UC Berkeley

Title

Two-Level Chebyshev Filter Based Complementary Subspace Method: Pushing the Envelope of Large-Scale Electronic Structure Calculations

Authors

Banerjee, Amartya S
Lin, Lin
Suryanarayana, Phanish
et al.

Two-level Chebyshev filter based complementary subspace method: pushing the envelope of large-scale electronic structure calculations

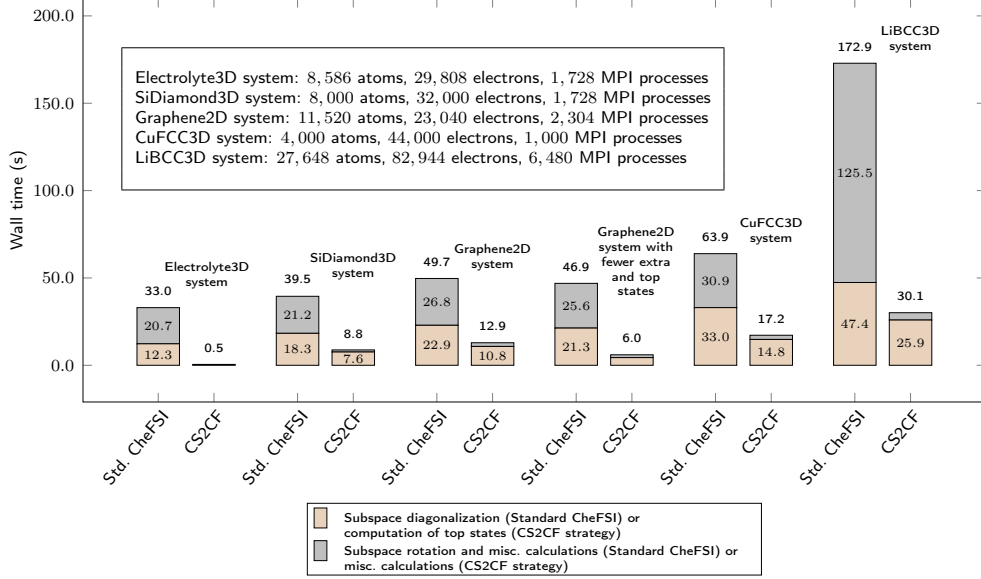Amartya S. Banerjee,*,† Lin Lin,*,‡,† Phanish Suryanarayana,*,¶ Chao Yang,*,† and

John E. Pask*,§

†*Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, U.S.A*

‡*Department of Mathematics, University of California, Berkeley, California 94720, U.S.A*

¶*College of Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332, U.S.A*

§*Physics Division, Lawrence Livermore National Laboratory, Livermore, California 94550, U.S.A*

E-mail: asb@lbl.gov; linlin@math.berkeley.edu; phanish.suryanarayana@ce.gatech.edu; cyang@lbl.gov; pask1@llnl.gov

Electrolyte3D system: 8,586 atoms, 29,808 electrons, 1,728 MPI processes
SiDiamond3D system: 8,000 atoms, 32,000 electrons, 1,728 MPI processes
Graphene2D system: 11,520 atoms, 23,040 electrons, 2,304 MPI processes
CuFCC3D system: 4,000 atoms, 44,000 electrons, 1,000 MPI processes
LiBCC3D system: 27,648 atoms, 82,944 electrons, 6,480 MPI processes

Subspace diagonalization (Standard CheFSI) or computation of top states (CS2CF strategy)
Subspace rotation and misc. calculations (Standard CheFSI) or misc. calculations (CS2CF strategy)

(Graphic for the Table of Contents / Graphical Abstract)

Abstract

We describe a novel iterative strategy for Kohn-Sham density functional theory calculations aimed at large systems (> 1000 electrons), applicable to metals and insulators alike. In lieu of explicit diagonalization of the Kohn-Sham Hamiltonian on every self-consistent field (SCF) iteration, we employ a two-level Chebyshev polynomial filter based complementary subspace strategy to: 1) compute a set of vectors that span the occupied subspace of the Hamiltonian; 2) reduce subspace diagonalization to just partially occupied states; and 3) obtain those states in an efficient, scalable manner via an inner Chebyshev-filter iteration. By reducing the necessary computation to just partially occupied states, and obtaining these through an inner Chebyshev iteration, our approach reduces the cost of large metallic calculations significantly, while eliminating subspace diagonalization for insulating systems altogether. We describe the implementation of the method within the framework of the Discontinuous Galerkin (DG) electronic structure method and show that this results in a computational scheme that can effectively tackle bulk and nano systems containing tens of thousands of electrons, with chemical accuracy, within a few minutes or less of wall clock time per SCF iteration on large-scale computing platforms. We anticipate that our method will be

2

instrumental in pushing the envelope of large-scale *ab initio* molecular dynamics. As a demonstration of this, we simulate a bulk silicon system containing $8,000$ atoms at finite temperature, and obtain an average SCF step wall time of 51 seconds on $34,560$ processors; thus allowing us to carry out 1.0 ps of *ab initio* molecular dynamics in approximately 28 hours (of wall time).

1   Introduction

Kohn-Sham density functional theory (KS-DFT)[1,2] is among the most widely used approaches in the computational chemistry, condensed matter, and materials research communities. Over the years, KS-DFT has provided unparalleled insights and robust predictions for the gamut of materials properties[3–5], as a result of which, much research has been devoted to enable calculations of ever larger and more complex systems[6–10].

Conventionally, the Kohn-Sham equations are solved self-consistently, wherein the linear eigenvalue problem arising from discretization in a chosen basis is solved on each self-consistent field (SCF) iteration until a fixed point is reached in the electronic density or potential; whereupon energies, forces, and other quantities of interest are computed.[2] Solution of this eigenvalue problem via direct or iterative diagonalization methods scales cubically with the number of electronic states in the system (and hence, also cubically with the number of atoms). The computational cost of this procedure can become prohibitive, however, as the system size grows beyond a few thousand electronic states. This has led to the development of computational techniques which scale more favorably with respect to the number of electronic states in the system, by avoiding explicit diagonalization of the Kohn-Sham Hamiltonian.[11–18] However, these work best on insulating systems with substantial band gaps[18,19], metallic systems with low dimension (such as in the case of the pole expansion and selected inversion (PEXSI) method)[20,21], or systems at high electronic temperature.[16,17] Consequently, while impressive large-scale Kohn-Sham calculations of various insulating systems have been demonstrated by several groups[6–8,22], large-scale, well converged, chemically

3

accurate calculations of realistic semiconducting or metallic systems have appeared only rarely[16,18,19,23]. Many of the aforementioned methods rely on the nearsightedness principle[24] for obtaining the electron density from the Kohn-Sham Hamiltonian in an efficient manner. This presents practical issues while dealing with bulk metallic systems at moderate electronic temperatures (up to a few thousand Kelvins). The relatively slow decay of the density matrix associated with such systems (even for the case of simple metals like aluminum) often results in a computational method having a favorable algorithmic scaling but relatively large prefactor.[25] In practical treatments of large bulk metallic systems containing several thousands of atoms, this is likely to result in a larger computational wall time per SCF iteration with the use of these methods[1] than with the use of diagonalization based methods[25].

In this work, we take a different approach to make Kohn-Sham calculations of large, metallic systems more computationally feasible. Our approach is to revert to the use of (cubically scaling) partial diagonalization of the Kohn-Sham Hamiltonian via iterative methods, but to ensure that the resulting computational strategy has a low prefactor (in the sense of computational complexity estimates) and good parallel scaling efficiency. In practice, with the use of a sufficiently large number of processors, this then enables larger systems sizes and/or shorter simulation wall times than attainable heretofore.

With the aforementioned goal in mind, we have focused on two principal strategies for achieving it. The first of these has been to ensure that we use a discretization scheme that is highly efficient. Specifically, we utilize basis functions which can produce systematically improvable, high quality numerical results while keeping the number of basis functions per atom required for doing so, small. To this end, we employ so called Adaptive Local Basis (ALB) functions[26] that are generated on the fly on every SCF step during the course of Kohn-Sham calculations. These basis functions are able to capture the local materials physics in electronic structure calculations, and when used in conjunction with the interior penalty Discontinuous Galerkin (DG) formalism for the Kohn-Sham equations, they allow high quality

---

[1]Aggressive truncation of the density matrix entries can also lead to SCF iteration instabilities, further slowing calculations with these methods[16,25].

energies and Hellman-Feynman forces to be computed with only a few tens of basis functions per atom[26–28]. The DG approach for solving the Kohn-Sham equations using ALB functions has been incorporated into a massively parallel software package, DGDFT[20,29]. The second strategy has been to employ a well suited iterative diagonalization strategy for the discretized Kohn-Sham Hamiltonian. To this end, we have made use of Chebyshev polynomial filtered subspace iteration (CheFSI)[28]. For a number of reasons[28], this technique works particularly well within the framework of DGDFT — both in terms of computational wall times and overall parallel scaling efficiency. We have recently demonstrated that the CheFSI technique significantly outperforms existing alternatives in carrying out SCF iterations in DGDFT, particularly in the context of large bulk systems[28].

Our experiences with the DGDFT-CheFSI strategy have revealed that the principal computational bottleneck in treating large, metallic systems occurs in the Rayleigh-Ritz process[30] to obtain approximate eigenvalues and vectors in each SCF iteration. In the context of the DGDFT-CheFSI methodology, this process is generally as follows[28]:

1. Compute a basis spanning the occupied subspace using a Chebyshev polynomial filter.

2. Orthonormalize the basis.

3. Project the DG Hamiltonian matrix onto the subspace.

4. Diagonalize the projected Hamiltonian to obtain Ritz values and projected vectors.

5. Rotate the basis according to projected vectors to obtain Ritz vectors.

The Ritz values and vectors so obtained then provide approximate eigenvalues and vectors of the DG Hamiltonian in each SCF iteration of the Kohn-Sham solution. In the following, we focus mainly on steps (4) and (5) of the process, which we shall refer to as *subspace diagonalization* and *subspace rotation*, respectively. In the absence of fractionally occupied states (i.e., for insulating systems), these steps can be avoided altogether (see Section 2.1.1) and in that case, the computational bottleneck arises from the orthonormalization of the basis that spans the occupied subspace. Indeed, due to their focus on insulating systems, a number

of previous authors[12,31–33] have focused on reducing or eliminating this orthonormalization cost, instead of the cost associated with the aforementioned projected Hamiltonian eigenvalue problem. For non-insulating (e.g., metallic and semiconducting) systems, however, fractional occupation numbers have to be computed and the projected Hamiltonian eigenvalue problem cannot be directly avoided.[2] In this situation, detailed wall time studies (e.g., Section 3.2) reveal that the most significant hindrance to pushing the computational efficiency of the DGDFT-CheFSI approach lies in the subspace diagonalization and subspace rotation steps of the Rayleigh-Ritz process. While the orthonormalization cost can become significant for large problems, its contribution to the simulation wall time is less than that of the above steps, and its parallel scalability is appreciably better. This provides the incentive to devise a computational strategy that reduces or eliminates the computational cost incurred due to the subspace diagonalization and subspace rotation steps in the DGDFT-CheFSI approach.

In this work, we formulate and implement a two-level Chebyshev polynomial filter based complementary subspace strategy to address the above issues. In this methodology, only the relatively few fractionally occupied states of the projected Hamiltonian, and not those fully occupied or empty, are calculated, thus reducing the computational cost of the subspace diagonalization and subspace rotation steps significantly. Moreover, exploiting the spectral properties of the projected Hamiltonian, we employ CheFSI iterations to obtain the fractionally occupied states, thereby yielding a highly efficient iterative scheme that uses Chebyshev polynomial filtering on two levels. We refer to the resulting computational methodology as

---

[2]In principle, it is possible to employ techniques such as Fermi Operator Expansion[11] (FOE) to arrive at the projected density matrix, starting from the projected Hamiltonian, without going through the intermediate step of computing the eigenstates of the projected Hamiltonian[34]. However, there are a number of challenges in taking such an approach. The projected Hamiltonian is in general a dense matrix, especially for metallic systems at moderate electronic temperatures[25] (e.g., less than 3000 Kelvin). Furthermore, the FOE technique requires a large number of matrix-matrix products to be computed[34]. Moreover, the method becomes less efficient at lower (e.g., ambient) electronic temperatures due to the increased number of terms in the expansion. Finally, an efficient application of the FOE technique requires a localization procedure to be carried out, which can be computationally expensive[34] and can also lead to SCF convergence issues[35]. Due to these factors, application of the FOE technique is expected to be significantly more expensive than the methodology developed in this work (based on computing only the topmost states of the projected Hamiltonian using iterative methods, and so involving only matrix-vector products). Indeed, for these reasons, it is expected that such application of the FOE technique would be more expensive than direct computation of the eigenstates of the projected Hamiltonian, and so we do not pursue this direction here.

CS2CF, i.e., *Complementary Subspace strategy with 2 levels of Chebyshev Filtering.*

The idea of exploiting completeness to reduce subspace computations to just fractionally occupied states was presented recently by Michaud-Rioux et al.[36] in their partial Rayleigh-Ritz method for large-scale Kohn-Sham calculations, as implemented in the RESCU Matlab electronic structure code.[3] The idea has also been exploited previously in particle-hole duality formulations to accelerate density matrix computations in $O(N)$ electronic structure methods[39–41]. In the context of subspace diagonalization, in line with the key idea of obtaining a large subspace from its much smaller complement, we refer to the approach as the *Complementary Subspace* (CS) strategy here.

