

A robust solver for wavefunction-based Density Functional Theory calculations

J.-L. Fattebert

Computational Sciences and Engineering Division, Oak Ridge National Laboratory

(Dated: January 5, 2022)

A new iterative solver is proposed to efficiently calculate the ground state electronic structure in Density Functional Theory calculations. This algorithm is particularly useful for simulating physical systems considered difficult to converge by standard solvers, in particular metallic systems. The effectiveness of the proposed algorithm is demonstrated on various applications.

PACS numbers: 71.15.-m 71.15.Dx 71.15.Ap 71.15.Pd

I. INTRODUCTION

In recent years, growing computational resources have changed the way computer simulations are carried out, in particular in the domain of Density Functional Theory (DFT) [1], and its applications to materials science, physics, chemistry, and biology. Instead of focusing on a single physical system and its physical properties, one can now use high-throughput calculations to explore many systems, searching for the best candidate for a specific target property, or with the goal of designing and optimizing materials properties [2–4]. The search for atomic potentials through Machine Learning (ML) techniques also involves solving the DFT equations numerous times for various atomic configurations [5, 6]. These configurations are sometimes even random [7]. While innovative work flows have been developed to spawn numerous DFT calculations in an intelligent manner (see [8] for instance), all of these techniques rely on robust solvers for the DFT equations.

The electronic structure of transition metal compounds, open-shell transition metals or heavy elements can be difficult to converge reliably. Metallic systems constitute a particular numerical challenge for large-scale DFT calculations, due to the strong non-linearity of the Hamiltonian. This effect can become even more pronounced when dealing with large systems, or for "high-Z metals" (*i.e.*, those of high atomic number), as recent advances in computer power allows simulations not only of small unit cells, but larger systems with hundreds of atoms. This strong non-linearity can lead to slower convergence for standard iterative solvers, or even non-convergence and thus limits our simulations capabilities in that field. The phenomenon known as "charge sloshing" is typically worse for large cells [9]. To prevent sloshing instabilities the new electronic density can be mixed with a fraction of the previous density. This leads users to spend some effort determining, if possible, iterative solver parameters that give reliably convergence for the class of compounds they are interested in. While convergence failures are often not reported in the literature if they can be fixed "manually" this issue is gathering more attention as we move away from single application/single set of parameters type simulations. Techniques to overcome convergence failures during first-principles molecular dynamics have been proposed [10], and ML models have been developed to predict failures [11].

Many research papers have been published in recent years proposing innovative algorithms to solve the difficult problem of converging self-consistent iterations in difficult systems. Most published work however addresses semiconductor systems and systems of much smaller sizes than those we can address today [12, 13]. Here the focus is on direct solvers for wavefunctions based discretizations, that is solvers that attempt at solving directly the non-linear Kohn-Sham (KS) equations by computing a set of wavefunctions and their occupations, without solving surrogate linear eigenvalue problems associated with a frozen Hamiltonian. Wavefunctions based discretizations include the popular Plane Waves approach or a Finite Difference discretization of the KS equations. In this category of direct solvers, a very elegant one was proposed by Marzari, Vanderbilt, and Payne (MVP) [14]. It is based on an energy functional that depends on two sets of independent degrees of freedom, the fundamental variables to optimize for: the electronic wavefunctions and their occupations. The new solver proposed in this paper is a more robust generalization of the MVP algorithm. This solver is described in detail in Section II. Some important implementation details are

¹ This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>).

given in Section III and numerical results showing the convergence properties of the iterative solver are shown in Section IV.

II. KOHN-SHAM SOLVER

Kohn-Sham theory [15] is a widely used model for first-principles calculations. It states that the electronic ground state of a physical system can be described by a system of orthogonal one-particle electronic wave functions $\phi_j, j = 1, \dots, N$ that minimizes the KS total energy functional E_{KS} . For a molecular system in a computational domain Ω , the KS energy functional can be written as (in atomic units)

$$\begin{aligned} E_{KS}[\{\phi_i\}_{i=1}^N, X] &= \sum_{i,j=1}^N X_{i,j} \int_{\Omega} \phi_i^*(\mathbf{r}) \left(-\frac{1}{2} \nabla^2 \right) \phi_j(\mathbf{r}) d\mathbf{r} \\ &+ \frac{1}{2} \int_{\Omega} \int_{\Omega} \frac{\rho_e(\mathbf{r}_1) \rho_e(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2 \\ &+ E_{XC}[\rho_e] + \sum_{i,j=1}^N X_{i,j} \int_{\Omega} \phi_i^*(\mathbf{r}) (V_{ext} \phi_j)(\mathbf{r}) d\mathbf{r}. \end{aligned} \quad (1)$$

