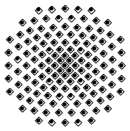
Finite Element Implementation of Orbital-Free Density Functional Theory for Electronic Structure Calculations

Soumya Swayamjyoti

Institute of Applied Mechanics (CE) · Micromechanics of Materials Group
University of Stuttgart, 70550 Stuttgart, Pfaffenwaldring 7, Germany



Contents

Master Thesis supervised by Dipl.-Phys. V. Schauer & Jun-Prof. Dr. C. Linder
Stuttgart, April 2012

1. Acknowledgements

First of all, I would like to thank Prof. Christian Linder for giving me the opportunity to do my master thesis in his research group. I thank him for his encouragement and support during this interesting study. Sincere thanks to Prof. Linder for his support during my PhD applications.

I want to thank Dipl.-Phys Volker Schauer for his guidance and supervision of this thesis. It was a great learning experience to work with him. The discussions we had pertaining to this thesis and about physics in general motivated my interest in computational physics to a great extent. I seize this opportunity to wish him the best in his future endeavors.

Many thanks to Prof. Christian Miehe for institutional support and the Institute of Applied Mechanics, at University of Stuttgart.

Thanks to our collaborators Prof. Vikram Gavini and his group (in particular, Mr. Phani Sudheer Motamarri) for useful inputs.

Thanks to all my colleagues in the Micromechanics of Materials Group and the Institute of Applied Mechanics (Chair I) of University of Stuttgart for the cordial work atmosphere.

Special mention to Mr. Gautam Ethiraj (Stuttgart) and Ms. Lelia Zielonka (Barcelona) for their help with administrative affairs during my studies in the Erasmus Mundus Master of Science in Computational Mechanics. I thank my teachers at University of Stuttgart and UPC Barcelona for helping me build up my knowledge base in computational mechanics. Financial support of the European Commission is acknowledged. My master studies in Europe were funded by the Erasmus Mundus Scholarship. I thank the selection consortium comprising CIMNE (International Center for Numerical Methods in Engineering, Barcelona), University of Stuttgart, Ecole Centrale Nantes and Swansea University.

- -

Year 2022 and 2023: I acknowledge the contribution of Shashwatha Mitra G B, a student from my alma- mater, National Institute of Technology Karnataka (NIT-K), for helping me with the formatting of the final document. Students who have worked with me, and who have taken interest in this field, seeing them excel in their scientific endeavors, would be the outcome intended by this thesis.

2. Introduction

At the fundamental level, ground state properties of materials depend only on electron density and hence are obtainable from first principles. However, a complete ab-initio calculation by solving the Schrodinger equation is computationally ambitious for a wide range of length scales involved in material modeling and thus separation of scales become necessary. At the microscopic scale, one can aim at ab-initio electronic structure calculations in the framework of approximate theories developed over the last decades, the most prominent being the density functional theory which was pioneered by [1, 2]. In the Kohn-Sham approach (KS-DFT) the kinetic energy is computed by solving an eigenvalue problem involving the wave-functions or orbitals of a material system. The computational complexity associated with this calculation is high and thus limits the potential system size to a few hundred atoms. This has inspired interest in orbital-free kinetic energy functionals. This form of density functional theory where the kinetic energy is modeled by orbital-free kinetic energy functionals is referred as Orbital-Free Density Functional Theory (OF-DFT). This theory has its roots in the Thomas-Fermi model which finds mention in the early papers of [3, 4, 5]

In the present work, we have implemented and developed numerical methods for Orbital-Free Density Functional Theory. In Chapter 3, we have set up the OF-DFT framework. Subsequently, we present the methodologies which we have developed and implemented in Chapter 4. The numerical implementation and algorithmical controls are detailed in Chapter 5. We have come up with very good and reasonably accurate numerical results of our implementation in Chapter 6. Chapter 7 concludes our work giving future directions to this research.

2.1. Relevance of Orbital-Free DFT in Multiscale Modeling of Materials

In the context of material science, the electronic structure of a material determines the behaviour of the material at the macroscopic level, which is the length scale of interest for engineering applications. Processes that occur at a certain length scale govern the behaviour of the material at larger length scales. Hence, it becomes necessary to successfully bridge scales that can couple quantum mechanical, molecular mechanical (classical atomistic) and continuum mechanics simulations in a unified theory. A feature of these multiscale simulations is to use accurate and computationally complex techniques to treat regions of small length scale and "coarse-grain" (use of less accurate but also less expensive methods) the rest of the system. An important aspect of the multiscale treatment is to have matching boundary conditions at the interface of two simulation methods, a virtual boundary that arises due to the coupling technique. This boundary must not introduce any physical consequences in order to correctly simulate the actual system. While coupling a KS-DFT based quantum mechanical simulation method with a classical method, boundary conditions on the electron wave functions must be imposed at the interface of the two simulation methods, which is complicated. A straight-forward and simpler approach is to use OF-DFT as an input which only requires boundary conditions to be imposed on the electron density. An efficient coupling method for simple metals based on OF-DFT has been implemented in [6]. A quasi-continuum scheme based on OF-DFT (QC-OFDFT) has been developed by [7] which concurrently bridges atomistic and continuum length scales. [7] have used finite-element bases in their quasi-continuum scheme which has the added advantage that there are no restrictions on boundary con-

ditions, unlike plane-wave bases which represents problems only with periodic boundary conditions. DFT based multiscale (QC) material modeling approaches have also been discussed in [8, 9, 10]. It's notable that [9] use OF-DFT as their sole input for calculation of the energetics. Figure 1 represents a multiscale approach.

Figure 1: Multiscale Modelling of Materials

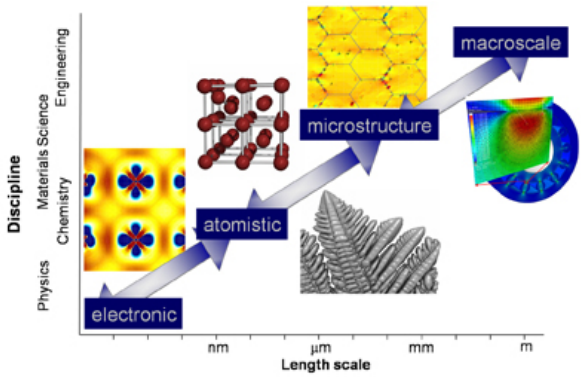


Figure from http://www.icams.de/content/research-at-icams/research-index.html

3. Orbital-Free Density Functional Theory

In this chapter, the OF-DFT framework is set up. The OF-DFT energy functional is described and different kinetic energy terms of the functional are discussed. In addition to that, the relation with respect to Kohn-Sham DFT is presented. Different solution procedures and basis functions used by other researchers solving OF-DFT are detailed. Furthermore, the development of FEM based OF-DFT implementations in the last decade is taken note of. Subsequently, the variational problem is presented followed by the discrete formulation of the problem in finite element spaces.

3.1. The Orbital-Free Density Functional Theory Framework

The Orbital-Free Density Functional Theory is used to study atomic systems. An atomic system consists of nuclei and surrounding electrons. A neutral atomic system consists of equal amount of positive charge in the nuclei and negative charge as electrons. The OF-DFT uses electron density as an input which is the number of electrons per unit volume. The negative charge in the system is thus represented by a smeared and smooth eletron density which has higher concentration around a nucleus or an atomic core. The electron density is denoted as ρ which has the properties $\rho \geq 0$ and $\int \rho \, \mathrm{d}\mathbf{r} = N$ where N is the total number of electrons. In this study, we use atomic units for our study of atomic systems. In the atomic units, which is the conventional units used in atomic physics, the four physical constants, namely the electron mass, the elementary charge, the Planck's constant and the Coulomb's constant are defined as unity. The unit of energy we use is Hartree, which is the atomic unit of energy.

The ground state energy functional in density functional theory is given by; [11]

$$E(\rho, \mathbf{R}) = T_s(\rho) + E_{xc}(\rho) + E_H(\rho) + E_{ext}(\rho, \mathbf{R}) + E_{zz}(\mathbf{R}) \tag{1}$$

where ρ is the electron-density and $\mathbf{R}$ is the vector containing the nuclear positions in the system.

T_s is the kinetic energy of non-interacting electrons. In orbital-free density functional theory, T_s is modeled by orbital-free kinetic energy functionals. The first direct approximations of the kinetic energy dates back to the Thomas-Fermi model; [4]. In the Thomas-Fermi model, the kinetic energy is expressed as;

$$T_s(\rho) = C_F \int_\Omega \rho^{5/3}(\mathbf{r}) \, \mathrm{d}\mathbf{r} \tag{2}$$

where $C_F = \dfrac{3}{10}(3\pi^2)^{2/3}$

It is usual practice to write the energy functional for interacting electrons as a sum of the kinetic energy and the interaction terms comprising of the electrostatic energy and the exchange-correlation energy where the exchange-correlation term is a quantum correction term. As the true form of the energy functional is unknown, approximations for $E_{xc}(\rho)$ must be used. The most widely used approximation is the local-density approximation (LDA), where the functional depends locally only on the electron density.

The Local Density Approximation (LDA) [12] used in the present work, is written in the form by

$$E_{xc}(\rho) = \int_\Omega \epsilon_{xc}(\rho(\mathbf{r}))\rho(\mathbf{r}) \, \mathrm{d}\mathbf{r} \tag{3}$$

where $\epsilon_{xc} = \epsilon_x + \epsilon_c$ is the exchange and correlation energy per electron given by

$$\epsilon_x(\rho) = -\frac{3}{4}(\frac{3}{\pi})^{1/3}\rho^{1/3} \tag{4}$$

$$\epsilon_c(\rho) = \begin{cases} \dfrac{\gamma}{1+\beta_1\sqrt{r_s}+\beta_2 r_s} & \text{if } r_s \geq 1 \\ A\log r_s + B + Cr_s\log r_s & \text{if } r_s < 1 \end{cases} \tag{5}$$

where $r_s = \left(\dfrac{3}{4\pi\rho}\right)^{(1/3)}$. For unpolarized medium, the values of the constants are

$\gamma = -0.1471, \beta_1 = 1.1581, \beta_2 = 0.3446, A = 0.0311, B = -0.048, C = 0.0014, D = -0.0108$

The last three terms in the energy functional are electrostatic:

$$E_H(\rho) = \underbrace{\frac{1}{2}\int_\Omega\int_\Omega \frac{\rho(\mathbf{r})\rho(\mathbf{r'})}{|\mathbf{r}-\mathbf{r'}|}\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r'}}_{electron-electron} \tag{6}$$

$$E_{ext}(\rho,\mathbf{R}) = \underbrace{\int_\Omega \rho(\mathbf{r})V_{ext}(\mathbf{r})\mathrm{d}\mathbf{r}}_{electron-nuclei} \quad , \quad V_{ext} = \int_\Omega \frac{b(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,\mathrm{d}\mathbf{r}' \tag{7}$$

$$E_{zz}(\mathbf{R}) = \underbrace{\frac{1}{2}\sum_{I,J,I\neq J}\frac{Z_I Z_J}{|R_I - R_J|}}_{nuclei-nuclei} \tag{8}$$

The point charge Z_I is regularized with a smooth function $Z_I\delta(\mathbf{r})$.

$$E_{zz}(\mathbf{R}) = \frac{1}{2}\int_\Omega\int_\Omega \frac{b(\mathbf{r})b(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,\mathrm{d}\mathbf{r}\,\mathrm{d}\mathbf{r}' \tag{9}$$

where $b(\mathbf{r}) = \sum_{I=1}^{M} Z_I\delta(\mathbf{r})$

This differs from the earlier formulation by the self-energy of the nuclei. As the self-energy of the nuclei is a constant, it does not affect the subsequent derivations.

E_H is the Hartree energy, which is the interaction energy of electron-density. E_{ext} is the interaction energy with external field induced by nuclear charges. E_{zz} is the repulsive energy between nuclei.

The Thomas-Fermi model has several shortcomings; [13]. A major drawback of this model is that it does not predict atomic binding to form molecules and solids, as proven in the Teller non-binding theorem. In light of this, Weizsäcker introduced a correction to the kinetic energy functional and the derived model is known as Thomas-Fermi-Weizsäcker (TFW) model which has the form:

$$T_s(\rho) = C_F \int_\Omega \rho^{5/3}(\mathbf{r})\ \mathrm{d}\mathbf{r} + \frac{\lambda}{8}\int_\Omega \frac{|\nabla\rho(\mathbf{r})|^2}{\rho(\mathbf{r})}\mathrm{d}\mathbf{r} \tag{10}$$

where $C_F = \frac{3}{10}(3\pi^2)^{2/3}$ and λ is a parameter. The Weizsäcker correction facilitates prediction of atomic binding. $\lambda = 1$ is the Weizsäcker correction and is suitable for rapidly varying electron densities. $\lambda = 1/9$ is suitable for slowly varying electron densities. $\lambda = 1/6$ includes the fourth order effects and $\lambda = 0.186$ is determined from analysis of large atomic-number limit of atoms; [12]. Weizsäcker type corrections include different values of λ and the corresponding models are known as TFλW models. In the present work, $\lambda = 0.2$ is used unless otherwise stated. This value of λ is the most optimum value; [14]. A comparative study (TFDW and HF models) of variation of relative deviation of ground state energies with λ as a parameter has been presented in [14]. The relative deviation of the total ground state energies has been studied with λ as a parameter and at $\lambda = 0.2$, we find the least deviation. Hence we use $\lambda = 0.2$ for our computations. Unfortunately, the kinetic energy given by the TFW model is not significantly better than the TF model [13], the reason being the highly non-local nature of the exact kinetic energy functional. There have been several attempts to accurately model the kinetic energy functional as studied in [15, 16, 17, 18, 19, 20, 21]. The new kinetic energy kernels developed in the aforementioned papers are non-local in nature and hence model the kinetic energy functional more accurately. A prominent non-local kinetic energy functional is shown below, taken from [22]:

$$T_s(\rho) = C_F \int_\Omega \rho^{5/3}(\mathbf{r})\ \mathrm{d}\mathbf{r} + \frac{\lambda}{8}\int_\Omega \frac{|\nabla\rho(\mathbf{r})|^2}{\rho(\mathbf{r})}\mathrm{d}\mathbf{r} + T_k(\rho) \tag{11}$$

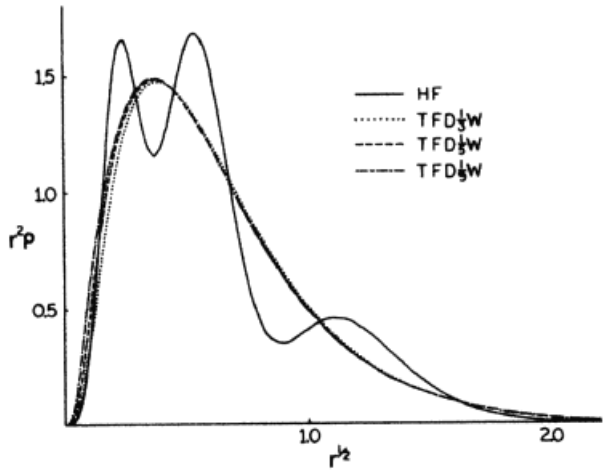
$$T_k(\rho) = \int_\Omega\int_\Omega \rho^\alpha(\mathbf{r})K(|\mathbf{r}-\mathbf{r'}|,\rho(\mathbf{r}),\rho(\mathbf{r'}))\rho^\beta(\mathbf{r'})\ \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r'} \tag{12}$$

where α and β are parameters. The kernels can be defined as Density Dependant (DD) or Density Independent depending on whether K is dependant on $\rho(\mathbf{r})$ or not, respectively [22].

But the disadvantage of the newly developed kinetic energy kernels is that it's computationally expensive to evaluate the non-local terms. This computational expense balances out the computational simplicity inherent in the OF-DFT models when compared with KS-DFT. To see the latest developments in this field interested readers can refer [23, 24, 25, 26].

