Commun. Comput. Phys.
doi: 10.4208/cicp.190111.090911a

Vol. **12**, No. 1, pp. 135-161
July 2012

Preconditioners and Electron Density Optimization in Orbital-Free Density Functional Theory

Linda Hung[1], Chen Huang[2] and Emily A. Carter[1,3,*]

[1] *Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544, USA.*
[2] *Department of Physics, Princeton University, Princeton, NJ 08544, USA.*
[3] *Department of Mechanical and Aerospace Engineering and Andlinger Center for Energy and the Environment, Princeton University, Princeton, NJ 08544, USA.*

Received 19 January 2011; Accepted (in revised version) 9 September 2011

Communicated by Weinan E

Available online 20 January 2012

Abstract. Orbital-free density functional theory (OFDFT) is a quantum mechanical method in which the energy of a material depends only on the electron density and ionic positions. We examine some popular algorithms for optimizing the electron density distribution in OFDFT, explaining their suitability, benchmarking their performance, and suggesting some improvements. We start by describing the constrained optimization problem that encompasses electron density optimization. Next, we discuss the line search (including Wolfe conditions) and the nonlinear conjugate gradient and truncated Newton algorithms, as implemented in our open source OFDFT code. We finally focus on preconditioners derived from OFDFT energy functionals. Newly-derived preconditioners are successful for simulation cells of all sizes without regions of low electron-density and for small simulation cells with such regions.

AMS subject classifications: 65Z05, 74G65, 49M15, 81V99

Key words: Density functional theory, truncated Newton method, conjugate gradient method, constrained optimization, benchmarks.

1 Introduction

When modeling materials, a quantum mechanical theory's utility is limited by its computational cost. Part of the cost comes from the evaluation of energy functionals. In this respect, orbital-free density functional theory (OFDFT) [1] is relatively inexpensive

*Corresponding author. *Email addresses:* linda.hung@polytechnique.edu (L. Hung), chenh@lanl.gov (C. Huang), eac@princeton.edu (E. A. Carter)

compared to the more-popular Kohn-Sham density functional theory (KSDFT). OFDFT describes the electronic energy of a system solely using electron density and an ionic external potential; it does not use any wavefunctions and all energy functionals can be formulated to be quasilinear scaling ($\mathcal{O}(N\log(N))$ with a small prefactor [2, 3]. These computational advantages come as a trade-off with accuracy, such that materials with large spatial fluctuations in electron density are not yet well represented by kinetic energy density functionals (KEDFs) in OFDFT. However, main group metals such as aluminum (Al) can be modeled with energies reproducing KSDFT results to $\sim$10 meV [4, 5].

The second part of the computational cost in materials science simulations is due to the optimization method used to minimize the energy. Optimizations must be performed on several levels to fully minimize the total energy of the system. To optimize the cell lattice vectors, cell stresses are minimized. To optimize ion positions, the forces are minimized. And to evaluate the ground state energies, forces, or stresses for a fixed geometry, the electron density must be fully optimized. Princeton Orbital-Free Electronic Structure Software (PROFESS) [2, 6], which uses OFDFT to compute energies, optimizes the electron density, atomic configurations, and cell lattice vectors using iterative methods. In this work, we describe recent improvements to electron density optimization algorithms implemented within PROFESS and report benchmark results.

In Section 2, we introduce the specific OFDFT optimization problem and the treatment of constraints. Unlike wavefunction-based methods, there is no need to orthogonalize orbitals; only two physical conditions must be satisfied. First, the total number of electrons N_e must remain constant. Second, the electron density at any given point must be non-negative. These constraints are satisfied by a judicious choice of optimization variable and the use of a Lagrange multiplier.

Next, we describe our calculational details and benchmarking metrics in Section 3.

In Section 4, we detail the iterative line-search optimization methods implemented within PROFESS. We first describe a line search that conserves the total number of electrons in the simulation [8], and derive the line-search-termination criteria that are analogous to the Wolfe conditions for a standard line search. We then review the nonlinear conjugate gradient (CG) and the truncated Newton (TN) methods, which determine the descent direction to take during the line search. Some benchmarks comparing CG and TN performance for simulation cells containing Al are also included in this section.

In Section 5, we derive and test preconditioners for the inner CG loop of the TN method. In the past, a preconditioner based on the von Weizsäcker (vW) [9] KEDF was found to accelerate convergence in bulk crystals but caused optimization to fail in the presence of vacuum [10]. Here, we propose preconditioners based on Lindhard linear response and preconditioners that include both a Hartree energy term together with a kinetic energy term. The preconditioners are benchmarked with simulation cells of bulk material (no surfaces/vacuum) and with simulation cells containing vacuum.

2 Optimization constraints

In OFDFT, we perform a constrained optimization of the electron density ρ, which is defined in 3-dimensional space, to minimize the electronic energy

$$E[\rho]=T_S[\rho]+J[\rho]+E_{xc}[\rho]+E_{i-e}[\rho]. \tag{2.1}$$

Here, T_S is the KEDF, J is the Hartree energy density functional (electron-electron Coulombic repulsion energy), E_{xc} is the exchange-correlation density functional, and E_{i-e} is the energy due to interaction of the electrons with the ions (here defined as nuclei screened by the core electrons). Several KEDFs currently in use [4, 11–13] can be further broken into the components

$$T_S[\rho]=T_{TF}[\rho]+T_{vW}[\rho]+T_x[\rho], \tag{2.2}$$

where T_{TF} is the Thomas-Fermi (TF) KEDF, which is exact for a homogeneous electron gas [14, 15], T_{vW} is the vW KEDF, exact for a single orbital system [9], and T_x is a nonlocal term, which, when combined with the TF and vW KEDFs, allows the KEDF to reproduce Lindhard linear response [16, 17].

In minimizing E, the first constraint is that the electron density ρ at any given point $\mathbf{x}$ within the simulation cell must always remain non-negative, i.e.,

$$\rho(\mathbf{x})\geqslant 0. \tag{2.3}$$

An effective way to satisfy this constraint is to redefine the optimization variable [2]. In Lignères's thesis [10], the substitutions ϕ (with $\phi^2=\rho$) and χ (with $\exp(\chi)=\rho$) were compared. While both variables guarantee that the non-negativity constraint is satisfied, χ has the advantage of a one-to-one mapping with ρ, while ϕ has a more relevant physical interpretation as being analogous to a wavefunction. In practice, ϕ is found to be much more stable and efficient, and is commonly used for OFDFT optimization [2, 8, 18–20]. Therefore, we focus our discussion on ϕ, which we term the "pseudo-wavefunction" since it satisfies $|\phi|^2=\rho$.

While optimization algorithms using the pseudo-wavefunction always retain a non-negative ρ, ϕ still has the potential to cause problems when it changes sign when using a common form for the vW KEDF,

$$T_{vW}[\phi]=-\frac{1}{2}\int_\Omega \sqrt{\rho(\mathbf{x})}\nabla^2\sqrt{\rho(\mathbf{x})}d\mathbf{x}=-\frac{1}{2}\int_\Omega |\phi(\mathbf{x})|\nabla^2|\phi(\mathbf{x})|d\mathbf{x}, \tag{2.4}$$

where Ω is the simulation cell domain. In this formulation, numerical artifacts may appear in the energy density and potential whenever ϕ changes sign. Fortunately, this obstacle is easily overcome by using the alternative expression

$$T_{vW}[\phi]=-\frac{1}{2}\int_\Omega \phi(\mathbf{x})\nabla^2\phi(\mathbf{x})d\mathbf{x}, \tag{2.5}$$

where the only difference in the resulting energy densities occurs when ϕ is close to zero. Some researchers have suggested that (2.5) might cause problems, since it becomes possible for two discretized pseudo-wavefunctions that produce the same electron density distribution (ϕ_1 and ϕ_2, with $\phi_1^2=\phi_2^2=\rho$) to have different energies [2, 19]. One example where this would occur would be if ϕ_1 on a grid was purely positive and ϕ_2 had changes in sign. Previous researchers have even suggested resetting ϕ at the start of each optimization iteration. However, this reintroduces the numerical artifacts as in (2.4) and makes CG optimization algorithms invalid, since previous conjugate directions no longer apply to the reset wavefunction.

Fortunately, this non-uniqueness problem – where the kinetic energy is not uniquely defined by ρ – is not an issue if ρ and ϕ are differentiable. This differentiability is assumed in our case, since CG and TN are used to determine ϕ. Then even if ϕ_1 and ϕ_2 both satisfy $\phi^2=\rho$ on a grid, they do not correspond to the same differentiable ϕ or ρ if one keeps the same sign and the other changes sign. Therefore, there is only one kinetic energy associated with any given ρ when using (2.5) to evaluate the vW KEDF, and there is no need to constrain ϕ to be non-negative everywhere. Without the non-negativity constraint, there are two equivalent local minima (ϕ^* and $-\phi^*$), but regardless of the minimum approached with CG and TN optimizations, both give identical energies and the same ground state electron configuration.

