

# Applied Hartree-Fock methods

Atomic and diatomic energy computations

**Anders Larsen**  
**Rolf Sommer Poulsen**

*Supervisor :*  
*Thomas Garm Pedersen*

Masters project



**AALBORG UNIVERSITY**

STUDENT REPORT

Department of Physics

Aalborg University

Denmark

Spring, 2015





**AALBORG UNIVERSITY**  
STUDENT REPORT

**Department of Physics and  
Nanotechnology**  
Aalborg University  
<http://www.aau.dk>

**Title:**

Applied Hartree-Fock methods

**Theme:**

Atomic and diatomic energy calculations

**Project Period:**

Spring semester 2015

**Project Group:**

anro-F15

**Participants:**

Anders Larsen  
Rolf Sommer Poulsen

**Supervisor:**

Thomas Garm Pedersen

**Copies:** 4

**Page Numbers:** 68

**Date of Completion:**

June 2, 2015

**Abstract:**

The aim of this project was to calculate the electron and potential energy functions of diatomic molecules using the Hartree-Fock method. In this report the theory behind the Hartree-Fock method is derived, explained and the implementation through Hartree-Fock-Roothaan equations has been presented. To demonstrate the Hartree-Fock method of central field problems it was used to calculate the electron energy of neutral atoms from  $Z = 1$  through  $Z = 103$ , as well as for anions and cations of the 52 first elements with good accuracy. Calculations on static atomic polarizabilities has been attempted. Results for the Hydrogen atom has been presented, but further research is needed for the rest of the atoms. Applying a Cartesian Gaussian basis the energy functions and potential energy functions of the molecules  $H_2$ ,  $HeH$ ,  $He_2$ ,  $LiH$  and  $Li_2$  were calculated. The calculations yielded ambiguous results and need further research.





**AALBORG UNIVERSITET**  
STUDENTERRAPPORT

Institut for Fysik og Nanoteknologi  
Aalborg Universitet  
<http://www.aau.dk>

**Titel:**

Applied Hartree-Fock methods

**Tema:**

Atomic and diatomic energy calculations

**Projektperiode:**

Spring semester 2015

**Projektgruppe:**

anro-F15

**Deltagere:**

Anders Larsen  
Rolf Sommer Poulsen

**Vejleder:**

Thomas Garm Pedersen

**Oplag:** 4

**Sidetal:** 68

**Afleveringsdato:**

2. juni 2015

**Abstract:**

Målet med dette projekt var at beregne elektron energier og potentialer for diatomare molekyler ved anvendelse af Hartree-Fock metoder. I denne rapport udledes og forklares teorien bag Hartree-Fock metoden, og implementeringen gennem Hartree-Fock-Roothaan ligningerne er blevet præsenteret.

For at demonstrere Hartree-Fock metoden anvendt på et centralpotentialproblem er elektronenergiene for atomerne  $Z = 1$  til og med  $Z = 103$  blevet beregnet, ligesom energierne for anioner og kationer for de 52 første atomer er præsenteret med acceptabel nøjagtighed.

Beregninger for atomare polarisabiliteter er blevet forsøgt. Resultater for Hydrogen er præsenteret, men yderligere undersøgelser er nødvendige for de resterende atomer.

Ved at anvende en kartesisk Gaussisk basis er elektronenergi og potentialer for molekylerne  $H_2$ ,  $HeH$ ,  $He_2$ ,  $LiH$  and  $Li_2$  blevet beregnet. Da beregningerne viser tvetydige resultater er der brug for nærmere undersøgelser.

*Rapportens indhold er frit tilgængeligt, men offentliggørelse (med kildeangivelse) må kun ske efter aftale med forfatterne.*



# Preface

This report is written by two FYS10 students during the spring of 2015 at Aalborg University. The aim of the report is to model multi-particle quantum mechanical systems, namely electron energies of atoms and diatomic molecules. Furthermore the project sought to calculate the polarizability of the elements using the Hartree-Fock method.

The target group is students who are familiar with basic quantum mechanics, differential equations, calculus and linear algebra.

Figures and equations will be denoted and referred to with two numbers like  $(x.y)$  where  $x$  denotes the chapter and  $y$  denotes the number of the given figure or equation. Citations are referred to using the Vancouver method where a reference is denoted by a number like [3]. In the report the following notations have been used: A vector named  $r$  is denoted  $\mathbf{r}$ . A unit vector in the  $z$ -direction is denoted in  $\mathbf{e}_z$ . The complex conjugate of  $a$  is denoted  $a^*$ , matrices are denoted like  $M$  and operators are denoted like  $\hat{H}$ .

The report is typeset in L<sup>A</sup>T<sub>E</sub>X. Illustrations are made in TikZ and all calculations and figures have been produced in Python 3.0 using matplotlib, the numpy and scipy packages. To some of the analytical calculations the CAS-tool Wolfram Mathematica 10 has been used.

The used scripts are presented on a DVD attachment in appendix D

---

Anders Larsen

---

Rolf Sommer Poulsen





# Contents

<b>1 Preliminaries</b>	<b>1</b>
1.1 The many-electron problem . . . . .	1
1.2 The Aufbau principle . . . . .	2
1.3 The Adiabatic approximation . . . . .	2
1.4 Slater Determinants . . . . .	4
1.5 Matrix operations on Slater Determinants . . . . .	4
1.6 The Hartree-Fock equations . . . . .	6
1.7 The Hartree-Fock-Roothaan equations . . . . .	9
<b>2 Implementation for atoms</b>	<b>11</b>
2.1 Gaussian basis functions . . . . .	11
2.2 The matrix elements . . . . .	12
2.3 Self Consistent Field procedure . . . . .	13
2.4 Convergence schemes . . . . .	13
2.5 Results . . . . .	15
2.6 Discussion . . . . .	17
2.7 Ions . . . . .	18
2.8 Conclusion . . . . .	21
<b>3 Polarizability calculations</b>	<b>23</b>
3.1 Polarizability . . . . .	23
3.2 The Hamiltonian with an external electric field . . . . .	23
3.3 Implementation . . . . .	24
3.4 Results . . . . .	24
<b>4 Implementation for molecules</b>	<b>27</b>
4.1 Calculation of molecular matrix elements . . . . .	27
4.2 Procedure . . . . .	34
4.3 Results . . . . .	35
4.4 Discussion . . . . .	37
4.5 Vibrational states . . . . .	40
4.6 Conclusion . . . . .	44
<b>5 Conclusion</b>	<b>45</b>
<b>A Matrix elements for atomic calculations</b>	<b>47</b>
A.1 The coulomb elements . . . . .	47
A.2 The exchange elements . . . . .	48

<b>B Hermite Polynomials</b>	<b>51</b>
<b>C Python code</b>	<b>53</b>
C.1 Hartree_Fock.py . . . . .	53
C.2 Polarizability.py . . . . .	56
C.3 Molecules.py . . . . .	60
C.4 Vibrational_Diatomic.py . . . . .	63
<b>D DVD with code</b>	<b>65</b>

# Chapter 1

## Preliminaries

The statements of this chapter can be found in any typical text book on quantum mechanics such as [3] or [1].

### 1.1 The many-electron problem

As it is commonly known the single electron Schrödinger equation in atomic units for a single-nucleus atom is

$$\left(-\frac{1}{2}\nabla^2 - \frac{Z}{r}\right)\psi = E\psi. \quad (1.1)$$

Let the Hamiltonian for this problem here forward be denoted  $\hat{h}_i$  such that

$$\hat{h}_i = -\frac{1}{2}\nabla_i^2 - \frac{Z}{r_i}. \quad (1.2)$$

Adding  $N$  electrons to the atom gives rise to an electron repulsion term so that the Hamiltonian becomes

$$\left(\sum_{i=1}^N \hat{h}_i + \sum_{i<j}^N \frac{1}{r_{ij}}\right)\Psi = E\Psi. \quad (1.3)$$

The collected wave functions,  $\Psi$ , contains information about both the spacial coordinates and spin of all electrons. Moreover the term

$$r_{ij} = |\mathbf{r}_i - \mathbf{r}_j| \quad (1.4)$$

simply describes the distance between two electrons.

If the many-electron Hamiltonian is introduced as

$$\hat{H} = \sum_{i=1}^N \hat{h}_i + \sum_{i<j}^N \frac{1}{r_{ij}}, \quad (1.5)$$

the collected Schrödinger equation for the many-electron atom becomes

$$\hat{H}\Psi = E\Psi. \quad (1.6)$$

## 1.2 The Aufbau principle

In order to understand and investigate the electronic energies of the atoms it is important to know the electronic states around the nucleus. More specifically the electron configuration of the atom is desired. Using numerous chemical and spectroscopic evidence a few guidelines for placing electrons around the nucleus of the atom have been derived. These rules are known as the Aufbau principle.

The idea behind the Aufbau principle is that the formation of an atom can be considered as a successive capturing of electrons and maintaining them in their most stable conditions. In this way the hydrogen atom is created when a proton capture a single electron and holding it in its most stable state called the  $1s$  orbital. The chemical and spectroscopic evidence suggests that a given spacial state only can hold two electrons resulting in the fact that the electrons of the Lithium atom are in two different spacial states. Since the atom tend to be in the most stable condition the two different spacial states of the Lithium atom must be the ones with the lowest possible energies being the  $1s$  and  $2s$  orbitals. The electron configuration of Lithium is then stated as  $1s^2 2s^1$ . Using this procedure the Aufbau principle can be stated as the following rules:

The electron configuration of an element may be determined by filling the orbitals after increasing values of  $n + l$ . For a fixed value of  $n + l$  orbitals with lower  $n$ -values are filled first.

Here  $n$  is the principal quantum number whereas  $l$  is the azimuthal quantum number.

Due to repulsion between electrons and the nucleus not all electron configurations obey the Aufbau principle but its a good guideline for the construction of atoms.

## 1.3 The Adiabatic approximation

In all the quantum mechanical calculations of this project the adiabatic approximation also known as the Born-Oppenheimer approximation is used to simplify the problems. The approximation assumes that because the electrons are so much lighter than the nuclei they rearrange instantaneously according to the positions of the nuclei. In this picture the nuclei can be considered as stationary and independent of the electron motion such that the molecular wave function may be written

$$\Psi_{\text{mol}}(\mathbf{x}, \boldsymbol{\rho}) = \Psi(\mathbf{x}; \boldsymbol{\rho}) \kappa(\boldsymbol{\rho}), \quad (1.7)$$

where  $\mathbf{x}$  denotes the spin and radial coordinates of the electrons and  $\boldsymbol{\rho}$  denotes the coordinates of the nuclei. Hereby the molecular wave function is written as a product of an electron wave function  $\Psi(\mathbf{x}; \boldsymbol{\rho})$  of electrons at  $\mathbf{x}$  assuming that the nuclei are at position  $\boldsymbol{\rho}$  and a nuclear wave function  $\kappa(\boldsymbol{\rho})$ . This approximation yields a easy way to implement the quantum mechanical calculations used in this project. Consider a system consisting of  $N$  electrons and  $M$  nuclei the Hamiltonian neglecting spin in atomic units yields

$$\hat{H}_{\text{mol}} = - \sum_{k=1}^M \frac{1}{2M_k} \nabla_k^2 + \sum_{k<l}^M \frac{Z_k Z_l}{r_{kl}} - \sum_{i=1}^N \frac{1}{2} \nabla_i^2 - \sum_{k=1}^M \sum_{i=1}^N \frac{Z_k}{r_{ik}} + \sum_{i<j}^N \frac{1}{r_{ij}}, \quad (1.8)$$

where  $k$  and  $l$  labels the nuclei and  $i$  and  $j$  the electrons.  $M_k$  is the mass of the  $k$ th nucleus. Using this Hamiltonian the molecular Schrödinger equation may be written as

$$\hat{H}_{\text{mol}}\Psi_{\text{mol}}(\mathbf{x}, \boldsymbol{\rho}) = E_{\text{mol}}\Psi_{\text{mol}}(\mathbf{x}, \boldsymbol{\rho}). \quad (1.9)$$

We rewrite the molecular Hamiltonian as

$$\hat{H}_{\text{mol}} = -\sum_{k=1}^M \frac{1}{2M_k} \nabla_k^2 + \sum_{k<l}^M \frac{Z_k Z_l}{r_{kl}} + \hat{H}, \quad (1.10)$$

where

$$\hat{H} = \sum_i \hat{h}_i + \sum_{i<j} \frac{1}{r_{ij}} \quad (1.11)$$

is an electronic Hamiltonian concerning the coordinates of the electrons.

Using the Born-Oppenheimer approximation in equation 1.7 we can write the Schrödinger equation as

$$\begin{aligned} \frac{-1}{\Psi(\mathbf{x}; \boldsymbol{\rho})} \sum_{k=1}^M \frac{1}{2M_k} \nabla_k^2 \Psi(\mathbf{x}; \boldsymbol{\rho}) \kappa(\boldsymbol{\rho}) + \sum_{k<l}^M \frac{Z_k Z_l}{r_{kl}} + \frac{1}{\Psi(\mathbf{x}; \boldsymbol{\rho})} \hat{H} \Psi(\mathbf{x}; \boldsymbol{\rho}) \\ = E_{\text{mol}}. \end{aligned} \quad (1.12)$$

If we then assume that  $\Psi(\mathbf{x}; \boldsymbol{\rho})$  varies very slowly with  $\boldsymbol{\rho}$  we may write

$$\frac{-1}{\kappa(\boldsymbol{\rho})} \sum_{k=1}^M \frac{1}{2M_k} \nabla_k^2 \kappa(\boldsymbol{\rho}) + \sum_{k<l}^M \frac{Z_k Z_l}{r_{kl}} + \frac{1}{\Psi(\mathbf{x}; \boldsymbol{\rho})} \hat{H} \Psi(\mathbf{x}; \boldsymbol{\rho}) = E_{\text{mol}}. \quad (1.13)$$

From this equation it is clear that the third term, which is the only term regarding electron coordinates, must simplify to a function independent from  $\mathbf{x}$ . Calling this function  $E(\boldsymbol{\rho})$  we get the requirement

$$\hat{H} \Psi(\mathbf{x}; \boldsymbol{\rho}) = E(\boldsymbol{\rho}) \Psi(\mathbf{x}; \boldsymbol{\rho}). \quad (1.14)$$

Using this the Schrödinger equation may be written as

$$\left( -\sum_{k=1}^M \frac{1}{2M_k} \nabla_k^2 + U(\boldsymbol{\rho}) \right) \kappa(\boldsymbol{\rho}) = E_{\text{mol}} \kappa(\boldsymbol{\rho}), \quad (1.15)$$

where

$$U(\boldsymbol{\rho}) = E(\boldsymbol{\rho}) + \sum_{k<l}^M \frac{Z_k Z_l}{r_{kl}}. \quad (1.16)$$

Now the Schrödinger equation of the entire system is written as two separate equations. One regarding the electrons, 1.14, which must be solved for every nuclear configurations and yields the electronic energy function  $E(\boldsymbol{\rho})$ . The other equation, 1.15, is the Schrödinger equation of the nuclei. The function  $U(\boldsymbol{\rho})$  is called the potential energy function and is the sum of the electronic energy function and the nuclear repulsion.

## 1.4 Slater Determinants

When calculating quantum mechanical problems the aim is to find a wave function which is a solution to the Schrödinger equation in equation 1.6. This wave function does not only describe the motion of one electron but instead it explains the behaviour of all the electrons of the calculated system. One way of constructing such a wave function is the so-called Hartree product defined as the product of all the one-electron wave functions of the system

$$\Psi(\mathbf{x}) = \psi_1(\mathbf{x}_1) \psi_2(\mathbf{x}_2) \cdots \psi_N(\mathbf{x}_N). \quad (1.17)$$

However this approach of defining the many-electron wave function is not valid for electrons due to the Pauli exclusion principle. This states that the wave function of electrons should be antisymmetric, a property the wave function in equation 1.17 does not possess. However by specific permutations of the arguments of the one-electron wave functions in equation 1.17 one can create an antisymmetrical many-electron wave function. This permutation is then called the Slater determinant and is defined to be

$$\Psi(\mathbf{x}) = |\psi_1 \psi_2 \cdots \psi_N| = \frac{1}{\sqrt{N!}} \begin{vmatrix} \psi_1(\mathbf{x}_1) & \psi_1(\mathbf{x}_2) & \cdots & \psi_1(\mathbf{x}_N) \\ \psi_2(\mathbf{x}_1) & \psi_2(\mathbf{x}_2) & \cdots & \psi_2(\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_N(\mathbf{x}_1) & \psi_N(\mathbf{x}_2) & \cdots & \psi_N(\mathbf{x}_N) \end{vmatrix}. \quad (1.18)$$

It is assumed that the single electron wave functions are orthonormal. Due to the mathematical properties of the determinant the Slater determinant obey the antisymmetric behaviour of the many-electron wave function.

## 1.5 Matrix operations on Slater Determinants

This section aims to derive rules for calculating matrix elements of the form  $\langle \Psi | \hat{F} | \Psi' \rangle$  for different operators acting on the Slater determinants.

The Slater determinant may be written as

$$\Psi = |\psi_1 \psi_2 \cdots \psi_N| = \sqrt{\frac{1}{N!}} \sum_P (-1)^p \hat{P} \psi_1(\mathbf{x}_1) \psi_2(\mathbf{x}_2) \cdots \psi_N(\mathbf{x}_N), \quad (1.19)$$

where  $\hat{P}$  is a permutation operator acting on the arguments such that

$$\hat{P} \psi_1(\mathbf{x}_1) \psi_2(\mathbf{x}_2) \cdots \psi_N(\mathbf{x}_N) = \psi_1(P\mathbf{x}_1) \psi_2(P\mathbf{x}_2) \cdots \psi_N(P\mathbf{x}_N), \quad (1.20)$$

with  $P$  as a permutation. The sum of equation 1.19 is over all the  $N!$  permutations and  $p$  denotes the number of transpositions required to generate the given permutation. The matrix element  $\langle \Psi | \hat{F} | \Psi' \rangle$  is given by

$$\langle \Psi | \hat{F} | \Psi' \rangle = \int \Psi^* \hat{F} \Psi' d\mathbf{x}, \quad (1.21)$$

where  $d\mathbf{x} = d\mathbf{x}_1 d\mathbf{x}_2 \cdots d\mathbf{x}_N$ . By using the reformulation of the Slater determinant in equation 1.19 on  $\Psi$ , a general term in the  $N!$  sums become

$$(-1)^p \frac{1}{N!} \int \psi_1^*(P\mathbf{x}_1) \psi_2^*(P\mathbf{x}_2) \cdots \psi_N^*(P\mathbf{x}_N) \hat{F} \Psi' d\mathbf{x}. \quad (1.22)$$

If  $\Psi'$  is expanded using equation 1.19 the factor  $(-1)^p$  cancels and if the integration variable  $\mathbf{x}_1$  is switched with  $P\mathbf{x}_1$  up to  $\mathbf{x}_N$  and  $P\mathbf{x}_N$  and assuming that  $\hat{F}$  is symmetric the expression in equation 1.22 becomes

$$\frac{1}{N!} \int \psi_1^*(\mathbf{x}_1) \psi_2^*(\mathbf{x}_2) \cdots \psi_N^*(\mathbf{x}_N) \hat{F} \Psi' d\mathbf{x}. \quad (1.23)$$

By adding  $N!$  of these identical terms an expression for the matrix element  $\langle \Psi | \hat{F} | \Psi' \rangle$  is obtained

$$\langle \Psi | \hat{F} | \Psi' \rangle = \int \psi_1^*(\mathbf{x}_1) \psi_2^*(\mathbf{x}_2) \cdots \psi_N^*(\mathbf{x}_N) \hat{F} \Psi' d\mathbf{x}. \quad (1.24)$$

The matrix element still needs to be evaluated under the assumptions that  $\hat{F}$  is either a sum of one-electron operators or two-electron operators. In the first case the following rules can be obtained

- a.  $\langle \Psi | \hat{F} | \Psi' \rangle = 0$   
if  $\Psi$  and  $\Psi'$  differ in more than one spin orbital.
- b.  $\langle \Psi | \hat{F} | \Psi' \rangle = \langle \psi_k | \hat{f} | \psi'_k \rangle$  (1.25)  
if  $\Psi'$  may be obtained from  $\Psi$  by replacing  $\psi'_k$  with  $\psi_k$ .
- c.  $\langle \Psi | \hat{F} | \Psi \rangle = \sum_{i=1}^N \langle \psi_i | \hat{f} | \psi_i \rangle$

The first rule comes from the fact that the contribution of  $\hat{f}(\mathbf{x}_1)$  to the matrix element is

$$\sum_P (-1)^P \langle \psi_1 | \hat{f}(\mathbf{x}_1) | \psi'_{P1} \rangle \langle \psi_2 | \psi'_{P2} \rangle \cdots \langle \psi_N | \psi'_{PN} \rangle, \quad (1.26)$$

where  $\psi'_{Pi}$  is the permutation of  $\psi'_i$  rising from the expansion of  $\Psi'$  in equation 1.24. The contribution in equation 1.26 vanishes unless

$$\psi'_{P2} = \psi_2, \dots, \psi'_{PN} = \psi_N. \quad (1.27)$$

The same approach can be made for  $\hat{f}(\mathbf{x}_2)$  up to  $\hat{f}(\mathbf{x}_N)$  thereby yielding the first rule in 1.25. For the second rule consider the case

$$\Psi = |\psi_1 \psi_2 \cdots \psi_{k-1} \psi_k \psi_{k+1} \cdots \psi_N| \quad (1.28)$$

$$\Psi' = |\psi_1 \psi_2 \cdots \psi_{k-1} \psi'_k \psi_{k+1} \cdots \psi_N| \quad (1.29)$$

Assume that  $\psi_k$  and  $\psi'_k$  are in identical positions. The contribution to the matrix element is then

$$\sum_P (-1)^P \langle \psi_1 | \psi'_{P1} \rangle \cdots \langle \psi_k | \hat{f}(\mathbf{x}_k) | \psi'_{Pk} \rangle \cdots \langle \psi_N | \psi'_{PN} \rangle, \quad (1.30)$$

but since only the identity permutation can contribute due to orthogonality the only contribution to the matrix element must be  $\langle \psi_k | \hat{f}(\mathbf{x}_k) | \psi'_k \rangle$  thereby showing the second rule. The third rule is obtained since the contribution of  $\hat{f}(\mathbf{x}_1)$  must be that of equation 1.26 since the conditions in equation 1.27 are

present. Since the same approach can be taken for  $\hat{f}(\mathbf{x}_2)$  up to  $\hat{f}(\mathbf{x}_N)$  the matrix element must be given by the third rule of 1.25.