Once the subspace computation has been reduced to just fractionally occupied states, it is then crucial to obtain these as efficiently as possible. As we detail in Section 2.2.2, to accomplish this, we exploit key characteristics of the subspace eigenvalue problem to obtain the fractionally occupied states both efficiently and scalably via low-order CheFSI iterations. This strategy is particularly well suited to large-scale parallel implementation and, as we show, integrates particularly well within the framework of the massively parallel DGDFT code. Consequently, it allows us to attack metallic systems containing tens of thousands of electrons in a few minutes or less of wall time per SCF iteration. Finally, all results reported here are computed to chemical accuracy (i.e., energy and force errors below 10^{-3} Ha/atom and 10^{-3} Ha/Bohr, respectively), as typical in production simulations. This has allowed us to carry out accurate, energy-conserving *ab initio* molecular dynamics simulations

---

[3]There are significant differences, however, between the partial Rayleigh-Ritz method of Michaud-Rioux et al.[36] and the CS2CF approach presented here. Specifically, in their implementation of the partial Rayleigh-Ritz method, the authors have employed direct diagonalization and/or the iterative LOBPCG method[37] to obtain the fractionally occupied states, neither of which appears to yield significant savings in the large-scale parallel context. Furthermore, their formulation, as described, is limited to the calculation of energies alone and implementation is limited to MATLAB[38]. In contrast, the use of an inner-level of CheFSI for computing the fractionally occupied states turns out to be particularly well suited to large-scale parallel implementation for a number of reasons (see Section 2.2.2) and increases the computational efficiency of the procedure dramatically. Additionally, the CS2CF strategy integrates particularly well within the framework of the massively parallel DGDFT code (see Section 2.3). Consequently, it allows us to attack metallic systems containing tens of thousands of electrons in a few minutes or lesser of wall time per SCF step (See Section 3). Finally, all the results reported here are done with full chemical accuracy (i.e., the discretization parameters were chosen to result in well converged energies and forces), which enables us to carry out *ab initio* molecular dynamics simulations of large scale non-insulating systems conveniently.

of systems with many thousands of atoms in a few minutes per MD step. Without the use of the CS2CF strategy, the advantages afforded by the ALB discretization, and the highly efficient, massively parallel DGDFT code, this would not be possible.

It is important to note that the utility of the complementary subspace strategy (and in particular, the CS2CF methodology) is not restricted to the DGDFT code. *Any* other Kohn-Sham code can benefit from this methodology as well, although as mentioned above (also see Section 2.3) the CS2CF strategy integrates well and performs particularly efficiently within DGDFT. Overall, we view this work as the development of a Kohn-Sham equation solution strategy that has been well tuned to minimize wall times in practical simulation scenarios of large systems. In our view, it is an important step towards a robust and efficient methodology for carrying out *ab initio* molecular dynamics of large metallic and semiconducting systems.

The remainder of this paper is organized as follows. In Section 2, we outline the the mathematical formulation of the complementary subspace strategy, efficient two-level Chebyshev filter based solution, and large-scale parallel implementation in the context of the discontinuous Galerkin electronic structure method. In Section 3, we present results for a range of systems, from insulating to semiconducting to metallic, and comparisons with existing methods. We conclude and comment on future research directions in Section 4.

2   Methodology

We describe the formulation of the complementary subspace strategy as well as its implementation within the adaptive local basis set based discontinuous Galerkin electronic structure method (specifically, the DGDFT code[20]) in this section. For simplicity, we consider Γ-point calculations of non-spin-polarized periodic systems, as is typical in large-scale *ab initio* molecular dynamics simulations, although this assumption is not required in what follows.

2.1 Formulation of complementary subspace strategy

2.1.1 Density matrix and projected density matrix

We consider a system with a discretized Kohn-Sham Hamiltonian matrix H of size $N_b \times N_b$, with N_b denoting the total number of (orthonormal) basis functions used for the discretization. The Hamiltonian is a function of the real-space electron density $\rho(x)$ from the discretized Hamiltonian that must be determined iteratively in a self-consistent field (SCF) iteration.

The conventional procedure for achieving this is through the intermediate computation of the (discretized) Kohn-Sham orbitals, i.e., the eigenvectors of H.[2,42] Within the above setting, each Kohn-Sham orbital is a $N_b \times 1$ sized real valued vector. For a system containing N_e electrons (per unit cell), the lowest N_s eigenstates of H need to be computed in each SCF step. For an insulator, each orbital is doubly occupied, and N_s can be taken as $N_e/2$. In contrast, for a metallic system, it is customary to use $N_s = N_e/2 + N_x$, where N_x denotes extra states that are used to accommodate fractional occupations.[36,43,44] It usually suffices to take N_x to be about $5 - 10\%$ of $N_e/2$ while dealing with electronic temperatures up to a few thousand Kelvins. In this case, the occupation numbers associated with the states lying beyond the lowest N_s can be conveniently set to 0 without compromising the accuracy of the solution (ground-state energies and forces, for example) or aggravating SCF convergence.

The computation of the lowest N_s eigenvalues $\{\epsilon_i\}_{i=1}^{N_s}$ and the corresponding eigenvectors $\{\psi_i\}_{i=1}^{N_s}$ of H can be carried out through the use of direct or iterative eigensolvers. Subsequent to the computation of the eigenstates, the occupation fractions $\{f_i\}_{i=1}^{N_s}$ (with $0 \leq f_i \leq 1$) can be computed from the Fermi-Dirac function[45]

$$f_i = f_F(\epsilon_i), \text{ with } f_F(\epsilon) = \frac{1}{1 + \exp\left(\frac{\epsilon - \epsilon_F}{k_B \Theta_e}\right)}, \tag{1}$$

where Θ_e is the electronic temperature, k_B denotes the Boltzmann constant, and the Fermi

level ϵ_F can be determined by solving the constraint equation

$$2 \sum_{i=1}^{N_s} f_i = N_e \, . \tag{2}$$

The use of fractional occupation (also known as smearing)[18,43,44] allows us to overcome numerical difficulties associated with possible degeneracy of eigenstates near ϵ_F.

Using the results from the above computations, the (discretized) density matrix (also referred to as the Fermi matrix at finite electronic temperature) of the system can be calculated. This $N_b \times N_b$ sized matrix is defined as

$$P = f_F(H) \, , \tag{3}$$

and using the fact that $f(\epsilon_i) = 0$ for $i > N_s$, it can be rewritten using the eigenvectors of H as

$$P = \sum_{i=1}^{N_s} f_i \, \psi_i \, \psi_i^T \, . \tag{4}$$

Denoting the collection of the eigenvectors $\{\psi_i\}_{i=1}^{N_s}$ as the $N_b \times N_s$ matrix X, and the $N_s \times N_s$ diagonal matrix of occupation numbers as $\mathfrak{F}$ (i.e., $\mathfrak{F}_{i,i} = f_i$ for $i = 1, \ldots, N_s$), a more compact matrix form of the above expression (i.e., Eq. 4) is

$$P = X \, \mathfrak{F} \, X^T \, . \tag{5}$$

The matrix P contains all the information required for progressing with the SCF iterations — in particular, if the basis functions used for the discretization are denoted as $\left\{e_j(x)\right\}_{j=1}^{N_b}$,

then the real-space electron density can be expressed[4] using the matrix entries of P as

$$\rho(x) = 2 \sum_{j=1}^{N_b} \sum_{j'=1}^{N_b} P_{j,j'} \, e_j(x) \, e_{j'}(x) \,. \tag{6}$$

In the process of computing the density matrix P, it is often simpler to compute an alternate set of orthonormal vectors $\{\phi_i\}_{i=1}^{N_s}$ that span the same subspace as the eigenvectors (i.e., the occupied subspace). If the collection of these alternate vectors is expressed as an $N_b \times N_s$ matrix Y, there must exist an orthogonal $N_s \times N_s$ matrix Q such that $X = YQ$, and Eq. (5) then takes the form

$$P = Y \left(Q \, \mathfrak{F} \, Q^T \right) Y^T \,. \tag{7}$$

The $N_s \times N_s$ matrix $\tilde{P} = Q \, \mathfrak{F} \, Q^T$ will be referred to as the *projected density matrix*. Equation (7) indicates that the density matrix P may be computed using alternative vectors $\{\phi_i\}_{i=1}^{N_s}$ if the projected density matrix is available along with the vectors $\{\phi_i\}_{i=1}^{N_s}$.

A straightforward way of computing such a set of alternate orthonormal vectors is through the use of Chebyshev polynomial filtering followed by explicit orthonormalization.[30] Specifically, we may start with a $N_b \times N_s$ block of linearly independent vectors Y_0, and apply a Chebyshev polynomial filter matrix $p_m(H)$ to Y_0. The filter polynomial $p_m(\cdot)$ can be specifically scaled and a sufficiently high filter order m can be chosen so that the eigenvectors $\{\psi_i\}_{i=1}^{N_s}$ are amplified in the resulting filtered vectors $Y_1 = p_m(H) Y_0$.[47–49] To avoid linear dependencies, we may then orthonormalize the vector block Y_1. The resulting set of orthonormal vectors will (approximately) span the occupied subspace. This strategy has been combined with subspace iteration techniques for use in various electronic structure codes[28,46–48,50–52] and it can successfully deal with metallic as well as insulating systems.

---

[4]Alternately, the eigenvectors can be expressed in real space as $\psi_i(x) = \sum_{j=1}^{N_b} \psi_{i,j} \, e_j(x)$, whereupon the electron density can be obtained as $\rho(x) = 2 \sum_{i=1}^{N_s} f_i \left[\psi_i(x) \right]^2$. This approach is conventionally used, for example, in planewave and other such codes[46] employing global basis sets, but is not as suitable in the context of localized basis sets[28], as we employ here.

We note that a special situation arises when the system in question is an insulator. In this case, the matrix of occupation numbers $\mathfrak{F}$ is the identity matrix, and so Eq. 7 reduces to

$$P = Y Y^T,\tag{8}$$

as Q is an orthogonal matrix. Thus, for an insulating system, we only need a way of computing a set of orthonormal vectors that span the occupied subspace in order to compute the density matrix P. For a metallic system, additional work is needed to compute the projected density matrix.

2.1.2 Direct computation of the projected density matrix

Considering the expression for the projected density matrix $\tilde{P} = Q\,\mathfrak{F}\,Q^T$, we see that the evaluation of this expression requires the computation of the occupation numbers $\{f_i = f_F(\epsilon_i)\}_{i=1}^{N_s}$ as well as the matrix Q. These quantities can be computed by carrying out an eigenvalue decomposition of the *projected Hamiltonian matrix*, i.e., the $N_s \times N_s$ matrix $\tilde{H} = Y^T H Y$. The occupation numbers can be computed using the eigenvalues of $\tilde{H}$ since these are the same as the lowest N_s eigenvalues of H (i.e., $\{\epsilon_i\}_{i=1}^{N_s}$). Furthermore, the eigenvectors of $\tilde{H}$ are the columns of the matrix Q.

To verify this, we first write down the eigendecomposition of H (for the lowest N_s states) as $H X = X \Lambda$. Next, using $X = Y Q$, we get

$$H Y Q = Y Q \Lambda.\tag{9}$$

Premultiplying with Y^T and using the orthonormality of the column vectors in Y, we get

$$\left(Y^T H Y\right) Q = \tilde{H} Q = Q \Lambda.\tag{10}$$

Additionally, the expressions $\tilde{P} = Q \mathfrak{F} Q^T$ and $\tilde{H} = Q \Lambda Q^T$ allow us to interpret the projected density matrix in terms of the projected Hamiltonian matrix as $\tilde{P} = f_F(\tilde{H})$.

In the context of Krylov subspace projection methods,[30,53] the steps involving the construction of the projected Hamiltonian matrix, computation of its eigendecomposition, and computation of (approximate) eigenvectors of H using the expression $X = Y Q$ constitute the Rayleigh-Ritz process, wherein the last step corresponds to subspace rotation. As the system size grows, the dimension of the Hamiltonian N_b and number of states N_s grow as well. Correspondingly, the eigendecomposition and subspace rotation steps begin to consume more and more computation time, thus making it infeasible to directly compute the projected density matrix as described above. Instead, a complementary subspace strategy may be formulated to mitigate these issues, as we now describe.

2.1.3 Complementary subspace computation of the projected density matrix

In light of the above discussion, it appears that for a generic system (i.e., one with some degree of fractional occupation), we require all the N_s eigenstates of $\tilde{H}$ to be computed on every SCF iteration in order to progress with the SCF iterations. A crucial observation, however, is that for electronic temperatures typically encountered in practice (e.g., $\Theta_e \lesssim 3000$ Kelvin), a large majority of the occupation numbers $\{f_i\}_{i=1}^{N_s}$ are equal to 1. We denote states 1 through N_1 as those with occupation numbers equal to 1. The remaining states, from $N_1 + 1$ through N_s, have occupations numbers less than 1. Let N_t be the number of these fractionally occupied states, i.e., $N_t = N_s - N_1$. Denoting the eigenvectors of the projected density matrix as $\{\tilde{\psi}_i\}_{i=1}^{N_s}$ (the columns of the matrix Q), we may rewrite the expression for

the projected density matrix as

$$\tilde{P} = \sum_{i=1}^{N_s} f_i \, \tilde{\psi}_i \, \tilde{\psi}_i^T$$

$$= \sum_{i=1}^{N_1} \tilde{\psi}_i \, \tilde{\psi}_i^T + \sum_{i=N_1+1}^{N_s} f_i \, \tilde{\psi}_i \, \tilde{\psi}_i^T$$

$$= \sum_{i=1}^{N_1} \tilde{\psi}_i \, \tilde{\psi}_i^T + \sum_{i=N_1+1}^{N_s} \tilde{\psi}_i \, \tilde{\psi}_i^T - \sum_{i=N_1+1}^{N_s} \tilde{\psi}_i \, \tilde{\psi}_i^T + \sum_{i=N_1+1}^{N_s} f_i \, \tilde{\psi}_i \, \tilde{\psi}_i^T$$

$$= \sum_{i=1}^{N_s} \tilde{\psi}_i \, \tilde{\psi}_i^T - \sum_{i=N_1+1}^{N_s} \tilde{\psi}_i \, \tilde{\psi}_i^T + \sum_{i=N_1+1}^{N_s} f_i \, \tilde{\psi}_i \, \tilde{\psi}_i^T \tag{11}$$

$$= \tilde{\mathcal{I}} - \sum_{i=N_1+1}^{N_s} (1 - f_i) \, \tilde{\psi}_i \, \tilde{\psi}_i^T . \tag{12}$$

In Eq. 12 above, $\tilde{\mathcal{I}}$ denotes the identity matrix of dimension $N_s \times N_s$. That the first term of Eq. 11 is the identity matrix follows from the fact that the vectors $\{\tilde{\psi}_i\}_{i=1}^{N_s}$ are the eigenvectors of $\tilde{H}$, a symmetric matrix, and so form a resolution of the identity.