The first term represents the kinetic energy of the electrons. The second one represents the electrostatic energy of interaction between electrons and is typically computed by solving the corresponding Poisson problem. E_{XC} models the exchange and correlation between electrons. Finally, the last term represents the energy of the electrons in the potential V_{ext} generated by all the atomic cores. The electronic density is defined as

$$\rho_e(\mathbf{r}) = \sum_{i,j=1}^N X_{i,j} \phi_i(\mathbf{r}) \phi_j(\mathbf{r}) \quad (2)$$

where X is a single particle density matrix (DM) which describes the occupation of the electronic wave functions $\phi, i = 1, \dots, N$. To facilitate the later discussion, X is not assumed diagonal here, even though it could be made diagonal for a particular choice of $\phi_i, i = 1, \dots, N$ — the eigenfunctions of the Kohn-Sham Hamiltonian — as it is done most of the time when introducing the KS equations. X is subject to the constraints $Tr(X) = n_e$ where n_e is the total number of electrons in the system, and its eigenvalues, the occupations numbers f_i of the electronic eigenfunctions, are in the interval $[0, 1]$.

At finite electronic temperature T_e , the functional to minimize is the free energy functional $A = E_{KS} - T_e S$ [14, 16] which depends on both the wave functions $\{\phi_i\}_{i=1}^N$ and the occupation numbers $\{f_i\}_{i=1}^N$ — through X —

$$A_N[\{\phi_i\}_{i=1}^N, X] = E_{KS}[\{\phi_i\}_{i=1}^N] - T_e S[X] \quad (3)$$

where the entropic term is given by

$$S[X] = \sum_{i=1}^N [-f_i \ln(f_i) - (1 - f_i) \ln(1 - f_i)]$$

and $0 \leq f_i \leq 1, i = 1, \dots, N$ are the eigenvalues of X . At this functional's minimum, the occupations numbers follow a Fermi-Dirac thermal distribution

$$f_i = f_T(\epsilon_i - \mu) = \frac{1}{e^{(\epsilon_i - \mu)/kT_e} + 1} \quad (4)$$

where μ is the chemical potential whose value is such that the sum of the occupation numbers f_i is equal to the total number of electrons n_e .

Because of the non-linearity and the typical problem size, one commonly uses iterative solvers to search for the minimum of the energy functional (3). This generally requires as a key ingredient the steepest descent direction, which can be combined with a preconditioner K . Denoting the set of discretized trial wave functions $\{\phi_i\}_{i=1}^N$ as an $M \times N$ matrix Φ , these search directions can be written as

$$-K \nabla_{\Phi} E_{KS}.$$

also an $M \times N$ matrix whose matrix elements are the gradient of E_{KS} with respect to each matrix element of Φ . We will use the notation $col_j(\Phi)$ to denote the j^{th} column of a matrix Φ , and $row_i(\Phi)$ to denote the i^{th} row of a matrix Φ . Here we have $col_j(\Phi) = \phi_j$.

For insulators or semiconductor systems at low temperatures, with a non-vanishing band gap, numerous solvers are being used in the scientific community, including various flavors of non-linear conjugate gradient or accelerated gradient methods, such as the Direct Inversion of the Iterative Subspace (DIIS) [17]. Typically, one works with one set of wave functions which are updated iteratively until convergence is achieved. Probably the most popular approach for dealing with metallic systems consists in solving a series of surrogate problems with linearized operators and occupations numbers set to satisfy (4). Their solutions are then combined together using some extrapolation scheme until convergence. Convergence of this process is not guaranteed, is very system dependent, and sometimes problematic for reasons often not well understood. In practice, it can lead to a lot of parameter tuning, trial, and errors.

