OFDFT Minimization Algorithms

Marc Lane

Email: lanem2@tcd.ie

1 Overview

In this document, I tried to collect together some of the main techniques used in direct energy minimizations in OFDFT simulations. This document also functions as a short and simplified review of the papers that accompanied the release of the OFDFT software PROFESS[1], DFTpy [2], Atlas [3], CONUNDrum [4] and DRAGON [5]. Many of the important details about how the minimization is implemented comes from this useful paper [6]. I reused the same notation from the textbook [7]. To focus on the fundamental principles of the direct minimization in OFDFT, I have omitted discussions about complex functionals, pseudopotentials, and other implementation-specific details required in practical OFDFT algorithms. At the end of the document, I also briefly explain some of the newer, alternative OFDFT algorithms that employ some extra tricks that avoid some of the difficulties that come with direct minimization.

2 Minimization Algorithm

The direct minimization of the OFDFT Lagrangian is a highly multi-dimensional and non-linear problem (where the number of parameters we need to optimize is proportional to the number of gridpoints). Thus, developing a stable, robust and fast numerical minimization algorithm presents several difficulties. The first difficulty is to uphold the requirements that the electron density (ρ) remains normalized $\Big(\int \rho(\mathbf{r})d\mathbf{r} = N_e\Big)$ and non-negative $\Big(\rho(\mathbf{r}) \geq 0\Big)$ throughout the minimization process. The second difficulty is to ensure the algorithm remains numerically stable.

In essence, an OFDFT algorithm must minimize the Lagrangian

$$L[\rho] = E[\rho] - \mu\bigg[\int \rho(\mathbf{r})d\mathbf{r} - N_e\bigg], \tag{1}$$

where $E[\rho]$ is the energy functional (explained in background theory document), μ is the chemical potential and N_e is the number of electrons.

Conceptually, the direct minimization algorithm operates numerically with a starting density $\rho(\mathbf{r})$, represented as an array with elements that represent its value at discrete points along $\mathbf{r}$. To find the ρ that minimizes L, we need to move the starting ρ closer to the minimizing ρ by adding to it another array p. The step direction p is the functional derivative of the Lagrangian, which is

$$p = \frac{\delta L[\rho]}{\delta \rho} = \frac{\delta E[\rho]}{\delta \rho} - \mu. \tag{2}$$

We can use p to get iteratively closer to the ground-state density starting from an arbitrary initial guess density. In principle, the algorithm should proceed at iteration k as

$$\rho_{k+1} = \rho_k + \alpha_k p_k \tag{3}$$

where ρ_{k+1} is an improved density that is closer to the true ground-state density than ρ_k, and α_k is the step size. Convergence of this algorithm is achieved when the condition that

$$\left[\int \left(\frac{\delta L[\rho_{k+1}]}{\delta \rho_{k+1}}\right)^2 d\mathbf{r}\right]^{\frac{1}{2}} < \tau \tag{4}$$

is satisfied for a small tolerance value τ. Once successful minimization has been indicated by the evaluation of Eq. (4), we can conclude that the Euler-Lagrange equation of Eq. (2) is satisfied and ρ_{k+1} has converged on the ground-state density with precision determined by the tolerance value τ.

2.1 Change of variable ($\rho \to \phi$)

In general, OFDFT programs deal with the issue of upholding normalization and non-negativity by reformulating the problem in terms of a new optimization variable $\phi(x)$ [1–5]. We can recast the Euler-Lagrange equation in terms of $\phi(x)$ instead of $\rho(x)$ by multiplying it by $\frac{\delta\rho}{\delta\phi}$ and applying the chain rule, such that

$$\frac{\delta \rho}{\delta \phi}\left(\frac{\delta E[\rho]}{\delta \rho} - \mu\right) = \frac{\delta E[\phi]}{\delta \phi} - \frac{\delta \rho}{\delta \phi}\mu = 0. \tag{5}$$

