RECENT ADVANCES IN SOFTWARE FOR A DENSITY FUNCTIONAL THEORY OF MOLECULAR FRAGMENTS

by

Victor Hugo González Chávez

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Department of Chemistry

West Lafayette, Indiana

May 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Adam Wasserman, Chair
Department of Chemistry

Dr. Lyudmila Slipchenko
Department of Chemistry

Dr. Shelley Claridge
Department of Chemistry

Dr. Sabre Kais
Department of Chemistry

Approved by:
Dr. Christine Hrycyna

This is the Art of Pretending to Swim.

ACKNOWLEDGMENTS

Adam, I have yet to meet someone more humble yet diligent. Thank you for agreeing to work with me even when I ~~was~~ am clueless about Partition Density Functional Theory. You allowed me to have the space to grow in my own direction. Thank you for your skillful mentoring. Your quiet guidance spoke volumes. If I know how to tell a story in front of an audience, it is because I learned from the best. I am deeply indebted for all your support. Thank you for all the Fridays we shared over a cup of coffee. I am sorry for ordering the one that takes forever to make. In my defense, I still yet have to meet the "Chief Coffee Maker". En retrospectiva, siempre fui un caminante sin camino. Gracias por todo, Adam.

Thank you, Prof. Edgar, for introducing me to the engulfing life of non-interacting electrons. I have been lucky enough to witness the passion you have for this field, but more importantly, for life. It is still hard to believe that you turned the CONACYT upside down so that I could attend the 2016 APS Meeting.

Thank you for giving your talk at that same APS meeting, Prof. Staroverov. After its conclusion, I ran and sat on the carpet of the Baltimore Conference Center and devoured your book chapter. I still hold dear the copy I printed in London. It is, after all, my New Testament. Thank you for teaching me that logic always presents itself. I still hope we can work together on the Laplacian. Food for thought.

Thank you for always rooting for me, Prof. Lyuda. Your constant "I thought you graduated already" is a compliment that will forever lift my spirits. It was an honor and bliss to teach under your wing. I am confident that I will soon appreciate the acquaintance of the effective fragment potential. I someday hope to encounter you during a frigid adventure in the forest.

Thanks to the members of The New Suspenders: Dr. Jonathan Nafziger, Dr. Daniel Whitenack, Dr. Kaili Jiang, Dr. Kelsie Niffenegger, Dr. Daniel Jensen, and Dr. Martín Mosquera. This thesis is a testament to your fantastic work.

Ян and Wendy, thank you for your friendship (10/10). Vienna is not been the same since you left. Thanks for your willingness to go over a subject a hundred million times. It was during our discussions when I learned best.

Thanks to Yuming, I hope our baby thrives now that it is out in the wild. I could not have survived the APS meeting without your help and encouragement.

Finally, thanks to Kui, Yi, and the rest of the folks who have and will join Adam's squad. This is a journey best enjoyed in tandem.

Thank you to everyone at the Slipchenko group. You were the first to make me feel at home. My sister Kbi, la princesa, la parcera. I thrived after I met you, and I never got over you leaving. Thank you for always providing the best company. Yen, thanks for your overwhelming energy and confidence. I hope one day I am also able to embody it.

Thanks to everyone at the Staroverov group. You folks made me feel at home since day one. Thank you, Dr. Egor Ospadov, for your kind actions to help me settle down. Thank you, Dasha. I was the happiest during the shared time on the bench outside the Chemistry building. Your company was always soothing and nurturing. If anyone says I was successful in Canada, it was only because of you.

Thanks to all of the Software Scientists at MolSSI. Dr. Jessica Nash, your down-to-earth personality and kindness are why I love engaging with the Institute. Thank you for teaching me all I know about Python. Your ability to lead has no bounds. I hope I get to keep learning from you. Dr. Daniel G. A. Smith, thank you for creating the tools that helped define my work. I have the utmost respect for you and your craft. I hope I get to work with you in the future. My wonderful mentor, Dr. Taylor A. Barnes, thank you for keeping up with me for so long. I hope my work did not disappoint.

Thank you, Prof. Ashley Ringer-McDonald. I am still in shock that a random curiosity for Psi4Education led me to one of the dearest projects of my time as a student. I am deeply in debt for all the invitations to teach along your side. You are the coolest professor I know. I hope we can discuss together the perfect texture of a focaccia one day. Thanks to Dr. Lori Burns for always steering me along well-defined $PATHs. Last but not least, thanks to Prof. Ryan Fortenberry, Prof. Brandon Magers, soon-to-be Prof. Dom Sirianni, and soon-to-be Dr. Ben Peyton for being a fantastic crew.

Although I only got to interact very briefly with them, I want to acknowledge these people for sharing their brilliance with me: Thank you, Prof. Mel Levy, for sharing your passion for the adiabatic connection during our time in Switzerland. Prof. Staroverov witnessed that your PRA from 1982 is what kept me in this field.

Thank you, Dr. Alex Gaiduk, for our fruitful conversations. I still owe you and your family dinner.

Thank you, Prof. Paul Ayers, for your kind words about my research. Our conversation made a substantial positive impact on my perception of my work. Thanks for letting me know I am on the right track.

Thank you, Dr. Rogelio Cuevas Saavedra, for sharing your last night in London over a drink with Francisco and me. I hope someday I get to tell you por qué chingados quise un doctorado.

Thank you, Dr. Ask Hjorth Larsen. I only learned to *really* program after working with the codes you wrote. Thank you for our multiple conversations about complex scaled density functionals.

Thank you, Dr. Kiril Datchev, for all the conversations about the discrete spectrum of the Schrödinger operators. It still breaks my heart that our project never came to be.

I have much appreciation for the staff in the department. Cindy, thank you for being a mom away from home. Our conversations brought me so much comfort and literally kept me sane. Trisha, thank you for providing a nook in your office and your endless support in all the most stressful aspects of life.

Thank you, Ilse, Jon, Piña, and Renee. You folks are my family.

Thank you, Justin. We made it.

Thank you, mum and pop, for your unconditional support. Despite the scarcity, you found a way to see me succeed. I love you both.

Finally, I am very grateful to the Department of Chemistry at Purdue University for financial awards in the form of the Andrews Fellowship and the Charles H Viol Memorial Fellowship, to Prof. T. Daniel Crawford from The Molecular Science Software Institute, who supported me with the Seed Fellowship and the Investment Fellowship, and to Prof. Adam Wasserman who supported me through a Grant from the National Science Foundation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$\mathscr{H}$	Hamiltonian
$\mathscr{C}$	Arbitrary constant $\in \mathbb{R}$
E	total energy
ε	molecular orbital eigenvalue
ϕ_{i}	Kohn-Sham orbitals
M	number of nuclei
N	number of electrons
$\mathbf{r}$	position vector
$\mathbf{R}$	position vector of a nucleus
R_{e}	equilibrium internuclear separation
v_{ext}	external potential
v_{ks}	Kohn-Sham potential
v_{H}	Hartree potential
v_{Hxc}	Hartree-exchange correlation potential
v_{xc}	exchange-correlation potential
Z	nuclear charge

ABBREVIATIONS

2OA	Two-Orbital Approximation
DFA	Density Functional Approximation
CMS	Computational Molecular Sciences
DFA	Density Functional Approximation
DFT	Density Functional Theory
GGA	Generalized Gradient Approximation
HOMO	highest occupied molecular orbital
LDA	Local Density Approximation
LUMO	lowest unoccuiped molecular orbital
mRKS	The modified Ryabinkin, Kohut, and Staroverov method
OC	The Ou and Carter method
PBE	Purdue-Burke-Ernzerhof
PDE	partial differential equation
P-DFT	Partition Density Functional Theory
SCF	self-consistent error
WY	The Wu and Yang method
ZMP	The Zhao, Morrison, and Parr method

ABSTRACT

Partition Density Functional Theory (P-DFT) is a quantum chemistry method in which the system is fragmented into non-interacting components, and the energy is given by functionals of the fragment densities. The method is unique in the sense that it corrects for density functional approximation errors and sheds light on the individual structure of fragments within a molecule. In this work, we discuss the fundamental aspects of the theory as well as its challenges, and we introduce two software packages that were written to advance the understanding and applicability of the theory. The first, n2v focuses on the numerical procedure to obtain a potential that generates a given density, and the second, pyCADMium performs very accurate P-DFT calculations in diatomic molecules. Both packages are fully open-source and thus can be used and repurposed with any intention. We hope that these advances can be used to develop everyday embedding calculations.

1. INTRODUCTION

Partition Density Functional Theory (P-DFT) is a quantum chemistry method that relies on breaking down a full molecular calculation into smaller calculations on fragments. This approach does not only mean that we are able to achieve a linear scaling of N with respect to the number of fragments (acquiring a significant amount of time reduction as well as allowing fragment calculations to be run in parallel). The other intrinsic benefit is that many of the problems that are ubiquitous in Density Functional Theory, such as bond-breaking, can be easily be fixed. This work is organized as in the following manner: in Chapter 2 we review the theoretical foundations on Density Functional Theory. This chapter introduces all of the definitions that are relevant for P-DFT. In Chapter 3 we discuss how fragmenting a system into atoms or molecules can bring new benefits but also unexpected challenges. In Chapter 4 we introduce n2v, a new module that allows to perform different inversion methods that are of the essence for the understanding and development of P-DFT. In Chapter 5, we introduce pyCADMium , an open-source version of our P-DFT code for diatomic molecules. Finally, in Chapter 6, we go over the two-orbital approximation and the modified-Orbital approximation for P-DFT that allow us to approximate the non-additive kinetic energy functional in terms on the fragment densities.

2. DENSITY FUNCTIONAL THEORY

Consider an electronic system of N particles in the Born-Oppenheimer formalism [1]. If we consider atomic units (as we will be using throughout this text), the Hamiltonian in the position representation is

$$\mathscr{H}(\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N) = \underbrace{-\sum_{i=1}^{N} \frac{1}{2}\nabla^2_{\mathbf{r}_i}}_{\text{Kinetic Energy}} + \underbrace{\frac{1}{2}\sum_{i=1}^{N}\sum_{i\neq j}^{N} \frac{1}{|\,\mathbf{r}_i - \mathbf{r}_j\,|}}_{\text{Electron Repulsion}} - \underbrace{\sum_{i=1}^{N}\sum_{i=\alpha}^{M} \frac{Z_\alpha}{|\,\mathbf{r}_i - \mathbf{R}_\alpha\,|}}_{\text{Nuclear Attraction}} \tag{2.1}$$

where $\mathbf{r}_i$ refer to the 3N spatial coordinates of electrons and $\mathbf{R}_\alpha$ refer to the 3N spatial coordinates of the nuclei. Equation (2.1) in practice is expressed as a sum of operators independent of the representation used

$$\hat{\mathscr{H}} = \hat{T} + \hat{V}_{ee} + \hat{V}_{ext} \tag{2.2}$$

To find the electronic ground state of a system of interest we find the eigenvalues and eigenvectors of the previous operator. This is known as the Schrodinger Equation [2]. Using Dirac's notation [3], we can write it in a representation-independent formalism

$$\hat{\mathscr{H}}\,|\Psi\rangle = E\,|\Psi\rangle\,, \tag{2.3}$$

where the solution to Equation (2.3) is a wavefunction dependent on all 3N spatial coordinates and N spin coordinates. The solution is antisymmetric with respect to the exchange of an odd number of particle indices. The antisymmetry condition is forced upon the solution by constructing a linear combination of Slater determinants Φ [4],

$$\Phi(\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_N) = (N!)^{-1/2} \begin{vmatrix} \chi_i(\mathbf{x}_1) & \chi_j(\mathbf{x}_1) & \dots & \chi_k(\mathbf{x}_1) \\ \chi_i(\mathbf{x}_2) & \chi_j(\mathbf{x}_2) & \dots & \chi_k(\mathbf{x}_2) \\ \vdots & \vdots & & \vdots \\ \chi_i(\mathbf{x}_N) & \chi_j(\mathbf{x}_N) & \dots & \chi_k(\mathbf{x}_N) \end{vmatrix} \tag{2.4}$$

where each of the elements are spin orbitals $\chi(\mathbf{x})$, $\quad x := \{(\mathbf{r}_i, \sigma_i) \mid \mathbf{r}_i \in \mathbb{R},\ \ \sigma_i \in \{\uparrow, \downarrow\}$.

Finally, the electronic density can be found by integrating all but one coordinates in the wavefunction

$$n(\mathbf{r}) = N \int \cdots \int \mid \Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \mid^2 d\sigma d\mathbf{x}_2 \ldots d\mathbf{x}_N \tag{2.5}$$

Additionally, the electron density is normalized to the number of electrons

$$\int n(\mathbf{r}) \cdot d\mathbf{r} = N \tag{2.6}$$

To find the electronic ground-state energy we make use of the variational principle. It establishes that if exists, $|\Psi\rangle$ can be found as the function that minimizes the expectation value of the Hamiltonian, and in turn, the eigenvalue is the ground state energy E_0.

$$E_0 = \min_{\Psi} \langle \Psi | \hat{\mathscr{H}} | \Psi \rangle \tag{2.7}$$

2.1 1964 - Hohenberg & Kohn

Consider an isolated electronic system that is composed of N particles bound in space by a set of nuclei that fully constitutes an external potential $v_{\text{ext}}(\mathbf{r})$. We can express any atomic system we can think of by replacing the third term in the Hamiltonian of Equation (2.1). This is because the only term that provides an identity to the Hamiltonian is the external potential, i.e. $\mathscr{H}[v_{\text{ext}}(\mathbf{r})]$. In this way, by multiplying the Hamiltonian by a wavefunction on both sides and integrate in all of space, we are able to define a universal functional

$$F[n] = \langle \Psi[n] \big| \hat{T} + \hat{W}_{\text{ee}} \big| \Psi[n] \rangle \tag{2.8}$$

When we solve the Schrödinger equation, we find the eigenvalues and eigenvectors associated to a Hamiltonian, and from them, we build an electronic density. This implies that there is a path that takes us from the external potential to an electronic density

$$v_{\text{ext}}(\mathbf{r}) \underbrace{\longrightarrow}_{①} \Psi \underbrace{\longrightarrow}_{②} n(\mathbf{r}) \tag{2.9}$$

In the Fall of 1964, Pierre Hohenberg and Walter Kohn published an article that proved that not only $v_{\text{ext}}(\mathbf{r})$ maps to $n(\mathbf{r})$, but also that the universal functional from Equation (2.8) has as its minimum value the correct ground-state energy associated with the external potential $v_{\text{ext}}(\mathbf{r})$, making it possible to use the electronic density as a vehicle to find the properties of the system [5].