The energy components of Orbital-Free DFT are different to those of Orbital-Based DFT, but sum up to a total energy that is same. This is due to the fact that OF-DFT computations lead to a difference in the electron density, which does not show the characteristic shell structure as shown in figure . It is this correctness of total energies together with its numerical advantages, which makes the Orbital-Free DFT also interesting

Figure 2: Electron Density of Argon for Various Models; [11]



for coupling to macroscopic scales, as was already shown in the quasicontinuum framework [7]. The accuracies required in chemistry, e.g. for the evaluation of reaction paths, cannot be met by the orbital-free DFT, however qualitative insights into the mechanical behavior of materials are still possible. The absence of the orbitals makes the solution process simpler and faster.

To discuss the properties of the Thomas-Fermi energy functional, we refer to [4]. The uniqueness of the solutions of the Thomas-Fermi equations has been established in this paper. These solutions also minimize the Thomas-Fermi energy functional. The Thomas-Fermi functional is strictly convex which plays an important role in the uniqueness of the solutions. The Thomas-Fermi functional is non-linear which poses the biggest challenge in numerical solvability of the same.

3.2. The Variational Problem

The electrostatic interaction energy of the electrons and repulsive energy of the nuclei are non-local terms. The evaluation of these terms is the most computationally intensive part of the energy functional. Hence, the problem is reformulated into a local variational problem; [12]; as follows:

$$E_H(\rho) + E_{ext}(\rho, \mathbf{R}) + E_{zz}(\mathbf{R}) = - \inf_{\phi \in H^1(\mathbb{R}^3)} \left\{ \frac{1}{8\pi} \int_{\mathbb{R}^3} |\nabla \phi|^2 \mathrm{d}\mathbf{r} \; - \; \int_{\mathbb{R}^3} (\rho(\mathbf{r}) + b(\mathbf{r}))\, \phi(\mathbf{r})\, \mathrm{d}\mathbf{r} \right\} \tag{13}$$

where ϕ is the electrostatic potential and ρ is the electron density.

Now, we introduce Euler-Lagrange equations which are differential equations whose solutions are functions for which the functional is stationary. Euler-Lagrange equation corresponding to the variational problem above is,

$$\frac{-1}{4\pi} \Delta \phi = \rho + b \tag{14}$$

These have an unique solution:

$$\phi(\mathbf{r}) = \int_\Omega \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,\mathrm{d}\mathbf{r}' + \int_\Omega \frac{b(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,\mathrm{d}\mathbf{r}' = \int_\Omega \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,\mathrm{d}\mathbf{r}' + V_{ext} \tag{15}$$

where

$$V_{ext} = \int_\Omega \frac{b(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,\mathrm{d}\mathbf{r}' \tag{16}$$

Substituting the above ansatz into the variational problem gives us back the euler-lagrange equation.

The energy functional in the local form is then transformed to,

$$E(\rho, \mathbf{R}) = \sup_{\phi \in H^1(\mathbb{R}^3)} L(\rho, \mathbf{R}, \phi) \tag{17}$$

The complete expression then reads as a saddle-point problem which is thus given by,

$$\begin{aligned} E_0(\rho, \mathbf{R}) &= \inf_{\rho \in Y,\, R \in \mathbb{R}^3} \sup_{\phi \in H^1(\mathbb{R}^3)} L \\ \text{subject to} \quad \rho(\mathbf{r}) \geq 0 \quad &\text{and} \quad \int \rho(\mathbf{r})\ \mathrm{d}\mathbf{r} = N \end{aligned} \tag{18}$$

where the Lagrangian is given by:

$$\begin{aligned} L(\rho, \mathbf{R}, \phi) \;:=\; & C_F \int_\Omega \rho^{5/3}(\mathbf{r})\ \mathrm{d}\mathbf{r} + \frac{\lambda}{8} \int_\Omega \frac{|\nabla \rho(\mathbf{r})|^2}{\rho(\mathbf{r})} \mathrm{d}\mathbf{r} + \int_\Omega \epsilon_{xc}(\rho(\mathbf{r}))\rho(\mathbf{r})\,\mathrm{d}\mathbf{r} \\ - \;& \frac{1}{8\pi} \int_{(\mathbb{R})^3} |\nabla \phi(\mathbf{r})|^2 \mathrm{d}\mathbf{r} \;+\; \int_{\mathbb{R}^3} (\rho(\mathbf{r}) + b(\mathbf{r}))\,\phi(\mathbf{r})\ \mathrm{d}\mathbf{r} \end{aligned} \tag{19}$$

The constraint $\rho(\mathbf{r}) \geq 0$ can be imposed by substituting,

$$\rho = u^2 \tag{20}$$

The problem is re-stated as,

$$\inf_{u^2 \in H_0^{-1}, R \in \mathbb{R}^{3M}} \sup_{\phi \in H^1(\mathbb{R}^3)} L(u, \mathbf{R}, \phi) \tag{21}$$

subject to

$$\int_\Omega u^2(\mathbf{r})\,\mathrm{d}\mathbf{r} = N \tag{22}$$

Note that the above problem is equivalent to solving the TFW equations along with the subsidiary condition of $\int_\Omega u^2(\mathbf{r})\,\mathrm{d}\mathbf{r} = N$

3.3. Finite Element Formulation

There have been several implementations of the OF-DFT in plane-wave basis and spline basis; [14]. But such implementations can only simulate systems with periodic boundary conditions. To predict defects in materials such as, vacancies, cracks, dislocations etc, plane wave basis sets are inadequate. In lieu of the same, in the present work a finite element implementation of OF-DFT has been performed similar to the methodologies developed in [12, 27, 22]. Finite-element formulation of OF-DFT has been successful in studying defects in materials as detailed in; [28, 10, 27, 29, 30]. Additionally, finite element are local bases unlike plane wave bases. With finite elements the domain can be adaptively refined which is an added advantage.

The electron density and the electrostatic potential are discretized using a finite-element scheme using 4-noded tetrahedral elements with linear shapefunctions, as follows,

$$u = \sum u_i N_i(x) \tag{23}$$

$$\phi = \sum \phi_i N_i(x) \tag{24}$$

Plugging in the above discretizations into the lagrangian, we obtain,

$$\begin{aligned} L(u, \mathbf{R}, \phi) \;:=\; & C_F \sum_{e=1}^{M} \int_{\Omega^e} \Big(\sum_{i=1}^{4} u_i N_i(x) \Big)^{10/3} \mathrm{d}\mathbf{r} \\ + \; & \frac{\lambda}{2} \sum_{e=1}^{M} \int_{\Omega^e} \Big| \sum_{i=1}^{4} u_i \nabla N_i(x) \Big|^2 \mathrm{d}\mathbf{r} \\ + \; & \sum_{e=1}^{M} \int_{\Omega^e} \epsilon_{xc} \Big(\big(\sum_{i=1}^{4} u_i N_i(x) \big)^2 \Big) \Big(\sum_{i=1}^{4} u_i N_i(x) \Big)^2 \mathrm{d}\mathbf{r} \\ - \; & \frac{1}{8\pi} \sum_{e=1}^{M} \int_{\Omega^e} \Big| \sum_{i=1}^{4} \phi_i \nabla N_i(x) \Big|^2 \mathrm{d}\mathbf{r} \\ + \; & \sum_{e=1}^{M} \int_{\Omega^e} \Big(\big(\sum_{i=1}^{4} u_i N_i(x) \big)^2 + b(\mathbf{r}) \Big) \cdot \Big(\sum_{i=1}^{4} \phi_i N_i(x) \Big) \mathrm{d}\mathbf{r} \end{aligned} \tag{25}$$

where M in the above expression is the number of elements.

Thus L is a multi-dimensional function of the nodal-coefficients.

$$L = L(u_1, u_2, u_3, ..., u_N, \phi_1, \phi_2, \phi_3, ..., \phi_N) = L(\{u_i\}, \{\phi_i\}) \tag{26}$$

where N in the above expression is the number of Degrees of Freedom (DOFs).

Multi-dimensional optimization algorithms are required to solve the above posed discrete problem in finite element spaces. This is discussed in the next chapter.

4. Optimization Methods for Orbital-Free Density Functional Theory

The discrete problem posed in earlier chapters, can be solved by carrying out a multi-dimensional minimization of the energy functional. Several minimization algorithms can be employed to this end, as detailed in [24]. A direct minimization algorithm to find the energy minimum is by traversing the direction opposite to the gradient direction, which is otherwise known as the steepest-descent method. In the present study though, we will use a conjugate gradient (CG) method which has significant advantages over the steepest descent method; [24, 31, 32].

4.1. Steepest Descent Method

The steepest descent method is a minimum finding method where the traversal direction is the negative of the gradient direction. The gradient direction represents the direction of maximum increase of the function and hence we move in the negative gradient direction. The iterative scheme is given by

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{r}^{(k)} \tag{27}$$

where $\alpha^{(k)} = -\frac{\langle \mathbf{r}^{(k)}, \mathbf{r}^{(k)} \rangle}{\langle \mathbf{r}^{(k)}, \mathbf{A}\mathbf{r}^{(k)} \rangle}$, which is obtained by solution of a 1-D minimization problem. $\langle \cdot \rangle$ represents an inner product and this notation for inner product is used throughout this study.

Figure 3: Iterations of Steepest Descent Method for a Quadratic Problem; Figure from CIMNE Virtual Center

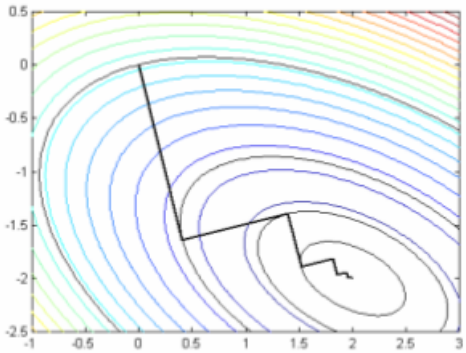


Figure 3 shows the iterations of the steepest descent method for a quadratic problem. In the steepest descent method, the advance directions are often repeated and it takes into account only the neighbourhood behavior of the function. In contrast, in the conjugate gradient method each direction is constructed taking into account previous directions which is why the CG method is more efficient than the steepest descent method. Therefore, we use the conjugate gradient method in our implementation.

4.2. Conjugate Gradient Method

The conjugate gradient Method is a method to find the minimum of a function whose gradient can be computed. In this method, the advance direction is always the conjugated negative gradient direction. A key feature of this method is that the advance directions

are never repeated. Two vectors $\mathbf{u}$ and $\mathbf{v}$ are said to be A-conjugate of each other if $\mathbf{u}^T\mathbf{A}\mathbf{v} = 0$ where $\mathbf{A}$ is a symmetric positive definite matrix. In the CG method the advance directions are chosen to be A-conjugate and are given by,

$$\mathbf{p}^{(k)} = \begin{cases} \mathbf{r}^{(0)} & k = 0 \\ \mathbf{r}^{(k)} + \beta^{(k)}\mathbf{p}^{(k-1)} & k > 0 \end{cases} \tag{28}$$

where $\beta^{(k)} = -\frac{\langle \mathbf{r}^{(k)}, \mathbf{A}\mathbf{p}^{(k-1)} \rangle}{\langle \mathbf{p}^{(k-1)}, \mathbf{A}\mathbf{p}^{(k-1)} \rangle}$. Thus we see that, the new direction in the conjugate gradient method is constructed from a linear combination of the new residual and the previous direction. In the above equations, $\mathbf{r}$ is the residual or the gradient, $\mathbf{p}$ is the direction vector and β is a scalar.

Figure 4: Iterations of Conjugate Gradient Method for a Quadratic Problem; Figure from CIMNE Virtual Center

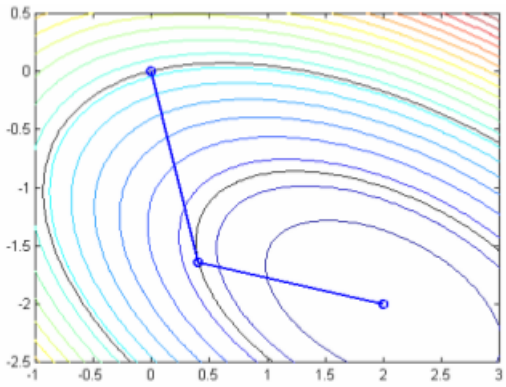


Figure 4 represents the iteration of a conjugate gradient Method applied to a quadratic landscape. For a quadratic problem, if n is the system dimension, the CG method converges in n iterations at the most. But usually, this method converges much before n iterations. The convergence of the conjugate gradient method often depends on the condition number of the system matrix which is given by $k = \frac{\lambda_{max}}{\lambda_{min}}$ where λ_{max} and λ_{min} are the maximum and minimum eigenvalues of the system matrix. The use of a preconditioner enables us to reduce the condition number which leads to faster convergence. A preconditioned conjugate gradient method is given as follows; [32]:

$$\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{r}^{(0)}, \tag{29}$$

$$\boldsymbol{d}^{(0)} = \boldsymbol{M}^{-1}\mathbf{r}^{(0)}, \tag{30}$$

$$\alpha^{(k)} = -\frac{\langle \mathbf{r}^{(i)}, \boldsymbol{M}^{-1}\mathbf{r}^{(i)} \rangle}{\langle \boldsymbol{d}^{(i)}, \mathbf{A}\mathbf{p}^{(i)} \rangle}, \tag{31}$$

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \alpha^{(i)}\boldsymbol{d}^{(i)}, \tag{32}$$

$$\mathbf{r}^{(i+1)} = \mathbf{r}^{(i)} - \alpha^{(i)}\mathbf{A}\boldsymbol{d}^{(i)}, \tag{33}$$

$$\beta^{(i+1)} = -\frac{\langle \mathbf{r}^{(i+1)}, \boldsymbol{M}^{-1}\mathbf{r}^{(i+1)}\rangle}{\langle \mathbf{r}^{(i+1)}, \boldsymbol{M}^{-1}\mathbf{r}^{(i+1}\rangle}, \tag{34}$$

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{M}^{-1}\mathbf{r}^{(i+1)} + \beta^{(i+1)}\boldsymbol{d}^{(i)}. \tag{35}$$

where $\boldsymbol{M}$ is the preconditioner matrix, $\mathbf{r}$ is the residual vector, $\boldsymbol{d}$ is the preconditioned residual, α and β are scalars.

For application of the conjugate gradient method to our formulation, the problem of solving the saddle-point formulation in finite element spaces is broken down into two sub-problems, that is, the Poisson Problem and the Thomas-Fermi Problem. The energy functional is a multi-dimensional function of the nodal-coefficients $\{u_i\}$ and $\{\phi_i\}$. We adopt a solution procedure wherein the Poisson Problem and the Thomas-Fermi Problem are solved iteratively till self-consistency is achieved. The goal of the Poisson Problem is to maximize the energy functional with respect to the $\{\phi_i\}$ and the goal of the Thomas-Fermi Problem is to minimize the energy functional with respect to the $\{u_i\}$. This is the way to deal with the inf-sup OF-DFT problem. To this end, we apply the conjugate-gradient method [32] to optimize each of the two sub-problems. The problem thus reduced to the following form;

$$\begin{aligned} L(u,\mathbf{R},\phi) \;:=\; & C_F\int_\Omega u^{10/3}(\mathbf{r})\,\mathrm{d}\mathbf{r} + \frac{\lambda_{kin}}{2}\int_\Omega |\nabla u(\mathbf{r})|^2\mathrm{d}\mathbf{r} + \int_\Omega \epsilon_{xc}(u^2(\mathbf{r}))u^2(\mathbf{r})\,\mathrm{d}\mathbf{r} \\ - \;& \frac{1}{8\pi}\int_{\mathbb{R}^3} |\nabla\phi(\mathbf{r})|^2\mathrm{d}\mathbf{r} \;+\; \int_{\mathbb{R}^3}\left(u^2(\mathbf{r}) + b(\mathbf{r})\right)\phi(\mathbf{r})\,\mathrm{d}\mathbf{r} + \underbrace{\mu\left(\int_\Omega u^2(\mathbf{r})\mathrm{d}\mathbf{r} - N\right)^2}_{penalty-term} \end{aligned} \tag{36}$$

such that: $\nabla_\phi L = 0$, $\nabla_u L = 0$

In the above expression, μ is the penalty parameter to treat the constraint $(\int_\Omega u^2(\mathbf{r})\mathrm{d}\mathbf{r} - N) = 0$. This parameter is adaptively handled in our implementation. We start with a small value of penalty (of the order of 10) and increase the penalty parameter by a factor after each self-consistency loop. This factor is arbitrarily chosen such that the we achieve fast convergence. If the penalty parameter is too high at the start, the solution can either diverge or lead to very slow convergence. The stricter treatment of the constraint with high values of the penalty parameter towards the end makes sure that the constraint is accurately satisfied.