There are many choices of ϕ available (such as $\rho = \phi^2$ or $\rho = e^{\phi}$), but the consensus is that $\rho(x) = \phi(x)^2$ is the most stable, efficient and commonly used option in OFDFT minimization programs. The benefit of this reformulation is that $\phi(x)$ can take negative values, but $\phi(x)^2$ will still obey the non-negativity constraint. In this case, one can evaluate analytically that $\frac{\delta\rho}{\delta\phi} = 2\phi$ and $\frac{\delta E[\phi]}{\delta\phi} = 2\phi\frac{\delta E[\rho]}{\delta\rho}$, as shown in [8]. By substituting these expressions in, we reformulate the minimization problem as

$$\frac{\delta L[\phi]}{\delta \phi} = 2\phi\left(\frac{\delta E[\phi^2]}{\delta \phi^2}\right) - 2\phi\mu = 0. \tag{6}$$

This change of variable also modifies the convergence condition. Expressing it in terms of ϕ, the condition of successful minimization now becomes

$$\left[\int_0^1 \left(\frac{\delta L[\phi]}{\delta \phi}\right)^2 dx\right]^{\frac{1}{2}} < \tau. \tag{7}$$

Once this condition is satisfied, our optimization variable $\phi(x)$ has been optimized. We can then take its square to calculate the ground-state density $\phi(x)^2 = \rho(x)$.

As a side note, the new optimization variable $\phi(x)$ may be labelled the 'pseudo-wavefunction' [6], since it has the property that $\phi^2 = \rho$. Taking this interpretation further, we could rearrange the Euler-Lagrange equation for ϕ into a Schrödinger-like equation. Dividing both sides by two and rearranging we get

$$\phi \frac{\delta E[\phi^2]}{\delta \phi^2} = \phi \mu, \tag{8}$$

where the chemical potential μ resembles the energy and $\frac{\delta E[\phi^2]}{\delta \phi^2}$ the Hamiltonian operator. This equation has a strong kinship with the Kohn-Sham (KS) equation, but for only one orbital [9]. Historically, it was expected that a variation of the standard KS self-consistent field (SCF) method could be reused to straightforwardly solve Eq. (8). Instead, [10] found that more specialized direct minimization techniques were needed to make OFDFT calculations possible without reverting to $\mathcal{O}(N_e^3)$ scaling [9]. Recently however, new one-orbital ensemble SCF (OE-SCF) methods have been developed, making KS methods viable for OFDFT [11]. Since the OE-SCF technique is still relatively recent, I opted instead to follow the majority of the literature on OFDFT optimization by directly minimizing Eq. (1).

The negative of $\frac{\delta L[\phi]}{\delta \phi}$ is the search direction p_k that is calculated at the beginning of each iteration to move the density closer to the ground-state. In order to calculate p_k, we wish to approximate the chemical potential at the start of each iteration. We use the same expression for the approximate chemical potential u_0 as in [6], given as

$$\mu_0 = \frac{1}{2} \int_0^1 \left(\frac{\delta E[\phi]}{\delta \phi} \phi \right) dx. \tag{9}$$

As explained in [6], this expression for μ_0 is exact at the ground-state, but may be inaccurate when $\phi(x)$ is far from the optimized density. An inaccurate μ_0 causes a violation of the constraint that $\int \rho(x) dx = 1$, so $\phi(x)$ is renormalized by scaling at the end of each iteration.

Once a search direction p_k is calculated (using Eq. (6)) for the k^{th} iteration, one can update ϕ_k with

$$\phi_{k+1} = \phi_k + \alpha_k p_k, \tag{10}$$

where α_k is a step length determined by a line search.