The second constraint in OFDFT is that the total number of electrons in the simulation cell must remain a constant N_e, i.e.,

$$\int_\Omega \rho(\mathbf{x})\,d\mathbf{x}=\int_\Omega \phi^2(\mathbf{x})\,d\mathbf{x}=N_e. \tag{2.6}$$

This conservation of total electron number during energy minimization can be implemented using the method of Lagrange multipliers, with the Lagrange function defined as [21]:

$$L[\phi,\mu]=E[\phi]-\mu\left(\int_\Omega \phi^2 d\mathbf{x}-N_e\right). \tag{2.7}$$

The minimized energy $E[\phi]$ subject to constraint (2.6) then lies at the saddle point

$$\max_{\mu}\min_{\phi} L. \tag{2.8}$$

The potential $\delta L/\delta\phi$ vanishes at this constrained minimum, $\delta E/\delta\phi$ is parallel to pseudo-wavefunction ϕ, and the maximizing μ is the ratio between the two such that

$$0=\frac{\delta L}{\delta\phi}=\frac{\delta E}{\delta\phi}-2\mu\phi \quad\rightarrow\quad 2\mu\phi=\frac{\delta E}{\delta\phi}. \tag{2.9}$$

In PROFESS, the constrained minimum is reached by approximating the μ that maximizes L with μ_0 and performing an unconstrained minimization on $L[\phi,\mu_0]$. The expression for μ_0 is equivalent to the maximizing μ at the saddle point (2.9), with

$$\mu_0=\frac{1}{2N_e}\int_\Omega\left(\frac{\delta E}{\delta\phi}\phi\right)d\mathbf{x}. \tag{2.10}$$

Of course, other approximations for μ can also reduce to (2.9) at the saddle point. Another candidate tested, $\tilde{\mu}_0 = 0.5(N_e)^{-1/2}\int(\delta E/\delta\phi)d\mathbf{x}$, does not converge as fast as (2.10).

While guaranteed to be exact at the constrained minimum, and close to an optimal μ when ϕ is close to optimized, an approximate μ_0 may be far from the true value of μ that maximizes the Lagrange function when ϕ is far from the optimized electronic structure. Because an inaccurate μ_0 can prevent convergence, this possible problem is addressed in two parts. First, when the potential is large, optimizations to determine $\min_\phi L[\phi,\mu_0]$ may proceed for multiple steps with fixed μ_0. By choosing how often to update μ_0, optimization can be tuned to be faster or more stable. Second, because inaccurate μ_0 may cause the electron count N_e to deviate far from the initial value – which, in turn, may make μ_0 even less accurate – the pseudo-wavefunction ϕ is rescaled at every iteration so that it always satisfies constraint (2.6). Because the rescaling occurs during the line search and must satisfy the modified Wolfe conditions described in the next section, it does not adversely affect convergence. This use of rescaling is only a preventative measure for instabilities and alone cannot replace Lagrange multipliers in optimization algorithms.

3 Benchmark setups

In the following sections, optimization algorithms are benchmarked to assess their performance using a version of the PROFESS that includes preconditioners not implemented in the publicly-released software [2,6]. All simulation cells have uniform 3-dimensional grids with periodic boundary conditions imposed. For calculation of energies and potentials that take place at each optimization iteration, an overall $\mathcal{O}(N\log(N))$ scaling with system size occurs via fast Fourier transforms (FFTs), which are used to make all other calculations local in real or reciprocal space and scaling linearly [2]. Although some optimization wall times are included in the results, benchmarks are mostly determined by counting the number of FFTs, which is a good metric since they are relatively expensive and are used in the calculation of most functionals.

At the beginning of all electron density optimizations in this work, the pseudo-wavefunction is uniform throughout the simulation cell. Electron density is optimized until the norm of the potential, defined as

$$\left\|\frac{\delta L}{\delta\phi}\right\| = \sqrt{\left(\int_\Omega\left(\frac{\delta L}{\delta\phi}\right)^2 d\mathbf{x}\right)\left(\int_\Omega d\mathbf{x}\right)^{-1}}, \tag{3.1}$$

is sufficiently low, or when the energy of the simulation cell has sufficiently converged for three successive iterations.

Energy differences reported in graphs are defined differently from the energy convergence stopping criterion. Any energy differences listed in the following sections are relative to the lowest energy obtained for the system among all the optimization methods tested. This allows us to better compare algorithms, since a method that produces a

significantly higher energy than other methods has not reached the ground state electron configuration, despite any apparent convergences in energy or potential.

Unless otherwise noted, benchmarks use the Wang-Govind-Carter 1999 (WGC99) KEDF with or without vacuum damping (depending whether large regions of low electron density are present in the simulation cell) [4, 22, 23], the local density approximation for the exchange-correlation energy [24, 25], the Huang-Carter pseudopotential for Al [5], and a plane-wave kinetic energy cutoff of 600 eV (grid density of ~ 4 Å^{-1}). WGC99 parameters are $\rho^* = 0.1919$, $\gamma = 2.7$, and $\{\alpha,\beta\} = \{(5\pm\sqrt{5})/6\}$. In some benchmarks, the Huang-Carter (HC) KEDF is also used, with parameters $\lambda = 0$ and $\beta = 0.51$ [13].

4 Optimization methods

4.1 Line search for modified Wolfe conditions

Returning to the discussion of optimization algorithms, we wish to minimize the Lagrange function described in (2.7), while approximating the Lagrange multiplier μ as the function of $\delta E/\delta\phi$ and ϕ given as μ_0 in (2.10). In PROFESS, we use line search methods to minimize the Lagrange function.

Before each line search iteration, a search direction p_k is first calculated; a typical line search then determines an appropriate step size a_k in that direction. If an exact line search is performed, the chosen step size a_k gives the minimum energy along the line search direction. However, finding the precise minimum in a line search can be expensive. An inexact line search is faster; the energy and $\partial E/\partial a_k$ must only satisfy conditions of sufficient decrease and curvature (Wolfe conditions) before terminating the search [7]. The Wolfe conditions are as follows:

$$L[\phi_{k+1}] \leqslant L[\phi_k] + c_1 a_k \nabla L[\phi_k]\cdot p_k, \tag{4.1}$$

$$\nabla L[\phi_{k+1}]\cdot p_k \geqslant c_2 \nabla L[\phi_k]\cdot p_k, \tag{4.2}$$

where $c_1 \in (0,1)$ and $c_2 \in (c_1,1)$ are constants that depend on the algorithm being used. Equation (4.1) is a condition for sufficient decrease while (4.2) is a curvature condition. After determining the step size, a standard line search optimization would update the pseudo-wavefunction at each iteration with

$$\phi_{k+1} = \phi_k + a_k p_k. \tag{4.3}$$

Unfortunately, the standard line search defined in (4.3) presents some difficulties when applied to OFDFT. In particular, the line search does not keep the total electron count in the simulation constant (constraint (2.6)). As mentioned earlier, the Lagrange multiplier is approximated with μ_0 while minimizing $L[\phi,\mu_0]$, so fluctuations of N_e that are too large can potentially make μ_0 very inaccurate and the entire optimization unstable.

Instead of (4.3), a line search should keep the total number of electrons constant without arbitrarily rescaling. For this, we use a mixing method proposed in [8]. This method

does not step in a straight line, but instead moves in a curved direction that conserves total electron count. It is expressed as

$$\phi_{k+1} = \phi_k^{\perp} \sin(\theta_k) + \phi_k \cos(\theta_k), \tag{4.4}$$

where $\phi_k^{\perp}$ is the part of the search direction p_k perpendicular to ϕ_k, normalized to $\sqrt{N_e}$, i.e,

$$\phi_k^{\perp} = \frac{\sqrt{N_e}}{|\phi_k^{\perp,unnormalized}|} \phi_k^{\perp,unnormalized}, \tag{4.5}$$

$$\phi_k^{\perp,unnormalized} = p_k - \left(\frac{\phi_k \cdot p_k}{|\phi_k|^2} \right) \phi_k. \tag{4.6}$$

In addition to conserving $\int_\Omega |\phi|^2 d\mathbf{x} = N_e$, (4.4) has the advantage of being a bracketed search; we require that $0 < \theta_k < \pi/2$ in order to give a mix of the old and new pseudo-wavefunctions. This provides a stability advantage over other electron-conserving line searches [2, 19] that are unbracketed.

In reference [8], the step size along the descent direction was approximated analytically for the mixing method, which sped up computation by skipping the iterations inherent in a line search, but also was found to be unstable in some cases. We use a line search in our implementation. For most optimization problems in OFDFT, the curved line search (4.4) is nearly linear – approximately (4.3) – since the step size θ_k is typically very small; Wolfe conditions (4.1)-(4.2) are sufficient for convergence. However, for larger values of θ, the changes in search direction become more significant (p_k no longer fixed), and the standard Wolfe conditions should be modified to maintain the same convergence properties. The appropriate criteria for line search termination, analogous to the Wolfe conditions, become

$$L[\phi_{k+1}] \leqslant L[\phi_k] + c_1 \theta_k \nabla L[\phi_k] \cdot q_k, \tag{4.7}$$

$$\nabla L[\phi_{k+1}] \cdot q_k \geqslant c_2 \nabla L[\phi_k] \cdot q_k. \tag{4.8}$$

Here, the original line search variable a_k is replaced by the new line search variable θ_k, and p_k is replaced by

$$q_k(\theta_k) = \frac{\phi_{k+1} - \phi_k}{\theta_k} = \phi_k^{\perp} \frac{\sin(\theta_k)}{\theta_k} + \phi_k \frac{\cos(\theta_k) - 1}{\theta_k} \tag{4.9}$$

for $\theta_k \in (0, \pi/2]$, and

$$q_k(0) = \phi_k^{\perp}. \tag{4.10}$$

In addition, q_k must be a descent direction. In practice, (4.9) can be evaluated as a Taylor expansion for better precision, due to the small values of θ_k in line searches. When using the modified Wolfe conditions (4.7)-(4.8) and implementing periodic restarts from steepest descent, global convergence of line search algorithms like TN and CG is guaranteed

for certain differentiable functions because the Zoutendijk condition is satisfied; proofs closely follow those of the Wolfe conditions outlined in [7].