When the operator  $\hat{F}$  is a sum of two-electron operators the following rules can be stated

- a.  $\langle \Psi | \hat{F} | \Psi' \rangle = 0$   
if  $\Psi$  and  $\Psi'$  differ in more than two spin orbitals.
- b.  $\langle \Psi | \hat{F} | \Psi' \rangle = \langle \psi_k \psi_l | \hat{f} | \psi'_k \psi'_l \rangle - \langle \psi_k \psi_l | \hat{f} | \psi'_l \psi'_k \rangle$ ,  
if  $\Psi'$  may be obtained from  $\Psi$  by replacing  $\psi'_k$  by  $\psi_k$  and  $\psi'_l$  by  $\psi_l$ .
- c.  $\langle \Psi | \hat{F} | \Psi' \rangle = \sum_{l \neq k} (\langle \psi_k \psi_l | \hat{f} | \psi'_k \psi'_l \rangle - \langle \psi_k \psi_l | \hat{f} | \psi'_l \psi'_k \rangle)$ ,  
if  $\Psi'$  may be obtained from  $\Psi$  by replacing  $\psi'_k$  by  $\psi_k$ .
- d.  $\langle \Psi | \hat{F} | \Psi \rangle = \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N (\langle \psi_k \psi_l | \hat{f} | \psi_k \psi_l \rangle - \langle \psi_k \psi_l | \hat{f} | \psi_l \psi_k \rangle)$ .

(1.31)

Since the argument for the rules in 1.31 are very similar to the rules in 1.25 they will not be mentioned here.

## 1.6 The Hartree-Fock equations

The Hartree-Fock equations are derived from applying the variational principle to Slater determinants. Small variations of the type  $\psi_i + \delta\psi_i$  in the wave functions result in a higher energy so it is preferable to minimise the variation

$$\delta \langle \Psi | \hat{H} | \Psi \rangle = 0 \quad (1.32)$$

The expectation value of the Hamiltonian, and hence the energy, is

$$\langle \Psi | \hat{H} | \Psi \rangle = \sum_{i=1}^N \langle \psi_i | \hat{h}_i | \psi_i \rangle + \sum_{i < j}^N (\langle \psi_i \psi_j | \frac{1}{r_{ij}} | \psi_i \psi_j \rangle - \langle \psi_i \psi_j | \frac{1}{r_{ij}} | \psi_j \psi_i \rangle). \quad (1.33)$$

Applying the variation  $\psi_i + \delta\psi_i$  and only considering variations of the first order gives

$$\begin{aligned} \delta \langle \Psi | \hat{H} | \Psi \rangle &= \sum_{i=1}^N \langle \delta\psi_i | \hat{h}_i | \psi_i \rangle + \sum_{i < j}^N (\langle \delta\psi_i \psi_j | \frac{1}{r_{ij}} | \psi_i \psi_j \rangle - \langle \delta\psi_i \psi_j | \frac{1}{r_{ij}} | \psi_j \psi_i \rangle) + \\ &\quad \sum_{i=1}^N \langle \psi_i | \hat{h}_i | \delta\psi_i \rangle + \sum_{i < j}^N (\langle \psi_i \psi_j | \frac{1}{r_{ij}} | \delta\psi_i \psi_j \rangle - \langle \psi_i \psi_j | \frac{1}{r_{ij}} | \psi_j \delta\psi_i \rangle). \end{aligned} \quad (1.34)$$

It is required that the spin orbitals are orthogonal even under the variation as described by

$$\delta \langle \psi_i | \psi_j \rangle = \langle \delta\psi_i | \psi_j \rangle + \langle \psi_i | \delta\psi_j \rangle = 0. \quad (1.35)$$

Introducing the Fock operator

$$\hat{F} = \hat{h} + \sum_{j=1}^N (\hat{J}'_j - \hat{K}'_j) \quad (1.36)$$



where

$$\hat{J}'_j \psi(\mathbf{x}_1) = \int \psi_j(\mathbf{x}_2)^* \psi_j(\mathbf{x}_2) \frac{1}{r_{ij}} d\mathbf{x}_2 \psi(\mathbf{x}_1), \quad (1.37)$$

and

$$\hat{K}'_j \psi(\mathbf{x}_1) = \int \psi_j(\mathbf{x}_2)^* \psi(\mathbf{x}_2) \frac{1}{r_{ij}} d\mathbf{x}_2 \psi_j(\mathbf{x}_1), \quad (1.38)$$

are the Coulomb and exchange operators of the electrons. This greatly reduces the variation term

$$\delta \langle \Psi | \hat{H} | \Psi \rangle = \sum_{i=1}^N (\langle \delta \psi_i | \hat{F} | \psi_i \rangle + \langle \psi_i | \hat{F} | \delta \psi_i \rangle) = 0 \quad (1.39)$$

If we only consider variations on a single spin orbital that

$$\delta \psi_i = c \psi_\lambda, \quad \delta \psi_k = 0, \quad k \neq i \quad (1.40)$$

that is, only if the functions is divided into an occupied function space of the form

$$\psi_I = \sum_{k=1}^N c_k \psi_k \quad (1.41)$$

and a complementary function space of the unoccupied orbitals

$$\psi_{II} = \sum_{\mu=N+1}^{\infty} c_\mu \psi_\mu. \quad (1.42)$$

Spin orbitals belonging to the occupied function space are denoted with Roman subscripts and the unoccupied spin orbitals are denoted with Greek subscripts accordingly. The variations of equation 1.40 correspond to variations of a single spin orbital into the unoccupied function space. Inserting this into equation 1.39 gives

$$c^* \langle \psi_\lambda | \hat{F} | \psi_i \rangle + c \langle \psi_i | \hat{F} | \psi_\lambda \rangle = 0, \quad (1.43)$$

and replacing  $ic$  and multiplying by  $i$  gives

$$c^* \langle \psi_\lambda | \hat{F} | \psi_i \rangle - c \langle \psi_i | \hat{F} | \psi_\lambda \rangle = 0. \quad (1.44)$$

This implies that both terms must cancel out. This result holds for every combination of occupied and unoccupied spin orbitals, and thus

$$\hat{F} \psi_i = \sum_{k=1}^N \epsilon_{ki} \psi_k + \sum_{\mu=N+1}^{\infty} \epsilon_{\mu i} \psi_\mu \quad (1.45)$$

must terminate after the  $N$ 'th term. The coefficients can be determined as

$$\epsilon_{ki} = \langle \psi_k | \hat{F} | \psi_i \rangle, \quad (1.46)$$

$$\epsilon_{\mu i} = \langle \psi_\mu | \hat{F} | \psi_i \rangle, \quad (1.47)$$

but the second one vanish due to equation 1.43 and equation 1.44. Lastly we arrive at the Hartree-Fock equations

$$\hat{F} \psi_i = \sum_{k=1}^N \epsilon_{ki} \psi_k, \quad i = 1, 2, \dots, N. \quad (1.48)$$

The eigenvalues  $\epsilon_{ij}$  from equation 1.48 is called *Lagrangian multipliers* which can be derived from the requirement that

$$I = \langle \Psi | \hat{H} | \Psi \rangle - \sum_{i=1}^N \sum_{j=1}^N \epsilon_{ij} \langle \psi_i | \psi_j \rangle \quad (1.49)$$

should be minimum for all variations of  $\psi_i$ 's and  $\psi_j$ 's.

By a unitary transformation of a matrix with elements  $\epsilon_{ij}$  and by choosing a suitable linear combination of spin orbitals the Hartree-Fock equations can take on the following canonical form

$$\hat{F}\psi_i = \epsilon_i \psi_i. \quad (1.50)$$

By assuming the spin orbitals are eigenfunctions of  $\hat{s}_z$  and that the Slater determinant is ordered so that the  $n$  first spin orbitals are spin up and the following  $N - n$  are spin down

$$\Psi(x_1, x_2, \dots, x_N) = |\psi_1^+ \psi_2^+ \dots \psi_{n-1}^+ \psi_n^+ \psi_{n+1}^- \dots \psi_{N-1}^- \psi_N^-| \quad (1.51)$$

Integrating over the spin coordinate in the Coulomb and exchange operators gives

$$\hat{J}'_j \psi(\mathbf{x}_1) = \hat{J}_j \psi(\mathbf{r}_1) \quad \hat{K}'_j \psi(\mathbf{x}_1) = \delta(m_{si}, m_{sj}) \hat{K}_j \psi(\mathbf{r}_1) \quad (1.52)$$

and thus there is no exchange interaction between electrons with opposite spins. When the Fock operator operates on a spin up orbital it only counts exchange interactions from the first  $n$  terms and vice versa. The Hartree-Fock equation yields two solutions

$$\hat{F}^\uparrow \psi_i = \epsilon_i \psi_i. \quad i = 1, 2, \dots, n \quad (1.53)$$

$$\hat{F}^\downarrow \psi_i = \epsilon_i \psi_i. \quad i = n + 1, n + 2, \dots, N \quad (1.54)$$

The Fock operators under spin restrictions are then

$$\hat{F}^\uparrow = \hat{h} + \sum_{j=1}^N \hat{J}'_j - \sum_{j=1}^n \hat{K}'_j \quad (1.55)$$

$$\hat{F}^\downarrow = \hat{h} + \sum_{j=1}^N \hat{J}'_j - \sum_{j=n+1}^N \hat{K}'_j \quad (1.56)$$

The physical significance of the eigenvalues  $\epsilon_i$  of equation 1.50 can be examined by determining them through equation 1.50 by left multiplication and integrating obtaining

$$\epsilon_i = \langle \psi_i | \hat{h} | \psi_i \rangle + \sum_{j=1}^N \langle \psi_i | \hat{J}'_j - \hat{K}'_j | \psi_i \rangle \quad (1.57)$$

If compared to equation 1.33 then it is easily recognized that the energy is found by

$$E = \sum_{i=1}^N \epsilon_i - \sum_{i < j} \langle \psi_i | \hat{J}'_j - \hat{K}'_j | \psi_i \rangle \quad (1.58)$$

or accordingly

$$E = \frac{1}{2} \sum_{i=1}^N (\epsilon_i + \langle \psi_i | \hat{h}_i | \psi_i \rangle). \quad (1.59)$$

The Hartree-Fock equations have now been derived, however in their present form they are still very hard to implement in order to get decent results. One way of solving this problem is by introducing the Hartree-Fock-Roothaan equations.

## 1.7 The Hartree-Fock-Roothaan equations

Solving multiple electron problems is generally an analytically impossible task. Clemens C. J. Roothaan sought to find an approximating solution where each of the spin orbitals can be expressed as a linear combination of hydrogenic orbitals

$$\psi_\mu = \sum_p c_{p\mu} \varphi_{p\mu}(\mathbf{x}). \quad (1.60)$$

The subscript  $\mu$  is short for the set of quantum numbers describing the given electron. The goal is to determine the coefficients or weights of the individual hydrogenic orbitals. By taking the Hartree-Fock operator of equation 1.55 and letting it operate on  $\psi_\mu$  yields

$$\hat{h}\psi_\mu + \sum_{i=1}^N \hat{J}_i \psi_\mu - \sum_{i=1}^n \hat{K}_i \psi_\mu = \epsilon_i \psi_\mu. \quad (1.61)$$

By writing  $\psi_\mu$  as in equation 1.60 and left multiplying by  $\langle \varphi_{p\mu} |$  we obtain the following relation

$$\begin{aligned} \sum_q c_{q\mu} \langle \varphi_{p\mu} | \hat{h} | \varphi_{q\mu} \rangle + \sum_q \sum_{i=1}^N c_{q\mu} \langle \varphi_{p\mu} | \hat{J}_i | \varphi_{q\mu} \rangle - \sum_q \sum_{i=1}^n c_{q\mu} \langle \varphi_{p\mu} | \hat{K}_i | \varphi_{q\mu} \rangle \\ = \epsilon_\mu \sum_q c_{q\mu} \langle \varphi_{p\mu} | \varphi_{q\mu} \rangle. \end{aligned} \quad (1.62)$$

By introducing the matrix elements

$$h_\mu^{pq} = \langle \varphi_{p\mu} | \hat{h} | \varphi_{q\mu} \rangle \quad (1.63)$$

$$S_\mu^{pq} = \langle \varphi_{p\mu} | \varphi_{q\mu} \rangle \quad (1.64)$$

$$J_\mu^{pq} = \sum_{i=1}^N \langle \varphi_{p\mu} | \hat{J}_i | \varphi_{q\mu} \rangle \quad (1.65)$$

$$K_\mu^{pq} = \sum_{i=1}^n \langle \varphi_{p\mu} | \hat{K}_i | \varphi_{q\mu} \rangle \quad (1.66)$$

the relation becomes

$$\sum_q c_{q\mu} h_\mu^{pq} + \sum_q c_{q\mu} J_\mu^{pq} - \sum_q c_{q\mu} K_\mu^{pq} = \epsilon_\mu \sum_q c_{q\mu} S_\mu^{pq}, \quad (1.67)$$

where each choice of  $p$  gives a new equation. The collected set of equations for  $p$  and  $q$  can be described by a matrix equation

$$\mathbf{F}_\mu \mathbf{c}_\mu = \epsilon_\mu \mathbf{S}_\mu \mathbf{c}_\mu \quad (1.68)$$

where  $\mathbf{F}_\mu = \mathbf{h}_\mu + \mathbf{J}_\mu - \mathbf{K}_\mu$ . Similar derivation can be performed for the opposite spin. This is a generalised eigenvalue problem where  $\epsilon_\mu$  is the eigenvalue for the linear combinations determined by the expansion coefficients contained in  $\mathbf{c}_\mu$ . The eigenvalues are by some referred to as orbital energies [13].

Since the basis is not necessarily orthonormal the equations has to be properly normalised as determined by

$$\mathbf{c}_\mu \mathbf{S}_\mu \mathbf{c}_\mu = 1. \quad (1.69)$$

We are now ready to implement the Hartree-Fock-Roothaan theory and thereby calculate the electron energies starting with atoms. This consists of choosing an appropriate basis and make it possible to calculate the necessary matrix elements.

## Chapter 2

# Implementation for atoms

Now that the general Hartree-Fock theory have been derived and improved using the Hartree-Fock-Roothaan equations, it is possible to use the equations to calculate the electron energy of the atoms. In this chapter the implementation of the Hartree-Fock theory will be described and results of electronic energies of the elements will be presented and discussed.

As mentioned in 1.7 the implementation of the equations will be a lot easier if a known basis is used such that the problem reduces to solving eigenvalue problems. In this project we use the Gaussian basis functions [2].

### 2.1 Gaussian basis functions

The known basis function used in equation 1.60 can generally be written as

$$\varphi_{i\mu}(\mathbf{x}) = R_{nl}(r) Y_{lm}(\theta, \phi) \eta_{\gamma_s}(\varsigma), \quad (2.1)$$

where  $R_{nl}(r)$  contains the radial part of the atomic orbital,  $Y_{lm}(\theta, \phi)$  a spherical harmonic function and  $\eta_{\gamma_s}(\varsigma)$  a function taking care of the spin. The spherical harmonics and spin functions are often well defined, however the use of the radial part of the orbital function vary from method to method. Different type of functions have both advantages and disadvantages. As mentioned we implement Gaussian functions as the radial part of the known basis functions. The advantages of the Gaussian functions is that the matrix elements are easier calculated however the functions does not have the close nor long range behaviour of a proper orbital. We compensate for this by using several different Gaussian functions. The un-normalised Gaussian function of an electron centred around atom  $i$  positioned at  $\mathbf{R}_i$  may be written

$$R_{nl}(\mathbf{r}) = r^l e^{-b_i|\mathbf{r}-\mathbf{R}_i|^2}. \quad (2.2)$$

The reason why the computation of the matrix elements are much easier using Gaussian functions as the known basis is due to the Gaussian product theorem. This theorem states multiplying for example two  $s$ -type orbitals yields

$$\begin{aligned} R_{s,i}(\mathbf{r}) R_{s,j}(\mathbf{r}) &= e^{-b_i|\mathbf{r}-\mathbf{R}_i|^2} e^{-b_j|\mathbf{r}-\mathbf{R}_j|^2} \\ &= e^{-\gamma_{ij}} e^{-b_{ij}|\mathbf{r}-\mathbf{R}_{ij}|^2}, \end{aligned} \quad (2.3)$$

where

$$b_{ij} = b_i + b_j, \quad \gamma_{ij} = \frac{b_i b_j}{b_{ij}} |\mathbf{R}_i - \mathbf{R}_j|^2, \quad \mathbf{R}_{ij} = \frac{b_i \mathbf{R}_i + b_j \mathbf{R}_j}{b_{ij}}. \quad (2.4)$$

This means that the product of two Gaussian functions located on different atoms still is a Gaussian function located on a virtual point in between the two atoms.

Having determined the appropriate basis function we are now ready to calculate the atomic matrix elements.

## 2.2 The matrix elements

The four different matrix elements to be calculated are given by equation 1.63 through equation 1.66. Only the last two integrals are troublesome due to the  $1/r_{ij}$  term. Using Gaussian basis functions the overlap and single electron energy integrals can be calculated straightforward to

$$h_{\mu}^{ij} = -\frac{1}{2} (b_{i,nl} + b_{j,nl})^{-5/2-l} ((b_{i,nl} + b_{j,nl})^{3/2} Z \Gamma(l+1) - 2b_{i,nl} b_{j,nl} \Gamma(5/2+l)) \quad (2.5)$$

$$S_{\mu}^{ij} = \frac{1}{2} (b_{i,nl} + b_{j,nl})^{-3/2-l} \Gamma(3/2+l), \quad (2.6)$$

where  $l$  is the azimuthal quantum number of the electron,  $Z$  the nucleus electronic charge and  $\Gamma(x)$  is the Gamma function. However in order to evaluate the two electron matrix elements it is an advantage to express  $1/r_{ij}$  in terms of the integration variables. Often the reciprocal distance is expanded in terms of spherical harmonics [13]. Doing so the term is given by

$$\frac{1}{r_{ij}} = \sum_{l,m} \frac{4\pi}{2l+1} Y_{lm}^*(\theta_1, \phi_1) Y_{lm}(\theta_2, \phi_2) \frac{r_{\min}^l}{r_{\max}^{l+1}}, \quad (2.7)$$

where  $r_{\min} = \min(r_1, r_2)$  and  $r_{\max} = \max(r_1, r_2)$ . Using the Gaussian basis functions and equation 2.7 the matrix elements may be expressed as

$$J_{\mu\lambda}^{ij} = \sum c_{i\lambda} c_{j\lambda} \sum_{l,m} \frac{4\pi}{2l+1} \langle Y_{l_1, m_1} | Y_{lm}^*(\theta_1, \phi_1) | Y_{l_1, m_1} \rangle \\ \times \langle Y_{l_2, m_2} | Y_{lm}(\theta_2, \phi_2) | Y_{l_2, m_2} \rangle \mathbb{J} \quad (2.8)$$

and

$$K_{\mu\lambda}^{ij} = \sum c_{i\lambda} c_{j\lambda} \sum_{l,m} \frac{4\pi}{2l+1} \langle Y_{l_1, m_1} | Y_{lm}^*(\theta_1, \phi_1) | Y_{l_2, m_2} \rangle \\ \times \langle Y_{l_2, m_2} | Y_{lm}(\theta_2, \phi_2) | Y_{l_1, m_1} \rangle \mathbb{K}, \quad (2.9)$$

where  $\mathbb{J}$  and  $\mathbb{K}$  are integrals involving the radial part of the wavefunctions and are given by

$$\mathbb{J} = \int_0^\infty \int_0^\infty r_1^{2l_1+2} r_2^{2l_2+2} e^{-(b_{i,n_1l_1}+b_{j,n_1l_1})r_1^2} e^{-(b_{p,n_2l_2}+b_{q,n_2l_2})r_2^2} \frac{r_{\min}^l}{r_{\max}^{l+1}} dr_1 dr_2 \quad (2.10)$$

$$\mathbb{K} = \int_0^\infty \int_0^\infty r_1^{l_1+l_2+2} r_2^{l_1+l_2+2} e^{-(b_{i,n_1l_1}+b_{p,n_2l_2})r_1^2} e^{-(b_{j,n_2l_2}+b_{q,n_1l_1})r_2^2} \frac{r_{\min}^l}{r_{\max}^{l+1}} dr_1 dr_2. \quad (2.11)$$

A way of evaluating these integrals is by splitting one of the integrals into two separate integrals. One going from 0 to  $r_1$  and another going from  $r_1$  to  $\infty$ . In this way  $r_{\min} = r_2$  and  $r_{\max} = r_1$  in the first integral and vice versa in the other and the radial integrals may then be written

$$\begin{aligned} \mathbb{J} = \int_0^\infty r_1^{2l_1+2} e^{-(b_{i,n_1l_1}+b_{j,n_1l_1})r_1^2} & \left( \int_0^{r_1} r_2^{2l_2+2} e^{-(b_{p,n_2l_2}+b_{q,n_2l_2})r_2^2} \frac{r_2^l}{r_1^{l+1}} dr_2 \right. \\ & \left. + \int_{r_1}^\infty r_2^{2l_2+2} e^{-(b_{p,n_2l_2}+b_{q,n_2l_2})r_2^2} \frac{r_1^l}{r_2^{l+1}} dr_2 \right) dr_1 \end{aligned} \quad (2.12)$$

$$\begin{aligned} \mathbb{K} = \int_0^\infty r_1^{l_1+l_2+2} e^{-(b_{i,n_1l_1}+b_{j,n_2l_2})r_1^2} & \left( \int_0^{r_1} r_2^{l_1+l_2+2} e^{-(b_{q,n_2l_2}+b_{p,n_1l_1})r_2^2} \frac{r_2^l}{r_1^{l+1}} dr_2 \right. \\ & \left. + \int_{r_1}^\infty r_2^{l_1+l_2+2} e^{-(b_{q,n_2l_2}+b_{p,n_1l_1})r_2^2} \frac{r_1^l}{r_2^{l+1}} dr_2 \right) dr_1 \end{aligned} \quad (2.13)$$

From here the matrix elements may be calculated using a CAS-program. We have used Mathematica, and the calculation of the matrix elements can be seen in appendix A.

