–560.
- [101] C.G. Broyden, A class of methods for solving nonlinear simultaneous equations, *Math. Comput.* 19 (92) (1965) 577–593.
- [102] P. Pulay, Convergence acceleration of iterative sequences: the case of scf iteration, *Chem. Phys. Lett.* 73 (2) (1980) 393–398.
- [103] D. Johnson, Modified Broyden's method for accelerating convergence in self-consistent calculations, *Phys. Rev. B* 38 (18) (1988) 12807.
- [104] K.N. Kudin, G.E. Scuseria, E. Cancès, A black box self consistent field convergence algorithm: one step closer, *J. Chem. Phys.* 116 (2002) 8255.
- [105] H. Fang, Y. Saad, Two classes of multisection methods for nonlinear acceleration, *Numer. Linear Algebra Appl.* 16 (3) (2009) 197–221.

- [106] J. Kohanoff, *Electronic Structure Calculations for Solids and Molecules: Theory and Computational Methods*, 1st edition, Cambridge University Press, 2006.
- [107] G. Kresse, J. Hafner, Ab initio molecular dynamics for liquid metals, *Phys. Rev. B* 47 (1993) 558–561.
- [108] R.P. Brent, *Algorithms for Minimization Without Derivatives*, Courier Dover Publications, 1973.
- [109] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, F. Rossi, *GNU Scientific Library Reference Manual*, 3rd edition, Network Theory Ltd., 2009.
- [110] G.H. Golub, J.H. Welsch, Calculation of Gauss quadrature rules, *Math. Comput.* 23 (106) (1969) 221–230.
- [111] S. Goedecker, M. Teter, J. Hutter, Separable dual-space Gaussian pseudopotentials, *Phys. Rev. B* 54 (1996) 1703–1710.
- [112] C. Fiolhais, J.P. Perdew, S.Q. Armster, J.M. MacLaren, M. Braszewska, Dominant density parameters and local pseudopotentials for simple metals, *Phys. Rev. B* 51 (20) (1995) 14001.
- [113] H. Chen, X. Gong, L. He, Z. Yang, A. Zhou, Numerical analysis of finite dimensional approximations of Kohn–Sham models, *Adv. Comput. Math.* 38 (2) (2013) 225–256.
- [114] S. Kotochigova, Z.H. Levine, E.L. Shirley, M. Stiles, C.W. Clark, Local-density-functional calculations of the energy of atoms, *Phys. Rev. A* 55 (1) (1997) 191.
- [115] P. Suryanarayana, K. Bhattacharya, M. Ortiz, A mesh-free convex approximation scheme for Kohn–Sham density functional theory, *J. Comput. Phys.* 230 (13) (2011) 5226–5238.
- [116] F. Nogueira, C. Fiolhais, J. He, J.P. Perdew, A. Rubio, Transferability of a local pseudopotential based on solid-state electron density, *J. Phys. Condens. Matter* 8 (3) (1996) 287.
- [117] C. Fiolhais, J.P. Perdew, S.Q. Armster, J.M. MacLaren, M. Braszewska, Dominant density parameters and local pseudopotentials for simple metals, *Phys. Rev. B* 53 (19) (1996) 13193 (Erratum).
- [118] C. Hartwigsen, S. Goedecker, J. Hutter, Relativistic separable dual-space Gaussian pseudopotentials from H to Rn, *Phys. Rev. B* 58 (1998) 3641–3662.
- [119] J. Fang, X. Gao, A. Zhou, A Kohn–Sham equation solver based on hexahedral finite elements, *J. Comput. Phys.* 231 (8) (2012) 3166–3180.
- [120] Q.-B. Yan, X.-L. Sheng, Q.-R. Zheng, L.-Z. Zhang, G. Su, Family of boron fullerenes: general constructing schemes, electron counting rule, and ab initio calculations, *Phys. Rev. B* 78 (20) (2008) 201401.
- [121] W.H. Green, S.M. Gorun, G. Fitzgerald, P.W. Fowler, A. Ceulemans, B.C. Titeca, Electronic structures and geometries of C₆₀ anions via density functional calculations, *J. Phys. Chem.* 100 (36) (1996) 14892–14898.
- [122] J. Murray, K. Sen, *Molecular Electrostatic Potentials: Concepts and Applications*, 1st edition, *Theor. Comput. Chem.*, Elsevier, 1996.
- [123] O. Gunnarsson, J. Harris, R. Jones, Density functional theory and molecular bonding. I. First-row diatomic molecules, *J. Chem. Phys.* 67 (9) (1977) 3970–3979.
- [124] R.M. Dickson, A.D. Becke, Local density-functional polarizabilities and hyperpolarizabilities at the basis-set limit, *J. Phys. Chem.* 100 (40) (1996) 16105–16108.
- [125] J. Guan, P. Duffy, J.T. Carter, D.P. Chong, K.C. Casida, M.E. Casida, M. Wrinn, Comparison of local-density and Hartree–Fock calculations of molecular polarizabilities and hyperpolarizabilities, *J. Chem. Phys.* 98 (6) (1993) 4753–4765.
- [126] D. Chadi, M.L. Cohen, Special points in the Brillouin zone, *Phys. Rev. B* 8 (12) (1973) 5747.
- [127] R.A. Evarestov, V.P. Smirnov, Special points of the Brillouin zone and their use in the solid state theory, *Phys. Status Solidi B* 119 (1) (1983) 9–40.
- [128] A.R. Edmonds, *Angular Momentum in Quantum Mechanics*, fourth printing edition, Princeton University Press, 1996.
- [129] G.B. Folland, *A Course in Abstract Harmonic Analysis*, 1st edition, *AMS/IP Stud. Adv. Math.*, Taylor & Francis, 1994.
- [130] A.O. Barut, R. Raczka, *Theory of Group Representations and Applications*, second revised edition, World Scientific Publishing Company, 1986.
- [131] R.D. James, Objective structures, *J. Mech. Phys. Solids* 54 (11) (2006) 2354–2390.