

Martin Vågseter Jakobsen

The Projector Augmented Wave method in DFT calculations for periodic clay structures

Master's thesis in Master of Science in Physics

Supervisor: Jon Andreas Støvneng

June 2022

Martin Vågseter Jakobsen

The Projector Augmented Wave method in DFT calculations for periodic clay structures

Master's thesis in Master of Science in Physics
Supervisor: Jon Andreas Støvneng
June 2022

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

Abstract

This thesis contains an extensive discussion regarding the plane wave approach to DFT calculations for periodic structures using pseudopotentials. The vdW-DF and DFT-D methods are suggested as options for the inclusion of dispersion effects in the system. The quasi-Newtonian BFGS algorithm is introduced as a possible method for geometry optimization. An introductory guide for the two open-source DFT simulation softwares Quantum ESPRESSO and GPAW is presented. The fluorohectorite clay mineral is investigated as a subject for periodic modeling in Quantum ESPRESSO.

Sammendrag

Denne avhandlingen inneholder en omfattende diskusjon angående plane bølger-tilnærmingen til DFT-metoden for periodiske strukturer ved bruk av pseudo-potensialer. Metodene vdW-DF og DFT-D er angitt som mulige alternativer for å inkludere dispersjonseffekter i systemet. Den kvasi-Newtonske BFGS algoritmen er introdusert som en mulig metode for geometrioptimering. En innledende guide til to open-source programpakker for DFT-simuleringer, Quantum ESPRESSO og GPAW, blir presentert. Leiremineralet fluorhektoritt blir undersøkt som et tema for periodisk modellering i Quantum ESPRESSO.

Preface

This thesis marks the end of my five year long journey of studying physics at the Norwegian University of Science and Technology (NTNU). It is a culmination of what I have learned within the realm of numerical quantum physics, from both my Bachelor's degree and over the two-year long programme of a Master of Science in Physics.

I want to give a special thanks to my supervisor, Prof. Jon Andreas Støvneng for the many engaging and fun conversations, both on the topic of this thesis and otherwise.

I want to thank my best friends Magnus and Javid for five fun physics years together, and for being supportive all throughout my Master's work. I would like to thank my former teachers Åsmund and Gøran for making physics a fun and interesting subject worth pursuing. I would like to thank Siv-Marie and Mustafa for helping me understand DFT calculations in the context of clay materials. I would also like to thank Friele for providing good coffee. Last but not least, I want to thank my family for all the love and support.

Contents

1	Introduction	1
1.1	Clay and Fluorohectorite	1
1.2	Software Considerations	2
1.3	Structure of the Report	2
2	Density Functional Theory	5
2.1	The Many-body Schrödinger Equation	5
2.2	The Hartree-Fock Method	6
2.3	The Kohn-Sham Equations	9
2.4	Exchange-Correlation Functionals	12
2.5	Dispersion Correction Methods	13
2.5.1	Non-localized van der Waals Functionals	13
2.5.2	Grimme's DFT-D Method	14
2.6	Periodicity in DFT	15
2.6.1	Reciprocal Space and Bloch's Theorem	15
2.6.2	Sampling of k-points	16
2.7	Geometry Optimization	17
2.7.1	Potential Energy Surfaces	17
2.7.2	The Hellmann-Feynman Theorem	19
2.7.3	Relaxation Scheme	20
3	Computational Approach	21

3.1	Plane Wave Method and Energy Cutoff	21
3.1.1	Alternative Basis Sets	22
3.2	Pseudopotentials	23
3.3	The BFGS Algorithm	24
3.4	Parameters and Convergence Factors	25
3.4.1	Occupation Number Smearing	26
4	Software Choices	29
4.1	Quantum ESPRESSO	29
4.1.1	Pseudopotential Resources	30
4.1.2	Input Script Structure	30
4.1.3	Parallel Runs	33
4.1.4	Using Shell Scripts	34
4.2	GPAW	36
4.2.1	Script Structure	36
4.2.2	Parallel Runs	39
4.3	Clay Calculations	39
5	Conclusion	43
5.1	Concluding Remarks	43
5.2	Future Work	43
	References	45

Abbreviations

DFT | Density Functional Theory

SCF | Self-Consistent Field

XC | Exchange-Correlation

LDA | Local Density Approximation

GGA | Generalized Gradient Approximation

PBE | Perdew-Burke-Ernzerhof

vdW-DF | van der Waals Density Functional

BZ | Brillouin Zone

PES | Potential Energy Surface

TS | Transition State

LCAO | Linear Combination of Atomic Orbitals

GTO | Gaussian Type Orbitals

STO | Slater Type Orbitals

USPP | Ultra-Soft Pseudopotential

PAW | Projector Augmented Wave

BFGS | Broyden-Fletcher-Goldfarb-Shanno

QE | Quantum ESPRESSO

BASH | Bourne-Again Shell

ASE | Atomic Simulation Environment

CHAPTER

1

INTRODUCTION

In the world of quantum chemistry and materials science we are working at the scale of atoms, where the nature is bound by the Schrödinger equation. When considering complex many-body systems, this equation can be hard to solve. Density functional theory (DFT) aims to make the process of solving this problem easier, by making use of numerical computations in a variational approach. Implementing this method into simulation software can lead to a new understanding of materials, once modeled into a distribution of electrons and nuclei.

1.1 Clay and Fluorohectorite

Clay minerals are abundant, non-toxic, cheap and reusable [1]. They have large surface areas, which makes them great candidates for applications where we want to adsorb other molecules on the surface. Due to the layered structure in these materials, inserting different molecules between layers can lead to adsorption that results in strong binding energies. Today, studying the possibilities for carbon storage in clay has become an active field of research.

In Fig. 1.1 we see the typical form of a smectite, with its center-most octahedral layer surrounded by tetrahedral layers on either side. We will in this thesis consider a specific type of clay material in the smectite group called *fluorohectorite*. The formula for fluorohectorite is written as



where X denotes some form of an intercalated ion between clay layers. From this expression we see that whenever $y = 0$ we have the pure form of fluorohectorite.

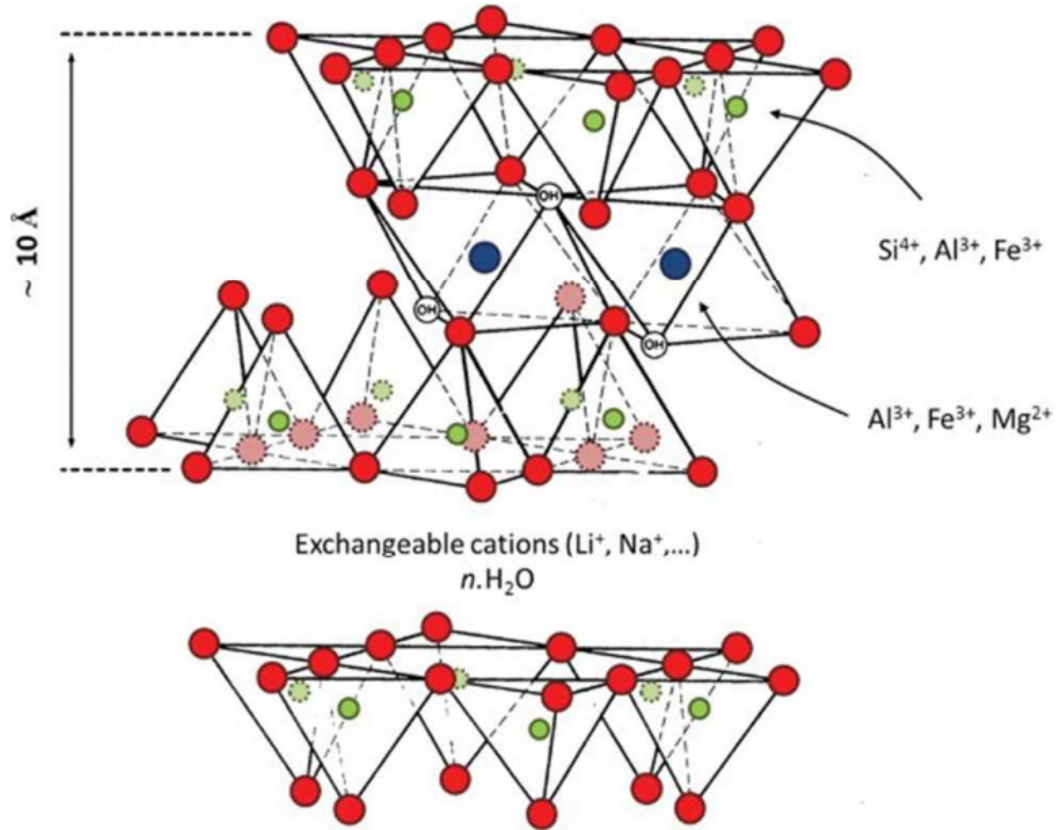


Figure 1.1: The typical structure of a smectite [2]. Each layer of clay sheets contains a stack of two tetrahedral structures and an octahedral layer in the middle.

1.2 Software Considerations

Due to an expired license for a previously used DFT software at NTNU, we have a motivating factor in finding a new software that is versatile, user-friendly and efficient. In this project, we will consider two different open-source DFT suites, both relying on sophisticated methods that will be explained throughout this thesis.

1.3 Structure of the Report

In this thesis we will mainly be discussing different methods that can be applied for DFT calculations. We will start by looking at the many-body Schrödinger equation, and go historically through the different approaches to simplifying the problem. In chapter 3 we will consider how DFT methods can be implemented

in combination with geometry optimization. In chapter 4 we will compare DFT softwares and test some clay calculations, and in chapter 5 we will finish with some concluding remarks.

CHAPTER

2

DENSITY FUNCTIONAL THEORY

In scientific fields such as physical chemistry and materials physics, the objective is often to investigate properties of molecular structures at the quantum level. To calculate these properties, we require a solution to the Schrödinger equation. The task of finding such a solution proves non-trivial for many-body systems, as the computational cost typically scales exponentially with the complexity of the system. Among the many proposed methods to tackle this problem, DFT tends to strike a good balance between computational time and accuracy in the results, and it will be our method of interest for this project. In this chapter we will discuss the DFT method in detail, starting with the history and fundamental principles behind it.

2.1 The Many-body Schrödinger Equation

When we are studying many-body systems, we need to consider multiple nuclei simultaneously. This translates to a more complex Hamiltonian in the Schrödinger equation for the system, and calls for simplifications to find a solution faster.

The main simplification that is utilized is known as the *Born-Oppenheimer approximation* [3]. The idea is here to separate the dynamics of the nuclei from the dynamics of electrons in the system, given that the nuclei are much heavier than the electrons. When solving for the motion of the electrons we may consider each nuclei I as having a fixed position \mathbf{R}_I in space. Given a configuration of M nuclei we can form an energy landscape $E(\mathbf{R}_1, \dots, \mathbf{R}_M)$ that can be minimized, which will be discussed further in later sections. The combination of Coulomb potentials

from all nuclei will act as an external electrostatic field V for the electrons in the system. The time-independent Schrödinger equation for the electrons can be written as

$$\hat{H}\psi = \left[-\frac{\hbar^2}{2m} \sum_{i=1}^N \nabla_i^2 + \sum_{i=1}^N V(\mathbf{r}_i) + \sum_{i=1}^N \sum_{j<i}^N U(\mathbf{r}_i, \mathbf{r}_j) \right] \psi = E\psi, \quad (2.1)$$

where \hat{H} is the Hamiltonian and E denotes the ground state energy. The wave function is here referring to the product of all individual electron wave functions, which reads $\psi = \psi_1(\mathbf{r}_1)\psi_2(\mathbf{r}_2)\dots\psi_N(\mathbf{r}_N)$ for a given number of electrons N . The three terms inside the brackets represent the kinetic energy of the electrons, the energy due to electron-nuclei interactions with the external field V , and the energy from all electron-electron interactions respectively. We note that this form of the equation is non-relativistic and does not account for electronic spin effects.

2.2 The Hartree-Fock Method

One of the early approaches to solving the many-body Schrödinger equation was proposed by Douglas R. Hartree in 1928, where he reviewed the wave mechanics of a system where all electron-electron Coulomb effects were neglected [4]. Following this idea we can rewrite the Hamiltonian in Eq. 2.1 as

$$\hat{H} = \sum_{i=1}^N \left[-\frac{\hbar^2}{2m} \nabla_i^2 + V(\mathbf{r}_i) \right] = \sum_{i=1}^N \hat{h}_i, \quad (2.2)$$

where \hat{h} denotes the Hamilton operator for a single electron. The time-independent Schrödinger equation for a single electron can be expressed as $\hat{h}\chi = E\chi$, where χ describes some eigenfunction. In order to include spin we introduce the concept of a *spin-orbital*, denoted $\chi_j(\mathbf{x}_i)$ [5, p. 20]. The indices $j = 1, 2, \dots$ are here referring to energy levels E_j associated with each orbital, while \mathbf{x}_i is a vector including both the position values and spin state of the given electron i . In Hartree's original procedure the total electron wave function of the system was simply expressed by multiplying together spin-orbitals, known as the *Hartree product*, which reads

$$\psi(\mathbf{x}_1, \dots, \mathbf{x}_N) = \chi_{j1}(\mathbf{x}_1)\chi_{j2}(\mathbf{x}_2), \dots, \chi_{jN}(\mathbf{x}_N). \quad (2.3)$$

The problem with this approach is that interchanging two electrons yields the same wave function, thus violating the fundamental *anti-symmetry principle* for

fermions [6, p. 190]. In order to rectify this issue, John C. Slater introduced what is now known as the *Slater determinant* [7]. The wave function for N electrons is here expressed as

$$\psi(\mathbf{x}_1, \dots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \chi_1(\mathbf{x}_1) & \chi_2(\mathbf{x}_1) & \dots & \chi_N(\mathbf{x}_1) \\ \chi_1(\mathbf{x}_2) & \chi_2(\mathbf{x}_2) & \dots & \chi_N(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1(\mathbf{x}_N) & \chi_2(\mathbf{x}_N) & \dots & \chi_N(\mathbf{x}_N) \end{vmatrix}, \quad (2.4)$$

where the factor in front of the determinant simply acts to normalize the wave function. This form of the wave function not only satisfies the anti-symmetry principle, but also the *Pauli exclusion principle*, which forbids two electrons from occupying the same quantum state [8]. We can verify both of these properties for a simple two-electron system with the wave function

$$\psi(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{\sqrt{2}} [\chi_j(\mathbf{x}_1)\chi_k(\mathbf{x}_2) - \chi_j(\mathbf{x}_2)\chi_k(\mathbf{x}_1)], \quad (2.5)$$

where the two orbital states are denoted by indices j and k . If we swap the electrons 1 and 2 we see anti-symmetry in that $\psi(\mathbf{x}_1, \mathbf{x}_2) = -\psi(\mathbf{x}_2, \mathbf{x}_1)$, and if we try to occupy the same orbital state by setting $j = k$ we find that the wave function goes to zero.