The above expression suggests that if the N_t top eigenvectors $\tilde{\psi}_i$ and corresponding occupation numbers f_i are known, the projected density matrix $\tilde{P}$ may be computed. Thus, instead of determining the full $N_s \times N_s$ set of vectors, we need to determine only an extremal block of vectors (of dimension $N_s \times N_t$), corresponding to the states $i = N_1 + 1$ to N_s.

To compute the corresponding occupation numbers, we rewrite the equation $2\sum_{i=1}^{N_s} f_i = N_e$ as

$$2\Big(\sum_{i=1}^{N_1} f_i + \sum_{i=N_1+1}^{N_s} f_i\Big) = N_e \implies \sum_{i=N_1+1}^{N_s} f_i = N_e/2 - N_1. \tag{13}$$

The above algebraic equation may be solved for the Fermi level ϵ_F and occupation numbers $\{f_i\}_{i=N_1+1}^{N_s}$.

Once the projected density matrix $\tilde{P}$ has been obtained, the full (i.e., $N_b \times N_b$) density matrix can be obtained as $P = Y\tilde{P}Y^T$. To simplify this further, let us denote by $\tilde{C}_{\tilde{\mathfrak{F}}}$ the

$N_s \times N_t$ matrix consisting of the vectors $\{\sqrt{1-f_i}\,\tilde{\psi}_i\}_{i=1}^{N_t}$ (i.e., each of the top eigenvectors of $\tilde{H}$ scaled by the quantity $\sqrt{1-f_i}$). Then Eq.12 can be written in terms of $\tilde{C}_{\mathfrak{F}}$ as

$$\tilde{P} = \tilde{I} - \tilde{C}_{\mathfrak{F}}\tilde{C}_{\mathfrak{F}}^T \tag{14}$$

whereupon we obtain

$$P = Y\tilde{P}Y^T = YY^T - Y\tilde{C}_{\mathfrak{F}}\tilde{C}_{\mathfrak{F}}^T Y^T$$
$$= YY^T - (Y\tilde{C}_{\mathfrak{F}})(Y\tilde{C}_{\mathfrak{F}})^T. \tag{15}$$

As we explain later, this expression is particularly easy to evaluate in a localized basis set. In particular, within the DGDFT code, evaluation of the diagonal portions of the above expression can be carried out in a manner that avoids interprocess communication. This is sufficient for evaluating the real-space electron density and proceeding with SCF iterations.

Ground-state KS-DFT calculations also require computation of the band energy $E_b = \sum_{i=1}^{N_s} f_i\,\epsilon_i$. Since only the fractionally occupied states are in the complementary subspace scheme, we rewrite this as

$$E_b = 2\left(\sum_{i=1}^{N_s} f_i\,\epsilon_i\right) = 2\left(\sum_{i=1}^{N_1} f_i\,\epsilon_i + \sum_{i=N_1+1}^{N_s} f_i\,\epsilon_i\right)$$
$$= 2\left(\sum_{i=1}^{N_1}\epsilon_i + \sum_{i=N_1+1}^{N_s} f_i\,\epsilon_i\right) = 2\left(\sum_{i=1}^{N_s}\epsilon_i + \sum_{i=N_1+1}^{N_s} f_i\,\epsilon_i - \sum_{i=N_1+1}^{N_s}\epsilon_i\right)$$
$$= 2\left(\mathrm{Tr}(\tilde{H}) - \sum_{i=N_1+1}^{N_s}(1-f_i)\,\epsilon_i\right). \tag{16}$$

Thus, the trace of the projected Hamiltonian matrix, the top eigenvalues, and their corresponding occupation numbers are sufficient to compute the band energy.

The electronic entropy can be obtained from just the fractionally occupied states as well.

The electronic entropy is given by

$$S = -2k_B \sum_{i=1}^{N_s} f_i \log f_i + (1 - f_i) \log (1 - f_i) . \tag{17}$$

By inspection, we see that the contribution of a state i to the electronic entropy goes to zero as the occupation number for that state f_i goes to 0 or 1. Hence, within the complementary subspace scheme, only the contribution of the fractionally occupied states is considered. This allows the simplification of Eq. 17 to

$$S = -2k_B \sum_{i=N_1+1}^{N_s} f_i \log f_i + (1 - f_i) \log (1 - f_i) , \tag{18}$$

which can be readily computed.

2.2 Computation of top states

We now discuss strategies for computing the N_t topmost occupied states of the projected Hamiltonian matrix $\tilde{H}$. This is the key step in the CS2CF methodology. To the extent that the top states of the projected Hamiltonian can be obtained more quickly than all states, the methodology will outperform standard dense and sparse-direct solvers which obtain all states.

To obtain the top states as efficiently as possible, we exploit two key properties of the projected Hamiltonian. First, by construction (i.e., projection onto the occupied subspace) the projected Hamiltonian has quite limited spectral width, with maximum eigenvalue limited to that of the highest occupied state. Second, since we seek only the top few (typically $\leq 10\%$) states, we have an extremal eigenvalue problem for a relatively small fraction of the spectrum.

The above properties suggest that iterative solution methods stand to be efficient to obtain the desired states. We explored two such approaches: (1) the LOBPCG method[37,54,55],

as employed in Ref.[36], and (2) CheFSI.

2.2.1 Use of LOBPCG

We first implemented an unpreconditioned version of the LOBPCG method[37,54,55]. The topmost N_t states of $\tilde{H}$ were obtained by computing the bottommost N_t states of $-\tilde{H}$. The initial vectors for the LOBPCG iterations were obtained from the results of a direct diagonalization of $\tilde{H}$ (using LAPACK / ScaLAPACK) from a previous SCF step. Computation of matrix-vector products was carried out directly by the use of dense linear algebra (BLAS) routines.

Overall, this strategy works reasonably well in practice. When compared against results from the direct diagonalization of $\tilde{H}$, a few iterations of LOBPCG are typically enough to obtain the top eigenstates to desired accuracy at a fraction of the cost. However, as the system size increases, so does the total number of states N_s and number of top states N_t. Under these circumstances, the well known computational bottlenecks of the LOBPCG algorithm associated with dense linear algebra operations begin to become apparent. Replacing serial dense linear algebra operations in LOBPCG with the corresponding parallel versions (i.e., PBLAS and ScaLAPACK routines[56–58]) did not significantly improve performance since the computational bottlenecks of LOBPCG also suffer from scalability issues. We therefore turned to a different strategy, as we describe below.

2.2.2 Use of CheFSI — two-level polynomial filtering strategy

To mitigate the aforementioned issues, we replaced the LOBPCG algorithm with Chebyshev polynomial filtered subspace iteration (CheFSI). This turns the overall iterative strategy into one that employs two levels of Chebyshev polynomial filtering on every SCF step. The first (or outer) level allows the computation of a set of orthonormal vectors that (approximately) span the occupied subspace of H. The second (or inner) level uses CheFSI to compute the N_t topmost states of $\tilde{H}$, or equivalently, the N_t lowest states of $-\tilde{H}$.

This turns out to be a much more effective strategy for a number of reasons. First, by virtue of the limited spectral width of the projected Hamiltonian, a low-order polynomial filter suffices for the inner CheFSI iterations. In fact, for all calculations reported here, we found a filter order of 4 or lower to be sufficient. Secondly, as explained in previous work[49], depending on the initial guess provided, as well as the spectral width of the matrix, the use of CheFSI to determine eigenstates of sufficient accuracy often requires the application of multiple CheFSI cycles. These factors appear to work in our favor and we found that 5 or fewer CheFSI cycles were sufficient in all cases considered, provided that the starting vectors for the inner CheFSI iteration were obtained using results from the previous SCF step. Finally, a significant fraction of the time involved in the inner CheFSI iteration is spent on evaluation of the Chebyshev polynomial filter $p_{\widetilde{m}}(-\tilde{H})$ as applied to an $N_s \times N_t$ block of vectors. This operation is based on matrix-matrix multiplications (i.e., GEMM operations in BLAS) and it parallelizes quite efficiently when PBLAS routines[56] are used. Hence, the scalability of the inner CheFSI operation turns out to be more favorable as compared to LOBPCG.

It is worthwhile to discuss the computational complexity of the above procedure for determining the top states of $\tilde{H}$. If an inner Chebyshev filter of order $\widetilde{m}$ is employed, the computational cost of applying the Chebyshev filter is $O(\widetilde{m}\, N_s^2\, N_t)$. The subsequent orthonormalization and projection, diagonalization, and rotation steps associated with the inner problem incur costs of $O(N_s\, N_t^2 + N_t^3)$, $O(N_t^3)$, and $O(\,N_s\, N_t^2)$, respectively, leading to an overall cost of $O(\widetilde{m}\, N_s^2\, N_t + N_s\, N_t^2 + N_t^3)$ for each inner CheFSI cycle. This estimate makes clear that it is advantageous to reduce N_t as much as possible in practical calculations using the CS2CF strategy, as long as this reduction does not adversely affect the accuracy or convergence of the calculation. As described later, this can be done in some cases based on numerical arguments (e.g., there is no need to account for fractionally occupied states which do not affect energies and forces appreciably) or physical ones (certain systems might be expected to have only a few fractionally occupied states, based on symmetry arguments,

for example).

If $\widetilde{m}$ is small (as in practical calculations), the above estimate appears to be of lower complexity than the $O(N_s^3)$ cost associated with the direct diagonalization of the projected Hamiltonian matrix. However, noting that N_t is a small fraction of N_s (typically less than 10%), we see that the asymptotic complexity associated with the inner CheFSI procedure is the same as that of direct diagonalization of $\tilde{H}$ (i.e., $O(N_s^3)$) but with a lower prefactor. As we show in Section 3, this lower prefactor does indeed result in significantly lower computational wall times when the inner CheFSI technique is used in lieu of explicit diagonalization of $\tilde{H}$.

2.3   Implementation

Eq. 15 suggests that for the success of the complementary subspace strategy, it is essential to be able to compute the full density matrix P in an efficient manner once the vector blocks Y and $C_{\mathfrak{F}}$ are available. As described above, in the two-level CheFSI scheme, the outer Chebyshev polynomial filtering iterations allow us to compute the vector block Y (using H) while the inner CheFSI iterations allow us to compute the vector block $C_{\mathfrak{F}}$ (using $\tilde{H}$). Using these computed quantities, evaluating Eq. 15 naively would incur a computational cost of $O(N_b^2 N_s + N_b N_s N_t + N_b^2 N_t)$. However, if the basis set used for the discretization is strictly localized, the computation of certain entries of the density matrix P can be avoided during the SCF iterations, thus resulting in significant reductions in computational cost. Specifically, according to Eq. 6, if the basis functions $e_j(x)$ and $e_{j'}(x)$ have nonoverlapping support, then it is redundant to compute the density matrix entry $P_{j,j'}$ since this term does not contribute to the real-space electron density. With this observation and a few additional factors (as detailed below) in mind, we have implemented the two-level CheFSI based complementary subspace strategy within the framework of the discontinuuos Galerkin electronic structure method (specifically, the DGDFT code), as we now describe.

2.3.1 Background on discontinuous Galerkin electronic structure method and DGDFT code

The discontinuous Galerkin (DG) electronic structure method employs an adaptive local basis (ALB) set to solve the equations of KS-DFT in a discontinuous Galerkin framework.[26,27] The methodology has been implemented in the Discontinuous Galerkin Density Functional Theory (DGDFT) code for large-scale parallel electronic structure calculations.[20,28] The DGDFT approach to solving the KS-DFT equations involves partitioning the global simulation domain into a set of subdomains (or elements). The Kohn-Sham equations are then solved locally in and around each element. These local calculations are used to generate the ALBs (in each element) and the Kohn-Sham equations in the global simulation domain are then discretized using them. The ALBs form a discontinuous basis set globally with discontinuities occurring at the element boundaries. Subsequent to the generation of the ALBs, the interior penalty discontinuous Galerkin approach[59] is used for constructing the Hamiltonian matrix. This formulation ensures that the global continuity of the relevant Kohn-Sham eigenstates and related quantities such as the electron density is sufficiently maintained.