An iterative minimization procedure is defined by the sequence $u_i^{(n+1)} = u_i^{(n)} + \lambda^{(n)}h'^{(n)}$ where h' is normalized h, i.e $h' = \frac{h}{||h||}$. The residual $\mathbf{r} = -\nabla_{\mathbf{u}}L$ is substituted into the CG routine. For the scalar $\beta^{(k)}$ in equation X, several approximations have been researched. Two prominent substitutions for $\beta^{(k)}$ are the Fletcher-Reeves formula $\beta^{(k)} = \frac{\mathbf{r}^{(k+1)}.\mathbf{r}^{(k+1)}}{\mathbf{r}^{(k)}.\mathbf{r}^{(k)}}$ and the Polak-Ribiere formula $\beta^{(k)} = \frac{\mathbf{r}^{(k+1)}.(\mathbf{r}^{(k+1)}-\mathbf{r}^{(k)})}{\mathbf{r}^{(k)}.\mathbf{r}^{(k)}}$. In our current implementation, we use the Polak-Ribiere formula for efficiency purposes.

The above steps are followed to minimize L with respect to the coefficients $\{u_i\}$. A similar CG method is also adopted to maximize L with respect to $\{\phi_i\}$.

4.2.1. Poisson Problem. As formulated earlier, the Lagrangian can be expressed as a multi-dimensional function of $\{u_i\}$ and $\{\phi_i\}$:

$$L = L(u_1, u_2, u_3, ..., u_N, \phi_1, \phi_2, \phi_3, ..., \phi_N) = L(\{u_i\}, \{\phi_i\}) \tag{37}$$

The Poisson Problem is equivalent to solving the supremum part of the inf-sup saddle-point OF-DFT problem, that is maximizing the Lagrangian with respect to $\{\phi_i\}$ while keeping the $\{u_i\}$ fixed. This involves computing the partial derivatives of the Lagrangian with respect to $\{\phi_i\}$ as below and using this gradient in the conjugate gradient method.

$$\left.\frac{\partial L}{\partial \phi_i}\right|_{u_i=const.} = \frac{-1}{4\pi}\int_{\mathbb{R}^3} \nabla\phi(\mathbf{r})\nabla N_i(x)\,\mathrm{d}\mathbf{r} + \int_{\mathbb{R}^3}(u^2(\mathbf{r}) + b(\mathbf{r}))N_i(x)\,\mathrm{d}\mathbf{r} \tag{38}$$

4.2.2. Thomas-Fermi Problem. The Thomas-Fermi problem involves solving the infinimum part of the inf-sup saddle-point OF-DFT formulation, that is, minimizing the Lagrangian with respect to $\{u_i\}$ while keeping the $\{\phi_i\}$ fixed. The partial derivative of the Lagrangian with respect to the $\{u_i\}$ is computed as below and is used in the conjugate gradient algorithm for minimizing the Lagrangian.

$$\left.\frac{\partial L}{\partial u_i}\right|_{\phi_i=const.} = \frac{\partial T_s}{\partial u_i} + \frac{\partial E_{xc}}{\partial u_i} + \frac{\partial (ES)}{\partial u_i} + \frac{\partial (Penalty)}{\partial u_i} \tag{39}$$

where T_s is the kinetic energy functional, E_{xc} is the exchange-correlation functional, ES is the electrostatic part and $Penalty$ is the penalty term.

Since the Thomas-Fermi problem is a non-linear problem, during it's conjugate gradient minimization, a line-search technique is introduced during the CG iterations. For every new direction, we take a step in the descent direction. But, we do not know, where is the minimum in that small step. This would not have been the case had the problem is quadratic. So, unlike the Poisson Problem, in the Thomas-Fermi problem, we need to find the minimum on that line for every new CG direction. The goal here is to obtain the μ, as computed below:

L can be expressed as a function of μ as L is minimized by a one-dimensional line search; [32].

$$f(\mu) = L(u_i^{(n)} + \mu^{(n)} h^{(n)}) \tag{40}$$

For L to be minimum, $f'(\mu) = 0$. To find the root of the equation $f'(\mu) = 0$, secant method is used, as below:

$$\mu^{(k+1)} = \mu^{(k)} - \frac{f'(\mu^{(k)})}{f''(\mu^{(k)})} \tag{41}$$

The second order derivative is approximated by,

$$f''(\mu^{(k)}) = \frac{f'(\mu^{(k)}) - f'(\mu^{(k-1)})}{\mu^{(k)} - \mu^{(k-1)}} \tag{42}$$

Substituting, we get,

$$\mu^{(k+1)} = \mu^{(k)} - \frac{(\mu^{(k)} - \mu^{(k-1)})f'(\mu^{(k)})}{f'(\mu^{(k)}) - f'(\mu^{(k-1)})} \tag{43}$$

where $f'(\mu) = (\boldsymbol{h}^{(n)})^T \nabla_{\mathbf{u}} L(u_i^{(n)} + \mu^{(k)} \boldsymbol{h}^{(n)})$. The line search is stopped when $abs(\mu^{(k)} - \mu^{(k-1)}) < tol$. μ for a particular line is thus determined as, $\mu^{(n)} = \mu^{(k)}$

The above line search is embedded into the conjugate gradient method presented in the previous section.

4.3. Root-Finding Newton Method

The Newton method is used to find the stationary point of a function. It assumes a quadratic curvature in the vicinity of the optimum which is why it is also important to start the iterations close enough to the real solution. A Taylor series approximation around the iterate neglecting higher order terms is used in this method, as shown below:

$$\mathbf{f}(\mathbf{x_k} + \mathbf{\Delta x}) = \mathbf{f}(\mathbf{x_k}) + \nabla \mathbf{f}(\mathbf{x_k})^{\mathbf{T}} \mathbf{\Delta x} + \frac{\mathbf{1}}{\mathbf{2}} \mathbf{\Delta x^T B \Delta x} \tag{44}$$

where $\mathbf{B}$ is the Hessian matrix of the second derivatives of $\mathbf{f}$ and $\nabla \mathbf{f}(\mathbf{x_k})$ is the gradient of $\mathbf{f}$ at the iterate.

$$\nabla \mathbf{f}(\mathbf{x_k} + \mathbf{\Delta x}) = \nabla \mathbf{f}(\mathbf{x_k}) + \mathbf{B \Delta x} \tag{45}$$

Setting the gradient to zero, we get the Newton step:

$$\mathbf{\Delta x} = -\mathbf{B^{-1}} \nabla \mathbf{f}(\mathbf{x_k}) \tag{46}$$

Figure 5: Quadratic Energy Landscapes

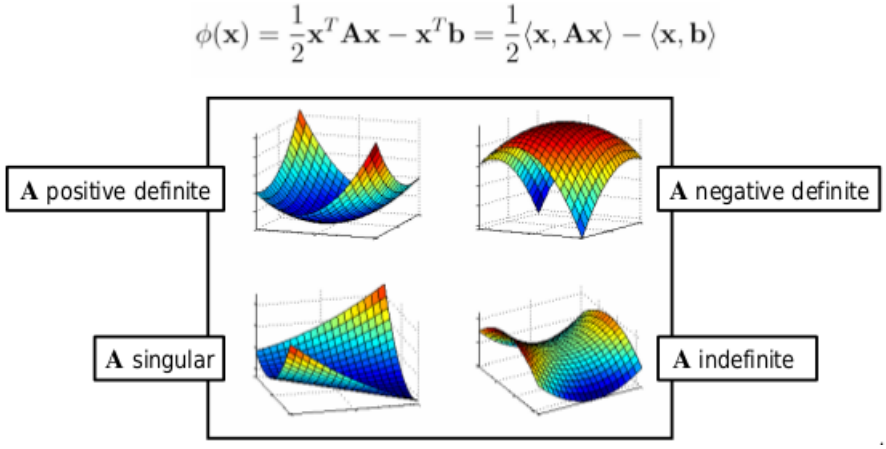


Figure 5 shows the quadratic approximations of multi-dimensional energy landscapes. The multi-dimensional root-finding Newton method is a method developed in the present work, which is identical to the staggered solution procedure adopted in [22]. In this method, we use the hessian of the lagrangian, that is the matrix of the second derivatives of the lagrangian with respect to the nodal coefficients, to locate the minimum of the functional. A lagrange multiplier is introduced in the energy functional to perform constrained optimization. If L has an extremum for the original constrained problem, then there exists λ which maps to a stationary point for the Lagrange function (stationary points are those points where the partial derivatives of L are zero).The functional is at an extremum point when the gradient is zero. What we need to do is to find the roots of

the equation that equates the gradient to zero. The problem formulation of this method is presented below,

$$\begin{aligned} L(u,\mathbf{R},\phi) \;&:=\; C_F\int_\Omega u^{10/3}(\mathbf{r})\,\mathrm{d}\mathbf{r} + \frac{\lambda_{kin}}{2}\int_\Omega |\nabla u(\mathbf{r})|^2\mathrm{d}\mathbf{r} + \int_\Omega \epsilon_{xc}(u^2(\mathbf{r}))u^2(\mathbf{r})\,\mathrm{d}\mathbf{r} \\ &-\; \frac{1}{8\pi}\int_{\mathbb{R}^3}|\nabla\phi(\mathbf{r})|^2\mathrm{d}\mathbf{r} \;+\; \int_{\mathbb{R}^3}(u^2(\mathbf{r})+b(\mathbf{r}))\,\phi(\mathbf{r})\,\mathrm{d}\mathbf{r} + \underbrace{\lambda(\int_\Omega u^2(\mathbf{r})\mathrm{d}\mathbf{r}-N)}_{lagrange-term} \end{aligned} \tag{47}$$

such that: $\nabla_{\{\phi_i\}}L = 0$, $\nabla_{\{u_i\}}L = 0$ and $\nabla_\lambda L = 0$ where λ in the above expression and subsequently in this chapter is the lagrange multiplier introduced to handle the constraint $(\int_\Omega u^2(\mathbf{r})\mathrm{d}\mathbf{r} - N) = 0$.

Thus, the energy functional is a multi-dimensional function of $\{u_i\}$, $\{\phi_i\}$ and λ as shown below,

$$L = L(u_1,u_2,u_3,...,u_N,\phi_1,\phi_2,\phi_3,...,\phi_N,\lambda) = L(\{u_i\},\{\phi_i\} \tag{48}$$

It's important to discuss the above treatment of the OF-DFT problem. First of all, what we have done is to approximate the non-linear Thomas-Fermi functional in a quaratic form. The Poisson Problem is already in the quadratic form, so that is straight-forward to solve. But, that is not the case with the Thomas-Fermi part. So, we expand this Thomas-Fermi part of the functional as a Taylor series expansion and then neglect the higher-order terms. This treatment is being dealt with in the subsequent sections.

An obvious question that arises here is, why did not we treat the constraint using a penalty term as we did in the conjugate gradient method presented earlier. The technical reason behind it is that, if we take the second derivative of the penalty term, we obtain a non-sparce term which makes the whole hessian non-sparce. Thus, it becomes computationally impractical for us to store and solve this non-sparce hessian. Hence, we have reformulated here the optimization problem to make use of a lagrange multiplier instead. The mathematical explanation is presented below,

$$Penalty := \mu(\int_\Omega u^2(\mathbf{r})\,\mathrm{d}\mathbf{r} - N)^2 \tag{49}$$

$$\frac{\partial(Penalty)}{\partial u_i \partial u_j} = 8\mu\Big(\int_\Omega uN_i(x)\,\mathrm{d}\mathbf{r}\Big)^+ \underbrace{4\mu\Big(\int_\Omega u^2(\mathbf{r})\,\mathrm{d}\mathbf{r} - N\Big)\int_\Omega N_i(x)N_j(x)\,\mathrm{d}\mathbf{r}}_{non-sparce} \tag{50}$$

4.3.1. Poisson Problem. The Poisson part is a quadratic problem, that is a quadratic function of the $\Phi = \{\phi_i\}$. The Lagrangian is expressed as a Taylor series expansion of which neglecting the higher order terms, we get,

$$L(\Phi_0 + \Delta\Phi) \approx \frac{1}{2}\Delta\Phi^T\mathbf{A}\Delta\Phi + \mathbf{b}^T\Delta\Phi + c$$

The system $\mathbf{Ax} = -\mathbf{b}$ is solved in every Newton iteration and the nodal coefficients are updated as $\{\phi_i\} = \{\phi_0\} + \{x\}$, where the Poisson Hessian is given as,

$$\frac{\partial L}{\partial \phi_i \partial \phi_j} = A_{ij} = \frac{-1}{4\pi} \underbrace{\int_\Omega \nabla N_i(x) \nabla N_j(x) \, \mathrm{d}\mathbf{r}}_{Stiffness-Matrix} \tag{51}$$

A linear potential mixing scheme is also incorporated to achieve convergence.

4.3.2. Thomas-Fermi Problem. An approach similar to the Poisson Problem is adopted for the Thomas-Fermi Problem. The important difference here is that, the Thomas-Fermi problem is a non-linear problem unlike the Poisson Problem. Hence, we need to model the non-linear part into a quadratic form by neglecting the higher order terms of it's Taylor series expansion as below,

The lagrangian can be approximated as a Taylor Series Approximation (neglecting higher order terms)

$$L(\mathbf{u}_0 + \Delta\mathbf{u}) \approx \frac{1}{2}\Delta\mathbf{u}^T \mathbf{A} \Delta\mathbf{u} + \mathbf{b}^T \Delta\mathbf{u} + c \tag{52}$$

Taking the gradient,

$$\nabla L(\mathbf{u}_0 + \Delta\mathbf{u}) = \mathbf{A}\Delta\mathbf{u} + \mathbf{b} \stackrel{!}{=} \mathbf{0} \tag{53}$$

Setting $\Delta\mathbf{u} = 0$, we get,

$$\nabla L(\mathbf{u}_0) = \mathbf{b} \tag{54}$$

The system $\mathbf{A}\Delta\mathbf{u} = -\mathbf{b}$ is solved in every newton iteration and the nodal coefficients are updated as $\mathbf{u} = \mathbf{u}_0 + \Delta\mathbf{u}$, where the Thomas-Fermi Hessian is given by,

$$\left.\frac{\partial L}{\partial u_i \partial u_j}\right|_{\phi_i = const.} = \frac{\partial T_s}{\partial u_i \partial u_j} + \frac{\partial E_{xc}}{\partial u_i \partial u_j} + \frac{\partial (ES)}{\partial u_i \partial u_j} + \frac{\partial (\text{Lagrange})}{\partial u_i \partial u_j} \tag{55}$$

The Thomas-Fermi hessian is now,

$$\mathbb{R}^{(N+1)\times(N+1)} \ni \mathbf{A} = \begin{bmatrix} \mathbf{B} & c \\ c & 0 \end{bmatrix}.$$

where $B_{ij} = \left.\dfrac{\partial L}{\partial u_i \partial u_j}\right|_{\phi_i = const}$ and $c = \dfrac{\partial L}{\partial u_i \partial \lambda}$

The lagrange multiplier λ is considered here as another coefficient, that is, $\lambda = u_{N+1}$. The system $\mathbf{Ax} = -\mathbf{b}$ is solved in every newton iteration and the nodal coefficients are updated as $\boldsymbol{\{}\phi_i\} = \alpha\boldsymbol{\{}\phi_0\} + \mathbf{x}$. α is initially chosed to be 1, which implies that the full newton step is taken. If the gradient computed is higher than the old gradient, then α is halvened. This process is repeated till calculated gradient is less than the old gradient.