In a standard line search, TN is initialized with $a_{k,0}=1$, while CG does not have a recommended value for $a_{k,0}$. Therefore, to most closely reproduce the same initial step size and direction in the electron-conserving line search, we initialize $\theta_{k,0}$ so that the new wavefunction is parallel to ϕ_k+p_k and normalized to $\sqrt{N_e}$. Because the component of ϕ_k+p_k that is perpendicular to the current wavefunction ϕ_k is

$$\frac{|\phi_k+p_k|}{|\phi_k^\perp|}\sin(\theta_{k,0})\phi_k^\perp=(\phi_k+p_k)^\perp=\frac{\left|\phi_k^{\perp,unnormalized}\right|}{|\phi_k^\perp|}\phi_k^\perp, \tag{4.11}$$

we initialize the line search variable with

$$\theta_{k,0}\approx\sin(\theta_{k,0})=\frac{\left|\phi_k^{\perp,unnormalized}\right|}{|\phi_k+p_k|}. \tag{4.12}$$

Here, we approximate $\theta_{k,0}\approx\sin(\theta_{k,0})$, since the line search variable θ_k starts small and decreases, making the initialization more accurate as the optimal electron density is approached.

Note that we use the Lagrange function $L[\phi,\mu_0]$ while describing the modified Wolfe conditions. While $L[\phi,\mu_0]$ and $E[\phi]$ are equivalent at any point of the electron-conserving line search (since the Lagrange multiplier term in (2.7) is zero), the difference between the energy and Lagrange function becomes important when implementing the inexact line search termination criteria. The modified Wolfe conditions (4.7)-(4.9) are not satisfied at the same time for $E[\phi]$ and $L[\phi,\mu_0]$, and because we are searching for the saddle point of the Lagrange function, the optimization can converge faster if we monitor the modified Wolfe condition based on the Lagrange function instead of the energy. The benchmark in Fig. 1 compares the termination criteria applied to CG optimization for an equilibrium 4-atom face-centered cubic (fcc) Al bulk simulation cell. The CG optimization of bulk fcc Al using the line search checking for convergence of the Lagrange function is twice as fast as an optimization that checks the energy.

However, for most other systems, the difference in using Lagrange and energy line searches is much less dramatic. In some cases where the approximation μ_0 for the Lagrange multiplier is not sufficiently accurate, convergence speeds can even be reversed, since using energy and $\delta E/\delta\phi$ (i.e., fixing $\mu=0$) to check Wolfe conditions is more stable than allowing μ_0 to vary. The choice of E or L for line searches is therefore problem-dependent. Nevertheless, for the benchmarks in the remainder of this work, line searches are performed for the Lagrange function L, with the Lagrange multiplier μ_0 fixed at the approximated value calculated at the beginning of each line search.

4.2 Nonlinear conjugate gradient and truncated Newton

In the previous section, we discussed inexact line search criteria that ensure convergence as long as p_k and q_k are descent directions. Here, we explore the performance of the

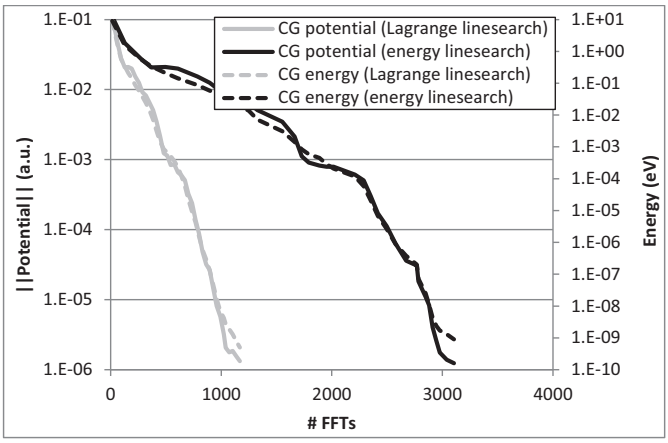

Figure 1: Convergence of electron density optimization for an Al fcc cubic unit cell (4 atoms), comparing the performance of an optimization using a line search with the Lagrange function vs. an optimization using a line search with energy.

optimization depending on the choice of search direction p_k.

Electron density optimization in OFDFT is a very large scale problem even when modeling materials at the nanoscale; a simulation cell containing 10^4 atoms has ϕ with $\sim 10^7$ dimensions. As a result, OFDFT requires optimization algorithms that have minimal computational and storage costs. Two families of line search optimization methods that satisfy these requirements are the nonlinear CG and inexact Newton methods [7]. Neither method requires the Hessian to be computed, and relatively little memory is required since Newton methods do not use previous search directions and CG only uses the immediately prior search direction to determine the next search direction.

The search direction p_k at iteration k can be expressed as

$$p_k = -B_k^{-1}\nabla L_k, \tag{4.13}$$

where $-\nabla L_k$ is the negative potential, which gives the steepest descent direction of the Lagrange function (2.7) at iteration k, and the expression for B_k^{-1} depends on the specific algorithm (CG or Newton).

For linear CG, the search direction is chosen to be conjugate to previous search directions. In nonlinear CG (for problems like OFDFT), a variety of algorithms generate distinct search directions, although all reduce to linear CG when applied to a linear problem. These include the standard Polak-Ribière method and the newer Hager-Zhang method [26], two algorithms that are implemented in PROFESS.

On the other hand, inexact Newton methods use approximate Hessian matrices in determining search directions, with

$$B_k \approx H_{\phi,k} \equiv \frac{\delta^2 L_k}{\delta\phi(\mathbf{x})\delta\phi(\mathbf{x}')}. \tag{4.14}$$

In TN (also known as the line search Newton-CG method) the search direction is found by solving for an inexact solution of the linear equation

$$H_{\phi,k}p_k = -\nabla L_k \tag{4.15}$$

using linear CG. When the convergence tolerance is reached or when the Hessian is found to be non-positive definite, this inner CG loop is terminated. Even though an analytical form of the Hessian $H_{\phi,k}$ is known in OFDFT, an iterative method is used to solve (4.15) since the computation and storage of $H_{\phi,k}^{-1}$, not to mention its application to $-\nabla L_k$ to compute p_k, would scale more than quadratically with system size. On the other hand, the calculation of the projected Hessian, $H_{\phi,k}p_k$, at each iteration scales quasilinearly with system size (like the energy and potential calculations), leaving the optimization scaling dependent on the inner CG loop. The projected Hessian can be evaluated either directly or as a finite difference of the potential, with little difference in computational efficiency. In PROFESS, the projected Hessian is evaluated as a first-order finite difference of the potential.

Because an inner CG loop is used to solve (4.15) for each TN iteration, the scaling of the TN minimization itself (number of iterations needed due to system size) is essentially the same as with CG minimization. For both, scaling is highly dependent on the nature of the problem being studied. If suitable preconditioners were implemented, scaling could be made independent of system size. Of course, finding a good preconditioner is almost as difficult as solving the initial optimization problem. Current preconditioners for OFDFT, which work well in bulk, still often fail for simulation cells containing vacuum. These preconditioners are discussed in Section 5.

Despite similarities in scaling with system size, TN converges quadratically once ϕ is sufficiently close to the solution while CG can converge superlinearly. This is reflected in the faster rate of convergence for TN relative to CG in benchmarks of Al simulation cells. In Fig. 2, TN and CG electron density optimizations are benchmarked for a 4-atom fcc unit cell representing bulk Al (3.97 Å $\times$ 3.97 Å $\times$ 3.97 Å) and a 1110-atom simulation cell representing a stretched Al nanowire of 1 nm diameter surrounded by vacuum (19.00 Å $\times$ 20.00 Å $\times$ 234.45 Å). The norm of the potential (3.1) and the energy (relative to the smallest energy computed using any method) are shown to converge faster for TN in both cases, with significantly better performance observed with TN in the nanowire test. To reduce the potential norm below 10^{-5} a.u. for the 4-atom cell on a single core of a 2.67 GHz Intel Nehalem processor, the CG optimization takes 0.17 s (960 FFTs), and TN optimization takes 0.12 s (864 FFTs). To reduce the potential to below 10^{-5} a.u. for the nanowire on one node (8 cores) of the same processor, CG takes 701.2 s (8896 FFTs) and TN takes 310.8 s (4304 FFTs).

In the benchmarks above, the approximate Lagrange multiplier μ_0 is updated at every iteration for CG and at every inner CG loop iteration for TN. However, as mentioned earlier, frequent updates to μ_0 can sometimes cause problems with convergence. In Fig. 3, an implementation of TN that updates μ_0 at every inner CG iteration is compared to a

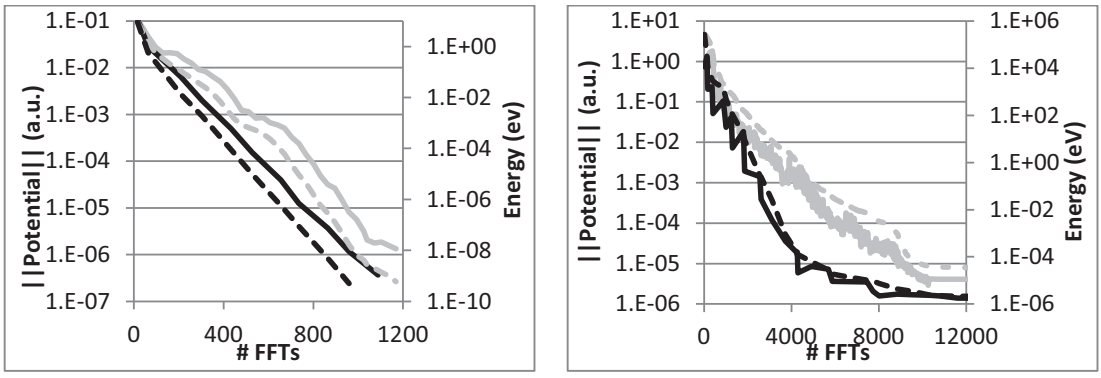

Figure 2: Benchmarked performance of CG (gray) compared to TN (black) electron density optimization on an Al fcc unit cell (left) and an Al nanowire of 1 nm diameter (right). The norms of the potentials are indicated by solid lines and energies (relative to the minimum energy) are dotted lines.

faster one that updates μ_0 only once per TN iteration. This benchmark, using the WGC99 KEDF with parameters $\rho^* = 0.21$ Å^{-3} and $\gamma = 2.5$, is performed on a simulation cell representing a quasi-two-dimensional crack in Al bulk (7828 atoms in a 2.81 Å $\times$ 232.84 Å $\times$ 224.13 Å cell) that does not exhibit any convergence instabilities. The fastest convergence is still observed when μ_0 is updated at every iteration in the inner CG loop for TN. Although almost twice as slow, TN with less frequent updates to μ_0 still converges both faster and to a more optimized configuration than CG. On 32 cores, TN with frequent μ_0 recalculations can optimize electron density below the potential cutoff of 10^{-5} a.u. in 508 s for TN (11696 FFTs). When μ_0 is updated only once per TN iteration, time increases to 827 s (20769 FFTs), which is still an improvement from the 1194 s needed for CG (26512 FFTs).