With the inclusion of the Slater determinant and the Pauli exclusion principle for the N -electron wave function, we now seek to find an approximation for the spin-orbitals. The Schrödinger equation for one electron can be expressed as

$$\left[-\frac{\hbar^2}{2m} \nabla_j^2 + V(\mathbf{r}) + V_H(\mathbf{r}) + V_X(\mathbf{r}) \right] \chi_j(\mathbf{x}) = E_j \chi_j(\mathbf{x}), \quad (2.6)$$

where $\chi_j(\mathbf{x})$ is the spin-orbital for the electron, and $V_X(\mathbf{r})$ denotes the exchange interaction due to the Pauli principle. This equation is similar to Eq. 2.1 in that it assumes a combined Coulomb potential $V(\mathbf{r})$ from stationary nuclei around it. Similarly, the electron-electron interactions are here reintroduced as an average Coulomb potential $V_H(\mathbf{r})$ from all surrounding electrons. This potential, which is often referred to as the *Hartree potential*, is defined as

$$V_H(\mathbf{r}) = e^2 \int \frac{n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d^3r', \quad (2.7)$$

where $n(\mathbf{r})$ denotes the electron density of the system. We may also express the electron density using the wave functions themselves. In the standard notation of

single-electron wave functions ψ_i we now write

$$n(\mathbf{r}) = 2 \sum_i \psi_i^*(\mathbf{r})\psi_i(\mathbf{r}), \quad (2.8)$$

where the factor 2 accounts for dual spin-states per electron. By inserting back into Eq. 2.7 we now note that solving Eq. 2.6 has become a non-linear procedure. From the *variational principle* we know that any expectation value of the Hamiltonian will be larger or equal to the ground state energy of a system [9, p. 8]. With this in mind, we can approach the problem iteratively, whereby starting with some suggested ansatz of χ_j we can calculate $n(\mathbf{r})$, leading to new estimates of χ_j and E_j to repeat the procedure with. We then minimize with respect to E_j to find the ground state. This approach is often referred to as the *self-consistent field* method (SCF), and the flowchart shown in Fig. 2.1 illustrates how the process eventually leads to a satisfactory solution given some predetermined convergence criteria.

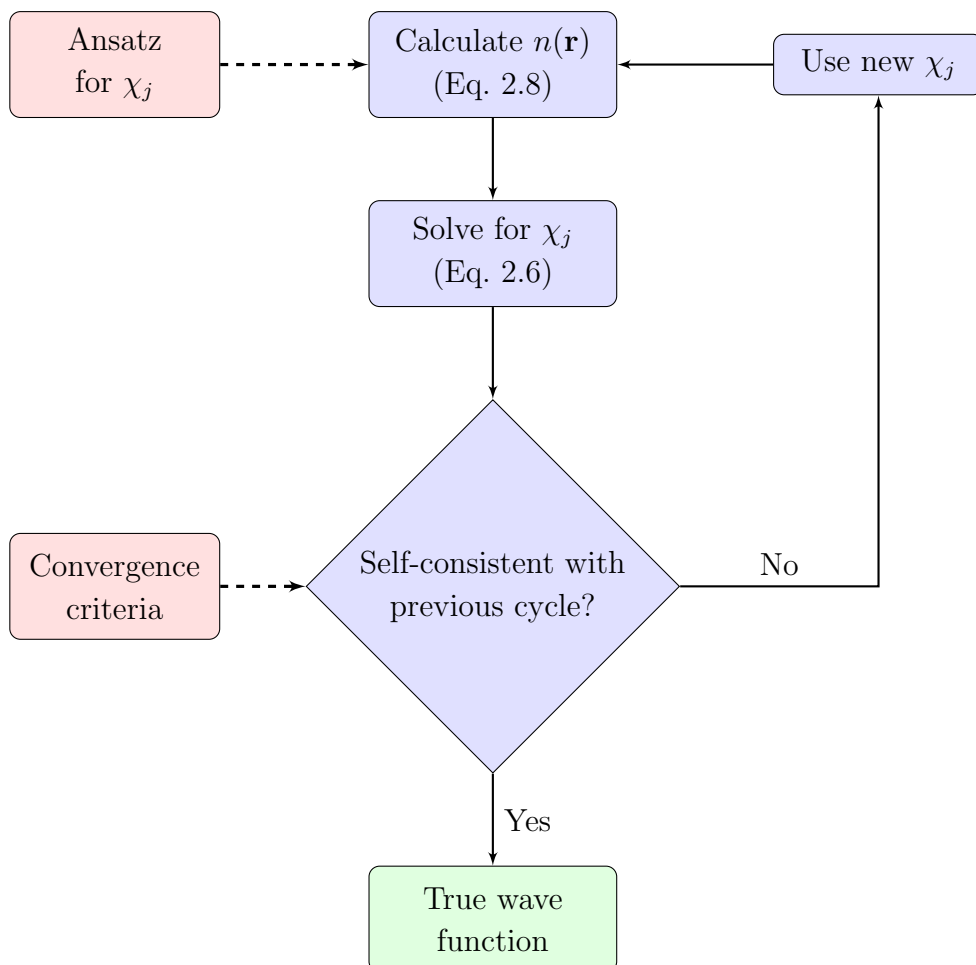


Figure 2.1: The Hartree-Fock calculation scheme. Finds the self-consistent true wave function based on input quantities from the red boxes.

As discussed above, we need to start the calculation with an ansatz of what the spin-orbitals could look like. We can express a spin-orbital $\chi_j(\mathbf{x})$ as

$$\chi_j(\mathbf{x}) = \sum_{i=1}^K \alpha_{ji} \eta_i(\mathbf{x}), \quad (2.9)$$

where the finite selection of expansion coefficients α_{ji} and basis functions $\eta_i(\mathbf{x})$ approximate the exact shape of the orbital. In the Hartree-Fock method we choose our *basis set* $\{\eta_i\}_{i=1}^K$ before any calculations, which means we only need to find new expansion coefficients throughout the SCF cycles to describe the orbitals. Since the Hartree-Fock method is *ab initio*, i.e. derived from fundamental principles within physics, we can technically be as accurate as we want, given a large enough basis set and strict enough convergence criteria.

We note that the Hartree potential in Eq. 2.7 only accounts for a portion of the electron-electron interactions, and that any *correlation energy* between electrons is left out of the Hartree-Fock method.

2.3 The Kohn-Sham Equations

The DFT method was first founded by Walter Kohn and Pierre Hohenberg in the 1960s, where they formulated two mathematical theorems related to the connection between electrons in an external potential and the resulting ground state energy of the system [10]. The first theorem can be transcribed as follows: *The ground state energy of a system from Schrödinger's equation is a unique functional of the electron density* [5, p. 11]. This means that for every given electron density $n(\mathbf{r})$ there exists a corresponding energy functional $E[n(\mathbf{r})]$ and vice versa. A functional is defined as a real-valued function that takes another function as its argument. To give an example we can consider the equation

$$F[f] = \int_{-1}^1 f(x) dx, \quad (2.10)$$

where F is a scalar-valued functional of the function $f(x)$. The advantage of solving the Schrödinger equation using a functional becomes clear when we consider the dimensions of the problem; solving for N electrons usually requires calculations for $3N$ variables, but by introducing a functional we can reduce this to just 3 variables of the electron density $n(\mathbf{r})$.

The first theorem tells us that an energy functional must exist, but it does not help us define what it actually looks like. The second theorem proposed by Kohn and Hohenberg states the following: *The electron density that minimizes the energy of the overall functional is the true electron density corresponding to the full solution of the Schrödinger equation* [5, p. 11]. This suggests that we can use the variational principle to find the true ground state electron density $n_0(\mathbf{r})$ once we have an adequate description of the energy functional.

In the spirit of Hartree-Fock, we will now discuss which energy contributions should go into the Schrödinger equation. We already know the form of some of the energy contributions in the Hartree-Fock approach, namely the nuclei-electron potential V , the Hartree potential V_H , and the kinetic energy term which we will denote T_e . The energy from ionic nuclei-nuclei interactions E_{ion} that was previously omitted is now also included. We can write the total energy in functional form as

$$E[n] = T_e[n] + V[n] + V_H[n] + E_{\text{ion}} + E_{\text{XC}}[n] = E_{\text{known}}[n] + E_{\text{XC}}[n], \quad (2.11)$$

where in addition to the known contributions we have the functional $E_{\text{XC}}[n(\mathbf{r})]$ describing the *exchange-correlation* energy. We remember from the definition in Eq. 2.8 that the electron density can be described using the sum of all one-electron wave functions ψ_i . This means that the energy functional can also be expressed by the set of these wave functions as $E[\{\psi_i\}]$. By considering a single-electron wave function ψ_i we can readily write out the known contributions as

$$\begin{aligned} E_{\text{known}}[\psi_i] = & -\frac{\hbar^2}{2m} \sum_i \int \psi_i^* \nabla^2 \psi_i \, d^3r + \int V(\mathbf{r})n(\mathbf{r}) \, d^3r \\ & + \frac{e^2}{2} \int \int \frac{n(\mathbf{r})n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \, d^3r \, d^3r' + E_{\text{ion}}, \end{aligned} \quad (2.12)$$

in the order corresponding to $T_e + V + V_H + E_{\text{ion}}$ in Eq. 2.11.

Kohn proved through his collaboration with with Lu J. Sham that it was possible to arrive at the right electron density by solving a set of one-electron equations

$$\left[-\frac{\hbar^2}{2m} \nabla_i^2 + V(\mathbf{r}) + V_H(\mathbf{r}) + V_{\text{XC}}(\mathbf{r}) \right] \psi_i(\mathbf{r}) = \epsilon_i \psi_i(\mathbf{r}), \quad (2.13)$$

today fittingly known as the *Kohn-Sham equations* [5, p. 12]. We see an almost identical structure to that of the many-body Schrödinger equation in Section 2.1,

but notice the lack of summations. The last term inside the brackets will hereby be referred to as the exchange-correlation potential, and it is defined as

$$V_{\text{XC}} = \frac{\delta E_{\text{XC}}(\mathbf{r})}{\delta n(\mathbf{r})}, \quad (2.14)$$

denoting the *functional derivative* of the exchange-correlation energy $E_{\text{XC}}(\mathbf{r})$ with respect to $n(\mathbf{r})$. The sum of the potentials in Eq. 2.13 is often referred to as the effective potential

$$V_{\text{eff}}(\mathbf{r}) = V(\mathbf{r}) + V_H(\mathbf{r}) + V_{\text{XC}}(\mathbf{r}). \quad (2.15)$$

The flowchart in Fig. 2.2 shows the procedure of solving Kohn-Sham equations. It is similar to the SCF method presented in Fig. 2.1, but instead of starting with an ansatz for spin-orbitals χ_j , we use the electron density $n(\mathbf{r})$ directly.