5. Numerical Implementation

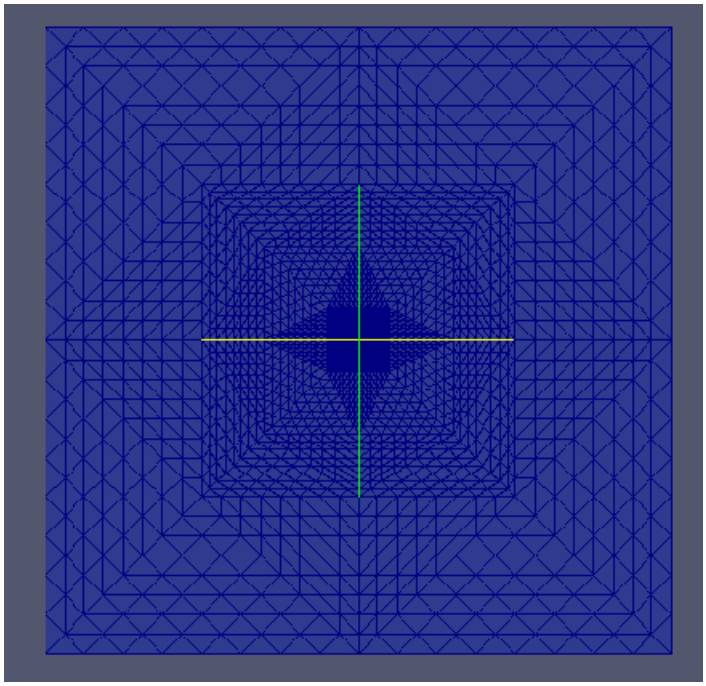
In the present work, the numerical implementation of the Orbital-Free Density Functional Theory has been carried out in a finite-element basis using the DUNE Numerics package; [33, 34, 35, 36, 37, 38, 39].

5.1. Mesh Generation

The meshes that we use have been created using C programs. We have created tailor-made meshes for the problems that we intend to solve. We apply our OF-DFT implementation to individual atoms, di-atomic molecules and small aluminium clusters.

The 3-D meshes for single atoms are generated such that the mesh is highly refined around the center of the domain where the atom will eventually be placed. There is a node at the center of the domain at which the atom is placed. The mesh coarsens away gradually as we go away from the center. Figure 6 shows a single atom mesh.

Figure 6: Single Atomic Mesh



Special meshes are also created for di-atomic molecules such as CO and N2. For generation of these meshes, the inter-atomic distance and the domain size are taken as modifiable parameters. The mesh is refined around the atom positions and gradually coarsened away towards the boundary. Figure 7 represents a typical mesh for a di-atomic molecule.

The meshes for aluminium clusters are made up of two parts, the inner atom part and the basic larger grid which is of the order of the lattice constant. The lattice constant and the number of atoms are the modifiable parameters for generation of these meshes. The meshes are so created that the region around the atom is highly refined and new nodal points are placed such that the mesh coarsens away as we move farther from the atom positions. Figure 8 shows the mesh for a small aluminium cluster.

Figure 7: Di Atomic Mesh

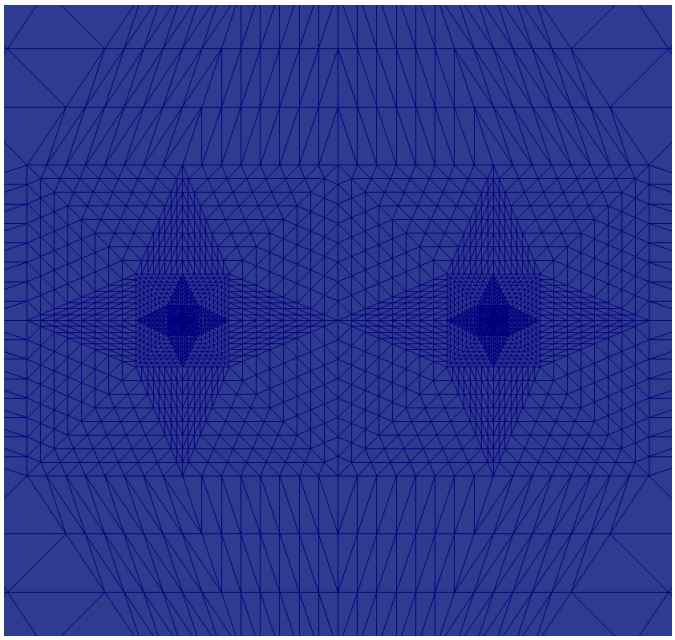
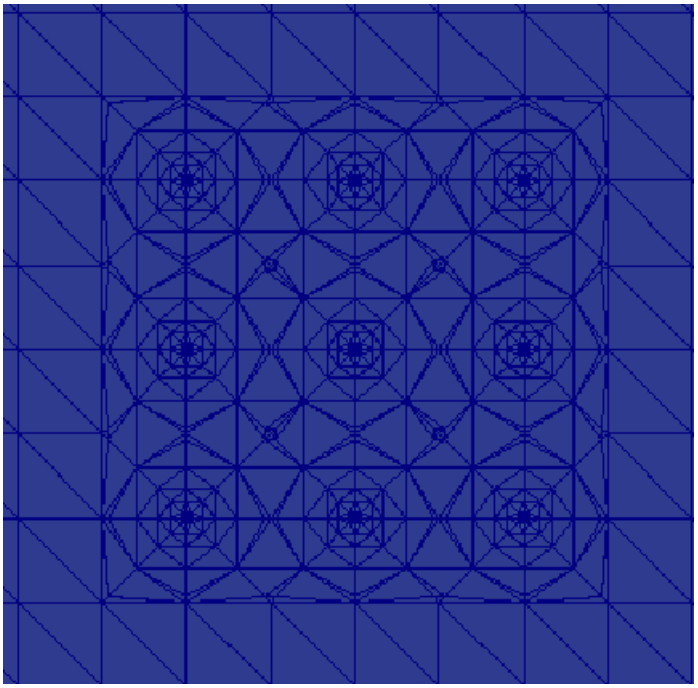


Figure 8: Mesh for Aluminium Clusters



5.2. Implementation with DUNE Numerics

As mentioned in the previous section, we use linear 3D 4-noded tetrahedral finite elements for our implementation. Subsequently, the domain is discretized using these finite elements.

The optimization procedures are applied on the Grid Function Space (GFS) of the FE mesh. Dirichlet boundary conditions are enforced such that $\{u_i\}$ and $\{\phi_i\}$ are zero on the boundary.

For numerical integration, 4th order quadrature rule has been used. A domain integral is broken down into sum of integrals over the elements. Each element integral is computed by iterating over the shape-functions of that element. In case of linear elements, the number of shapefunctions of the element is same as the number of vertices of that element. To sum it up, a domain integral is computed by nested iterations over the elements and then over the quadrature rule and then over the shapefunctions of the element.

It is important to figure out how the local node numbering for each element is mapped to global node numbers. This is achieved by using the subindex functionality of DUNE. The subindex function takes the local node number, the element and the co-dimension of the entity as input and returns the global nodal number of that particular vertex.

The complete picture of our algorithm is shown in figure 9. The algorithm starts with an initial value of the $\{\rho_i\}$ vector as the input. The initial guess of $\{\rho_i\}$ is such chosen that, it resembles the final $\{\rho_i\}$ the most, that is a gaussian distribution over the domain. In particular, $\{\rho_i\}$ is chosen to be $e^{-\alpha \mathbf{r}}$ at the start of the algorithm where $\mathbf{r}$ is the radial distance from the center of the domain. The algorithm is then taken through the conjugate gradient optimization procedures for both the Poisson problem and the Thomas-Fermi problem. With conjugate gradient, we obtain a configuration of $\{\rho_i\}$ which is in the vicinity of the real solution. The output $\{\rho_i\}$ from the CG method is used as the input for the Newton method which solves the Poisson problem and the Thomas-Fermi problem more efficiently to reach the real minimum of the energy functional with respect to the $\{\rho_i\}$. The final configuration of $\{\rho_i\}$ is stored and the energies at that point gives us the ground state energy of the system.

5.2.1. Conjugate Gradient Method. This method uses the gradients of the energy functional to carry out optimizations. The Poisson Problem requires the gradients with respect to $\{\phi_i\}$ whereas the Thomas-Fermi Problem asks for gradients with respect to $\{u_i\}$. The $\{u_i\}$ vector is initialized to a gaussian distribution over the domain. The gradients are preconditioned to speed-up the conjugate gradient iterative procedure. The preconditioner used for the Poisson Problem is the inverse of the Poisson Hessian Diagonal Matrix with all the non-diagonal elements set to zero. The preconditioner used for the Thomas-Fermi problem is the inverse of the Thomas-Fermi Diagonal Matrix with all non-diagonal elements set to zero. The standard conjugate gradient method is operated on the preconditioned gradients to obtain updates of $\{u_i\}$ and $\{\phi_i\}$.

The flowchart in figure 10 depicts the conjugate gradient implementation order. The Poisson problem is solved first and then the Thomas-Fermi problem till gradients for both these problems are below a certain tolerance. The tolerance criteria is slightly loose for this method unlike the Newton method. This is because, we employ the CG method just to get closer to the real solution whereas in the Newton method, we actually need to converge at the real solution.

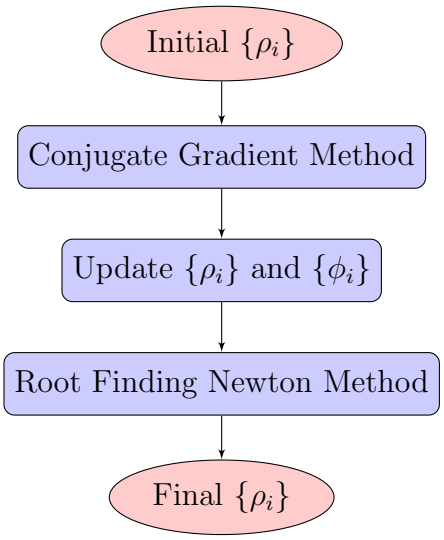


Figure 9: Complete Algorithm

5.2.2. Root-Finding Newton Method. The crux of this method is in assembling the right hessian matrices and solving them. The hessian matrix for the Poisson Problem and Thomas-Fermi Problem are assembled first. To this end, sparcity patterns for both the matrices are worked out. All codim-1 entities of the current element are traversed and all pairs of vertices are stored. The sparcity pattern of the assembler is set with the information gained during the storage of the adjacency data. An index is added to the hessian matrix for each non-zero element and later these indices are used while assembling the hessian. The hessian matrices are assembled in the standard procedure by iterating over the elements, then over the quadrature rule and then over the shapefunctions. The boundary conditions for these matrices are set such that diagonal elements are set to unity and the rows corresponding to the boundary are set to zero.

The right-hand side vectors of the systems to be solved are also assembled and assigned the appropriate values. Thereafter, the $\mathbf{Ax} = \mathbf{b}$ systems are solved using DUNE solvers. To ease and hasten the solving process DUNE preconditioners are used before solving the systems. For the Poisson Problem, SeqSOR preconditioner is used and BiCGSTABSolver is used for solving the system. For the Thomas-Fermi Problem, the preconditioner used is SeqILUn and again we use the BiCGSTABSolver to solve the system.

The control of the Newton algorithm is reversed as compared to the conjugate gradient method, that is, the Thomas-Fermi problem is solved first and then the Poisson problem. This is because, for the Newton method, the initial solution should always be in the vicinity of the actual solution. When we switch from the CG method to the Newton method, we have $\{\rho_i\}$ from the CG which is in the vicinity of the real solution. Hence, we input that $\{\rho_i\}$ into the Thomas-Fermi Newton algorithm unchanged. The figure 11 provides an overall picture of the Newton method.