When far from the saddle point of the Lagrange function, the minimization algorithm becomes unstable if we update μ_0 at each inner CG iteration. Therefore the schedule for updating μ_0 must be determined dynamically. A good balance of speed and stability can be achieved if μ_0 is updated only once each TN iteration when the potential is larger than 0.01 a.u., and if μ_0 is updated at every inner CG loop when the potential norm drops below 0.01 a.u.. However, benchmarks in this work are performed with μ_0 updated at every inner CG loop iteration regardless of the potential norm.

From these benchmarks, we see that the CG and TN optimization routines are not size independent, and more FFTs are required for convergence for larger simulations (in addition to the quasilinear scaling of the cost of each FFT with system size). Overall scaling is therefore somewhat worse than quasilinear when the electron density is being optimized to the ground state configuration. However, the convergence of the nanowire and the cracked bulk is especially difficult due to the presence of vacuum. Large simulation cells containing no vacuum would be expected to have a smaller size scaling effect.

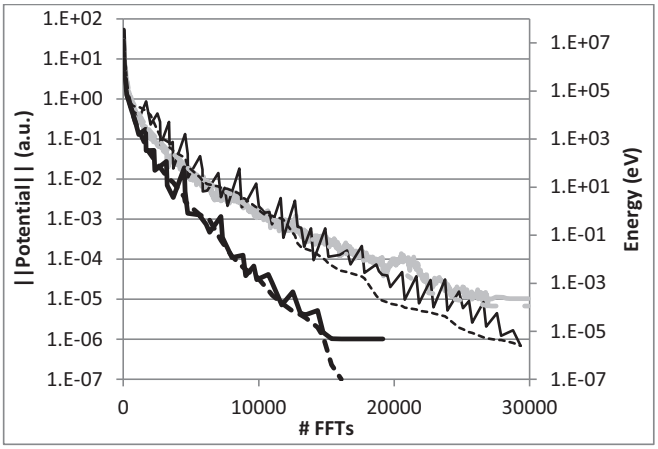

Figure 3: Convergence of potential (solid lines) and energy (dashed lines) for electron density optimization of an edge crack in bulk fcc Al. The thicker black lines indicate TN where the Lagrange multiplier μ is updated at every iteration of the inner CG loop. The thin black lines indicate TN when μ is only updated once per TN iteration. Gray is CG optimization.

5 Preconditioners

5.1 Deriving Newton direction preconditioners

As mentioned in the previous section, the scaling of both CG and TN is primarily determined by the number of iterations required for the convergence of CG loops. The difference between the two is that the CG loop in TN solves a problem in the form of a linear system of equations (specifically, determining p in $H_\phi p = -\nabla L$) while the nonlinear CG optimization solves a problem in the form of an energy minimization. Because preconditioners are easier to determine in the first case and because TN convergence is already more efficient than CG, we focus on preconditioning the CG loop in TN.

The speed at which a linear CG optimization converges depends on the distribution of eigenvalues of matrix H_ϕ. The larger the range of eigenvalues, the longer it takes to converge, and convergence is fastest if there are few eigenvalues or else eigenvalues clustered at only a few values. Preconditioned linear CG performs a change of variables to redistribute eigenvalues using a matrix C to form the new optimization problem [7]:

$$C^{-T} H_\phi C^{-1} p = C^{-T}(-\nabla L). \tag{5.1}$$

The change of basis that takes place in preconditioned CG requires minimal alterations to the standard CG algorithm. The single change at each iteration is to add a step solving for a preconditioned residual y from the residual $r = H_\phi p + \nabla L$ with

$$My = r, \tag{5.2}$$

where $M = C^T C$ is symmetric positive definite. If $M = H_\phi$ is used, the solution is determined in a single iteration after preconditioning (although the inner loop to solve $My = r$

may require multiple iterations), and if $M=I$ (the identity matrix) the algorithm reverts to standard CG. Optimally, M reflects the eigenvalue distribution of H_ϕ, and $y=M^{-1}r$ should be computable with at most quasilinear scaling. To derive a good preconditioner M that reproduces the ill-conditioned parts of H_ϕ, we first examine H_ϕ component by component.

Recall that $H_\phi=\delta^2 L/\delta\phi(\mathbf{x})\delta\phi(\mathbf{x}')$, where the Lagrange function is made up of the electronic energy of the system and an electron conserving term:

$$\begin{aligned} L[\phi] &= E[\phi]-\mu\left(\int_\Omega \phi^2 d\mathbf{x}-N_e\right) \\ &= T_s[\phi]+J[\phi]+E_{xc}[\phi]+E_{i-e}[\phi]-\mu\left(\int_\Omega \phi^2 d\mathbf{x}-N_e\right). \end{aligned} \tag{5.3}$$

The second derivative of the ion-electron energy does not contribute to H_ϕ, since its first derivative is constant with respect to the pseudo-wavefunction (we use local pseudopotentials). We also neglect the exchange-correlation and Lagrange multiplier terms when deriving preconditioners. The exchange-correlation term is smaller than the kinetic energy, and any derivative of the Lagrange multiplier term is also small when total electron density is close to the constrained value. In addition, the nonlocal KEDF term is smaller than TF and vW terms (2.2), so it may be less important when deriving a preconditioner.

In summary, the TF, vW, and Hartree energy density functionals are the primary contributors to H_ϕ. The energies and the first and second functional derivatives relative to ϕ for each of these terms are as follows.

The TF KEDF is local in real space:

$$E^{TF}=c_{TF}\int_\Omega \phi^{10/3}(\mathbf{x})d\mathbf{x}, \tag{5.4}$$

$$V^{TF}=\frac{10}{3}c_{TF}\phi^{7/3}(\mathbf{x}), \tag{5.5}$$

$$H_\phi^{TF}=\frac{70}{9}c_{TF}\phi^{4/3}(\mathbf{x})\delta\left(\mathbf{x}-\mathbf{x}'\right), \tag{5.6}$$

where c_{TF} is $0.3(3\pi)^{2/3}$.

The vW terms depend on the gradient of the pseudo-wavefunction ϕ, which is local in reciprocal space. Since functional derivatives are taken relative to ϕ, simpler forms of the potential and Hessian are obtained than if taking the derivatives relative to ρ:

$$E^{vW}=\int_\Omega \phi(\mathbf{x})\left(-\frac{1}{2}\nabla^2\right)\phi(\mathbf{x})d\mathbf{x}, \tag{5.7}$$

$$V^{vW}=\frac{\delta E^{vW}[\phi]}{\delta\phi(\mathbf{x})}=-\nabla^2\phi(\mathbf{x}), \tag{5.8}$$

$$H^{vW}=-\nabla^2\delta\left(\mathbf{x}-\mathbf{x}'\right). \tag{5.9}$$

Here, $\delta(\mathbf{x})$ is the delta distribution. In the past, the second functional derivative of the vW energy has been moderately successful in preconditioning CG optimization in OFDFT for simulation cells containing only bulk [10,19].

The second derivatives of TF and vW can be discretized into Hessian matrices to form kinetic energy preconditioners

$$M^{TF}=H_{\phi}^{TF}=\frac{70}{9}c_{TF}D_{\phi^{4/3}}, \tag{5.10}$$

$$M^{vW}=H^{vW}=\mathcal{F}^{-1}D_{q^2}\mathcal{F}, \tag{5.11}$$

where q is a vector containing the magnitudes of the reciprocal space coordinates, D_{ψ} indicates a diagonal matrix whose diagonal is the vector ψ, and $\mathcal{F}$ and $\mathcal{F}^{-1}$ indicate the forward and reverse discrete Fourier transform matrices, which act on vectors with

$$\mathcal{F}[f(\mathbf{l})]=\frac{1}{N}\sum_{l_1=0}^{N_1-1}\sum_{l_2=0}^{N_2-1}\sum_{l_3=0}^{N_3-1}f(\mathbf{l})\times e^{-2\pi i(l_1m_1/N_1+l_2m_2/N_2+l_3m_3/N_3)}, \tag{5.12}$$

$$\mathcal{F}^{-1}[f(\mathbf{m})]=\sum_{m_1=0}^{N_1-1}\sum_{m_2=0}^{N_2-1}\sum_{m_3=0}^{N_3-1}f(\mathbf{m})\times e^{2\pi i(l_1m_1/N_1+l_2m_2/N_2+l_3m_3/N_3)}, \tag{5.13}$$

where N_i is the total number of gridpoints in dimension i, $N=N_1\cdot N_2\cdot N_3$, and $\mathbf{l}=(l_1,l_2,l_3)$ and $\mathbf{m}=(m_1,m_2,m_3)$ are the real and reciprocal space indices, respectively.