Marzari et al. [14] proposed to solve directly the minimization problem (3) by an iterative process where one alternates between wave functions updates — for fixed occupation numbers — and occupation numbers optimization — for fixed wave functions. This was found to be the most robust solver by Woods et al. [18], albeit often a slow one. One issue with wave function updates for a system with fractional occupation numbers is that the gradient of the energy functional with respect to an eigenfunction includes the occupation number of that eigenfunction as a multiplying factor. This means vanishing corrections for wave functions associated with small occupation numbers and results in slow convergence. In [14] Marzari et al. proposed to precondition the gradient in a way that results in calculating gradients as if all the wave functions were equally (fully) occupied. We observed however that such an approach leads to numerical instabilities, with relatively large updates of the wave functions near the Fermi level, which leads to significantly reduce step length in the line minimization procedure in order to keep the iteration stable. This results in slow convergence as well.

To overcome this problem, we propose to use gradient-based corrections not directly to update the wave functions in an additive fashion, but instead to use these gradient-based corrections to extend the subspace in which to optimize the single particle density matrix. That is we propose to improve the MVP algorithm by combining the idea of occupation numbers optimization with the restarted Davidson algorithm idea of extending the search subspace for wave functions [19]. We find this modification and extension of the algorithm proposed by Marzari et al. to be more robust, precluding the convergence issues observed with the original formulation.

The Davidson algorithm is widely used in the electronic structure community, but typically to solve an eigenvalue problem with a linear (frozen) Hamiltonian at each step of a self-consistent solver. Here we propose to apply it to solve for the non-linear Hamiltonian problem, updating the electronic density and the nonlinear part of the KS potential when optimizing the DM. Yang et al. [20] proposed a similar approach for systems with a finite band gap at $T_e = 0$, to deal with difficult semiconductor systems, without the need to optimize the occupations numbers. They actually propose to use a $3N$ -dimensional space made not only of the trial vectors and their preconditioned gradients, but also the trial vectors from the previous iteration, as in the Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) algorithm [21]. The algorithm presented here generalizes this idea to metallic systems without a band gap.

Let us consider a trial solution Φ_k , an $M \times N$ matrix whose columns are the discretized wave functions $\phi_k, i = 1, \dots, N$ at step k . Let us also assume we have an initial guess X_k for the single particle density matrix associated with that trial solution. Knowing Φ_k and X_k , one can compute $\rho_{e,k}$, the potential $V(\rho_{e,k})$, and thus the Hamiltonian operator at step k , H_k . Given a preconditioner K , we compute the matrix of preconditioned steepest descent vector projected out of the subspace spanned by Φ_k as $P_k = ortho((I - \Phi_k \Phi_k^T)K(H_k \Phi_k - \Phi_k(\Phi_k^T H_k \Phi_k)))$, where $ortho(A)$ denotes a matrix obtained by orthonormalizing the columns of A . Here it means that the matrices Φ_k and P_k together constitute a set of $2N$ orthonormal vectors. In the subspace spanned by Φ_k and P_k , one can build the projected $2N \times 2N$ Hamiltonian matrix

$$\tilde{H}_{k,2N} = \begin{pmatrix} \Phi_k^T H_k \Phi_k & \Phi_k^T H_k P_k \\ P_k^T H_k \Phi_k & P_k^T H_k P_k \end{pmatrix} \quad (5)$$

Then, as proposed by Marzari et al., but in a space twice as large, we compute a search direction for improving X_k based on the solution of the $2N \times 2N$ symmetric eigenvalue problem

$$\tilde{H}_{k,2N} Z = Z \Lambda \quad (6)$$

where Λ is a diagonal matrix with diagonal entries $\epsilon_i, i = 1, \dots, 2N$ in increasing order, $\epsilon_i \leq \epsilon_{i+1}$. From the solution of that eigenvalue problem, one then computes a single particle density matrix

$$\tilde{X}_{2N} = Z F Z^* \quad (7)$$

where F is a diagonal matrix with entries $f_i = f_T(\epsilon_i - \mu)$. \bar{X}_{2N} is a DM in the space spanned by Φ_k and P_k .