The one-page proof by contradiction is divided in two parts (① and ②), each part ensures that each mapping in (2.9) is unique:

- ① Consider two Hamiltonians, each defined by their external potential $\mathscr{H}_1 := \mathscr{H}[v_1(\mathbf{r})]$ and $\mathscr{H}_2 := \mathscr{H}[v_2(\mathbf{r})]$ so that they differ more than a constant $v_1(\mathbf{r}) - v_2(\mathbf{r}) \neq \mathscr{C}$. Let us assume that we can find the minimum energy ε_0 for each system with the same Ψ.

$$\underbrace{\mathscr{H}_1 |\Psi\rangle = \varepsilon_0 |\Psi\rangle \qquad \mathscr{H}_2 |\Psi\rangle = \varepsilon_0 \Psi(\mathbf{r})}_{\text{Subtract from each other}}$$

$$(v_1(\mathbf{r}) - v_2(\mathbf{r}))\,|\Psi\rangle = (\varepsilon_1 - \varepsilon_2)\,|\Psi\rangle$$

  The previous equality implies that $v_1(\mathbf{r}) - v_2(\mathbf{r}) \neq \varepsilon_1 - \varepsilon_2$ which is a contradiction to the initial statement since $\varepsilon \in \mathbb{R}$. We assume that the position representation of $|\Psi\rangle$ is a function that is non-zero for all spatial coordinates $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, ...)$, which is a reasonable assumption for isolated systems with an external potential originating from nuclei. This proves the uniqueness of the mapping between the external potential and the wavefunction.

- ② Assume that two different ground-state wavefunctions $|\Psi_1\rangle$ and $|\Psi_2\rangle$ are the lowest energy eigenfunctions of each Hamiltoinan $\mathscr{H}_1$ and $\mathscr{H}_2$, and they give rise to the same ground-state density $n(\mathbf{r})$. Next, we can build the following set of inequalities,

$$E_1 = \langle\Psi_1|\,\hat{\mathscr{H}}_1\,|\Psi_1\rangle < \langle\Psi_2|\,\hat{\mathscr{H}}_1\,|\Psi_2\rangle = \langle\Psi_2|\,\hat{\mathscr{H}}_2 + (v_1(\mathbf{r}) - v_2(\mathbf{r}))\,|\Psi_2\rangle = E_2 + \int (v_1(\mathbf{r}) - v_2(\mathbf{r}))n(\mathbf{r})d\mathbf{r}$$

$$E_2 = \langle\Psi_2|\,\hat{\mathscr{H}}_2\,|\Psi_2\rangle < \langle\Psi_1|\,\hat{\mathscr{H}}_2\,|\Psi_1\rangle = \langle\Psi_1|\,\hat{\mathscr{H}}_1 + (v_2(\mathbf{r}) - v_1(\mathbf{r}))\,|\Psi_1\rangle = E_1 + \int (v_2(\mathbf{r}) - v_1(\mathbf{r}))n(\mathbf{r})d\mathbf{r}$$

And proceed to add them together,

$$E_1 + E_2 < E_1 + E_2$$

Again, a contradiction. Therefore, two different wavefunctions cannot give rise to the same electronic density. This proves the uniqueness of the mapping between the wavefunction and the density. We arrive at the conclusion that two external potentials differing only by a constant, cannot generate the same non-degenerate ground state density.

$$n(\mathbf{r}) \longrightarrow v_{\text{ext}}(\mathbf{r}) + \mathscr{C}.$$

2.2 1965 - Kohn & Sham

Walter Kohn and Lu Jeu Sham proposed a way to rewrite the functional in Equation (2.8) hoping to circumvent the two-body operator $\hat{V}_{\text{ee}}$ [6]. To do so, the system of interacting particles is mapped into a system of non-interacting ones. To compensate for the missing interaction, the non-interacting particles are under the influence of an effective multiplicative potential that allows electrons to have the exact same density as the interacting system. With only one-body operators left, our problem becomes separable.

Mathematically, this is done by building a Hamiltonian as in Equation (2.2) in which the electron-electron term is scaled by a constant λ. By setting $\lambda \in [0, 1]$ we tune the electron interaction and connect the interacting system with the non-interacting system. Using Levy's formulation [7]

$$F_\lambda[n] = \left\langle \Psi_n^{min,\lambda} \middle| \hat{T} + \lambda \cdot \hat{V}_{\text{ee}} \middle| \Psi_n^{min,\lambda} \right\rangle \tag{2.10}$$

where $\Psi_n^{min,\lambda}$ is the single-determinant wavefunction that minimizes the expectation value of $\hat{T} + \lambda \hat{V}_{\text{ee}}$ as well as generating the density $n(\mathbf{r})$. The physical system is represented by $\lambda = 1$ and the auxiliary system of non-interacting electron is represented by $\lambda = 0$ so that:

$$F_0[n] = T_{\text{s}}[n] = -\frac{1}{2} \sum_{\text{i}=1}^{N} \langle \phi_{\text{i}} | \nabla^2 | \phi_{\text{i}} \rangle \tag{2.11}$$

where the $\{\phi(\mathbf{r})_i\}$ are the Kohn-Sham orbitals to be defined later. Additionally, the density is given by

$$n(\mathbf{r}) = \sum_{i=1}^{N} \mid \phi_i(\mathbf{r}) \mid^2 \tag{2.12}$$

The process of expressing our problem as a non-interacting one allows us to express the universal functional as

$$F[n] = T_s[n] + E_H[n] + E_{xc}[n] \tag{2.13}$$

where $T_s[n]$ is the kinetic energy computed using the known functional of Equation (2.11). E_H[n] is the Hartree energy functional

$$E_H[n] = \frac{1}{2} \int \int d\mathbf{r}\ d\mathbf{r}'\ \frac{n(\mathbf{r})\ n(\mathbf{r}')}{\mid \mathbf{r} - \mathbf{r}' \mid} \tag{2.14}$$

that represents the electrostatic repulsion energy generated by the electron density $n(\mathbf{r})$, and $E_{xc}[n(\mathbf{r})]$ is defined as the difference between the interacting components and the non-interacting components

$$E_{xc}[n] = \Big(T[n] - T_s[n]\Big) + \Big(V_{ee}[n] - V_H[n]\Big) \tag{2.15}$$

In summary, the full Kohn-Sham functional becomes

$$E[n] = T_s[n] + E_H[n] + E_{xc}[n] + \int d\mathbf{r}\ n(\mathbf{r})\ v(\mathbf{r}) \tag{2.16}$$

To minimize the previous expression, let us apply a functional derivative of the Kohn-Sham functional with respect to $n(\mathbf{r})$ to obtain

$$\frac{\delta T_s[n]}{\delta n(\mathbf{r})} + \int d\mathbf{r}' \frac{n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + \frac{\delta E_{xc}[n]}{\delta n(\mathbf{r})} + v_{ext}(\mathbf{r}) = 0 \tag{2.17}$$

where the second term is know as the Hartree potential:

$$v_H[n(\mathbf{r})] = \frac{\delta E_H[n]}{\delta n(\mathbf{r})} = \int d\mathbf{r}' \frac{n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \tag{2.18}$$

and the third term is known as the exchange-correlation potential:

$$v_{\mathrm{xc}}[n] = \frac{\delta E_{\mathrm{xc}}[n]}{\delta n(\mathbf{r})} \tag{2.19}$$

The electronic density used to evaluate the functional is obtained from the so called Kohn-Sham orbitals that are the solution to the set of differential equations known as the Kohn-Sham Equations.

$$\left[-\frac{1}{2}\nabla^2 + v_{\mathrm{ext}}(\mathbf{r}) + v_{\mathrm{H}}[n] + v_{\mathrm{xc}}[n] \right] \phi_{\mathrm{i}}(\mathbf{r}) = \varepsilon_{\mathrm{i}} \phi_{\mathrm{i}}(\mathbf{r}) \tag{2.20}$$

where $\phi(\mathbf{r})$ are the Kohn-Sham orbitals, ε_{i} are the orbital's energies. The Kohn-Sham equations are exact, in the sense that they recover the electronic density of the interacting system. $E_{\mathrm{xc}}[n(\mathbf{r})]$ is not known and must be approximated through density functional approximations (DFA). In practice, when it is said that Density Functional Theory does not yield accurate results, it is the DFA that fails, not DFT. Notice that $v_{\mathrm{H}}[n]$ and $v_{\mathrm{xc}}[n]$ are implicitly dependant on the orbitals, the set of functions that we are trying to find, and thus, this set of equation is solved self-consistently.

2.3 The Adiabatic Connection

Practical applications of DFT require Equation (2.15) to be approximated. The definition of the exchange-correlation energy ensures that all the physical effects missing from the non-interacting scheme are recovered. From a physical perspective, this includes effects from both the motion of electron as well as repulsion due to charge and spin. Associating many physical effects to a single variable that we seek to approximate is not very elegant, nor practical. Let us attempt to fix this issue. Remember that the Hellman-Feynman Theorem states that

$$\frac{\delta F_{\lambda[\mathbf{r}]}}{\delta \lambda} = \left\langle \Psi_n^{min,\lambda} \right| \hat{V}_{\mathrm{ee}} \left| \Psi_n^{min,\lambda} \right\rangle \tag{2.21}$$

where $F_\lambda[n]$ is the scaled universal function of Eq (2.10). Integrate the expression in the domain of $\lambda \in [0, 1]$ while maintaining the density constant:

$$\int_0^1 \frac{\delta F_{\lambda[\mathbf{r}]}}{\delta \lambda} d\lambda = F_1[n] - F_0[n] = E_{\mathrm{xc}}[n] + E_{\mathrm{H}}[n] \tag{2.22}$$

Combine Equation (2.21) with (2.22) to find the adiabatic connection formula

$$E_{\mathrm{xc}}[n] = \int_0^1 \left\langle \Psi_n^{min,\lambda} \right| \hat{V}_{\mathrm{ee}} \left| \Psi_n^{min,\lambda} \right\rangle d\lambda - E_{\mathrm{H}}[n] \tag{2.23}$$

We have removed any kinetic dependence from our exchange-correlation expression. The drawback is that we now require to know how this functional behaves as λ changes from zero to one. The adiabatic connection formula is a common starting point for many density functional approximations [8].

In practice the exchange-correlation energy is separated into two parts:

$$E_{\mathrm{xc}}[n] = E_{\mathrm{x}}[n] + E_{\mathrm{c}}[n] \tag{2.24}$$

so that each can be approximated individually.

2.4 Ingredients to Construct Density Functional Approximations

To construct an approximation for $E_{\mathrm{xc}}[n]$, the only real guideline is to have a functional that depends on the density. The general form is given by

$$E_{\mathrm{xc}}[n] = \int \varepsilon_{\mathrm{xc}}(n) d\mathbf{r} \tag{2.25}$$

where $\varepsilon_{\mathrm{xc}}$ is a quantity called energy density given its dimensions $\frac{energy}{volume}$.

The Local Density Approximation (LDA) is a simple yet powerful functional that solely depends on the density [6]. Unfortunately for most practical calculations, it is easy to find that the dependence of $E_{\mathrm{xc}}[n]$ on the density is often highly non-local [9], making small changes on the density to cause large changes on the functional derivative of $E_{\mathrm{xc}}[n]$. To compensate, other ingredients need to be added to account for this non-locality. The most

common ingredients after the density are the gradient of the density ($|\nabla n(\mathbf{r})|$) and laplacian of the density ($\nabla^2 n(\mathbf{r})$), and the kinetic energy density

$$\tau = \frac{1}{2}\sum_{\mathrm{i}}^{N} |\nabla\phi_{\mathrm{i}}|^2 \tag{2.26}$$

that integrates to the Kohn-Sham kinetic energy $T_{\mathrm{s}}[n]$. By doing so, we can then construct an energy density with any arbitrary number of different ingredients

$$\varepsilon_{\mathrm{xc}} := \varepsilon_{\mathrm{xc}}(n, \nabla n, \nabla^2 n, \tau, ...) \tag{2.27}$$

In theory, as we increase the number of ingredients, we expect a more accurate functional. The modern development of functionals depends on the conventions introduced by John Perdew. According to his methodology, by ensuring that our approximations are in agreement with known exact conditions for $E_{\mathrm{xc}}[n]$, the closer we move towards the true exact functional [10].

3. WHAT IS THE SHAPE OF ATOMS IN MOLECULES?

“What is the shape of atoms in molecules?" is a question without an answer. Atoms in molecules do not have a real, independent existence. When we say that a water molecule “has" two hydrogen atoms and one oxygen atom, what we mean is that an appropriate supply of energy can split the molecule into these three atoms. Each of these atoms, when isolated, is spherical, but the ground-state electronic density of the water molecule is not simply the sum of three spheres. However, it is approximately equal to the sum of three spheres (see Figure 3.1), and can be written exactly as the sum of three distorted spheres. Thus, the question of the title may be vague but it is not meaningless. All molecular information (i.e. what makes a molecule different from the simple sum of its isolated constituents) is coded into a set of atomic density distortions.

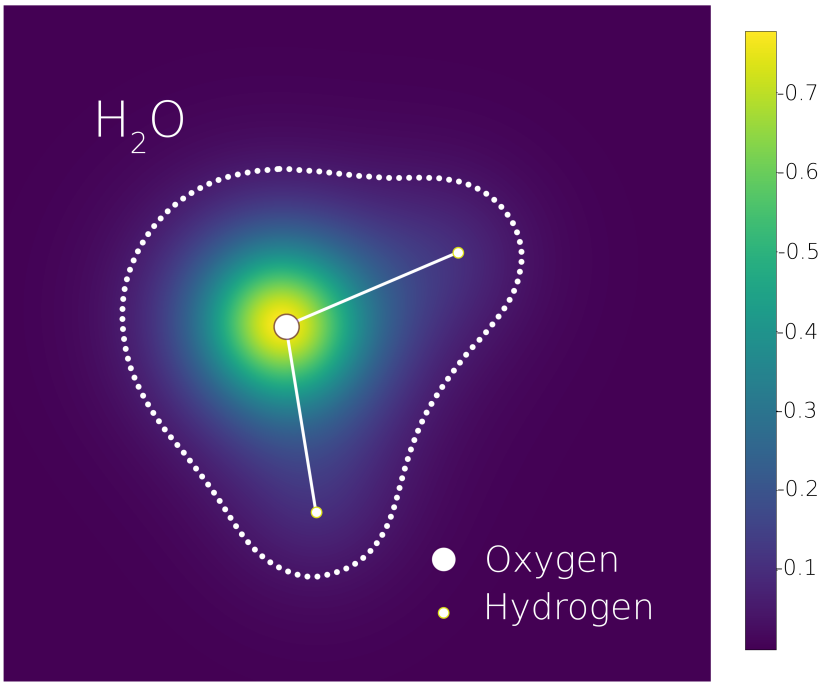


Figure 3.1. Ground-state density of a water molecule on the plane of the three atoms. Calculated using Psi4 [11], [12] with CCSD(T)/UGBS. The dotted line indicates an iso-density contour.

There is no unique way of defining these atomic density distortions. In fact, there are infinitely many ways in which one can decompose a given molecular density $n(\mathbf{r})$ into the sum of atomic-like functions $n_\alpha(\mathbf{r})$ (we use the subscript α to label atoms in this chapter). However, there is a unique set of densities $\{n_\alpha(\mathbf{r})\}$ that sum up to $n(\mathbf{r})$ while minimizing the sum of the atomic energies. It is this special set of densities to which we turn our attention in this chapter.