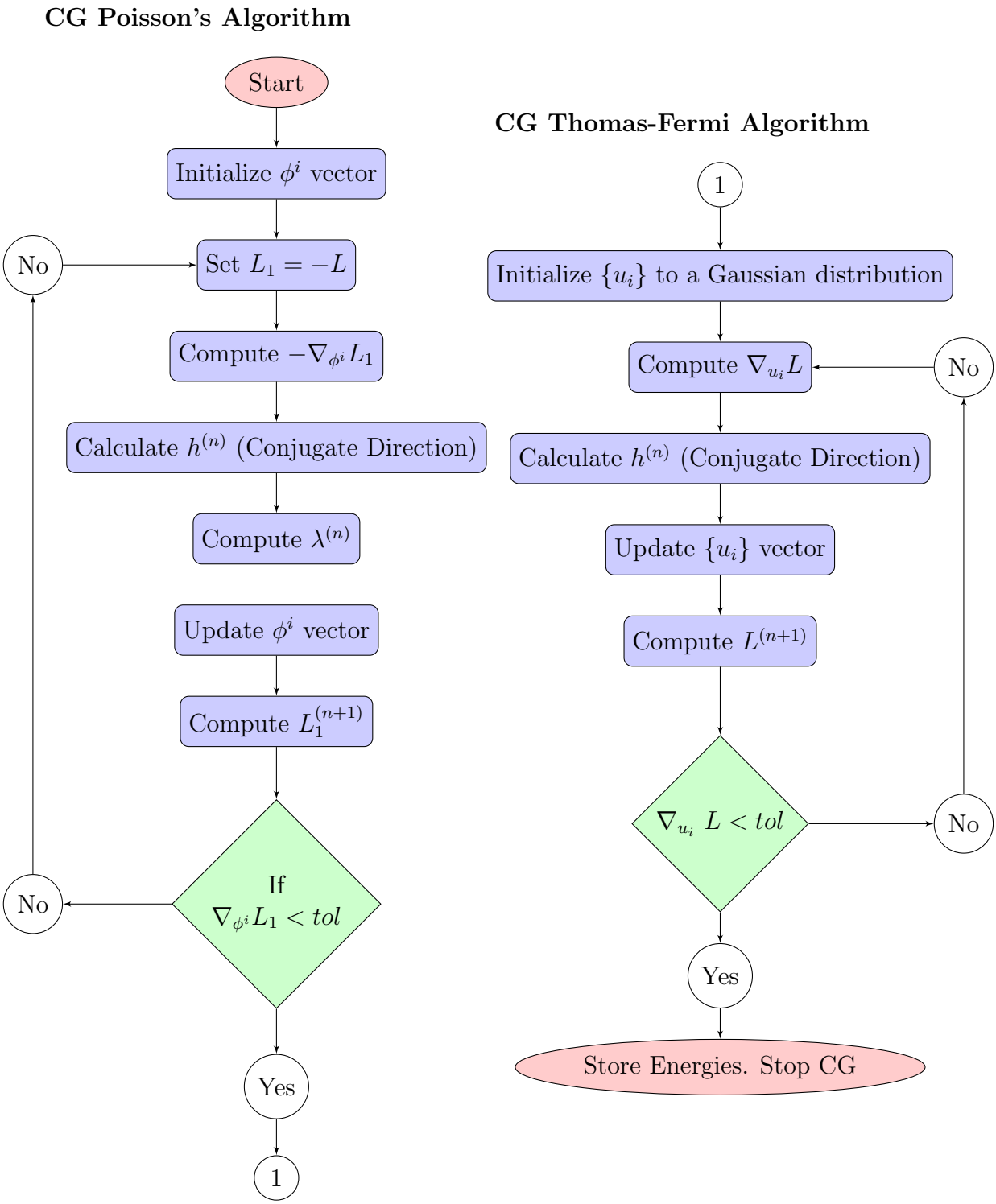


Figure 10: CG Algorithm

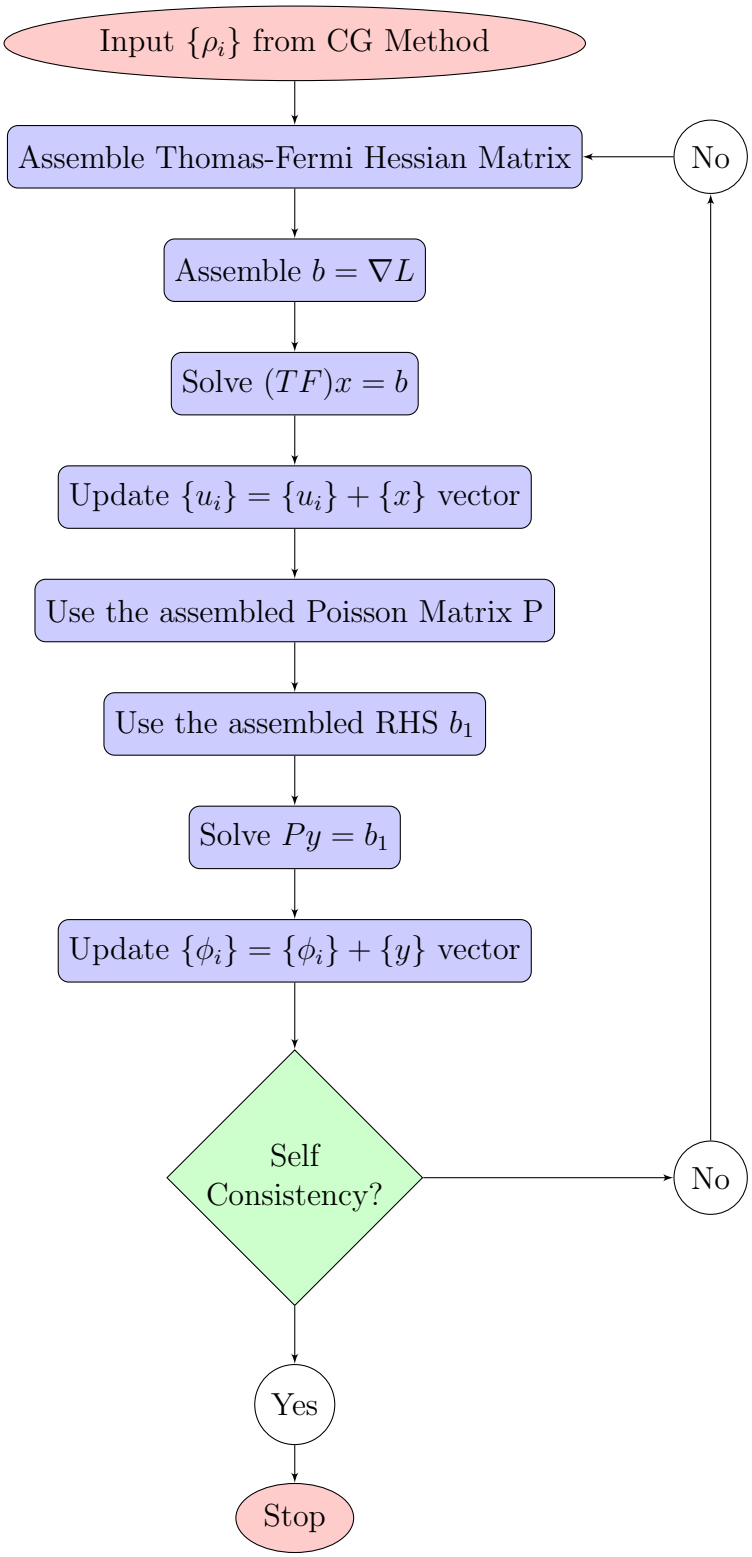


Figure 11: Overall Sketch of Newton Algorithm

6. Numerical Examples

In the previous chapter, two different methods of solving the saddle-point formulation of OF-DFT have been discussed. The first method is basically a conjugate-gradient method which treats the constraint in the optimization problem with a penalty parameter. In the second method, the constraint is handled by a lagrange multiplier. For convenience, in the following sections the two methods have been denominated as the 'conjugate-gradient method' and the 'Newton method' respectively. A coupled approach which combines the conjugate gradient method and the Newton method is used to obtain the final saddle-point configuration and the corresponding energy minimum. This is done by passing the final $\{u_i\}$ and $\{\phi_i\}$ configuration of the conjugate-gradient method into the Newton method. The Newton method requires the starting density, that is, the initial solution to be in the vicinity of the actual solution. The conjugate-gradient coupled Newton method achieves the same, as the conjugate-gradient method brings down the gradients low enough to pass on values to the Newton method which are close to the real minimum configuration. This approach of solving the OF-DFT problem is validated by the simulations presented in the next sections.

6.1. Atoms

The test case is the neon atom, which is the largest atom in our simulations. We are going to do a comparative study of our results with previous OF-DFT studies. To this purpose, we compare with the FE-OFDFT work by [12] who use $\lambda = 2/9$ as against $\lambda = 2/10$ used by us for the Weizsäker term. We also compare our results with [14] who use spline basis in their work. [14] use a slightly different functional than us as they do not have the correlation term in the energy functional. They employ though the same λ value of 2/10 as we do for the Weizsäcker term.

In the present study, we use $\lambda = 0.2$, which is the optimal value of the parameter λ; [14]. The ground-state energies of various atoms computed in our simulations have been tabulated in Table 1. The energies are compared with OF-DFT studies by [12] & [14] and Kohn-Sham-DFT values of NIST. We observe excellent agreement of the results of our simulations with literature.

Table 1: Energies of atoms, in atomic units

	Present	Stich et al. [14]	Gavini et al. [12]	NIST
He	-2.704	-2.818	-2.91	-2.834
Li	-7.218	-7.323	-7.36	-7.335
C	-37.967	-38.033	-	-37.425
N	-54.881	-54.943	-	-54.025
O	-74.637	-75.577	-	-74.473
Ne	-128.700	-128.80	-123.02	-128.233

Figure 12 shows the electron density around the neon atom in a 2-D mid-plane cut. Electron density is highest at the nucleus. This simulation of the neon atom is run with a polygonal mesh as shown in figure 13.

Figure 14 and figure 16 shows the convergence of our finite-element appoximation. We compare the convergence of the ground-state energy of the neon atom with two different

meshing schemes. The meshing schemes are shown in figure 13 and figure 15. Figure 13 represents a polygonal mesh, where we start with a mesh of 277 DOFs and uniformly refine it three times to obtain the final mesh. The intermediate meshes used in the convergence plot in figure 14 contain 277, 2025, 15729 and 124513 DOFs respectively. The second meshing scheme is shown in figure 15, which is a cubical mesh. In the second scheme, we start with a mesh of 183 DOFs and uniformly refine it three times to obtain the final mesh. The intermediate meshes used in the convergence plot in figure 16 contain 183, 1301, 9993 and 78737 DOFs respectively. It seems to us that the polygonal 3-D mesh is more suitable for our calculations as it converges faster, although a strong inference cannot be drawn on this because it has more elements for each uniform sub-division.

We have made a good observation in Table 2, where we compare the energy components of neon in our orbital-free calculations and Kohn-Sham DFT calculations obtained from NIST. The energy contributions are different to those of Kohn-Sham DFT, but sum up to a total energy that is similar again. The reason for this behavior is explained in previous theory chapters.

Table 2: Comparison of orbital-free and orbital based methods, energy components of neon

Ne	OF-DFT	KS-DFT
E_{total}	-128.700	-128.233
E_{kin}	+123.713	+127.738
E_{xc}	-10.209	-11.710
E_{coul}	+54.204	+65.726
E_{enuc}	-296.409	-309.988

At the end of this section, we would like to make a note that, we obtained the ground-state energy of neon from our calculations as -128.80 Hartree, which is as accurate as possible. That said, we also observed an extremum point at -127.414 Hartree. If we switch from the conjugate-gradient method to the Newton method earlier than needed, we end up getting trapped in this extremum point. At this moment, we cannot really comment if this is a local minimum, a maximum or a saddle-point at an excited state. We have two methods in the present work, first a conjugate-gradient method, which is really searching for a minimum in the functional and a 'Newton method', which finds zeros of the gradient, which can therefore also end up in saddle points or (local) maxima of the functional. To be sure one would have to calculate the hessian and see whether its positive or not, something we have not investigated in the scope of the present work. An alternative is to run the conjugate- gradient method again after the Newton method, so that we have a 'conjugate-gradient coupled Newton coupled conjugate-gradient' method. This method should find the minimum. We ran this simulation on the neon atom and found that the ground-state energy of -128.80 Hartree is really a minimum, nevertheless we have not been able to identify if the extremum at -127.414 Hartree is a local minimum, maximum or a saddle-point. This makes the energy landscape of neon quite interesting though and calls for further analysis.

Figure 12: Electron density around a neon atom calculated with 124513 DOFs with the mesh in figure 13

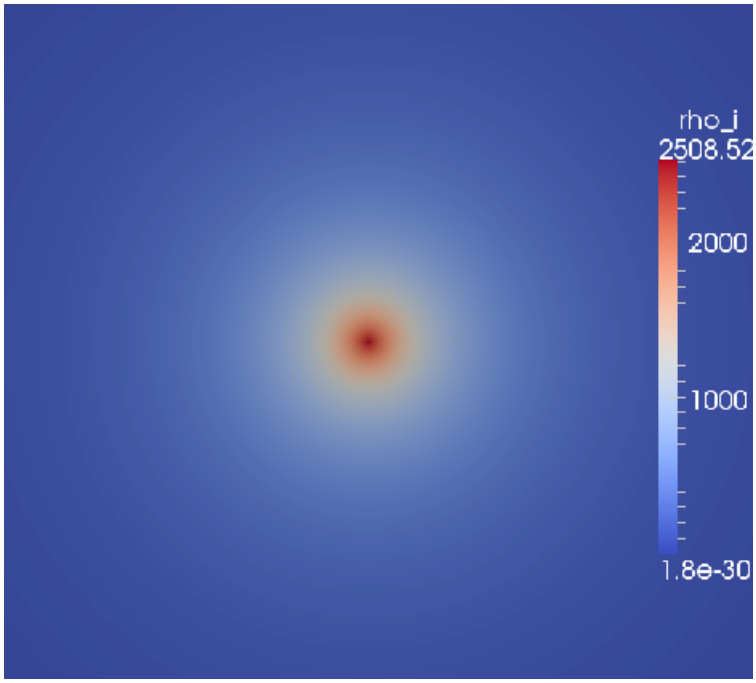


Figure 13: Polygonal mesh with 124513 DOFs

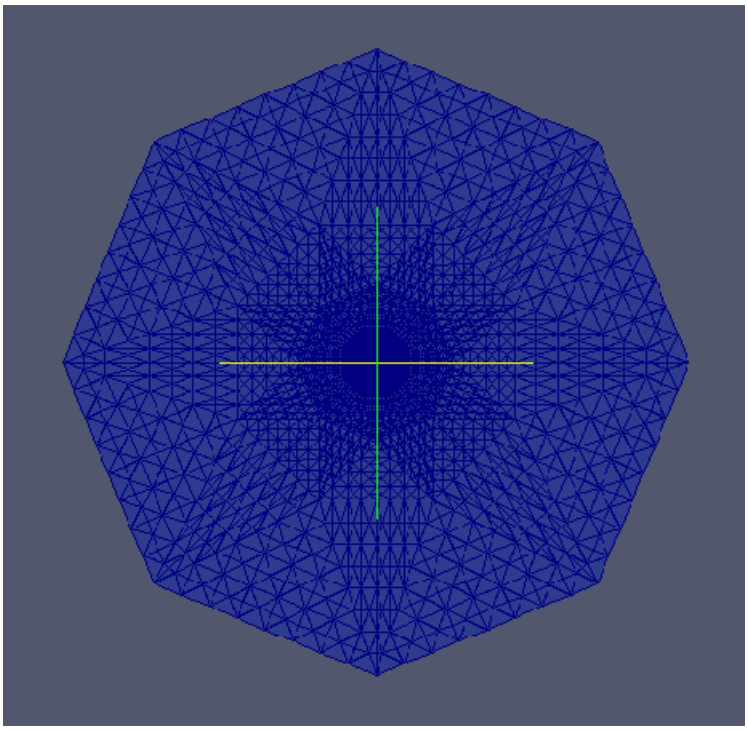


Figure 14: Convergence plot of Neon atom with polygonal mesh

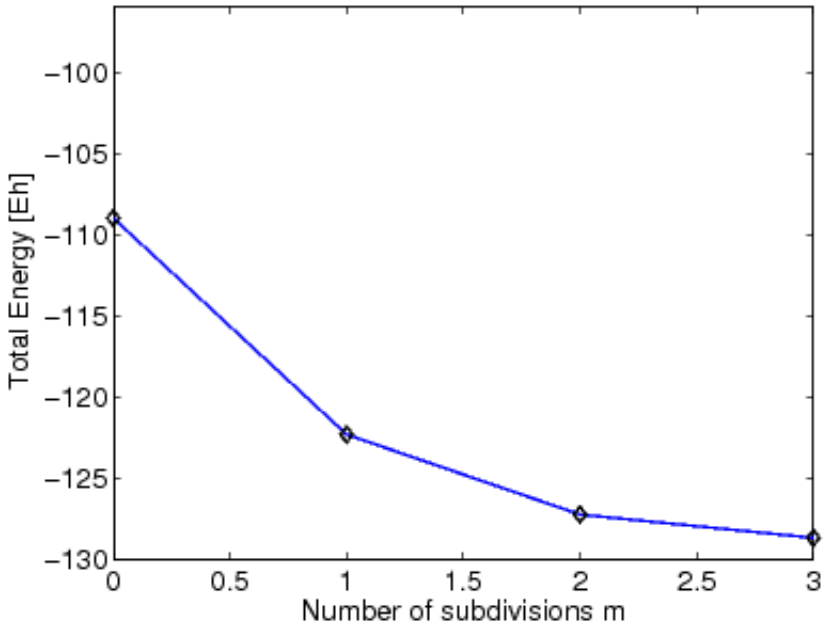
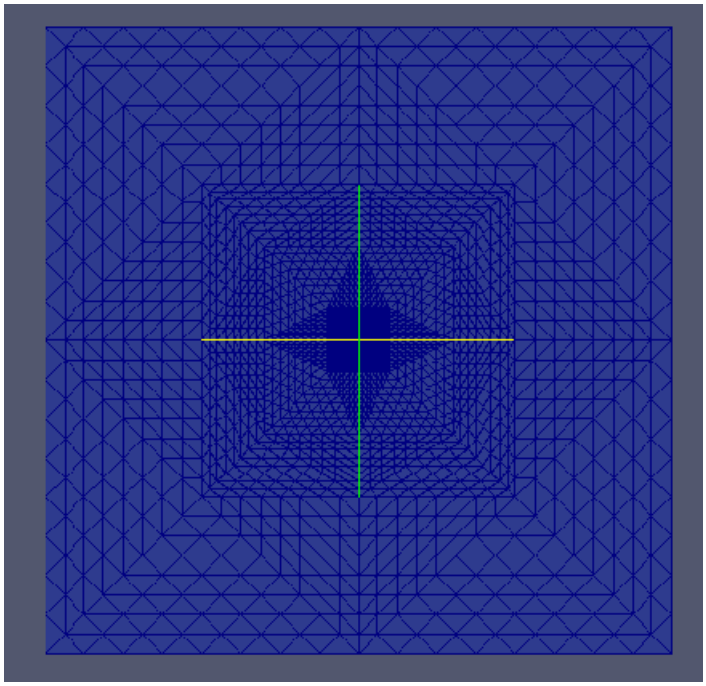


Figure 15: Mesh with a cubical domain Shape with 78737 DOFs



6.2. Molecules

The next application of our OF-DFT implementation is to simulate binding in molecules. It should be mentioned here that, the Thomas-Fermi model is poor at predicting binding, refer Teller non-binding theorem mentioned in earlier chapters; [5]. It is the Weizsacker correction in the OF-DFT functional which helps us predict binding of CO and N_2 molecules in the present study. Figure 17 shows the mesh for a di-atomic molecule. Figure 18 and figure 19, show the electron density around CO and N_2 molecules respectively and we can clearly visualize the binding between atoms. It's important to point out that the meshes for these molecules are so designed that the regions near the atoms and between the atoms are highly refined compared to the outer regions. This helps us capture binding. Figure 20 shows the plot over line of a mid-plane section of the CO contour of electron density. This figure depicts the assymetry of the CO molecule, where C is a smaller atom compared to O and hence the electron density at the center of C atom is lower compared to the electron density at the center of O atom.