We then define

$$A_{k,2N}(\beta) := A_{2N}[\{\Phi_k, P_k\}, \tilde{X}_{2N} + \beta(\bar{X}_{2N} - \tilde{X}_{2N})]$$

which is the energy functional (3) in a 2N-dimensional space. Note that in that 2N-dimensional space the \tilde{X}_{2N} is initialized from X_k as

$$\tilde{X}_{2N} = \begin{pmatrix} X_k & 0 \\ 0 & 0 \end{pmatrix}.$$

We do a line search to find the optimal density matrix along the line direction, looking for

$$\min_{0 \leq \beta \leq 1} A_{k,2N}(\beta) \quad (8)$$

To do that, we calculate $A_k[\beta = 0]$, $A'_k[\beta = 0]$ and $A_k[\beta = 1]$ and find the minimum of the quadratic polynomial fitting these three values. $A_k[\beta = 0]$ is easy to compute since we already have ρ_e and the potential for the pair $\{\Phi_k, X_k\}$. To compute $A_k[\beta = 1]$, we need to calculate $\rho_e[\{\Phi_k, P_k\}, \bar{X}_{2N}]$, given explicitly by

$$\bar{\rho}_e(\mathbf{r}) = \sum_{i,j=1}^{2N} (\bar{X}_{2N})_{i,j} v_i(\mathbf{r}) v_j(\mathbf{r}) \quad (9)$$

where $v_m := \text{col}_m(\Phi_k)$, $m = 1, \dots, N$, and $v_m := \text{col}_{m-N}(P_k)$, $m = N + 1, \dots, 2N$. From that electronic density, we can evaluate E_{XC} as well as the Coulomb energy. The other terms in the energy are linear and can simply be computed using the matrix elements associated with the pairs of wavefunctions and the atomic potentials. We compute $A'_k[\beta = 0]$ partially by finite differences, evaluating $S'_k[\beta = 0] = (S_k[\beta = \epsilon] - S_k[\beta = 0])/\epsilon$, and then $A'_k[\beta = 0] = \text{Tr}(H_{k,2N}(\bar{X}_{2N} - \tilde{X}_{2N})) - TS'_k[\beta = 0]$. This way we avoid two problems encountered in the MVP approach: (i) there is no need to evaluate the derivative of the entropy function, which is numerically problematic since it involves evaluating logarithms close to 0 and 1, and (ii) this derivative takes fully into account the subspace rotation in the line search direction, which is apparently neglected in MVP. Note that evaluating the entropy S_k for a specific value of β involves computing the eigenvalues for the corresponding DM.

From the result of that line minimization problem, β^* , one obtains a new trial DM in the 2N-dimensional space given by

$$\tilde{X}_{2N} \leftarrow \tilde{X}_{2N} + \beta^*(\bar{X}_{2N} - \tilde{X}_{2N}). \quad (10)$$

Using the updated \tilde{X}_{2N} , a new electronic density is computed

$$\rho_e(\mathbf{r}) = \sum_{i,j=1}^{2N} (\tilde{X}_{2N})_{i,j} v_i(\mathbf{r}) v_j(\mathbf{r}) \quad (11)$$

where $v_m := \text{col}_m(\Phi_k)$, $m = 1, \dots, N$, and $v_m := \text{col}_{m-N}(P_k)$, $m = N + 1, \dots, 2N$.

The above process of optimizing the DM within a 2N-dimensional space can be repeated several times, from defining a new DM target using Eqs. (6) and (7), to doing a new line minimization between the latest DM and this new target, to updating DM. In practice, it is not necessary to converge that process and a fixed number of line minimizations (2 or 3) is sufficient. Once this iterative process is complete, we can extract a subset of N vectors out of the 2N spanning the subspace in which we optimized X . To do that, the last \tilde{X}_{2N} is diagonalized, solving the symmetric eigenvalue problem

$$\tilde{X}_{2N} V = V \Lambda \quad (12)$$

where Λ is a diagonal matrix with real diagonal entries λ_i , $i = 1, \dots, 2N$ in increasing order, $\lambda_i \leq \lambda_{i+1}$, and keep the N vectors corresponding to the highest eigenvalues λ_i , $i = N + 1, \dots, 2N$, that is

$$\Phi_{k+1} = (\Phi_k, P_k) V_2 \quad (13)$$

where V has been split into two $2N \times N$ matrices, $V = (V_1, V_2)$. The N highest eigenvalues should all be between 0 and 1, corresponding to the fully and partially occupied states of $X_{k,2N}$. The discarded eigenvalues should all be 0, corresponding to fully empty states: if they are not, it would mean more empty states should be included in the

calculations to fully cover all the partially occupied states. Note that the diagonalization of the density matrix can lead to several degenerate eigenvalues — around 0 — and thus lead to eigenvectors being very sensitive to numerical noise or the specific algorithm used for diagonalization. While this does not affect the final result, it does affect intermediate results before convergence for this solver.