Density Functional Theory (DFT) [5], [6] establishes that any electronic property P of a molecule is a functional of its ground-state electronic density, $P = P[n]$. The uniqueness of the set $\{n_\alpha(\mathbf{r})\}$ for a given density then allows one to understand molecular properties, in principle, as functionals $P[\{n_\alpha(\mathbf{r})\}]$ of that set. Decades of research in DFT have taught us how the total density $n(\mathbf{r})$ can be used as the main variable in molecular calculations, as explained briefly in Section 3.1. Our group is investigating how the atomic densities, as opposed to the total molecular density, may be used as the main variables, and we discuss this in Section 3.2. The change of perspective brings with it advantages and disadvantages. The most obvious advantage is a significant lowering of the computational cost of the calculations because instead of having to solve the N-electron Schrödinger equation (a second-order differential equation on $3N$ coupled variables), one solves only for a small number of independent equations, each for less than N electrons. A second advantage will be explained and illustrated in Section 3.3: By focusing on atomic densities, rather than on total molecular densities, one can fix pervasive errors of approximate density-functional approximations and significantly improve the accuracy of certain calculations; a third advantage will be discussed in Section 3.4: The chemical reactivity between two atoms or molecular fragments that approach each other is best described in a theory that employs atomic (or fragment) densities as the main variables. On the downside, many of the theorems and techniques that have been explored over the last six decades to describe molecular systems are not directly applicable to the new set of variables, so entirely new methods and approximations need to be developed. A growing community of researchers working under the umbrella of “embedding" methods [13]–[16], including the authors, are pursuing this direction. We highlight here some of the recent developments.

3.1 The Struggles of Using the Molecular Density

An ample literature exists documenting the successes and failures of different approximations to $E[n]$, [17]–[19], but in a nutshell, some of the most pervasive failures of approximate KS-DFT appear when stretching chemical bonds. The large errors observed in these cases encompass both fractional charge (or delocalization) and fractional-spin (or static-correlation) errors [20], [21], which are ultimately due to the inability of the approximate XC-functionals to reduce the molecular density to the correct atomic densities (or spin-densities) when bonds are stretched. We illustrate both types of errors below for the Local Density Approximation (LDA), the simplest and earliest approximation for $E[n]$ on which the modern ladder of approximations is built [22].

Fractional-charge error: Consider stretching H_2^+. The true ground-state density has left-right symmetry, with $\frac{1}{2}$ electron on the left atom and $\frac{1}{2}$ electron on the right atom. The physical state at infinite separation must break this symmetry and produce a neutral hydrogen atom on one side and a bare proton on the other. Both solutions (broken-symmetry and symmetric) should therefore have the same energy at infinite separation, but this feature is not achieved by the LDA or other approximations built upon it. The LDA energy of a H atom with half an electron is much lower than what it should (half the energy of infinitely stretched H_2^+), leading to the incorrect binding shown in dashed pink in Figure 3.2.

Fractional-spin error: Now consider stretching H_2. The issues of the previous paragraph are no longer a problem because each atom has exactly one electron at infinite separation. However, an analogous problem arises for fractional spins. The true ground state of H_2 is spin-unpolarized and must remain so at very large separations in the absence of environmental perturbations. However, an isolated H atom is spin-polarized so the energy of two spin-polarized H atoms should be identical to that of two spin-unpolarized H atoms (each having half-spin up, half-spin down). This condition is again violated by the LDA, leading to the significant overestimation of the binding that can be appreciated in Figure 3.3.

These errors, illustrated here for the two simplest open-shell and closed-shell molecules, are ubiquitous in quantum chemistry. Every time a bond is stretched, as in transition states along chemical reactions, a combination of these errors can creep into and contaminate the

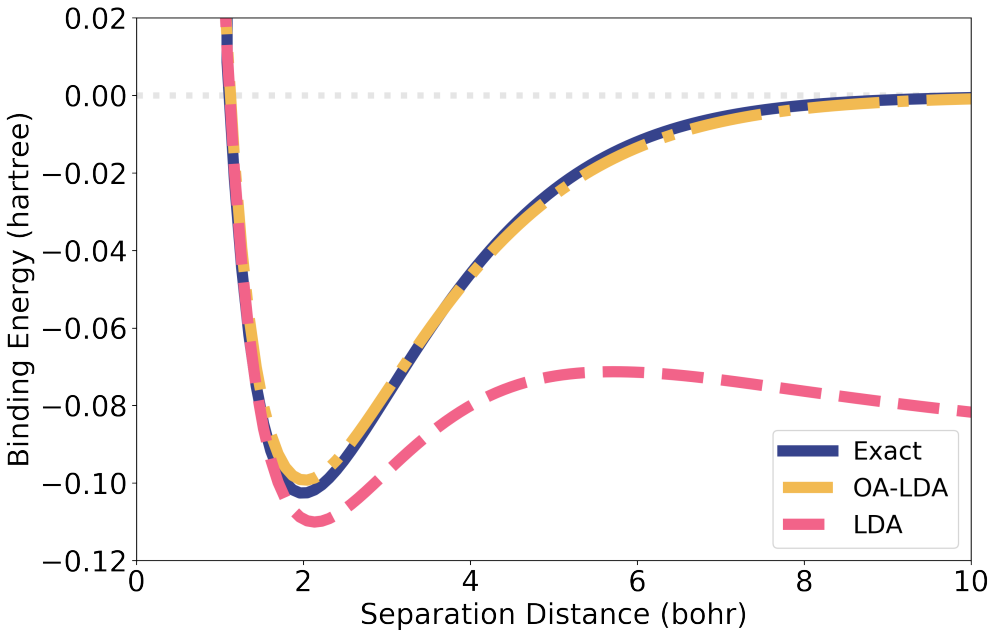


Figure 3.2. Electronic binding energy of H_2^+. Exact (solid, blue), KS-LDA (dashed, pink), and OA-LDA (dash-dotted, yellow) from [23], as explained in Section 3.3

DFT calculations. Cancellation between the two errors can sometimes occur (note they have opposite signs) and lead to accidentally accurate results for complex systems, but predicting such cancellations is generally extremely difficult and not something DFT users want to or should rely on. Results from approximate KS-DFT calculations are thus often suspect. A theory that uses atomic densities as the main variables, as opposed to the total molecular density, has the potential the fix such errors. Furthermore, it has the potential to provide information about individual atomic density distortions, along with chemical insight into the reactivity of individual fragments.

3.2 Atomic Densities as the Main Variable

As mentioned in the Introduction if this chapter, there is a unique set of atomic densities $\{n_\alpha(\mathbf{r})\}$ that minimizes $E_f \equiv \sum_\alpha E_\alpha[n_\alpha(\mathbf{r})]$ while satisfying the density constraint:

$$\sum_\alpha n_\alpha(\mathbf{r}) = n(\mathbf{r}) \tag{3.1}$$

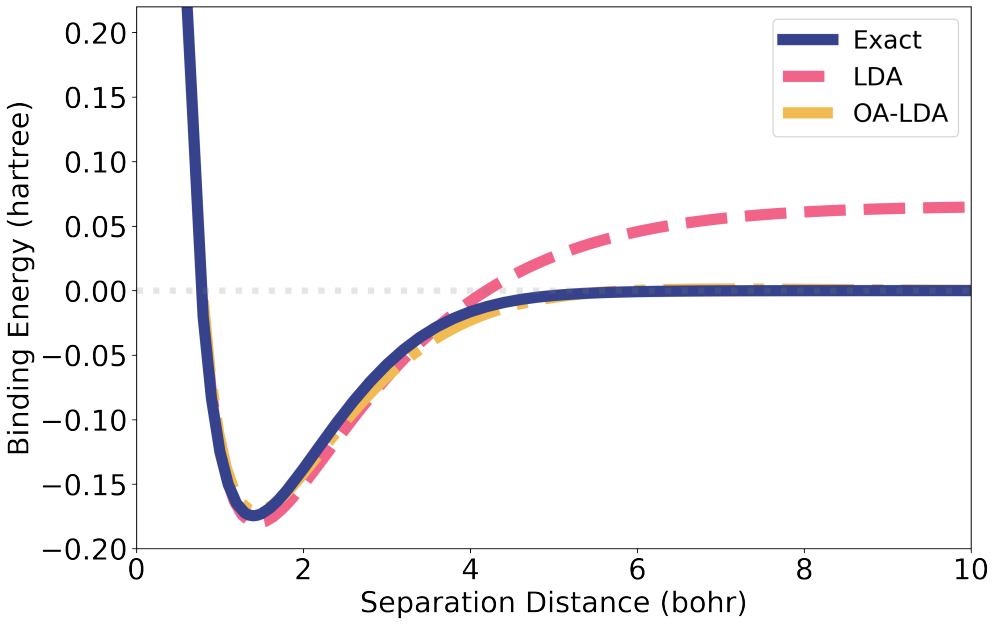


Figure 3.3. Electronic binding energy of H_2. Exact (solid, blue), KS-LDA (dashed, pink), and OA-LDA (dash-dotted, yellow). from [19] as explained in Section 3.3

The atomic energies $E n_\alpha(\mathbf{r})]$ in the above definition of E_f are not true ground-state energies, but rather given by $E_\alpha[n_\alpha(\mathbf{r})] = F[n_\alpha(\mathbf{r})] + \int d\mathbf{r} v_\alpha(\mathbf{r}) n_\alpha(\mathbf{r})$, where $v_\alpha(\mathbf{r})$ is the α-atomic potential. The constraint of Equation (3.1) prevents $n_\alpha(\mathbf{r})$ from being the ground-state density of the corresponding $v_\alpha(\mathbf{r})$. However, each of the n_α can be shown to be the ensemble ground-state density of $v_\alpha(\mathbf{r})+v_p(\mathbf{r})$, where $v_p(\mathbf{r})$ is a unique α-independent potential. More specifically, the Partition Potential Theorem (PPT) [24] establishes the following: If a molecular potential $v(\mathbf{r})$ is decomposed into atomic potentials $\{v_\alpha(\mathbf{r})\}$, i.e. $v(\mathbf{r}) = \sum_\alpha v_\alpha(\mathbf{r})$, then, for a set of fragment occupations $\{N_\alpha\}$ there is a unique local potential $v_p(\mathbf{r})$ such that, when added to the individual $v_\alpha(\mathbf{r})$'s leads to ensemble ground-state densities $\{n_\alpha(\mathbf{r})\}$ summing up to the correct total density $n(\mathbf{r})$.

Simple Illustration of the PPT: Figure 3.4 provides the simplest illustration of this theorem. Consider first one electron moving in the 1D-potential $v_1(x)$ (dotted grey line in middle panel). Its ground-state density, when isolated, is $n_1^{(0)}(x)$ (dotted orange line in upper pannel). Similarly, one electron in $v_2(x)$ has density $n_2^{(0)}(x)$. Now consider two non-interacting electrons in the double-well potential $v(x) = v_1(x)+v_2(x)$ (dashed yellow line in middle panel).

The density of this system, $n(x)$, is not equal to $n^{(0)}(x) = n_1^{(0)}(x) + n_2^{(0)}(x)$, but it is close to it: $n(x) \approx n^{(0)}(x)$, especially if $v_1(x)$ overlaps weakly with $v_2(x)$. The PPT establishes that there is only one potential $v_p(x)$ (purple line in Figure 3.4) such that, when added separately to $v_1(x)$ (lower panel) and $v_2(x)$ leads to ground-state densities $n_1(x)$ (upper pannel) and $n_2(x)$ that differ from $n_1^{(0)}(x)$ and $n_2^{(0)}(x)$ in just the right way so that $n_1(x) + n_2(x) = n(x)$. The theorem applies to any number of interacting electrons in 3D and to any number of fragments [24].

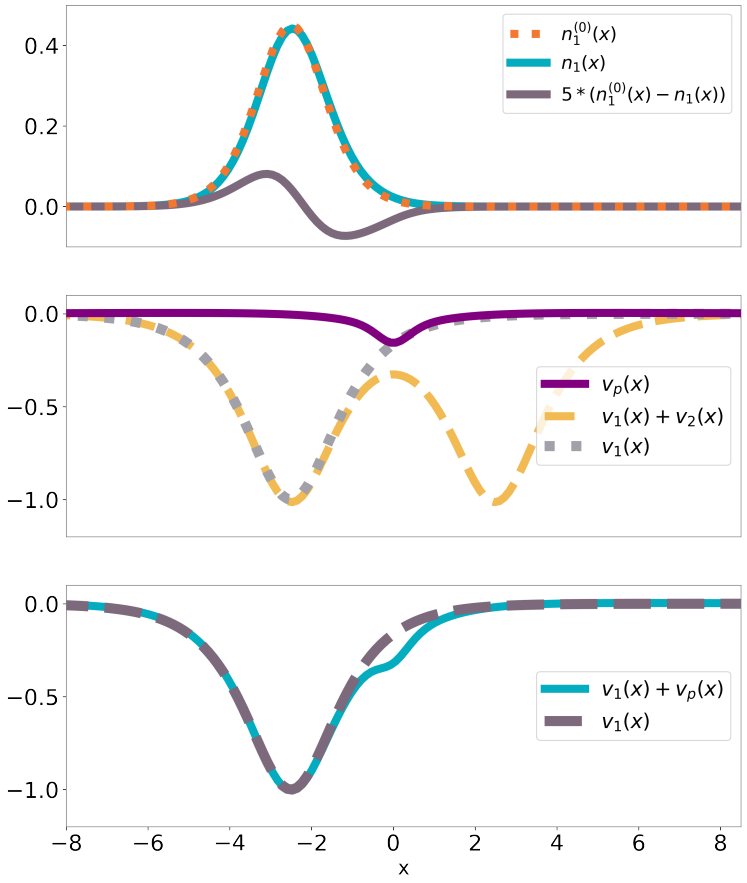


Figure 3.4. Graphic example of simplest PPT case for 2 non-interacting electrons in 1D potentials of the form $v_{1,2}(x) = -\cosh^{-2}(x \pm a)$, with $a = 2.5$. Upper panel: The gray line magnifies the atomic-density distortion $n_1(x) - n_1(x)^{(0)}$ by a factor of 5 to highlight what occurs upon formation of the chemical bond: The density of the isolated atom is pulled toward the bonding region. Middle panel: Left atomic potential $v_1(x)$ (dotted), total 'molecular' potential (dashed), and partition potential (solid, purple). Bottom panel: Comparison of the isolated atomic potential $v_1(x)$ and the effective potential $v_1(x) + v_p(x)$ for which the polarized density $n_1(x)$ is a ground-state density.

Algorithm to calculate $v_p(\mathbf{r})$: Several algorithms have been developed to solve the constrained optimization problem involved in calculating $v_p(\mathbf{r})$ (i.e. minimizing $E_f[\{n_\alpha(\mathbf{r})\}]$ under the constraint of Eqution (3.1)). The algorithm described here is perhaps conceptually the simplest:

1. Choose an approximation for $E[n]$, solve the KS equations of Equation(2.20) for the isolated atoms, and find their self-consistent densities $\{n_\alpha^{(0)}(\mathbf{r})\}$ and corresponding KS-potentials $\{v_{s,\alpha}^{(0)}(\mathbf{r})\}$.

2. Build an approximate molecular density as $n^{(0)}(\mathbf{r}) \approx \sum_\alpha n_\alpha(\mathbf{r})^{(0)}(\mathbf{r})$.

3. Invert the KS equations to find the effective KS-potential $v_s^{(0)}(\mathbf{r})$ corresponding to $n^{(0)}(\mathbf{r})$. For the exact density, this potential would be identical to $v(\mathbf{r}) + v_{\text{Hxc}}(\mathbf{r})$, but for the approximate density it is not. The difference between the two, therefore, can be used as a correction to generate an improved atomic KS potential $v_{s,\alpha}^{(1)}(\mathbf{r}) = v_{s,\alpha}^{(0)}(\mathbf{r}) + \{v(\mathbf{r}) + v_{\text{Hxc}}[n^{(0)}](\mathbf{r}) - v_s^{(0)}(\mathbf{r})\}$.