Figure 21 shows the convergence of our meshing scheme for 2-atom molecules. The convergence of ground-state energies of CO at uniform subdivisions of the initial mesh is plotted in figure 21. The initial 2-atom mesh has 305 DOFs and it's uniformly refined three times to obtain the final mesh. The intermediate meshes used in the convergence plot in figure 21 contain 305, 2217, 17201 and 136161 DOFs respectively.

In Table 3, we report the ground state energies of CO and N_2 molecules. We find that the ground state energies of these molecules at sample interatomic distances are in good agreement with ground state energies of the open source Kohn-Sham DFT code ORCA, applying the VWN-5 functional; [11]. These calculations have been made with a mesh of 136131 DOFs which is obtained by 3 uniform subdivisions of the initial mesh.

Table 3: Ground state energies of CO and N_2 molecule [136131 DOFs]

Molecule	Present [E_h]	ORCA [E_h]
CO[2.6 a.u.]	-112.047	-112.391
N_2[2.4 a.u.]	-106.542	-108.636

We then do a series of calculations of total energies of N_2 at different interatomic distances. This is done with a mesh of 17201 DOFs, which is obtained by 2 uniform subdivisions of the same initial mesh as used for comparison with ORCA. As the mesh is not highly refined, we can expect the total energy values to be not so accurate. These values are tabulated in Table 4. Using these values, we compute the bond length of N_2 as the interatomic distance where the total energy is minimum, see figure 22. We also observe strong repulsive energy at interatomic distance of 1.0 a.u. The minimum energy configuration corresponds to interatomic distance 2.0 a.u., which is the most stable configuration of N_2. The binding energy of N_2 is also calculated using the results of these computations which is tabulated in Table 5. Binding energy is calculated as, binding energy = E_{AB} - E_A - E_B, where E_{AB} is the ground-state energy of the molecule and E_A & E_B are the ground-state energies of the single atoms. The same mesh used for ground-state energy calculations of N_2 molecule is used for energy calculations of single atom N. It's assumed that the errors will cancel out due to the low-refinement of the mesh [17201 DOFs] during binding energy calculations. The conversion of 1 Hartree = 27.2107 eV is used for calculation of binding energy. It's important to note that orbital-free kinetic

Figure 16: Convergence plot of Neon atom with a mesh with a cubical domain shape

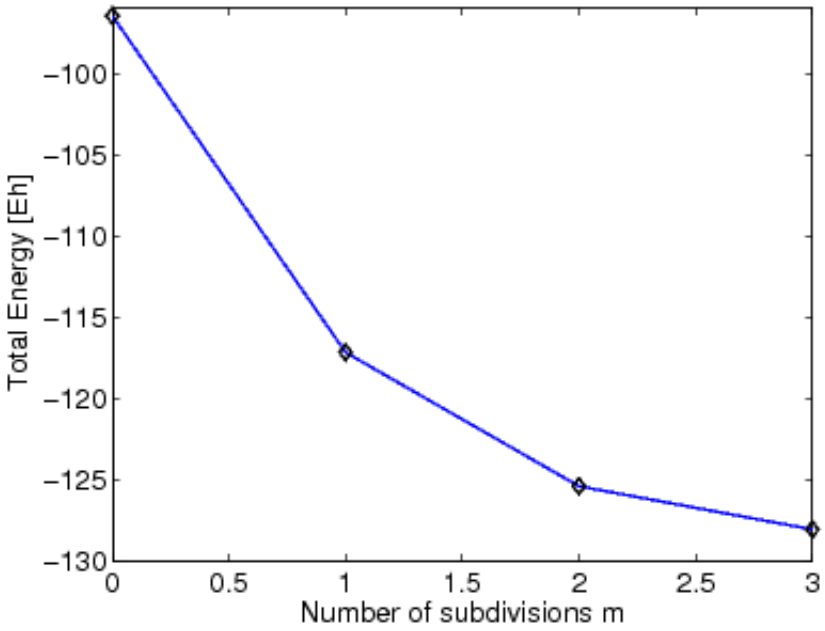


Figure 17: Di-Atomic Mesh

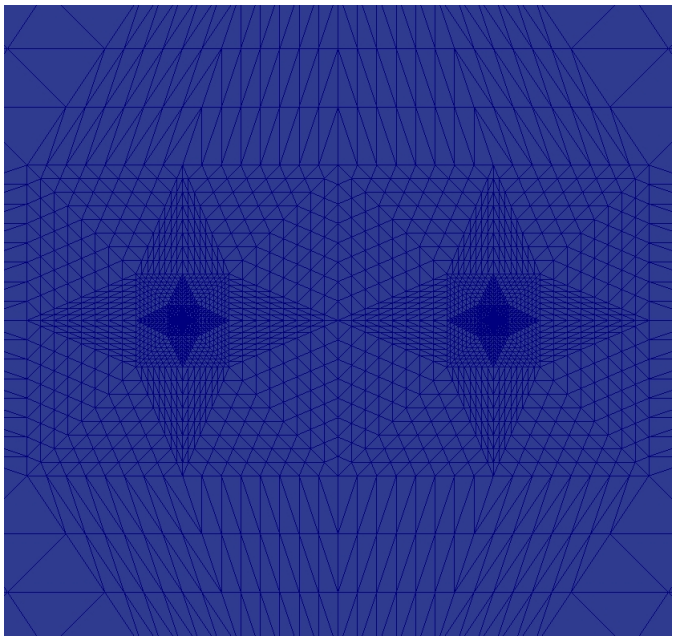


Figure 18: Electron density around a CO molecule

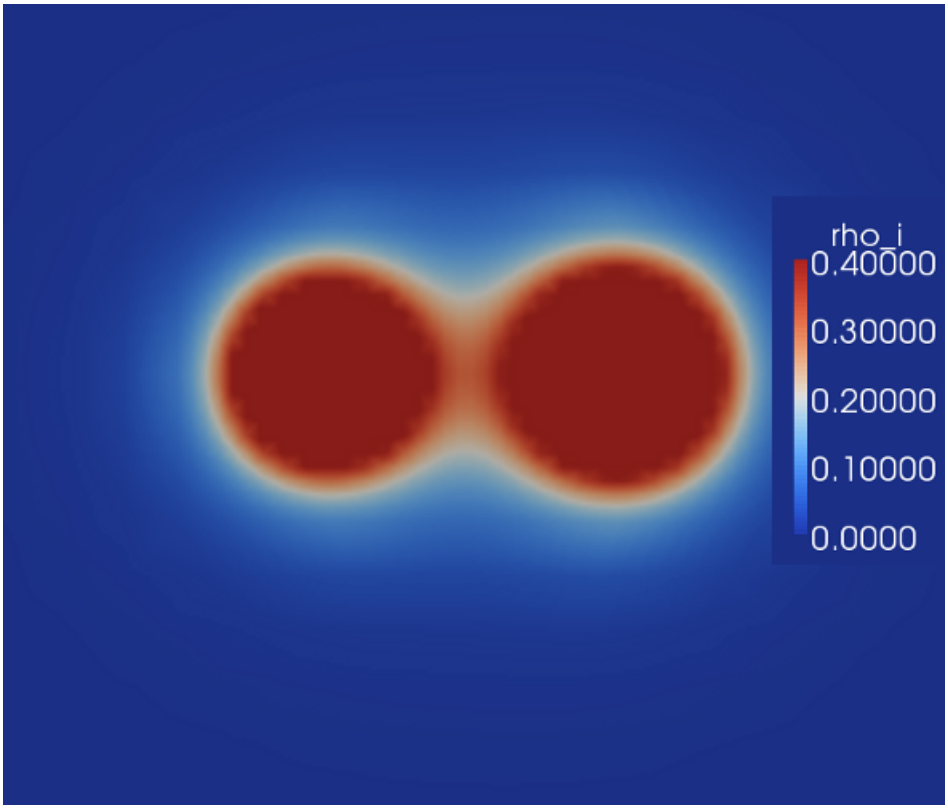


Figure 19: Electron density around a N_2 molecule

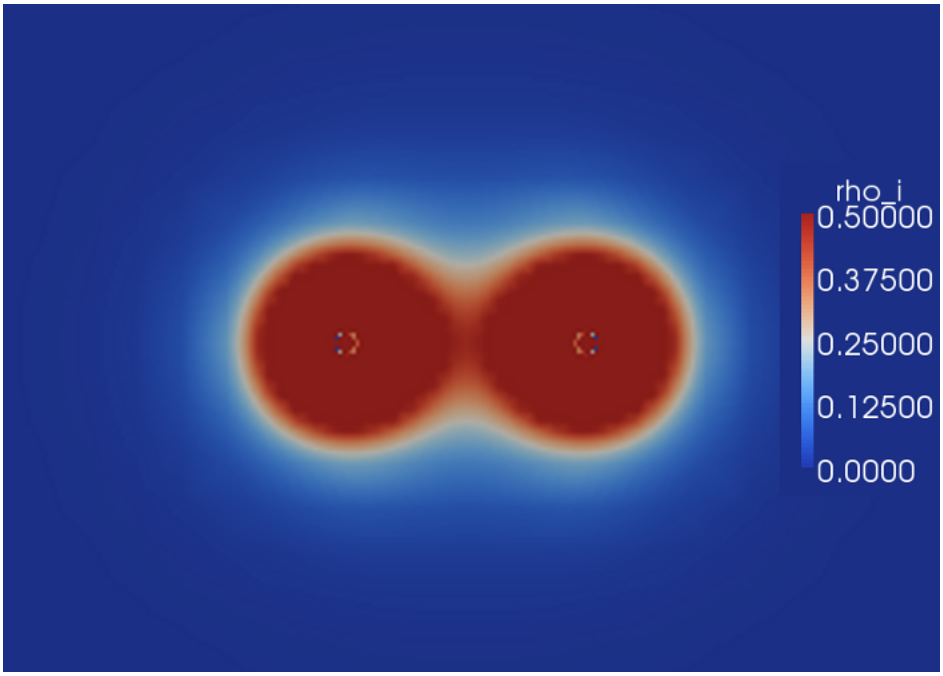


Figure 20: Plot over line for CO

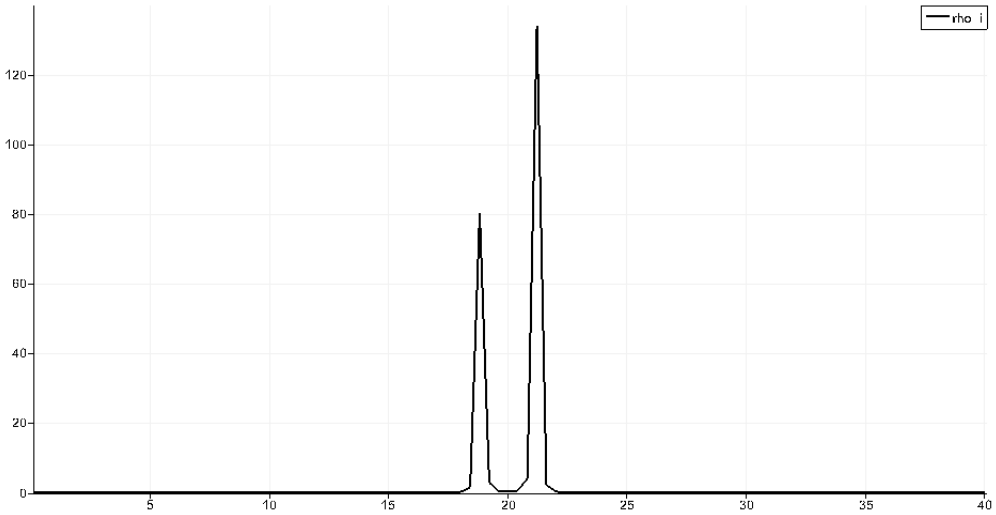
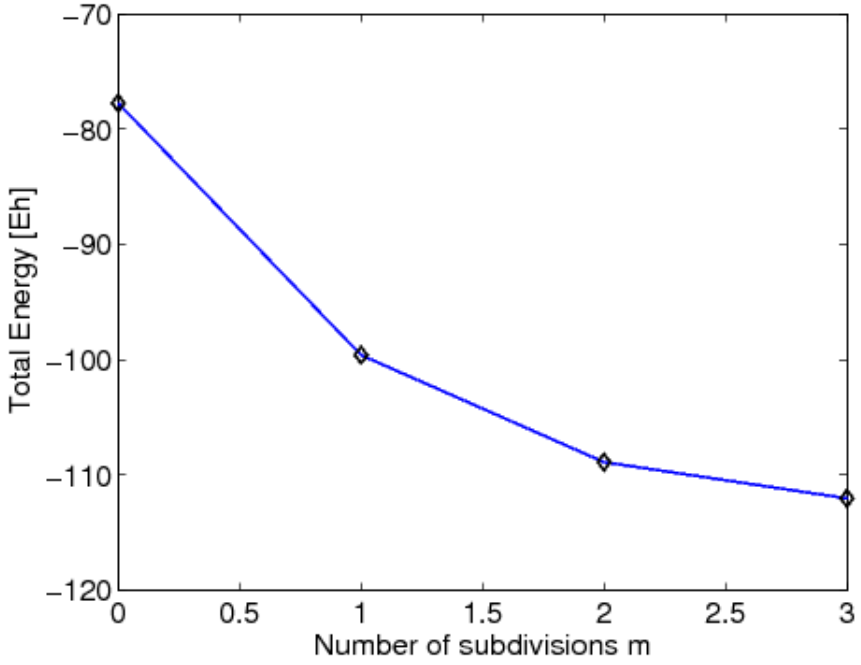


Figure 21: Convergence plot of CO



energy functionals have strong limitations in the presence of covalent bonds; [**?**] and the orbital-free theory is not suitable for molecules like N_2 and CO. It's just a demonstration of our methodology which we have presented here.