Once we are back to a set Φ of N vectors describing the electronic structure of the system, we can start again and evaluate a gradient-based search direction, and generate a new $2N$ -dimensional space in which to optimize the DM again. We will refer later to this part as the “outer” iteration, while the DM optimization inside that $2N$ -dimensional space is referred to as the “inner” iteration.

While we do not have convergence proof for the algorithm described above, by construction it should lead to non-increasing energies since every update is based on an energy minimization. If $A'_k[\beta = 0] < 0$, one should have $\beta^* > 0$ and the energy should be decreasing. However, while this property has been proven for systems with band gaps at $T_e=0$ [22], to the best of the author’s knowledge it has not been demonstrated for $T_e > 0$.

III. IMPLEMENTATION

The solver described in the previous section was implemented in the MGmol code, an open source finite difference DFT code [23]. Because the solver relies on an energy minimization (8), it is important for the discretized KS potential to be consistent with the discretized exchange-correlation energy. This comes out naturally for real-space discretization of the Local-density approximations (LDA) exchange-correlation with point-wise functional derivatives. For the Perdew–Burke–Ernzerhof (PBE) [24] functional, the exchange-correlation potential proposed by White and Bird [25] is used. For a discrete set of grid points and a finite difference evaluation of the gradients of the electronic density, it is consistent with the discretized form of the exchange-correlation energy. Besides that, there is nothing specific to a real-space discretization in the solver and convergence results are expected to be the same using a different discretization, such as the popular Plane Waves approach.

Given that the algorithm described above involves calculations in a $2N$ -dimensional space that can become quite expensive compared to simple updates in an N -dimensional space, it can be important to pay attention to some implementation details that may slow down the solver considerably in an unoptimized implementation. For instance, evaluating the electronic density ρ_e according to Eqs. (2) or (11) can be computationally expensive given the double sum over the electronic states, in particular when the sum goes up to $2N$, when evaluating A_k at $\beta = 0$, and $\beta = 1$. One notices however that the number of floating point operations necessary to carry out that sum is MN^2 , thus the same as computing the overlap matrix $S = \Phi^T \Phi$ for example. In practice however, while optimized libraries exist on most computing platforms to carry out matrix-matrix multiplications very efficiently, no such functionality exists for the expression (2) since it is not a standard linear algebra operation.

To take advantage of vendor optimized Basic Linear Algebra Subprograms (BLAS) libraries [26], let us rewrite (2) in a block form. Let ρ_e denote the vector made of the values of ρ at all the mesh points \mathbf{r} . It can be expressed as

$$\rho_e = \text{diag}(\Phi X \Phi^T) \quad (14)$$

where $\text{diag}(A)$ denotes a column vector consisting of the diagonal entries of matrix A . Its components i at \mathbf{r}_i is then given by

$$(\rho_e)_i = (\text{row}_i(\Phi X))^T \cdot \text{row}_i(\Phi). \quad (15)$$

To be computationally efficient, the idea is to first evaluate the matrix product ΦX , which requires MN^2 floating point iterations, followed by the M dot products of row vectors in Eq. (15) which require each N floating point iterations each, for a total of MN operations. The first operation is the expensive one — $O(N^3)$ for $M \sim N$ — and can be carried out with an optimized matrix-matrix product implementation (BLAS3). The second one requires much fewer operations — $O(N^2)$ for $M \sim N$ — and can be computed with M dot products (BLAS1).

Another place where computational efficiency can be enhanced is by using Kleinman-Bylander (KB) separable pseudopotentials [27]. Indeed, in the evaluation of the matrix elements of $\tilde{H}_{k,2N}$ in Eq. (5), the linear part corresponding to the pseudopotentials can then be computed using a set of pre-computed dot products $\langle \beta_{lm}, \phi_i \rangle$ between all the KB projectors from all of the atoms and all of the $2N$ wavefunctions, which can considerably simplify the whole calculation.