For the Hartree energy, the second functional derivative is more complicated than either H_{ϕ}^{TF} or H^{vW}, and is a sum of a term local in real space with one local in reciprocal space:

$$J=\frac{1}{2}\int_{\Omega}\int_{\Omega}\frac{\phi^2(\mathbf{x})\phi^2(\mathbf{x})}{|\mathbf{x}-\mathbf{x}'|}d\mathbf{x}d\mathbf{x}', \tag{5.14}$$

$$V^J=\phi(\mathbf{x})\int_{\Omega}\frac{\phi^2(\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|}d\mathbf{x}', \tag{5.15}$$

$$H_{\phi}^{J}=\delta(\mathbf{x}-\mathbf{x}')\int_{\Omega}\frac{\phi^2(\mathbf{x}'')}{|\mathbf{x}-\mathbf{x}''|}d\mathbf{x}''+\frac{2\phi(\mathbf{x})\phi(\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|}=H_{\phi}^{J1}+H_{\phi}^{J2}. \tag{5.16}$$

We show in benchmarks that the Hartree term alone is a bad preconditioner. However, the second term of the Hartree second functional derivative can be combined with kinetic energy-based Hessians. The discretized Hartree Hessian suggests a preconditioner of the form

$$M^{J2}=H_{\phi}^{J2}=8\pi D_{\phi}\mathcal{F}^{-1}D_{q^{-2}}\mathcal{F}D_{\phi}. \tag{5.17}$$

All the individual preconditioners proposed above can be easily inverted. The application of the inverse partial Hessian to some vector remains at most quasilinear scaling (Table 1), since the Fourier transform applied to any vector p scales quasilinearly with the size of p through the use of FFTs, and the computation of the diagonal matrices scales linearly.

We can derive more preconditioners if we assume that electron density is close to a homogenous electron gas (e.g., Al bulk). In this case, the pseudo-wavefunction is nearly constant, with $\phi \approx \sqrt{\rho_0}$, where ρ_0 is the average electron density. A Hartree preconditioner for bulk then can be reduced from (5.17) to

$$M^{J0} = 8\pi D_{\sqrt{\rho_0}} \mathcal{F}^{-1} D_{q^{-2}} \mathcal{F} D_{\sqrt{\rho_0}} = 8\pi\rho_0 \mathcal{F}^{-1} D_{q^{-2}} \mathcal{F}. \tag{5.18}$$

Because this expression is solely diagonal in reciprocal space, it can be combined with the vW Hessian (5.11) to form another preconditioner,

$$M^{vW+J0} = \mathcal{F}^{-1} \left(D_{q^2} + 8\pi\rho_0 D_{q^{-2}} \right) \mathcal{F}. \tag{5.19}$$

Similarly, a TF-like term associated with a uniform electron gas can be added to the vW preconditioner, giving two preconditioners

$$M^{TF0vW} = \mathcal{F}^{-1} \left(D_{q^2} + \frac{70 c_{TF} \rho_0^{2/3}}{9} I \right) \mathcal{F}, \tag{5.20}$$

$$M^{TF0vW+J0} = \mathcal{F}^{-1} \left(D_{q^2} + \frac{70 c_{TF} \rho_0^{2/3}}{9} I + 8\pi\rho_0 D_{q^{-2}} \right) \mathcal{F}. \tag{5.21}$$

The application of the inverse of these preconditioners to any vector is also quasilinear scaling since they are local in reciprocal space.

In addition to Hessian-based preconditioners, another family of preconditioners can be derived from Lindhard linear response. As mentioned above, numerous KEDFs have been derived to reproduce the following linear response:

$$\left. \frac{\delta^2 T_S[\rho]}{\delta\rho(\mathbf{x})\delta\rho(\mathbf{x}')} \right|_{\rho_0} = \mathcal{F}^{-1} \left(-\frac{1}{\chi_{Lind}(q)} \right), \tag{5.22}$$

where

$$\chi_{Lind} = -\frac{k_F}{\pi^2} \left(\frac{1}{2} + \frac{1-\eta^2}{4\eta} \ln \left| \frac{1+\eta}{1-\eta} \right| \right) \tag{5.23}$$

and $k_F = (3\pi^2\rho_0)^{1/3}$ and $\eta = q/2k_F$. Since (5.22) is the second derivative relative to electron density ρ instead of the pseudo-wavefunction ϕ, the chain rule is used to derive a total kinetic energy functional preconditioner of the form

$$M^L = \frac{\delta^2 T_S[\rho]}{\delta\phi(\mathbf{x})\delta\phi(\mathbf{x}')} \approx \frac{\delta\rho(\mathbf{x})}{\delta\phi(\mathbf{x})} \left. \frac{\delta^2 T_S[\rho]}{\delta\rho(\mathbf{x})\delta\rho(\mathbf{x}')} \right|_{\rho_0} \frac{\delta\rho(\mathbf{x}')}{\delta\phi(\mathbf{x}')}, \tag{5.24}$$

which, after substituting (5.22), is discretized as

$$M^L = 4 D_\phi \mathcal{F}^{-1} D_{-1/\chi_{Lind}(q)} \mathcal{F} D_\phi. \tag{5.25}$$

Table 1: List of analytical preconditioners tested in benchmarks.

Preconditioner	Discretized Preconditioner	Inverse Preconditioner
TF	(5.10)	$\frac{9}{70c_{TF}}D_{\phi^{-4/3}}$
vW	(5.11)	$\mathcal{F}^{-1}D_{q^{-2}}\mathcal{F}$
vW+J0	(5.19)	$\mathcal{F}^{-1}D_{(q^2+(8\pi\rho_0)/q^2)^{-1}}\mathcal{F}$
TF0vW	(5.20)	$\mathcal{F}^{-1}\left(D_{\left(q^2+(70c_{TF}\rho_0^{2/3})/9\right)^{-1}}\right)\mathcal{F}$
TF0vW+J0	(5.21)	$\mathcal{F}^{-1}\left(D_{\left(q^2+(70c_{TF}\rho_0^{2/3})/9+(8\pi\rho_0)/q^2\right)^{-1}}\right)\mathcal{F}$
L	(5.25)	$\frac{1}{4}D_{\phi^{-1}}\mathcal{F}^{-1}D_{-\chi_{Lind(q)}}\mathcal{F}D_{\phi^{-1}}$
L+J	(5.26)	$\frac{1}{4}D_{\phi^{-1}}\mathcal{F}^{-1}D_{\left(-1/\chi_{Lind(q)}+2\pi/q^2\right)^{-1}}\mathcal{F}D_{\phi^{-1}}$
L0	(5.27)	$\frac{1}{4\rho_0}\mathcal{F}^{-1}D_{-\chi_{Lind(q)}}\mathcal{F}$
L0+J0	(5.28)	$\frac{1}{4\rho_0}\mathcal{F}^{-1}D_{\left(-1/\chi_{Lind(q)}+2\pi/q^2\right)^{-1}}\mathcal{F}$

This is similar to the second term of the Hartree Hessian, and as a result, the sums of the Lindhard and Hartree matrices can also be inverted for use in preconditioners. Three additional preconditioners based on Lindhard linear response are

$$M^{L+J}=4D_{\phi}\mathcal{F}^{-1}\left(D_{-1/\chi_{Lind(q)}}+2\pi D_{q^{-2}}\right)\mathcal{F}D_{\phi}, \tag{5.26}$$

$$M^{L0}=4\rho_0\mathcal{F}^{-1}D_{-1/\chi_{Lind(q)}}\mathcal{F}, \tag{5.27}$$

$$M^{L0+J0}=4\rho_0\mathcal{F}^{-1}\left(D_{-1/\chi_{Lind(q)}}+2\pi D_{q^{-2}}\right)\mathcal{F}. \tag{5.28}$$

A summary of the preconditioners proposed in this section is included in Table 1. All preconditioners other than the TF preconditioner require an additional two FFTs per CG iteration. From their derivation, any of these preconditioners could be useful when a simulation uses a Lindhard-response-based KEDF.

5.2 Preconditioners for bulk samples

We test the preconditioners derived in Section 5.1 in comparison to several "preconditioners" that solve for the preconditioned residual $y=M^{-1}r$ (5.2) using an additional CG loop. These numerical CG preconditioners (which add another nested CG loop to the pre-existing CG loop of TN) have the forms TFvW, WGC, J (Hartree), TFvW+J, and WGC+J, and are computed iteratively since they contain terms that are local in both real and reciprocal space and cannot be easily inverted analytically. The CG preconditioners are iterated until the residual is sufficiently small (less than 10% of the initial residual), the residual stops changing (less than 10% change in subsequent iterations), or the pre-

Table 2: Benchmark for electron density optimization for equilibrium bulk fcc Al modeled using a 4-atom unit cell. The numbers of CG iterations, line search (LS) iterations, and FFTs for an electron density optimization within OFDFT are compared for the WGC99 and HC KEDFs for a variety of analytical (above the line) and numerical CG (below the line) preconditioners. The * indicates that the CG preconditioner is never positive definite. The unpreconditioned values are shown in boldface.

	WGC99			HC		
Preconditioner	#CG	#LS	#FFTs	#CG	#LS	#FFTs
L0+J0	7	6	222	8	7	3237
L0	8	6	240	8	7	3237
vW	8	7	256	8	7	3237
vW+J0	8	8	272	8	7	3237
TF0vW	10	8	308	10	9	4117
TF0vW+J0	11	8	326	10	7	3679
L+J	20	9	504	20	9	6359
L	22	9	540	26	10	7904
None	**57**	**11**	**1088**	**54**	**10**	**13648**
TF	63	10	1168	63	10	15637
WGC+J (CG)	15	8	1456	*	*	-
WGC (CG)	12	10	1402	*	*	-
TFvW+J (CG)	18	10	760	18	10	6264
TFvW (CG)	15	8	494	17	10	5933
J (CG)	884	99	38134	875	99	234636

conditioner M is no longer positive definite. We focus on the results of the iterative preconditioners and ignore the cost of their additional iterations; these preconditioners are used to check whether analytical preconditioners with a similar form are worth pursuing in future research.