As the number of ALBs is increased, the solution obtained by the above procedure converges systematically to the infinite basis set limit. Since the ALBs incorporate local materials physics into the basis, an efficient discretization of the Kohn-Sham equations can be obtained in which chemical accuracy in total energies and forces can be attained with a few tens of basis functions per atom.[26,27] Additionally, the rigorous mathematical foundations of the discontinuous Galerkin method allow the errors in the above approach to be systematically gauged by means of *a posteriori* error estimators.[60–62] Thus, DGDFT combines the key advantage of planewave basis sets in terms of systematic improvability with that of localized basis sets in reducing basis set size. The DG framework for solution of the Kohn-Sham equations (as implemented in the DGDFT code) has been successfully used to study complex materials problems involving many thousands of atoms.[20,28,29]

Despite the many successes of DGDFT in studying a wide variety of large-scale materials

20

problems, a persistent issue has been to obtain the electron density from the discretized Kohn-Sham Hamiltonian in an efficient and scalable manner for large systems (i.e., systems containing a thousand or more atoms). To address this issue, we have recently investigated the use of Chebyshev polynomial filtered subspace iteration (CheFSI) within DGDFT.[28] While this technique has the same asymptotic computational complexity as traditional diagonalization based methods (i.e., $O(N_s^3)$, with N_s denoting the number of Kohn-Sham states), it has a substantially lower prefactor compared to the existing alternatives (based on direct diagonalization using ScaLAPACK, for instance) within DGDFT. This stems from several favorable properties of the discretized Hamiltonian matrix in DGDFT. These include: a small dimension (e.g., a few tens times the number of atoms) which leads to lower linear algebra operation costs, a relatively low spectral width which ensures Chebyshev polynomials of relatively low order can be employed, and finally, an underlying block sparse structure which ensures that matrix vector products can be carried out with high computational efficiency. These features, along with the favorable parallel scalability of the DG-CheFSI approach have allowed us to tackle systems containing several thousands of atoms in minutes of wall time per SCF step on large-scale computational platforms.[28]

Our experience has shown that the limiting computational bottleneck in such large-scale calculations using the DG-CheFSI approach turns out to be associated with the subspace diagonalization and subspace rotation steps of the Rayleigh-Ritz process.[28] In light of this observation, we view the current contribution as one which directly confronts the computational bottlenecks associated with the above steps and replaces them with the complementary subspace strategy based on an inner level of Chebyshev filtering. In particular, the factors which lead to the success of the DG-CheFSI approach (i.e., favorable properties of the discretized Hamiltonian matrix, good parallel scalability of various operations, etc.) also ensure that the two-level CheFSI based complementary subspace strategy performs with great efficiency when implemented within the DGDFT framework. We now outline some specific implementation details of the above strategy within the DGDFT code.

2.3.2 Implementation details

We highlight a few important details of the implementation of the two-level CheFSI based complementary subspace strategy within DGDFT. The first concerns the manner in which the code transitions from regular CheFSI based SCF iterations to the use of the complementary subspace strategy. As discussed in a previous section, the use of iterative solvers to evaluate the topmost states of the projected Hamiltonian $\tilde{H}$ requires good initial approximations in order to avoid excessive iterations. Consequently, in static (i.e., fixed atomic positions) Kohn-Sham calculations, we first carry out about 4–5 SCF iterations using the conventional CheFSI technique. The eigenstates of $\tilde{H}$ are computed directly by use of LA-PACK or ScaLAPACK routines during these iterations. The N_t top eigenvectors from the last conventional CheFSI iteration are subsequently used as the initial guess for the iterative solvers in the first complementary subspace based SCF iteration. Furthermore, in case of the two-level CheFSI strategy, the bounds for the inner-level Chebyshev polynomial filter (i.e., the one used to compute the topmost states of $\tilde{H}$) are computed using the eigenvalues evaluated in the previous (conventional CheFSI based) SCF step. Following this transition, the topmost eigenvalues and eigenvectors of $\tilde{H}$ are always stored between SCF iterations for use by the iterative solvers in subsequent SCF steps. During molecular dynamics or geometry optimization runs, we have used the above methodology only during the first ionic step. For subsequent ionic steps, the complementary subspace strategy is used exclusively in every SCF iteration.

The second detail pertains to the parallelization aspects of the two-level CheFSI based complementary subspace strategy. The parallelization strategies involved in the first-level Chebyshev polynomial filter computation (i.e., the one associated with H) are described in earlier work[28]. Parallelization of the various linear algebra operations associated with the second level of CheFSI iterations are carried out with the use of PBLAS and ScaLAPACK routines[56–58]. Accordingly, the various matrices involved in these computations are redistributed over two dimensional block cyclic process grids. The various redistribution and

parallel storage format interconversion routines (employing ScaLAPACK's `pdgemr2d` routine or otherwise) did not consume more than one percent of the total time spent in the complementary subspace strategy, even for the largest systems considered here.

Finally, the third detail pertains to the computation of the density matrix in DGDFT by using Eq. 15. Since the supports of the ALBs are confined to individual DG elements, the Hamiltonian matrix in DGDFT enjoys a block-sparse structure in which nonzero contributions arise due to an element and only its nearest neighbors.[20,28] The (full) density matrix enjoys this structure as well.[20] As noted above, the real-space electron density that must be updated in each SCF iteration only accumulates contributions from density matrix entries that are associated with basis functions with overlapping support. These factors combined imply that *only the diagonal blocks* of the density matrix are required in DGDFT when the complementary subspace strategy is used to update the density in each SCF iteration. This is a significant reduction in the number of operations relative to what a naive inspection of Eq. 15 would suggest and it is one of the primary reasons for the success of the present strategy within DGDFT.

The DGDFT code uses a two-level parallelization strategy implemented via Message Passing Interface (MPI) to handle inter-process communication.[20] At the coarse grained level, the parallelism is based on domain decomposition and work is distributed among processors by DG elements. Further, multiple processors are assigned to each element to achieve the second, finer level of parallelism. The observations made above imply that computation of the diagonal blocks of the density matrix incurs no communication between processors associated with different elements as long as the matrix $\tilde{C}_{\mathfrak{J}}$ is available locally on the processors working on a given element. Our implementation makes use of this observation to achieve a good balance of memory storage requirements and parallel scalability of linear algebra operations while working with the matrix $C_{\mathfrak{J}}$. Once SCF convergence has been achieved, the full density matrix (or, more precisely, all the nonzero blocks) needs to be computed. The two-level parallelization strategy implemented in DGDFT ensures that this

computation can be done efficiently in parallel with a small contribution to the overall wall time.

Figure 1 depicts the various steps of the CS2CF strategy within DGDFT for a static ground-state calculation. Table 1 summarizes the values of the various parameters used within the strategy for such calculations.

Table 1: Values of various parameters used for the CS2CF strategy in ground-state calculations at typical electronic temperatures. Note that, as suggested in previous work[28,49], the first SCF step of a ground-state calculation employs multliple (regular) CheFSI cycles (typically, $3 - 4$) while starting from randomly initialized wavefunctions.

Parameter	Criteria used for selecting parameter value	Value commonly used in this work
Total no. of electronic states (N_s)	Number of electrons N_e in system, type of system (metal/insulator, etc.)	$1.05 \times N_e/2$
No. of top states (N_t)	Type of system (metal/insulator, etc.)	$0.1 \times N_s$
Outer Chebyshev polynomial filter order	Spectral width of Hamiltonian H (depends on type of atoms in system)	$30 - 50$
Inner Chebyshev polynomial filter order	Spectral width of projected Hamiltonian $\tilde{H}$, density of fractionally occupied states	4
No. of inner CheFSI cycles (using $\tilde{H}$)	Spectral width of projected Hamiltonian $\tilde{H}$, density of fractionally occupied states	4
No. of initial SCF steps using regular CheFSI	Type of system (metal/insulator, etc.), nature of SCF convergence	$4 - 5$

3 Results and discussion

To demonstrate the accuracy, efficiency, and parallel scaling of the CS2CF methodology, we apply it to five prototypical systems encompassing metals, semimetals, semiconductors, and insulators, ranging is size from a few hundred atoms to over 27,000. The fundamental unit cells (i.e., atomic configurations replicated to generate the various large systems examined subsequently) of these systems are summarized in Table 2. The first system, referred to as
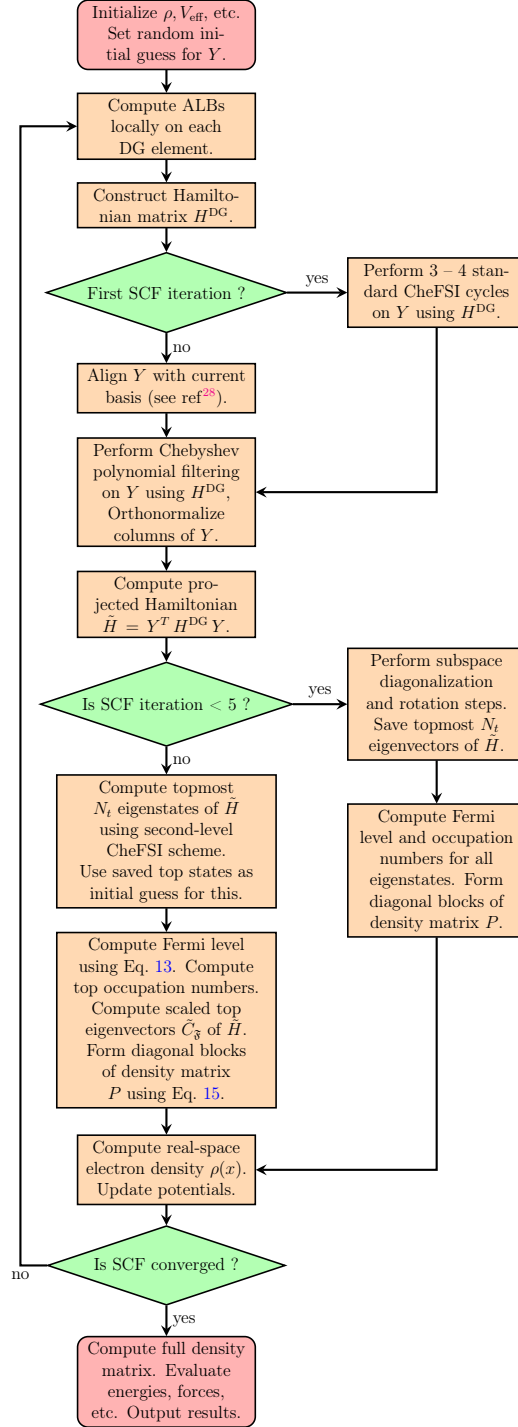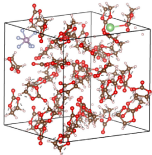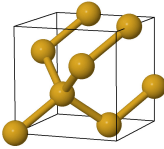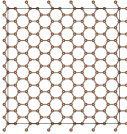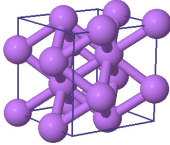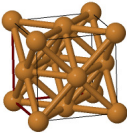
Figure 1: Flowchart depicting the various steps of the CS2CF strategy within DGDFT for a ground-state Kohn-Sham calculation. Note that Y is an $N_b \times N_s$ block of orthonormal vectors that spans the occupied subspace of the Kohn-Sham Hamiltonian H as shown in Eq. 7. H^{DG} denotes the Hamiltonian matrix in the adpative local basis set used in DGDFT.

Electrolyte3D, consists of a three-dimensional bulk lithium-ion electrolyte system originating from the design of energy storage devices. Atoms of hydrogen, lithium, carbon, phosphorus, oxygen, and fluorine, numbering 318 in total, are present in a single unit cell of this system. It serves as a protoypical bulk disordered insulating system. The second, referred to as *SiDiamond3D*, consists of atoms of crystalline silicon in the diamond structure, with 8 atoms in the unit cell. Silicon is a well known semiconductor and, in its crystalline form, has an LDA band gap of ~ 0.6 eV. Thus, it tends to have a small number of fractionally occupied states in Kohn-Sham calculations at room temperature. The third system, referred to as *Graphene2D*, consists of a sheet of graphene for which the unit cell contains 180 carbon atoms. This serves as a prototype for a two-dimensional semimetallic system. The fourth system consists of atoms of lithium in a body centered cubic configuration with 16 atoms in the unit cell. We will refer to this system as *LiBCC3D*. Finally, the fifth system consists of atoms of copper in a face centered cubic configuration with 4 atoms in the unit cell. We will refer to this system as *CuFCC3D*. The LiBCC3D and CuFCC3D systems serve as prototypical examples of simple and more complex bulk metallic systems, respectively. To remove periodicities in larger cells produced by replication, we added mild random perturbations to the atomic positions for all the crystalline/periodic systems mentioned above, before using them in calculations.

Together, these five systems were chosen for their technological relevance as well as the fact that KS-DFT calculations on large supercells based on these can be challenging. Additionally, the electronic properties of these systems cover a broad spectrum – this has helped us ensure that the computational strategy presented in this work is able to deal successfully with different kinds of materials systems, without any computational difficulties arising from the physical nature of the system.

In order to work with larger system sizes, we have employed multiple unit cells of the aforementioned five systems replicated along the coordinate axes. Thus, Electrolyte3D$_{1\times2\times2}$, for example, refers to a system in which the 318-atom unit cell has been replicated along Y and Z directions to produce a 1272-atom bulk system; and similarly, Graphene2D$_{2\times2}$ refers

Table 2: Unit cells of the systems considered in this work. The simulation results presented in Section 3 use supercells constructed by replicating these cells along coordinate axes to produce larger cells.

System	Type	No. of atoms in unit cell	Types of atoms (elements)	No. of electrons in unit cell	ALBs per atom in DGDFT	Representative image of unit cell
Electrolyte3D	Bulk, Insulating	318	C,H,F,Li,O	1104	40	
SiDiamond3D	Bulk, Semiconducting	8	Si	16	40	
Graphene2D	2D, Semimetallic	180	C	360	20	
LiBCC3D	Bulk, Metallic	16	Li	48	35	
CuFCC3D	Bulk, Metallic	4	Cu	44	50	

to a graphene sheet containing 720 atoms.