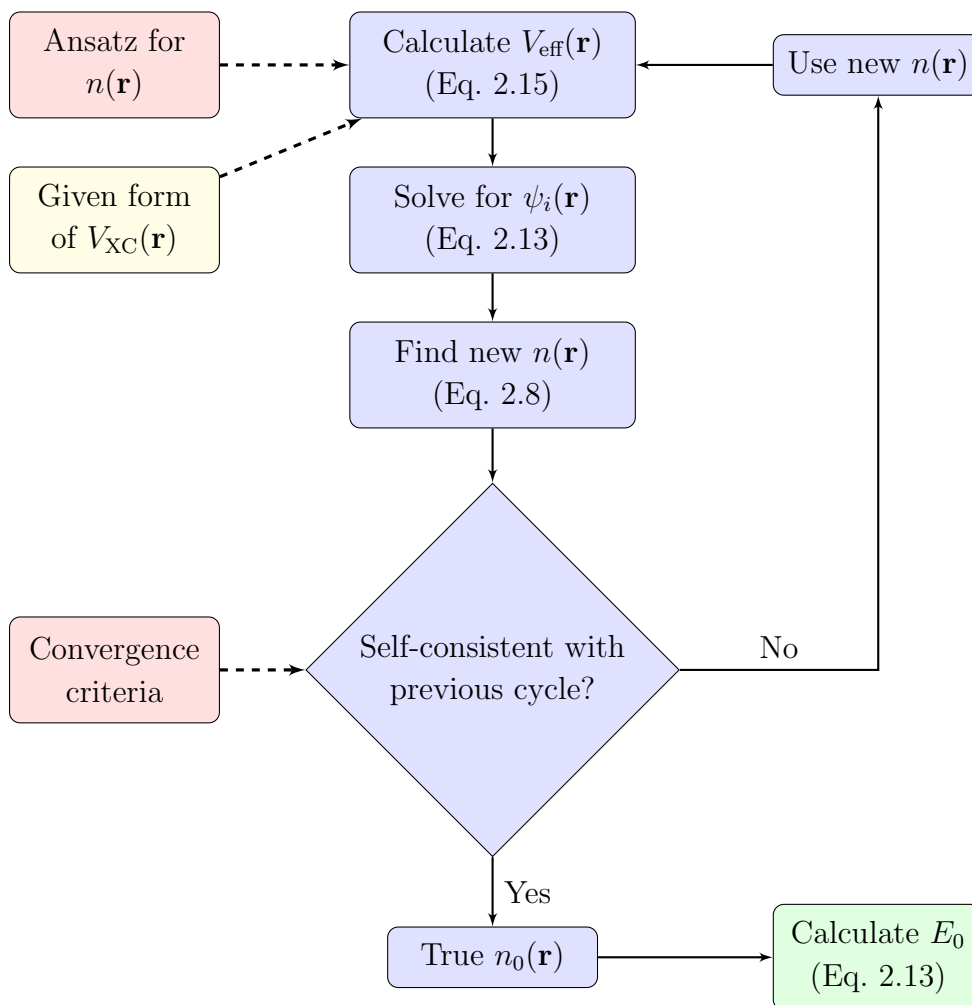


Figure 2.2: The Kohn-Sham equation calculation scheme. Calculates the ground state energy E_0 based on input quantities from the red boxes. The given exchange-correlation potential in the yellow box always applies in the calculation of $V_{\text{eff}}(\mathbf{r})$.

2.4 Exchange-Correlation Functionals

The difficult part of solving the Kohn-Sham equations is to define what the exchange-correlation energy contribution actually looks like. In theory, there should exist a functional $E_{\text{XC}}[n(\mathbf{r})]$ that contains all information about non-Coulombian electron-electron interaction energy contributions, but as of today it is yet to be found. Instead, we can approximate the effect of this functional by looking at situations where it can be derived exactly.

In the case of a uniform electron gas we can set the electron density $n(\mathbf{r})$ to be constant at all points in space. If we use this approximation at points \mathbf{r} in our system where $n(\mathbf{r})$ is given, we can then define the exchange-correlation potential locally as

$$V_{\text{XC}}(\mathbf{r}) = V_{\text{XC}}^{\text{gas}}[n(\mathbf{r})], \quad (2.16)$$

where $V_{\text{XC}}^{\text{gas}}$ is the exchange-correlation potential for a uniform electron gas given the value of $n(\mathbf{r})$. This simple approach to defining the exchange-correlation functional is known as the *local density approximation* (LDA)[11].

A more sophisticated approach that builds on the LDA functional is described in the *generalized gradient approximation* (GGA), which in addition to the local electron density takes into account the local gradient of $n(\mathbf{r})$ as

$$V_{\text{XC}}^{\text{GGA}}(\mathbf{r}) = V_{\text{XC}}[n(\mathbf{r}), \nabla n(\mathbf{r})]. \quad (2.17)$$

Due to the inclusion of more physical information about $n(\mathbf{r})$, GGA functionals are generally preferred over LDA. However, this approach is more computationally intensive, and it does not guarantee more accurate results. There are many different ways of incorporating the local electron density gradient into the approximation, and thus many variants of the GGA functional have been suggested. One of the most widely used GGA variants is the PBE functional proposed by Perdew, Becke and Ernzerhof in 1996 [12], which employs a non-empirical approach. The PBE functional has later been revised and made more accurate by Zhang and Yang [13], with the proposed name revPBE. There are functionals that include even more physical information, but for the purposes of this report we will mostly be focusing the GGA-PBE approach.

2.5 Dispersion Correction Methods

One of the shortcomings of the exchange-correlation functionals discussed so far is the disregard for weak forces, such as the van der Waals interactions between atoms. These dispersion interactions are inherently long-range, and they are thus not considered in the semi-localized approaches such as for the GGA and LDA functionals. If we want to take into account the effect of dispersion interactions we have two options; we either aim to construct a non-localized functional or we try to apply an empirical dispersion correction to the Kohn-Sham energy.

2.5.1 Non-localized van der Waals Functionals

The concept of a non-local van der Waals density functional (vdw-DF) has been under development since the early 2000s, carried forward by the efforts of Dion, Rydberg and others [14], [15]. The idea is in general terms to construct a correlation functional $E_c[n]$ which can then be coupled with the exchange part of a semi-local GGA functional such as revPBE. The early steps of defining $E_c[n]$ involve dividing up the total correlation functional as

$$E_c = E_c^0[n] + E_c^{\text{nl}}[n], \quad (2.18)$$

where the first term is approximated with a localized LDA approach, and the second term encapsulates the non-local correlation energy. The simplest form of the energy in the latter term is here expressed as

$$E_c^{\text{nl}} = \frac{1}{2} \int d^3r d^3r' n(\mathbf{r})\phi(\mathbf{r}, \mathbf{r}')n(\mathbf{r}'), \quad (2.19)$$

where $\phi(\mathbf{r}, \mathbf{r}')$ is some function dependent on difference vectors $\mathbf{r} - \mathbf{r}'$ and the local values of n around the respective coordinates. The process of finding a suitable expression for ϕ is quite complicated, and it involves the use of various models and parameters [15]. These details are beyond the scope of what we will discuss here, but we now have a general understanding of the concepts in the non-localized functional approach. Similarly as for the PBE functional, the vdW-DF approach has been updated with new parameter values in the vdW-DF2 functional.

2.5.2 Grimme's DFT-D Method

The dispersion correction approach has also been explored since the mid-2000s in the development of Stefan Grimme's semi-empirical DFT-D method [16], [17]. His approach has later been refined in the DFT-D3 method [18], which we will now take a closer look at. The contributions to the total DFT-D3 energy is expressed with a dispersion correction as

$$E_{\text{DFT-D3}} = E_{\text{KS-DFT}} + E_{\text{disp}}. \quad (2.20)$$

The first term on the right side is here referring to the self-consistent Kohn-Sham energy from a chosen density functional, while the dispersion correction term is further modeled as

$$E_{\text{disp}} = E^{(2)} + E^{(3)}. \quad (2.21)$$

The dispersion energy is here split up into a two-body term and a three-body term, where the most important two-body term reads as

$$E^{(2)} = \sum_{AB} \sum_{n=6,8,10,\dots} s_n \frac{C_n^{AB}}{r_{AB}^n} f_{\text{d},n}(r_{AB}). \quad (2.22)$$

The atoms are here pairwise denoted as AB , where r_{AB} is the distance between the two atoms. We note that this expression includes a damping factor $f_{\text{d},n}(r_{AB})$, a scaling factor s_n , and an averaged dispersion coefficient C_n^{AB} for a given atom pair. The dispersion coefficients are computed ab initio from the Casimir-Polder formula, while the damping factor is modeled from a proposed variant by Chai and Head-Gordon [19].

Using the same damping function as for the two-body term, we now express the three-body term as

$$E^{(3)} = \sum_{ABC} f_{\text{d},(3)}(\bar{r}_{ABC}) E^{ABC}, \quad (2.23)$$

where ABC denotes the triplet of atoms that interact with each other, and \bar{r}_{ABC} the averaged radii between them. The factor E^{ABC} is here a triple dipole term derived from third-order perturbation theory.

It is worth noting that there have been other efforts to introduce dispersion corrections, examples being the Exchange-hole dipole-moment model [20], the

Tkatchenko-Scheffler method [21], and the many-body dispersion (MDB) framework [22]. However, due to it being easier to implement for our software choices, we will only be concerned with Grimme's DFT-D3 method in this thesis.

2.6 Periodicity in DFT

In this section we will investigate how modeling our systems as periodic structures can lead to simplifications for more efficient DFT calculations. We will consider both the theory behind it, and how it can be implemented computationally.

2.6.1 Reciprocal Space and Bloch's Theorem

To further simplify the process of describing the dynamics of electrons in a many-body system, we can make use of the theory behind crystalline structures in solid state physics. The periodicity of crystals can be explained as a repeating lattice with corresponding lattice vectors \mathbf{a}_1 , \mathbf{a}_2 and \mathbf{a}_3 that define the unit cell in real space. This can also be expressed by the atomic coordinate $\mathbf{R} = n_1\mathbf{a}_1 + n_2\mathbf{a}_2 + n_3\mathbf{a}_3$, where the integers n_i are free to be chosen arbitrarily.

Fourier transforming the unit cell lattice structure gives us the corresponding representation of the crystal in reciprocal space, often referred to as \mathbf{k} -space. We consider the equivalent lattice vectors in \mathbf{k} -space as $\mathbf{G} = m_1\mathbf{b}_1 + m_2\mathbf{b}_2 + m_3\mathbf{b}_3$. The connection between real space and \mathbf{k} -space unit cell representations is given by the plane wave relation $e^{i\mathbf{G}\cdot\mathbf{R}} = 1$. This is the same as writing $\mathbf{a}_i \cdot \mathbf{b}_j = 2\pi\delta_{ij}$, where δ_{ij} is a Kronecker delta [23, p. 29]. The reciprocal lattice vectors may now be expressed by the real space vectors as

$$\mathbf{b}_1 = 2\pi \frac{\mathbf{a}_2 \times \mathbf{a}_3}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)}, \quad \mathbf{b}_2 = 2\pi \frac{\mathbf{a}_3 \times \mathbf{a}_1}{\mathbf{a}_2 \cdot (\mathbf{a}_3 \times \mathbf{a}_1)}, \quad \mathbf{b}_3 = 2\pi \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\mathbf{a}_3 \cdot (\mathbf{a}_1 \times \mathbf{a}_2)}. \quad (2.24)$$

The reciprocal unit cell that these lattice vectors define in \mathbf{k} -space is known as the first *Brillouin zone*, which plays a central role in band theory. We note that the relation between the volumes of the Brillouin zone (BZ) and the real space unit cell can be found as

$$V_{\text{BZ}} = \frac{(2\pi)^3}{V_{\text{cell}}},$$

indicating inverse proportionality in length between the two spaces.

To see the importance of periodicity in the context of describing electrons, we apply the solid state physics theory to the Schrödinger equation. The result of this is known as *Bloch's theorem*, which states that wave functions of electrons in a periodic potential must be of the form

$$\psi_k(\mathbf{r}) = e^{i\mathbf{k}\cdot\mathbf{r}}u_k(\mathbf{r}), \quad (2.25)$$

where u_k is a function with the same periodicity as the potential [6, p. 214]. The periodic form of such *Bloch states* implies that the first BZ should contain all physical information about the material. An illustration of the construction of a Bloch state is shown in Fig. 2.3.

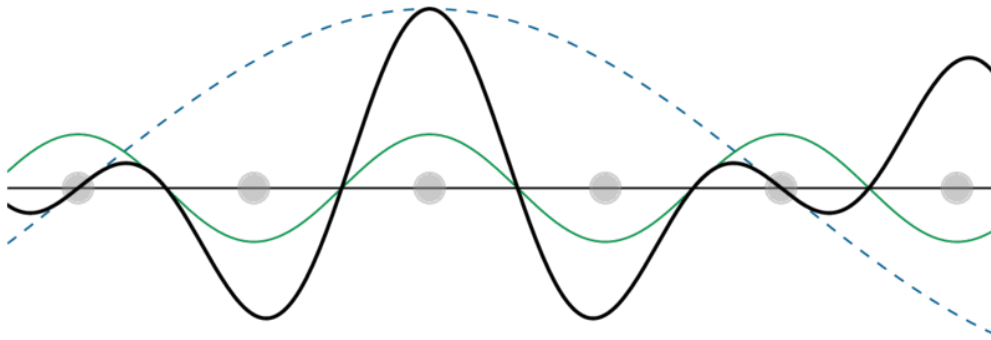


Figure 2.3: The real part of a one-dimensional Bloch function [24]. The Bloch state (black line) is a product of a plane wave (dashed blue line) and a periodic potential u_k (green line). The distribution of atoms is indicated by grey circles.

2.6.2 Sampling of k -points

DFT calculations for periodic systems often involve functions of the form

$$\bar{g} = \frac{V_{\text{cell}}}{(2\pi)^3} \int_{\text{BZ}} g(\mathbf{k})d\mathbf{k}, \quad (2.26)$$

where we need to integrate over the Brillouin zone in reciprocal space [5, p. 53]. In order to evaluate this integral efficiently, we discretize the space into a grid of k -points. Both the accuracy and computational cost of a given DFT calculation is proportional to the number of k -points in our grid. This means that we must allow some error if we want to achieve results in a reasonable amount of time. We can

use a convergence test to decide the optimal size of our k -points grid; we say that we are satisfied whenever increasing the number of k -points leaves the calculated energy unchanged up to some predetermined tolerance error.