## 2.3 Self Consistent Field procedure

In the Hartree-Fock theory equation 1.68 is often solved using the Self Consistent Field method. This is due to the fact that in order to create the Fock-operator used in equation 1.68 one actually needs a solution to the problem. In the Self Consistent Field method one starts with a reasonable guess of the desired wave functions and from these construct the Fock-operator and solve equation 1.68. Using the obtained solution one creates another Fock-operator and from it obtain a new solution. The method stops when the difference between two consecutive solutions is insignificant. However in order to reach this convergence of the solutions one might implement effective convergence schemes.

## 2.4 Convergence schemes

A great effort has been put into the problem of making the solutions of Hartree-Fock problems converge faster. This is of course of computational interest since

the convergence can be quite time consuming but iterations can also approach an infinite loop between two or more solutions. In order to overcome this problem we have implemented Pulay mixing named after Peter Pulay who introduced direct inversion of the iterative subspace (DIIS) or simply Pulay mixing in order to accelerate and stabilise the convergence of Hartree-Fock methods [9].

### Pulay mixing

After each iteration we obtain a set of vectors of coefficients that in a linear combination determined the weight of the basis function, of which the coefficient represents. Let the solution coefficient vector at iteration  $i$  be denoted  $\mathbf{c}_i$ . As the solution increments through iterations we consider a residuum vector at step  $i$  defined as

$$\Delta\mathbf{c}_i = \mathbf{c}_{i+1} - \mathbf{c}_i. \quad (2.14)$$

Pulay made the assumption that the final solution coefficient vector,  $\mathbf{c}_f$ , can be approximated as a linear combination of the previous

$$\mathbf{c}_f = \sum_i^m p_i \mathbf{c}_i \quad (2.15)$$

where the size of  $m$  determines the amount of previous steps to be taken into consideration. In practise this number rarely exceeds the number 5 as there has been shown no considerable improvement using more previous steps. The coefficients  $p_i$  are obtained by requiring that the associated residual vector

$$\Delta\mathbf{c} = \sum_i^m p_i (\Delta\mathbf{c}_i) \quad (2.16)$$

approaches zero in a least-squares sense. We also need the constraint that the coefficients  $p_i$  add to one

$$1 = \sum_i^m p_i. \quad (2.17)$$

In order to accept this last requirement we can write each of the solutions,  $\mathbf{c}_i$ , as a sum between the final solution and an error term

$$\mathbf{c}_i = \mathbf{c}_f + \mathbf{e}_i \quad (2.18)$$

Putting this into equation 2.15 yields

$$\begin{aligned} \mathbf{c}_f &= \sum_i^m p_i (\mathbf{c}_f + \mathbf{e}_i) \\ &= \mathbf{c}_f \sum_i^m p_i + \sum_i^m p_i \mathbf{e}_i \end{aligned} \quad (2.19)$$

Through the iterations the second term should vanish as the error approaches zero. This leaves only the first term so in order to make the above relation valid the coefficients must sum to one as assumed.



We wish to minimise the norm of the residuum vector

$$\langle \Delta \mathbf{c} | \Delta \mathbf{c} \rangle = \sum_{ij}^m p_i^* p_j \langle \Delta \mathbf{c}_i | \Delta \mathbf{c}_j \rangle \quad (2.20)$$

Exchanging  $B_{ij} = \langle \Delta \mathbf{c}_i | \Delta \mathbf{c}_j \rangle$ , the requirements can be satisfied by minimizing the following function with Lagrangian multiplier,  $\lambda$

$$\mathcal{L} = \sum_{ij}^m p_i^* B_{ij} p_j - \lambda \left( 1 - \sum_i^m p_i \right) \quad (2.21)$$

Assuming real quantities, we can minimize  $\mathcal{L}$  with respect to a coefficient  $p_k$  to obtain

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial p_k} = 0 &= \sum_i^m p_i B_{ik} + \sum_j^m p_j B_{kj} - \lambda \\ &= 2 \sum_i^m p_i B_{ik} - \lambda \end{aligned} \quad (2.22)$$

We can absorb the factor of 2 into  $\lambda$  to obtain the following matrix equation

$$\begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1m} & -1 \\ B_{21} & B_{22} & \cdots & B_{2m} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ B_{m1} & B_{m2} & \cdots & B_{mm} & -1 \\ -1 & -1 & \cdots & -1 & 0 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix}. \quad (2.23)$$

Solving this matrix equation yields the optimum linear combination coefficients for the previous term coefficients in order to minimise the residuum vector. Thus the new predicted optimal term coefficient vector can be found by using equation 2.15.

Using Pulay mixing and the calculated matrix functions we are ready to calculate the electron binding energies of the elements.

## 2.5 Results

The script used for the calculations is presented in appendix C.1

The atomic computations has been carried out using a set of uncontracted Gaussian basis set developed by [5], that has been shown to be chemically reliable even in larger molecules.

The results are shown in table form through table 2.1. The results are compared with those of [5] showing some quite accurate resemblance. The energy difference are presented in figure 2.1 showing a maximum energy difference of around 1 Hartree.

$Z$	Atom name	Configuration	$E$ (Computed)	$E$ (From [5])
1	Hydrogen	1s <sup>1</sup>	-0.49994557	-0.49994557

Table 2.1: Computed atomic energies from H through L compared with the calculations of [5]. Energy units are in Hartrees (Continued on next page).

$Z$	Atom name	Configuration	$E$ (Computed)	$E$ (From [5])
2	Helium	$1s^2$	-2.86115334	-2.86115334
3	Lithium	$1s^2 2s^1$	-7.43271968	-7.43269569
4	Beryllium	$1s^2 2s^2$	-14.5729681	-14.5729681
5	Boron	$1s^2 2s^2 2p^1$	-24.4144654	-24.5289676
6	Carbon	$1s^2 2s^2 2p^2$	-37.5310547	-37.6884715
7	Nitrogen	$1s^2 2s^2 2p^3$	-54.4042654	-54.4007133
8	Oxygen	$1s^2 2s^2 2p^4$	-74.6191049	-74.8090732
9	Fluorine	$1s^2 2s^2 2p^5$	-99.1639672	-99.4088900
10	Neon	$1s^2 2s^2 2p^6$	-128.546472	-128.546472
11	Sodium	[Ne] $3s^1$	-161.858600	-161.858570
12	Magnesium	[Ne] $3s^2$	-199.614215	-199.614215
13	Aluminum	[Ne] $3s^2 3p^1$	-241.802199	-241.876368
14	Silicon	[Ne] $3s^2 3p^2$	-288.757442	-288.853976
15	Phosphorus	[Ne] $3s^2 3p^3$	-340.718822	-340.718336
16	Sulfur	[Ne] $3s^2 3p^4$	-397.384664	-397.504352
17	Chlorine	[Ne] $3s^2 3p^5$	-459.338687	-459.481433
18	Argon	[Ne] $3s^2 3p^6$	-526.816781	-526.816781
19	Potassium	[Ar] $4s^1$	-599.164348	-599.164292
20	Calcium	[Ar] $4s^2$	-676.757668	-676.757668
21	Scandium	[Ar] $4s^2 3d^1$	-759.553865	-759.735143
22	Titanium	[Ar] $4s^2 3d^2$	-848.054450	-848.405360
23	Vanadium	[Ar] $4s^2 3d^3$	-942.482641	-942.883632
24	Chromium	[Ar] $4s^1 3d^5$	-1043.35589	-1043.35553
25	Manganese	[Ar] $4s^2 3d^5$	-1149.86888	-1149.86538
26	Iron	[Ar] $4s^2 3d^6$	-1262.18252	-1262.44269
27	Cobalt	[Ar] $4s^2 3d^7$	-1380.93099	-1381.41347
28	Nickel	[Ar] $4s^2 3d^8$	-1506.33054	-1506.86971
29	Copper	[Ar] $4s^1 3d^{10}$	-1638.96277	-1638.96228
30	Zinc	[Ar] $4s^2 3d^{10}$	-1777.84664	-1777.84664
31	Gallium	[Ar] $4s^2 3d^{10} 4p^1$	-1923.18595	-1923.26029
32	Germanium	[Ar] $4s^2 3d^{10} 4p^2$	-2075.26686	-2075.35902
33	Arsenic	[Ar] $4s^2 3d^{10} 4p^3$	-2234.23911	-2234.23794
34	Selenium	[Ar] $4s^2 3d^{10} 4p^4$	-2399.75947	-2399.86687
35	Bromine	[Ar] $4s^2 3d^{10} 4p^5$	-2572.31642	-2572.44056
36	Krypton	[Ar] $4s^2 3d^{10} 4p^6$	-2752.05419	-2752.05419
37	Rubidium	[Kr] $5s^1$	-2938.35681	-2938.35673
38	Strontium	[Kr] $5s^2$	-3131.54500	-3131.54500
39	Yttrium	[Kr] $5s^2 4d^1$	-3331.55545	-3331.68347
40	Zirconium	[Kr] $5s^2 4d^2$	-3538.75135	-3538.99434
41	Niobium	[Kr] $5s^1 4d^4$	-3753.43518	-3753.59688
42	Molybdenum	[Kr] $5s^1 4d^5$	-3975.55206	-3975.54863
43	Technetium	[Kr] $5s^2 4d^5$	-4204.79397	-4204.78792
44	Ruthenium	[Kr] $5s^1 4d^7$	-4441.23215	-4441.53853
45	Rhodium	[Kr] $5s^1 4d^8$	-4685.53924	-4685.88071
46	Palladium	[Kr] $4d^{10}$	-4937.91980	-4937.91980
47	Silver	[Kr] $5s^1 4d^{10}$	-5197.69786	-5197.69740
48	Cadmium	[Kr] $5s^2 4d^{10}$	-5465.13210	-5465.13210
49	Indium	[Kr] $5s^2 4d^{10} 5p^1$	-5740.10075	-5740.16830
50	Tin	[Kr] $5s^2 4d^{10} 5p^2$	-6022.84999	-6022.93086
51	Antimony	[Kr] $5s^2 4d^{10} 5p^3$	-6313.48607	-6313.48449
52	Tellurium	[Kr] $5s^2 4d^{10} 5p^4$	-6611.69122	-6611.78321
53	Iodine	[Kr] $5s^2 4d^{10} 5p^5$	-6917.87550	-6917.98003
54	Xenon	[Kr] $5s^2 4d^{10} 5p^6$	-7232.13748	-7232.13748
55	Cesium	[Xe] $6s^1$	-7553.93311	-7553.93302
56	Barium	[Xe] $6s^2$	-7883.54325	-7883.54325
57	Lanthanum	[Xe] $6s^2 5d^1$	-8220.95071	-8221.06620
58	Cerium	[Xe] $6s^2 4f^1 5d^1$	-8566.37167	-8566.87215
59	Praseodymium	[Xe] $6s^2 4f^3$	-8920.39371	-8921.18034
60	Neodymium	[Xe] $6s^2 4f^4$	-9283.04490	-9283.88222
61	Promethium	[Xe] $6s^2 4f^5$	-9654.39094	-9655.09820

Table 2.1: Computed atomic energies from H through L compared with the calculations of [5]. Energy units are in Hartrees (Continued on next page).

$Z$	Atom name	Configuration	$E$ (Computed)	$E$ (From [5])
62	Samarium	[Xe]6s <sup>2</sup> 4f <sup>6</sup>	-10034.5278	-10034.9517
63	Europium	[Xe]6s <sup>2</sup> 4f <sup>7</sup>	-10423.5496	-10423.5421
64	Gadolinium	[Xe]6s <sup>2</sup> 4f <sup>7</sup> 5d <sup>1</sup>	-10820.5365	-10820.6604
65	Terbium	[Xe]6s <sup>2</sup> 4f <sup>9</sup>	-11225.8464	-11226.5674
66	Dysprosium	[Xe]6s <sup>2</sup> 4f <sup>10</sup>	-11640.4860	-11641.4515
67	Holmium	[Xe]6s <sup>2</sup> 4f <sup>11</sup>	-12064.2689	-12065.2887
68	Erbium	[Xe]6s <sup>2</sup> 4f <sup>12</sup>	-12497.2944	-12498.1516
69	Thulium	[Xe]6s <sup>2</sup> 4f <sup>13</sup>	-12939.6584	-12940.1731
70	Ytterbium	[Xe]6s <sup>2</sup> 4f <sup>14</sup>	-13391.4548	-13391.4548
71	Lutecium	[Xe]6s <sup>2</sup> 4f <sup>14</sup> 5d <sup>1</sup>	-13851.6806	-13851.8067
72	Hafnium	[Xe]6s <sup>2</sup> 4f <sup>14</sup> 5d <sup>2</sup>	-14321.0157	-14321.2486
73	Tantalum	[Xe]6s <sup>2</sup> 4f <sup>14</sup> 5d <sup>3</sup>	-14799.5544	-14799.8114
74	Tungsten	[Xe]6s <sup>2</sup> 4f <sup>14</sup> 5d <sup>4</sup>	-15287.3729	-15287.5451
75	Rhenium	[Xe]6s <sup>2</sup> 4f <sup>14</sup> 5d <sup>5</sup>	-15784.5427	-15784.5320
76	Osmium	[Xe]6s <sup>2</sup> 4f <sup>14</sup> 5d <sup>6</sup>	-16290.4713	-16290.6474
77	Iridium	[Xe]6s <sup>2</sup> 4f <sup>14</sup> 5d <sup>7</sup>	-16805.8003	-16806.1119
78	Platinum	[Xe]6s <sup>1</sup> 4f <sup>14</sup> 5d <sup>9</sup>	-17330.8587	-17331.0686
79	Gold	[Xe]6s <sup>1</sup> 4f <sup>14</sup> 5d <sup>10</sup>	-17865.3992	-17865.3987
80	Mercury	[Xe]6s <sup>2</sup> 4f <sup>14</sup> 5d <sup>10</sup>	-18408.9902	-18408.9902
81	Tallium	[Xe]6s <sup>2</sup> 4f <sup>14</sup> 5d <sup>10</sup> 6p <sup>1</sup>	-18961.7587	-18961.8237
82	Lead	[Xe]6s <sup>2</sup> 4f <sup>14</sup> 5d <sup>10</sup> 6p <sup>2</sup>	-19523.9305	-19524.0070
83	Bismuth	[Xe]6s <sup>2</sup> 4f <sup>14</sup> 5d <sup>10</sup> 6p <sup>3</sup>	-20095.5875	-20095.5854
84	Polonium	[Xe]6s <sup>2</sup> 4f <sup>14</sup> 5d <sup>10</sup> 6p <sup>4</sup>	-20676.4142	-20676.4999
85	Astatine	[Xe]6s <sup>2</sup> 4f <sup>14</sup> 5d <sup>10</sup> 6p <sup>5</sup>	-21266.7841	-21266.8807
86	Radon	[Xe]6s <sup>2</sup> 4f <sup>14</sup> 5d <sup>10</sup> 6p <sup>6</sup>	-21866.7713	-21866.7713
87	Francium	[Ra]7s <sup>1</sup>	-22475.8581	-22475.8580
88	Radium	[Ra]7s <sup>2</sup>	-23094.3030	-23094.3030
89	Actinium	[Ra]7s <sup>2</sup> 6d <sup>1</sup>	-23722.0873	-23722.1915
90	Thorium	[Ra]7s <sup>2</sup> 6d <sup>2</sup>	-24359.4372	-24359.6219
91	Protoactinium	[Ra]7s <sup>2</sup> 5f <sup>2</sup> 6d <sup>1</sup>	-25006.5117	-25007.1092
92	Uranium	[Ra]7s <sup>2</sup> 5f <sup>3</sup> 6d <sup>1</sup>	-25663.5826	-25664.3376
93	Neptunium	[Ra]7s <sup>2</sup> 5f <sup>4</sup> 6d <sup>1</sup>	-26330.6626	-26331.4543
94	Plutonium	[Ra]7s <sup>2</sup> 5f <sup>6</sup>	-27008.4196	-27008.7186
95	Americium	[Ra]7s <sup>2</sup> 5f <sup>7</sup>	-27695.8997	-27695.8864
96	Curium	[Ra]7s <sup>2</sup> 5f <sup>7</sup> 6d <sup>1</sup>	-28392.6577	-28392.7704
97	Berkelium	[Ra]7s <sup>2</sup> 5f <sup>9</sup>	-29099.5106	-29099.8308
98	Californium	[Ra]7s <sup>2</sup> 5f <sup>10</sup>	-29816.6874	-29817.4180
99	Einsteinium	[Ra]7s <sup>2</sup> 5f <sup>11</sup>	-30544.2078	-30544.9712
100	Fermium	[Ra]7s <sup>2</sup> 5f <sup>12</sup>	-31282.1408	-31282.7766
101	Mendelevium	[Ra]7s <sup>2</sup> 5f <sup>13</sup>	-32030.5533	-32030.9320
102	Nobelium	[Ra]7s <sup>2</sup> 5f <sup>14</sup>	-32789.5111	-32789.5111
103	Lawrencium	[Ra]7s <sup>2</sup> 5f <sup>14</sup> 7p <sup>1</sup>	-33557.6110	-33557.9495

Table 2.1: Computed atomic energies from H through L compared with the calculations of [5]. Energy units are in Hartrees (Continued from previous pages).

## 2.6 Discussion

When studying table 2.1 it is immediately clear that the results are in very good agreement to the calculated values of [5]. However there are some atoms where the difference is rather large. Atoms like Holmium where the difference is about one Ha. To gain a better overview we plot the difference as a function of the atomic number. The result of this can be seen in figure 2.1. One might think more electrons leads to larger deviations, but figure 2.1 proves that this is

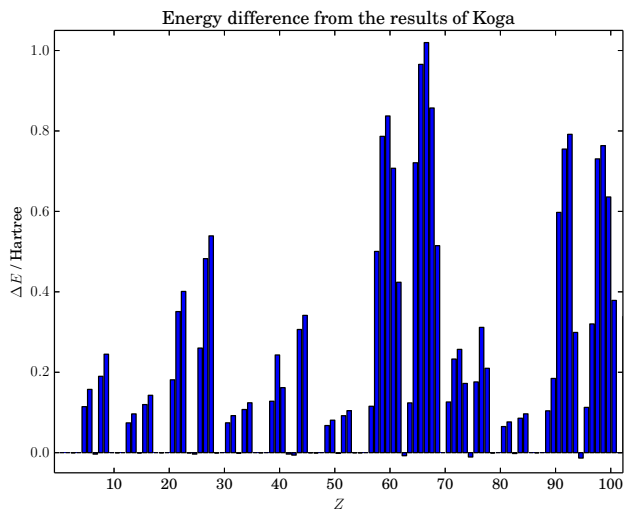


Figure 2.1: Energy difference between our computed atomic energies and those of [5]. The atoms are sorted after element number.

not the case. Instead the main reason why the calculations for some atoms are inaccurate is due to an approximation used in the implementation. To simplify the calculations we have adopted what can be called a spherical average. By adopting the spherical average we do not distinguish electrons with the same angular momentum projections. To investigate this inaccuracy we have plotted the differences as a function of the total angular momentum. The result can be seen in figure 2.2. As seen in the figure the calculations become increasingly inaccurate as the total angular momentum increases. What is also noticed is that when the total angular momentum is zero, physically meaning that the atom has filled shells, the calculations become exact. This is due to the fact that when the total angular momentum is zero the spherical average approximation becomes exact.