The solver described in the previous section was also implemented for systems with spin. In that case, one has a DM for each spin. The line minimization however should be done for the overall energy of the total system, resulting in a single β^* in (10). An application with spin is shown in Section IV to illustrate that case. This strategy should be used for calculations with k-points too, but has not been implemented and will not be shown here.

system	type	computational domain (Bohr)	XC	spin	# w.f.	mesh spacing (Bohr)	# inner iterations	T_e (K)
F	single atom	21. x 21. x 21.	PBE	no	5	0.33	2	300
Au ₁₃	molecule	56.7 x 56.7 x 28.35	LDA	no	174	0.29	3	300
Au ₃₂	surface	15.4 x 15.4 x 30.8	LDA	no	404	0.24	3	600
Fe(salen)(NO)	molecule	25.6 x 25.6 x 38.4	PBE	S=1	69	0.2	2	300
Cu ₃₂	surface	13.6 x 13.6 x 27.2	LDA	no	368	0.28	3	300
Cu ₁₀₈	bulk	20.4 x 20.4 x 20.4	LDA	no	1242	0.28	3	300
La ₅₆	bulk	28.50 x 24.68 x 22.98	LDA	no	358	0.2	3	300

TABLE I: List of physical systems used to evaluate convergence rates. The unreconstructed “surfaces” were generated by simply extending the computational domain for a cubic cell made of 32 atoms by a factor 2 in one direction.

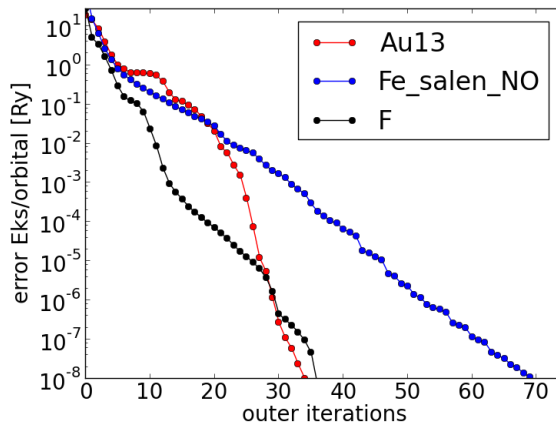


FIG. 1: Convergence as a function of number of outer iterations for three finite systems. The reference energies are the values obtained after a tight convergence of the iterative solver.

IV. NUMERICAL RESULTS

A series of test problems was setup to evaluate the iterative solver proposed above. The focus is on the algorithm performance and no particular attention was made to pick the optimal DFT functional for each system from a physics point of view. The list of test problems and their parameters are described in Tab. I. These includes finite systems, bulk, and surfaces. The geometry for the spin-crossover complex Fe(salen)(NO) was taken from [28]. The geometry for the Au₁₃ cluster is from [29] (Au₁₃(1)),

The Optimized norm-conserving Vanderbilt pseudopotentials (ONCVSP) as proposed by Hamann [30] were used for all the numerical experiments shown here. For the simulations using the PBE exchange and correlation functional, the ONCVSP potentials as parameterized by Schlipf and Gygi [31] were used. The number of valence electrons/atom for these pseudopotentials are 1 for H, 4 for C, 5 for N, 6 for O, 16 for Fe, 19 for Au, 19 for Cu, and 11 for La. Note that for all the simulations presented here except for the spin-crossover complex Fe(salen)(NO), that is the systems without spin, the maximum number of electrons/wavefunction is set to two to take advantage of degeneracy and reduce computational cost.

Fig. 1 shows convergence of the iterative solver for three finite systems: a gold cluster, an open shell transition metal complex, and a single F atom. Fig. 2 shows convergence for two metallic surfaces: gold and copper. Fig. 3 shows convergence for a really large bulk system with 108 copper atoms and 1242 wavefunctions. Finally Fig. 4 shows convergence for a bulk system made of 56 Lanthanum atoms. It can be observed that most systems converge to a tolerance of 10^{-8} Ry/wavefunction within less than 70 outer iterations. The exception is La₅₆ which takes more than 200 outer iterations to converge to that same tolerance. As expected from the algorithm properties, convergence is monotonic for all systems.