The sampling of k -points can be done in different ways. If we want to distribute the k -points evenly in reciprocal space, the method proposed by Monkhorst and Pack makes sense [25]. Their approach revolves around specifying the number of k -points along each reciprocal lattice vector as $M_1 \times M_2 \times M_3$, where we adjust the indices based on the length of each lattice vector. We mentioned earlier that there is an inverse proportionality in length between real space and reciprocal space, which means we can decide on the Monkhorst-Pack grid based on our real space unit cell; if we consider two real space lattice vectors where $|\mathbf{a}_1| > |\mathbf{a}_2|$, we can then select indices $M_1 < M_2$ to get an even k -points distribution in reciprocal space.

2.7 Geometry Optimization

When we want to calculate the potential energy of a molecular system, we need to consider more than just the electrons. So far we have considered the nuclei of a many-body system to be stationary in a given structure of positions $\mathbf{R}_1, \dots, \mathbf{R}_M$. However, in order to reach an overall stable state, we need to optimize these coordinates in such a way that we can minimize the total energy of the system.

2.7.1 Potential Energy Surfaces

The concept of a *potential energy surface* (PES) refers to the idea that for every structural configuration of a molecule, we have a corresponding energy landscape that is unique to this geometry. Finding the local minima of the PES corresponds to a geometry optimization for the molecular structure.

The dimensionality of the PES for a system of M atoms is given as $3M - 6$, where we have subtracted the degrees of freedom related to translation and rotation. From this, we can already note that the process of finding energy minima is a non-trivial task. As a first step, knowing the curvature of the PES can help to make the process easier. We describe the curvature of a system with M atoms by

considering the second derivative of the potential energy at each individual atomic coordinate. The resulting *Hessian matrix* \mathbf{H} is written as

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 E}{\partial x_1^2} & \frac{\partial^2 E}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 E}{\partial x_1 \partial x_{3M}} \\ \frac{\partial^2 E}{\partial x_2 \partial x_1} & \frac{\partial^2 E}{\partial x_2^2} & \cdots & \frac{\partial^2 E}{\partial x_2 \partial x_{3M}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial x_{3M} \partial x_1} & \frac{\partial^2 E}{\partial x_{3M} \partial x_2} & \cdots & \frac{\partial^2 E}{\partial x_{3M}^2} \end{bmatrix}, \quad (2.27)$$

where all $3M$ positional coordinates are taken into account. Whenever all non-zero eigenvalues λ_i of the Hessian matrix are positive, we know that the structure is close to a local minima, while cases with negative eigenvalues indicate local maxima [26]. The physical interpretation of the eigenvalues λ_i is that they help to define the characteristic vibrational frequencies

$$\nu_i = \frac{1}{2\pi} \sqrt{\frac{\lambda_i}{m}}, \quad (2.28)$$

known as the *normal modes* of the system. We recognize that depending on the system we are looking at, the behavior of normal modes can be an interesting property to consider, but we will not be elaborating any further on this topic in the remainder of this thesis.

When discussing strategies for geometry optimization we are often interested in the *transition state* (TS) between two local minima. Transition states can be categorized as having a Hessian matrix with exactly one negative eigenvalue, acting as a maxima along an otherwise minimized energy path. The illustration in Fig. 2.4 shows how two minima \mathbf{A} and \mathbf{B} are separated by a transition state \mathbf{TS} acting as an energy barrier along the *minimum energy path* between the two minima. We define the *activation energy* E_a required to cross this barrier as

$$E_a = E(\mathbf{TS}) - E(\mathbf{A}), \quad (2.29)$$

assuming that the path starts at \mathbf{A} and ends at \mathbf{B} . The concept of an activation energy over a transition state is most often associated with chemical reactions, where the \mathbf{A} refers to the reactants and \mathbf{B} refers to the products. However, in our case we are only interested in understanding the PES concepts in the context of geometry optimizations for molecular structures.

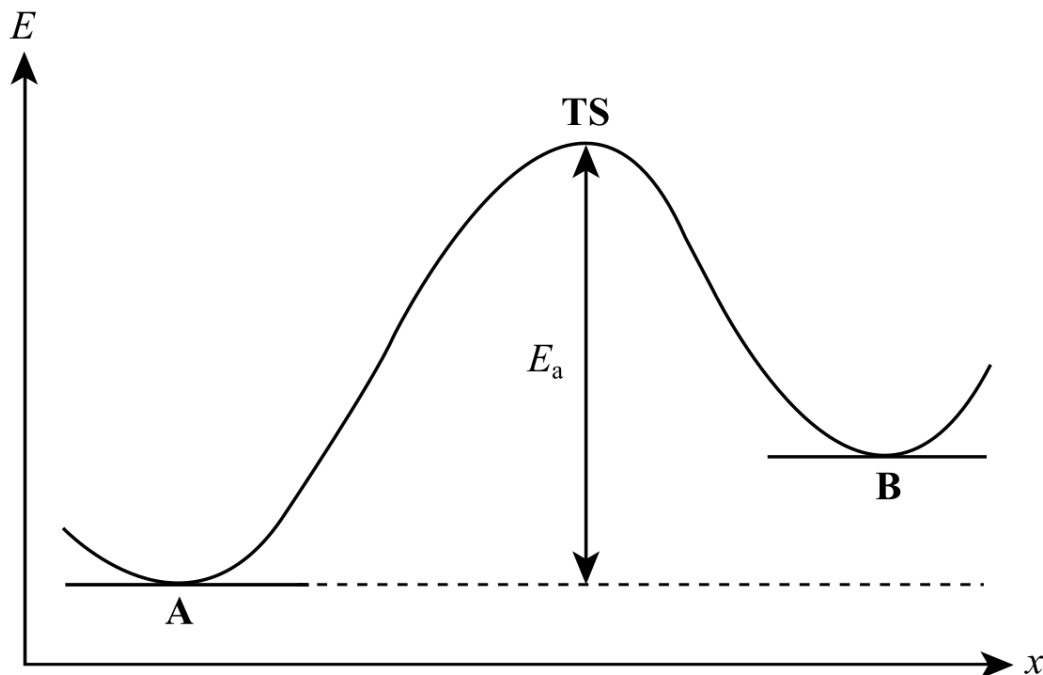


Figure 2.4: One-dimensional minimum energy path between two energy minima, going from **A** to **B**. The transition state **TS** is labelled along with its associated activation energy E_a .

2.7.2 The Hellmann-Feynman Theorem

Looking back to Section 2.2 and Section 2.3, we remember how both the Hartree-Fock SCF method and the Kohn-Sham approach are reliant on some form of the variational principle. We can expand upon this principle by introducing parameter dependencies in the Hamiltonian. For a given parameter λ , we may express the energy expectation value E_λ of the parameter-dependent Hamiltonian as

$$E_\lambda = \langle \psi_\lambda | \hat{H}_\lambda | \psi_\lambda \rangle, \quad (2.30)$$

where the solution ψ_λ is the normalized wave function in Dirac notation [27]. If we try to differentiate this equation we end up with two terms, one of which vanishes by the variational principle [9, p. 9]. This leaves us with the expression

$$\frac{dE_\lambda}{d\lambda} = \left\langle \psi_\lambda \left| \frac{\partial \hat{H}_\lambda}{\partial \lambda} \right| \psi_\lambda \right\rangle, \quad (2.31)$$

known as the *Hellmann-Feynman Theorem* [28]. The terms in the Hamiltonian that involve interactions with the nuclei are inherently dependent on the atomic

positions \mathbf{R}_I . Since we have built our models around keeping these positions fixed, we can consider the coordinates as being parameters for the Hamiltonian. With the parameter-dependency $\lambda = \mathbf{R}_I$ we now use Eq. 2.31 to get the relation

$$\mathbf{F}_I = -\frac{\partial E}{\partial \mathbf{R}_I} = \left\langle \psi \left| -\frac{\partial \hat{H}_\lambda}{\partial \mathbf{R}_I} \right| \psi \right\rangle, \quad (2.32)$$

giving us the *Hellmann-Feynman forces* acting on a given nucleus I .

2.7.3 Relaxation Scheme

Optimizing the geometry into an equilibrium is now a similar procedure to the variational approach for SCF cycles; in order to find the most stable state, we can repeatedly inact variations on the positions \mathbf{R}_I until we are in a PES minima with forces $\mathbf{F}_I = 0$ for all nuclei. For each step in the optimization process we need to calculate the relevant properties required to proceed; this means running a self-consistent DFT-calculation on the electrons at each step, given the current molecular structure. Since we are aiming to optimize the structure into a state that is more energetically favorable, we often refer to geometry optimizations as a relaxation of the system. The relaxation scheme in Fig. 2.5 illustrates how we can relax an initial geometry into a final ground state.

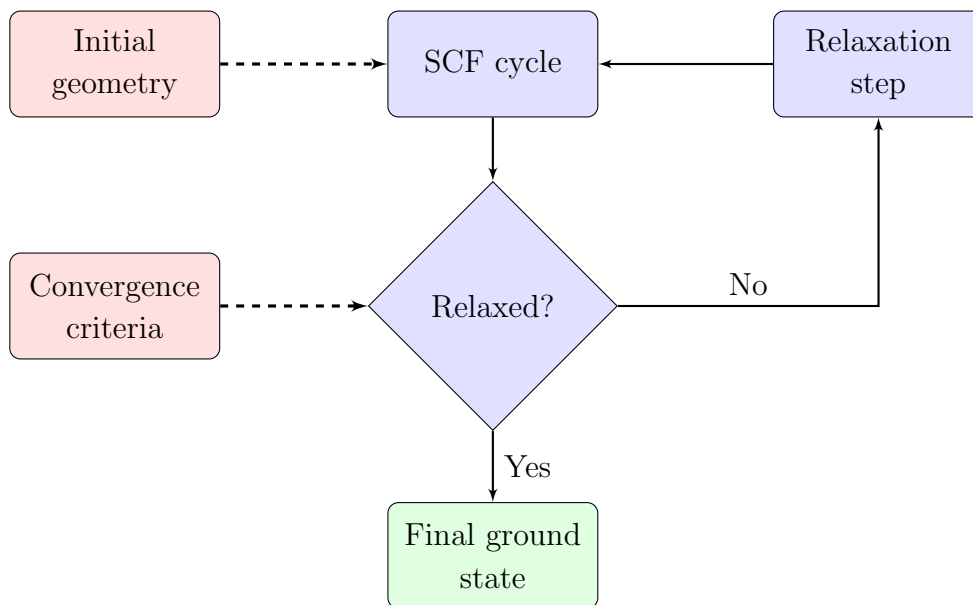


Figure 2.5: Geometry optimization scheme from the initial geometry to the final ground state. We apply criteria for energy convergence and acceptable force values.

CHAPTER

3

COMPUTATIONAL APPROACH

As we have seen so far, the DFT framework includes many sophisticated models and approximations that help describe the interactions within many-body systems. However, to apply all this theory in actual calculations on real systems, we also need to accept some numerical limitations. In this chapter we will present the plane wave pseudopotential approach to solving the Kohn-Sham equations, and explain how we can optimize the geometry of systems within this method.

3.1 Plane Wave Method and Energy Cutoff

We consider a system where the electron wave functions obey Bloch's theorem as described in Eq. 2.25. The function $u_k(\mathbf{r})$ has the same periodicity as the potential that the given electrons are moving in, and can thus be expanded using a basis set of plane waves as

$$u_k(\mathbf{r}) = \sum_{\mathbf{G}} c_{\mathbf{G}} e^{i\mathbf{G}\cdot\mathbf{r}}, \quad (3.1)$$

where we sum over all vectors \mathbf{G} defined in reciprocal space [5, p. 61]. If we put this back into Eq. 2.25 we get the expression

$$\psi_k(\mathbf{r}) = \sum_{\mathbf{G}} c_{\mathbf{k}+\mathbf{G}} e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}}, \quad (3.2)$$

which still contains an infinite sum over the \mathbf{G} -vectors. Having an infinite sum in our expression does not seem promising for calculations, but upon closer inspection we may recognize that this form of the wave function is a simple solution to the Schrödinger equation, where the kinetic energy is given by

$$E = \frac{\hbar^2}{2m} |\mathbf{k} + \mathbf{G}|^2. \quad (3.3)$$

With the physical interpretation of the wave function as a combination of high-energy and low-energy plane waves, we can try to approximate the full solution by excluding some of the high-energy terms. With an upper bound *cutoff energy* given as

$$E_{\text{cut}} = \frac{\hbar^2}{2m} G_{\text{cut}}^2, \quad (3.4)$$

we can now truncate the sum in Eq. 3.2, so that it only includes the terms that satisfy the relation $|\mathbf{G} + \mathbf{k}| < G_{\text{cut}}$. Software implementations of this concept usually come with a recommended energy cutoff for each element in the periodic table. In situations with different types of atoms, we choose the cutoff energy that matches the highest value from the respective atom types, and we should always use the same cutoff energy when comparing energy differences between systems.

3.1.1 Alternative Basis Sets

In situations where we are modeling our systems to be non-periodic, we require a different type of basis set to describe the electron wave functions. We have already seen how the electron wave function expressed in Eq. 2.9 uses a linear combination of atomic orbitals (LCAO) to approximate the spin-orbital representation of the wave function, so we now only need to specify the basis functions η_i .

Among the most commonly used basis functions we find Gaussian type orbitals (GTO) and Slater type orbitals (STO). The GTO basis set uses atomic orbitals of the form

$$\eta_i^{\text{GTO}} = C r^\beta e^{-\alpha r^2}, \quad (3.5)$$

which can be easily implemented in its analytical form. The Slater type orbitals are defined as

$$\eta_i^{\text{STO}} = C' r^\beta e^{-\alpha r} Y_{lm}(\theta, \phi), \quad (3.6)$$

where the function $Y_{lm}(\theta, \phi)$ is a spherical harmonic with quantum numbers l and m . We can use multiple STOs in a given basis set to increase the accuracy of calculations; the minimal basis set is referred to as SZ (single zeta), followed by the doubled basis set DZ (double zeta), then TZ (triple zeta) and so forth. These basis functions can then further be augmented with polarization functions as TZP, TZ2P, etc.