We have used the local density approximation for the exchange-correlation functional with a rational function parametrization[63] of the correlation energy calculated by Ceperley-Alder[64]. Hartwigsen-Goedecker-Hutter[63,65] and Optimized Norm-Conserving Vanderbilt (ONCV) pseudopotentials[66,67] are employed to remove inert core electrons from the computations. Whenever required, SCF convergence was accelerated by means of Pulay's scheme[68,69] and an electronic temperature of 300 K was used for Fermi-Dirac occupation. Additionally, a Kerker preconditioner[44,70] was employed to minimize charge sloshing while treating metallic systems. The various discretization related parameters in DGDFT (specifically, the number of ALBs per atom, DG penalty parameter, and fineness of real-space grid) were chosen such that chemical accuracy could be attained[26,27] (i.e., error in total energies and forces less than 10^{-3} Ha/atom and 10^{-3} Ha/Bohr, respectively, relative to reference planewave results). This ensures that the calculations presented here are carried out at accuracies typical in practice.

We have typically employed 5% of extra states to accommodate fractional occupations (i.e., $N_x = 5\%$ of $N_e/2$, or equivalently, $N_s = 1.05 \times N_e/2$). Unless specified otherwise,[5] the complementary subspace calculations used the topmost 10% of states, i.e., $N_t = 0.1 \times N_s$. For calculations where LOBPCG was used to compute top states, $10 - 15$ LOBPCG iterations per SCF step were used. For calculations using the CS2CF strategy, the order of the inner Chebyshev filter and number of inner CheFSI cycles were both set to 4. Other parameters relevant to the CS2CF strategy were chosen according to the values shown in Table 1.

All calculations described here were performed on the Edison platform at the National Energy Research Scientific Computing (NERSC) center. Edison has 5462 Cray XC30 nodes. Each node has 64 GB of memory and 24 cores partitioned among two Intel Ivy Bridge processors, running at 2.4GHz. Edison employs a Cray Aries high-speed interconnect with Dragonfly topology for inter-node communication.

---

[5]In practice, for the cases considered here, these choices ensure that the states up to index N_1 have occupation numbers that differ from 1.0 by less than 10^{-10}. In our experience, this is quite conservative, since as long as accuracies of 10^{-7} or better are maintained with respect to occupation numbers, total energies and forces do not change appreciably and SCF convergence is maintained.

3.1 SCF convergence and accuracy

As a first test of the CS2CF methodology we first verified that it reproduces the results of the standard CheFSI methodology (with full diagonalization of the subspace Hamiltonian) with comparable SCF convergence. Accordingly, we first compared the SCF convergence behavior of the complementary subspace strategy against the corresponding behavior of standard CheFSI (as implemented in DGDFT) for a range of systems containing from $500 - 1272$ atoms. Figure 2 shows that for all systems considered, the overall convergence behavior of the complementary subspace strategy is comparable to that of standard CheFSI, as should be the case since the methods are equivalent if eigenvectors are computed exactly. Also, convergence of the complementary subspace strategy is comparable whether LOBPCG or CheFSI is used for computing the top states of the projected Hamiltonian.

Next, we verified that when SCF convergence is reached (i.e., when the relative norm of the electron density residual $\frac{\|\rho_{\text{out}} - \rho_{\text{in}}\|}{\|\rho_{\text{in}}\|} < 10^{-6}$ and the energy per atom has converged to 5×10^{-6} Ha), the energy per atom and the atomic forces obtained by the complementary subspace strategy are in agreement with the results obtained from the standard CheFSI approach to well below discretization error ($\sim 10^{-3}$ Ha/atom). Table 3 shows that for the above systems, the energy per atom difference is on the order of 10^{-5} Ha or less while the maximum difference in force components is on the order of 10^{-4} Ha/Bohr or less, an order of magnitude or more below discretization error.

It is worth pointing out a difference between the scenario depicted in Fig. 2a and the other cases shown in Fig. 2. As discussed in Section 2.1.1, it is not necessary to carry out the subspace diagonalization step or its complementary subspace counterpart while treating insulating systems (like the electrolyte considered in Fig. 2a). Hence, there is no distinction between the LOBPCG and CheFSI based complementary subspace strategies in the case of Fig. 2a, thus leading to the single curve for the complementary subspace strategy.
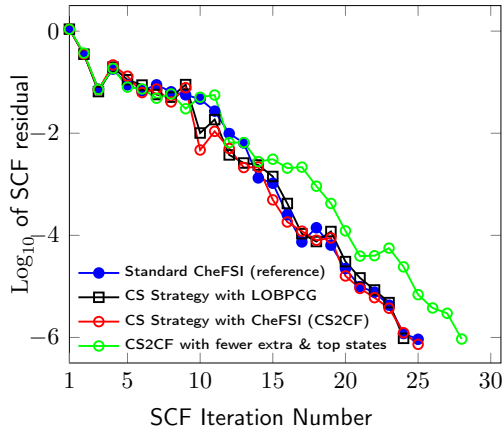
We note also the case of graphene shown in Fig. 2c. Semimetallic systems such as this tend to have relatively few fractionally occupied states near the Fermi level at moderate electronic
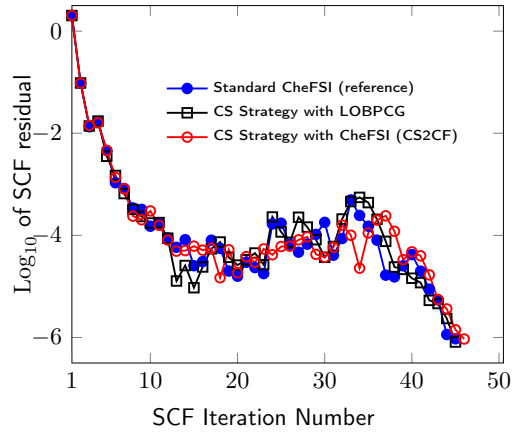
29
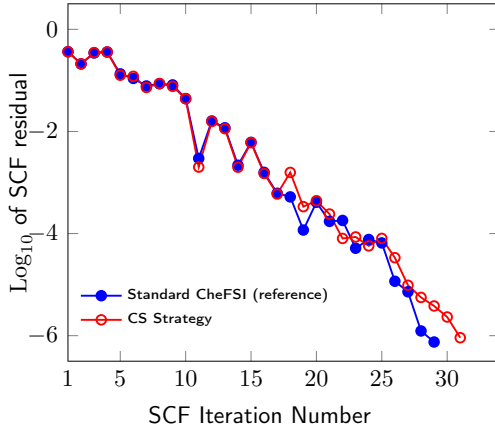
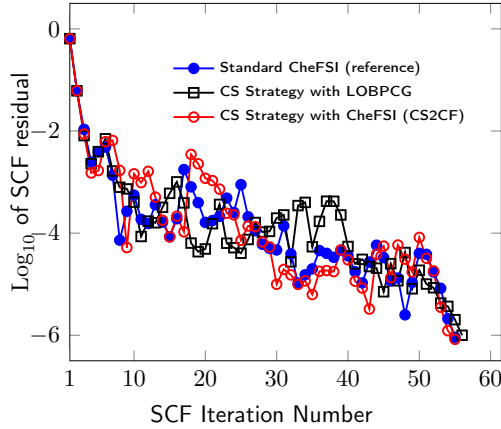(a) Electrolyte3D$_{1\times2\times2}$ system (1272 atoms).
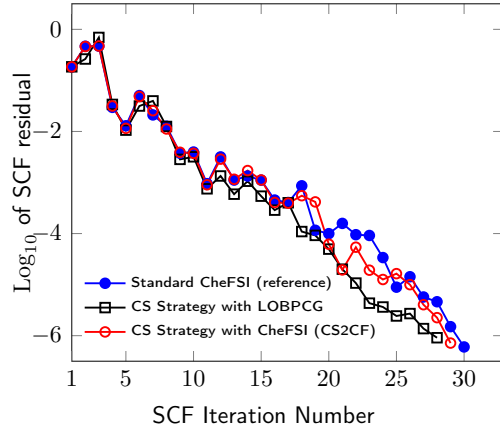
(b) SiDiamond3D$_{5\times5\times5}$ system (1000 atoms).

(c) Graphene2D$_{2\times2}$ system (720 atoms).

(d) CuFCC3D$_{5\times5\times5}$ system (500 atoms).

(e) LiBCC3D$_{4\times4\times4}$ system (1024 atoms).

Figure 2: SCF convergence of the complementary subspace (CS) strategy and standard CheFSI method for systems considered in this work. The top states of the projected Hamiltonian $\tilde{H}$ can be computed using LOBPCG as well as CheFSI (the CS2CF strategy) and results for both are shown.

Table 3: Accuracy of complementary subspace (CS) strategy using LOBPCG and CheFSI (CS2CF) methods to compute top states. Shown are energy per atom and force component differences from standard CheFSI results. Differences are well below discretization error in all cases.

System	CS Strategy with LOBPCG for top states		CS Strategy with CheFSI for top states (CS2CF strategy)	
	Energy per atom difference (Ha)	Max. force component difference (Ha/Bohr)	Energy per atom difference (Ha)	Max. force component difference (Ha/Bohr)
Electrolyte3D$_{1\times2\times2}$	5×10^{-5}	2×10^{-4}	5×10^{-5}	2×10^{-4}
SiDiamond3D$_{5\times5\times5}$	4×10^{-6}	4×10^{-5}	2×10^{-6}	1×10^{-5}
Graphene2D$_{1\times2\times2}$	7×10^{-6}	8×10^{-5}	9×10^{-6}	8×10^{-5}
CuFCC3D$_{5\times5\times5}$	8×10^{-6}	9×10^{-5}	4×10^{-6}	7×10^{-5}
LiBCC3D$_{4\times4\times4}$	7×10^{-6}	9×10^{-5}	5×10^{-6}	8×10^{-5}

temperature.[71] Hence, it is possible to apply (finite electronic temperature) smearing to such systems with fewer extra states in the calculation, thus reducing N_s. Furthermore, the complementary subspace strategy can be made to use fewer top states, thus reducing N_t. This reduces computational effort without significantly impacting accuracy or SCF convergence. As Fig. 2c shows, even with fewer extra and top states in the computation (specifically, the CS2CF strategy used $N_s = 1.025 \times N_e/2$ rather than the usual $N_s = 1.05 \times N_e/2$, and $N_t = 5\%$ of N_s), SCF convergence is not significantly affected. We also verified that the converged energies and forces agreed with reference CheFSI results to well below discretization error. As demonstrated in the next section, however, the computational gains from using fewer states can be quite noticeable, especially for large systems.

3.2 Computational efficiency and parallel scaling

We now carry out a systematic comparison of the computational efficiency and parallel scaling of the new CS2CF and standard CheFSI methodologies.

In order to carry out the comparison, we investigate the wall time for the construction and solution of the subspace problem for a number of large systems. Within the context of the CheFSI or CS2CF strategies, this is the time spent in the sequence of computational steps that lead to the diagonal blocks of the (full) density matrix, after the Chebyshev polynomial filtered vectors given by the columns of Y have been computed using H. Therefore, this time includes contributions from steps that are common to both standard CheFSI and CS2CF strategies, such as orthonormalization of the Chebyshev polynomial filtered vectors (i.e., columns of Y) and formation of the projected Hamiltonian $\tilde{H} = Y^T(HY)$ from the vector blocks Y and $Z = HY$. Additionally, for the standard CheFSI method, it includes the time spent on subspace diagonalization and subspace rotation steps. In contrast, for the CSF2CF strategy, it includes the time spent on computing the top states of the projected Hamiltonian (using the inner level of CheFSI on $\tilde{H}$), and any additional computation required for evaluating the diagonal blocks of the density matrix (see Eq. 15). Thus, within the context of the standard CheFSI or CS2CF strategies, the subspace problem construction and solution wall time provides an estimate of the total wall time spent on every SCF step in (distributed) dense linear algebra operations.

The systems considered for this comparison contain between $4,000$ and $27,648$ atoms and between $23,040$ and $82,944$ electrons. In each case, an identical number of computational cores were allocated to both methods[6] and the ScaLAPACK process grids used were kept as close to square geometries as possible. The results are shown in Figures 3 and 4.