The Hartree-Fock method is not limited by the total charge of the given problem. To investigate this the next section calculates the electron energy of the anions and cations of the 52 first elements.

## 2.7 Ions

Using the Hartree-Fock method virtually any configuration of core charge and number of electrons can be applied. Optimal basis sets for ions can be found in [6], but in this section we implemented another, simpler approach for our ionic calculations.

In order to keep the task of finding the appropriate electron configuration as simple as possible, we loaded the basis set and electron configuration for the appropriate atom belonging to the number of electrons and then corrected the nuclear charge. If the ion to be computed was  $Z^n$ , where  $n$  is the net charge of the ion, we loaded the electron configuration and basis set of the neutral atom

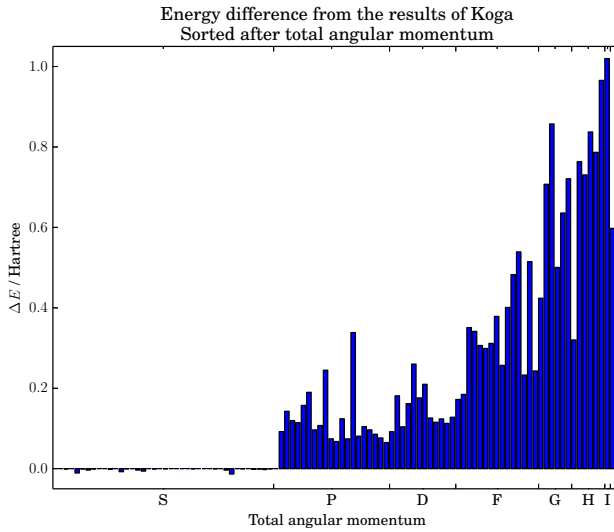


Figure 2.2: Energy difference between our computed atomic energies and those of [5]. The atoms are sorted after total angular momentum.

$(Z - n)$  and corrected the nuclear charge to that of  $Z$ .

We realised that this implementation was not exactly too accurate in describing the ions for the following reason. As the charge of the core is increased so is the coulomb attraction between electrons and core, thus pulling the electrons towards the center and vice versa if the core charge is reduced. This may not be depicted in the used basis, since this is optimised for the neutral atoms. Our solution to this was to add 4 basis functions to each azimuthal basis, two diffuse exponents  $\frac{1}{2}b_{l,\min}$  and  $\frac{1}{4}b_{l,\min}$ , where  $b_{l,\min}$  was the former smallest Gaussian exponent for the azimuthal value  $l$ . Likewise we added  $2b_{l,\min}$  and  $4b_{l,\min}$  for better depiction of the electrons close to the core.

Results are given in table 2.2. The Numerical Hartree-Fock limit values (NMF) is given in [6] which also yielded more accurate results for the energies. However, our results shows how the case of neutral atoms can be put into the case of ions in a simple matter. Also many of the results are accurate down to the 5th or 6th figure, so the approximations done to the electron configurations and basis is not entirely in error.

Further research need to be conducted to determine the reasons for the errors. Are they related to electron configurations in error, or are they manifestations of the spherical average as in the atoms?

Z	Symbol	Energies (Hartree)			
		Anions		Cations	
		Our results	NMF	Our results	NMF
2	He	-1.998686244		-2.829773837	
3	Li	-7.232887583	-7.236415201	-7.405404957	-7.428232061

Table 2.2: Computed ionic energies from H through Te compared with the calculations of [6]. Energy units are in Hartrees. NMF values are the numerical Hartree-Fock limit of the given configuration (Continued on next page)

Z	Symbol	Energies (Hartree)			
		Anions		Cations	
		Our results	NMF	Our results	NMF
4	Be	-14.27397675	-14.27739481	-14.46426381	
5	B	-24.23629313	-24.23757518	-24.40619392	-24.51922137
6	C	-37.11311415	-37.29222377	-37.69814209	-37.70884362
7	N	-53.66997517	-53.88800501	-54.18510745	-54.32195889
8	O	-74.37649652	-74.37260568	-74.59947209	-74.78974593
9	F	-98.57046183	-98.83172020	-99.44815874	-99.45945391
10	Ne	-127.5086495	-127.8178141	-128.5082245	
11	Na	-161.6757598	-161.6769626	-161.8279992	-161.8551260
12	Mg	-199.3604662	-199.3718097	-199.5233196	
13	Al	-241.6667369	-241.6746705	-241.7935389	-241.8782653
14	Si	-288.4665366	-288.5731311	-288.8785902	-288.8896602
15	P	-340.2240030	-340.3497759	-340.5932184	-340.6988736
16	S	-397.1731064	-397.1731828	-397.4126215	-397.5384302
17	Cl	-458.8984458	-459.0485907	-459.5680776	-459.5769253
18	Ar	-526.1035653	-526.2745343	-526.7900620	
19	K	-599.0165109	-599.0175794	-599.1456809	-599.1619170
20	Ca	-676.5641364	-676.5700126	-676.5624412	
21	Sc	-759.4579119	-759.5391440	-759.4345725	-759.6887738
22	Ti	-847.8030832	-848.2034008	-848.0509535	-848.3725498
23	V	-942.0556906	-942.6707837	-942.7517567	-942.8631322
24	Cr	-1042.406419	-1043.139393	-1043.321945	-1043.337097
25	Mn	-1149.648030	-1149.649383	-1149.529828	-1149.729110
26	Fe	-1262.126273	-1262.213012	-1261.971903	-1262.367074
27	Co	-1380.679266	-1381.128750	-1380.882061	-1381.351810
28	Ni	-1505.859484	-1506.591099	-1570.483454	-1506.821133
29	Cu	-1637.859898	-1638.728242	-1638.940294	-1638.964145
30	Zn	-1777.564036	-1777.567545	-1777.745649	
31	Ga	-1923.055879	-1923.059722	-1923.177982	-1923.260381
32	Ge	-2074.983722	-2075.086491	-2075.386442	-2075.394742
33	As	-2233.773329	-2233.888335	-2234.127078	-2234.222940
34	Se	-2399.558847	-2399.558574	-2399.792972	-2399.904726
35	Br	-2571.915145	-2572.045211	-2572.529689	-2572.536273
36	Kr	-2751.423206	-2751.567394	-2752.031895	
37	Rb	-2938.218921	-2938.219931	-2938.342206	-2938.354900
38	Sr	-3131.367043	-3131.373777	-3131.378175	
39	Y	-3331.467796	-3331.472882	-3331.461714	-3331.683116
40	Zr	-3538.538111	-3538.809305	-3538.809241	-3538.994500
41	Nb	-3752.920050	-3753.389513	-3753.555146	-3753.578216
42	Mo	-3975.055632	-3975.333703	-3975.520048	-3975.526268
43	Tc	-4204.595304	-4204.594360	-4204.430695	-4204.764631
44	Ru	-4441.133156	-4441.321956	-4441.164831	-4441.528477
45	Rh	-4685.219380	-4685.664172	-4685.709565	-4685.875582
46	Pd	-4937.161218	-4937.675930	-4937.888076	-4937.891544
47	Ag	-5197.479958	-5197.481334	-5197.682097	-5197.700050
48	Cd	-5464.876405	-5464.878609	-5465.045432	

Table 2.2: Computed ionic energies from H through Te compared with the calculations of [6]. Energy units are in Hartrees. NMF values are the numerical Hartree-Fock limit of the given configuration (Continued on next page)

Z	Symbol	Energies (Hartree)			
		Anions		Cations	
		Our results	NMF	Our results	NMF
49	In	-5739.975893	-5739.978392	-5740.101561	-5740.175141
50	Sn	-6022.588935	-6022.678323	-6022.966041	-6022.972657
51	Sb	-6313.067201	-6313.165941	-6313.398175	-6313.481518
52	Te	-6611.503815	-6611.503394	-6611.731825	-6611.827949

Table 2.2: Computed ionic energies from H through Te compared with the calculations of [6]. Energy units are in Hartrees. NMF values are the numerical Hartree-Fock limit of the given configuration.

## 2.8 Conclusion

In this chapter the electron energies of atoms  $Z = 1$  through  $Z = 103$  have been calculated with acceptable accuracy when compared to [5]. Likewise the energies of anions and cations from  $Z = 1$  through  $Z = 52$  has been calculated and compared with [6] and yielded acceptable results with respect to the used basis set. Any deviations in the electron energy of the atoms are likely to rise from the fact that the calculations adopts a spherical average approximation. However for the ions the deviation may also rise from the fact that the used basis has not been optimised to the given problems.





## Chapter 3

# Polarizability calculations

Now that the electronic energies of the elements have been calculated we wondered if it was possible to calculate the atomic polarizability of the elements using the Hartree-Fock theory. Similar calculations have been performed by [11], [12] and [14] on which we base this chapter.

### 3.1 Polarizability

When an external electric field is applied to electronic systems the charged particles will shift accordingly to the applied field. This is known as an induced dipole

$$\mathbf{p} = \alpha \mathcal{E}, \quad (3.1)$$

where  $\mathbf{p}$  is the induced dipole moment,  $\mathcal{E}$  is the applied electric field and  $\alpha$  is the dipole polarizability. The resulting energy change can be expanded in a Taylor series assuming a static electric field

$$E = E^{(0)} + \mathcal{E}E^{(1)} + \mathcal{E}^2E^{(2)} + \dots \quad (3.2)$$

The first order term is the so-called permanent dipole moment, however the second order term is related to the dipole polarizability  $\alpha$  in the sense that

$$\mathcal{E}^2E^{(2)} = -\frac{1}{2}\alpha\mathcal{E}^2 \quad (3.3)$$

By applying an external static field to the Fock-operator one should be able to obtain expressions for the energy corresponding to a certain external field strength and hence determine the polarizability,  $\alpha$ .

### 3.2 The Hamiltonian with an external electric field

An external electric field is applied in the  $\mathbf{e}_z$ -direction giving rise to the potential,  $z\mathcal{E}$ , where  $\mathcal{E}$  is the field strength. The perturbation to the Hamiltonian is then

$$\Delta\hat{H}_{\mathcal{E}} = \hat{z}\mathcal{E}. \quad (3.4)$$

This can be solved separately from the usual Hamiltonian matrix and involves integrals of the form

$$\langle \psi_\mu | \hat{z}\mathcal{E} | \psi_\lambda \rangle = \mathcal{E} \int_{-\infty}^{\infty} \int_0^{2\pi} \int_0^\pi \psi_\mu^* \psi_\lambda \cos(\theta) r^3 \sin(\theta) d\phi d\theta dr, \quad (3.5)$$

where  $\hat{z} = r \cos(\theta)$  has been used. Since  $\hat{z}$  is an odd operator this integral will vanish when the electrons have the same angular momentum.

The integrals has been evaluated in Wolfram Mathematica to

$$\langle \psi_{i,l=0} | \hat{z}\mathcal{E} | \psi_{j,l=1} \rangle = \frac{\mathcal{E}\sqrt{3\pi}}{8(b_i + b_j)^{5/2}} \quad (3.6)$$

$$\langle \psi_{i,l=1} | \hat{z}\mathcal{E} | \psi_{j,l=0} \rangle = \langle \psi_{i,l=0} | \hat{z}\mathcal{E} | \psi_{j,l=1} \rangle$$

$$\langle \psi_{i,l=1} | \hat{z}\mathcal{E} | \psi_{j,l=2} \rangle = \frac{\mathcal{E}\sqrt{15\pi}}{8(b_i + b_j)^{7/2}} \quad (3.7)$$

$$\langle \psi_{i,l=2} | \hat{z}\mathcal{E} | \psi_{j,l=1} \rangle = \langle \psi_{i,l=1} | \hat{z}\mathcal{E} | \psi_{j,l=2} \rangle$$

$$\langle \psi_{i,l=2} | \hat{z}\mathcal{E} | \psi_{j,l=3} \rangle = \frac{9\mathcal{E}\sqrt{35\pi}}{32(b_i + b_j)^{9/2}} \quad (3.8)$$

$$\langle \psi_{i,l=3} | \hat{z}\mathcal{E} | \psi_{j,l=2} \rangle = \langle \psi_{i,l=2} | \hat{z}\mathcal{E} | \psi_{j,l=3} \rangle.$$

The rest of the integrals vanishes. The Hamiltonian with an applied electron field is then the usual Hamiltonian of chapter 2 added with the polarization elements in equations 3.6 to 3.8.

### 3.3 Implementation

As mentioned in the previous section,  $\hat{z}\mathcal{E}$  is an odd function so the expectation value vanishes for same angular momenta. This will cause the effect of an external electric field seem to vanish for atoms with only  $s$ -orbitals occupied.

This is of course not the case so we needed to adapt an  $p$ -basis for the atoms with no angular momenta. This has been implemented in the sense that the orbital exponents belonging to the outermost angular momentum has been copied to the following angular momentum. Atoms with only  $s$ -orbitals gets both  $s$ - and  $p$ -orbitals, atoms with both  $s$ - and  $p$ -orbitals gets an additional  $d$ -basis etc. We find the polarizability by calculating the electron energy as a function of the field strength. According to equation 3.3 when the electron energies are plotted as a function of the field strength a parabola should appear. From this we fit the second-degree polynomial thereby obtaining the polarizability.

### 3.4 Results

The script used for the calculations is presented in appendix C.2

During the project we have tried to calculate the polarizabilities of the elements. However we have only managed to get a decent result when working with Hydrogen. We know that the error is caused by the fact that the Coulomb and exchange matrix elements were not correctly calculated. However when dealing with Hydrogen they effectively vanish, which is why we, when cancelling these

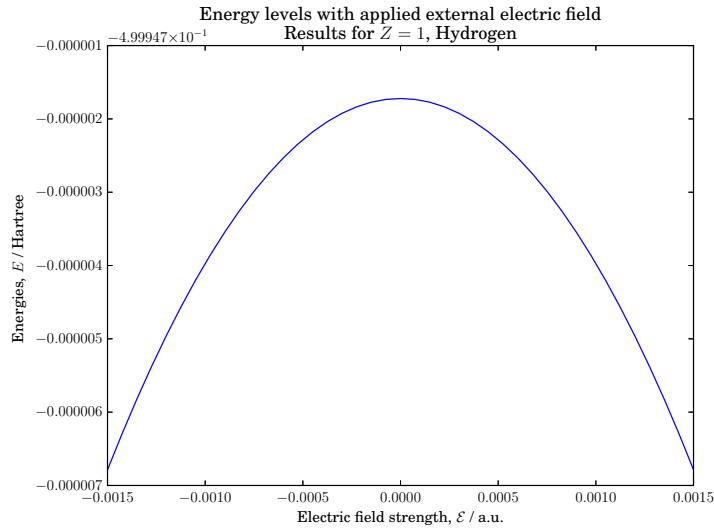


Figure 3.1: Plot of electron energies as a function of a static field strength for a hydrogen atom.

integrals, obtained reasonably results for the polarizability of Hydrogen. A plot of the electron energies as a function of the field strength of hydrogen can be seen in figure 3.1. From his plot we have obtained a polarizability for hydrogen of  $4.47a_0^3$ . The actual value is  $4.50a_0^3$  [10]. When trying to calculate the polarizability of the next element Helium we already see deviations of about 300%. We conclude therefore that our Coulomb and exchange matrix elements are wrong.

However since we are not able to calculate the polarizability of the other atoms, we instead try to improve the calculations on Hydrogen. We do this by increasing the basis size by adding two smaller exponents to the wave functions. These additions are respectively one half and quarter of the smallest exponent of the used basis. By doing this the calculated polarizability of Hydrogen becomes  $4.50a_0^3$ . We can thereby conclude that we have managed to calculate the polarizability of Hydrogen to an acceptable accuracy.



## Chapter 4

# Implementation for molecules

Now that the electron energy of the elements and some ions have been calculated, it is time to implement the Hartree-Fock theory to calculate electronic energies and bond length of different molecules. In this chapter the bond length of diatomic molecules consisting of atoms with electrons orbiting in  $s$ -orbitals will be calculated. The chapter will not only present the results but also try to discuss and explain the flaws and inaccuracy of the calculations. As in chapter 2 we need to calculate the matrix elements of the theory.

### 4.1 Calculation of molecular matrix elements

This section aims to present a derivation of a method used to analytically evaluate the given integrals present in the molecular Hartree-Fock calculations of the project. The analytical expressions will be given for un-normalised Cartesian Gaussian functions. This section is based on [8].

#### Strategy

The strategy of the analytical derivation is to rewrite the Cartesian Gaussian functions in such a way that the given integrals, one needs to compute is done over a simple Gaussian function, that of an  $s$ -type Gaussian orbital. In order to determine the higher Gaussian orbitals appropriate operators are then used on these integrals. The method of this chapter uses the Gaussian Product Theorem described in section 2.1. Furthermore the Gaussian functions are rewritten with the use of Hermite polynomials which are described in appendix B.

The Hermite polynomials are given by

$$H_n(x) = n! \sum_{m=0}^{\lfloor \frac{1}{2}n \rfloor} \frac{(-1)^m (2x)^{n-2m}}{m!(n-2m)!}. \quad (4.1)$$

The expansion of a given monomial in term of a Hermite polynomial is given

by

$$x^n = \frac{n!}{2^n} \sum_{m=0}^{\lfloor \frac{1}{2}n \rfloor} \frac{H_{n-2m}(x)}{m!(n-2m)!}. \quad (4.2)$$

Furthermore the Rodrigues' formula regarding Hermite polynomials reads

$$H_n(x) = \frac{(-1)^n}{\exp(-x^2)} \frac{d^n}{dx^n} e^{-x^2}. \quad (4.3)$$

Using equations 4.1 to 4.3 a given Gaussian orbital may be written as a product of a appropriate operator and a Gaussian function exemplified here with the  $x$ -component of an arbitrary Gaussian Type Orbital

$$(x - Ax)^l e^{-b(x-Ax)^2} = \frac{l!}{2^l} \sum_{i=0}^{\lfloor \frac{1}{2}l \rfloor} \frac{1}{i!(l-2i)!b^{l-i}} \frac{\partial^{l-2i}}{\partial Ax^{l-2i}} e^{-b(x-Ax)^2} \quad (4.4)$$

Using these equations as foundation it is possible to derive the analytical expression of the molecular matrix elements.