2.2 Line Search

Ideally, a line search could be performed exactly to find the α_k that minimizes the Lagrangian described along the direction of p_k. However, this is generally too computationally expensive, and instead inexact line search methods that find a step size of sufficient decrease and curvature are preferred [7]. The Wolfe conditions described in [6] and [7] are a popular choice for determining the optimum step size, which are given as

$$L[\phi_{k+1}] \leq L[\phi_k] + c_1 \alpha_k \frac{\delta L[\phi_k]}{\delta \phi_k} \cdot p_k \tag{11}$$

$$\frac{\delta L[\phi_{k+1}]}{\delta \phi_{k+1}} \cdot p_k \geq c_2 \frac{\delta L[\phi_k]}{\delta \phi_k} \cdot p_k \tag{12}$$

where $L[\phi_k]$ is the Lagrange function from in terms of ϕ instead of ρ, $c_1 \in (0, 1)$ and $c_2 \in (c_1, 1)$. Eq. (11) is the condition of sufficient decrease (called the Armijo condition) and Eq. (12) is the curvature condition [7]. The optimum value for c_1 is a small number of approximately $c_1 = 10^{-4}$ and the optimum value for c_2 is $c_2 = 0.1$ for the non-linear conjugate gradient method (described later) [7]. An illustration of how a line search is implemented using the Wolfe conditions for a function f is given in Figure 1.