3.2 Pseudopotentials

Using plane waves as a basis set is practical, but there are a couple of concerns that need to be addressed. The electrons that are tightly bound to the core of a nucleus are associated with rapid oscillations in the wave function [5, p. 63-64]. Consequently, the all-electron wave function Ψ_{AE} inherits this behavior near an atomic core. This means that if we wanted to describe the wave function accurately, we would have to include high-energy plane waves, which in turn would require a high cutoff energy, making our calculations more costly and less efficient.

The core electrons are computationally problematic, but we also know from a physical standpoint that the core electrons are not as interesting as the valence electrons. This gives us the motivation to try approximating the wave function near the core of each nucleus. The approach illustrated in Fig. 3.1 shows how we can replace the nuclei-electron potential V with a non-diverging *pseudopotential* in order to smoothen the all-electron wave function within a cutoff radius near the core. The system will then remain unchanged outside the range of the cutoff radius, while getting the benefit of a greatly reduced the requirement for the cutoff energy. Pseudopotentials were first introduced by Hans Hellmann in 1934 [29].

The plane wave pseudopotential approach is often combined with the so-called *frozen core approximation*, which aims to keep the properties of the core electrons fixed throughout calculations. There are also varying levels of "hardness" across different types of pseudopotentials, meaning some have smoother potentials and wave functions associated with them than others. The cutoff radius can also be adjusted depending on the needs and assumptions for different systems.

One of the most popular types of pseudopotentials are the so-called *ultra-soft pseudopotentials* (USPPs). They are designed to require comparably lower cutoff energy values, while relying more on empirical parameters. An arguably more reliable alternative to USPPs is the *projector augmented-wave* method (PAW), which tends to give similar results to USPPs, but with better accuracy for some materials [5, p. 64-65]. The PAW method takes root in transformation theory, and was first introduced by Blöchl in the mid 1990s [30]. Its implementation for plane wave calculations was later carried out by Kresse and Joubert [31].

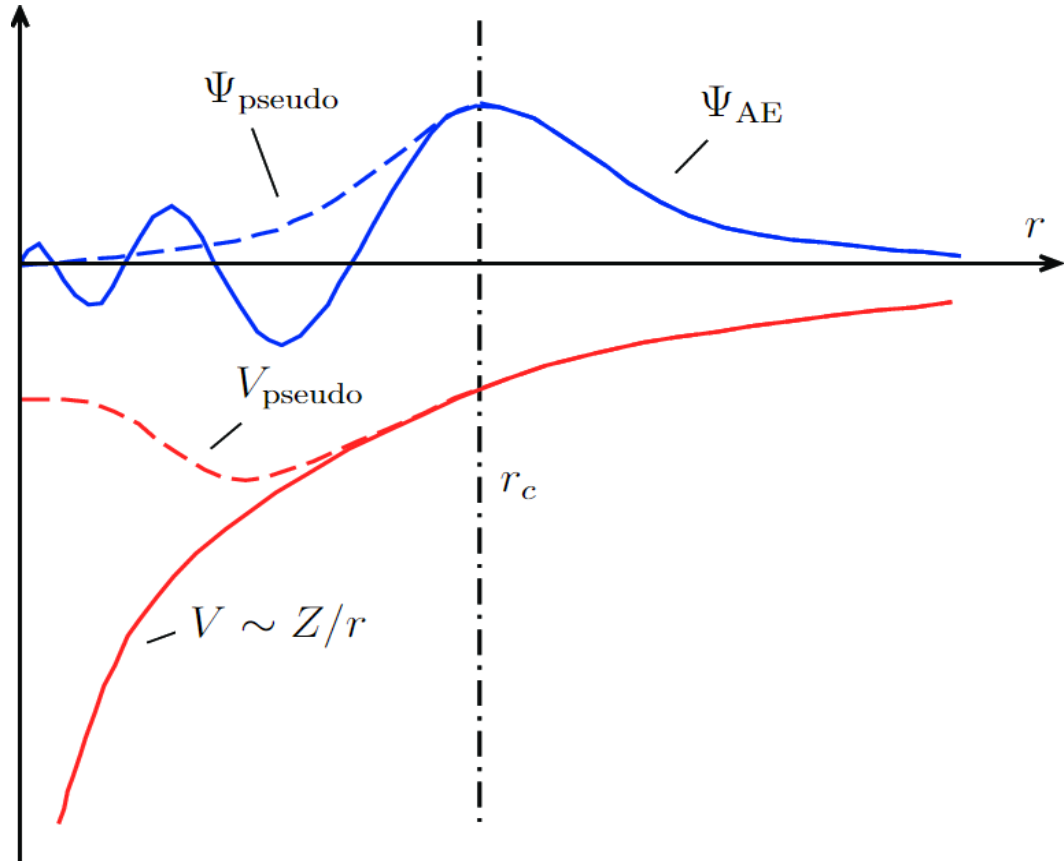


Figure 3.1: The concept of a pseudopotential (modified from [32]). The nucleielectron potential V is replaced by the pseudopotential V_{pseudo} inside the bounds of the cutoff radius r_c . The all-electron wave function Ψ_{AE} is as a result of this smoothed into the wave function Ψ_{pseudo} .

3.3 The BFGS Algorithm

In Section 2.7 we discussed the idea behind the process of structural relaxation, but we have yet to show a way to implement this numerically. With this in mind, we will now present some of the theory behind an efficient algorithm to be used for numerical optimization.

As a starting point, we consider *Newton's method* for numerical optimization. Newton's method is a well-known algorithm designed to find the roots of a given real-valued function in an iterative manner. Given the function $f(x)$ and its first derivative $f'(x)$, we define the iteration scheme for Newton's method as

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad (3.7)$$

where each step is bringing us closer to a root of $f(x)$. If we now consider the function $f(x)$ to be twice differentiable, we may replace $f(x)$ with $f'(x)$ in Eq. 3.7 to instead approximate the points x at which $f'(x) = 0$. Generalized to n dimensions, we now write the minimizing scheme

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [H_f(\mathbf{x}_k)]^{-1} \nabla f(\mathbf{x}_k), \quad (3.8)$$

where $H_f(x)$ is the Hessian containing all second-order partial derivatives of $f(x)$.

Next, we consider the optimization approach of the *quasi-Newton methods*. This class of optimization methods aims to modify Newton's method by approximating the Hessian matrix $H_f(x)$ using a positive-definite matrix B [33]. The nature of B is dependent on which quasi-Newton method we choose, but we always require that it satisfies the relation

$$B_{k+1} [\mathbf{x}_{k+1} - \mathbf{x}_k] = y_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k). \quad (3.9)$$

Now that we have defined the framework for all quasi-Newton methods, we can finally specify our method of choice. The BFGS algorithm is defined by the update formula for the matrix B given as

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T \Delta x_k} - \frac{B_k \Delta x_k (B_k \Delta x_k)^T}{\Delta x_k^T B_k \Delta x_k} \quad (3.10)$$

The computational complexity of the BFGS method can be shown to be $\mathcal{O}(n^2)$, which compared to $\mathcal{O}(n^3)$ for Newton's method is significantly more efficient. The BFGS method was developed in 1970 through the combined effort of Broyden [34], Fletcher [35], Goldfarb [36] and Shanno [37], hence the acronym BFGS.

With an optimization method in place, we can now contextualize its utility in the case of geometry relaxation. If we interpret the function $f(x)$ as representing the potential energy in the system, we can draw parallels to the PES discussion in Section 2.7.1. The matrix B is here an approximation of the Hessian matrix in Eq. 2.27, leading us to a minimization of potential energy into a local minima.

3.4 Parameters and Convergence Factors

We have now introduced the main methods and approximations that are used for plane wave pseudopotential calculations and system relaxations. The approach we

have chosen relies on a few different parameters that can be adjusted depending on our needs; we can adjust the cutoff energy E_{cut} for our plane wave basis, the number of k -points in our Monkhorst-Pack grid, and we can adjust the shape of our unit cell with the choice of real space lattice vectors \mathbf{a}_i . We can also choose between different types of pseudopotentials to smoothen the wave function, select density functionals that are either local, semi-local or non-local, and we can even choose to add dispersion corrections on top of our Kohn-Sham energy.

The most important parameters that we need to specify are the convergence criteria. The flowcharts shown in Fig. 2.1, Fig. 2.2 and Fig. 2.5 all require that we supply the self-consistency cycles with definite tolerance values for discrepancies between iterations. We may for instance specify a threshold energy $E_{\text{thr}} = 10^{-7}$ eV, and define a convergence criteria where we demand that the energy difference ΔE between SCF iterations should fulfill the relation $\Delta E \leq E_{\text{thr}}$. A similar criteria can be specified for the Hellmann-Feynman forces, where we instead monitor the deviations from $F_I = 0$.

When doing SCF calculations and relaxation simulations we may sometimes end up in a situation where the energy oscillates between iterations, and thus does not converge as expected. This can indicate that we need to change the pseudopotential, adjust parameters like E_{cut} , E_{thr} , or increase the number of k -points, etc.

3.4.1 Occupation Number Smearing

In systems with significant band gaps, the positioning of the Fermi level plays a crucial role in determining electrical properties. If the Fermi level is inside of a band gap, we get a discontinuity in the occupation numbers at the Fermi surface. Whenever we try to apply the plane wave pseudopotential method to these kinds of systems, we are prone to run into issues with computations in reciprocal space; due to the discontinuous nature of the Fermi surface, the integrals presented in Eq. 2.26 can become problematic.

To remedy this situation we introduce the concept of *smearing functions*. The point of these functions is to help "smear out" the occupancies at the Fermi sur-

face to extend over a larger area, allowing for fewer k -points to be necessary for the detection of energy contributions. As shown in Fig. 3.2, using a Gaussian type smearing function can distribute the occupation numbers more smoothly around the Fermi level. Other smearing function options include the Fermi-Dirac distribution, Methfessel-Paxton first-order spreading [38] and Marzari-Vanderbilt-DeVita-Payne cold smearing [39].

We note that smearing functions are only used as a mathematical trick to better the rate of convergence in calculations, and that it should not affect the underlying physics of the system.

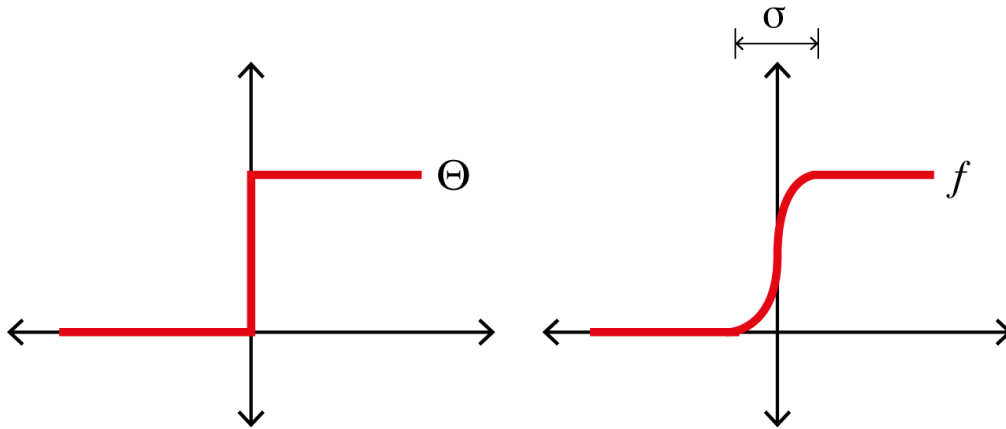


Figure 3.2: Comparison of distribution functions at the Fermi level [40]. The sharp Dirac step function Θ on the left vs. the smooth Gaussian function f on the right. The width of the Gaussian spread is here denoted by σ .

CHAPTER

4

SOFTWARE CHOICES

There are many different DFT software options that support the plane wave pseudopotential approach in way we have described in this thesis. Some simulation softwares are open-source, while others require a license. In this chapter we will look at two different types of open-source DFT softwares, going through the script structures and how we can run calculations efficiently from a Linux terminal. Lastly, we will also briefly discuss how to model a fluorohectorite clay system within this framework.

4.1 Quantum ESPRESSO

The first software we have chosen to look at is the open-source suite known as *Quantum ESPRESSO* (QE). This simulation software is based entirely in the plane wave pseudopotential approach described in this thesis, and is thus it is inherently limited to periodic models. Although it offers some options for two-dimensional slab calculations, we must always assume the system to be periodic in at least two spacial directions. The QE suite is able to perform ground-state calculations, structural optimizations and molecular dynamics simulations, find response properties and spectroscopic properties, look at quantum transport, and more [41]. The default units for lengths and energies are in this software set as a.u. and Ry respectively. In this section we will look at self-consistent plane wave calculations (PWscf) in the context of structural optimizations, starting with a discussion around the pseudopotentials and input scripts that go into the plane wave calculation program *pw.x* within the QE suite.

4.1.1 Pseudopotential Resources

The QE suite only accepts potentials formatted in the Unified Pseudopotential Format (UPF). Generated UPF files can be found listed in several different online libraries. The best places to look for pseudopotentials are either in the *PSlibrary* resource listed on the QE home page, or in the more actively updated library of *standard solid-state pseudopotentials* (SSSP) at the materials cloud webpage [42]. Depending on the element, we use either USPPs or PAW potentials, both of which can be found in the *SSSP Efficiency* library. Whenever we run PWscf calculations, the density functional is chosen based on the data found in the pseudopotential.