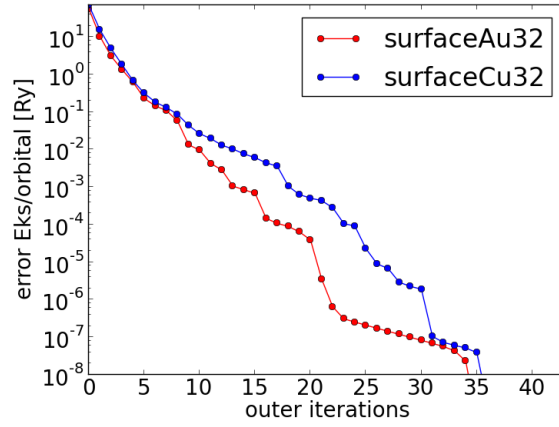


FIG. 2: Convergence as a function of number of outer iterations for two surfaces (Gold and Copper). The reference energies are the values obtained after a tight convergence of the iterative solver.

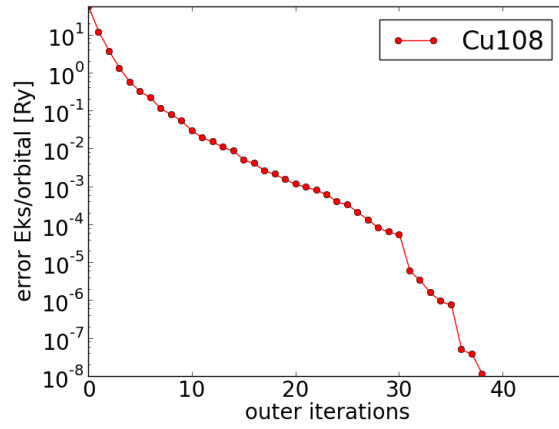


FIG. 3: Convergence as a function of number of outer iterations for bulk Copper (108 atoms cell). The reference energies is the value obtained after a tight convergence of the iterative solver.

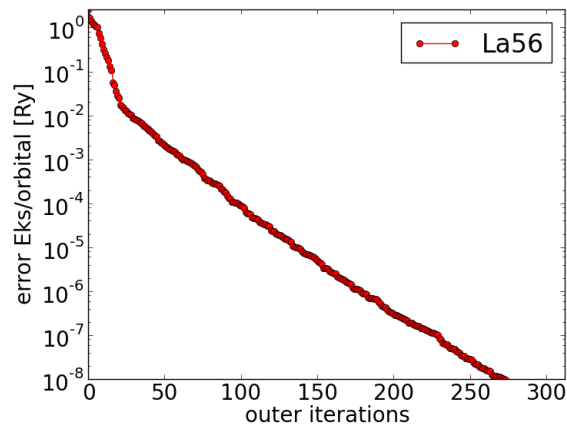


FIG. 4: Convergence as a function of number of outer iterations for bulk Lanthanum (56 atoms cell). The reference energy is the value obtained after a tight convergence of the iterative solver.

V. CONCLUDING REMARKS

A new algorithm to solve for the Density Functional Theory equations was presented for wavefunction-based discretizations. Its key property is robustness and a guaranteed non-increase of the energy of the system from iteration to iteration. Convergence is observed to be fast as well, typically requiring less than 100 wavefunction updates. The computational cost to pay for that is to work in a 2N-dimensional space that include the trial wavefunctions and their correction directions, but this is similar to what the popular restarted Davidson algorithm requires too, albeit in a non-linear context.

A possible extension of the algorithm could be considered by working in a 3N-dimensional space instead of 2N, similar to the LOBPCG algorithm [21], including not only the trial wavefunctions and the correction directions, but in addition the trial wavefunctions from the previous step. This would likely improve convergence given the larger search space for the DM optimization. While time-to-solution is clearly implementation dependent, it would be interesting to see if the extra cost of a 3N-dimensional space would be compensated for by a substantially smaller number of iterations. Another interesting study would be to evaluate that algorithm for atomistic systems modeled using multiple k-points. While the author does not anticipate any issues in that case, that investigation remains to be done.

Also, note that the algorithm presented here does not rely on a specific type of pseudopotentials. While norm-conserving pseudopotentials were used for the results presented in Sec. IV and Sec. III describes some implementation details for Kleinman-Bylander (KB) separable pseudopotentials, the solver could be implemented for other potentials as well, or within the Projector augmented wave method as done for the Block Davidson algorithm [32].