Figure 22: Bond Length of N_2

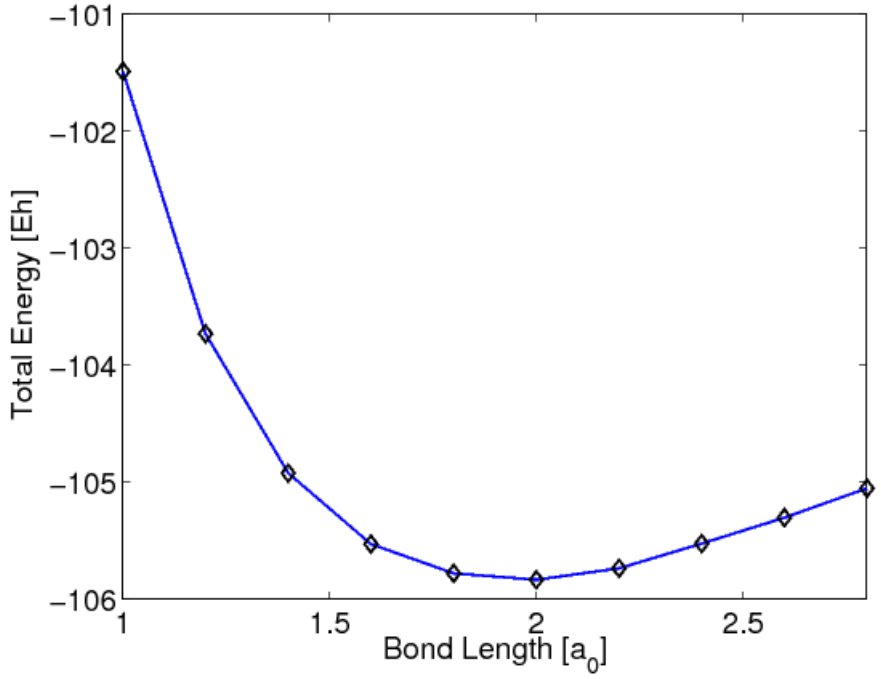


Table 4: Ground state energies of N_2 molecule [17201 DOFs]

Dist.[a.u.]	Ground-state energy [E_h]
1.0	-101.492
1.2	-103.733
1.4	-104.922
1.6	-105.529
1.8	-105.778
2.0	-105.833
2.2	-105.735
2.4	-105.524
2.6	-105.303
2.8	-105.051

To conclude this section, we try to point out a few possible inadequacies of our methodology which might lead to discrepancies and something one should always be careful about. Firstly, we have used a conjugate-gradient coupled Newton method in our calculations. It's important to point out that the Newton method is not a minimum finding algorithm but rather an extremum finding method. We cannot therefore guarantee that the ground-state energy values that we obtain are real minima. Rather, they can be excited state saddle points. This eventuality can be ruled out by checking for the positive definiteness of the hessian matrix. Secondly, for the binding energy and bond length calculations

Table 5: Binding energy and bond length of N_2 molecule [17201 DOFs]

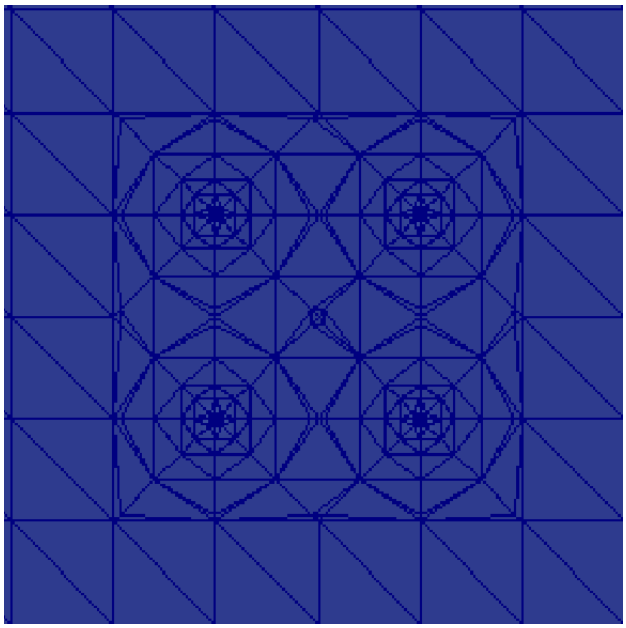
Property	Present	Gavini. et al. [12]	Experiments [12]	HF [12]	KS-LDA
Binding energy (eV)	-6.6	-12.6	-11.2	-7.9	-9.6
Bond length (a.u)	2.0	2.7	2.07	2.01	2.16

of molecules, we have done a series of computations with meshes specifically designed for 2-atom molecules at different interatomic distances. This can get tricky, because in our solution procedure, we solve for the minimum of the OF-DFT functional in the GFS (Grid Function Space). So, if the meshes generated for 2-atom molecules at different interatomic distances are not refined enough or are not similar enough, then there can be discrepancies in the binding energy vs interatomic dist. curve and deviations from the original nature of this curve. For us, it has not been easy to reproduce the exact binding energy vs interatomic dist. curves presented in [12]. We attribute these discrepancies to the 2-atom meshes we have used.

6.3. Aluminium Clusters

We have applied our OF-DFT implementation to simulate the electron density around small aluminium clusters. Figure 23 and figure 24 shows the mesh for a $2\times2\times2$ aluminium cluster and the electron density around a $2 \times 2 \times 2$ aluminium cluster respectively. The mesh is highly refined near the atoms. Similarly, figure 25 and figure 26 represents the mesh for a $3 \times 3 \times 3$ aluminium cluster and the corresponding electron density countour plot respectively.

Figure 23: Mesh for $2 \times 2 \times 2$ Aluminium cluster



Similar to the binding energy calculations of molecules in the previous section, binding energy per atom of aluminium clusters can be computed from the ground-state energy values of these alumnium clusters. Binding energy can be calculated as, binding energy $= (E(n) - nE_0)/n$ where E(n) is the energy of the cluster per unit cell consisting of n atoms and E_0 is the energy of a single atom. However, it's pertinent to point out that,

Figure 24: Electron density around a $2 \times 2 \times 2$ Aluminium cluster

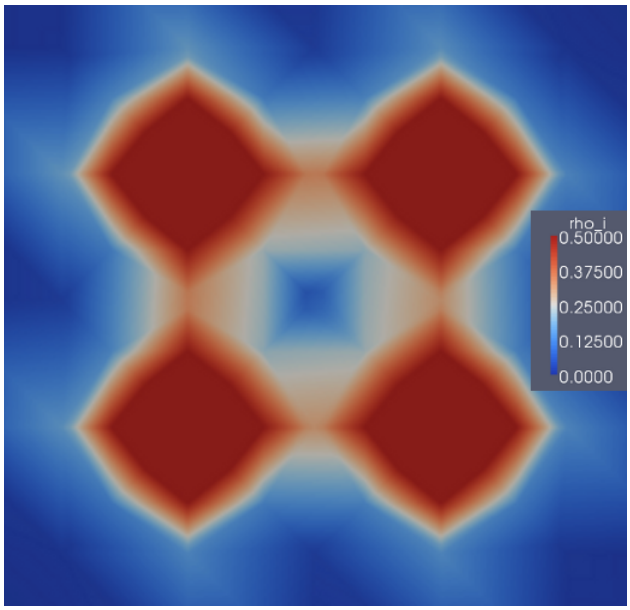
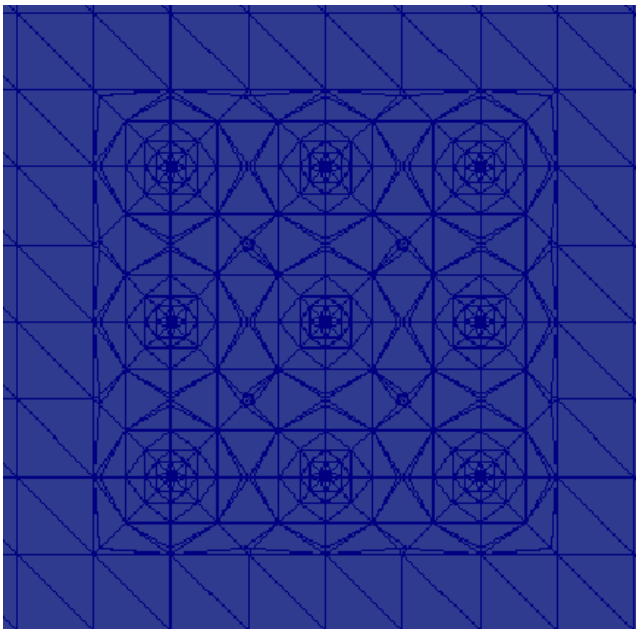


Figure 25: Mesh for $3 \times 3 \times 3$ Aluminium cluster



aluminium clusters have covalent bonds and OF-DFT is not the best method for systems with covalent bonds.

6.4. Computational Time

We now have a look at the computational time of our simulations using the Valgrind program. The code has been profiled and we have run a test simulation on the neon atom with a mesh of 131 DOFs. We find that the conjugate gradient method is much slower than the Newton method for our test case. This is noted by running both the methods separately till convergence and measuring the time for each run. It's also observed that the majority of the simulation time is taken by gradient calculations, see figure 27. It is interesting to note that, solving the poisson problem is considerably less expensive than the Thomas-Fermi problem; figure 27. The gradient calculation by the conjugate gradient method takes up around 53% of the total time which is the most expensive function. This is because, for every iteration of the conjugate-gradient gradient calculation function, there is a line-search loop embedded in it. This line-search includes another gradient calculation, thus making the outer function quite expensive. 35% of the time is taken by the gradient computation done by the Newton method. Also, computational time significantly depend on the size of the hessian matrix which is proportional to the number of DOFs. Figure 28 and figure 29 shows the percentage break-up of the conjugate-gradient gradient calculation and the Newton gradient calculation functions respectively. It should be noted that the pow functions (in the exchange-correlation terms) take the bulk of the time in these gradient calculations. It is important to note that the above observations are specific to our test case.

With our experience of these simulations, we have seen that the use of preconditioners in the penatly method has significantly reduced computational time. In our study, the Newton method seems to converge faster than the conjugate-gradient method, but we cannot draw a definite conclusion on this. If the initial guess is a good one, the method will achieve convergence faster. Also, for the conjugate-gradient method, we use the simplest preconditioner now, which is the digonalized hessian matrix. Moving to better preconditioners can reduce the computational time of the conjugate-gradient method. Additionally, parallelizing the code is strongly recommended as a future work of this study.

Figure 26: Electron density around a $3 \times 3 \times 3$ Aluminium cluster

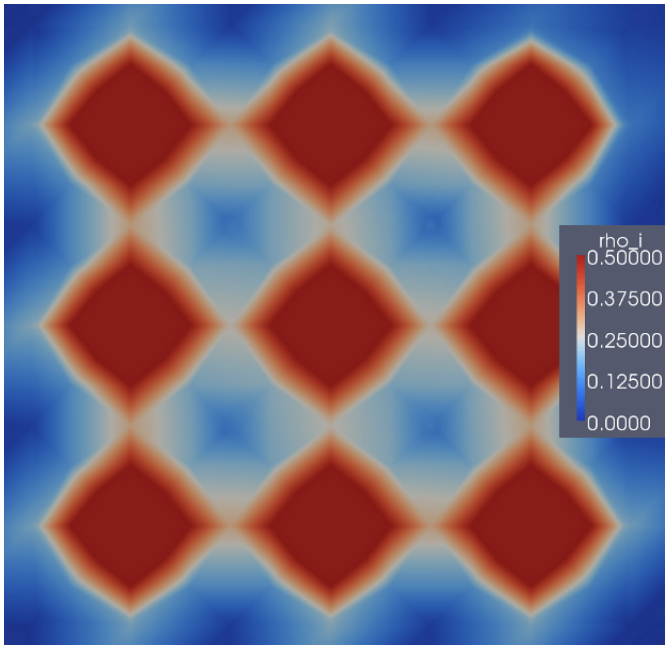


Figure 27: % break-up of the main function

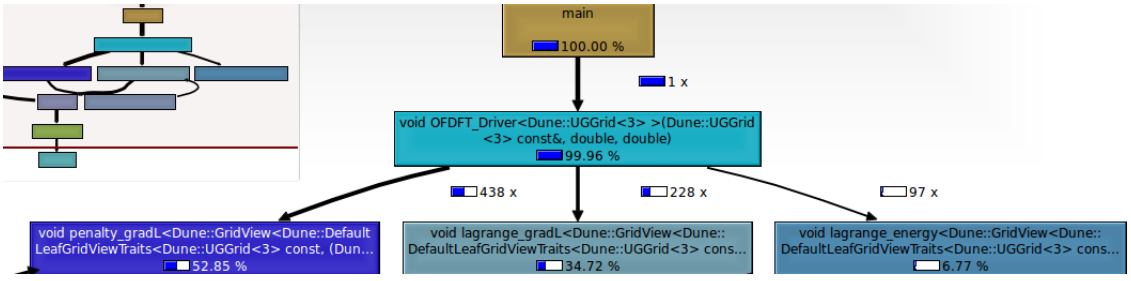


Figure 28: % break-up of the conjugate-gradient gradL function

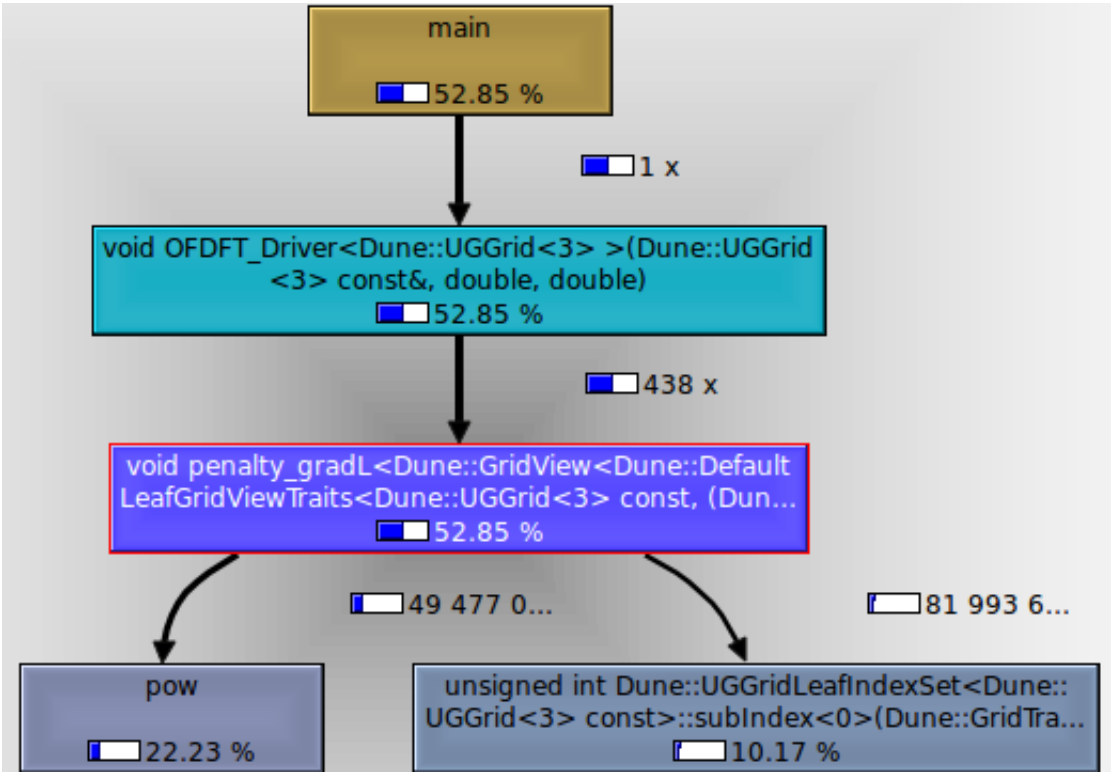
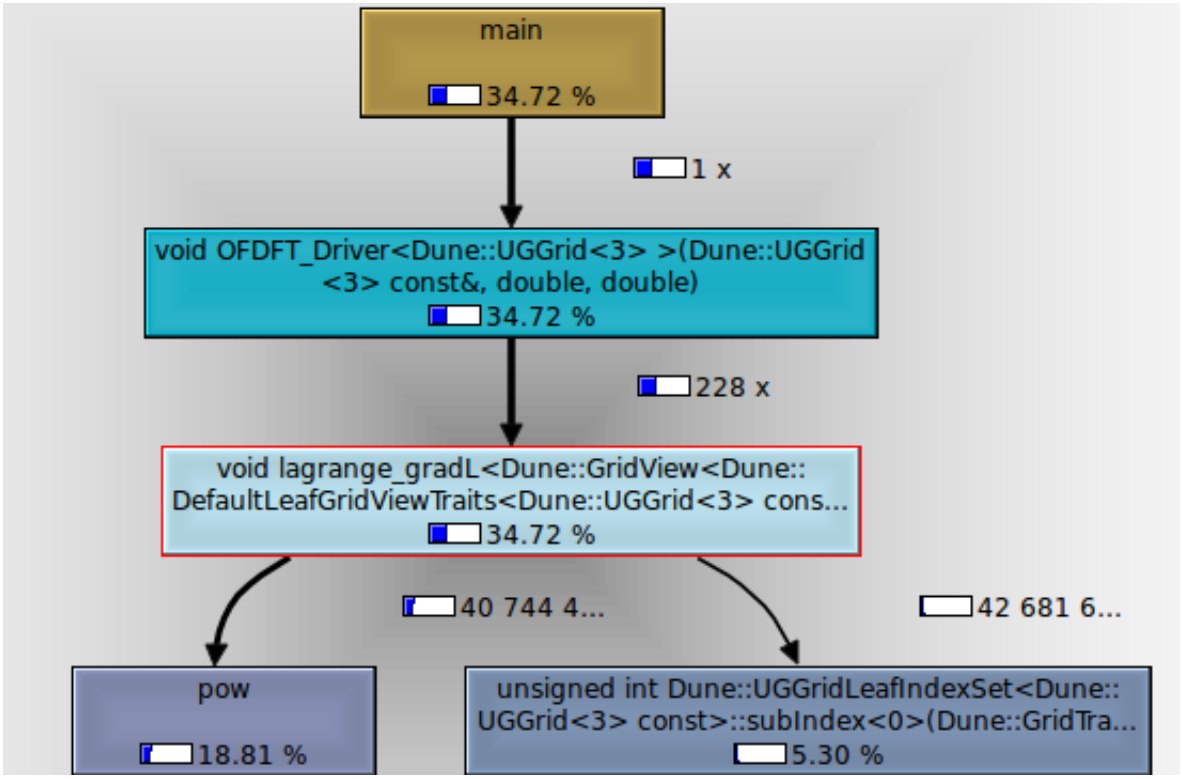


Figure 29: % break-up of the Newton gradL function



7. Conclusion and Future Directions

To sum up the present work, two different methods of solving the OF-DFT problem have been implemented. Both methods are compared and the results are found to be matching. Excellent agreement with literature has been achieved for the energies of single atoms. The ground-state energies of the molecules have been found to be in reasonable agreement with other DFT codes. Additionally, good contour plots for electron densities of atoms, molecules and small aluminium clusters have been obtained. Furthermore, the computational time of calculation by both the methods is studied and it is found that the Newton method is significantly faster than the conjugate gradient method. We also infer that the problem is highly mesh dependant because the optimization routines are performed on the Grid Function Spaces (GFS). We have achieved significantly good results compared with past FE-OFDFT work [12], with far less number of finite elements and without parallelization.