4. Solve the KS equations for the atoms again with the improved atomic KS potential and repeat until self-consistency is achieved. If convergence is achieved after iterating k times, then the atomic KS potentials $v_{s,\alpha}^{(k)}(\mathbf{r})$ are given by $v_\alpha(\mathbf{r}) + v_p(\mathbf{r}) + v_{\text{Hxc}}[n_\alpha(\mathbf{r})^{(k-1)}](\mathbf{r})$. That is, the partition potential emerges as an α-independent piece to be added to the α-nuclear potential.

Just as for the model system of Figure 3.4, the main feature of $v_p(\mathbf{r})$ in real diatomic molecules is an attractive well in between the nuclei. The 'job' of this well is to distort the density of each isolated atom by pulling it toward the bonding region. There are generally also positive plateaus in $v_p(\mathbf{r})$ that are due to kinetic effects [15] (contributions from the functional derivatives of $T_s[n_\alpha(\mathbf{r})]$), and a singularity at the nuclei whose strength is proportional to the value of the density of one atom at the location of the other [15]. Modifying the isolated-atom density of the nuclei according to Kato's cusp condition. This is our solution to the question without an answer: The shape of atoms in molecules is determined by $v_p(\mathbf{r})$.

Now define the partition energy E_p as the difference between the total energy E and the sum of atomic energies E_f. Using a bold $\mathbf{n}(\mathbf{r})$ to denote the set of atomic densities, $\mathbf{n}(\mathbf{r}) \equiv \{n_\alpha(\mathbf{r})\}$,

$$E_p[\mathbf{n}] = E[n] - E_f[\mathbf{n}] \tag{3.2}$$

With the KS energy-decomposition of Equation (2.16), the partition energy is given by:

$$E_p[\mathbf{n}] = T_s^{\text{nad}}[\mathbf{n}] + E_{\text{Hxc}}^{\text{nad}}[\mathbf{n}] + V_{\text{ext}}^{\text{nad}}[\mathbf{n}] \quad , \tag{3.3}$$

where the "nad" superscript is used to indicate non-additive quantities, i.e. the difference between the total and the sum of the fragments. Thus

$$T_s^{\text{nad}}[\mathbf{n}] = T_s[\mathbf{n}] - \sum_\alpha T_s[n_\alpha(\mathbf{r})] \tag{3.4}$$

$$E_{\text{Hxc}}^{\text{nad}}[\mathbf{n}] = E_{\text{Hxc}}[\mathbf{n}] - \sum_\alpha E_{\text{Hxc}}[n_\alpha(\mathbf{r})] \tag{3.5}$$

$$V_{\text{ext}}^{\text{nad}} = \int d\mathbf{r} \cdot n(\mathbf{r}) \cdot v(\mathbf{r}) - \sum_\alpha \int d\mathbf{r} \cdot n_\alpha(\mathbf{r}) v_\alpha(\mathbf{r}) \tag{3.6}$$

We have again combined the non-additive exchange-correlation and Hartree terms of Equation (2.16) together into $E_{\text{Hxc}}^{\text{nad}}$. For a given approximation to $E_{\text{xc}}[n]$, the algorithm described in the previous section exactly reproduces the results of the corresponding KS calculation, including all of its errors. It can be shown [25] that the partition potential is the functional derivative of the partition energy with respect to any of the atomic densities, at the minimum. Since $V_{\text{ext}}^{\text{nad}}$ is known exactly, the key to improving over Kohn-Sham is to propose adequate approximations for the other two components of Equation (3.3), T_s^{nad} and $E_{\text{Hxc}}^{\text{nad}}$. We discuss each separately:

3.3.1 Approximating $E_{\rm Hxc}^{\rm nad}$

The origin of common errors of approximate XC functionals is well understood especially for homonuclear diatomic molecules [19] (i.e. see discussion above for H_2 and H_2^+). We have shown how a simple overlap approximation (OA) to $E_{\rm Hxc}^{\rm nad}$ can fix both of these errors simultaneously [26], something that no approximate XC functional can achieve with comparable accuracy within KS-DFT. The OA is defined by:

$$E_{\rm Hxc}^{\rm nad,\ OA}[\mathbf{n}] = E_{\rm H}^{\rm nad}[\mathbf{n}] + S[\mathbf{n}]E_{\rm xc}^{\rm nad}[\mathbf{n}] + (1 - S[\mathbf{n}])\Delta E_{\rm H}^{\rm nad}[\mathbf{n}] \quad , \tag{3.7}$$

where $S[\mathbf{n}]$ is a measure of the overlap between the two atomic densities:

$$S[\mathbf{n}] = \mathrm{erf}\left(2\int d\mathbf{r}\,\sqrt{n_A(\mathbf{r})n_B(\mathbf{r})}\right) \tag{3.8}$$

The term $\Delta E_{\rm H}^{\rm nad}[\mathbf{n}]$ in Equation (3.7) is a correction to the non-additive Hartree contribution, properly defined in ref.[26] so that both fractional-charge and fractional-spin errors are suppressed as the molecule is stretched, as shown in ref.[26]. The results shown in yellow in Figure 3.2 and labeled "OA-LDA" go a step further [23] and replace $E_{\rm Hxc}^{\rm nad}$ in Equation (3.7) by the non-additive exact-exchange functional, canceling completely the self-interaction error and leading to the exact $E_p[\mathbf{n}]$ in this case [23]. The only deviation from the exact binding here is due to the effect that the slightly incorrect LDA fragment densities have on E_f. We stress that the correct energy is obtained here as the molecule is stretched without symmetry breaking: The ground state of H_2^+ retains left-right symmetry, and the ground-state of H_2 remains a spin-singlet throughout the entire range of separations. This is a proof-of-principle demonstration that it is possible to use a simple functional of the density for the atoms (not for the molecule), while approximating $E_{\rm Hxc}^{\rm nad}$ to fix the underlying errors due to fragmentation. The route is complementary to the efforts of many others to develop sophisticated XC-functionals of the total density [27]–[31].

3.3.2 Approximating T_s^{nad}

Even with a robust and accurate functional for $E_{\text{Hxc}}^{\text{nad}}[\mathbf{n}]$ (based perhaps on future generalizations of the OA), an explicit approximation for $T_s^{\text{nad}}[\mathbf{n}]$ is needed if one wants to reach the goal of linear-scaling calculations. Step 3 of the algorithm described in Section 3.2 relies on iterative inversions that make the method computationally impractical (at least, not more efficient than regular KS-DFT). The most obvious way to avoid such inversions is by resorting to orbital-free DFT (OF-DFT) [32] to approximate $T_s^{\text{nad}}[\mathbf{n}]$ as an explicit functional of the set of atomic densities. Kinetic-energy functionals are famously difficult to approximate and state-of-the-art functionals are still far from reaching chemical accuracy, but we are looking for approximations to the non-additive part of T_s, which is altogether a different challenge. The non-interacting kinetic energies for the atoms are still calculated exactly via orbitals, and we wish to approximate the much smaller $T_s^{\text{nad}}[\mathbf{n}]$. Recent work shows that the route is promising. For example, writing $T_s^{\text{nad}} = \int d\mathbf{r} n(\mathbf{r}) t_s^{\text{nad}}(\mathbf{r})$, an expression for the non-additive kinetic energy density $t_s^{\text{nad}}(\mathbf{r})$ of the type:

$$t_s^{\text{nad}}(\mathbf{r}) = Q[\mathbf{n}] t_s^{\text{nad, vW}}(\mathbf{r}) + (1 - Q[\mathbf{n}]) t_s^{\text{nad,TF}}(\mathbf{r}) \tag{3.9}$$

has been shown [33] to provide an excellent approximation to the exact T_s^{nad} in covalent σ-bonds. In Equation (3.9), $t_s^{\text{nad, vW}}$ is the von Weiszäcker kinetic-energy density [34], which is exact for one-orbital systems, $t_s^{\text{nad, TF}}$ is the Thomas-Fermi kinetic-energy density [35], [36], exact for uniform-density systems, and $Q[\mathbf{n}]$ is a switching functional whose role is analogous to that of $S[\mathbf{n}]$ in Equation (3.7), as described in detail in Ref.[33]. It determines the spatial regions where a one-orbital description should dominate.

3.4 Towards a Quantum Theory of Chemical Reactivity

We glossed over one key feature of the Partition Potential Theorem (PPT) of Section 3.2 The minimization of $E_f[\mathbf{n}]$ is performed under the constraint that the $n_\alpha(\mathbf{r})$ add to $n(\mathbf{r})$, without individual normalization constraints for the atomic densities. The $n_\alpha(\mathbf{r})$ can integrate to fractional numbers of electrons N_α as long as the total density integrates to the correct

number of electrons, i.e. $\sum N_\alpha = N$. A sensible interpretation of fractional-number densities is provided by an ensemble description where the fractional number arises as an ensemble average over integer-number components. One result of the extension of DFT for fractional electron numbers [37] is that for N_α between the integers p_α and $p_{\alpha+1}$, the minimizing density (for the exact $E_{\rm xc}[n]$) is given by $n_\alpha(\mathbf{r}) = (1-\omega_\alpha)n_{p_\alpha}(\mathbf{r}) + \omega_\alpha n_{p_{\alpha+1}}(\mathbf{r})$, where $0 \le \omega_\alpha \le 1$. The minimization of $E_f[\mathbf{n}]$ is to be performed over the set of the $\{\omega_\alpha\}$, leading to possibly fractional-number densities. One immediate advantage of such fractional densities is that chemical reactivity indices involving derivatives of various properties with respect to electron number become sharply defined for the atoms. For example, Fukui functions are given directly by

$$f_\alpha^+(\mathbf{r}) = \left.\frac{\partial n_\alpha(\mathbf{r})}{\partial N_\alpha^+}\right|_{v_\alpha} = n_{p_\alpha+1}(\mathbf{r}) - n_{p_\alpha}(\mathbf{r}) \tag{3.10}$$

as normally defined [38], but here the bordering-integer densities include the polarizing effect of the partition potential which accounts for the detailed environment of the atom in the molecule. Similarly, the $\{\omega_\alpha\}$ are those that lead to electronegativity equalization [24], [39].

3.5 Next Steps

The approximations of Equations (3.7) and (3.9) need to be extended to be applicable to realistic, complex chemical systems. The roads to robust approximations of general applicability, and to efficient and accurate linear-scaling algorithms will be long and winding. Therefore, one might wonder whether these roads are worth taking in the first place, especially given that: (a) As computers get more powerful and machine-learning conquers the quantum-chemistry landscape, KS-DFT calculations with sophisticated approximations to $E_{\rm xc}$ will become applicable to an ever-expanding frontier of chemical complexity; (b) Several other fragment-based [40] and embedding methods [41]–[43] are enabling multi-million atom and multi-scale calculations where individual fragment densities are of little use; and (c) Quantum computing, when finally here, will allow for the direct calculation of many-electron wavefunctions rendering DFT-based methods obsolete.

Our current take on these three valid concerns is the following: (a) Further improvements of approximate XC functionals will continue via two directions: A non-empirical approach in which more exact constraints will be incorporated, especially perhaps in the framework of Generalized-Kohn-Sham [44], [45], and an empirical approach exploiting large data sets of chemical information through machine-learning tools [46], [47]. However, all of these positive developments can be readily incorporated into the framework described in Section 3.2. Furthermore, the fragments do not need to be atoms but could be functional groups, protein backbones, etc., and calculations will greatly benefit from improved approximations to E_{xc}; (b) There is plenty of room in quantum chemistry for more than one type of embedding method. When minimizing the total energy is the only goal, our approach is admittedly not essential. When, however, in addition to minimizing the energy, one is interested in examining the individual fragment density distortions or in understanding the reactivity of one fragment in a specific chemical environment, then our approach offers a unique, useful perspective; (c) Yes: Some day quantum computers will be ready to solve the many-electron Schrödinger equation for large molecules. However, one will always want to understand the results. Understanding the results involves determining how individual atoms or fragments in the molecule are distorted due to the interactions with neighboring atoms or fragments. The tools described here allow us to accomplish precisely this, regardless of the type of computers employed.

4. THE INVERSE KOHN-SHAM PROBLEM

4.1 The Kohn-Sham Inversion Formula.

Assume you've successfully completed an SCF calculation using a DFA of your choice. Given the Kohn-Sham orbitals along with its eigenvalues, could we recover the Kohn-Sham potential by solving for v_{KS}?:

$$v_{\mathrm{KS}}(\mathbf{r}) = \frac{1}{2} \cdot \frac{\nabla^2 \phi_{\mathrm{i}}(\mathbf{r})}{\phi_{\mathrm{i}}(\mathbf{r})} + \epsilon \tag{4.1}$$

Notice that we obtain the v_{KS} using a single orbital, while the Kohn-Sham equations result in a set of N Kohn-Sham orbitals. Thus, this definition results in a different potential for a different orbital.

Another alternative is consider using all of the orbitals to build a weighted average of orbital-specific potentials, where each of the weights is $\mid \phi_{\mathrm{i}}(\mathbf{r}) \mid^2 n(\mathbf{r})$

$$v_{\mathrm{KS}}(\mathbf{r}) = \frac{1}{n(\mathbf{r})} \sum_{\mathrm{i}=1}^{N} \left[\frac{1}{2} \phi_{\mathrm{i}}^{*}(\mathbf{r}) \nabla^2 \phi_{\mathrm{i}}(\mathbf{r}) + \epsilon \mid \phi_{\mathrm{i}}(\mathbf{r}) \mid^2 \right] \tag{4.2}$$

In the following, any function that carries the subscript "in" (as in $n_{\mathrm{in}}(\mathbf{r})$ below) represents a quantity that enters iKS as input.

This direct method is rarely useful in practice because it requires KS orbitals that are already the result of a KS calculation. Nevertheless, Equation (4.1) has been used for determining exchange-correlation potentials [48], [49], where the direct $v(\mathbf{r})$ is less obvious to obtain directly on a grid for a high level XC functional. It has also been used for determining the functional derivative of a kinetic energy functional [50], and for constructing orbital-averaged exchange-correlation potentials for orbital-dependent functionals [51].

Unfortunately, the procedure indicated by Equation (4.1) suffers from numerical difficulties when the $\psi_{\mathrm{i}}(\mathbf{r})$'s are expanded on finite basis-sets[52]. These numerical issues are almost exclusively dependent on the basis-sets employed. Gaiduk et al.[53] showed how one could build an oscillation profile from a simple method such as the Slater exchange functional. The profile can then be used to subtract the errors from any other more sophis-

ticated density-functional approximations. This method along with its basis-set correction is the first method available in n2v.