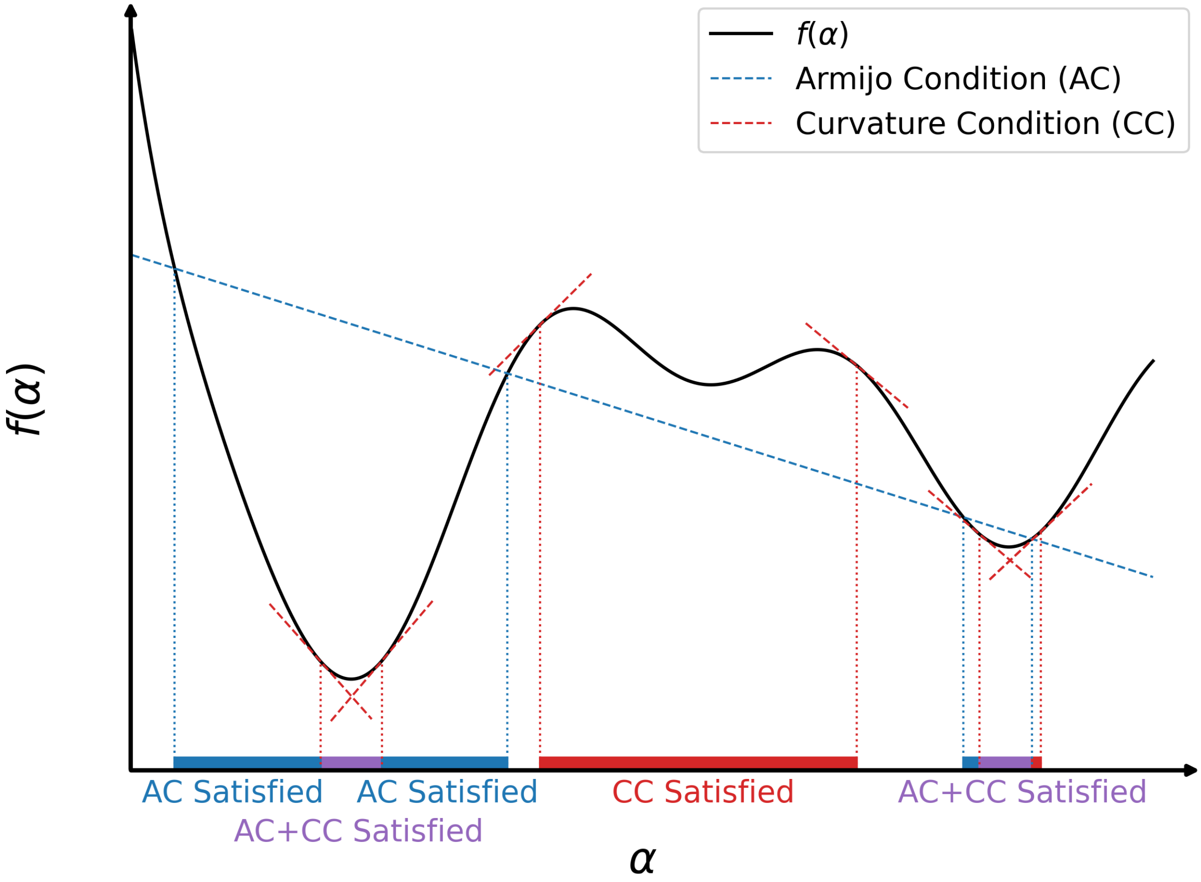

Figure 1: A blueprint of how a line search is performed for a function f (in black) to find the step size α that fulfils the Wolfe conditions. Different values of α are tested to find the optimum f_{k+1} for the next iteration. The blue segments of the α-axis are the sections for which the Armijo condition (AC) (i.e. Eq. (11)) is satisfied, the red sections satisify the curvature condition (CC) (i.e. Eq. (12)) and the purple sections satisify both conditions.

2.3 Orthogonalization Scheme

The algorithm as currently formulated using the Wolfe conditions runs into numerical instabilities. This is primarily due to the fact that the line search, by and of itself, doesn't obey the constraint that $\int \rho(x)dx = 1$, with instead this constraint being imposed at the end of the iteration by arbitrarily rescaling $\phi(x)$ to achieve normalization (which causes more issues). To force the algorithm to comply with the normalization constraint, we carry out the orthogonalization scheme proposed in [12] and implemented in [1, 2, 3]. Instead of moving in the direction of steepest descent p_k, the algorithm follows a curved direction through the constrained space that automatically prescribes that $\int \rho(x)dx = 1$. ϕ is now updated as

$$\phi_{k+1} = \phi_k^{\perp} \sin(\theta_k) + \phi_k \cos(\theta_k), \tag{13}$$

where $\phi_k^{\perp}$ is the part of p_k perpendicular to ϕ_k given by

$$\phi_k^{\perp} = p_k - \left(\frac{\phi_k \cdot p_k}{|\phi_k|^2} \right), \tag{14}$$

and $\phi_k^{\perp}$ is normalized before being substituted into Eq. (13). An additional benefit of this algorithm is the replacement of the step size α_k with θ_k, where we require that $0 < \theta_k < \pi/2$. This modification means the line search for θ_k is now bracketed, thereby enhancing the stability over standard unbracketed line searches [6]. The Wolfe conditions for line search convergence must also be modified to be expressed in terms of θ_k. We follow the Wolfe conditions for the θ_k line search used in [6], expressed as

$$L[\phi_{k+1}] \leq L[\phi_k] + c_1 \theta_k \frac{\delta L[\phi_k]}{\delta \phi_k} \cdot q_k \tag{15}$$

$$\frac{\delta L[\phi_{k+1}]}{\delta \phi_{k+1}} \cdot q_k \geq c_2 \frac{\delta L[\phi_k]}{\delta \phi_k} \cdot q_k, \tag{16}$$

where q_k is the new search direction given as

$$q_k = \frac{\phi_{k+1} - \phi_k}{\theta_k}. \tag{17}$$

2.4 Descent Direction

2.4.1 Steepest Descent Method

Thus far, the descent direction p_k is ultimately calculated from Eq. (6). This approach is a variation of the steepest descent method, in which each new direction is based only on the direction of steepest descent at that iteration. I found that this method consistently diverged or progressed too slowly.

2.4.2 Non-linear Conjugate Gradient Method

An improved scheme is to insist that descent directions from previous iterations must be conjugate to one another, to avoid repeated minimization along the same directions [1]. Due to non-linear terms in Eq. (6), the non-linear conjugate gradient method [7] is used to determine the descent direction p_k. This method has typically been found to be the most stable algorithm used in direct OFDFT minimizations [1, 2, 4, 5]. To implement this we use the Polak-Ribière method [13], where the conjugate gradient direction g_k is expressed as

$$g_k = -p_k + \frac{(p_k - p_{k-1}) \cdot p_k}{|p_{k-1}|^2} g_{k-1}. \tag{18}$$

This formula ensures that g_k is always conjugate to g_{k-1}, except for the first iteration where the conjugate gradient method reduces to the steepest descent method. g_k now replaces p_k as the descent direction used in Eq. (10). The non-linear conjugate gradient method used here is in reality a generalization of the linear conjugate gradient method used in linear algebra to solve the $Ax = b$ problem.