### Analytical derivations of the elements

In this section the matrix elements will be derived using the  $x$ -component of an arbitrary Gaussian orbital as an example in the overlap, kinetic, nuclear attraction and electron repulsion integral. During the derivation it is assumed that the electron states may be described by the following four un-normalized Gaussian orbitals

$$\phi_A(\mathbf{r}) = (x - Ax)^{l_1} (y - Ay)^{n_1} (z - Az)^{m_1} e^{-b_1(\mathbf{r}-\mathbf{A})^2} \quad (4.5)$$

$$\phi_B(\mathbf{r}) = (x - Bx)^{l_2} (y - By)^{n_2} (z - Bz)^{m_2} e^{-b_2(\mathbf{r}-\mathbf{B})^2} \quad (4.6)$$

$$\phi_C(\mathbf{r}) = (x - Cx)^{l_3} (y - Cy)^{n_3} (z - Cz)^{m_3} e^{-b_3(\mathbf{r}-\mathbf{C})^2} \quad (4.7)$$

$$\phi_D(\mathbf{r}) = (x - Dx)^{l_4} (y - Dy)^{n_4} (z - Dz)^{m_4} e^{-b_4(\mathbf{r}-\mathbf{D})^2} \quad (4.8)$$

### The overlap integral

The overlap integral is defined as

$$\langle A|B \rangle = \int_{-\infty}^{\infty} \phi_A(\mathbf{r}) \phi_B(\mathbf{r}) d\mathbf{r}. \quad (4.9)$$

Using the Gaussian product theorem and equation 4.4 the  $x$ -component of an arbitrary overlap may be written

$$\begin{aligned} \langle A|B \rangle_x &= \frac{l_1!l_2!}{2^{l_1+l_2}} \sum_{i_1=0}^{\lfloor \frac{1}{2}l_1 \rfloor} \sum_{i_2=0}^{\lfloor \frac{1}{2}l_2 \rfloor} (i_1!i_2!(l_1-2i_1)!(l_2-2i_2)!b_1^{l_1-i_1}b_2^{l_2-i_2})^{-1} \\ &\quad \times \frac{\partial^{l_1-2i_1}}{\partial Ax^{l_1-2i_1}} \frac{\partial^{l_2-2i_2}}{\partial Bx^{l_2-2i_2}} e^{-\eta_p(Ax-Bx)^2}. \end{aligned} \quad (4.10)$$

Using equations 4.1 and 4.3 the derivatives with the respect to atom positions are found to be

$$\begin{aligned} \frac{\partial^{k_1}}{\partial Ax^{k_1}} \frac{\partial^{k_2}}{\partial Bx^{k_2}} e^{-\eta_p(Ax-Bx)^2} &= (-1)^k \eta_p^{\frac{1}{2}k} H_k(\sqrt{\eta_p}(Ax-Bx)) e^{-\eta_p(Ax-Bx)^2} \\ &= (-1)^k k! \sum_{i=0}^{\frac{1}{2}k} \frac{(-1)^i 2^{k-2i} \eta_p^{k-i} (Ax-Bx)^{k-2i}}{i!(k-2i)!} \\ &\quad \times e^{-\eta_p(Ax-Bx)^2}, \end{aligned} \quad (4.11)$$

where  $k = k_1 + k_2$ . If the same is done for the  $y$  and  $z$  component any overlap matrix element may be written as

$$\langle A|B \rangle = \left( \frac{\pi}{\gamma_p} \right)^{3/2} e^{-\eta_p(A-B)^2} \sum_{i_1, i_2, o} S_x \sum_{j_1, j_2, p} S_y \sum_{k_1, k_2, q} S_z, \quad (4.12)$$

where

$$\begin{aligned} \sum_{i_1, i_2, o} S_x &= \frac{(-1)^{l_1} l_1! l_2!}{\gamma_p^{l_1+l_2}} \sum_{i_1=0}^{\lfloor \frac{1}{2}l_1 \rfloor} \sum_{i_2=0}^{\lfloor \frac{1}{2}l_2 \rfloor} \sum_{o=0}^{\lfloor \frac{1}{2}\Omega \rfloor} \frac{(-1)^o \Omega! b_1^{l_2-i_1-2i_2-o} b_2^{l_1-2i_1-i_2-o}}{4^{i_1+i_2+o} i_1! i_2! o!} \\ &\quad \times \frac{\gamma_p^{2(i_1+i_2)+o} (Ax-Bx)^{\Omega-2o}}{(l_1-2i_1)!(l_2-2i_2)!(\Omega-2o)!}, \end{aligned} \quad (4.13)$$

where  $\Omega = l_1 + l_2 - 2(i_1 + i_2)$ .  $S_y$  and  $S_z$  are defined in a similar manner.

### Kinetic integral

The kinetic integral is a one-electron term and is defined as

$$\left\langle A \left| -\frac{1}{2} \nabla^2 \right| B \right\rangle = -\frac{1}{2} \int_{-\infty}^{\infty} \phi_A \nabla^2 \phi_B d\mathbf{r}. \quad (4.14)$$

Applying the Laplacian to the second Gaussian orbital any x-component may be written

$$\begin{aligned} -\frac{1}{2} \nabla^2 (x-Bx)^{l_2} e^{-b_2(x-Bx)^2} &= e^{-b_2(x-Bx)^2} \\ &\quad \times \left( b_2(2l_2+1)(x-Bx)^{l_2} - 2b_2^2(x-Bx)^{l_2+2} \right. \\ &\quad \left. - \frac{1}{2} l_2(l_2-1)(x-Bx)^{l_2-2} \right) \end{aligned} \quad (4.15)$$

In this way the kinetic integral is written as a sum of individual overlap integrals each with different angular momenta. Using this fact the kinetic integral of any Gaussian orbitals is found to be

$$\begin{aligned} \left\langle A \left| -\frac{1}{2} \nabla^2 \right| B \right\rangle &= \frac{1}{2} \left( b_2(4(l_2+m_2+n_2)+6) \langle A|B \rangle \right. \\ &\quad - 4b_2^2 (\langle A|B, l_2+2 \rangle + \langle A|B, m_2+2 \rangle + \langle A|B, n_2+2 \rangle) \\ &\quad - l_2(l_2-1) \langle A|B, l_2-2 \rangle - m_2(m_2-1) \langle A|B, m_2-2 \rangle \\ &\quad \left. - m_2(m_2-1) \langle A|B, m_2-2 \rangle \right), \end{aligned} \quad (4.16)$$

where the notation

$$|B, l_2 + 2\rangle = (x - Bx)^{l_2+2} (y - By)^{n_2} (z - Bz)^{m_2} e^{-b_1(\mathbf{r}-\mathbf{B})^2} \quad (4.17)$$

has been used.

### Nuclear attraction integral

Using the notation  $r_c = \|\mathbf{r} - \mathbf{r}_c\|$  the nuclear attraction integral can be defined as

$$\left\langle A \left| -\frac{Z_c}{r_c} \right| B \right\rangle = -Z_c \int_{-\infty}^{\infty} \phi_A \frac{1}{r_c} \phi_B d\mathbf{r} \quad (4.18)$$

By using the known integral result

$$\frac{1}{2} \sqrt{\frac{\pi}{a}} = \int_0^{\infty} e^{-ax^2} dx, \quad (4.19)$$

the denominator of equation 4.18 may be written as

$$\frac{1}{\|\mathbf{r} - \mathbf{r}_c\|} = \frac{2}{\sqrt{\pi}} \int_0^{\infty} e^{-u^2(\mathbf{r}-\mathbf{r}_c)^2} du. \quad (4.20)$$

equation 4.18 may then be written as

$$\begin{aligned} \left\langle A \left| -\frac{Z_c}{r_c} \right| B \right\rangle &= -\frac{2Z_c}{\sqrt{\pi}} \int_{-\infty}^{\infty} \int_0^{\infty} e^{-b_1(\mathbf{r}-\mathbf{A})^2 - b_2(\mathbf{r}-\mathbf{B})^2 - u^2(\mathbf{r}-\mathbf{r}_c)^2} du d\mathbf{r} \\ &= \frac{2Z_c}{\sqrt{\pi}} e^{-\eta_p(\mathbf{A}-\mathbf{B})^2} \int_0^{\infty} e^{-\gamma_p \mathbf{P}^2 - u^2 \mathbf{r}_c^2} du \\ &\quad \times \int_{-\infty}^{\infty} e^{-(\gamma_p + u^2)\mathbf{r}^2 + 2(\gamma_p \mathbf{P} + u^2 \mathbf{r}_c)\mathbf{r}} d\mathbf{r} \end{aligned} \quad (4.21)$$

where the Gaussian product theorem has been used. The last integral in equation 4.21 can be solved by using the relation

$$\int_0^{\infty} e^{-(au^2 + 2bu + c)} du = \sqrt{\frac{\pi}{4a}} \left( 1 - \Phi \left( \frac{b}{\sqrt{a}} \right) \right) e^{(b^2 - ac)/a},$$

which implies

$$\int_{-\infty}^{\infty} e^{-(au^2 + 2bu)} du = \sqrt{\frac{\pi}{a}} e^{(b^2)/a}. \quad (4.22)$$

Using this relation 4.21 becomes

$$\left\langle A \left| -\frac{Z_c}{r_c} \right| B \right\rangle = -2\pi Z_c e^{-\eta_p(\mathbf{A}-\mathbf{B})^2} \int_0^{\infty} (\gamma_p + u^2)^{-3/2} e^{-\gamma_p u^2 (\mathbf{P}-\mathbf{r}_c)^2 / (\gamma_p + u^2)} du \quad (4.23)$$

By a change of variables to

$$t^2 = u^2 (\gamma_p + u^2) \quad \Rightarrow \quad du = \frac{(\gamma_p + u^2)^{3/2}}{\gamma_p} dt, \quad (4.24)$$



using these the new limits of the integral are obvious and equation 4.21 becomes

$$\left\langle A \left| -\frac{Z_c}{r_c} \right| B \right\rangle = -\frac{2\pi Z_c}{\gamma_p} e^{-\eta_p(\mathbf{A}-\mathbf{B})^2} \int_0^1 e^{-\gamma_p(\mathbf{P}-\mathbf{r}_c)^2 t^2} dt. \quad (4.25)$$

From here we identify that the expression is similar to equation 4.10 but with two functions to differentiate. The differentiation of  $\exp(-\eta_p(\mathbf{A}-\mathbf{B})^2)$  is found earlier, and by using Leibniz's theorem for differentiating a product the  $x$ -component of the differentiation of the integral becomes

$$\begin{aligned} \frac{\partial^{j_1}}{\partial A_x^{j_1}} \frac{\partial^{j_2}}{\partial B_x^{j_2}} \int_0^1 e^{-\gamma_p(\mathbf{P}-\mathbf{r}_c)^2 t^2} dt &= b_1^{j_1} b_2^{j_2} j! \sum_{u=0}^{\lfloor \frac{1}{2}j \rfloor} \frac{(-1)^{j+u} 2^{j-2u} (P_x - r_{cx})^{j-2u}}{u!(j-2u)! \gamma_p^u} \\ &\times \int_0^1 t^{2(j-u)} e^{-\gamma_p(\mathbf{P}-\mathbf{r}_c)^2 t^2} dt, \end{aligned} \quad (4.26)$$

where  $j = j_1 + j_2$ . The last integral also appears for the  $y$ - and  $z$ -components, and can be evaluated by

$$F_\nu(u) = \int_0^1 t^{2\nu} e^{-ut^2} dt = \frac{(2\nu)!}{2\nu!} \left( \frac{\sqrt{\pi}}{4^\nu u^{\nu+1/2}} \Phi(\sqrt{u}) - \sum_{k=0}^{\nu-1} \frac{(\nu-k)!}{4^k (2\nu-2k)! u^{k+1}} \right). \quad (4.27)$$

Using this equation 4.26 may be written

$$\begin{aligned} \frac{\partial^{j_1}}{\partial A_x^{j_1}} \frac{\partial^{j_2}}{\partial B_x^{j_2}} \int_0^1 e^{-\gamma_p(\mathbf{P}-\mathbf{r}_c)^2 t^2} dt &= b_1^{j_1} b_2^{j_2} j! \sum_{u=0}^{\lfloor \frac{1}{2}j \rfloor} \frac{(-1)^{j+u} 2^{j-2u} (P_x - r_{cx})^{j-2u}}{u!(j-2u)! \gamma_p^u} \\ &\times F_{j-u}(\gamma_p(\mathbf{P}-\mathbf{r}_c)^2) \end{aligned} \quad (4.28)$$

From this, one can find that the complete expression for the nuclear attraction integral is given by

$$\begin{aligned} \left\langle A \left| -\frac{Z_c}{r_c} \right| B \right\rangle &= -\frac{2\pi Z_c}{\gamma_p} e^{-\eta_p(\mathbf{A}-\mathbf{B})^2} \sum_{\substack{i_1, i_2, \\ o_1, o_2, \\ r, u}} A_x \sum_{\substack{j_1, j_2, \\ p_1, p_2, \\ s, v}} A_y \\ &\times \sum_{\substack{k_1, k_2, \\ q_1, q_2, \\ t, w}} A_z 2F_\nu(\gamma_p(\mathbf{P}-\mathbf{r}_c)^2), \end{aligned} \quad (4.29)$$

where

$$\gamma_p = b_1 + b_2, \quad \eta_p = \frac{b_1 b_2}{\gamma_p}, \quad \mathbf{P} = \frac{1}{\gamma_p} (b_1 \mathbf{A} + b_2 \mathbf{B}) \quad (4.30)$$

and

$$\begin{aligned} \sum_{\substack{i_1, i_2, \\ o_1, o_2, \\ r, u}} A_x &= (-1)^{l_1+l_2} l_1! l_2! \sum_{i_1} \sum_{i_2} \sum_{o_1} \sum_{o_2} \sum_r \frac{(-1)^{o_2+r} (o_1+o_2)!}{4^{i_1+i_2+r} i_1! i_2! o_1! o_2! r!} \\ &\times \frac{b_1^{o_2-i_1-r} b_2^{o_1-i_2-r} (A_x - B_x)^{o_1+o_2-2r}}{(l_1-2i_1-o_1)! (l_2-2i_2-o_2)! (o_1+o_2-2r)!} \\ &\times \sum_u \frac{(-1)^u \mu_x! (P_x - r_{cx})^{\mu_x-2u}}{4^u u! (\mu_x-2u)! \gamma_p^{o_1+o_2-r+u}}, \end{aligned} \quad (4.31)$$

where  $\mu_x = l_1 + l_2 - 2(i_1 + i_2) - o_1 - o_2$  and  $\nu = \mu_x + \mu_y + \mu_z - u - v - w$ .  $A_y$  and  $A_z$  are defined in a similar way for the  $y$ - and  $z$ -components. The limits of the sums are

$$i_1 = 0 \rightarrow \left\lfloor \frac{1}{2}l_1 \right\rfloor, \quad i_2 = 0 \rightarrow \left\lfloor \frac{1}{2}l_2 \right\rfloor, \quad o_1 = 0 \rightarrow l_1 - 2i_1,$$

$$o_2 = 0 \rightarrow l_1 - 2i_2, \quad r = 0 \rightarrow \left\lfloor \frac{1}{2}(o_1 + o_2) \right\rfloor, \quad u = 0 \rightarrow \left\lfloor \frac{1}{2}\mu_x \right\rfloor.$$

### Coulomb and Exchange integrals

Both the Coulomb and exchange integrals are so-called two-electron integrals. In order to calculate them we present them as an electron repulsion integral defined as

$$\left\langle A, C \left| \frac{1}{r_{12}} \right| B, D \right\rangle = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\phi_A(\mathbf{r}_1)\phi_C(\mathbf{r}_1)\phi_B(\mathbf{r}_2)\phi_D(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2 \quad (4.32)$$

By appropriate modification of basis coefficients one can create both the Coulomb and the exchange integral from equation 4.32. To solve the integral the denominator is found in a similar way as in equation 4.20 and using the Gaussian product theorem and some rewriting to use the function in equation 4.22 the integral can be written

$$\begin{aligned} \left\langle A, C \left| \frac{1}{r_{12}} \right| B, D \right\rangle &= \frac{2}{\sqrt{\pi}} e^{-\eta_p(\mathbf{A}-\mathbf{B})^2} e^{-\eta_q(\mathbf{C}-\mathbf{D})^2} \\ &\quad \times \int_0^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-u^2\mathbf{r}_2^2 - \gamma_p\mathbf{P} - \gamma_q(\mathbf{r}_2 - \mathbf{Q})^2} \\ &\quad \times e^{-((\gamma_p+u^2)\mathbf{r}_1^2 - 2(\gamma_p\mathbf{P} + u^2\mathbf{r}_2) \cdot \mathbf{r}_1)} d\mathbf{r}_1 d\mathbf{r}_2 du, \end{aligned} \quad (4.33)$$

where

$$\gamma_p = b_1 + b_2, \quad \gamma_q = b_3 + b_4, \quad \eta_p = \frac{b_1 b_2}{\gamma_p}, \quad \eta_q = \frac{b_3 b_4}{\gamma_q},$$

$$\mathbf{P} = \frac{1}{\gamma_p} (b_1\mathbf{A} + b_2\mathbf{B}), \quad \mathbf{Q} = \frac{1}{\gamma_q} (b_3\mathbf{C} + b_4\mathbf{D}).$$

To use equation 4.22 we identify  $a = \gamma_p + u^2$  and  $b = \gamma_p\mathbf{P} + u^2\mathbf{r}_2$  and after integration over  $\mathbf{r}_1$  the integral becomes

$$\begin{aligned} \left\langle A, C \left| \frac{1}{r_{12}} \right| B, D \right\rangle &= 2\pi e^{-\eta_p(\mathbf{A}-\mathbf{B})^2} e^{-\eta_q(\mathbf{C}-\mathbf{D})^2} \\ &\quad \times \int_{-\infty}^{\infty} \int_0^{\infty} (\gamma_p + u^2)^{-3/2} e^{(\gamma_p\mathbf{P} + u^2\mathbf{r}_2)^2 / (\gamma_p + u^2)} \\ &\quad \times e^{-u^2\mathbf{r}_2^2 - \gamma_q(\mathbf{r}_2 + \mathbf{Q})^2} du d\mathbf{r}_2. \end{aligned} \quad (4.34)$$

Again to use equation 4.22 we rearrange such that  $a = (\gamma_p\gamma_q + (\gamma_p + \gamma_q)u^2) / (\gamma_p + u^2)$  and  $b = (\gamma_p u^2\mathbf{P} + (\gamma_p + u^2)\gamma_q\mathbf{Q}) / (\gamma_p + u^2)$  and after integration over  $\mathbf{r}_2$  we

obtain

$$\begin{aligned} \left\langle A, C \left| \frac{1}{r_{12}} \right| B, D \right\rangle &= \frac{2\pi^{5/2}}{(\gamma_p + \gamma_q)^{3/2}} e^{-\eta_p(\mathbf{A}-\mathbf{B})^2} e^{-\eta_q(\mathbf{C}-\mathbf{D})^2} \\ &\times \int_0^\infty (\eta + u^2)^{-3/2} e^{-\eta u^2(\mathbf{P}-\mathbf{Q})^2/(\eta+u^2)} du, \end{aligned} \quad (4.35)$$

with  $\eta = \frac{\gamma_p \gamma_q}{\gamma_p + \gamma_q}$ . By changing variables to  $t^2 = \frac{u^2}{\eta+u^2}$  we can write

$$\left\langle A, C \left| \frac{1}{r_{12}} \right| B, D \right\rangle = \frac{2\pi^{5/2}}{\gamma_p \gamma_q (\gamma_p + \gamma_q)^{1/2}} e^{-\eta_p(\mathbf{A}-\mathbf{B})^2} e^{-\eta_q(\mathbf{C}-\mathbf{D})^2} \int_0^1 e^{-\eta t^2(\mathbf{P}-\mathbf{Q})^2} dt. \quad (4.36)$$

In a similar fashion as with the nuclear attraction integral the x-component of the differentiation of the last integral becomes

$$\begin{aligned} &\frac{\partial^{j_1}}{\partial A_x^{j_1}} \frac{\partial^{j_2}}{\partial B_x^{j_2}} \frac{\partial^{j_3}}{\partial C_x^{j_3}} \frac{\partial^{j_4}}{\partial D_x^{j_4}} \int_0^1 e^{-\eta t^2(\mathbf{P}-\mathbf{Q})^2} dt \\ &= \frac{b_1^{j_1} b_2^{j_2} b_3^{j_3} b_4^{j_4}}{\gamma_p^{j_1+j_2} \gamma_q^{j_3+j_4}} j! \sum_{u=0}^{\lfloor \frac{1}{2}j \rfloor} \frac{(-1)^{u+j_1+j_2} (P_x - Q_x)^{j-2u} \eta^{j-u}}{u!(j-2u)!} \\ &\times F_{j-u}(\eta(\mathbf{P}-\mathbf{Q})^2), \end{aligned} \quad (4.37)$$

with  $j = j_1 + j_2 + j_3 + j_4$  and similar for the  $y$ - and  $z$ -component. Since the differentiation of  $\exp(-\eta_q(\mathbf{C}-\mathbf{D})^2)$  is similar to the differentiation of  $\exp(-\eta_p(\mathbf{A}-\mathbf{B})^2)$  the complete solution to the electron repulsive integral becomes

$$\begin{aligned} \left\langle A, C \left| \frac{1}{r_{12}} \right| B, D \right\rangle &= \frac{\pi^{5/2}}{\gamma_p \gamma_q \sqrt{\gamma_p + \gamma_q}} e^{-\eta_p(\mathbf{A}-\mathbf{B})^2} e^{-\eta_q(\mathbf{C}-\mathbf{D})^2} \\ &\times \sum_{\substack{i_1, i_2, i_3, i_4, \\ o_1, o_2, o_3, o_4, \\ r_1, r_2, u}} J_x \sum_{\substack{j_1, j_2, j_3, j_4, \\ p_1, p_2, p_3, p_4, \\ s_1, s_2, v}} J_y \sum_{\substack{k_1, k_2, k_3, k_4, \\ q_1, q_2, q_3, q_4, \\ t_1, t_2, w}} J_z \\ &\times 2F_\nu(\eta(\mathbf{P}-\mathbf{Q})^2), \end{aligned} \quad (4.38)$$

where

$$\gamma_p = b_1 + b_2, \quad \gamma_q = b_3 + b_4, \quad \eta_p = \frac{b_1 b_2}{\gamma_p}, \quad \eta_q = \frac{b_3 b_4}{\gamma_q},$$

$$\mathbf{P} = \frac{1}{\gamma_p} (b_1 \mathbf{A} + b_2 \mathbf{B}), \quad \mathbf{Q} = \frac{1}{\gamma_q} (b_3 \mathbf{C} + b_4 \mathbf{D}), \quad \eta = \frac{\gamma_p \gamma_q}{\gamma_p + \gamma_q},$$

and

$$\begin{aligned}
\sum_{\substack{i_1, i_2, i_3, i_4, \\ o_1, o_2, o_3, o_4, \\ r_1, r_2, u}} J_x &= \frac{(-1)^{l_1+l_2} l_1! l_2!}{\gamma_p^{l_1+l_2}} \sum_{i_1} \sum_{i_2} \sum_{o_1} \sum_{o_2} \sum_{r_1} \frac{(-1)^{o_2+r_1} (o_1 + o_2)!}{4^{i_1+i_2+r_1} i_1! i_2! o_1! o_2! r_1!} \\
&\times \frac{b_1^{o_2-i_1-r_1} b_2^{o_1-i_2-r_1} \gamma_p^{2(i_1+i_2)+r_1} (A_x - B_x)^{o_1+o_2-2r_1}}{(l_1 - 2i_1 - o_1)! (l_2 - 2i_2 - o_2)! (o_1 + o_2 - 2r_1)!} \\
&\times \frac{l_3! l_4!}{\gamma_q^{l_3+l_4}} \sum_{i_3} \sum_{i_4} \sum_{o_3} \sum_{o_4} \sum_{r_2} \frac{(-1)^{o_3+r_2} (o_3 + o_4)!}{4^{i_3+i_4+r_2} i_3! i_4! o_3! o_4! r_2!} \\
&\times \frac{b_3^{o_4-i_3-r_2} b_4^{o_3-i_4-r_2} \gamma_q^{2(i_3+i_4)+r_2} (C_x - D_x)^{o_3+o_4-2r_2}}{(l_3 - 2i_3 - o_3)! (l_4 - 2i_4 - o_4)! (o_3 + o_4 - 2r_2)!} \\
&\times \sum_u \frac{(-1)^u \mu_x! \eta^{\mu_x-u} (P_x - Q_x)^{\mu_x-2u}}{4^u u! (\mu_x - 2u)!}, \tag{4.39}
\end{aligned}$$

where

$$\mu_x = l_1 + l_2 + l_3 + l_4 - 2(i_1 + i_2 + i_3 + i_4) - o_1 - o_2 - o_3 - o_4$$

$$\nu = \mu_x + \mu_y + \mu_z - u - v - w.$$

The limits of the sums are given by

$$i_1 = 0 \rightarrow \left\lfloor \frac{1}{2} l_1 \right\rfloor, \quad i_2 = 0 \rightarrow \left\lfloor \frac{1}{2} l_2 \right\rfloor, \quad i_3 = 0 \rightarrow \left\lfloor \frac{1}{2} l_3 \right\rfloor,$$

$$i_4 = 0 \rightarrow \left\lfloor \frac{1}{2} l_4 \right\rfloor, \quad o_1 = 0 \rightarrow l_1 - 2i_1, \quad o_2 = 0 \rightarrow l_2 - 2i_2, \quad o_3 = 0 \rightarrow l_3 - 2i_3,$$

$$o_4 = 0 \rightarrow l_4 - 2i_4, \quad r_1 = 0 \rightarrow \left\lfloor \frac{1}{2} (o_1 + o_2) \right\rfloor,$$

$$r_2 = 0 \rightarrow \left\lfloor \frac{1}{2} (o_3 + o_4) \right\rfloor, \quad u = 0 \rightarrow \left\lfloor \frac{1}{2} \mu_x \right\rfloor$$

The  $y$ - and  $z$ -components are calculated in a similar manner.