4.1.1 The Zhao-Morrison-Parr method

One of the earlier iKS methods is the Zhao-Morrison-Parr method (ZMP), named after their developers Zhao, Morrison and Parr [54] in the early 1990's, and recently generalized by Kumar and Harbola [55]. The method uses a self-consistent calculation to determine the one-electron effective potential from the target density $n(\mathbf{r})$. To do so, a constraint enforces a density to be equal to $n_{\text{in}}(\mathbf{r})$ at a certain limit:

$$C[n(\mathbf{r}), n_{\text{in}}(\mathbf{r})] = \frac{1}{2}\int\int \frac{\{n(\mathbf{r}) - n_{\text{in}}(\mathbf{r})\}\{n(\mathbf{r}') - n_{\text{in}}(\mathbf{r}')\}}{|\,\mathbf{r} - \mathbf{r}'\,|} \cdot d\mathbf{r} \cdot d\mathbf{r}' = 0 \tag{4.3}$$

By attaching a Lagrange multiplier λ, one finds the corresponding potential

$$v_c^{\lambda}(\mathbf{r}) = \lambda \int \frac{n(\mathbf{r}') - n_{\text{in}}(\mathbf{r}')}{|\,\mathbf{r} - \mathbf{r}'\,|} d\mathbf{r}' \tag{4.4}$$

So that we can solve the differential equations

$$\left[-\frac{1}{2}\nabla^2 + v_{\text{ext}}(\mathbf{r}) + (1 - \frac{1}{N})v_{\text{H}}(\mathbf{r}) + v_c^{\lambda}(\mathbf{r}) \right] \psi_{\text{i}}^{\lambda}(\mathbf{r}) = \varepsilon_{\text{i}}^{\lambda}\psi_{\text{i}}^{\lambda}(\mathbf{r}) \tag{4.5}$$

where $\psi_{\text{i}}^{\lambda}$ and $\varepsilon_{\text{i}}^{\lambda}$ are the eigenfunctions and eigenvalues found with v_c^{λ} with a particular λ, and the term $(1-\frac{1}{N})v_{\text{H}}(\mathbf{r})$ is the Fermi-Amaldi approximation. This term, besides being exact for 1 electron [56], ensures the correct $-1/r$ asymptotic behavior of the resulting potential.

In the limit $\lambda \to \infty$ Equation (4.4) becomes Equation (2.20) and we can retrieve the Kohn-Sham orbitals and orbital energies up to a constant. For finite λ, the solution will only be an approximation to the exact orbitals.

With a choice of λ, at convergence, the resulting $v_{\text{xc}}(\mathbf{r})$ is given by:

$$v^{\lambda}(\mathbf{r}) = v_c^{\lambda}(\mathbf{r}) - \frac{1}{N}v_{\text{H}}(\mathbf{r}). \tag{4.6}$$

Finite values of λ lead to numerical difficulties [54]. This issue is immediately clear by examining the results for large values of λ. Thus, the most important and time-consuming aspect of this method relies on carefully choosing one or several values of λ. The original work [54] uses an extrapolation technique in which the potentials from the values of $\lambda = \{100, 140, 200\}$ were expanded using a three-term power series in $1/\lambda$. To avoid fitting and make the method more generally applicable, the default for n2v is the iterative method used by Morrison and Zhao [57]. In this approach, one starts with a small value of $\lambda = \lambda_{\mathrm{i}}$, then increases its value to $\lambda = \lambda_{\mathrm{i+1}}$ and incorporates the previous self-consistent potential as a starting potential for the calculation of $\lambda_{\mathrm{i+1}}$. This process is iterated until the desired convergence criteria are reached.

4.1.2 The Wu-Yang Method.

The Wu-Yang method [58] implemented with finite potential basis sets is one of the most efficient methods for iKS problems [59]. With explicitly derived Hessian and gradient, a regular optimizer (e.g. trust-region) will usually converge within 10 iterations. Because of the absence of mesh points, the memory required is also usually acceptable.

The Wu-Yang method optimizes the KS non-interacting kinetic energy T_s under the constraint that the output density density $n(\mathbf{r})$ is the same as the target density $n_{\mathrm{in}}(\mathbf{r})$. By setting the Lagrangian multiplier as the KS potential $v(\mathbf{r})$, which is the condition for the stationary point, the Lagrangian can be built as:

$$W[v] = T_s[\Psi_{\mathrm{det}}[v]] + \int d\mathbf{r} v_{\mathrm{KS}}(\mathbf{r})[n(\mathbf{r}) - n_{\mathrm{in}}(\mathbf{r}], \tag{4.7}$$

where $\Psi_{\mathrm{det}}[v]$ is the Slater determinant corresponding to $v_{\mathrm{KS}}(\mathbf{r})$. $W[v]$ is proven to be concave [58]. Thus, in order to reach the stationary point, the Lagrangian can be optimized with its gradient:

$$\frac{\delta W[v]}{\delta v(\mathbf{r})} = n(\mathbf{r}) - n_{\mathrm{in}}(\mathbf{r}) \tag{4.8}$$

and hessian

$$\begin{aligned}\frac{\delta^2 W[\Psi_{\text{det}}[v_{\text{KS}}], v_{\text{KS}}]}{\delta v(\mathbf{r})\delta v(\mathbf{r}')} &= \frac{\delta n(\mathbf{r})}{\delta v(\mathbf{r}')} \\ &= 2\sum_{\text{i}}^{occ}\sum_{a}^{vir}\frac{\psi_{\text{i}}^*(\mathbf{r})\psi_a(\mathbf{r})\psi_{\text{i}}(\mathbf{r}')\psi_a^*(\mathbf{r}')}{\epsilon_{\text{i}} - \epsilon_a}.\end{aligned} \tag{4.9}$$

$v(\mathbf{r})$ can be decomposed similarly as in ZMP:

$$v(\mathbf{r}) = v_{\text{ext}}(\mathbf{r}) + v_0(\mathbf{r}) + v_{\text{PBS}}(\mathbf{r}), \tag{4.10}$$

where $v_0(\mathbf{r})$ is a guide potential designed to capture known features of the correct KS potential; the Fermi-Amaldi potential is the usual choice. The rest is expanded on a finite potential basis set (PBS) $\{\phi_t\}$

$$v_{\text{PBS}}(\mathbf{r}) = \sum_t b_t\phi_t(\mathbf{r}). \tag{4.11}$$

In n2v, finite difference checks are implemented for both the gradient and hessian in the PBS [59]. This can be a practically useful tool to check the error and convergence. Since the Wu-Yang optimization can be very sensitive for multiple reasons, regularization is important for practice [59].

4.1.3 PDE-Constrained Optimization.

Even though the iKS problem is generally ill-posed [60], it is still supposed that within a reasonably close region around the exact XC potential, the closer the KS density is to the exact density, the more accurate the potential will be [60], [61]. Following this intuition, a density error is defined and optimized under several constraints required by the KS model [60]–[62]. The density error is defined as:

$$N_{\text{error}} = \int d\mathbf{r} \mid n(\mathbf{r}) - n_{\text{in}}(\mathbf{r}) \mid^2 . \tag{4.12}$$

The Lagrangian is defined as:

$$\begin{aligned} L[v, \{\psi_i\}, \{\epsilon_i\}, \{p_i\}, \{\mu_i\}] = & \int (n(\mathbf{r}) - n_{in}(\mathbf{r}))^2 d\mathbf{r} \\ & + \sum_{i=1}^{N/2} \int p_i(\mathbf{r})(-\frac{1}{2}\nabla^2 + v(\mathbf{r}) - \epsilon_i)\psi_i(\mathbf{r}) d\mathbf{r} \\ & + \sum_{i=1}^{N/2} \mu_i(\int |\psi_i(\mathbf{r})|^2 \, d\mathbf{r} - 1), \end{aligned} \tag{4.13}$$

where $\{p_i\}$ and $\{\mu_i\}$ are Lagrange multipliers for the constraints that $\{\psi_i\}$ are the KS orbitals of v with corresponding eigenvalues $\{\epsilon_i\}$ (Equation 2.20) and that $\{\psi_i\}$ are normalized. The normal equations with respect to p_i, μ_i, ψ_i, ϵ_i and v are:

$$(-\frac{1}{2}\nabla^2 + v(\mathbf{r}))\psi_i(\mathbf{r}) = \epsilon_i\psi_i(\mathbf{r}), \tag{4.14a}$$

$$\int |\psi_i(\mathbf{r})|^2 \, d\mathbf{r} = 1, \tag{4.14b}$$

$$\left[-\frac{1}{2}\nabla^2 + v(\mathbf{r}) - \epsilon_i \right] p_i(\mathbf{r}) = 8(n_{in}(\mathbf{r}) - n(\mathbf{r}))\psi_i(\mathbf{r}) - 2\mu_i\psi_i(\mathbf{r}), \tag{4.14c}$$

$$\int p_i(\mathbf{r})\psi_i(\mathbf{r}) d\mathbf{r} = 0, \tag{4.14d}$$

$$\frac{\delta L}{\delta v(\mathbf{r})} = \sum_{i=1}^{N/2} p_i(\mathbf{r})\psi_i(\mathbf{r}). \tag{4.14e}$$

Equations (4.14a) with (4.14b) are solved for the $\{\psi_i\}$; Equations (4.14c) with (4.14d) are solved for the $\{p_i\}$. Then $\{\psi_i\}$ and $\{p_i\}$ can be plugged into (4.14e) to solve for the gradient.

The PDE-Constrained Optimization has been implemented into n2v in finite basis sets as described in [59]. Since the Lagrangian (Equation (4.13)) is not concave as in WY (Equation 4.7) [58], [59], there is no guarantee that the optimization will perform well. Usually a good guide potential (such as Fermi-Amaldi) in Equation (4.10) with a zero initial guess in Equation (4.11) are sufficient to achieve convergence. Moreover, special care should be given to the singularity introduced by subtracting the Hamiltonian operator by its eigenvalues as in Equation (4.14c) and the orthogonality as defined in Equation (4.14d) [61].

4.1.4 The Modified Ryabinkin-Kohut-Staroverov Method.

The three methods above are pure KS inversion methods, in which only the densities are taken as input. The modified Ryabinkin-Kohut-Staroverov (mRKS) method [63], [64] is an accurate method that makes use of one-electron reduced density matrices $\gamma_{\text{in}}(\mathbf{r}, \mathbf{r}')$ and two-electron reduced density matrices $\Gamma_{\text{in}}(\mathbf{r}, \mathbf{r}_2; \mathbf{r}', \mathbf{r}_2')$ obtained from higher-level wavefunction calculations (WF). In mRKS, the $v(\mathbf{r})$ is solved self-consistently on Kohn-Sham systems (KS) in equation:

$$v(\mathbf{r}) = v^{\text{hole}}(\mathbf{r}) + \bar{\epsilon}^{\text{KS}}(\mathbf{r}) - \bar{\epsilon}^{\text{WF}}(\mathbf{r}) + \frac{\tau_P^{\text{WF}}(\mathbf{r})}{n^{\text{WF}}(\mathbf{r})} - \frac{\tau_P^{\text{KS}}(\mathbf{r})}{n^{\text{KS}}(\mathbf{r})}. \tag{4.15}$$

The potential of the exchange-correlation hole (v^{hole}), the average local electron energy ($\bar{\epsilon}$), and the Pauli kinetic energy density (τ_P) are defined as:

$$v^{\text{hole}}(\mathbf{r}) = \int d\mathbf{r}_2 \frac{n(\mathbf{r}, \mathbf{r}_2)}{\mid \mathbf{r} - \mathbf{r}_2 \mid}, \tag{4.16a}$$

$$\bar{\epsilon}^{\text{KS}}(\mathbf{r}) = \frac{2}{n^{\text{KS}}(\mathbf{r})} \sum_{\text{i}=1}^{N/2} \epsilon_{\text{i}} \mid \psi_{\text{i}}(\mathbf{r}) \mid^2, \tag{4.16b}$$

$$\bar{\epsilon}^{\text{WF}}(\mathbf{r}) = \frac{2}{n^{\text{WF}}(\mathbf{r})} \sum_{k=1}^{M} \lambda_{kk}(\mathbf{r}) \mid^2, \tag{4.16c}$$

$$\tau_P^{\text{WF}}(\mathbf{r}) = \frac{2}{n^{\text{WF}}(\mathbf{r})} \sum_{k<l}^{M} n_k n_l \mid \chi_k(\mathbf{r}) \nabla \chi_l(\mathbf{r}) - \chi_l(\mathbf{r}) \nabla \chi_k(\mathbf{r}) \mid^2, \tag{4.16d}$$

$$\tau_P^{\text{KS}}(\mathbf{r}) = \frac{2}{n^{\text{KS}}(\mathbf{r})} \sum_{\text{i}<\text{j}}^{M} n_{\text{i}} n_{\text{j}} \mid \psi_{\text{i}}(\mathbf{r}) \nabla \psi_{\text{j}}(\mathbf{r}) - \psi_{\text{j}}(\mathbf{r}) \nabla \psi_{\text{i}}(\mathbf{r}) \mid^2, \tag{4.16e}$$

where the exchange-correlation hole $n(\mathbf{r}, \mathbf{r}_2)$ is

$$n(\mathbf{r}, \mathbf{r}_2) = \frac{\Gamma_{in}(\mathbf{r}, \mathbf{r}_2; \mathbf{r}, \mathbf{r}_2)}{n_{in}(\mathbf{r})} - n_{in}(\mathbf{r}_2), \tag{4.17}$$

The $\{\chi_k\}$ are the natural orbitals of $\gamma_{in}(\mathbf{r}, \mathbf{r}')$, electron density from wavefunction method is $n^{\text{WF}}(\mathbf{r}) = \gamma_{in}(\mathbf{r}, \mathbf{r})$. $\{\lambda_k\}$, $\{f_k(\mathbf{r})\}$ are the eigenvalues and eigenfunctions of the generalized Fock matrix:

$$F(\mathbf{r}, \mathbf{r}') = \left(-\frac{\nabla_{\mathbf{r}}^2}{2} + v(\mathbf{r}) \right) \gamma_{in}(\mathbf{r}, \mathbf{r}') + \int d\mathbf{r}_2 \frac{\Gamma_{in}(\mathbf{r}, \mathbf{r}_2; \mathbf{r}', \mathbf{r}_2)}{\mid \mathbf{r} - \mathbf{r}_2 \mid}. \tag{4.18}$$

and $n_{i,j,k,l}$ is the orbital occupation number. When a new v is obtained from Equation (4.15), it is plugged in into Equation (2.20) to solve for quantities that will fill in the right-hand-side of Equation (4.15). Superscripts “WF" and “KS" are used to distinguish the results of a wavefunction calculation (input) from those of a KS calculation (fKS).