One of the simplest cases for the study of preconditioners is the same 4-atom fcc cell of bulk Al benchmarked in Section 4, with optimization stopping when the norm of the potential is less than 10^{-6} a.u.. Simulations using the WGC99 KEDF are compared with those using the HC KEDF (Table 2) [13]. For this setup, the TFvW and TFvW+J CG-based preconditioners accelerate convergence for bulk Al with both KEDFs. However, the CG-based J preconditioner does not accelerate convergence for either the WGC99 or HC KEDF. Also, differing results are obtained with the WGC preconditioner depending on whether the WGC99 or HC KEDF is used. This indicates a WGC preconditioner may be too specialized to be used as an all-purpose preconditioner for KEDFs derived using Lindhard response. We do not test the J or WGC preconditioner in further benchmarks.

All analytical preconditioners that contain a Lindhard-based term or a vW term accelerate convergence. The L0, L0+J0, vW, vW+J0, TF0vW, and TF0vW+J0 perform the best, cutting the total number of FFTs to less than 30% of the non-preconditioned value when using either the WGC99 or HC KEDF. This is better than the performance of the CG-based preconditioners, in terms of both CG loop iterations and line search steps. The L and L+J preconditioners also improve optimization time from the non-preconditioned

Table 3: Preconditioner performance for a 1848-atom cell of bulk Al with an fcc-like structure. The first set of results corresponds to a simulation cell where the displacement of an atom in any given dimension is Gaussian with standard deviation σ of 0.1 Å, and the second has standard deviation 0.25 Å. The numbers of CG iterations, line search (LS) iterations, and FFTs for an electron density optimization are shown, compared to the unpreconditioned case (values in boldface).

	$\sigma=0.1$			$\sigma=0.25$		
Preconditioner	#CG	#LS	#FFTs	#CG	#LS	#FFTs
L0+J0	12	8	344	16	9	432
vW+J0	15	8	398	19	8	470
TF0vW+J0	18	9	468	22	9	540
L+J	25	9	594	47	11	1022
None	**77**	**12**	**1424**	**95**	**17**	**1792**
TF	85	12	1552	113	15	2048
L0	84	12	1704	95	12	1902
L	84	12	1704	142	15	2796
TF0vW	92	12	1848	94	12	1884
vW	723	54	13878	963	58	18262
TFvW (CG)	81	13	2520	91	11	2830
TFvW+J (CG)	17	10	764	23	12	1024

system, but less dramatically; 46% of the original FFTs are needed for WGC99 and 47% for HC. The TF preconditioner is unsuccessful and slows optimization convergence.

From these results, it appears that TF and J individually do not account for the ill-conditioned nature of the Hessian. This is as expected, since the valence electron density of bulk Al (and most systems for which OFDFT is most accurate) is close to a uniform electron gas. Therefore, ρ and ϕ are fairly smooth, and functionals and energy densities that are locally dependent on those variables similarly do not vary much. On the other hand, energy densities that depend on $\nabla\phi$ have larger variations that are more important to represent in preconditioners. However, the L and L+J preconditioners, which have an additional $1/\phi$ real space dependence through the use of the chain rule, may over-exaggerate these fluctuations. All of the most successful analytical preconditioners are local in reciprocal space and have no dependence on ϕ.

Now we examine a more complicated bulk system. The previous Al_4 cell is reproduced with 6 cells in the x-direction, 7 in the y-direction, and 11 in the z-direction, resulting in a cell containing 1848 atoms. In one benchmark, atoms are moved from their equilibrium positions by a random Gaussian displacement with standard deviation 0.1 Å in each dimension. In the second large bulk benchmark, atoms are displaced with standard deviation 0.25 Å. The same convergence criterion (potential's norm less than 10^{-6} a.u.) is used. The effect of preconditioning the CG loop of TN optimization for these systems is summarized in Table 3.

Here, only the preconditioners that have both a kinetic energy and a Hartree component are faster than the non-preconditioned optimization, with L0+J0 needing only 24% of the original number of FFTs for optimization. The TF, TF0vW, L0, and L pre-

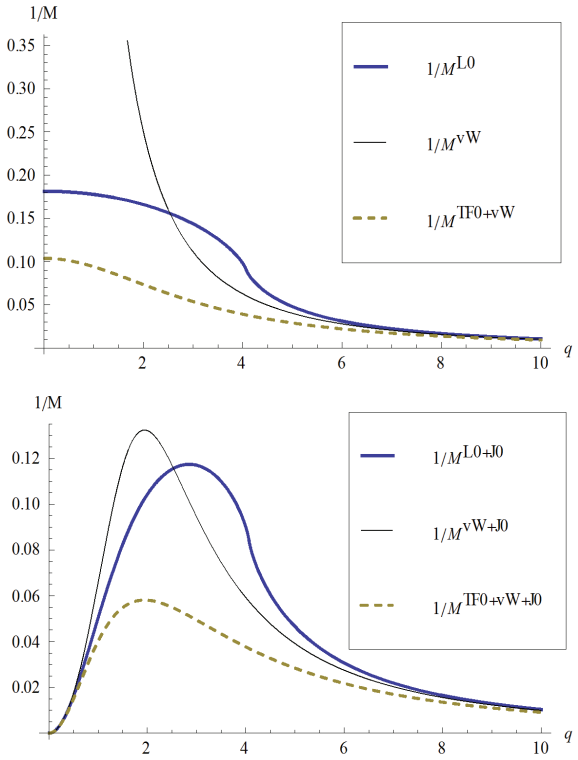

Figure 4: Preconditioners obtained by inverting kinetic energy and Hartree energy Hessian matrices in reciprocal space. The L0 and TF0+vW preconditioners with $\rho_0 = 0.2839$ a.u. (mean electron density for all bulk Al tests for preconditioner benchmarks) are plotted with the vW preconditioner on the upper figure. The lower figure plots preconditioners with both kinetic and Hartree (J0) terms.

conditioners are slightly worse than the non-preconditioned system. The vW preconditioner eventually converges, but requires more than nine times as many FFTs as the non-preconditioned system. The CG-based preconditioners follow a similar trend as the analytical preconditioners. The TFvW preconditioner produces similar performance to the unpreconditioned system, and TFvW+J improves performance, although not as much as with analytical preconditioners.

The success of a combined kinetic energy and Hartree energy Hessian preconditioner can be explained after comparing the preconditioners that are evaluated purely in reciprocal space (Fig. 4). The deficiencies of the vW, TF0vW, L0, or L preconditioners alone only become apparent for larger simulation cells, where the reciprocal space grid becomes denser and behavior at small q becomes increasingly important. Without the J0 term, either the preconditioner increases to some positive finite value for small q, as is the case for the TF0vW and L0 preconditioners, or it approaches infinity, as is the case for the vW preconditioner. When a Hartree term is included, J0 dominates at small q and the kinetic energy term dominates at large q. This best reproduces the ill-conditioned parts of the second derivative of the total Lagrange function for bulk simulation cells.

The L+J preconditioner (5.26) behaves similarly to the L0+J0 preconditioner (5.28) since the fluctuations of the pseudo-wavefunction ϕ are relatively small.

From these benchmarks, it appears that the L0+J0, vW+J0, and TF0vW+J0 preconditioners represent a big improvement over the vW preconditioner for simulation cells with bulk, with L0+J0 a little better than vW+J0, which in turn is slightly better than TF0vW+J0. For all the preconditioners tested, optimization scaling still has a system-size dependence.

5.3 Preconditioners for simulation cells containing vacuum

We now study whether any of the preconditioners suggested in Table 1 are successful for improving optimization for simulation cells containing regions with low electron density ("vacuum"). Vacuum regions are present in any simulation cells that model finite or semi-infinite systems such as surfaces, molecules, nanowires, or nanoclusters. For such cells, the electron density can no longer be assumed to be constant throughout the cell, and local changes in reciprocal space may no longer reproduce the most ill-conditioned parts of the Hessian.

But before presenting benchmark results, we first discuss a problem with TN optimization caused by vacuum. If a pseudo-wavefunction has regions where $\phi \approx 0$, the Hessian matrix H_ϕ is not only more ill-conditioned – it may no longer be positive definite. If H_ϕ is detected to not be positive definite during the inner CG loop (for calculating the Newton direction), the CG optimization cannot continue and terminates with a less accurate descent direction. Since the inaccurate descent direction is used in the subsequent line search, convergence can stall once the value of ϕ in the vacuum regions becomes too low. Tests of various Al simulation cells indicate the associated H_ϕ may no longer be positive definite when there are points $\mathbf{x}$ with $\phi(\mathbf{x}) < 10^{-7}$ a.u..