## 4.2 Procedure

Using the matrix elements and the Born-Oppenheimer approximation we try to calculate the bond length of a few molecules. These molecules are: H<sub>2</sub>, HeH, LiH, He<sub>2</sub> and Li<sub>2</sub>. In the calculations we have used much of the same procedure as we did in chapter 2. However since the electron configuration is different for the molecules we have changed the interactions such that the configuration of the molecules behaves accordingly to the Aufbau Principle. So in principle we just place two atoms at a distance away from each other and then use the matrix elements of the appropriate molecule. To further simplify the calculations we always place one of the atoms in origin and the other along the  $z$ -axis such that the distance between the atoms only depend on one parameter. The positions of the molecules can be seen in figure 4.1

The results are shown as plots of the electronic energy and the potential energy function as a function of the internuclear distance in the next section

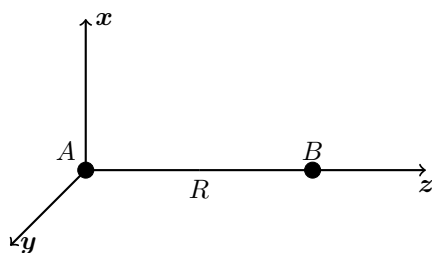


Figure 4.1: A diatomic molecule with nuclei centered at  $A$  and  $B$ . The internuclear axis is chosen as the  $z$ -axis.

### 4.3 Results

Here we show the calculations of the electron energy and potential energy functions as a function of the internuclear distance of the molecules  $H_2$ ,  $HeH$ ,  $LiH$ ,  $He_2$  and  $Li_2$ .

The scripts used to perform the calculations can be found in appendix C.3

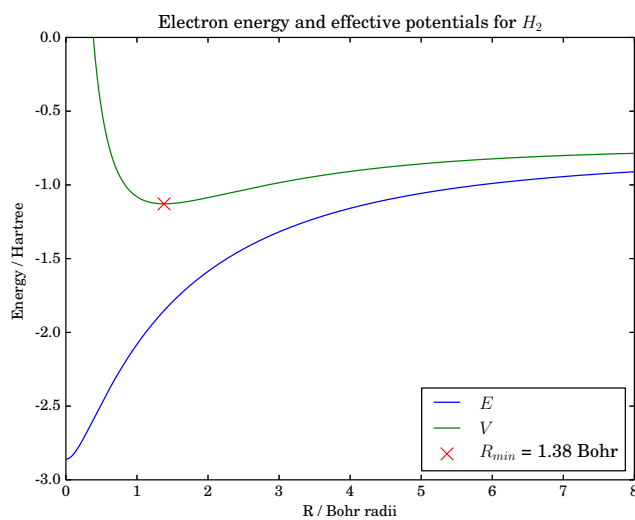


Figure 4.2: Electron energy and potential energy function as a function of internuclear distance for  $H_2$ . The bond length is marked with a red cross.

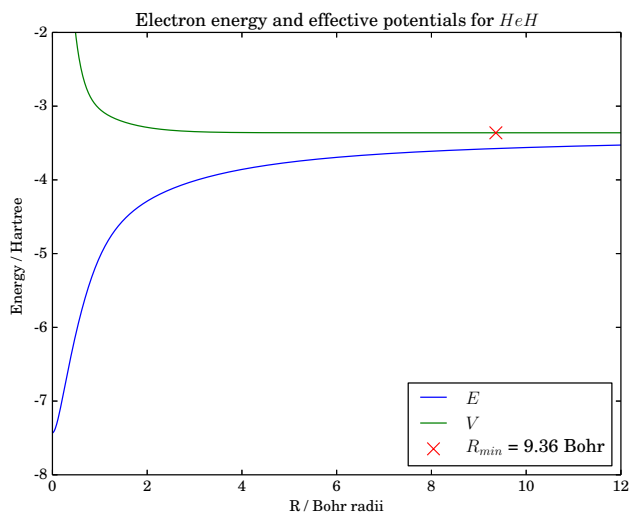


Figure 4.3: Electron energy and potential energy function as a function of internuclear distance for  $HeH$ . The bond length is marked with a red cross.

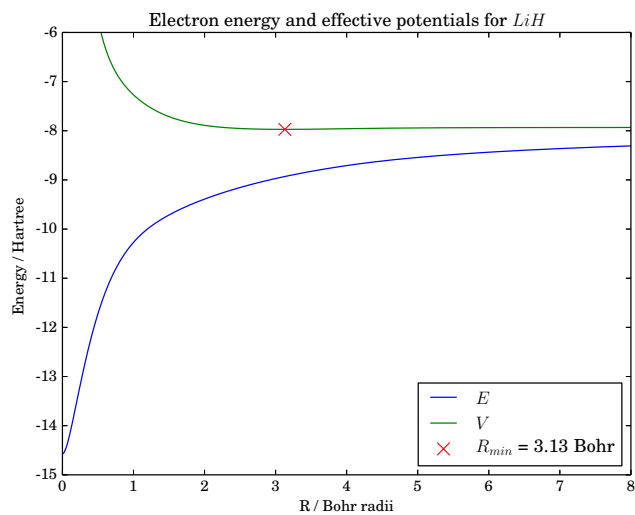


Figure 4.4: Electron energy and potential energy function as a function of internuclear distance for  $LiH$ . The bond length is marked with a red cross.

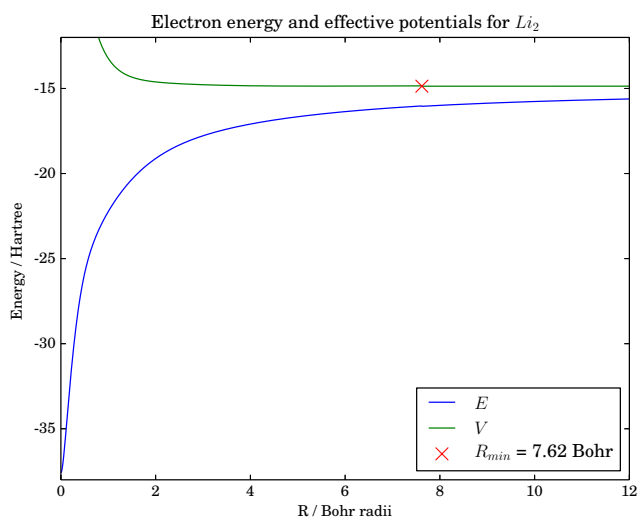


Figure 4.5: Electron energy and potential energy function as a function of internuclear distance for  $Li_2$ . The bond length is marked with a red cross.

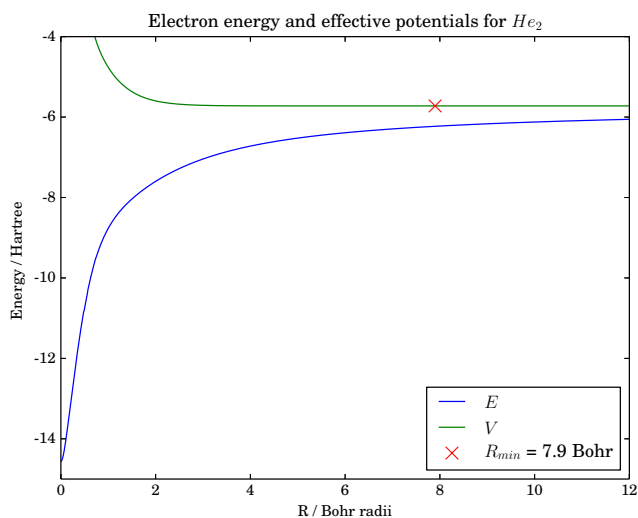


Figure 4.6: Electron energy and potential energy function as a function of internuclear distance for  $He_2$ . The bond length is marked with a red cross.

## 4.4 Discussion

This section aims to discuss the results of the calculation of the electron energy and the potential energy functions of the different molecules. At first we study the electron energies of the molecule. In all the molecules the total electron energy graphs have shape as expected. However as the internuclear distance

tends towards infinity we notice that the energy gets above the sum of the electron energy of the atoms of which the molecule is composed. Hereby goes the electron energy of  $H_2$  above two times the energy of the hydrogen atom which is  $-0.4999$  Ha, and similar for the other molecules. This result is a known mathematical flaw of the Hartee-Fock method as the used basis only uses coupled molecular orbitals. For large  $R$  the constituent atoms should tend to dissociate into separated atoms but our basis does not allow that. In the other limit as the distance tends towards zero the results obey our expectations. In this limit the electron energy tends towards the electron energy of the atom which has the same number of electrons as the molecule. This is of course expected since at  $R = 0$  the problem basically becomes a central field problem as the one solved for the atoms.

When studying the potential energy function of the molecules also these look as expected. The limit value of the potential energy function as  $R \rightarrow \infty$  should be the same as the electron energy since the nuclear repulsion term tends to zero as the internuclear distance tends towards infinity. The nuclear repulsion term is also responsible for the limit of the potential energy functions as the internuclear distance tends to zero. In this limit the potential energy tends to infinity because of the repulsion of the positive nucleus. The most interesting thing to gather from the potential energy functions is the bond length of the molecules, which per definition is the distance at which the potential energy function reaches a minimum. However in the figures of section 4.3 it is difficult to see the exact minimum, so here we bring figures where the minimum of the potential energy functions of the molecules  $He_2$ ,  $LiH$  and  $Li_2$  are highlighted.

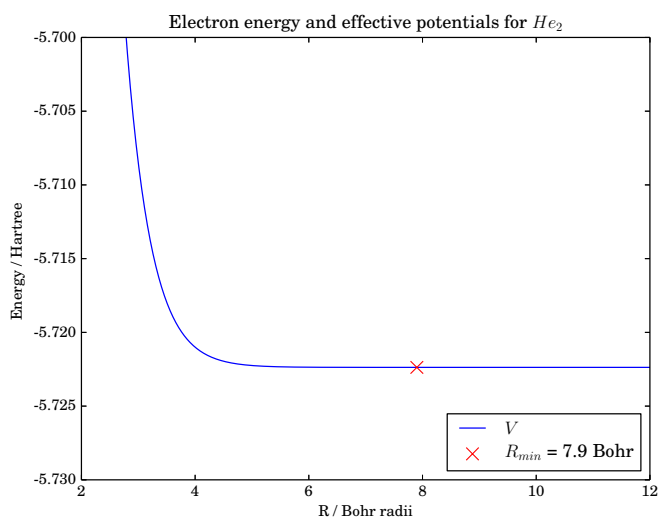


Figure 4.7: Plot showing the minimum potential energy function and bond length of  $He_2$



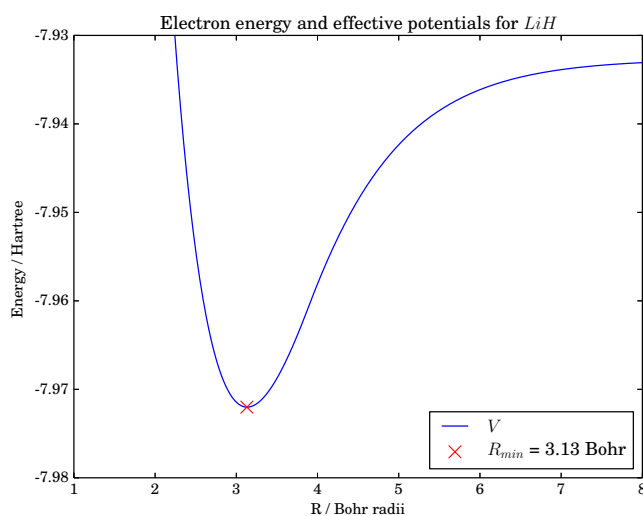


Figure 4.8: Plot showing the minimum potential energy function and bond length of  $LiH$

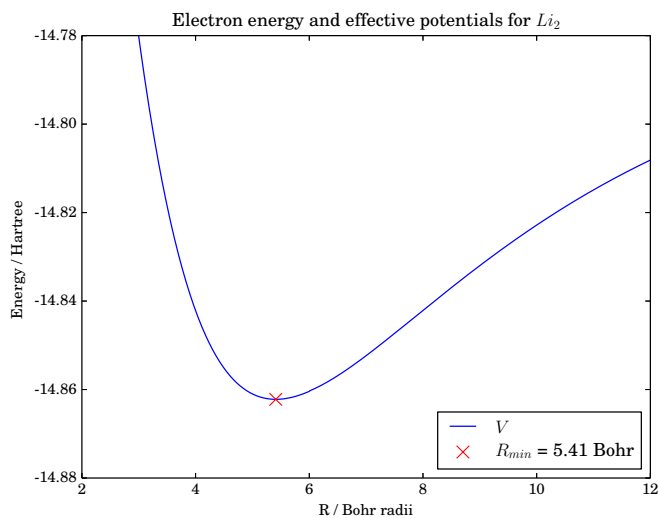


Figure 4.9: Plot showing the minimum potential energy function and bond length of  $Li_2$

The figures clearly show the potential well causing a specific bond length for the molecules  $LiH$  and  $Li_2$ . We notice however that in the  $HeH$  molecule the potential energy function does not form a specific potential well, thereby causing a very weak bonding between the Hydrogen and the Helium atoms. The same is seen when the potential energy function of  $He_2$  is studied. This finding agrees with the fact that Helium is an inert gas. The bond lengths of the molecules

can be found summarized in the following table.

Molecule	Our $R_e$	Reference $R_e$ [4]
H <sub>2</sub>	1.38	1.4011
HeH	9.36	-
LiH	3.13	3.015
He <sub>2</sub>	7.9	5.61
Li <sub>2</sub>	5.41	5.051

Table 4.1: Calculated and actual bond length of the molecules from [4]. All units are in Bohr radii.

We notice that the bond lengths in general are calculated fairly well with the exceptions of the molecules containing the Helium atom.

In order to compare the results of the calculations on another parameter than the bond length, the next section aims to calculate the vibrational states of the molecules in order to compare the calculated dissociation energy with experimental values.

## 4.5 Vibrational states

The dissociation energy is defined as the energy of the lowest electron energy and the limit value of the potential energy functions as  $R$  approaches infinity. In order to compare the calculations using the dissociation energy we first need to find the vibrational states of the molecules. Introducing a coordinate shift of the constituent atoms of the molecule and describing their positions in terms of the translational motion of their centre of mass as well as their relative motion with respect to each other, the relative motion can be separated in both a rotational and a vibrational Schrödinger equation [3]. The one of interest here is the vibrational one

$$-\frac{\hbar^2}{2\mu} \frac{d^2\phi(x)}{dx^2} + V(x)\phi(x) = E_{\text{vib}}\phi(x), \quad (4.40)$$

where  $\mu = \frac{M_1 M_2}{M_1 + M_2}$  is the reduced mass of the nuclei and  $V(x)$  is the effective potential given by the solution to the Hartree-Fock equations of this particular molecule.  $x$  is the difference from the positions of the nuclei to the equilibrium distance.

Now, in order to solve the Schrödinger equation consider trial functions of the form

$$\phi = \sum_m c_m \chi_m, \quad (4.41)$$

where  $\chi_n$  form a complete orthogonal set as solutions to the Schrödinger equation without the potential

$$-\frac{\hbar^2}{2\mu} \frac{d^2\chi_m(x)}{dx^2} = E_m \chi_m(x), \quad (4.42)$$

with eigenvalues  $E_m$ . Assuming all  $\chi_m$  to be normalised the orthogonality condition requires that

$$\langle \chi_m | \chi_n \rangle = \delta_{mn}, \quad (4.43)$$

which will be exploited later on. By substituting equation 4.41 into equation 4.40 we get

$$\left\{ -\frac{\hbar}{2\mu} \frac{d^2}{dx^2} + V(x) \right\} \sum_m c_m \chi_m = E_{\text{vib}} \sum_m c_m \chi_m, \quad (4.44)$$

of which we can multiply by the complex conjugate of  $\phi$  and integrate yielding

$$\sum_n \sum_m c_n^* c_m \langle \chi_n | \hat{T} + V(x) | \chi_m \rangle = \sum_n \sum_m E_{\text{vib}} c_n^* c_m \langle \chi_n | \chi_m \rangle. \quad (4.45)$$

The left side of this equation can be split up into two component whereas the first component becomes

$$\left\langle \chi_n \left| -\frac{\hbar}{2\mu} \frac{d^2}{dx^2} \right| \chi_m \right\rangle = E_m \langle \chi_n | \chi_m \rangle = E_m \delta_{nm}, \quad (4.46)$$

and the second part becomes just

$$\langle \chi_n | V(x) | \chi_m \rangle = \int \chi_n^* V(x) \chi_m dx \equiv V_{nm}. \quad (4.47)$$

Equation 4.45 can in fact be described by a matrix equation

$$\mathbf{H}\mathbf{c} = E_{\text{vib}}\mathbf{c}, \quad (4.48)$$

of which the elements  $H_{nm} = E_m \delta_{nm} + V_{nm}$  can be evaluated by applying functions that obey equation 4.42. By raising two infinitely tall potential walls in the potential, one close to  $x = -R_e$  and another wall on the other side of the quantum well, the trial functions can be thought of as standing sine waves as in an infinitely deep quantum well yielding

$$\chi_n(x) = (2/d)^{1/2} \sin \frac{n\pi x}{d}, \quad n = 1, 2, 3, 4, 5, \dots \quad (4.49)$$

where  $d$  is the distance between the raised quantum walls and  $\chi_n$  is normalised. Substituting  $\chi_n$  into equation 4.42 gives rise to the eigenvalues

$$E_n = \frac{1}{2\mu} \left( \frac{n\pi}{d} \right)^2. \quad (4.50)$$

After which only  $V_{nm}$  stands to be numerically evaluated.