4.1.5 The Ou-Carter method.

Ou and Carter developed a pure iKS method inspired by mRKS [65]. Instead of a cancellation between one equation derived from a wavefunction calculation and one from the KS system, only the KS system is utilized to derive a self-consistent equation for $v(\mathbf{r})$ from the KS equation:

$$v(\mathbf{r}) = \bar{\epsilon}^{\mathrm{KS}}(\mathbf{r}) - \frac{\tau_L^{\mathrm{KS}}(\mathbf{r})}{n^{\mathrm{KS}}(\mathbf{r})} - v_{\mathrm{ext}}(\mathbf{r}) - v_{\mathrm{H}}(\mathbf{r}), \tag{4.19}$$

where τ_L^{KS} is the KS kinetic energy density and

$$\frac{\tau_L^{\mathrm{KS}}(\mathbf{r})}{n^{\mathrm{KS}}(\mathbf{r})} = \frac{|\nabla n^{\mathrm{KS}}(\mathbf{r})|^2}{8\,|n^{\mathrm{KS}}(\mathbf{r})|^2} - \frac{\nabla^2 n^{\mathrm{KS}}(\mathbf{r})}{4n^{\mathrm{KS}}(\mathbf{r})} + \frac{\tau_P^{\mathrm{KS}}(\mathbf{r})}{n^{\mathrm{KS}}(\mathbf{r})}. \tag{4.20}$$

By replacing the KS density everywhere with the accurate input density and the external potential $v_{\mathrm{ext}}(\mathbf{r})$ by an effective $\tilde{v}_{\mathrm{ext}}(\mathbf{r})$, the final expression for $v(\mathbf{r})$ is:

$$v(\mathbf{r}) = \bar{\epsilon}^{\mathrm{KS}}(\mathbf{r}) + \frac{\nabla^2 n_{\mathrm{in}}(\mathbf{r})}{4n_{\mathrm{in}}(\mathbf{r})} - \frac{|\nabla n_{\mathrm{in}}(\mathbf{r})|^2}{8\,|n_{\mathrm{in}}(\mathbf{r})|^2} - \frac{\tau_P^{\mathrm{KS}}(\mathbf{r})}{n^{\mathrm{KS}}(\mathbf{r})} - \tilde{v}_{\mathrm{ext}}(\mathbf{r}) - v_{\mathrm{H}}[n_{\mathrm{in}}](\mathbf{r}). \tag{4.21}$$

Because in this method there is no error cancellation between wavefunction results and KS-DFT results as in mRKS, using $\tilde{v}_{\mathrm{ext}}(\mathbf{r})$ is necessary to eliminate the errors (mostly numerical) in each component [66], [67]. This is achieved by doing one extra calculation with a known (and simple) functional (usually LDA) on the same basis set to handle the error that comes from finite basis sets [66]:

$$\tilde{v}_{\mathrm{ext}}(\mathbf{r}) = \bar{\epsilon}^{\mathrm{KS}}(\mathbf{r}) - \frac{\tau_L^{LDA}(\mathbf{r})}{n^{LDA}{}_{\in}} - v_{\mathrm{H}}[n_{\mathrm{in}}^{LDA}](\mathbf{r}) - v^{LDA}[n_{\mathrm{in}}^{LDA}](\mathbf{r}), \tag{4.22}$$

where $\frac{\tau_L^{LDA}(\mathbf{r})}{n^{LDA}{}_{\in}}$ is defined by Equation 4.20.

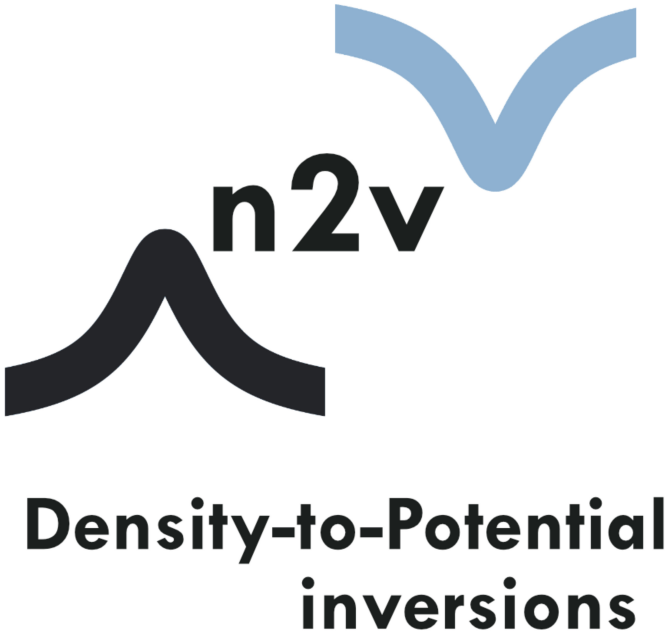


Figure 4.1. The logo for n2v: density-to-potential inversions.

In this section we present the library n2v (n2v stands for “density-to-potential"). n2v is a free, open-source library that offers many of the widely used and recent inversion methods. Six selected methods are implemented and studied: the direct KS-inversion method [51], the modified Ryabinkin-Kohut-Staroverov method (mRKS) [63], [64], the Wu-Yang method (WY) [58], the error function partial-differentiation-equation constrained-optimization method (PDE-CO) [60], [61], the Zhao-Morrison-Parr method (ZMP) [54] and the Ou-Carter methods (OC) [65]. mRKS is accurate but expensive to run [67]. WY is the lightest and most efficient method. ZMP and WY are most widely used. PDE-CO was implemented in various approaches and shows promising features and capabilities. OC is a pure iKS method that balances well the efficiency and accuracy. Our library is written in the high-level programming language Python, allowing us to focus on readability and rapid

development. This enables users to quickly learn and use the library as well as the inverse Kohn-Sham methodology. We aim to follow the rapid-prototyping scheme introduced by Psi4Numpy[12], a repository of reference implementations and interactive tutorials. By using this methodology, we provide access to a set of inversion methods that can be learned and modified quickly, leading to faster implementations of new applications and methods.

In a nutshell, n2v implements high-level details of the inversion algorithms but relies on other computational chemistry packages to perform other lower-level operations such as computing the orbitals, densities and energies.

n2v has an upstream dependence on many other canonical Python libraries. For example, Numpy[68] is used for general numerical operations; Scipy[69] is used to minimize functionals; opt_einsum[70] is used for efficient contraction of matrices; Pylibxc[71] allows computation of exchange-correlation functionals, and Matplotlib[72] is used for visualization.

4.2.1 Code Overview.

From its early design, we have focused on making n2v easy to learn, use, maintain and extend. We believe we have accomplished this by focusing on the following points:

- Readability. In addition to providing an extensive documentation for every component of our code, we use an automatically generated documentation through the use of Sphinx[73]. This can be accessed through the website: wasserman-group.github.io/n2v/. Additionally, we created a set of tutorials in Jupyter Notebooks, providing instructions into using our code and details of the algorithm for every method. The Notebooks can be accessed through the following repository: github.com/wasserman-group/n2v_examples. All of the results shown in this work are computed in the same fashion as the examples.

- Accessibility. Users can run our code in the Windows (through the use of Windows Subsystem), Linux, and macOS operating systems. We recommend the use of conda and the Python Package Index (PyPI). To begin with, at least one engine should be installed (see Section 3.2). For example, Psi4 can be installed with conda install -c psi4 psi4, and Pyscf can be installed with pip install pyscf. Additionally, libxc is required for the Ou-Carter method. To access libxc through python, one must obtain libxc

through conda and proceed to manually install pylibxc. More details are available on the repository. Proceed to install n2v through github.com/wasserman-group/n2v or by direct install with pip install ntov.

- Modern software development. We follow the "best practices" [74] paradigm defined by The Molecular Science Software Institute [75] that covers testing, code coverage, and continuous integration.

- Licensing We released our code under the BSD 3-clause license, a permissive license that allows for our code's modifications and redistribution.

4.2.2 Code Structure.

The software ecosystems within the CMS community are vast, and with the increasing popularity of high-level languages such as Python[76]–[81] or Julia[82]–[84], they will only continue to grow. Nowadays, developers designing software must keep in mind that users do not want to start using a new package to access different features[85]. To aid in lowering the barrier of using n2v, we introduce engines. An engine is an abstract class in Python that specifies what a computational chemistry code interfaced with n2v is expected to provide. These include basic information like geometry and basis set, as well as ingredients needed for each inversion method, such as the orbitals and densities in the atomic orbital basis set and, in some cases, on the grid.

The Psi4[11], [86], [87] and PySCF[88], [89] engines are available in the current release. Psi4 interfaces with our code through PsiAPI, its application programming interface (API). The API allows users to access most of Psi4's core C++ libraries at the python level. On the other hand, PySCF is written in Python and thus can be connected to n2v directly. The Psi4 engine can perform all the available methods, while the PySCF engine can perform all except the Ou-Carter and mRKS methods. Given the intrinsic dependence on the grid of these two methods, they are still under development, but they will be ready in a future release.

Two other classes are relevant to understanding the structure of n2v. The first one is the Inverter, a high-level class that requests tasks from the engine. Additionally, the Inverter collects all of the information needed for an inversion, generates a guiding potential and routes it to the requested method. Each of the methods from section 2 is contained within its class, so the Inverter inherits properties from all of them. In this way, once an Inverter object has been initialized, the user can access all of the different methods with the same object.

To perform some of the methods and visualize the resulting potentials, the class Grider is supplied to each engine. A Grider object computes relevant quantities any given grid, such as density, orbitals and their derivatives, and more. Each of the available engines (Psi4 and PySCF) uses their default spherical grid to compute the DFT kernel on the grid. For example, Psi4 uses a Lebedev-Treutler grid using 75 radial and 302 spherical points. Although the default is robust enough to express exchange-correlation potentials on the grid, it is advised that users refine the parameters according to each system and method.

We encourage other developers to write their engines and make them available for future releases through pull requests in the repository. Our licensing allows any user to use and modify any code component that is useful in their projects.

4.2.3 Using n2v.

To illustrate how to perform a calculation using n2v, we present a minimal example of a Neon atom using the Wu-Yang inversion method (See Figure 4.2) using the Psi4 engine, for a comparison with the PySCF engine, please look at Figure 4.3. The example below is general enough to provide users with an idea of how to use any of the methods discussed in this paper. This example was run in a Jupyter Notebook and can be accessed through the link github.com/wasserman-group/n2v_examples. To better describe its usage, the calculation is separated into three groups highlighted by each cell in the Jupyter Notebook. These correspond to 1) the calculation of target density using Psi4, 2) defining an Inverter object and the inversion itself, and 3) the generation of the potential on the grid handled by the Grider and additional data (Figure 4.2).

1) Calculation of target density. We define the geometry as a standard Psi4 geometry. Units should be set to atomic units and symmetry should be set to C1. We are currently working on exploiting the irreducible representations that would allow different symmetries to be used. Our code takes the reference used – restricted or unrestricted – from the Psi4 set option. Except for the mRKS method, every other method is available in the unrestricted scheme. In this example, we obtain the target density using the Coupled-Cluster with singles and doubles determinants (CCSD) method. Psi4 does not compute the CCSD density from a simple energy calculation. This means that we need to generate a property such as the "Dipole". Other SCF methods won't depend on this step but users should refer to the documentation of each engine for more information.

2) Inverting the density. Initialize an Inverter object by selecting an engine as well as setting the system and the target components. Alternatively, one can simply initialize it by providing a wfn object from the calculation. If no additional basis-set is given, the basis set used will be the same as the energy calculation. Here we specified a larger basis set to be exclusively used for to express the inverted potential. To perform the inversion, we use the method "invert" that requires as basic arguments the "method", followed by the"guide_component" that allows to select a starting point for the inversion. In the example, we use the "Fermi-Amaldi Potential" that ensures that the resulting potential has the correct decay as r tends to infinity.

3)Visualizing the resulting potential. To visualize the inverted potential we require to prepare all quantities on the grid. In this case, we are interested on visualizing a component of the Kohn-Sham effective potential, the exchange-correlation potential. We build a grid by using the linspace function in numpy. Then we make use of the esp function of the Grider to calculate the Hartree and Fermi-Amaldi potentials, followed by the actual potential inverted (rest). This is necessary since the optimizer did not have to invert the full potential, but only a part of it. By adding all of the components accordingly one can visualize the potential. It should be noted that not every method generates the same output; users should refer to the documentation of each method to ensure that they are visualizing the appropriate potential.

[1]:
```python
import n2v
import psi4
import numpy as np
import matplotlib.pyplot as plt

# Define a Psi4 geometry. Set symmetry as C1. Set units as Bohr.
Ne = psi4.geometry("""
Ne
symmetry c1
units bohr
""")

psi4.set_options({"reference" : "rhf"})

# Perform Calculation. Store wavefunction.
# The energy method not always constructs the density. Obtain a property instead.
wfn = psi4.properties('ccsd/cc-pvqz', molecule=Ne, return_wfn=True, properties=['dipole'])[1]

# Extract data needed for n2v.
da, db = np.array(wfn.Da()), np.array( wfn.Db())
ca, cb = np.array(wfn.Ca_subset('AO', 'OCC')) , np.array(wfn.Ca_subset('AO', 'OCC'))
ea, eb = np.array(wfn.epsilon_a()), np.array(wfn.epsilon_b())
```

[2]:
```python
# Initialize inverter object manually.
ine = n2v.Inverter( engine='psi4' )
ine.set_system( Ne, 'cc-pvqz', pbs='aug-cc-pvqz', wfn=wfn )
ine.nalpha = wfn.nalpha()
ine.nbeta = wfn.nbeta()
ine.Dt = [da, db]
ine.ct = [ca, cb]
ine.et = [ea, eb]

# Or extract information directly from the wavefunction object.
ine = n2v.Inverter.from_wfn(wfn, pbs='aug-cc-pvqz')

# Perform Wu Yang Inversion with "Fermi-Amaldi" potential as guide.
ine.invert("WuYang", guide_components="fermi_amaldi")
```

```
Optimization Successful within 3 iterations! |grad|=8.50e-04
```

[3]:
```python
# Generate plotting grid
x = np.linspace(0,5,3001)[:,None]
y = np.zeros_like(x)
z = np.zeros_like(x)
grid = np.concatenate((x,y,z), axis=1).T

# Extract quantities on the grid.
ext, ha ,esp = ine.eng.grid.esp(grid=grid)
vrest = ine.eng.grid.ao(ine.v_pbs, grid=grid, basis=ine.eng.pbs)
fa = (1-1/(ine.nalpha + ine.nbeta)) * ha
vxc = fa + vrest - ha
```

Figure 4.2. The input as a Jupyter Notebook for calculating the $v_{\mathrm{xc}}[n]$ for the Neon atom with a CCSD target density.

Figure 4.3. Side-by-side comparison of n2v running with two different Engines a)Psi4 and b)PySCF. Each cell highlights the similarities and differences related to using the two engines.

5. DFT IN A PROLATE SPHEROIDAL GRID

Diatomic molecules are among the most revealing systems in quantum chemistry. They're small in size which make their computational requirements feasible, and thus we can often compare an approximation with the exact result. Additionally, they highlight many of the present and ubiquitous problems of modern quantum chemistry approximations. Finally, if we consider an embedding calculation with only two fragments having one atom each, diatomics are the canonical example where embedding methods can be implemented and tested. This is why an open-source code that can provide numerically exact results for DFT calculation is an extremely desirable tool in the CMS community.

In this chapter we introduce pyCADMium, a Python module that uses a prolate spheroidal (PS) coordinate grid to accurately perform computational chemistry calculations on atoms and diatomic molecules. The name is an acronym for "Chemical Atoms in Molecules" that references the idea of chemically significant fragments within a molecule. pyCADMium originated in a proprietary programming language but has been rewritten from the ground up as an open-source alternative. Figure 5.1 shows the logo for the project.