2.4.3 Truncated Newton Method

The truncated Newton method for OFDFT was first proposed in [14] and is implemented in [1,2,3]. Here, the search direction p_k is calculated by inexactly solving the equation

$$\mathcal{H}_k p_k = \frac{\delta L[\phi]}{\delta \phi} \tag{19}$$

where $\mathcal{H}_k$ is the Hessian matrix of the Lagrangian function $L[\phi]$, computed numerically by a second conjugate gradient loop during each iteration [1]. Making use of a suitable preconditioner can make the computation of this linear equation more suitable for the minimization process, as discussed in [6, 15].

The truncated Newton method is faster than the non-linear conjugate gradient method [PROFESS PAPER], but less stable [1]. Since my project suffered from numerical stability issues, I opted to use the non-linear conjugate gradient method which can be easily implemented by converting p_k (from steepest descent) into g_k using Eq. (18).

2.5 Stability Issues

The direct minimization method presented so far, despite being one of the most stable algorithms presented in the literature, was found to fail in achieving convergence, due to instabilities likely caused by the divergence of the vW functional as $\rho \to 0$. Even in situations where the density takes a large value for all x in the [0,1] interval, the requirement that $\rho(x) = 0$ at the boundaries introduces a discontinuity in the density that causes the vW functional to diverge. It is suspected that this issue doesn't often arise in most OFDFT programs since it they primarily simulate periodic many-electron systems where the density never approaches zero. One might suggest the tactic to handle this would be to apply periodic boundary conditions to prevent $\rho \to 0$. If this was implemented, the formulation laid out in Section 1.3 would no longer be exact, and it wouldn't be possible to compare the converged densities with the Schrödinger equation.

The strategy taken to overcome this was to apply a Gaussian filter to p_k at the start of each iteration, to smooth out any outlier gridpoints where the vW functional diverged. Such a Gaussian filter works by convolving the signal with a Gaussian kernel . I used the function `SciPy.ndimage.gaussian_filter1d` with sigma = 5 to stabilize p_k at the start of each iteration. The Gaussian filter removes any of the outlier points in p_k that are introduced by instabilities of the vW functional.

3 Alternate Methods

The method I used to implement OFDFT is a direct minimization method, involving the direct minimization of the Lagrangian. In this section I briefly overview some of the more recently developed in-direct minimization methods that may avoid some of the numerical instability issues experienced by direct minimization. An investigation into the effectiveness of these methods to determine if TRIM can handle the divergences of the vW functional as $\rho \to 0$ would be worthwhile.

3.1 OE-SCF Method

The OE-SCF [9] method (mentioned earlier in this document) reuses techniques from the well-developed Kohn-Sham method to solve Eq. (8). It was found that the previous failure of SCF methods in OFDFT was due to HOMO-LUMO orbital swapping. Using a modified SCF procedure, OE-SCF overcomes this issue and promises to combine the best of OFDFT and KSDFT, by retaining the accuracy of KSDFT and the computational efficiency of OFDFT.

3.2 TRIM

Another possible alternative approach to the direct minimization method used in this project could be the Trust-Region Image Method (TRIM) [16]. This method attempts to address the fundamental issues with the instability of OFDFT optimization algorithms. TRIM implements a minimax optimization by simultaneously maximization the chemical potential μ and minimizing the total energy functional. It is a second-order optimization scheme that involves the calculation of the Hessian matrix with respect to the variation in μ and the electron density. So far, no comparisons have been made between the non-linear conjugate gradient method and TRIM [9].