4.1.2 Input Script Structure

When using the *pw.x* program for PWscf calculations, our main input quantities are the shape of the unit cell, the positions of atoms within said unit cell, and the types of pseudopotentials that are used for each of the respective elements. The geometrical properties can be written directly in the input script, while the pseudopotentials require that we specify the name and path to the UPF files.

The structure of a *pw.x* input script consists of a set of *namelists* and *cards* that categorize the parameter specifications related to the electrons, ions, atomic positions, distribution of *k*-points, etc. In order to explain what each namelist and card is responsible for, we consider a simple PWscf relaxation calculation example for a single CO₂ molecule in a large unit cell. We start with the `&control` namelist, which in our SCF calculation takes the following form:

```
&control
  calculation='relax',
  restart_mode='from_scratch',
  prefix='co2-relax',
  pseudo_dir='../pseudo/',
  outdir='/work/temp'
```

The `calculation` parameter lets us choose between different types of simulations; the string 'scf' corresponds to a simple SCF calculation, whereas writing the string

'relax' will result in a geometry optimization for the molecular structure. We can also choose the calculation type 'vc-relax', which in addition to the geometry also tries to optimize the lattice vectors in the unit cell with a variable cell approach. The `prefix` parameter, here specified by the string 'co2-relax', is used to label our system so that we can find the corresponding temporary files in a directory with this label. The two parameters `pseudo_dir` and `outdir` specify the path to our pseudopotentials and temporary files respectively. As demonstrated above, paths can both be specified locally ('./pseudo') or defined with an absolute path from the root directory ('/work/temp'). We can set the `restart_mode` to 'restart' to recover a stopped calculation.

Next, we consider the `&system` namelist. As seen in the snippet below, this is where parameters such as the number of atoms `nat` and the number of types of atoms `ntyp` are defined. The cutoff energies `ecutwfc` and `ecutrho` should be set to the recommended values for the given pseudopotentials; in this case the values are based on the PAW potential for oxygen found in version 1.1.2 of the SSSP Efficiency library. We have applied a Gaussian smearing method through the specifications of `occupations` and `smearing`, with a spread value given as `degauss = 0.01 Ry`. The `ibrav` parameter specifies the nature of our unit cell; the values 1, 2 and 3 correspond to simple cubic, face centered cubic and body centered cubic structures respectively. We have here chosen `ibrav = 0`, which means we to have to specify the unit cell later on in the script. It is also possible to include dispersion in the `&system` namelist, either by changing the functional as `input_dft='vdW-DF2'` or by adding a correction term `vdw_corr='DFT-D3'`.

```
&system
  ibrav=0,
  nat=3,
  ntyp=2,
  ecutwfc=50,
  ecutrho=400,
  occupations='smearing',
  smearing='gaussian',
  degauss=0.01
```

The `&electrons` namelist specifies the parameters that are related to the actual SCF calculations. In the snippet below we have defined the SCF cycle convergence criteria as `conv_thr = 10-8 Ry`, and required through `electron_maxstep = 100` and `scf_must_converge` that a given SCF cycle must converge within 100 SCF steps. The value of `electron_maxstep` is by default already set to 100, but we include it here to get a better overview of which parameters are involved.

```
&electrons
  electron_maxstep = 100
  scf_must_converge = .TRUE.
  conv_thr = 1.0d-8
```

The `&ions` namelist is only relevant whenever we are running a geometry relaxation or some other form of molecular dynamics. In the snippet below we have specified the `ion_dynamics` to run the BFGS algorithm, as described by the theory in Section 3.3. The QE suite also offers other methods, such as the FIRE algorithm using a semi-implicit Euler integration scheme [43].

```
&ions
  ion_dynamics='bfgs'
```

We now consider the two *cards* that define the atoms and their positions within the unit cell. The `ATOMIC_SPECIES` card specifies the symbol, mass and pseudopotential associated with the different types of atoms in the system, and the `ATOMIC_POSITIONS` card defines the coordinates of each of these atoms in units of Angstroms. We have in the snippet below named our two pseudopotential files 'C-PAW.UPF' and 'O-PAW.UPF', and placed them in the directory './pseudo/' so that the *pw.x* program is able to find and read the data.

```
ATOMIC_SPECIES
C 12.0107 C-PAW.UPF
O 15.9990 O-PAW.UPF

ATOMIC_POSITIONS {angstrom}
C -5.9811199593 1.4385953275 -6.9996149920
O -7.0912251134 1.0617547448 -6.9996202962
O -4.8711439886 1.8153981458 -6.9996001485
```

We now finally consider the last two cards, namely the `CELL_PARAMETERS` and `K_POINTS` cards. The `CELL_PARAMETERS` card lets us directly define the three lattice vectors that make up the unit cell, acting as the alternative to the definition using the `ibrav` parameter discussed earlier. For this system we chose a $10 \times 10 \times 10$ unit cell, where we again specified Angstroms as the unit of length. Due to it being such a small system, we set the k -points distribution to only the gamma point.

```
CELL_PARAMETERS {angstrom}
10 0 0
0 10 0
0 0 10
K_POINTS {gamma}
```

For larger systems we can replace the `{gamma}` argument with the `{automatic}` keyword, as shown in the snippet below. We can then instead specify the three Monkhorst-Pack parameters `nk1`, `nk2` and `nk3`, and set the three offset values `sk1`, `sk2` and `sk3` to zero.

```
K_POINTS {automatic}
nk1 nk2 nk3 sk1 sk2 sk3
```

4.1.3 Parallel Runs

To run a `pw.x` simulation we need to make use of the Linux terminal. In general, for a given input script `system.in` and output file `system.out`, we can run a PWscf calculation on our system by writing the command

```
$ pw.x -in system.in > system.out
```

Writing it this way will only occupy one of the cores on our computer. We can drastically improve calculation time if we have a way to run calculations in parallel over multiple cores at once. With a version of *Open MPI* installed on our computer, we can write the command for a parallel run over N cores as

```
$ mpirun -np N pw.x -in system.in > system.out
```

4.1.4 Using Shell Scripts

So far we have assumed that we need to write commands directly in the terminal to start calculations, but this is typically not very efficient. If we wanted to run a convergence test over one of the parameters in the input script, we would have to type out the commands for every time we changed the parameter. Instead, we introduce *shell scripts* that can automate some of this work for us.

As an example, let us consider a convergence test of the cutoff energy for the CO₂ molecule. In the shell script 'co2-cutoff.sh' we can then write out the commands we want to run using the syntax of the command language *BASH* (Bourne-Again Shell), and loop over the system with varying values of `ecutwfc` and `ecutrho`. At the start of the shell script shown on the next page, we see an expression consisting of a shebang followed by the path to the shell '/bin/sh'; this is how we can identify that it indeed is a shell script. Throughout the loop in the shell script we are defining variables that go back into the *pw.x* input script definition. We are only running SCF calculations here, without any relaxation for the ions. The calculations are performed using 8 cores.

For each loop we are only interested in the total energy in the output files. The *pw.x* output files are constructed in a way where this self-consistent energy is always marked with an exclamation mark. Instead of scrolling through each output file manually, we use the *grep* command to look lines that follow this pattern. After the total energy is found, we can then write the results of the given cutoff energy to a data file.

In order to run a shell script with Bash, we need to revisit the Linux terminal. We continue with the shell script 'co2-cutoff.sh' in the CO₂ cutoff convergence example, and write the following into the terminal:

```
$ bash 'co2-cutoff.sh'
```

We should now have the most recent input and output files available, and a filled out relation between the cutoff energy and its corresponding SCF energy at the given geometry of the system.

```
#!/bin/sh
# Varying the cutoff energy
for cutoff in 10 20 30 40 50 60 70 80; do

# Define new cutoff values
ecutwfc=$cutoff
ecutrho=$((($cutoff * 10))

# Making a SCF calculation input script
cat > 'co2-cutoff.in' << EOF
&control
    calculation='scf',
    ...
/
&system
    ecutwfc=$ecutwfc,
    ecutrho=$ecutrho,
    ...
...
    K_POINTS {gamma}
EOF

# Performing calculations in parallel on 8 cores
mpirun -np 8 pw.x -in 'co2-cutoff.in' > 'co2-cutoff.out'

# Obtaining total energy from output file
etot=`grep -e ! $output_name | awk '{print $(NF-1)}' | tail -1`

# Writing results to data file
echo $cutoff $etot >> co2-cutoff.data
done
exit 0
```

4.2 GPAW

GPAW is a DFT Python code based on the projector augmented wave approach for calculating the self-consistent ground state total energy. It is usually used in combination with the atomic simulation environment (ASE), which is a framework that allows for structural optimizations and calculations of other properties. In contrast to calculations in the QE suite, the GPAW software has the capability to use other approaches such as finite-difference methods or the LCAO approach for non-periodic systems. We can here replace the plane wave basis set with atom-centered basis functions constructed as a radial part multiplied with spherical harmonics. The GPAW code replaces the concept of pseudopotentials with what are known as *atomic PAW setups*; similarly to pseudopotentials, setups are data sets that aim help to represent the nature of atoms in the frozen core approximation. The default units for GPAW are Angstroms and electron volts eV for lengths and energies respectively.

4.2.1 Script Structure

Since GPAW and ASE are based in Python, we can easily install them as Python packages. The structure of GPAW scripts are no different from that of a usual Python project; we import the desired modules and take advantage of the objects and functions that are readily implemented within these modules. As shown in the snippet below, we can import ASE resources to represent `Atoms`, and allow for parallel calculations with a single output file made accessible with `paropen`. We will follow the same type of BFGS optimization as we did for QE; a relaxation of a periodic CO2 system in a large unit cell. Since we have a periodic system, we import the plane wave method `PW` alongside our GPAW calculator.

```
from ase import Atoms
from ase.optimize import BFGS
from ase.parallel import paropen
from gpaw import GPAW, PW
```

Next, we look to make the same system specifications as we did using the QE

suite. As seen in the next snippet, we define our system in a single object separate from the parameters related to the SCF calculations. We specify the `symbols` and `positions` of our atoms, and we define the shape of the unit `cell`. We want the system to be periodic in all spacial directions, so we specify the periodic boundary conditions `pbcs` to be "turned on" in all three directions with values of 1. All this information is then stored in the ASE `Atoms` object that we called `system`.

```
# System specifications
system = Atoms(symbols=['C', 'O', 'O'],
               positions=[
                   [-5.9811199593, 1.4385953275, -6.9996149920],
                   [-7.0912251134, 1.0617547448, -6.9996202962],
                   [-4.8711439886, 1.8153981458, -6.9996001485]],
               cell=[
                   [10, 0, 0],
                   [0, 10, 0],
                   [0, 0, 10]],
               pbc=[1, 1, 1])
```

We now want to define the parameters that go into the SCF calculation models. The `dft` object in the snippet below is a GPAW type calculator based on the same parameters as in the general input script for QE calculations. We have here defined a cutoff energy of 700 eV for the plane waves, which is roughly comparable with the previous QE cutoff energy. We have chosen the Fermi-Dirac distribution as our smearing function, and a k -points distribution of only the gamma point.

```
# GPAW DFT calculator (PBE functional)
dft = GPAW(mode=PW(700),
            xc='PBE',
            label='co2-relax',
            setups='paw',
            txt='co2-relax.out',
            occupations={'name': 'fermi-dirac', 'width': 0.05},
            kpts={'size': (1, 1, 1), 'gamma': True})
```

We note that the atomic setups obtained from `setups = 'paw'` in the snippet above are included in the installation of GPAW, and it is therefore not necessary to download files from online libraries, like we do with QE.

The next step is to attach the GPAW calculator to the system through the methods allowed by the ASE Atoms object, and start the geometry optimization using the BFGS algorithm.

```
# Attaching the calculator to the system
system.calc = dft

# Preparing for geometry optimization
relax = BFGS(atoms=system,
             trajectory='co2-relax.traj',
             logfile='co2-relax.log')

# Doing the relaxation calculation
relax.run(fmax=0.005)
```

Instead of looking through the output file 'co2-relax', we may now obtain the properties of our system simply by calling methods such as `get_positions()` and `get_potential_energy()`:

```
# Obtaining the final state of the system
geometry = system.get_positions()
energy = system.get_potential_energy()

# Writing the results to a data file
f = paropen('co2-relax.data', 'a')
print(f'Geometry:\n', geometry, file=f)
print(f'Energy (eV):', energy, file=f)
f.close()
```

This concludes the brief introduction to how we can carry out calculations for structural relaxations using a GPAW Python script.

4.2.2 Parallel Runs

We can do parallel runs for GPAW scripts in a similar fashion to how we approached the QE input scripts. For a given number of cores N and a GPAW script 'system.py', we can write the Linux terminal command as

```
$ mpirun -np N gpaw python 'system.py'
```

In some cases where the computer architecture is different, we might be required to oversubscribe in the following way:

```
$ mpirun --oversubscribe -np N gpaw python 'system.py'
```

Similarly to what we discussed for QE input scripts in Section 4.1.4, we can use shell scripts to automate parts of a convergence test using GPAW. It can in essence be constructed in the exact same way for GPAW as we did for QE, and so we will not discuss it any further.