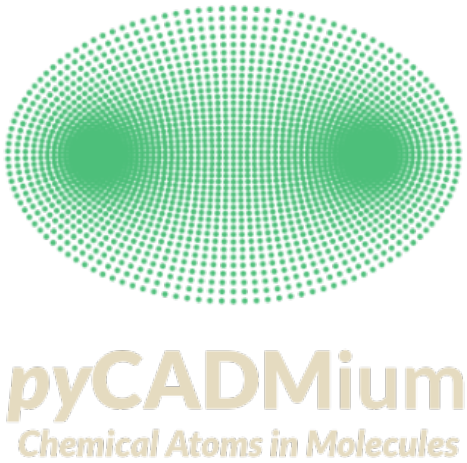


Figure 5.1. Logo of pyCADMium

Most practical calculations use a basis-set to represent operators and quantities like potentials, orbitals, and densities [90], [91]. The lack of an accurate space representation of these and other operators gives rise to the basis-set incompleteness error. This error can be minimized by increasing the number of basis functions used, but it cannot be entirely eliminated in practice[1]. On the other hand, grid-based methods intrinsically allow for an accurate representation [92]. Nevertheless, the number of points required to achieve a significant accuracy can become quickly unmanageable. We wrote pyCADMium , which uses a PS grid to circumvent these issues. Atoms and diatomics are ideally suited for this coordinate system given that each foci can be used to allocate an atom. Additionally, the PS grid is denser around the foci in its Cartesian representation, where we expect functions of molecular systems to change rapidly [93].

5.1 The Coordinate System

The PS grid is a two-center orthogonal coordinate system formed by revolving an elliptic coordinate plane around the line intersecting two foci. These planes are formed by ellipses and hyperbolae that share the same focus [94]. Assume that we place the foci in the Cartesian coordinates $(0, 0, -a)$ and $(0, 0, a)$. We can transform from one coordinate system to the other with

$$\begin{aligned} x &= a\ sinh(\mu)\ sin(\nu)\ cos(\phi) \\ y &= a\ sinh(\mu)\ sin(\nu)\ sin(\phi) \\ z &= a\ cosh(\mu)\ cos(\nu) \end{aligned} \tag{5.1}$$

Where a is the distance between the foci, and μ and ν correspond to the ellipsoidal and hyperboloid surfaces. In the case of an axially symmetric system, the molecular orbitals of interest are independent of the azimuth angle according to Equations (5.1). The ϕ coordinate can be treated analytically and our problem now becomes two-dimensional.

[1]↑We exclude any discussion about numerical procedures such as basis-set extrapolation techniques.

To allow a more accurate description of the systems around the nuclei, the coordinates can be mapped into a new set of variables

$$\mathfrak{e} = cosh^{-1}(\mu), \quad \mathfrak{e} \in [0, \infty) \tag{5.2}$$

$$\mathfrak{h} = cos^{-1}(\nu), \quad \mathfrak{h} \in [0, \pi] \tag{5.3}$$

In the new coordinates, the 'radial' part of the laplacian is

$$\nabla^2 = \frac{4}{a^2(\mu^2 - \nu^2)} \left\{ \frac{\partial^2}{\partial \mathfrak{e}^2} \frac{\mu}{\sqrt{\mu^2 - 1}} \frac{\partial}{\partial \mathfrak{e}} + \frac{\partial^2}{\partial \mathfrak{h}^2} + \frac{\nu}{\sqrt{1 - \nu^2}} \frac{\partial}{\partial \mathfrak{h}} - \mathbf{m_a^2} \cdot \left[\frac{1}{\mu - 1} + \frac{1}{1 - \nu^2} \right] \right\} \tag{5.4}$$

where $\mathbf{m_a^2}$ is an integer and it specifies the rotation symmetry of the orbitals [95]:

- $m_a = 0 \rightarrow \sigma$ orbitals
- $m_a = \pm 1 \rightarrow \pi$ orbitals
- $m_a = \pm 2 \rightarrow \delta$ orbitals
- $m_a = \pm 3 \rightarrow \phi$ orbitals

Orbitals that have a higher symmetry than ϕ are not relevant for diatomic molecules at the DFT level of theory. Orbitals that have the same radial part and the same $m = \pm m_a$ belong to the same shell.

5.1.1 Symmetry Considerations

These coordinates have a convenient correspondence with the distances between any point at either of the foci of the coordinate system.

$$\begin{aligned} r_1 &= 2a(cosh(\mathfrak{e}) + cos(\mathfrak{h})) = a(\mu + \nu) \\ r_2 &= 2a(cosh(\mathfrak{e}) - cos(\mathfrak{h})) = a(\mu - \nu) \end{aligned} \tag{5.5}$$

where r_1 and r_2 represent the distances of any point (x, y, z) to the foci. Additionally, the scales factors and the volume element for a PS grid are

$$h_1 = h_2 = (r_1 \cdot r_2)^{1/2} \tag{5.6}$$

$$h_3 = a \cdot sinh(\epsilon) \cdot sin(\hbar) \tag{5.7}$$

Further discussions on the scale factors can be found on the curvilinear coordinate system section of Reference [94]. Equations 5.5 reveal a lot of interesting properties of the transformation in Equations 5.2 and 5.3. If the sign of μ or ν is changed, then the point $(x, y, z) \rightarrow (-x, -y, z)$. Thus, a rotation by π radians leaves orbitals with even m unchanged but reverses the sign of the orbitals with odd m-values:

– Functions with $\sigma, \delta, ...$ symmetry are even functions of (μ, ν)

– Functions with $\pi, \phi, ...$ symmetry are odd functions of (μ, ν)

Mathematically, this is expressed as

$$f(\mu, \nu) = (-1)^m f(\mu, -\nu) \tag{5.8}$$

$$f(\mu, \nu) = (-1)^m f(-\mu, \nu) \tag{5.9}$$

$$f(\pi + \mu, \nu) = (-1)^m f(\pi - \mu, \nu) \tag{5.10}$$

Due to these properties, we are able to use the central difference formula in any function near the boundary line. These relations are also used to adjust f along $(0, \mu)$, (π, μ) and $(\mu, 0)$ boundary lines for σ functions. Any function that has a higher σ symmetry vanish at these boundary lines.

5.2 Discretization

To discretize our system, we build a two-dimensional mesh with two evenly spaced one-dimensional grids. Since this system is orthogonal, the mesh is a product of two independent one-dimensional meshes that generate a rectangle of N_a x N_r points, representing the number

of angular and radial points. With appropriate boundary conditions, only half of the rectangle is required for homonuclear diatomics, otherwise the whole rectangle is constructed. After having defined our space, we require our differentiation and integration operations. Integration is performed through

$$\int f(\mathbf{r})\, d\mathbf{r} \simeq \sum_{\mathrm{ij}}^{N_1} w_{\mathrm{ij}}\, f_{\mathrm{ij}} \tag{5.11}$$

To calculate the weights proceed as stated in the Appendix D. The first and second derivatives of the Laplacian defined on 5.4 are approximated by finite different expression from the central difference formula.

$$\frac{du}{dx_{\mathrm{j}}} = \frac{1}{dx} \cdot (c1 \cdot u_{[\mathrm{i-j}]} + c2 \cdot u_{[\mathrm{j}]} + c3 \cdot u_{[\mathrm{j+1}]}) \tag{5.12}$$

where $\{c_{\mathrm{i}}\}$ are the finite difference coefficients. An additional requirements for these operators is that they must have the appropriately boundary conditions for each symmetry class, therefore a distinct matrix is needed for each of the different orbitals symmetries.

Although we have computed the integration and differentiation operators, we have not discuss how to produce the potentials that are usually used in a KS-DFT calculation. Each of them needs to be addressed individually.

- The external potential is found by substitution of the transformation equations.

$$v_{ext}(\mathbf{r}) = -\frac{Z(cosh(\mathfrak{e}) - cos(\mathfrak{h}))}{a(cosh^2(\mathfrak{e}) - cos^2(\mathfrak{h}))} \tag{5.13}$$

- The Hartree or the Coulomb potential arises from the total electron density. Although we know its explicit form as an integral involving the electronic density, it is more commonly known to be determined by solving the Poisson equation

$$\nabla^2 v_{\mathrm{H}} = -4\pi n(\mathbf{r}) \tag{5.14}$$

The Poisson equation is not solved by direct substitution of the matrices that we produced in the previous section. Instead they are solved using an LU decomposed Laplacian followed by setting up the boundary conditions as suggested in Reference [95].

- The exchange-correlation potential is found through the library of exchange-correlation functionals libxc [96]. The library does not care about the coordinate system of the density, it only requires as input the value of the density and the gradient at the total discretized points. Currently no higher rungs than GGA are available in the current version of pyCADMium .

5.3 Usage

In this next section, we briefly go over each of the steps that one needs to do to use the many features in pyCADMium . We will start with defining the PS grid. To define a grid within pyCADMium , we need to specify the following parameters

```
#
from pycadmium import Psgrid
#
a  = 2.615/2                    # Distance between the two foci / 2 (a.u.)
NP = 7                          # Number of points per block
NM = [6,6]                      # Number of blocks [angular, radial]
L = np.arccosh(max_rad/a)       # Maximal radial coordinate value
loc = np.array(range(-4 , 5))   # Stencil for finite difference
#
grid = Psgrid(NP, NM, a, L, loc)
grid.initialize()               # Define and initialize grid
#
```

where a is half the distance between the two foci. NP refers to the number of points in a block, NM is the number of points per block, given separately for the 'angular' and 'radial' one dimensional grids to define the PS rectangle. The blocking of the points does not affect

Table 5.1. Ground State Configuration of the first row homonuclear diatomics.

B_2	$1\sigma_g^2\ 1\sigma_u^2\ 2\sigma_g^2\ 2\sigma_u^2\ 1\pi_u^2$	$^3\Sigma_g^-$
C_2	$1\sigma_g^2\ 1\sigma_u^2\ 2\sigma_g^2\ 2\sigma_u^2\ 1\pi_u^4$	$^1\Sigma_g^+$
N_2	$1\sigma_g^2\ 1\sigma_u^2\ 2\sigma_g^2\ 2\sigma_u^2\ 3\sigma_u^2\ 1\pi_u^4$	$^1\Sigma_g^+$
O_2	$1\sigma_g^2\ 1\sigma_u^2\ 2\sigma_g^2\ 2\sigma_u^2\ 3\sigma_g^2\ 1\pi_u^4\ 1\pi_g^2$	$^3\Sigma_g^-$
F_2	$1\sigma_g^2\ 1\sigma_u^2\ 2\sigma_g^2\ 2\sigma_u^2\ 3\sigma_g^2\ 1\pi_u^4\ 1\pi_g^4$	$^1\Sigma_g^+$

the calculation numerically, it only multiplies each of the values of NM so that the total number of points in the grid equals (NP-1) * (NM[0]) * (NP-1) * (NM[1]). We defined the size of the box as the maximal radial coordinate value. This quantity is modified according to section 5.1. The value of L must be contained within (-1,1). loc is required to select the coefficients for the finite difference approximation. Lastly, we provide all of the defined elements to generate a PSgrid object and we proceed to initialize it.

5.4 Molecules on the PS grid

Let us focus on the first row homonuclear diatomics B_2, C_2, N_2, O_2 and F_2. To use pyCADMium we stated that one needs to know about the orbital configuration of the diatomics. Table 5.1 shows the orbital configuration for these and other systems.

Let us use the F_2 molecule as the example of the chapter. To define the geometry in pyCADMium we need to provide the following

```
#
a   = 2.615/2       # Separation distance / 2 (a.u.)
Za  = 9.0           # Fluorine Atom 1 Nuclear Charne
Zb  = 9.0           # Fluorine Atom 2 Nuclear Charge
pol = 1             # Set unpolarized system
Nmo = [[5], [2]]    # Number of Molecular Orbitals
N   = [[10], [8]]   # Number of electrons
#
```

In the previous code, the a parameter is both the separation distance of the nuclei and the foci of the PS grid. We proceed to define the nuclear charges as well as the polarization of

the system. Setting pol = 2 would explicitly compute the α and β components of the density. Finally, the molecular configuration along with its symmetries are specied in the next two lines. Nmo requires the total number of orbitals per symmetry. As we can appreciate from 5.1, this system has a total of 5 σ orbitals and 2 π orbitals, where we allocate 10 and 8 electrons, respectively. Nothing prevents us from plugging in different values of the orbitals and the electrons. Thus one must be mindful of the desired configuration.

5.5 Kohn-Sham Calculation

We are ready to finally complete a calculation using pyCADMium . The first calculation that we are going to do is to obtain the LDA energy of the F_2 molecule. To do so, we need to supply the previous geometry to a Kohnsham object. Besides the geometry, we can supply an additional dictionary with several options to control how our calculation behaves. The simplest example can be constructed as

```
#
from pycadmium import Kohnsham
#
optKS = { # Options for the KS calculation
          'interaction_type' : 'dft',
          'sym'              : True,
          }
optSCF = { # Options for the SCF cycle
          'maxiter' : 100,
          'e_tol'   : 1e-9,
           }
#
KS = KohnSham(grid, Za, Zb, pol, Nmo, N, optKS)
KS.scf(optSCF)
#
```

where we requested a KS-DFT calculation. If no functional is provided, by default pyCADMium uses the LDA. The sym option specifies that only half of the PS plane needs to be constructed due to the symmetry of the F_2 molecule. The next set of options controls

the SCF procedure behaviour. Here we specified that we want a maximum of 100 iterations and that the procedure should stop if the differences in energy are less than $1e-9$. We proceed by defining the Kohnsham object and start the SCF procedure. The results of the calculations such as the energies and potentials is available as properties of the Kohnsham object. The full extent of options for the Kohnsham as well as the scf calculation can be found on Table 5.2

Table 5.2. SCF Options within pyCADMium

SCF Options			
Key	Type	Default	Meaning
disp	bool	True	Displays the steps of the scf cycle
continuing	bool	False	If False, sets the initial guess of the SCF to $v_{ext}(r) + v_{kin}(r)$ If True, uses the veff in memory
iterative	bool	True	If True, allows for an iterative calculation of the orbitals.
spinflipsym	bool	False	If True, it performs an ensemble of spin-up and spin-down for the system in question A system in a non-singlet state must be provided. See
maxiter	int	100	Sets the maximum number of iterations for the SCF procedure
e_tol	float	10e-6	Sets the energy threshold that defines convergence

Table 5.3. Kohn-Sham Options within pyCADMium

Kohn-Sham Options			
Key	Type	Default	Meaning
interaction_type	str	'dft'	Selects type of interaction. Choose between density functional theory ('dft') or non-interacting ('ni')
sym	bool	False	Specifies whether or not there is a homonuclear diatomic. If True, only half of the PS rectangle is generated.
fractional	bool	False	If True, fractional occupation for fragments is allowed.
xfunc_id	int	1	Specify what exchange functional to use. Only local and semilocal functionals are available. IDs are given by libxc: tddft.org/programs/libxc/functionals/
xfunc_id	int	12	Specify what correlation functional to use. Only local and semilocal functionals are available. IDs are given by libxc: tddft.org/programs/libxc/functionals/
xc_family	str	'lda'	Selects the family of exchange-correlation functional. Choose between {'lda', 'gga'}