3.3 ML Methods

An even more recent alternative OFDFT minimization method is explored in [17]. In [17], a Lagrangian-free optimization framework is implemented that reformulates the OFDFT minimization problem into a machine learning problem. Normalizing flows are utilized to find the ground-state, which circumvents the need to impose the normalization and non-negativity constraints on the density, as well as likely avoiding instability issues when simulating vacuum gridpoints.

4 References

[1] G. S. Ho, V. L. Lignères and E. A. Carter, *Introducing PROFESS: A new program for orbital−free density functional theory calculations*, Comput. Phys. Commun. 179, 839−854 (2008).

[2] X. Shao, K. Jiang, W. Mi, A. Genova and M. Pavanello, *DFTpy: An efficient and object-oriented platform for orbital−free DFT simulations*, Wiley Interdiscip. Rev. Comput. Mol. Sci. 11, e1482 (2021).

[3] W. Mi, X. Shao, C. Su, Y. Zhou, S. Zhang, Q. Li, H. Wang, L. Zhang, M. Miao, Y. Wang and Y. Ma, *ATLAS: A real−space finite−difference implementation of orbital−free density functional theory*, Comput. Phys. Commun. 200, 87−95 (2016).

[4] P. Golub and S. Manzhos, *CONUNDrum: A program for orbital−free density functional theory calculations*, Comput. Phys. Commun. 256, 107365 (2020).

[5] D. I. Mihaylov, S. X. Hu and V. V. Karasiev *Dragon: A multi−GPU orbital−free density functional theory molecular dynamics simulation package for modeling of warm dense matter*, Commun. Comput. Phys. 294, 108931 (2024).

[6] L. Hung, C. Huang and E. A. Carter, *Preconditioners and Electron Density Optimization in Orbital−Free Density Functional Theory*, Commun. Comput. Phys. 12, 135 (2012).

[7] J. Nocedal and S. Wright, *Numerical Optimization*, (New York, Springer, 2006).

[8] S. C. Watson and E. A. Carter, *Linear−scaling parallel algorithms for the first principles treatment of metals*, Comput. Phys. Commun. 128, 67 (2000).

[9] W. Mi, K. Luo, S. B. Trickey and M. Pavanello, *Orbital−Free Density Functional Theory: An Attractive Electronic Structure Method for Large−Scale First−Principles Simulations*, Chem. Rev. 123, 12039−12104 (2023).

[10] G. K.-L. Chan, A. J. Cohen and N. C. Handy, *Thomas−Fermi−Dirac−von Weizsäcker models in finite systems*, J. Chem. Phys. 114, 631−638 (2001).

[11] X. Shao, W. Mi and M. Pavanello, *Efficient DFT Solver for Nanoscale Simulations and Beyond*, J. Phys. Chem. Lett. 12, 4134−4139 (2021).

[12] H. Jiang and W. Yang, *Conjugate−Gradient Optimization Method for Orbital−free Density Functional Calculations*, J. Chem. Phys. 121, 2030−2036 (2004).

[13] E. Polak, *Computational Methods in Optimization*, (Academic Press, New York, 1971).

[14] C. J. García-Cervera, *An Efficient Real Space Method for Orbital−Free Density−Functional Theory*, Commun. Comput. Phys. 2, 334 (2007).

[15] L. Hung, *Orbital−free density functional theory at the nanoscale and beyond: Algorithms and applications*, (Princeton University, United States, New Jersey, 2011).

[16] M. S. Ryley, M. Withnall, T. J. P. Irons, T. Helgaker, and A. M. Teale, *Robust All−Electron Optimization in Orbital−Free Density−Functional Theory Using the Trust−Region Image Method*, J. Phys. Chem. A , 125, 459−475 (2021).

[17] A. de Camargo, R. T. Q. Chen and R. A. Vargas-Hernández, *Leveraging Normalizing Flows for Orbital−Free Density Functional Theory*, Mach. Learn. Sci. Technol. 5 035061 (2024).