To obtain more accuracy, use of higher order elements is recommended. Parallelization of the code is required for simulating larger aluminium clusters with good accuracy. It is important to appreciate that, the finite element method is an approximate technique and to obtain good accuracies we need higher refinements of the meshes we have used and hence parallelization is important. Use of higher order elements is an intelligent solution to this end.

Appendix

A. Required Derivatives

$$L(\rho, \mathbf{R}, \phi) := C_F \int_\Omega u^{10/3}(\mathbf{r})\,\mathrm{d}\mathbf{r} + \frac{\lambda_1}{2}\int_\Omega |\nabla u(\mathbf{r})|^2\,\mathrm{d}\mathbf{r} + \int_\Omega \epsilon_{xc}(u^2(\mathbf{r}))u^2(\mathbf{r})\,\mathrm{d}\mathbf{r} - \frac{1}{8\pi}\int_{\mathbb{R}^3} |\nabla\phi(\mathbf{r})|^2\,\mathrm{d}\mathbf{r} + \int_{\mathbb{R}^3} (u^2(\mathbf{r}) + b(\mathbf{r}))\phi(\mathbf{r})\,\mathrm{d}\mathbf{r} + \lambda(\int_\Omega u^2(\mathbf{r})\,\mathrm{d}\mathbf{r} - N) \tag{56}$$

such that: $\nabla_\Phi L = 0$, $\nabla_{\mathbf{u}} L = 0$ and $\nabla_\lambda L = 0$

$$\frac{\partial T_s}{\partial u_i} = C_F\left(\frac{10}{3}\right)\int_\Omega u^{7/3}(\mathbf{r})N_i(x)\,\mathrm{d}\mathbf{r} + \lambda\int_\Omega \nabla u(\mathbf{r})\nabla N_i(x)\,\mathrm{d}\mathbf{r} \tag{57}$$

$$\implies \frac{\partial T_s}{\partial u_i \partial u_j} = C_F\left(\frac{10}{3}\right)\left(\frac{7}{3}\right)\int_\Omega u^{4/3}(\mathbf{r})N_i(x)N_j(x)\,\mathrm{d}\mathbf{r} + \lambda\int_\Omega \nabla N_i(x)\nabla N_j(x)\,\mathrm{d}\mathbf{r} \tag{58}$$

$$\frac{\partial(ES)}{\partial u_i} = 2\int_{\mathbb{R}^3} u(\mathbf{r})\phi(\mathbf{r})N_i(x)\,\mathrm{d}\mathbf{r} \tag{59}$$

$$=> \frac{\partial(ES)}{\partial u_i \partial u_j} = 2\int_{\mathbb{R}^3} \phi(\mathbf{r})N_i(x)N_j(x)\,\mathrm{d}\mathbf{r} \tag{60}$$

$$\frac{\partial(Lagrange)}{\partial u_i} = \lambda\int_\Omega 2uN_i(x)\,\mathrm{d}\mathbf{r} \tag{61}$$

$$=> \frac{\partial(Lagrange)}{\partial u_i \partial u_j} = \int_\Omega 2N_i(x)N_j(x)\,\mathrm{d}\mathbf{r} \tag{62}$$

$$=> \frac{\partial(Lagrange)}{\partial\lambda\partial u_i} = \int_\Omega 2uN_i(x)\,\mathrm{d}\mathbf{r} \tag{63}$$

$$\frac{\partial E_{xc}}{\partial u_i} = \int_\Omega (\frac{\partial \epsilon_{xc}}{\partial u} u^2 + 2\epsilon_{xc} u) N_i(x)\, \mathrm{d}\mathbf{r} \tag{64}$$

$$\implies \frac{\partial E_{xc}}{\partial u_i \partial u_j} = \int_\Omega \frac{\partial^2 \epsilon_{xc}}{\partial u^2} u^2 N_i(x) N_j(x)\, \mathrm{d}\mathbf{r} + \int_\Omega \frac{\partial \epsilon_{xc}}{\partial u} (4u) N_i(x) N_j(x)\, \mathrm{d}\mathbf{r} + \int_\Omega 2\epsilon_{xc} N_i(x) N_j(x)\, \mathrm{d}\mathbf{r} \tag{65}$$

$$\frac{\partial^2 \epsilon_{xc}}{\partial u^2} = \frac{\partial^2 \epsilon_x}{\partial u^2} + \frac{\partial^2 \epsilon_c}{\partial u^2} \tag{66}$$

$$\epsilon_x = -\frac{3}{4}\left(\frac{3}{\pi}\right)^{1/3} u^{2/3} \tag{67}$$

$$\implies \frac{\partial \epsilon_x}{\partial u} = -\frac{1}{2}\left(\frac{3}{\pi}\right)^{1/3} u^{-1/3} \tag{68}$$

$$\implies \frac{\partial^2 \epsilon_x}{\partial u^2} = \frac{1}{6}\left(\frac{3}{\pi}\right)^{1/3} u^{-4/3} \tag{69}$$

$$\epsilon_c(\rho) = \begin{cases} \dfrac{\gamma}{1 + \beta_1 \sqrt{r_s} + \beta_2 r_s} \\ \\ A \log r_s + B + C r_s \log r_s + D r_s \end{cases} \tag{70}$$

for $r_s \geq 1$ and $r_s < 1$ respectively.

$$\implies \frac{\partial \epsilon_c}{\partial u} = \begin{cases} \dfrac{-\gamma}{(1 + \beta_1 \sqrt{r_s} + \beta_2 r_s)^2} \left(\dfrac{\beta_1}{2\sqrt{r_s}} + \beta_2\right) \dfrac{\partial r_s}{\partial u} \\ \\ \left(\dfrac{A}{r_s} + C + C \log r_s + D\right) \left(\dfrac{\partial r_s}{\partial u}\right) \end{cases} \tag{71}$$

for $r_s \geq 1$ and $r_s < 1$ respectively and where

$$r_s = \left(\frac{3}{4\pi u^2}\right)^{1/3}$$

and

$$\frac{\partial r_s}{\partial u} = \left(\frac{3}{4\pi}\right)^{1/3}\left(\frac{-2}{3}\right)u^{-5/3}$$

$$\frac{\partial^2 \epsilon_c}{\partial u^2} = \tag{72}$$

$$\begin{cases} \frac{-\gamma}{(1+\beta_1\sqrt{r_s}+\beta_2 r_s)^2}(\frac{\beta_1}{2\sqrt{r_s}}+\beta_2)\frac{\partial^2 r_s}{\partial u^2} + \frac{-\gamma}{(1+\beta_1\sqrt{r_s}+\beta_2 r_s)^2}(\frac{-\beta_1}{4r_s^{1.5}})(\frac{\partial r_s}{\partial u})^2 + \frac{2\gamma}{(1+\beta_1\sqrt{r_s}+\beta_2 r_s)^3}(\frac{\beta_1}{2\sqrt{r_s}}+\beta_2)^2(\frac{\partial r_s}{\partial u})^2 \\ (\frac{A}{r_s}+C+C\log r_s+D)(\frac{\partial^2 r_s}{\partial u^2}) + (\frac{-A}{r_s^2}+\frac{C}{r_s})(\frac{\partial r_s}{\partial u})^2 \end{cases} \tag{73}$$

for $r_s \geq 1$ and $r_s < 1$ respectively.

$$\left.\frac{\partial L}{\partial u_i \partial u_j}\right|_{\phi_i = const.} = C_{ij}|_{u_0} = \frac{\partial T_s}{\partial u_i \partial u_j} + \frac{\partial E_{xc}}{\partial u_i \partial u_j} + \frac{\partial (ES)}{\partial u_i \partial u_j} + \frac{\partial (Lagrange)}{\partial u_i \partial u_j} \tag{74}$$

References

[1] P Hohenberg and W Kohn. Inhomogeneous Electron Gas. *Physical Review*, 136:B864–B871.

[2] W Kohn and L J Sham. Self-Consistent Equations Including Exchange and Correlation Effects. *Physical Review*, 140:A1133–A1138.

[3] N H March. The Thomas-Fermi approximation in quantum mechanics. *Advances in Physics*, 6.

[4] Elliott H. Lieb and Barry Simon. The Thomas-Fermi Theory of Atoms, Molecules and Solids. *Advances in Mathematics*, 23:22–116.

[5] Elliott H. Lieb. Thomas-fermi and related theories of atoms and molecules. *Reviews of Modern Physics*, 53.

[6] Nicholas Choly, Gang Lu, Weinan E, and Efthimios Kaxiras. Multiscale simulations in simple metals:âA density-functional-based methodology. *Physical Review B*, 71:1–16.

[7] V Gavini, K Bhattacharya, and M Ortiz. Quasi-continuum orbital-free density-functional theory: A route to multi-million atom non-periodic DFT calculation. *Journal of the Mechanics and Physics of Solids*, 55:697–718.

[8] Gang Lu, E. Tadmor, and Efthimios Kaxiras. From electrons to finite elements: A concurrent multiscale approach for metals. *Physical Review B*, 73:1–4.

[9] Qing Peng, Xu Zhang, Linda Hung, Emily Carter, and Gang Lu. Quantum simulation of materials at micron scales and beyond. *Physical Review B*, 78:1–14.

[10] Matt Fago, Robin Hayes, Emily Carter, and Michael Ortiz. Density-functional-theory-based local quasicontinuum method: Prediction of dislocation nucleation. *Physical Review B*, 70:1–4.

[11] W. Yang R. Parr. *Density-Functional Theory of Atoms and Molecules*. Oxford University Press, 1989.

[12] Vikram Gavini and Jaroslaw Knap. Non-periodic finite-element formulation of orbital-free density functional theory. 55:669–696, 2007.

[13] Jeng-Da Chai. *Orbital-Free Density Functional Theory of Atoms, Molecules and Solids*. PhD thesis, University of Maryland, College Park, 2005.

[14] W. Stich, E.K.U. Gross, P. Malzacher and R.M. Dreizler. Accurate Solution of the Thomas-Fermi-Dirac-Weizsäcker Variational Equations for the Case of Neutral Atoms and Positive Ions. *Z. Phys. A - Atoms and Nuclei*, 309:5–11.

[15] Jeng-Da Chai and John D Weeks. Orbital-Free Density Functional Theory: Kinetic Potentials and Ab-Initio Local Pseudopotentials. *Physical Review B*, 75:1–14.

[16] Yan Wang, Niranjan Govind, and Emily Carter. Orbital-free kinetic-energy functionals for the nearly free electron gas. *Physical Review B*, 58:13465–13471.

[17] Zhou Baojing, Vincent L. Ligneres, and Emily A. Carter. Improving the orbital-free density functional theory description of covalent materials. *The Journal of chemical physics*, 122:044103.1–044103.10.

[18] Jeng-Da Chai, Vincent L Ligneres, Gregory Ho, Emily A Carter, and John D Weeks. Orbital-Free Density Functional Theory: Linear Scaling Methods for Kinetic Potentials, and Applications to Solid Al and S. *Chemical Physics Letters*, 473:263–267.

[19] E Smargiassi and Pa Madden. Orbital-free kinetic-energy functionals for first-principles molecular dynamics. *Physical review. B, Condensed matter*, 49:5220–5226.

[20] Lin-Wang Wang and Michael P Teter. Kinetic-energy functional of the electron density. *Phys. Rev. B*, 45:13196–13220.

[21] Nicholas Choly and Efthimios Kaxiras. Kinetic energy density functionals for non-periodic systems. *Solid State Communications*, 121:12.

[22] Phani Motamarri, Mrinal Iyer, Jaroslaw Knap, and Vikram Ganvini. A numerical analysis of the finite-element discretization of orbital-free density functional theory. *Arxiv preprint arXiv:1110.1280*, pages 1–43.

[23] Junchao Xia and Emily A. Carter Chen Huang, Ilgyou Shin. Can orbital-free density functional theory simulate molecules? *Journal of Chemical Physics*, 136.

[24] Gregory S. Ho, Vincent L. Lignères, and Emily a. Carter. Introducing PROFESS: A new program for orbital-free density functional theory calculations. *Computer Physics Communications*, 179:839–854.

[25] N. Govind, Y. A. Wang, and E. A. Carter. Electronic-Structure Calculations By First-Principles Density-Based Embedding Of Explicitly Correlated Systems. *The Journal of chemical physics*, 110:7677–7688.

[26] Huajie Chen and Aihui Zhou. Orbital-Free Density Functional Theory for Molecular Structure Calculations. *Numerical Mathematics: Theory, Methods and Applications*, 1:1–28.

[27] Vikram Gavini. *Electronic Structure Calculations at Macroscopic Scales*. PhD thesis, California Institute of Technology, 2007.

[28] Vikram Gavini. Role of Macroscopic Deformations in Energetics of Vacancies in Aluminum. *Physical Review Letters*, 101:1–4.

[29] Vikram Gavini, Kaushik Bhattacharya, and Michael Ortiz. Vacancy clustering and prismatic dislocation loop formation in aluminum. *Physical Review B*, 76:1–4.

[30] Balachandran Radhakrishnan and Vikram Gavini. Effect of cell size on the energetics of vacancies in aluminum studied via orbital-free density functional theory. *Physical review B. Condensed matter and materials physics*, 82.

[31] William W Hager and Hongchao Zhang. A survey of nonlinear conjugate gradient methods. *Science*.

[32] Jonathan Richard Shewchuk. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain.

[33] P Bastian, M Blatt, A Dedner, Ch. Engwer, J Fahlke, C Gräser, R Klöfkorn, M Nolte, M Ohlberger, and O Sander. DUNE Web page, 2011.

[34] P Bastian, M Blatt, A Dedner, C Engwer, R Klöfkorn, R Kornhuber, M Ohlberger, and O Sander. A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part II: Implementation and Tests in DUNE. 82:121–138.

[35] P Bastian, M Blatt, A Dedner, C Engwer, R Klöfkorn, M Ohlberger, and O Sander. A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part {I}: Abstract Framework. 82:103–119.

[36] Markus Blatt and Peter Bastian. The Iterative Solver Template Library. In Bo Kragström, Erik Elmroth, Jack Dongarra, and Jerzy Waśniewski, editors, *Applied Parallel Computing. State of the Art in Scientific Computing*, volume 4699 of *Lecture Notes in Computer Science*, pages 666–675. Springer, 2007.

[37] Markus Blatt and Peter Bastian. On the Generic Parallelisation of Iterative Solvers for the Finite Element Method. *Int. J. Comput. Sci. Engrg.*, 4(1):56–69, 2008.

[38] A Dedner, R Klöfkorn, M Nolte, and M Ohlberger. DUNE-FEM Web page, 2011.

[39] A Dedner, R Klöfkorn, M Nolte, and M Ohlberger. A Generic Interface for Parallel and Adaptive Scientific Computing: Abstraction Principles and the DUNE-FEM Module. 90:165–196.