The script used for the calculations is presented in appendix C.4

Results are given in figures 4.10, 4.11 and 4.12. Dissociation energies are given from the bottom of the quantum well,  $D_e$ , and from the lowest vibrational state,  $D_0$ . Furthermore the lowest vibrational transition is given,  $\omega_e$ . All units are in Hartrees. Wave functions are normalised arbitrarily to fit the energy scale. The dissociation energies are found with respect to the sum of the single-atom energies of the molecules. The energy scale has been shifted so that this

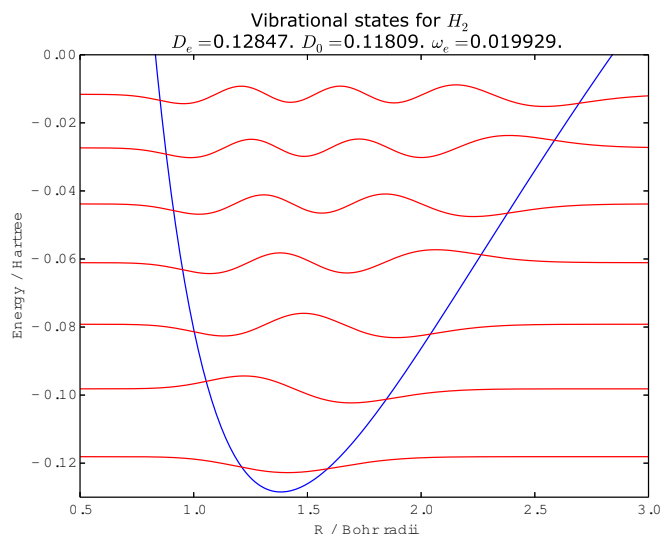


Figure 4.10: Vibrational states for  $H_2$ . All energy units are in Hartrees. Wave functions are normalised arbitrarily to fit the energy scale. The dissociation energies are found with respect to the sum of the single-atom energies of the molecules. The energy scale has been shifted so that this sum of the single-atom energies is zero.

sum of the single-atom energies is zero. The mass of the Hydrogen nucleus is taken to be  $m_H = 1836m_e$ , where  $m_e$  is the mass of an electron. The Lithium atom is taken to be  $7m_H$  as it most often consists of 3 protons and 4 neutrons.

The Dihydrogen molecule exhibits 7 vibrational states, but we note that the net potential exceeds the sum of single-atom energies. The quantum well is therefore expected to be too shallow and there may be more vibrational states. This is confirmed by the dissociation energy,  $D_0$ , which is found to be 0.11809 Hartree. For comparison [4] found this value to be 0.16457 Hartree. This confirms the suspicion of the quantum well being too shallow. The  $\omega_e$  value of [4] is 0.01057 Hartree, which suggests a wider quantum well than ours, since narrow quantum wells result in rotational states farther apart.

The LiH molecule exhibits 11 vibrational states and compared to the dissociation energy,  $D_0$ , of [4] of 0.08926 Hartree, our result of 0.03636 Hartree is far from accurate. This looks like the case of the  $H_2$  molecule and can be explained accordingly.

The  $Li_2$  molecule exhibits some unexpected properties. The energies of the potential lies above the reference energy of the sum of single-atom energies. This suggests no bound state for this molecule, but a bound state is documented by [4] with a dissociation energy of 0.03844 Hartree. This behaviour needs further research.

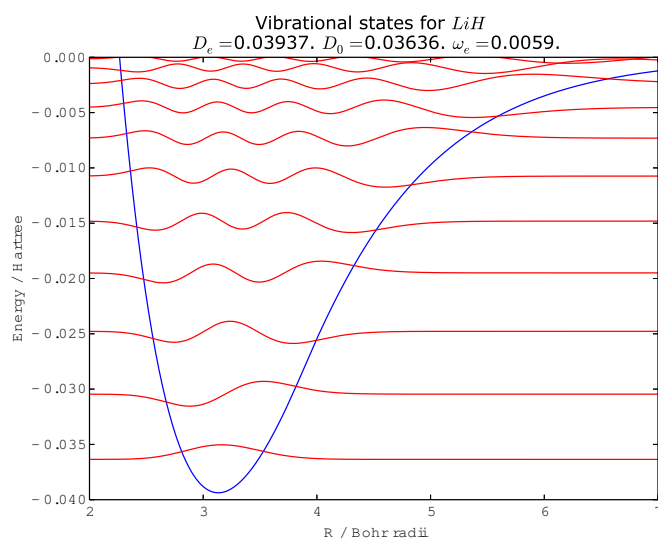


Figure 4.11: Vibrational states for  $LiH$ . All energy units are in Hartrees. Wave functions are normalised arbitrarily to fit the energy scale. The dissociation energies are found with respect to the sum of the single-atom energies of the molecules. The energy scale has been shifted so that this sum of the single-atom energies is zero.

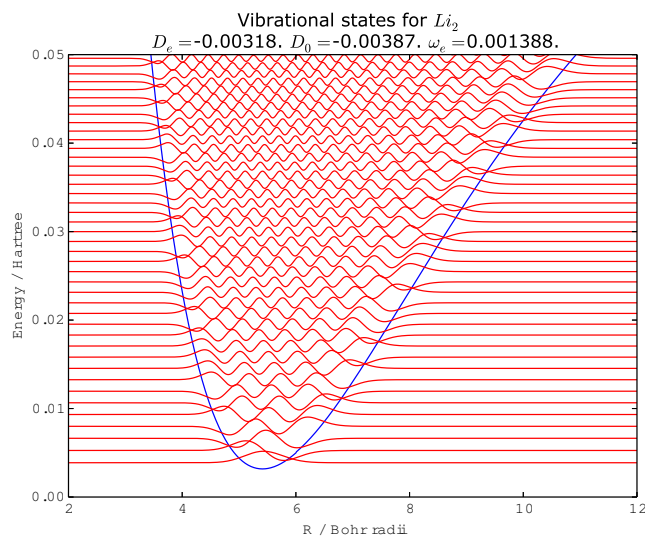


Figure 4.12: Vibrational states for  $Li_2$ . All energy units are in Hartrees. Wave functions are normalised arbitrarily to fit the energy scale. The dissociation energies are found with respect to the sum of the single-atom energies of the molecules. The energy scale has been shifted so that this sum of the single-atom energies is zero.

## 4.6 Conclusion

In this chapter the electronic energy as a function of the internuclear distance as well as the potential energy functions of different diatomic molecules have been calculated. The calculations yielded an electronic energy function that behaves according to our expectations of all the molecules. The behaviour of the potential energy function also matches our expectations and the calculated bond length agrees quite well to experimental data. The results of the molecules HeH and He<sub>2</sub> shows very weak bondings and long bond lengths agreeing to the fact that Helium is an inert gas. When comparing the calculations with experimental data of the dissociations energy the picture is very different. In these calculations the agreement between theory and experimental results is very low. In the molecules H<sub>2</sub> and LiH the disagreement is due to a flaw of the Hartree-Fock method where the energy scale exceeds the sum of the single-atom energies thereby resulting in a wrong shaped quantum well. However in the molecule Li<sub>2</sub> it is noticed that the potential energy functions does not reach below the sum of the single-atom energies thereby not encouraging to a bond formation.

## Chapter 5

# Conclusion

The aim of this project was to calculate the electron and potential energy functions of diatomic molecules using the Hartree-Fock method. In this report the theory behind the Hartree-Fock method is derived, explained and the implementation through Hartree-Fock-Roothaan equations has been presented.

Using Gaussian-type orbitals the theory was implemented in calculation of the electron energy of the elements. To ensure convergence of the Self Consistent Field procedure, Pulay mixing was implemented. To demonstrate the Hartree-Fock method of central field problems it was used to calculate the electron energy of neutral atoms from  $Z = 1$  through  $Z = 103$ , as well as for anions and cations of the 52 first elements with good accuracy. The deviations of the calculations are mostly due to the spherical average approximation adapted in the method. However the inaccuracy of the ionic calculations may be caused by the not fully optimised basis functions.

Using the electron energy of the elements the project sought to calculate the polarizability of the elements using the Hartree-Fock method. We were unable to compute good results but we suspect this is due to the Coulomb and exchange matrices. This needs further research.

Finally the Hartree-Fock method was implemented for diatomic molecules. Cartesian Gaussian basis functions were used to calculate the molecular matrix elements analytically. Using these matrix elements the energy functions and potential energy functions of the molecules  $H_2$ ,  $HeH$ ,  $He_2$ ,  $LiH$  and  $Li_2$  were calculated. The calculations yielded ambiguous results. The calculated bond lengths showed reasonable agreements to experimental results whereas comparisons in dissociation energies and vibrational states yielded unacceptable precision.





# Appendix A

## Matrix elements for atomic calculations

This appendix contains the Coulomb and exchange matrix elements used in the atomic calculations. The following has been computed using Wolfram Mathematica 10. Each basis function,  $\phi_{i\mu}$ , where  $\mu$  contains the significant quantum numbers of the electron, consists of the radial Gaussian basis and an angular part. Let the angular part be determined by the azimuthal quantum number and the respective Gaussian basis function by the Gaussian coefficient.

### A.1 The coulomb elements

Introducing the notation

$$J_{nl_i, nl_j} = \langle \phi_{i, nl_i} \phi_{j, nl_j} | \hat{J} | \phi_{i', n'l'_i} \phi_{j', n'l'_j} \rangle, \quad (\text{A.1})$$

and the definitions  $p = b_i + b_j$  and  $q = b_{i'} + b_{j'}$  the Coulomb matrix elements may be written:

$$J_{ns, ns} = \frac{\sqrt{\pi}}{8pq\sqrt{p+q}} \quad (\text{A.2})$$

$$J_{ns, np} = \frac{3p+2q}{2p(p+q)} \quad (\text{A.3})$$

$$J_{ns, nd} = \frac{15p^2 + 20pq + 8q^2}{4p^2(p+q)^2} \quad (\text{A.4})$$

$$J_{ns, nf} = \frac{3(35p^3 + 70p^2q + 56pq^2 + 16q^3)}{8p^3(p+q)^3} \quad (\text{A.5})$$

$$J_{np,ns} = \frac{2p + 3q}{2q(p + q)} \quad (\text{A.6})$$

$$J_{np,np} = \frac{3(2p^2 + 5pq + 2q^2)}{4pq(p + q)^2} \quad (\text{A.7})$$

$$J_{np,nd} = \frac{3(10p^3 + 35p^2q + 28pq^2 + 8q^3)}{8p^2q(p + q)^3} \quad (\text{A.8})$$

$$J_{np,nf} = \frac{3(70p^4 + 315p^3q + 378p^2q^2 + 216pq^3 + 48q^4)}{16p^3q(p + q)^4} \quad (\text{A.9})$$

$$J_{nd,ns} = \frac{8p^2 + 20pq + 15q^2}{4q^2(p + q)^2} \quad (\text{A.10})$$

$$J_{nd,np} = \frac{3(8p^3 + 28p^2q + 35pq^2 + 10q^3)}{8pq^2(p + q)^3} \quad (\text{A.11})$$

$$J_{nd,nd} = \frac{15(8p^4 + 36p^3q + 63p^2q^2 + 36pq^3 + 8q^4)}{16p^2q^2(p + q)^4} \quad (\text{A.12})$$

$$J_{nd,nf} = \frac{15(56p^5 + 308p^4q + 693p^3q^2 + 594p^2q^3 + 264pq^4 + 48q^5)}{32p^3q^2(p + q)^5} \quad (\text{A.13})$$

$$J_{nf,ns} = \frac{3(16p^3 + 56p^2q + 70pq^2 + 35q^3)}{8q^3(p + q)^3} \quad (\text{A.14})$$

$$J_{nf,np} = \frac{3(48p^4 + 216p^3q + 378p^2q^2 + 315pq^3 + 70q^4)}{16pq^3(p + q)^4} \quad (\text{A.15})$$

$$J_{nf,nd} = \frac{15(48p^5 + 264p^4q + 594p^3q^2 + 693p^2q^3 + 308pq^4 + 56q^5)}{32p^2q^3(p + q)^5} \quad (\text{A.16})$$

$$J_{nf,nf} = \frac{315(16p^6 + 104p^5q + 286p^4q^2 + 429p^3q^3 + 286p^2q^4 + 104pq^5 + 16q^6)}{64p^3q^3(p + q)^6} \quad (\text{A.17})$$

## A.2 The exchange elements

For the exchange matrix elements we have that  $p = b_i + b_{j'}$  and  $q = b_j + b_{i'}$

$$K_{ns,ns} = \frac{\sqrt{\pi}}{8pq\sqrt{p+q}} \quad (\text{A.18})$$

$$K_{np,ns} = \frac{1}{2(p+q)} \quad (\text{A.19})$$

$$K_{nd,ns} = \frac{3}{4(p+q)^2} \quad (\text{A.20})$$

$$K_{nf,ns} = \frac{15}{8(p+q)^3} \quad (\text{A.21})$$

$$K_{ns,ns} = \frac{1}{2(p+q)} \quad (\text{A.22})$$

$$K_{np,ns} = \frac{2p^2 + 7pq + 2q^2}{4pq(p+q)^2} \quad (\text{A.23})$$

$$K_{nd,ns} = \frac{4p^2 + 23pq + 4q^2}{8pq(p+q)^3} \quad (\text{A.24})$$

$$K_{nf,ns} = \frac{3(6p^2 + 47pq + 6q^2)}{16pq(p+q)^4} \quad (\text{A.25})$$

$$K_{ns,ns} = \frac{3}{4(p+q)^2} \quad (\text{A.26})$$

$$K_{np,ns} = \frac{4p^2 + 23pq + 4q^2}{8pq(p+q)^3} \quad (\text{A.27})$$

$$K_{nd,ns} = \frac{3(8p^4 + 40p^3q + 99p^2q^2 + 40pq^3 + 8q^4)}{16p^2q^2(p+q)^4} \quad (\text{A.28})$$

$$K_{nf,ns} = \frac{9(8p^4 + 52p^3q + 193p^2q^2 + 52pq^3 + 8q^4)}{32p^2q^2(p+q)^5} \quad (\text{A.29})$$

$$K_{ns,ns} = \frac{15}{8(p+q)^3} \quad (\text{A.30})$$

$$K_{np,ns} = \frac{3(6p^2 + 47pq + 6q^2)}{16pq(p+q)^4} \quad (\text{A.31})$$

$$K_{nd,ns} = \frac{9(8p^4 + 52p^3q + 193p^2q^2 + 52pq^3 + 8q^4)}{32p^2q^2(p+q)^5} \quad (\text{A.32})$$

$$K_{nf,ns} = \frac{9(80p^6 + 552p^5q + 1698p^4q^2 + 3607p^3q^3 + 1698p^2q^4 + 552pq^5 + 80q^6)}{64p^3q^3(p+q)^6} \quad (\text{A.33})$$



## Appendix B

# Hermite Polynomials

This appendix describe the Hermite polynomials used to calculate the molecular matrix elements of the project. The section is written with the aid of [3] and [7]. Solutions to the differential equation

$$\frac{\partial^2 F}{\partial x^2} - 2x \frac{\partial F}{\partial x} + 2nF = 0 \quad (\text{B.1})$$

are called Hermite polynomials and are often denoted  $H_n(x)$ . There are several ways of calculating the Hermite polynomials like the independent definition

$$H_n(x) = (-1)^n e^{x^2} \frac{\partial^n}{\partial x^n} e^{-x^2}. \quad (\text{B.2})$$

However they may also be calculated using the formula

$$H_n(x) = \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{(-1)^k n!}{k!(n-2k)!} (2x)^{n-2k} \quad (\text{B.3})$$

From these the first few Hermite polynomials can be found to be

$$H_0(x) = 1 \quad H_1(x) = 2x \quad H_2(x) = 4x^2 - 2 \quad H_3(x) = 8x^2 - 12x \quad (\text{B.4})$$

When normalized the Hermite polynomials form an orthonormal set on the interval  $(-\infty, \infty)$  such that

$$\langle H_n | H_m \rangle = \int_{-\infty}^{\infty} H_n(x)^* H_m(x) dx = \delta_{nm}. \quad (\text{B.5})$$

Using the properties of the Hermite polynomials it is possible to analytically solve quite difficult problems.



# Appendix C

## Python code

### C.1 Hartree\_Fock.py

```
1 '''
2 Created on Mar 3, 2015
3
4 @author: anders
5 '''
6 from Classes import LoadAtom, Atom
7 from numpy import dot, transpose, conj, divide, sqrt, add, zeros,
8     multiply,\
9     round, sum, copy, argmax
10 from scipy.linalg.decomp import eigh
11 from numpy.linalg.linalg import solve
12 from numpy.core.umath import subtract
13 from MatrixFunctions_Atoms import BuildTotalMatrices
14 from Petersson_Hellsing import BuildAtomicMatrices
15
16 def threedot(m1,m2,m3):
17     return dot(transpose(conj(m1)),dot(m2,m3))
18
19 def Normalise(c,S):
20
21     for i in range(len(c)):
22         norm = threedot(c[i],S,c[i])
23         c[i] = divide(c[i],sqrt(norm))
24
25     return c
26
27 def Eigenstates(H,S):
28
29     E,c = eigh(H,S)
30     c = transpose(c)
31     c = Normalise(c, S)
32
33     return E, c
34
35 def UpdateHF(NumberOfElectrons,nList,lList,sList,s,C,H0,J,K):
36
37     H = copy(H0)
38
39     for p in range(len(C[0])):
40         for q in range(p,len(C[0])):
41             Sum = 0
42             pold = 0
43             lold = 0
44             sold = 0
45             dH = 0
46             for e in range(NumberOfElectrons):
47                 if(nList[e] != pold or\
```

```

48         lList[e] != lold or\
49         sList[e] != sold):
50         dH = threedot(C[e], J[p][q], C[e])
51         if(sList[e] == s):
52             dH = dH - threedot(C[e], K[p][q], C[e])
53         pold = nList[e]
54         lold = lList[e]
55         sold = sList[e]
56         Sum = Sum + dH
57         H[p][q] = HO[p][q] + Sum
58         H[q][p] = H[p][q]
59
60     return H
61
62 def HartreeFock(H,S,J,K,MatrixIndices,NumberOfElectrons,
63               nList,lList,sList):
64     E = [None]*NumberOfElectrons
65     CO = [None]*NumberOfElectrons
66     C = copy(CO)
67     Cpulay = [[]]*NumberOfElectrons
68     Esumold = 1
69     HO = copy(H)
70
71     for it in range(100):
72         #print(it)
73         for s in range(-1,2,2):
74             if(it>0):
75                 H = UpdateHF(NumberOfElectrons,nList,lList,sList,s,C,HO,J,K)
76                 E0, c = Eigenstates(H,S)
77                 OrbitalIndex = OrbitalAnalyser(E0, c, MatrixIndices)
78                 for e in range(NumberOfElectrons):
79                     if(sList[e] == s):
80                         E[e] = E0[OrbitalIndex[nList[e]][lList[e]]]
81                         CO[e] = c[OrbitalIndex[nList[e]][lList[e]]]
82                         if(CO[e] != None and CO[e][0] < 0):
83                             CO[e] = multiply(-1,CO[e])
84                 Cpulay, C = Pulay(Cpulay,CO,S,NumberOfElectrons)
85                 Esum = sum(E)
86                 if(abs(Esum/Esumold-1)<1e-10):
87                     break
88                 Esumold = Esum
89
90     return HO, E, C
91
92 def CoefficientMix(C,CO,S,alpha=0.5):
93
94     if(C[0] == None):
95         return CO
96
97     for e in range(len(C)):
98         C[e] = add(multiply(alpha,CO[e]),multiply(1-alpha,C[e]))
99
100    C = Normalise(C, S)
101
102    return C
103
104 def Pulay(Cpulay,CO,S,NumberOfElectrons):
105
106     for e in range(NumberOfElectrons):
107         tempCpulay = copy(Cpulay[e])
108         tempCpulay = tempCpulay.tolist()
109
110         if(len(tempCpulay) < 5):
111             tempCpulay.append(CO[e])
112         else:
113             for i in range(len(tempCpulay)-1):
114                 tempCpulay[i] = tempCpulay[i+1]
115                 tempCpulay[-1] = CO[e]
116
117         if(len(tempCpulay)==5):
118             residual = list()
119             for k in range(len(tempCpulay)-1):
120                 op = subtract(tempCpulay[k+1],tempCpulay[k])

```





```

195         Atom.Configuration.Spin)
196
197     Energy = TotalEnergy(Atom,H,E,C)
198
199     return Energy
200
201 if __name__ == '__main__':
202
203     pass