From Figure 3 we see that for the large non-insulating systems considered here, the CS2CF strategy is able to bring down the wall time for the subspace diagonalization and subspace rotation steps in the standard CheFSI method by factors of 3.7–7.8. Additionally, for the particular case of the Electrolyte3D$_{3\times3\times3}$ insulating system, the CS2CF strategy

---

[6]As detailed in previous work[28], the number of MPI processes allocated for construction and solution of the subspace problem in DGDFT is typically set equal to the number of DG elements used in the particular calculation. The results shown in Figures 3 and 4 were obtained following this practice. However, fewer or more processors can be employed as needed. Accordingly, results in the following sections often use a larger number of processors for the subspace problem to bring down wall times for larger-scale calculations.
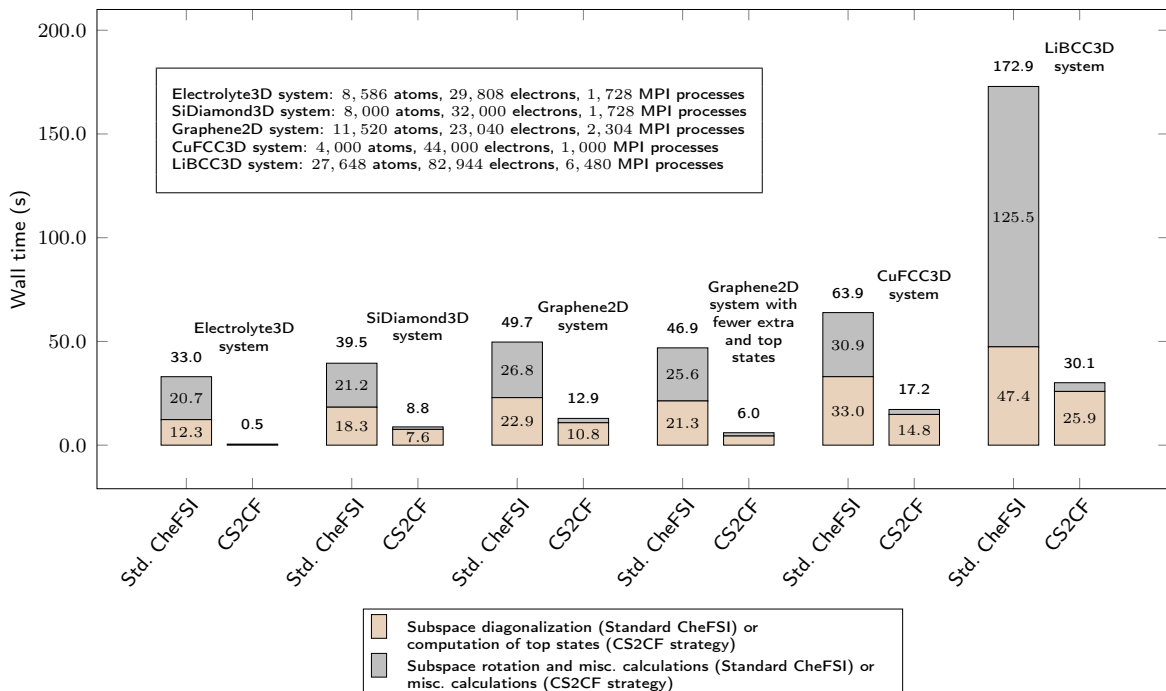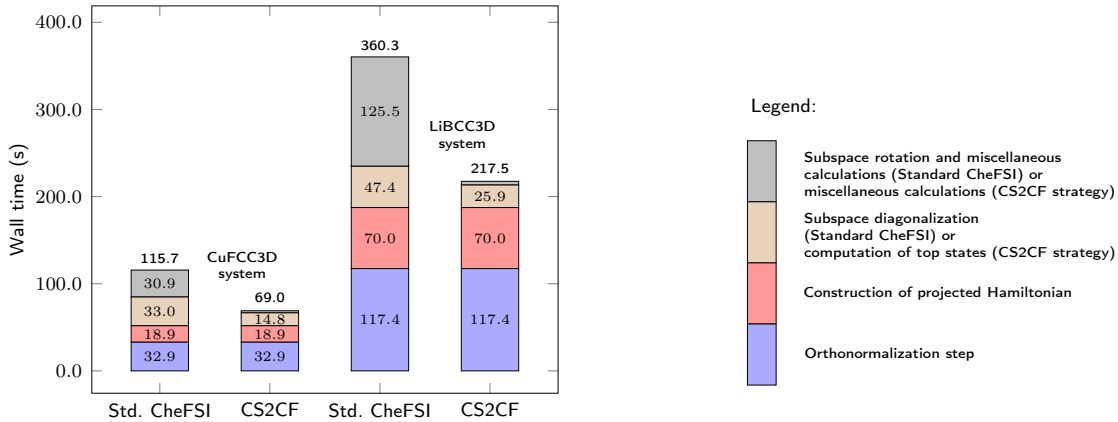
Figure 3: Wall times associated with solution of the subspace problem using standard CheFSI (subspace diagonalization, subspace rotation, miscellaneous calculations) and new CS2CF (computation of top states and miscellaneous calculations) strategies for a few large systems.

eliminates these steps altogether, and brings down the wall time by a factor exceeding 60. This dramatic reduction of the wall times of key steps of the standard CheFSI method leads us to expect that the overall subspace construction and solution time for these systems will be reduced significantly as well. This expectation turns out to be correct. From Figure 4, we see that the overall subspace problem construction and solution wall time is brought down by a factor of $\sim$ 1.7 to 2.2. These computational wall time savings are particularly significant in light of the fact that they occur on every SCF step. From the figure, it is also evident that the overall savings due to the replacement of the subspace diagonalization and subspace rotation steps by the corresponding CS2CF steps are most significant for systems in which these steps are the largest contributors to the subspace problem wall time. For the largest system considered here, i.e., LiBCC3D$_{12 \times 12 \times 12}$ (27, 648 atoms, 82, 944 electrons), the orthonormalization cost contributes to the subspace problem construction wall time in a significant manner and so the overall savings due to the CS2CF strategy, while substantial ($\sim$5.7x reduction in the time spent on the subspace diagonalization and subspace rotation

33
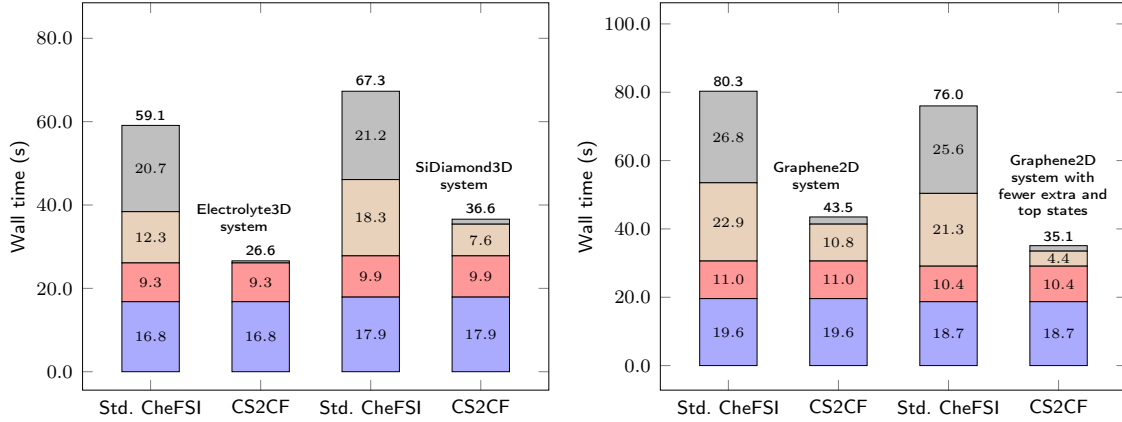
(a) Electrolyte3D$_{3\times3\times3}$ system ($8,586$ atoms, $29,808$ electrons, left two plots) and SiDiamond3D$_{10\times10\times10}$ system ($8,000$ atoms, $32,000$ electrons, right two plots). $1,728$ MPI processes used in all cases.

(b) Graphene2D$_{8\times8}$ system ($11,520$ atoms, $23,040$ electrons). Right two figures use fewer extra and top states. $2,304$ MPI processes used in all cases.

(c) CuFCC3D$_{10\times10\times10}$ system (4000 atoms, $44,000$ electrons, $1,000$ MPI processes, left two plots) and LiBCC3D$_{12\times12\times12}$ system ($27,648$ atoms, $82,944$ electrons, $6,480$ MPI processes, right two plots).

Figure 4: Wall times associated with construction and solution of the subspace problem for large systems using standard CheFSI and new CS2CF methods. Total wall times and contributions of key steps are shown.

steps, as shown in Fig. 3), are somewhat smaller (i.e., ∼1.7x reduction in overall subspace problem wall time) compared to other cases in Figure 4.

In light of earlier comments (see Sections 2.2.2 and 3.1), it is worth pointing out the computational benefits of using fewer extra and top states in the CS2CF strategy. As can be seen in Figures 3 and 4b, on reducing the number of extra states (i.e., N_x), and consequently, the total number of states N_s, the variations in wall times for the steps of the standard CheFSI strategy are not particularly significant. However, lowering N_x also allows us to lower the number of top states (i.e., N_t) in the CS2CF strategy, and this leads to a significant savings. Specifically, as Figure 3 shows, the total wall time spent in the routines associated with the CS2CF strategy decreases by more than a factor of 2 due to the value of N_t being halved.[7]

We also remark that while the above results demonstrate the computational advantages of the CS2CF strategy for large systems, the strategy works equally well (i.e., in terms of lower computational wall times) for smaller systems. For example, for a SiDiamond3D$_{1×1×4}$ system (32 atoms, 64 electrons) we found that the CS2CF strategy was able to reduce the (combined) subspace diagonalization and subspace rotation wall time to less than 0.001 s, from ∼ 0.003 s. The dense linear algebra operations for both strategies were carried out serially in this particular case. Hence, the CS2CF strategy appears to be a computationally advantageous replacement for the standard CheFSI strategy for a wide range of system sizes commonly encountered in Kohn-Sham calculations, as well as for sizes much larger.

Next, we examine the strong parallel scaling properties of the CS2CF strategy and contrast it with the standard CheFSI strategy. The parallel scaling properties of the polynomial filter application step associated with H (that appears in both standard CheFSI and CS2CF) have been detailed in previous work[28] and is identical for both strategies. Thus, we focus

---

[7]Such a striking demonstration of the virtue of lowering N_x and N_t in the CS2CF strategy led us us to investigate the utility of employing alternate smearing schemes[72]. Our experience suggests that while Gaussian and Methfessel-Paxton smearing[73,74] schemes do indeed allow N_x and N_t to be lowered significantly (without otherwise impacting SCF convergence or changing energies and forces beyond acceptable tolerances), the parallel implementation of the CS2CF strategy sometimes suffers scalability and performance issues in these cases (due to a suboptimal number-of-cores to problem-size ratio). We therefore retain Fermi-Dirac smearing in the calculations that follow.

on the subspace problem construction and solution steps here. Taking the LiBCC3D$_{12\times12\times12}$ system as an example, we plot the strong scaling efficiency of the principal steps involved in constructing and solving the subspace problem via the standard CheFSI and CSF2CF strategies in Figure 5. We have used the data points corresponding to 810 computational cores as the reference in this plot.
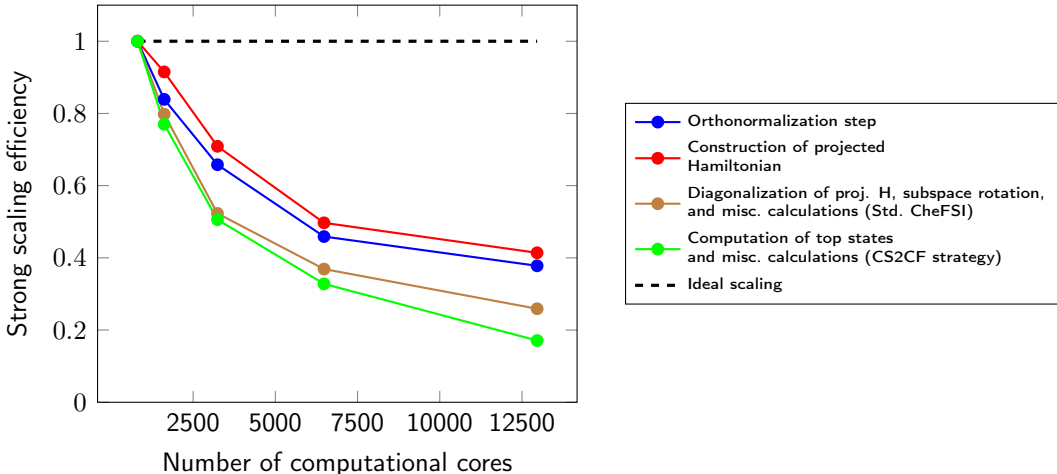

Figure 5: Strong scaling efficiencies of key steps in the standard CheFSI and new CSF2CF strategies for the LiBCC3D$_{12\times12\times12}$ system. While the steps associated with CS2CF scale less well than those associated with standard CheFSI, they yield a significantly lower wall time (by a factor of 5.4 for the case of 12,960 processors shown above). The steps common to both strategies scale somewhat better, reaching approximately 40% strong scaling efficiency at 12,960 processors in this case. The parallel scaling property of the polynomial filter application step associated with H (which appears in both standard CheFSI and CS2CF) is identical for both strategies and is not shown here (see previous work[28] for details).

From the plot, we see that the strong scaling efficiency of the steps strictly associated with the CS2CF strategy is somewhat worse than that of the steps associated with the standard CheFSI strategy, though not markedly so. This is attributed to the fact that the sizes of the matrices in the parallel dense linear algebra operations in the CS2CF approach are significantly smaller than those arising in the standard CheFSI approach. Thus, the PBLAS/ScaLAPACK routines do not parallelize as efficiently for the case of the CS2CF strategy. However, the steps associated with the CS2CF strategy execute significantly faster than their standard CheFSI counterparts — even for the case of 12,960 computational cores, the wall time associated with the CS2CF strategy turns out to be lower than that of the standard CheFSI strategy by a factor of 5.4. Thus, we may conclude that it is preferable

to use the CS2CF strategy regardless of the number of computational cores allocated to the subspace problem.

It is also worth noting that the orthonormalization and projected Hamiltonian construction steps common to both strategies fare somewhat better in terms of strong scaling performance (reaching approximately 40% strong scaling efficiency at 12,960 computational cores in this case). This suggests that it is worthwhile to allocate more computational cores to these parts of the calculation since the wall time for these steps can be reduced significantly. We employ this strategy in the large benchmark calculations presented in the next section.

3.3   Benchmark calculations on large systems

From the results presented above, it is clear that the CS2CF strategy is well suited for bringing down the computational wall times of large-scale Kohn-Sham calculations. We have already demonstrated[28] the superior computational efficiency of the standard CheFSI strategy within DGDFT compared to existing alternatives based on direct diagonalization (using ScaLAPACK) and certain sparse-direct solution strategies (namely, PEXSI[14,15,21]). Since the CS2CF strategy is successful in bringing down the wall times of the standard CheFSI approach, it is the method of choice for large-scale calculations in DGDFT. To demonstrate this, we display in Table 4 the wall time per SCF iteration for the large materials systems considered above, when the CS2CF strategy is employed.[8] For comparison, we also show the corresponding wall times associated with direct diagonalization of the DG Hamiltonian using ELPA[75,76] — a state-of-the-art massively parallel eigensolver library.[9] The ELPA eigensolver was made to use the same total number of computational cores as the CS2CF strategy for all cases considered.