H_ϕ is more likely to remain positive definite if the electron density is prevented from dropping too low. As a result, minor numerical artifacts in vacuum can counterintuitively help convergence. We showed in Section 2 that if ϕ is not constrained to be non-negative, the vW KEDF expressed as (2.5) can be computed without numerical artifacts. The corresponding potential (first functional derivative of (2.5)), which is used in determining the Newton direction, can also be calculated without artifacts when discretized as

$$\frac{\delta E^{vW}}{\delta \phi} = -\nabla^2 \phi = \mathcal{F}^{-1} D_{q^2} \mathcal{F} \phi. \tag{5.29}$$

If we would like to prevent electron density from becoming extremely low, the chain rule can be used to calculate the same potential, reintroducing numerical artifacts:

$$\frac{\delta E^{vW}}{\delta \phi} = \mathcal{F}^{-1} D_{q^2} \mathcal{F} |\phi| \cdot \frac{d|\phi|}{d\phi}. \tag{5.30}$$

We will refer to the first expression (5.29) as the direct computation of the vW potential, and the second expression (5.30) – where the functional derivative is taken relative to

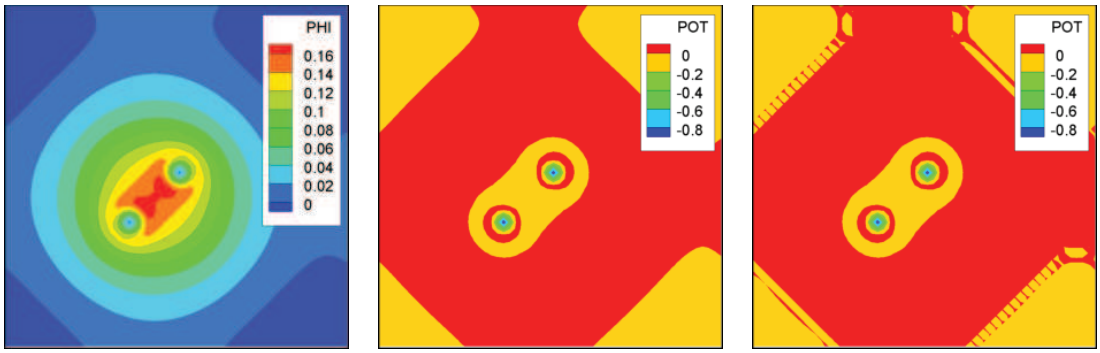

Figure 5: Potential ($\delta L/\delta\phi$) calculated for a 4-atom periodic cell of an Al nanowire (color online). The left panel displays the values of the pseudo-wavefunction on a slice perpendicular to the nanowire axis. The middle panel shows the potential calculated for that pseudo-wavefunction along the same slice using $\mathcal{F}^{-1}D_{q^2}\mathcal{F}\phi$, and the right panel plots the potential for the same pseudo-wavefunction and the same slice, calculated using $\mathcal{F}^{-1}D_{q^2}\mathcal{F}\phi\cdot d|\phi|/d\phi$.

$|\phi| = \sqrt{\rho}$ and the sign of ϕ is added later – as the chain rule computation of the vW potential.

In Fig. 5, we show how the total potential differs when the vW term is calculated with the two methods above. While the area of interest near the atomic cores remains quite similar, small differences are observed where $\phi \approx 0$. A smooth potential is produced when the Laplacian of ϕ is computed directly, and singularities appear when using the chain rule computation. The added noise for the chain rule computation means that the Newton direction chosen in vacuum regions is less accurate, and ϕ does not decrease as dramatically as when optimizing using the direct calculation of the vW potential. Since the magnitude of ϕ is larger, the Hessian matrix used during the CG loop is less likely to become non-positive definite and cause the CG loop to terminate early. However, ϕ is still sufficiently small in vacuum so that it has a minimal effect on the total energy. The overall result is that an optimization of a cell containing large regions of vacuum is then better able to proceed when there are minor numerical artifacts in vacuum, so optimizations using chain rule vW calculations generally converge faster than optimizations using direct vW calculation. Chain rule vW calculations were used in the benchmarks in Section 4.2.

The appearance of non-positive definite H_ϕ when ϕ is small adds to the difficulties in deriving preconditioners for simulation cells containing vacuum. If H_ϕ is not positive definite, optimization may converge slowly regardless of the preconditioner used. If the chain rule calculation is used to combat the non-positive definite problem, numerical artifacts will appear that cannot be treated with the analytical preconditioners. Therefore, we expect preconditioners to be of limited use for systems containing vacuum, especially when using the chain rule vW computation. Nevertheless, we test the preconditioners from the previous two sections to assess their effectiveness with such systems. The two methods of calculating the vW potential are compared, and the relative performance of

Table 4: 4-atom periodic cell Al nanowire preconditioner tests with the WGC99 KEDF. The numbers of CG iterations, line search (LS) iterations, and FFTs for an electron density optimization within OFDFT are compared for direct and chain rule computation of the vW term. The unpreconditioned values are shown in boldface. "No conv." denotes that optimizations did not reach a minimum.

	Direct			Chain rule		
Preconditioner	#CG	#LS	#FFTs	#CG	#LS	#FFTs
L0	24	11	608	50	11	1076
TF0vW	26	13	676	47	13	1054
None	**209**	**26**	**3760**	**121**	**15**	**2176**
L0+J0	315	63	6678	383	50	7694
vW+J0	391	64	8062	292	41	5912
TF0vW+J0	No conv.	No conv.	No conv.	360	50	7280
TFvW (CG)	126	18	3444	140	15	3854
TFvW+J (CG)	20	19	1704	26	19	2212

Table 5: 4-atom periodic cell Al nanowire preconditioner tests with the HC KEDF. The numbers of CG iterations, line search (LS) iterations, and FFTs for an electron density optimization within OFDFT are compared for direct and chain rule computation of the vW term. The unpreconditioned values are shown in boldface. "No conv." denotes that optimizations did not reach a minimum.

	Direct			Chain rule		
Preconditioner	#CG	#LS	#FFTs	#CG	#LS	#FFTs
L0	26	17	34333	103	19	101340
None	**139**	**18**	**130255**	**124**	**16**	**108932**
TF0vW	195	36	206483	64	17	65739
vW+J0	No conv.	No conv.	No conv.	282	39	238239
TF0vW+J0	No conv.	No conv.	No conv.	305	47	247074
L0+J0	No conv.	No conv.	No conv.	632	68	477428
TFvW (CG)	239	27	237354	103	16	103413
TFvW+J (CG)	56	26	73694	21	20	35319

each method, along with the effect of preconditioners, is benchmarked.

We begin by testing a small 4-atom periodic cell of an Al nanowire in an fcc-like configuration separated by its closest images by 10 Å vacuum (simulation cell size 13.97 Å $\times$ 13.97 Å $\times$ 3.97 Å). Benchmarks are terminated when the energy is converged to 10^{-5} eV or the potential's norm is converged to 10^{-6} a.u.. Results are listed in Table 4 (WGC99 KEDF with vacuum damping [23]) and Table 5 (HC KEDF [13]). Benchmarks for the preconditioners with an explicit dependence on ϕ (TF, vW, L, and L+J) do not converge for either KEDF, so they are not included in the tables or in other benchmarks of simulation cells containing vacuum.

In stark contrast with the bulk benchmarks, preconditioners that include both a kinetic and a J0 term slow the overall convergence, with some of these calculations not converging to the true minimum. Benchmarks for the L0+J0, vW+J0, and TF0vW+J0 preconditioners are not reported for other systems containing vacuum. Only the L0 and TF0vW preconditioners converge for both the WGC99 and the HC KEDF. The L0 preconditioner

is more successful overall, since it offers an improvement over the non-preconditioned optimization for both KEDFs and for both direct and chain rule vW computations. The TF0vW preconditioner is faster than L0 when using the chain rule vW calculation, but slower than an unpreconditioned optimization with the direct vW calculation. As with simulation cells containing only bulk, the preconditioning ability of L0 and TF0vW is likely due to their being almost constant at small q; they act only as large q preconditioners. On the other hand, the failure of the vW preconditioner and any preconditioners with a J0 term indicates that neither vW nor J0 accurately represents small q values for cells in vacuum.

Like the full Hessian matrix, the CG-based preconditioners also have a tendency to become non-positive definite in the presence of low electron density. This becomes a problem for the TFvW preconditioner. Since it is not positive definite when solving for the preconditioned residual in (5.2), the TFvW (CG) benchmarks do not accurately represent a true TFvW preconditioner. On the other hand, the TFvW+J (CG) preconditioner does not encounter this issue for the 4-atom nanowire simulation cell. If the cost of the preconditioner is not considered, the TFvW+J preconditioner is more effective than the analytical preconditioners in all cases except with the HC KEDF and direct calculation of vW.

The results above show that, while better preconditioners may still be derived, L0 speeds up optimization of small simulation cells containing vacuum with two KEDFs derived from Lindhard linear response, and that optimizations are faster with the TF0vW preconditioner when vW is computed using the chain rule. The results using the WGC99 KEDF are consistent with the results from bulk preconditioning tests, where the L0 and TF0vW preconditioners also improve convergence for the smaller simulation cells.

Although the L0 and TF0vW preconditioners do not similarly enhance convergence for large cells of bulk material, we test larger simulation cells containing a nanowire or a crack using the vacuum-modulated WGC99 KEDF to see whether improvements due to large q preconditioning are sufficient to improve optimization performance for simulation cells with vacuum. The same 1110-atom nanowire is used here as in the benchmarks in Section 4.2. The 7828-atom crack also uses the same energy functionals and parameters as the crack benchmarked in Section 4.2, but the cell is 2.81 Å $\times$ 232.55 Å $\times$ 222.80 Å, and has a plane-wave kinetic energy cutoff of 1000 eV (instead of the 600 eV used in all other benchmarks). As with the 4-atom nanowires, benchmarks are terminated when the energy is converged to 10^{-5} eV or the potential's norm is converged to 10^{-6} a.u..

Without a preconditioner, the chain rule calculation requires fewer FFTs than the direct calculation of the vW terms for the nanowire. However, the fastest computation occurs for a L0 preconditioned optimization when computing the vW terms directly. L0 and TF0vW are faster than non-preconditioned optimization for both direct and chain-rule vW computations. Like with the thinner 4-atom periodic cubic cell Al nanowire, the L0 preconditioner is best suited for the direct vW computation, and the TF0vW preconditioner is better for the chain rule vW computation.

The iterative CG preconditioners all have convergence problems associated with the

Table 6: [111] Al nanowire with 1 nm diameter. See previous tables for description of columns.

	Direct			Chain rule		
Preconditioner	#CG	#LS	#FFTs	#CG	#LS	#FFTs
L0	371	41	7334	419	101	9158
TF0vW	379	80	8102	380	42	7512
None	**1564**	**120**	**26944**	**618**	**59**	**10832**
TFvW (CG)	No conv.	No conv.	No conv.	508	49	10676
TFvW+J (CG)	895	87	45868	186	60	10232

Table 7: Al edge crack under tensile loading. See previous tables for description of columns. The * indicates that the CG preconditioner is never positive definite.