4.3 Clay Calculations

In this section we will implement the clay structures discussed in Section 1.1, and do some minor testing in the Quantum ESPRESSO suite. The structure of fluorohectorite includes a total number of atoms `nat = 38` within each unit cell, distributed in layers stacked upwards along the z -direction. As shown in Fig. 4.1, we can recognize the middle-most octahedral layer dominated by Mg atoms, and the tetrahedral layers of Si atoms on either side.

We have here run a variable cell relaxation calculation (`vc-relax`) to find a more optimized unit cell. Grimme's DFT-D3 dispersion correction has been included alongside pseudopotentials for the PBE functional. We chose a kinetic cutoff energy of 70 Ry, with applied Gaussian smearing at the Fermi level. Approximately 150 k -points were used. The entirety of the calculation took around 8 hours using an 8-core AMD Ryzen 7 5800X Processor.

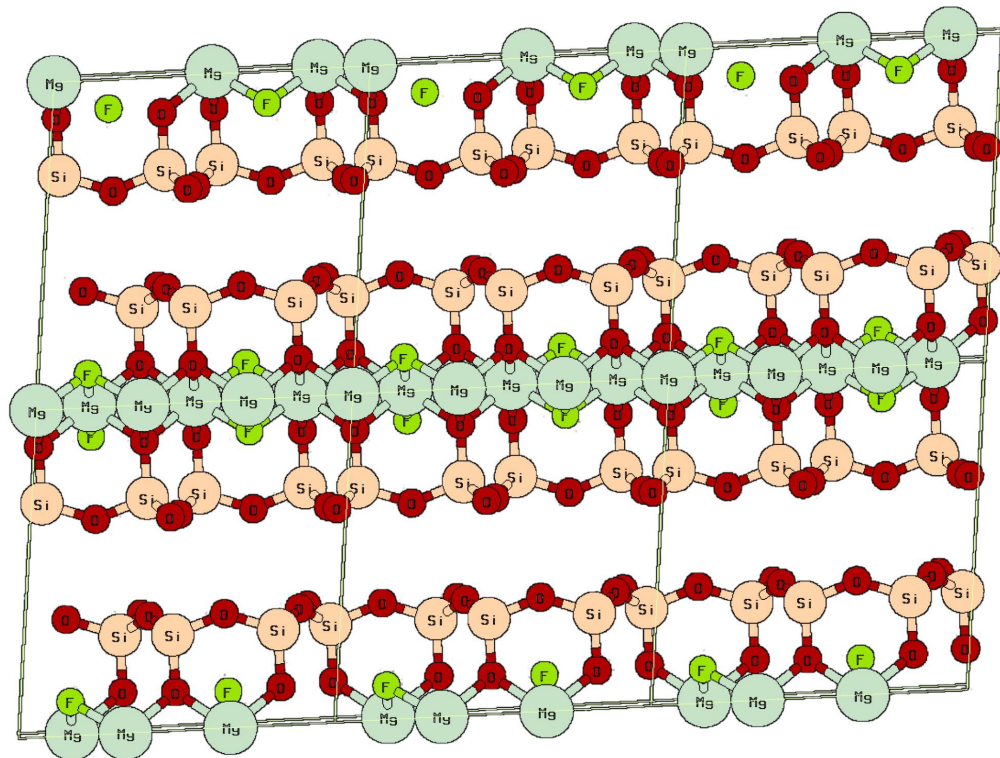


Figure 4.1: Optimized structure of fluorohectorite in the xz -plane. There are no intercalated ions present. The thin lines indicate the bounds of each unit cell.

Fig. 4.2 shows the effect of varying the distance d_{001} between layers in a fluorohectorite system with no intercalated ions. With the DFT-D3 dispersion correction added, we notice the Lennard-Jones-like shape of the function.

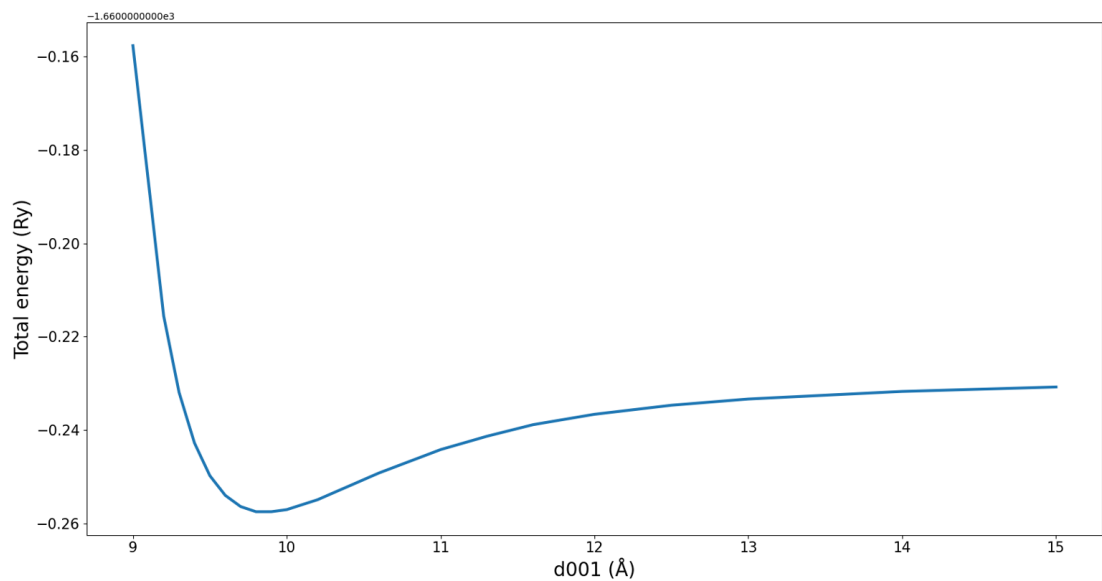
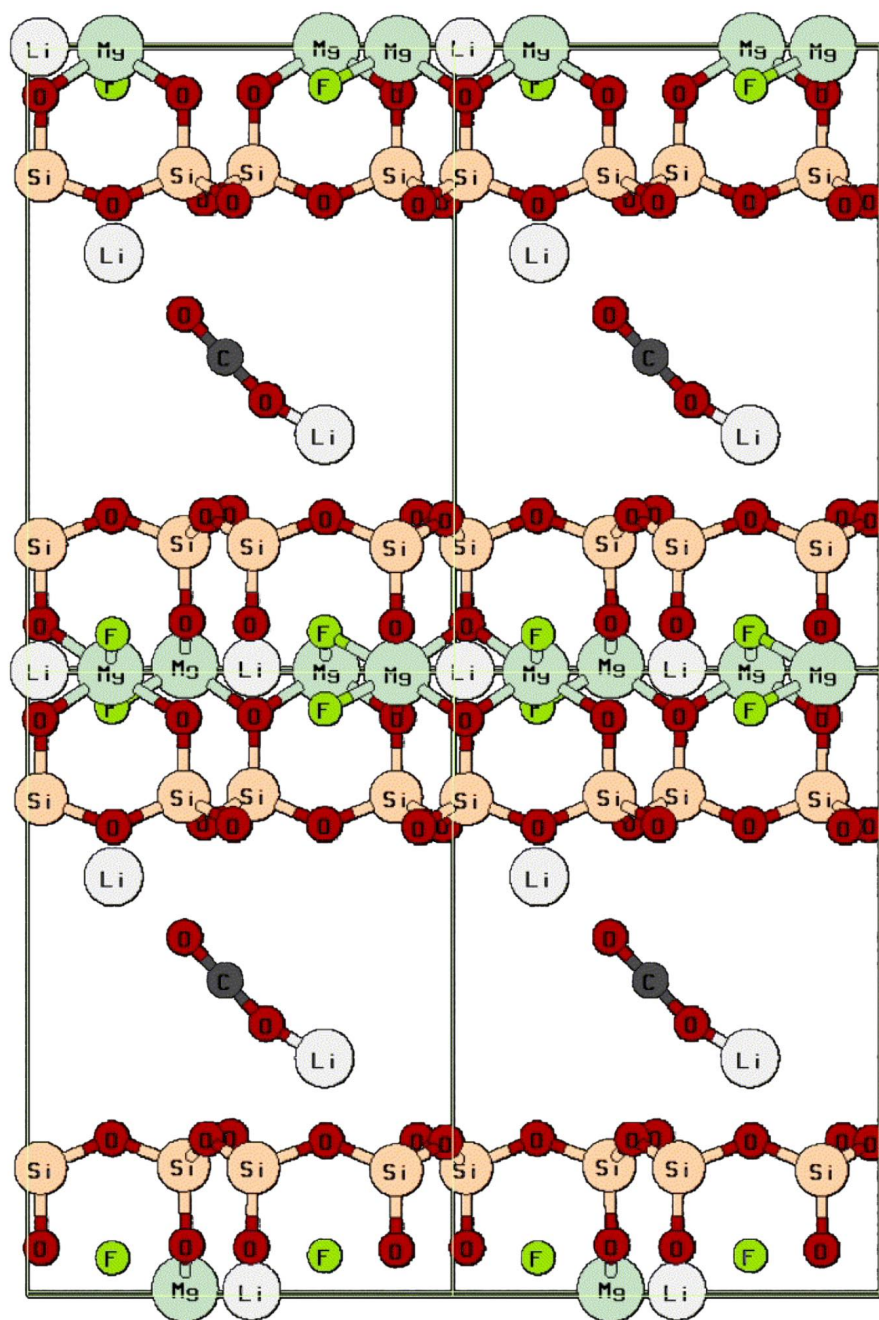
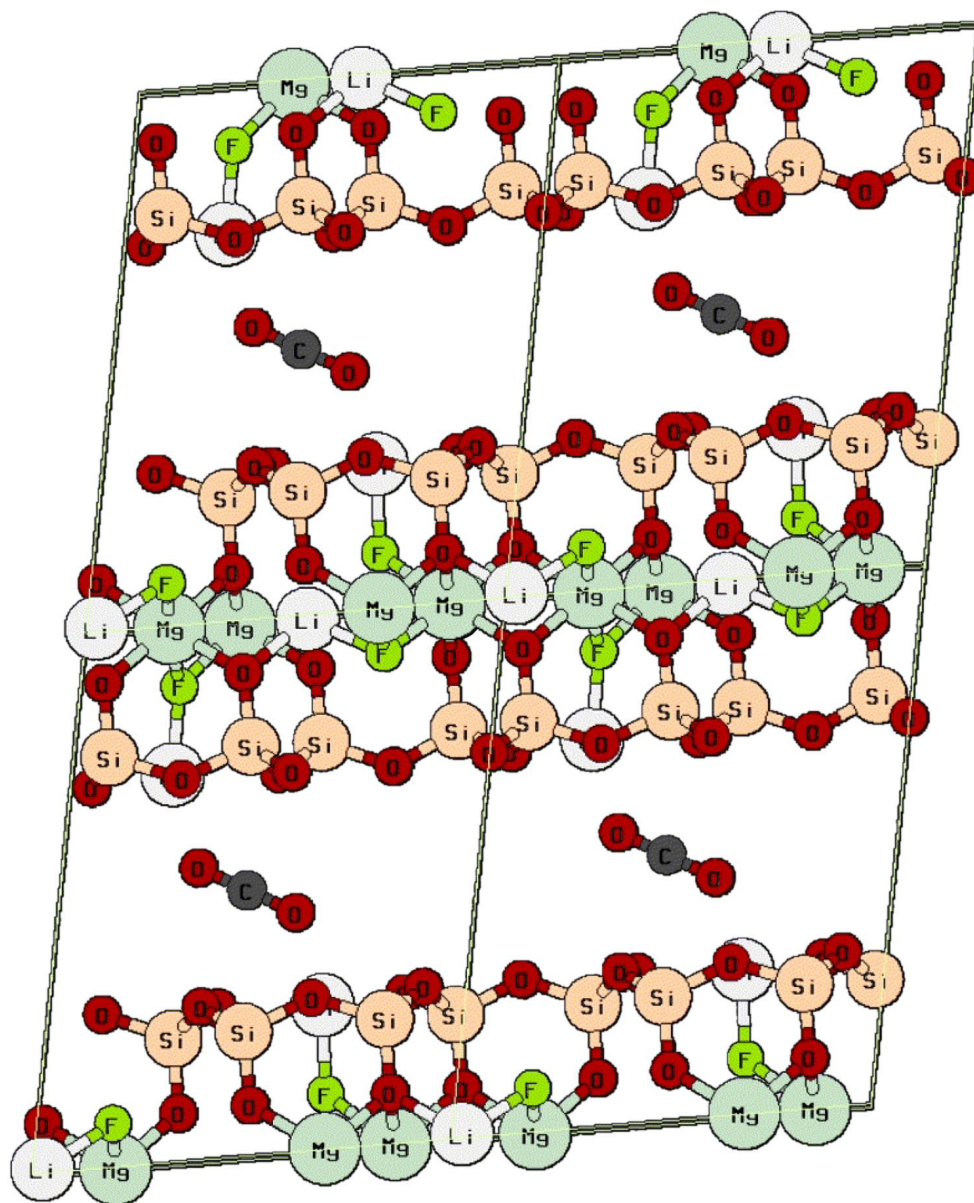


Figure 4.2: The total energy as a function of the distance d_{001} between layers.

Lastly we consider a clay system with two intercalated Li^+ ions and a CO_2 molecule. To compensate for the added charges in the system, we have replaced two of the Mg^{2+} ions in the octahedral layer with Li^+ ions. We simulated another variable cell geometry optimization over 135 SCF cycles until the system converged. This calculation took approximately 20 hours with the same hardware setup as for the clay-only calculations above. Fig. 4.3 shows the comparison between (a) the initial geometry and (b) the final relaxed state.