Acknowledgements

Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory (ORNL), managed by UT-Battelle, LLC for the U. S. Department of Energy under Contract No. De-AC05-00OR22725. This research used resources of the Compute and Data Environment for Science (CADES) at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

-
- [1] P. Hohenberg and W. Kohn, *Phys. Rev. B* **136**, 864 (1964).
 - [2] A. Jain, G. Hautier, C. J. Moore, S. Ping Ong, C. C. Fischer, T. Mueller, K. A. Persson, and G. Ceder, *Computational Materials Science* **50**, 2295 (2011).
 - [3] S. Curtarolo, G. Hart, M. Buongiorno Nardelli, N. Mingo, S. Sanvito, and O. Levy, *Nature Materials* **12**, 191 (2013).
 - [4] J. Saal, S. Kirklin, M. Aykol, B. Meredig, and C. Wolverton, *Journal of Metals* **65**, 1501 (2013).
 - [5] J. Behler, *The Journal of Chemical Physics* **145**, 170901 (2016).
 - [6] V. L. Deringer, M. A. Caro, and G. Csányi, *Advanced Materials* **31**, 1902765 (2019).
 - [7] V. L. Deringer, C. J. Pickard, and G. Csányi, *Phys. Rev. Lett.* **120**, 156001 (2018).
 - [8] J. H. Montoya and K. A. Persson, *npj Computational Materials* **3**, 14 (2017).
 - [9] G. P. Kerker, *Phys. Rev. B* **23**, 3082 (1981).
 - [10] L.-P. Wang and C. Song, *Journal of Chemical Theory and Computation* **15**, 4454 (2019).
 - [11] C. Duan, J. P. Janet, F. Liu, A. Nandy, and H. J. Kulik, *Journal of Chemical Theory and Computation* **15**, 2331 (2019).
 - [12] D. D. Johnson, *Phys. Rev. B* **38**, 12807 (1988).
 - [13] D. Raczowski, A. Canning, and L. W. Wang, *Phys. Rev. B* **64**, 121101 (2001).
 - [14] N. Marzari, D. Vanderbilt, and M. C. Payne, *Phys. Rev. Lett.* **79**, 10289 (1997).
 - [15] W. Kohn and L. J. Sham, *Phys. Rev. A* **140**, 1133 (1965).
 - [16] N. Mermin, *Phys. Rev.* **137**, 1441 (1965).
 - [17] G. Kresse and J. Furthmüller, *Phys. Rev. B* **54**, 11169 (1996).
 - [18] N. D. Woods, M. C. Payne, and P. J. Hasnip, *Journal of Physics: Condensed Matter* **31**, 453001 (2019).
 - [19] E. R. Davidson, *Computers in Phys.* **7**, 519 (1993).
 - [20] C. Yang, J. C. Meza, and L.-W. Wang, *J. Comput. Phys.* **217**, 709 (2006).
 - [21] A. V. Knyazev, *SIAM Journal on Scientific Computing* **23**, 517 (2002).
 - [22] E. Cancès, *The Journal of Chemical Physics* **114**, 10616 (2001).
 - [23] *MGmol: Real-space (Finite Difference) First-Principles Molecular Dynamics code*, <https://github.com/LLNL/mgmol> (2017–2021).
 - [24] J. P. Perdew, K. Burke, and M. Ernzerhof, *Phys. Rev. Lett.* **77**, 3865 (1996).
 - [25] J. A. White and D. M. Bird, *Phys. Rev. B* **50**, 4954 (1994).

- [26] J. J. Dongarra, J. Du Croz, S. Hammarling, and I. S. Duff, *ACM Trans. Math. Softw.* **16**, 1–17 (1990).
- [27] L. Kleinman and D. M. Bylander, *Phys. Rev. Lett.* **48**, 1425 (1982).
- [28] J. Conradie and A. Ghosh, *The Journal of Physical Chemistry B* **111**, 12621 (2007).
- [29] A. Kinaci, B. Narayanan, F. G. Sen, M. J. Davis, S. K. Gray, S. K. R. S. Sankaranarayanan, and M. K. Y. Chan, *Sci. Rep.* **6** (2016).
- [30] D. R. Hamann, *Phys. Rev. B* **88**, 085117 (2013).
- [31] M. Schlipf and F. Gygi, *Computer Physics Communications* **196**, 36 (2015).
- [32] A. Tackett, N. Holzwarth, and G. Matthews, *Computer Physics Communications* **135**, 348 (2001).