```

## C.2 Polarizability.py

```

1  '''
2  Created on Mar 3, 2015
3
4  @author: anders
5  '''
6  from Classes import LoadAtom, Atom
7  from numpy import dot, transpose, conj, divide, sqrt, add, zeros,
8      multiply,\
9      round, sum, copy, argmax, pi, empty, polyfit
10 from scipy.linalg.decomp import eigh
11 from MatrixFunctions_Atoms import Hamiltonian0Matrix, OverlapMatrix,
12     CoulombMatrix_new,\
13     ExchangeMatrix_new
14 from numpy.linalg.linalg import solve
15 from numpy.core.umath import subtract
16 from matplotlib.pyplot import plot, show, subplots, rc
17
18 def threedot(m1,m2,m3):
19
20     return dot(transpose(conj(m1)),dot(m2,m3))
21
22 def Normalise(c,S):
23
24     for i in range(len(c)):
25         norm = threedot(c[i],S,c[i])
26         c[i] = divide(c[i],sqrt(norm))
27
28     return c
29
30 def Eigenstates(H,S):
31
32     E,c = eigh(H,S)
33     c = transpose(c)
34     c = Normalise(c, S)
35
36     return E, c
37
38 def UpdateHF(Atom,s,C,H0,J,K):
39
40     H = copy(H0)
41
42     for p in range(len(C[0])):
43         for q in range(len(C[0])):
44             Sum = 0
45             pold = 0
46             lold = 0
47             sold = 0
48             for e in range(Atom.Configuration.Z):
49                 if(Atom.Configuration.Principal[e] != pold or\
50                     Atom.Configuration.Azimuthal[e] != lold or\
51                     Atom.Configuration.Spin[e] != sold):
52                     dH = threedot(C[e], J[p][q], C[e])
53                     if(Atom.Configuration.Spin[e] == s):
54                         dH = dH - threedot(C[e], K[p][q], C[e])
55                     pold = Atom.Configuration.Principal[e]
56                     lold = Atom.Configuration.Azimuthal[e]
57                     sold = Atom.Configuration.Spin[e]
58                     Sum = Sum + dH
59             print('p='+str(p)+' q='+str(q)+' Sum='+str(Sum))
60             H[p][q] = H[p][q] + Sum

```



```

126     if CO[e] != None and CO[e][0] < 0):
127         CO[e] = multiply(-1,CO[e])
128     Cpulay, C = Pulay(Cpulay,CO,S,Atom.Configuration.Z)
129     Esum = sum(E)
130     print('it='+str(it))
131     if(abs(Esum/Esumold-1)<1e-10):
132         break
133     Esumold = Esum
134     return H0, E, C
135
136 def CoefficientMix(C,CO,S,alpha=0.5):
137
138     if(C[0] == None):
139         return CO
140
141     for e in range(len(C)):
142         C[e] = add(multiply(alpha,CO[e]),multiply(1-alpha,C[e]))
143
144     C = Normalise(C, S)
145
146     return C
147
148 def Pulay(Cpulay,CO,S,Z):
149
150     for e in range(Z):
151         tempCpulay = copy(Cpulay[e])
152         tempCpulay = tempCpulay.tolist()
153
154         if(len(tempCpulay) < 5):
155             tempCpulay.append(CO[e])
156         else:
157             for i in range(len(tempCpulay)-1):
158                 tempCpulay[i] = tempCpulay[i+1]
159                 tempCpulay[-1] = CO[e]
160
161         if(len(tempCpulay)==5):
162             residual = list()
163             for k in range(len(tempCpulay)-1):
164                 op = subtract(tempCpulay[k+1],tempCpulay[k])
165                 residual.append(op)
166
167             M = zeros([len(residual)+1,len(residual)+1])
168
169             for i in range(len(residual)+1):
170                 for j in range(len(residual)+1):
171                     if((i == len(residual) or j == len(residual)) and i != j):
172                         M[i][j] = -1
173                     elif(i == len(residual) and j == len(residual)):
174                         M[i][j] = 0
175                     else:
176                         M[i][j] = dot(residual[i],residual[j])
177
178             N = zeros([len(residual)+1])
179             N[-1] = -1
180             x = solve(M,N)
181             tempC = 0
182             for i in range(len(tempCpulay)):
183                 tempC = tempC+multiply(x[i],tempCpulay[i])
184
185             Cpulay[e] = tempCpulay
186             alpha = 0.5
187             CO[e] = add(multiply(alpha,CO[e]),multiply(1-alpha,tempC))
188
189     C = Normalise(CO, S)
190
191     return Cpulay, C
192
193 def TotalEnergy(Atom,H,E,C):
194
195     Sum = 0
196     for e in range(Atom.Configuration.Z):
197         Sum = Sum + E[e]
198         Sum = Sum + threedot(C[e], H, C[e])
199

```

```

200     return round(Sum/2,10)
201
202 def OrbitalAnalyser(E,C,MatrixIndices):
203
204     startI = MatrixIndices[0]
205     endI = MatrixIndices[1]
206     tempIndex = [[],[ ]]
207     n = [1,2,3,4]
208
209     for e in range(len(E)):
210         Sum = [0]*len(C[0])
211         for l in range(len(startI)):
212             Sum[l] = sum(abs(C[e][startI[l]:endI[l]]))
213         lorb = argmax(Sum)
214         tempIndex[0].append(n[lorb])
215         tempIndex[1].append(lorb)
216         n[lorb] = n[lorb] + 1
217
218     OrbitalIndex = zeros([max(tempIndex[0])+1,4])
219
220     for i in range(len(tempIndex[0])):
221         OrbitalIndex[tempIndex[0][i]][tempIndex[1][i]] = i
222
223     return OrbitalIndex
224
225 def PolarizationElement(bi,bj,li,lj,EField):
226
227     if((li == 0 and lj == 1) or (li == 1 and lj == 0)):
228         return (EField*(3*pi)**(0.5))/(8*(bi + bj)**(5/2))
229     elif((li == 1 and lj == 2) or (li == 2 and lj == 1)):
230         return (EField*(15*pi)**(0.5))/(8*(bi + bj)**(7/2))
231     elif((li == 2 and lj == 3) or (li == 3 and lj == 2)):
232         return (9*EField*(35*pi)**(0.5))/(32*(bi + bj)**(9/2))
233     else:
234         return 0
235
236 def PolarizationMatrix(Basis,li,lj,EField):
237
238     polarizationMatrix = zeros([len(Basis[li]),len(Basis[lj])])
239
240     for i in range(len(Basis[li])):
241         for j in range(len(Basis[lj])):
242             polarizationMatrix[i][j] = PolarizationElement(Basis[li][i],
243                 Basis[lj][j], li, lj, EField)
244
245     return polarizationMatrix
246
247 def RunOneAtom(Z):
248
249     Atom = LoadAtom(Z,False,True,True)
250
251     print('Atom loaded: '+str(Atom.Configuration.Z)+' : '+Atom.Name)
252
253     Eknold = list()
254     res = 10
255     Efieldmax = 0.001
256
257     EField = empty(res+1)
258     for i in range(len(EField)):
259         if(i==0):
260             EField[i] = -Efieldmax
261         else:
262             EField[i] = EField[i-1]+2*Efieldmax/res
263
264     #polConstRel = [0.8,0.9,1,1.1,1.2]
265     H,E,Czero = HartreeFock(Atom, 0)
266
267     for i in range(len(EField)):
268         bestEnergy = 1000
269         #for const in range(len(polConstRel)):
270             H,E,C = HartreeFock(Atom,EField[i],None,1)
271             Energy = TotalEnergy(Atom,H,E,C)
272             #if(Energy<bestEnergy):
273                 bestEnergy = Energy

```

```

273     #bestpolConst = polConstRel[const]
274     Eknold.append(bestEnergy)
275     #print(bestpolConst)
276
277     #print(Atom.Koga.Energy)
278     #print(Eknold)
279
280     fit = polyfit(EField,Eknold,2)
281     print('alpha='+str(-fit[-3]*2))
282     #print('gamma='+str(-fit[-1]*24))
283
284     Eknoldtest = fit[-3]*EField**2+fit[-2]*EField+fit[-1]
285
286     rc('text', usetex=True)
287     rc('font', family='serif')
288
289     fig, ax = subplots()
290
291     ax.set_xlabel(r'Electric field strength,  $E$  / a.u.')
292     ax.set_ylabel(r'Energies,  $E$  / Hartree')
293     ax.set_title('Energy levels with applied external electric field\n
294                 Results for  $Z=1$ , Hydrogen')
295
296     rect = plot(EField,Eknold)
297
298     fig.tight_layout()
299
300     show()
301
302 if __name__ == '__main__':
303     pass

```

### C.3 Molecules.py

```

1  '''
2  Created on 08/04/2015
3
4  @author: Sommer
5  '''
6  from numpy import round, empty, argmin
7  from numpy.ma.core import copy, add
8  from numpy.core.umath import multiply
9  from Classes import LoadAtom
10 from numpy.core.multiarray import zeros
11 from MatrixFunctions_Molecules import BuildTotalMolMatrices,\
12     Overlap_Mol_Matrix, Hamilton_Mol_Matrix, Coulomb_M_Matrix
13 from builtins import sum
14 from Hartree_Fock import Eigenstates, threedot, Pulay,\
15     Normalise
16 import matplotlib.pyplot as pp
17 from posix import getcwd
18 import matplotlib
19 from matplotlib.pyplot import rc
20
21 def Index(n,l):
22
23     if(n==1 and l==0):
24         return 0
25     elif(n==2 and l==1):
26         return 2
27     elif(n==2 and l==0):
28         return 1
29     elif(n==3 and l==2):
30         return 3
31     elif(n==3 and l==1):
32         return 4
33     elif(n==3 and l==0):
34         return 2
35
36 def UpdateHF(N,principal,spin,s,C,H0,J,K):
37     H = copy(H0)

```

```

38 for p in range(len(C[0])):
39     for q in range(len(C[0])):
40         Sum = 0
41         pold = 0
42         sold = 0
43         dH = 0
44         for e in range(N):
45             if(principal[e] != pold or spin[e] != sold):
46                 dH = threedot(C[e], J[p][q], C[e])
47                 if(spin[e] == s):
48                     dH = dH - threedot(C[e], K[p][q], C[e])
49                 pold = principal[e]
50                 sold = spin[e]
51                 Sum = Sum + dH
52             H[p][q] = H0[p][q] + Sum
53
54     return H
55
56 def CoefMixing(C,C0,Z,alpha=0.9):
57
58     for e in range(Z):
59         C[e] = add(multiply(alpha,C0[e]),multiply(1-alpha,C[e]))
60
61     return C
62
63 def RunMolecule(Z1,Z2,ConfLists,Bx=0,By=0,Bz=0):
64
65     Atom1 = LoadAtom(Z1,WithDiffuse=True)
66     Atom2 = LoadAtom(Z2,WithDiffuse=True)
67
68     Number = Atom1.Configuration.Z+Atom2.Configuration.Z
69
70     size = BuildTotalMolMatrices(Atom1, Atom2)
71
72     Overlap,Basis = Overlap_Mol_Matrix(Atom1, Atom2, size,Bx,By,Bz)
73     Hamilton = Hamilton_Mol_Matrix(Atom1,Atom2,size,Bx,By,Bz)
74     Coulomb,Exchange,legleg,yoloExchange, yolocoulomb =
75         Coulomb_M_Matrix(Atom1,Atom2,size,Bx,By,Bz)
76
77     E = [None]*(Number)
78     C0 = [None]*(Number)
79     Cpulay = [[]]*(Number)
80     Esumold = 1
81     H0 = copy(Hamilton)
82
83     principal = ConfLists[0]
84     azimuthal = ConfLists[1]
85     spin = ConfLists[2]
86
87     for it in range(300):
88         print(it)
89         for s in range(-1,2,2):
90             if(it>0):
91                 Hamilton = UpdateHF(Number,principal,spin,s,C,H0,Coulomb,Exchange)
92                 E0, c = Eigenstates(Hamilton,Overlap)
93                 for e in range(Number):
94                     if(spin[e] == s):
95                         index = Index(principal[e], azimuthal[e])
96                         E[e] = E0[index]
97                         C0[e] = c[index]
98                         CO[e][len(CO[e])/2:len(CO[e])] = CO[e][0:len(CO[e])/2]
99                     if(CO[e] != None and CO[e][0] < 0):
100                         CO[e] = multiply(-1,CO[e])
101
102         if(it<1):
103             C = copy(C0)
104         else:
105             Cpulay, C = Pulay(Cpulay, C0, Overlap, Number)
106             C = Normalise(C, Overlap)
107             Esum = sum(E)
108             if(abs(Esum/Esumold-1)<1e-10):
109                 break
110             Esumold = Esum
111
112     return H0, E, C, Number

```

```

111 def TotalEnergy(N,H,E,C):
112
113     Sum = 0
114     for e in range(N):
115         Sum = Sum + E[e]
116         Sum = Sum + threedot(C[e], H, C[e])
117
118     return round(Sum/2,10)
119
120 def RunFullPotential(Z1,Z2,ConfLists,MolName,Rmax=8,Resolution=0.1):
121
122     R = empty(Rmax/Resolution)
123     E = empty(Rmax/Resolution)
124     V = empty(Rmax/Resolution)
125
126     for i in range(len(R)):
127         R[i] = (i+1)*Resolution
128
129     currentDirectory = getcwd()
130     fileObject = open(currentDirectory+'/mol_results/'+MolName+'.txt','w')
131
132     for i in range(len(R)):
133         #print(i)
134         H0, Eig, C, Number = RunMolecule(Z1, Z2,ConfLists, Bz=R[i])
135         E0 = TotalEnergy(Number,H0,Eig,C)
136         E[i] = E0
137         V0 = E0 + (Z1*Z2)/R[i]
138         V[i] = V0
139         fileObject.write(
140             str(R[i]).rjust(8)+str(E0).rjust(20)+str(V0).rjust(20)+'\n')
141         print(str(i)+'/'+str(len(R)))
142
143     imin = argmin(V)
144     Rmin = R[imin]
145
146     fileObject.close()
147
148     return E, V, R, Rmin
149
150 def PlotMolecule(R,E,V):
151
152     sort = 0
153
154     imin = argmin(V[sort:len(V)])
155     imin = imin + sort
156
157     fig, ax = pp.subplots()
158     line1, = pp.plot(R, E, label=r'$E$')
159     line2, = pp.plot(R, V, label=r'$V$')
160     dot1, = pp.plot([R[imin]],[V[imin]], 'rx', label=r'$R_{min}$ = '+str(R[imin])+' Bohr', markersize=10)
161
162     rc('text', usetex=True)
163     rc('font', family='serif')
164
165     pp.ylim((-38,-10))
166     pp.xlim((0,12))
167     pp.legend(loc=4, handler_map={dot1:matplotlib.legend_handler.HandlerLine2D(numpoints=1)})
168
169     ax.set_xlabel(r'$R$ / Bohr radii')
170     ax.set_ylabel(r'$Energy$ / Hartree')
171     ax.set_title(r'$Electron energy and effective potentials for $Li_2$')
172
173     pp.show()
174
175     pass
176
177 def LoadCalculatedResults(MolName):
178
179     currentDirectory = getcwd()
180     fileObject = open(currentDirectory+'/mol_results/'+MolName+'.txt','r')
181
182     R = list()

```



```

182 E = list()
183 V = list()
184
185 for line in fileObject:
186     splittedline = line.split()
187     R.append(float(splittedline[0]))
188     E.append(float(splittedline[1]))
189     V.append(float(splittedline[2]))
190
191 fileObject.close()
192
193 return R, E, V
194
195 if __name__ == '__main__':
196
197     #afstand = 0.001
198
199     #H,E,C,N = RunMolecule(1,1,0,0,afstand)
200     #Energy = TotalEnergy(N, H, E, C)
201     #print(Energy)
202     '''
203     Principal = [1,1,2,2,3,3]
204     Azimuthal = [0,0,0,0,0,0]
205     Spin = [-1,1,-1,1,-1,1]
206
207     ConfLists = [Principal, Azimuthal, Spin]
208
209     E, V, R, Rmin = RunFullPotential(3,
210                                     3,ConfLists,'Li2_final_eat',12,0.01)
211     '''
212
213     R, E, V = LoadCalculatedResults('Li2_final_combined')
214
215     PlotMolecule(R,E,V)
216
217     pass

```

## C.4 Vibrational\_Diatomic.py

```

1 '''
2 Created on Mar 18, 2015
3
4 @author: anders
5 '''
6 from numpy import add, argmin, sin, pi, trapz, empty, ones, zeros,
7     multiply
8 from scipy.linalg.decomp import eigh
9 from matplotlib.pyplot import subplots, plot, ylim, show, xlim
10 from Molecules import LoadCalculatedResults
11
12 def NumericalVibrationalStates(InternuclearDist,Potential,BasisSize):
13
14     EqIndex = argmin(Potential)
15     #EquilibriumDist = InternuclearDist[EqIndex]
16
17     X = add(InternuclearDist,-InternuclearDist[0])
18     V = add(Potential,-Potential[EqIndex])
19     D_e = -1-Potential[EqIndex]
20     d = InternuclearDist[-1]-InternuclearDist[1]
21
22     H_vib = empty([BasisSize,BasisSize])
23
24     for n in range(BasisSize):
25         for m in range(BasisSize):
26             chi_n = (2/d)**(1/2)*sin( (n+1)*pi*X/d )
27             chi_m = (2/d)**(1/2)*sin( (m+1)*pi*X/d )
28             if n != m:
29                 E = 0
30             else:
31                 E = 1/(2*((1836*1836)/(1836+1836))) * ( (n+1)*pi / d )**2
32                 Vnm = trapz(chi_n*V*chi_m,X)
33                 H_vib[n][m] = E + Vnm

```

```

33
34 e,c = eigh(H_vib)
35
36 return e, c, V, D_e, d,X
37
38 def PlotVibrationalStates(InternuclearDist,Potential,e,c,d,X,D_e):
39
40 V = add(Potential,-D_e)
41 chi = zeros([len(c),len(V)])
42
43 fig, ax = subplots()
44 rect = plot(InternuclearDist,V)
45 for i in range(len(e)):
46     chi = zeros(len(V))
47     if(e[i]-D_e<0):
48         for n in range(len(c[i])):
49             chi = add(chi,c[n][i]*(2/d)**(1/2)*sin((n+1)*pi*X/d))
50             flot = add(ones(len(InternuclearDist))*e[i],-D_e)
51             flot = add(flot,0.003*chi)
52             rect1 = plot(InternuclearDist,flot,'r')
53
54 ylim((-0.13,0))
55 xlim((0.5,3))
56
57 D_0 = D_e-e[0]
58 omega = abs(e[0]-e[1])
59
60 ax.set_xlabel(r'R / Bohr radii')
61 ax.set_ylabel(r'Energy / Hartree')
62 ax.set_title('Vibrational states for $H_2$ \n
        $D_e=${str(round(D_e,5))+'. $D_0=${str(round(D_0,5))+'.
        $\omega_e=${str(round(omega,6))+'.')
63
64 show()
65
66 if __name__ == '__main__':
67     '''
68     res = 0.01
69     Rmax = 8
70
71     R = empty(Rmax/res)
72     for i in range(len(R)):
73         R[i] = (i+1)*res
74     '''
75
76 R, E, V = LoadCalculatedResults('H2_final_withheat')
77 e, c, v, D_e, d,X = NumericalVibrationalStates(R,V,100)
78 print(D_e)
79 PlotVibrationalStates(R,v,e,c,d,X,D_e)
80
81
82 pass

```

## Appendix D

# DVD with code

This page has been left black intentionally for code attachments.



# Bibliography

- [1] Hans Albrecht Bethe, Roman W Jackiw, et al. Intermediate quantum mechanics. 1968.
- [2] S Francis Boys. Electronic wave functions. i. a general method of calculation for the stationary states of any molecular system. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 200, pages 542–554. The Royal Society, 1950.
- [3] Jens Peder Dahl and Bjarne Amstrup. *Introduction to the quantum world of atoms and molecules*, volume 1. World Scientific, 2001.
- [4] KP Huber and G Herzberg. Molecular structure and molecular spectra iv: constants of diatomic molecules. *Van Nostrand-Reinhold, New York*, 1979.
- [5] Toshikatsu Koga, Hiroshi Tatewaki, and Tsuyoshi Shimazaki. Chemically reliable uncontracted gaussian-type basis sets for atoms h to lr. *Chemical Physics Letters*, 328(4):473–482, 2000.
- [6] Toshikatsu Koga, Shinya Watanabe, Katsutoshi Kanayama, Ryuji Yasuda, and Ajit J Thakkar. Improved roothaan–hartree–fock wave functions for atoms and ions with  $n_l=54$ . *The Journal of chemical physics*, 103(8):3000–3005, 1995.
- [7] N.N. Lebedev. *Special Functions and their Applications*. Prentice-Hall, 1965.
- [8] Tomas Petersson and Bo Hellsing. A detailed derivation of gaussian orbital-based matrix elements in electron structure calculations. *Eur. Phys. J.*, 31(1):37, 2010.
- [9] Peter Pulay. Improved scf convergence acceleration. *Journal of Computational Chemistry*, 3(4):556–560, 1982.
- [10] Peter Schwerdtfeger. Table of experimental and calculated static dipole polarizabilities for the electronic ground states of the neutral elements (in atomic units). *Schwerdtfeger.-2012.-[URL] <http://ctcp.massey.ac.nz/dipole-polarizabilities>*, 2014.
- [11] Richard E Sitter Jr and RP Hurst. Hyperpolarizabilities for hartree-fock atoms. *Physical Review A*, 5(1):5, 1972.

- [12] Johannes Stiehler and Juergen Hinze. Calculation of static polarizabilities and hyperpolarizabilities for the atoms he through kr with a numerical rhf method. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 28(18):4055, 1995.
- [13] Jos Thijssen. *Computational physics*. Cambridge University Press, 2007.
- [14] Thomas Voegel, Juergen Hinze, and Frank Tobin. Numerical scf method for the calculation of static polarizabilities and hyperpolarizabilities for atoms, he through ne. *The Journal of Chemical Physics*, 70(3):1107–1111, 1979.