(a) Initial state of the system.



(b) Final relaxed state.

Figure 4.3: Structural optimization of a Li₂-Li₂-fluorohectorite system with an intercalated CO₂ molecule and a variable unit cell. The Li⁺ ions are moving away from the center to become part of the clay structure, while the CO₂ molecule remains central.

CONCLUSION

5.1 Concluding Remarks

We have now extensively discussed how the DFT method can help us find the ground state solution of the time-independent Schrödinger equation for many-body systems. Motivated by the need for simplifications in this problem, we have discovered many sophisticated methods from the fields of quantum chemistry, solid state physics and mathematical optimization theory. We have presented a way to set up self-consistent field calculations on the electron density, and seen how we can combine this with quasi-Newtonian numerical relaxation schemes to estimate local energy minima. The importance of periodicity in the plane wave basis set approach has been utilized in calculations to test out two different open-source DFT softwares. We have modeled the fluorohectorite smectite in the context of applying dispersion corrections to the Kohn-Sham energy in the DFT method, and we have seen the evolution of a relaxing clay structure with intercalated ions and molecules.

5.2 Future Work

In order to more thoroughly test the performance of the simulation softwares presented in this thesis, a good starting point can be to try and reproduce the results previously found with licensed software, particularly within the world of layered clay materials. Trying out non-periodic DFT calculations in the LCAO or finite difference approach might also be an interesting exercise for the future.

An active research topic here at NTNU is the study of CO₂ adsorption in fluorohectorite. It has been experimentally shown that intercalated Ni²⁺ ions can adsorb CO₂ better than most ions from group I and II. A suggested theory for this is that the Ni²⁺ ions will combine to form hydroxides connected to the layered clay surfaces. Testing this theory numerically with DFT calculations is definitely an interesting topic for the future.

References

- [1] L. Michels, J. O. Fossum, Z. Rozynek, H. Hemmen, K. Rustenberg, P. A. Sobas, G. N. Kalantzopoulos, K. D. Knudsen, M. Janek, T. S. Plivelic and G. J. da Silva, ‘Intercalation and Retention of Carbon Dioxide in a Smectite Clay promoted by Interlayer Cations’, *Scientific Reports*, vol. 5, no. 1, p. 8775, Mar. 2015. DOI: [10.1038/srep08775](https://doi.org/10.1038/srep08775).
 - [2] L. Bailey, H. N. W. Lekkerkerker and G. C. Maitland, ‘Smectite clay – inorganic nanoparticle mixed suspensions: Phase behaviour and rheology’, *Soft Matter*, vol. 11, no. 2, pp. 222–236, 2015. DOI: [10.1039/C4SM01717J](https://doi.org/10.1039/C4SM01717J).
 - [3] M. Born and R. Oppenheimer, ‘Zur Quantentheorie der Molekeln’, *Annalen der Physik*, vol. 389, no. 20, pp. 457–484, 1927. DOI: [10.1002/andp.19273892002](https://doi.org/10.1002/andp.19273892002).
 - [4] D. R. Hartree, ‘The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part II. Some Results and Discussion’, *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 24, no. 1, pp. 111–132, Jan. 1928. DOI: [10.1017/S0305004100011920](https://doi.org/10.1017/S0305004100011920).
 - [5] D. S. Sholl and J. A. Steckel, *Density functional theory: a practical introduction*. Hoboken, N.J: Wiley, 2009, OCLC: ocn245025462, ISBN: 9780470373170.
 - [6] P. C. Hemmer, *Kvantemekanikk: P.C. Hemmer*. Trondheim: Tapir akademisk forlag, 2005, OCLC: 489028564, ISBN: 9788251920285.
 - [7] J. C. Slater, ‘The Self Consistent Field and the Structure of Atoms’, *Physical Review*, vol. 32, no. 3, pp. 339–348, Sep. 1928. DOI: [10.1103/PhysRev.32.339](https://doi.org/10.1103/PhysRev.32.339).
 - [8] W. Pauli, ‘ber den Zusammenhang des Abschlusses der Elektronengruppen im Atom mit der Komplexstruktur der Spektren’, de, *Zeitschrift für Physik*, vol. 31, no. 1, pp. 765–783, Feb. 1925. DOI: [10.1007/BF02980631](https://doi.org/10.1007/BF02980631).
-

- [9] C. Fiolhais, F. Nogueira and M. A. L. Marques, Eds., *A primer in density functional theory*, ser. Lecture notes in physics 620. Berlin ; New York: Springer, 2003, ISBN: 9783540030836.
- [10] P. Hohenberg and W. Kohn, ‘Inhomogeneous Electron Gas’, *Physical Review*, vol. 136, no. 3B, B864–B871, Nov. 1964. DOI: 10.1103/PhysRev.136.B864.
- [11] L. Hedin and B. I. Lundqvist, ‘Explicit local exchange-correlation potentials’, *Journal of Physics C: Solid State Physics*, vol. 4, no. 14, pp. 2064–2083, Oct. 1971, Publisher: IOP Publishing. DOI: 10.1088/0022-3719/4/14/022.
- [12] J. P. Perdew, K. Burke and M. Ernzerhof, ‘Generalized Gradient Approximation Made Simple’, en, *Physical Review Letters*, vol. 77, no. 18, pp. 3865–3868, Oct. 1996, ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.77.3865.
- [13] Y. Zhang and W. Yang, ‘Comment on “Generalized Gradient Approximation Made Simple”’, *Physical Review Letters*, vol. 80, no. 4, pp. 890–890, Jan. 1998. DOI: 10.1103/PhysRevLett.80.890.
- [14] H. Rydberg, M. Dion, N. Jacobson, E. Schröder, P. Hyldgaard, S. I. Simak, D. C. Langreth and B. I. Lundqvist, ‘Van der Waals Density Functional for Layered Structures’, *Physical Review Letters*, vol. 91, no. 12, p. 126 402, Sep. 2003. DOI: 10.1103/PhysRevLett.91.126402.
- [15] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth and B. I. Lundqvist, ‘Van der Waals Density Functional for General Geometries’, *Physical Review Letters*, vol. 92, no. 24, p. 246 401, Jun. 2004. DOI: 10.1103/PhysRevLett.92.246401.
- [16] S. Grimme, ‘Accurate description of van der Waals complexes by density functional theory including empirical corrections’, *Journal of Computational Chemistry*, vol. 25, no. 12, pp. 1463–1473, Sep. 2004. DOI: 10.1002/jcc.20078.
- [17] —, ‘Semiempirical GGA-type density functional constructed with a long-range dispersion correction’, *Journal of Computational Chemistry*, vol. 27, no. 15, pp. 1787–1799, Nov. 2006. DOI: 10.1002/jcc.20495.
-

- [18] S. Grimme, J. Antony, S. Ehrlich and H. Krieg, ‘A consistent and accurate *ab initio* parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu’, *The Journal of Chemical Physics*, vol. 132, no. 15, p. 154104, Apr. 2010. DOI: 10.1063/1.3382344.
- [19] J.-D. Chai and M. Head-Gordon, ‘Long-range corrected hybrid density functionals with damped atom–atom dispersion corrections’, *Physical Chemistry Chemical Physics*, vol. 10, no. 44, p. 6615, 2008. DOI: 10.1039/b810189b.
- [20] A. D. Becke and E. R. Johnson, ‘Exchange-hole dipole moment and the dispersion interaction revisited’, *The Journal of Chemical Physics*, vol. 127, no. 15, p. 154108, Oct. 2007. DOI: 10.1063/1.2795701.
- [21] A. Tkatchenko and M. Scheffler, ‘Accurate Molecular Van Der Waals Interactions from Ground-State Electron Density and Free-Atom Reference Data’, *Physical Review Letters*, vol. 102, no. 7, p. 073005, Feb. 2009. DOI: 10.1103/PhysRevLett.102.073005.
- [22] A. Ambrosetti, A. M. Reilly, R. A. DiStasio and A. Tkatchenko, ‘Long-range correlation energy calculated from coupled atomic response functions’, *The Journal of Chemical Physics*, vol. 140, no. 18, 18A508, May 2014. DOI: 10.1063/1.4865104.
- [23] C. Kittel, *Introduction to solid state physics*, 8th ed. Hoboken, NJ: Wiley, 2005, ISBN: 9780471415268.
- [24] S. Harrison, ‘Exploring and Exploiting Charge-Carrier Confinement in Semiconductor Nanostructures’, 2016. DOI: 10.13140/RG.2.2.29386.57286.
- [25] H. J. Monkhorst and J. D. Pack, ‘Special points for Brillouin-zone integrations’, *Physical Review B*, vol. 13, no. 12, pp. 5188–5192, Jun. 1976. DOI: 10.1103/PhysRevB.13.5188.
- [26] J. Simons, *3.1: Strategies for Geometry Optimization and Finding Transition States*, 2021. [Online]. Available: [https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps / Advanced_Theoretical_Chemistry_\(Simons\) /03 % 3A_Characteristics_of_Energy_Surfaces / 3 . 01 % 3A_Strategies_for_Geometry_Optimization_and_Finding_Transition_States](https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Advanced_Theoretical_Chemistry_(Simons)/03%3A_Characteristics_of_Energy_Surfaces/3.01%3A_Strategies_for_Geometry_Optimization_and_Finding_Transition_States).
-

- [27] P. A. M. Dirac, ‘A new notation for quantum mechanics’, *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 35, no. 3, pp. 416–418, Jul. 1939. DOI: 10.1017/S0305004100021162.
- [28] R. P. Feynman, ‘Forces in Molecules’, *Physical Review*, vol. 56, no. 4, pp. 340–343, Aug. 1939. DOI: 10.1103/PhysRev.56.340.
- [29] P. Schwerdtfeger, ‘The Pseudopotential Approximation in Electronic Structure Theory’, *ChemPhysChem*, vol. 12, no. 17, pp. 3143–3155, Dec. 2011. DOI: 10.1002/cphc.201100387.
- [30] P. E. Blöchl, ‘Projector augmented-wave method’, *Physical Review B*, vol. 50, no. 24, pp. 17 953–17 979, Dec. 1994. DOI: 10.1103/PhysRevB.50.17953.
- [31] G. Kresse and D. Joubert, ‘From ultrasoft pseudopotentials to the projector augmented-wave method’, *Physical Review B*, vol. 59, no. 3, pp. 1758–1775, Jan. 1999. DOI: 10.1103/PhysRevB.59.1758.
- [32] M. Råsander, ‘A Theoretical perspective on the chemical bonding and structure of transition metal carbides and multilayers’, Ph.D. dissertation, Uppsala Universitet, 2010.
- [33] A. Lam, *BFGS in a Nutshell: An Introduction to Quasi-Newton Methods*, 2020. [Online]. Available: <https://towardsdatascience.com/bfgs-in-a-nutshell-an-introduction-to-quasi-newton-methods-21b0e13ee504>.
- [34] C. G. Broyden, ‘The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations’, *IMA Journal of Applied Mathematics*, vol. 6, no. 1, pp. 76–90, 1970. DOI: 10.1093/imamat/6.1.76.
- [35] R. Fletcher, ‘A new approach to variable metric algorithms’, *The Computer Journal*, vol. 13, no. 3, pp. 317–322, Mar. 1970. DOI: 10.1093/comjnl/13.3.317.
- [36] D. Goldfarb, ‘A family of variable-metric methods derived by variational means’, *Mathematics of Computation*, vol. 24, no. 109, pp. 23–26, 1970. DOI: 10.1090/S0025-5718-1970-0258249-6.
- [37] D. F. Shanno, ‘Conditioning of quasi-Newton methods for function minimization’, *Mathematics of Computation*, vol. 24, no. 111, pp. 647–656, 1970. DOI: 10.1090/S0025-5718-1970-0274029-X.
-

- [38] M. Methfessel and A. T. Paxton, ‘High-precision sampling for Brillouin-zone integration in metals’, *Physical Review B*, vol. 40, no. 6, pp. 3616–3621, Aug. 1989. DOI: 10.1103/PhysRevB.40.3616.
- [39] N. Marzari, D. Vanderbilt, A. De Vita and M. C. Payne, ‘Thermal Contraction and Disordering of the Al(110) Surface’, *Physical Review Letters*, vol. 82, no. 16, pp. 3296–3299, Apr. 1999. DOI: 10.1103/PhysRevLett.82.3296.
- [40] B. Murphy, ‘The physico-chemical properties of fullerenes and porphyrin derivatives deposited on conducting surfaces’, Ph.D. dissertation, Trinity College Dublin, 2014.
- [41] *What can QE do*, 2022. [Online]. Available: <https://www.quantum-espresso.org/what-can-qe-do/>.
- [42] P. Gianluca, M. Antimo, C. E Ivano, M. Nicolas, P. Elsa and M. Nicola, *A Standard Solid State Pseudopotentials (SSSP) library optimized for precision and efficiency*, en, Type: dataset, May 2021. DOI: 10.24435/MATERIALSCLOUD:RZ-77. [Online]. Available: <https://archive.materialscloud.org/record/2021.76>.
- [43] J. Guénolé, W. G. Nöhring, A. Vaid, F. Houllé, Z. Xie, A. Prakash and E. Bitzek, ‘Assessment and optimization of the fast inertial relaxation engine (fire) for energy minimization in atomistic simulations and its implementation in lammmps’, *Computational Materials Science*, vol. 175, p. 109584, Apr. 2020. DOI: 10.1016/j.commatsci.2020.109584.
-