	Direct			Chain rule		
Preconditioner	#CG	#LS	#FFTs	#CG	#LS	#FFTs
None	**874**	**71**	**15120**	**879**	**66**	**15120**
L0	930	68	17828	1119	70	21262
TF0vW	1128	85	21664	1071	70	20398
TFvW (CG)	*	*	-	*	*	-
TFvW+J (CG)	677	206	88848	262	122	27180

Hessian becoming non-positive definite. However, TFvW and TFvW+J are still able to improve on the unpreconditioned optimization when using the chain rule vW computation. The performance is sped up, even when the cost of the iterative preconditioning is taken into account.

For the edge crack, no difficulties are observed when using the direct computation, which converges faster than the chain rule computation. However, the analytical preconditioners do not aid convergence for either method of vW calculation. This is as expected, due to the increased importance of small q for large cells. While the nanowire also had a large dimension along one axis, preconditioners could still be used due to the nanowire's periodic atomic structure along the longest axis. The crack is not periodic in its longer real-space directions. The TFvW (CG) preconditioner is found to always be non-positive definite, regardless of the vW calculation method. The TFvW+J (CG) preconditioner, while occasionally having the same problem, is generally able to produce a preconditioned residual. It appears that a preconditioner with the form of TFvW+J would be effective, but an easily-inverted analytical form is unknown and the iterative implementation is too expensive to be useful.

For systems containing vacuum, optimization with and without preconditioners can be difficult. Even if the Hessian itself remains positive definite, preconditioners have to account for large variations in ϕ in real space. The two preconditioners observed to have moderate success do not have any explicit ϕ dependence, and their success is limited to small simulation cells. When the vW term is calculated directly, the L0 preconditioner works better, and when the vW term is calculated using the chain rule, the TF0vW

preconditioner is preferred. They both represent an improvement over the vW preconditioner, which does not allow any convergence in the presence of vacuum even for smaller cells. However, neither one helps in the key objective of preconditioning – improving the optimization speed for large simulation cells.

6 Conclusions

In this work, we present the optimization setup and algorithms used with OFDFT, and specifically implemented in PROFESS. Beginning with an introduction to the optimization problem defined by OFDFT, we discuss the performance of CG and TN optimization and derive preconditioners that are successful in bulk and accelerate convergence for small cells in vacuum.

Optimization schemes in OFDFT are used to determine the three-dimensional electron density distribution that minimizes the total energy. Following an established framework, we use a Lagrange function to conserve the total number of electrons, and a pseudo-wavefunction ϕ (with $\phi^2=\rho$) as the optimization variable to keep electron density non-negative [2,8,19]. We emphasize that any ϕ calculated using CG and TN is differentiable and does not need to be constrained to be positive; it can take any real value.

In addition, the approximation μ_0 used to estimate the Lagrange multiplier is discussed. Since minimizing the approximate Lagrange function is much faster than a full saddle point optimization, the Lagrange multiplier is expressed as a function of E and $\delta E/\delta\phi$. Unfortunately, this approximation makes the optimization less stable. The choice for how often μ_0 is updated also affects stability and speed, where fewer updates allow a more stable optimization, and more updates will allow the optimization to converge faster (assuming that the optimization is sufficiently stable).

For better stability, a line search that conserves total electron number is used [8], and a modified set of Wolfe conditions are derived to accommodate the curved inexact line search. Together with a CG or TN direction search, every iteration of the optimization algorithm conserves the total number of electrons in the system.

Benchmarks show that TN optimization converges faster than CG for simulation cells containing Al atoms. However, the costs of optimization for CG and TN scale similarly, since both algorithms rely on CG loops. Either the entire minimization is performed with nonlinear CG, or else an inexact Newton direction is calculated using linear CG, so the optimization scaling for CG and TN depend more on the complexity of the system than simply system size. In TN, for example, the distribution of eigenvalues of the Hessian matrix associated with a pseudo-wavefunction ϕ determines the convergence properties of the inner CG loop.

The CG loop within TN can be preconditioned to speed up optimization. We derive several prospective preconditioners based on the second derivative of energy functionals and the Lindhard linear response. All the analytical preconditioners have implementations that run with quasilinear scaling, since each one is a sequence of computations local

in real or reciprocal space, interspersed with FFTs.

For bulk samples, some of the new preconditioners improve on the vW preconditioner, which was shown to be successful in small bulk simulation cells [10]. While the vW preconditioner fails for larger cells, preconditioners that have both a Hartree and kinetic term speed up convergence for the larger bulk cells examined in this work. The fastest preconditioners – vW+J0, TF0vW+J0, and L0+J0 – have no dependence on ϕ and are fully diagonal in reciprocal space. Their large q values reflect the kinetic energy Hessian, and their small q values reflect the Hartree Hessian. While these three preconditioners have similar performance, the L0+J0 appears to be the best preconditioner for bulk samples, reducing the number of FFTs needed during optimization of bulk samples by $\sim 76-80\%$ in the systems of bulk Al studied.

Unfortunately, the same level of success is not obtained for preconditioners in the presence of vacuum. The J0 preconditioner, essential for preconditioning the small q values in bulk, is no longer is effective since there is a large difference in the electron density for vacuum and the region close to the ions. As a result, kinetic + J0 preconditioners fail. For small simulation cells, the L0 and TF0vW preconditioners can improve convergence by preconditioning large q values. However, when small q values begin to become more important (larger cells), neither preconditioner improves convergence. Still, this minimal success is an improvement on the vW preconditioner. If L0 and TF0vW preconditioners are used in large simulation cells with vacuum, optimization can still converge. If the vW preconditioner were used, optimization would fail completely.

For future work in speeding up optimization of electron density in the presence of vacuum, two challenges not present in bulk systems must be taken into account. First, the Hessian easily becomes non-positive definite if ϕ has large regions of vacuum. If a Hessian is not positive definite, TN cannot compute the Newton direction and convergence slows. This problem should be addressed in some way beyond the trick of adding numerical noise to artificially prevent electron densities from dropping too low in vacuum. In addition to problems with the positive definite nature of the Hessian matrices, any preconditioners proposed for simulation cells containing vacuum must contend with the large fluctuations of ϕ in real space. The fluctuations are difficult to represent in reciprocal space, and suggest that appropriate vacuum preconditioners should have a real-space ϕ dependence. However, the best preconditioners presented here are diagonal in reciprocal space. A preconditioner successful for vacuum will need terms local in both real and reciprocal space, and there is no guarantee that such a preconditioner can be easily and cheaply inverted.

Acknowledgments

We would like to thank the National Defense Science and Engineering Graduate Fellowship program (L.H.) and the National Science Foundation (E.A.C.) for funding.

References

[1] Y. A. Wang and E. A. Carter, in Theoretical Methods in Condensed Phase Chemistry, edited by S. D. Schwartz, (Kluwer, 2000) p. 117-184.
[2] G. S. Ho, V. L. Lignères, and E. A. Carter, Comput. Phys. Commun. 179, 839 (2008).
[3] L. Hung and E. A. Carter, Chem. Phys. Lett. 475, 163 (2009).
[4] Y. A. Wang, N. Govind, and E. A. Carter, Phys. Rev. B 60, 16350 (1999).
[5] C. Huang and E. A. Carter, Phys. Chem. Chem. Phys. 10, 7109 (2008).
[6] L. Hung, C. Huang, I. Shin, G. S. Ho, V. L. Lignères, and E. A. Carter, Comput. Phys. Commun. 181, 2208 (2010).
[7] J. Nocedal and S. Wright, Numerical Optimization (New York: Springer, 2006).
[8] H. Jiang and W. Yang, J. Chem. Phys. 121, 2030 (2004).
[9] C. F. von Weizsäcker, Z. Phys. 96, 431 (1935).
[10] V. L. E. Lignères, Advances in orbital-free density-functional theory (Princeton University, United States – New Jersey, 2008).
[11] L. Wang and M. P. Teter, Phys. Rev. B 45, 13196 (1992).
[12] D. García-Aldea and J. E. Alvarellos, Phys. Rev. A 76, 052504 (2007).
[13] C. Huang and E. A. Carter, Phys. Rev. B 81, 045206 (2010).
[14] L. H. Thomas, Proc. Cambridge Philos. Soc. 23, 542 (1927).
[15] E. Fermi, Z. Phys. 48, 73 (1928).
[16] J. Lindhard, K. Dan. Vidensk. Selsk. Mat. -Fys. Medd. 28, 8 (1954).
[17] N. W. Ashcroft and D. N. Mermin, Solid State Physics (Brooks Cole, 1976).
[18] S. C. Watson and E. A. Carter, Comput. Phys. Commun. 128, 67 (2000).
[19] C. J. García-Cervera, Commun. Comput. Phys. 2, 334 (2007).
[20] D. Garcia-Aldea and J. E. Alvarellos, Phys. Rev. A 77, 022502 (2008).
[21] M. P. Allen and D. J. Tildesley, Computer Simulation of Liquids (Clarendon Press, Oxford, 1987).
[22] Y. A. Wang, N. Govind, and E. A. Carter, Phys. Rev. B 64, 089903 (2001).
[23] I. Shin, A. Ramasubramaniam, C. Huang, L. Hung, and E. A. Carter, Philos. Mag. 89, 3195 (2009).
[24] D. M. Ceperley and B. J. Alder, Phys. Rev. Lett. 45, 566 (1980).
[25] J. P. Perdew and A. Zunger, Phys. Rev. B 23, 5048 (1981).
[26] W. W. Hager and H. Zhang, SIAM J. Optim. 16, 170 (2005).