From the table we see that, with the ability to leverage large-scale computational re-

---

[8]In order to reduce the wall time associated with the subspace problem, we employed a larger number of computational cores than used in Figures 3 and 4. Also, rectangular process grid geometries (instead of nearly square ones) were used for carrying out parallel dense linear algebra operations.

[9]The interface to the ELPA library in DGDFT is provided through the ELectronic Structure Infrastructure (ELSI) framework.[77]

Table 4: SCF iteration wall times (in seconds, rounded to the nearest whole number) for large systems using the CS2CF strategy in DGDFT. The contributions of key computational steps are also shown. In the third and sixth columns, numbers in parenthesis indicate the numbers of processors used for subspace problem construction and solution, and wall times for those operations, respectively. For comparison, corresponding wall times associated with direct diagonalization of the DG Hamiltonian using the ELPA library [75,76] are also shown in the last two columns. ELPA was made to use the same total number of computational cores as the CS2CF strategy.

System	No. of atoms (No. of electrons)	Total computational cores (cores used in subspace prob.) [s]	ALB generation [s]	Hamiltonian update [s]	CS2CF strategy (subspace prob. time) [s]	Total SCF wall time via CS2CF strategy [s]	Direct diagonalization via ELPA [s]	Total SCF wall time via ELPA [s]
Electrolyte3D$_{3\times3\times3}$	$8,586$ $(29,808)$	$34,560$ $(3,456)$	12	4	34 (19)	50	647	663
SiDiamond3D$_{10\times10\times10}$	$8,000$ $(32,000)$	$34,560$ $(3,456)$	9	2	40 (24)	51	648	659
Graphene2D$_{8\times8}$	$11,520$ $(23,040)$	$27,648$ $(4,608)$	4	2	35 (27)	41	262	268
CuFCC3D$_{10\times10\times10}$	$4,000$ $(44,000)$	$30,000$ $(3,000)$	20	9	75 (46)	104	199	228
LiBCC3D$_{12\times12\times12}$	$27,648$ $(82,944)$	$38,880$ $(12,960)$	22	13	180 (165)	215	5844	5879

sources, the CS2CF strategy within DGDFT is able to tackle several bulk and nano systems containing tens of thousands of electrons in less than a minute of wall time per SCF step. In terms of the number of electrons in the system, the LiBCC3D$_{12\times12\times12}$ case is the largest, and even in this case, the DGDFT-CS2CF methodology is able to complete each SCF iteration in a little over 3.5 minutes of wall time. In contrast, the wall times required for direct diagonalization of the DG Hamiltonian for these systems (via ELPA) are significantly longer, with the reductions achieved by the CS2CF strategy exceeding a factor of 15 in some cases. The direct diagonalization wall time appears to be the closest to that of the CS2CF strategy for the CuFCC3D$_{10\times10\times10}$ system (likely because the overall size of the DG Hamiltonian is smallest in this case); however, even in this situation, the CS2CF strategy appears to be faster by a factor of ~ 2.6. A comparison of the total SCF wall times between the CS2CF and direct diagonalization strategies also highlights significant gains, with an overall reduction factor of ~ 13 or higher in some cases.

We reiterate that the wall times presented above pertain to discretization parameter choices within DGDFT that lead to well converged (chemically accurate) energies and forces. In our opinion, this is one of the key differences with earlier attempts at simulating such

large scale metallic or semiconducting systems from first principles. To further highlight this point, we employed the DGDFT-CS2CF methodology to carry out a 1.0-ps *ab initio* molecular dynamics simulation of the SiDiamond3D$_{10\times10\times10}$ system initialized at 300 K ionic temperature. We used the NVE ensemble and a time step of 2.5 fs for integrating the equations of motion using the velocity-Verlet scheme[42,78]. We initialized the system by randomly perturbing the positions of the silicon atoms in the SiDiamond3D$_{10\times10\times10}$ configuration and assigning the atoms random velocities consistent with the initial temperature. We then let the system evolve and equilibrate for 50 fs before collecting data for 400 ionic time steps (i.e., 1 ps). To accelerate SCF convergence at each ionic step, we employed linear extrapolation of the real-space electron density and SCF converged wavefunctions from one ionic step were used as the starting point for the SCF iterations on the next ionic step. Results from the simulation are shown in Fig. 6. The mean and standard deviation of the total energy (i.e., kinetic energy of the ions + potential energy) are -3.96329 Ha per atom and 4.8×10^{-6} Ha per atom, respectively. Additionally, the drift in total energy (obtained using a linear fit) is less than 10^{-5} Ha/atom-ps. Thus, the scheme consistently produces high quality atomic forces and, consequently, excellent energy conservation.[10] The mean ionic temperature during course of the simulation[11] comes out to be about 274 K.

With the aid of the CS2CF strategy, the 1.0-ps *ab initio* MD simulation of the above 8,000-atom system can be carried out in $\sim$4.2 minutes per MD step, for a total of $\sim$28 hours of wall time on 34,560 computational cores. From the earlier discussion in Section 3.2, it is clear that doing a similar simulation without the use of the CS2CF strategy would have been far more expensive computationally, if not infeasible due to resource constraints.

---

[10] Although atoms in the system were randomly perturbed before the start of the simulation, the particularly small drift may be due in part to the fact that the system is a crystal in the near harmonic regime. In other ab initio molecular dynamics simulations using the CS2CF strategy we have found larger (but acceptable) drifts for typical discretizations. The Electrolyte3D system, for example, showed drifts of the order of 10^{-4} Ha/atom-ps in some of our calculations. In any case, since, as shown in Table 3, both energy and force errors associated with the CS2CF strategy are well below discretization error, drift for a given MD step size is controlled by discretization rather than eigensolution strategy.

[11]During the course of molecular dynamics simulations using the NVE ensemble, it is usual for the system to reach a mean temperature somewhat different from the initial temperature[78].

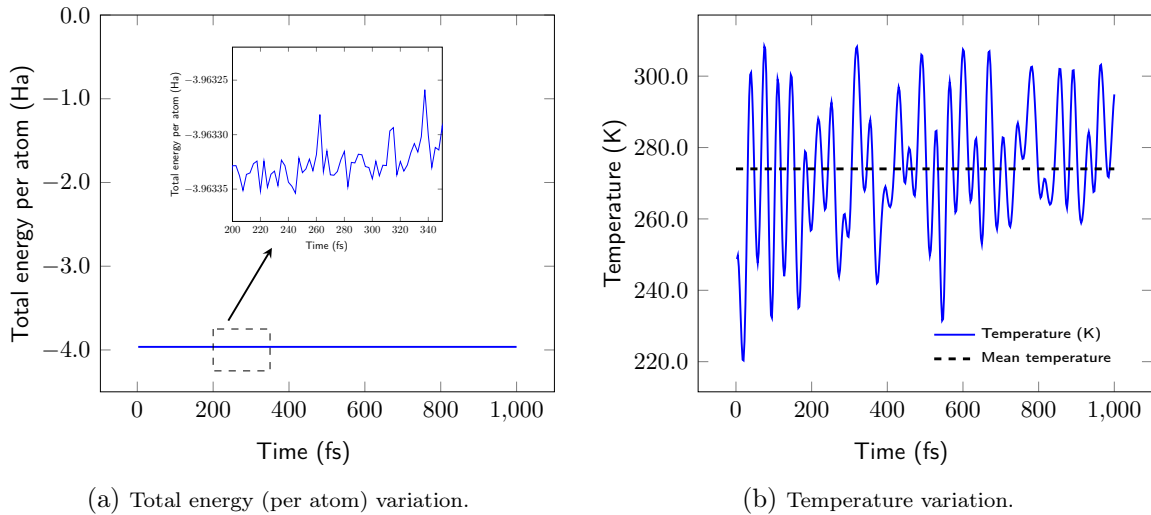(a) Total energy (per atom) variation.   (b) Temperature variation.

Figure 6: Results from 1.0-ps NVE *ab initio* molecular dynamics simulation of the SiDiamond3D$_{10\times10\times10}$ system (8000 atoms, 16,000 electrons) using the CS2CF strategy in DGDFT. Total energy is well conserved, with a drift of less than 10^{-5} Ha/atom over the course of the simulation.

4   Conclusions

In summary, we have presented a novel iterative strategy for KS-DFT calculations aimed at large system sizes, applicable to metals and insulators alike. Our CS2CF methodology combines a complementary subspace (CS) strategy and two-level Chebyshev polynomial filtering (2CF) scheme to reduce subspace diagonalization to just fractionally occupied states and obtain those states in an efficient and scalable way, exploiting the limited spectral width and extremal nature of the resulting eigenvalue problem. In so doing, the CS2CF approach reduces or eliminates some of the most computationally intensive and poorly scaling steps in large-scale Kohn-Sham calculations employing iterative solution methods. We showed that the approach integrates well within the framework of the massively parallel discontinuous Galerkin electronic structure method. Considering a variety of large systems, including metals, semimetals, semiconductors, and insulators, we then demonstrated that the use of the CS2CF strategy within the massively parallel DGDFT code allows us to tackle systems containing tens of thousands of electrons within a few minutes of wall time per SCF iteration on large-scale computing platforms. We found that the CS2CF strategy significantly outperforms alternatives based on direct diagonalization of the Hamiltonian. In particular,

40

the strategy makes possible *ab initio* molecular dynamics simulations of complex systems containing many thousands of atoms within a few minutes per MD step, as we demonstrate for bulk silicon.

With the use of the CS2CF strategy, the subspace diagonalization and subspace rotation steps cease to be the dominant parts of the calculation. For the largest systems considered here, the time for carrying out orthonormalization and forming the projected Hamiltonian then start to contribute significantly. We aim to confront these next. Once again, our focus will not necessarily be on lowering the computational complexity of these steps. Rather, any procedure that can lower the prefactor and/or improve the parallel scalability of these steps is likely be more effective in bringing down wall times in practice, without sacrificing accuracy, thus pushing the envelope of *ab initio* calculations to larger, more complex, and more realistic systems than feasible today.

Acknowledgement

cussions and for their help with improving the presentation of the manuscript. We thank Mitchell Ong for providing the lithium-ion electrolyte system and Donald Hamann for his assistance in generating soft and robust ONCV potentials. Finally, we would like to thank the anonymous reviewers for their comments and suggestions, which helped to improve the manuscript at a number of points.

References

(1) Kohn, W.; Sham, L. J. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **1965**, *140*, 1133–1138.

(2) Martin, R. M. *Electronic Structure: Basic Theory and Practical Methods*, 1st ed.; Cambridge University Press, 2004.

(3) Burke, K. Density Functional Theory: A great physics success story. *Bull. Am. Phys. Soc.* **2017**, *62*.

(4) Burke, K. Perspective on density functional theory. *J. Chem. Phys.* **2012**, *136*, 150901.

(5) Becke, A. D. Perspective: Fifty years of density-functional theory in chemical physics. *J. Chem. Phys.* **2014**, *140*, 18A301.

(6) Fattebert, J.-L.; Osei-Kuffuor, D.; Draeger, E. W.; Ogitsu, T.; Krauss, W. D. Modeling dilute solutions using first-principles molecular dynamics: computing more than a million atoms with over a million cores. In *SC '16*: Proccedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. 2016; pp 12–22.

(7) Bowler, D. R.; Miyazaki, T. Calculations for millions of atoms with density functional theory: linear scaling shows its potential. *J. Phys.: Condens. Matter* **2010**, *22*, 074207.

(8) VandeVondele, J.; Borstnik, U.; Hutter, J. Linear scaling self-consistent field calculations with millions of atoms in the condensed phase. *J. Chem. Theor. Comput.* **2012**, *8*, 3565–3573.

(9) Cohen, A. J.; Mori-Sánchez, P.; Yang, W. Insights into current limitations of density functional theory. *Science* **2008**, *321*, 792–794.

(10) Luo, S. Density Functional Theory: Toward Better Understanding of Complex Systems in Chemistry and Physics. Ph.D. thesis, University of Minnesota, Minneapolis, MN, 2014.

(11) Goedecker, S. Linear scaling electronic structure methods. *Rev. Mod. Phys.* **1999**, *71*, 1085.

(12) Bowler, D.; Miyazaki, T. O(N) methods in electronic structure calculations. *Rep. Prog. Phys.* **2012**, *75*, 036503.

(13) Skylaris, C. K.; Haynes, P. D.; Mostofi, A. A.; Payne, M. C. Introducing ONETEP: Linear-scaling density functional simulations on parallel computers. *J. Chem. Phys.* **2005**, *122*, 084119.

(14) Lin, L.; Lu, J.; Ying, L.; Car, R.; E, W. Fast algorithm for extracting the diagonal of the inverse matrix with application to the electronic structure analysis of metallic systems. *Commun. Math. Sci.* **2009**, *7*, 755.

(15) Lin, L.; Chen, M.; Yang, C.; He, L. Accelerating atomic orbital-based electronic structure calculation via pole expansion and selected inversion. *J. Phys.: Condens. Matter* **2013**, *25*, 295501.

(16) Suryanarayana, P.; Pratapa, P. P.; Sharma, A.; Pask, J. E. SQDFT: Spectral Quadrature method for large-scale parallel O (N) Kohn–Sham calculations at high temperature. *Comput. Phys. Commun.* **2018**, *224*, 288–298.

(17) Pratapa, P. P.; Suryanarayana, P.; Pask, J. E. Spectral Quadrature method for accurate O(N) electronic structure calculations of metals and insulators. *Comput. Phys. Commun.* **2016**, *200*, 96–107.

(18) Aarons, J.; Sarwar, M.; Thompsett, D.; Skylaris, C.-K. Perspective: Methods for large-scale density functional calculations on metallic systems. *J. Chem. Phys.* **2016**, *145*, 220901.

(19) Ratcliff, L. E.; Mohr, S.; Huhs, G.; Deutsch, T.; Masella, M.; Genovese, L. Challenges in large scale quantum mechanical calculations. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2017**, *7*.

(20) Hu, W.; Lin, L.; Yang, C. DGDFT: A massively parallel method for large scale density functional theory calculations. *J. Chem. Phys.* **2015**, *143*, 124110.

(21) Lin, L.; García, A.; Huhs, G.; Yang, C. SIESTA-PEXSI: massively parallel method for efficient and accurate ab initio materials simulation without matrix diagonalization. *J. Phys.: Condens. Matter* **2014**, *26*, 305503.

(22) Hine, N. D.; Haynes, P. D.; Mostofi, A. A.; Skylaris, C.-K.; Payne, M. C. Linear-scaling density-functional theory with tens of thousands of atoms: Expanding the scope and scale of calculations with ONETEP. *Comput. Phys. Commun.* **2009**, *180*, 1041–1053.

(23) Ruiz-Serrano, Á.; Skylaris, C.-K. A variational method for density functional theory calculations on metallic systems with thousands of atoms. *J. Chem. Phys.* **2013**, *139*, 054107.

(24) Prodan, E.; Kohn, W. Nearsightedness of electronic matter. *Proc. Natl. Acad. Sci. U.S.A.* **2005**, *102*, 11635–11638.

(25) Suryanarayana, P. On nearsightedness in metallic systems for O (N) Density Functional Theory calculations: A case study on aluminum. *Chem. Phys. Lett.* **2017**, *679*, 146–151.

(26) Lin, L.; Lu, J.; Ying, L.; Weinan, E. Adaptive local basis set for Kohn–Sham density functional theory in a discontinuous Galerkin framework I: Total energy calculation. *J. Comput. Phys.* **2012**, *231*, 2140–2154.

(27) Zhang, G.; Lin, L.; Hu, W.; Yang, C.; Pask, J. E. Adaptive local basis set for Kohn–Sham density functional theory in a discontinuous Galerkin framework II: Force, vibration, and molecular dynamics calculations. *J. Comput. Phys.* **2017**, *335*, 426–443.

(28) Banerjee, A. S.; Lin, L.; Hu, W.; Yang, C.; Pask, J. E. Chebyshev polynomial filtered subspace iteration in the discontinuous Galerkin method for large-scale electronic structure calculations. *J. Chem. Phys.* **2016**, *145*, 154101.

(29) Hu, W.; Lin, L.; Yang, C. Edge reconstruction in armchair phosphorene nanoribbons revealed by discontinuous Galerkin density functional theory. *Phys. Chem. Chem. Phys.* **2015**, *17*, 31397–31404.

(30) Saad, Y. *Numerical Methods for Large Eigenvalue Problems*, Revised ed.; SIAM, 2011.

(31) Car, R.; Parrinello, M. Unified approach for molecular dynamics and density-functional theory. *Phys. Rev. Lett.* **1985**, *55*, 2471.

(32) Lam, K.-Y.; Lee, H.-P. *Computational methods in large scale simulation*; World Scientific, 2005; Vol. 6.

(33) Mohr, S.; Ratcliff, L. E.; Genovese, L.; Caliste, D.; Boulanger, P.; Goedecker, S.; Deutsch, T. Accurate and efficient linear scaling DFT calculations with universal applicability. *Phys. Chem. Chem. Phys.* **2015**, *17*, 31360–31370.

(34) Motamarri, P.; Gavini, V. Subquadratic-scaling subspace projection method for large-scale Kohn-Sham density functional theory calculations using spectral finite-element discretization. *Phys. Rev. B* **2014**, *90*, 115127.

(35) Garcia-Cervera, C.; Lu, J.; Xuan, Y.; Weinan, E. Linear-scaling subspace-iteration algorithm with optimally localized nonorthogonal wave functions for Kohn-Sham density functional theory. *Phys. Rev. B* **2009**, *79*, 115110.

(36) Michaud-Rioux, V.; Zhang, L.; Guo, H. RESCU: A real space electronic structure method. *J. Comput. Phys.* **2016**, *307*, 593–613.

(37) Knyazev, A. Toward the Optimal Preconditioned Eigensolver: Locally Optimal Block Preconditioned Conjugate Gradient Method. *SIAM J. Sci. Comput.* **2001**, *23*, 517–541.

(38) MATLAB, The MathWorks Inc.: Natick, Massachusetts.

(39) Mazziotti, D. A. Towards idempotent reduced density matrices via particle-hole duality: McWeeny's purification and beyond. *Phys. Rev. E* **2003**, *68*, 066701.

(40) Mazziotti, D. A. Variational minimization of atomic and molecular ground-state energies via the two-particle reduced density matrix. *Phys. Rev. A* **2002**, *65*, 062511.

(41) Truflandier, L. A.; Dianzinga, R. M.; Bowler, D. R. Communication: Generalized canonical purification for density matrix minimization. *J. Chem. Phys.* **2016**, *144*, 091102.

(42) Marx, D.; Hutter, J. *Ab initio molecular dynamics: basic theory and advanced methods*, 1st ed.; Cambridge University Press, 2009.

(43) Kresse, G.; Hafner, J. Ab initio molecular-dynamics simulation of the liquid-metal–amorphous-semiconductor transition in germanium. *Phys. Rev. B* **1994**, *49*, 14251.

(44) Kresse, G.; Hafner, J. Ab initio molecular dynamics for liquid metals. *Phys. Rev. B* **1993**, *47*, 558–561.

(45) Mermin, N. D. Thermal Properties of the Inhomogeneous Electron Gas. *Phys. Rev.* **1965**, *137*, A1441–A1443.

(46) Banerjee, A. S.; Elliott, R. S.; James, R. D. A spectral scheme for Kohn–Sham density functional theory of clusters. *J. Comput. Phys.* **2015**, *287*, 226–253.

(47) Zhou, Y.; Saad, Y.; Tiago, M. L.; Chelikowsky, J. R. Self-Consistent-Field calculations using Chebyshev-filtered subspace iteration. *J. Comput. Phys.* **2006**, *219*, 172–184.

(48) Zhou, Y.; Saad, Y.; Tiago, M. L.; Chelikowsky, J. R. Parallel self-consistent-field calculations via Chebyshev-filtered subspace acceleration. *Phys. Rev. E* **2006**, *74*, 066704.

(49) Zhou, Y.; Chelikowsky, J. R.; Saad, Y. Chebyshev-filtered subspace iteration method free of sparse diagonalization for solving the Kohn–Sham equation. *J. Comput. Phys.* **2014**, *274*, 770–782.

(50) Ghosh, S.; Suryanarayana, P. SPARC: Accurate and efficient finite-difference formulation and parallel implementation of Density Functional Theory: Isolated clusters. *Comput. Phys. Commun.* **2017**, *212*, 189–204.

(51) Ghosh, S.; Suryanarayana, P. SPARC: Accurate and efficient finite-difference formulation and parallel implementation of Density Functional Theory: Extended systems. *Comput. Phys. Commun.* **2017**, *216*, 109–125.

(52) Banerjee, A. S.; Suryanarayana, P. Cyclic Density Functional Theory: A route to the first principles simulation of bending in nanostructures. *J. Mech. Phys. Solids* **2016**, *96*, 605–631.

(53) Parlett, B. N. *The symmetric eigenvalue problem*; SIAM, 1998.

(54) Knyazev, A. V.; Argentati, M. E.; Lashuk, I.; Ovtchinnikov, E. Block locally optimal preconditioned eigenvalue xolvers (BLOPEX) in HYPRE and PETSc. *SIAM J. Sci. Comput.* **2007**, *29*, 2224–2239.

(55) Deursch, J. A.; Gu, M.; Shao, M.; Yang, C. A robust and efficient implementation of LOBPCG. *arXiv:math/1704.07458. arXiv.org e-Print archive* **2017**, (accessed 12-October-2017).

(56) Choi, J.; Dongarra, J.; Ostrouchov, S.; Petitet, A.; Walker, D.; Whaley, R. C. *Applied Parallel Computing Computations in Physics, Chemistry and Engineering Science*; Springer, 1995; pp 107–114.

(57) Choi, J.; Demmel, J.; Dhillon, I.; Dongarra, J.; Ostrouchov, S.; Petitet, A.; Stanley, K.; Walker, D.; Whaley, R. C. *Applied Parallel Computing Computations in Physics, Chemistry and Engineering Science*; Springer, 1995; pp 95–106.

(58) Blackford, L. S.; Choi, J.; Cleary, A.; D'Azevedo, E.; Demmel, J.; Dhillon, I.; Dongarra, J.; Hammarling, S.; Henry, G.; Petitet, A.; Stanley, K.; Walker, D.; Whaley, R. C. *ScaLAPACK Users' Guide*; Society for Industrial and Applied Mathematics: Philadelphia, PA, 1997.

(59) Arnold, D. N. An Interior Penalty Finite Element Method with Discontinuous Elements. *SIAM J. Numer. Anal.* **1982**, *19*, 742–760.

(60) Kaye, J.; Lin, L.; Yang, C. A posteriori error estimator for adaptive local basis functions to solve Kohn-Sham density functional theory. *Commun. Math. Sci.* **2015**, *13*, 1741–1773.

(61) Lin, L.; Stamm, B. A posteriori error estimates for discontinuous Galerkin methods using non-polynomial basis functions Part I: Second order linear PDE. *ESAIM: Math. Modell. Numer. Anal.* **2016**, *50*, 1193–1222.

(62) Lin, L.; Stamm, B. A posteriori error estimates for discontinuous Galerkin methods using non-polynomial basis functions. Part II: Eigenvalue problems. *ESAIM: Math. Modell. Numer. Anal.* **2017**, *51*, 1733–1753.

(63) Goedecker, S.; Teter, M.; Hutter, J. Separable dual-space Gaussian pseudopotentials. *Phys. Rev. B* **1996**, *54*, 1703–1710.

(64) Ceperley, D. M.; Alder, B. J. Ground State of the Electron Gas by a Stochastic Method. *Phys. Rev. Lett.* **1980**, *45*, 566–569.

(65) Hartwigsen, C.; Goedecker, S.; Hutter, J. Relativistic separable dual-space Gaussian pseudopotentials from H to Rn. *Phys. Rev. B* **1998**, *58*, 3641–3662.

(66) Hamann, D. Optimized norm-conserving Vanderbilt pseudopotentials. *Phys. Rev. B* **2013**, *88*, 085117.

(67) Schlipf, M.; Gygi, F. Optimization algorithm for the generation of ONCV pseudopotentials. *Comput. Phys. Commun.* **2015**, *196*, 36–44.

(68) Pulay, P. Convergence acceleration of iterative sequences. The case of SCF iteration. *Chem. Phys. Lett.* **1980**, *73*, 393–398.

(69) Banerjee, A. S.; Suryanarayana, P.; Pask, J. E. Periodic Pulay method for robust and efficient convergence acceleration of self-consistent field iterations. *Chem. Phys. Lett.* **2016**, *647*, 31 – 35.

(70) Lin, L.; Yang, C. Elliptic preconditioner for accelerating the self-consistent field iteration in Kohn–Sham density functional theory. *SIAM J. Sci. Comput.* **2013**, *35*, S277–S298.

(71) Wikipedia, Semimetal — Wikipedia, The Free Encyclopedia. 2017; https://en.wikipedia.org/w/index.php?title=Semimetal&oldid=775884083, [Online; accessed 12-October-2017].

(72) Marzari, N. Ab-initio molecular dynamics for metallic systems. Ph.D. thesis, University of Cambridge, 1996.

(73) Methfessel, M.; Paxton, A. High-precision sampling for Brillouin-zone integration in metals. *Phys. Rev. B* **1989**, *40*, 3616.

(74) Kresse, G.; Furthmuller, J. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Comput. Mater. Sci.* **1996**, *6*, 15 – 50.

(75) Marek, A.; Blum, V.; Johanni, R.; Havu, V.; Lang, B.; Auckenthaler, T.; Heinecke, A.; Bungartz, H.-J.; Lederer, H. The ELPA library: scalable parallel eigenvalue solutions for electronic structure theory and computational science. *J. Phys.: Condens. Matter* **2014**, *26*, 213201.

(76) Auckenthaler, T.; Blum, V.; Bungartz, H.-J.; Huckle, T.; Johanni, R.; Kramer, L.; Lang, B.; Lederer, H.; Willems, P. R. Parallel solution of partial symmetric eigenvalue problems from electronic structure calculations. *Parallel Comput.* **2011**, *37*, 783–794.

(77) Yu, V. W.; Corsetti, F.; Garcia, A.; Huhn, W. P.; Jacquelin, M.; Jia, W.; Lange, B.; Lin, L.; Lu, J.; Mi, W.; Seifitokaldani, A.; Vazquez-Mayagoitia, A.; Yang, C.; Yang, H.; Blum, V. ELSI: A unified software interface for Kohn–Sham electronic structure solvers. *Comput. Phys. Commun.* **2018**, *222*, 267 – 285.

(78) Tadmor, E. B.; Miller, R. E. *Modeling Materials: Continuum, Atomistic and Multiscale Techniques*; Cambridge University Press, 2011.