5.6 P-DFT Calculation

One of the highlights of pyCADMium is its ability to treat each atom as a fragment. This feature allows it to be used for developing embedding methods. In this section, we will briefly discuss the algorithm for a P-DFT calculation. For more information refer to Chapter 3. Consider a set of two fragments $\{n_\alpha(\mathbf{r}), n_\beta(\mathbf{r})\}$, each having $\{N_a, N_b\}$ electrons respectively. We are interested in recreating the results of a KS-DFT calculation of the full system using only the information of the fragments. The number of electrons in each fragment must add up to N_m, the number of electrons in the full molecule. Additionally, we seek to minimize the sum of the energies of each fragments. Mathematically, this can be done through the unconstrained minimization of

$$G[n] = \min_{\{n_\alpha(\mathbf{r})\}} \left\{ E_f[\{n_\alpha(\mathbf{r})\}] + \int d\mathbf{r} \cdot v_p(\mathbf{r}) \cdot (n_f(\mathbf{r}) - n(\mathbf{r})) - \sum_\alpha \mu_\alpha \left(\int d\mathbf{r} \cdot n_\alpha(\mathbf{r}) - N_\alpha \right) \right\} \tag{5.15}$$

Where each $E_f[n_\alpha(\mathbf{r})]$ corresponds to the each of the fragment energies. $E_f[n_\alpha(\mathbf{r})] = T_s[n_\alpha(\mathbf{r})] + E_{\text{Hxc}}[n_\alpha(\mathbf{r})] + \int v_\alpha(\mathbf{r}) n_\alpha(\mathbf{r}) d\mathbf{r}$ and the two other terms are the Lagrange multipliers: the partition potential $v_p(\mathbf{r})$ and the chemical potential μ_α. To minimize, perform a functional derivative with respect to to the density $n_\alpha(\mathbf{r})$

$$\frac{\delta G}{\delta n_\alpha(\mathbf{r})} = 0 \tag{5.16}$$

$$\frac{\delta E_\alpha[\{n_\alpha(\mathbf{r})\}]}{\delta n_\alpha(\mathbf{r})} + v_p(\mathbf{r}) = \mu_\alpha \tag{5.17}$$

To solve the above equation, we use a set of KS systems as fragments, each of them in the influence of the multiplicative potential $v_\alpha(\mathbf{r}) + v_p(\mathbf{r})$. With the exact partition potential, we can solve the Kohn-Sham equations for the P-DFT problem

$$\left\{ -\frac{1}{2}\nabla^2 + v_{\alpha,eff}[\{n_\alpha\}](\mathbf{r}) + v_p[\{n_\alpha\}](\mathbf{r}) \right\} \phi_{i,\alpha}(\mathbf{r}) = \varepsilon_{i,\alpha} \phi_{i,\alpha}(\mathbf{r}) \tag{5.18}$$

Since the potentials inside the brackets depend on the fragment densities—the quantity we are looking for— we solve these equations self-consistently. At convergence, we obtain a set of fragment orbitals that we use to build the fragment densities as $n_\alpha(\mathbf{r}) = \sum_{i}^{N_f} | \phi_{i,\alpha}(\mathbf{r}) |^2$.

In pyCADMium we make use of the class Partition to define an embedded calculation. This class allows us to define the properties of each fragments as well as the properties of the full molecular system that we ought to solve. To define a Partition object, follow the F_2 example that we have been studying

```
#
from pycadmium import Partition
#
optPartition = { # Options for the object Partition
                 'kinetic_part_type' : 'inversion',
                 'ab_sym'            :  True,
                 'ens_spin_sym'      :  True
                 'isolated'          :  True}
               }
#
part = Partition(grid, Za, Zb, pol, MOa, Na, nua, MOb, Nb, nub,
optPartition)
#
part.scf()
#
```

In the previous example, we first set up the options for our object using a dictionary. The full set of options can be found in Table 5.4. By setting isolated = True, we won't be doing a P-DFT calculation just yet. Instead, we want to generate an initial guess for the calculation by simply adding the components of the individual fragments. Since the fragments are have no interaction between each other there is no partition potential, and thus no partition energy.

Table 5.4. Partition options within pyCADMium

Partition Options			
Key	Type	Default	Meaning
vp_calc_type	str	'component'	Choose between {'component', 'potential_inversion'}
hxc_part_type	str	'exact'	Choose between {'exact', 'overlap_hxc', 'overlap_xc', 'hartree'}
kinetic_part_type	str	'vonweiz'	Choose between {'vonweiz', 'inversion', 'libxcke', 'twoorbital', 'fixed'}
k_family	str	gga	Specifies what kinetic functional family to use. Only local and semilocal functionals are available. Used in conjunction with kinetic_part_type = 'libxcke'
ke_func_id	int	5	Specifies what kinetic functional to use. Only local and semilocal functionals are available. IDs are given by libxc: tddft.org/programs/libxc/functionals/ Used in conjunction with kinetic_part_type = 'libxcke'
disp	bool	False	Prints additional information about the calculation in question.
fractional	bool	False	Allows for fractional occupation of fragments.
isolated	bool	False	Are fragments isolated? (Not under the influence of an embedding potential)

Table 5.5. Inversion options within pyCADMium

Inverter Options			
Key	Type	Default	Meaning
use_iterative	bool	False	Speeds up the optimization for the inversion procedure. Currently not reliable.
invert_type	str	'wuyang'	Choose between {'wuyang', 'simple', 'orbitalinvert'}
tol_lin_solver	float	1e-2	Sets the tolerance for the linear solver. Used in conjuction with 'use_iterative'.
tol_invert	float	1e-12	Sets the threshold for the inversion in a single P-DFT step.
max_iter_lin_solver	int	2000	Sets the maximum number of iterations for the SciPy optimizer
max_iter_invert	int	20	Sets the maximum number of iterations to determine the inverted potential.

To find the exact partition potential, we make use of numerical inversions. This allows us to obtain the exact Kohn-Sham potential for the sum of fragment densities[2], as well as the kinetic potential of the sum of fragment densities needed to compute the partition potential.

In pyCADMium , we make use of the class Inverter to access a handful of methods to invert a density, including the already mentioned Wu-Yang method defined in Chapter 4. For brevity, we will only discuss the orbital invert method that has shown to be the most robust and reliable. The numerical inversion is essentially a direct search for orbitals that satisfy the KS equations while satisfying some constraints at each point of the two-dimensional space. The conditions form a set of residuals:

1. Orbitals solve the KS Equations $\rightarrow res_{\mathrm{ij}}^{KS} = \left\{ -\frac{1}{2}\nabla^2\phi_{\mathrm{i}} \right\}_{\mathrm{j}} + v_{s,\mathrm{j}}\phi_{\mathrm{i,j}} - \varepsilon_{\mathrm{i}}\phi_{\mathrm{i,j}}$
2. Orbitals are normalized $\rightarrow res_{\mathrm{ij}}^{N} = \sum_{\mathrm{j}} \mid \phi_{\mathrm{i,j}} \mid^2 -1$
3. Sum of the squares of each orbitals equals the target density $\rightarrow res_{\mathrm{ij}}^{n} = \sum_{\mathrm{j}} \mid \phi_{\mathrm{i,j}} \mid^2 -n_{\mathrm{j}}$

where the j index runs over grid points and the i point runs through the orbitals. Combine all residuals to form a vector function that takes the orbitals, energies and effective potential as its argument. If we find the root of this function, we find the effective potential that reproduces the given density. The Jacobian of the residual function happens to be a sparse square matrix that can be treated with the Newton-Raphson minimization to find which vector minimizes the residuals. Additionally, the HOMO eigenvalue is fixed to be zero to avoid shifting the potential, and the HOMO normalization constraint is removed allowing it to be satisfied by the overall density constraint. These numerical inversion are used at each step in the P-DFT scf procedure to determine the functional derivatives of $T_{\mathrm{s}}^{\mathrm{nad}}[n(\mathbf{r})]$.

[2]↑This quantity is not obtained in a "forward" manner, and thus, we don't have access to the Kohn-Sham potential that produces such a density

Remember that we already constructed an initial guess for our P-DFT calculation. Then, in order to complete it alongside with the inversion procedure, we write

```
#
from pycadmium import Pssovler, Inverter
#
optInverter = { # Options for the object Inverter
                'invert_type'     : 'orbitalinvert',
              }
#
# Inverter Requires a PSsolver Object
mol_solver = Pssolver(grid, MOm, Nm, {})
part.inverter = Inverter(grid, mol_solver, optInverter)
#
# We now want the fragments to be under the presence of the partition
potential
part.optPartition.isolated   = False
# 'continuing' ensures we use the generated initial guess
part.scf({"continuing" : True})
#
```

We will be using the 'orbitalinvert', this can be set in the options for the Inverter. The full set of options can be found in Table 5.5. A Psolver object is needed. This object simply stores the results from a calculation. Each Kohnsham object defines one per orbital. We define the Inverter as an attribute of the partition. We continue by specifying the Partition object that want to perform a P-DFT calculation without isolated fragments and we trigger the SCF calculation.

In this section, we do not discuss any results. To find a set of examples for different use cases of pyCADMium , please visit github.com/wasserman-group/pyCADMium_examples. Where the results for this and many other calculations are found.

6. APPROXIMATING THE KINETIC ENERGY FUNCTIONAL

Consider a system composed of a single electron, its kinetic energy is obtained through

$$T = -\frac{1}{2}\int d\mathbf{r}\cdot\Psi^{*}(\mathbf{r})\cdot\nabla^{2}\Psi(\mathbf{r}) \tag{6.1}$$

Since $\Psi(\mathbf{r})$ can be found with $\Psi(\mathbf{r}) = \sqrt{n(\mathbf{r})}$. von Weizsäcker derived a formula that can exactly describe the kinetic energy of a one-electron system in terms of the density, as opposed to the orbitals (the proof can be found in Appendix C):

$$T_{\mathrm{s}}[n(\mathbf{r})] = \frac{1}{8}\int d\mathbf{r}\cdot\frac{|\nabla n(\mathbf{r})|^{2}}{n(\mathbf{r})} \tag{6.2}$$

The relevance of this equation should not be underestimated. This is one of the only two cases where the kinetic energy is found as an explicit functional of the density. The other case being the Thomas-Fermi kinetic functional, which is exact for the uniform gas and it is given by:

$$T[n(\mathbf{r})] = C_{\mathrm{TF}}\int d\mathbf{r}\cdot n^{5/3}(\mathbf{r}) \tag{6.3}$$

with:

$$C_{TF} = \frac{3}{10}(3\pi^{2})^{2/3} \tag{6.4}$$

Finding approximate density-explicit functionals for any other system has proven to be extremely challenging [97]. This is because, alongside with the nuclear energy, the kinetic energy makes up for most of the electronic energy. Thus, any decent approximation results in terrible electronic energies. In the next section, we focus on different alternatives to build kinetic approximations in the context of embedding methodologies.

6.1 The Original Two-Orbital Approximation

We saw in Chapter 3 that practical P-DFT calculations rely on approximations for the non-additive kinetic energy functional

$$T_{s}^{\mathrm{nad}}[\{n_{\alpha}(\mathbf{r})\}] = T_{s}[\{n_{\alpha}(\mathbf{r})\}] - \sum_{\alpha} T_{s}[n_{\alpha}(\mathbf{r})] \tag{6.5}$$

where each term can be calculated using the definition of the kinetic energy

$$T_{\mathrm{s}}[\{\phi_{\mathrm{i}}(\mathbf{r})\}] = -\frac{1}{2}\sum_{\mathrm{i}=1}^{N}\int \phi_{\mathrm{i}}(\mathbf{r})\nabla^2\phi_{\mathrm{i}}(\mathbf{r})d\mathbf{r} \tag{6.6}$$

Notice that we have made explicit the dependence on the orbitals. Equation (6.5) shows a dependence on the density, but this is only true in an implicit manner. In reality, each term only depends on the Kohn-Sham orbitals which in turn, depend on the density. The goal of a kinetic functional approximation within P-DFT is to write the first term of Equation (6.5) in terms of the fragment densities. In contrast, since the fragment densities are obtained self-consistently, no additional approximation is needed for the second term.

Consider the simplest P-DFT calculation, i.e. a system composed of two fragments. The He_2 is a canonical example since it has two fragments and each fragment is a singlet state with one orbital. Let us perform a Kohn-Sham DFT calculation using the LDA. The solution is a set of two Kohn-Sham orbitals (see Figure 6.1). These Kohn-Sham orbitals have a gerade and ungerade symmetry, and so, we will be referring to them as $\phi^{\mathrm{ger}}(\mathbf{r})$ and $\phi^{\mathrm{ung}}(\mathbf{r})$, respectively. If we know the two Kohn-Sham orbitals, we can use the definition to compute the kinetic energy of this system

$$T_{\mathrm{s}}[\{\phi_{\mathrm{i}}(\mathbf{r})\}] = T_s^{ger}[\phi^{ger}(\mathbf{r})] + T_s^{ung}[\phi^{ung}(\mathbf{r})] \tag{6.7}$$

We want to be able to write $\phi^{\mathrm{ger}}(\mathbf{r})$ and $\phi^{\mathrm{ung}}(\mathbf{r})$ in terms of the known fragment densities. Given that P-DFT fragments are localized in space, we assume that these fragment densities can be constructed from a single orbital (in a similar fashion to how the von Weizsäcker functional is constructed). The orbitals in terms of the fragment densities are

$$\Psi_\alpha(\mathbf{r}) = \sqrt{n_\alpha(\mathbf{r})} \quad \& \quad \Psi_\beta(\mathbf{r}) = \sqrt{n_\beta(\mathbf{r})} \tag{6.8}$$

and we proceed to build an ungerade orbital from these functions

$$\phi^{ung}(\mathrm{r}) \approx \frac{1}{C^{\frac{1}{2}}} \cdot \left\{\Psi_\alpha(\mathbf{r}) - \Psi_\beta(\mathbf{r})\right\} \tag{6.9}$$

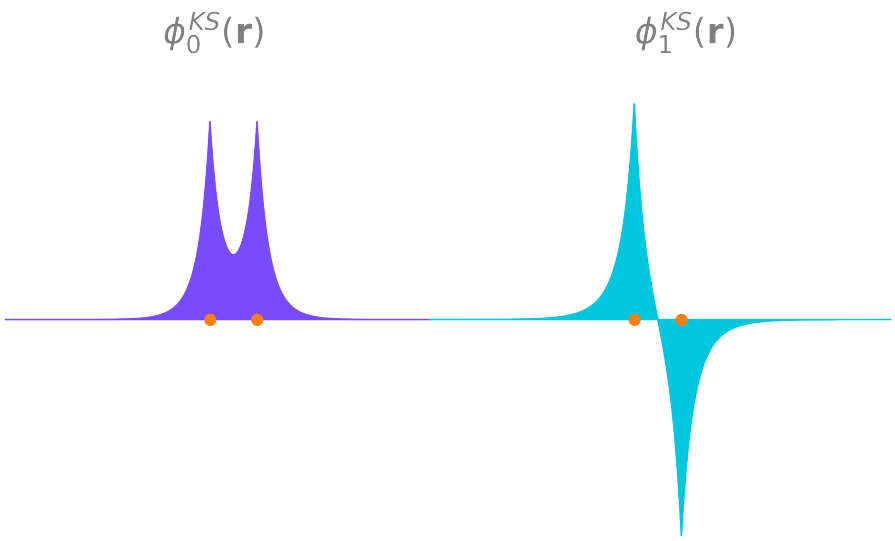


Figure 6.1. The two Kohn-Sham orbitals for He_2, first orbital presents a gerade symmetry whereas the second orbital presents an ungerade symmetry. The orange points represent the position of the two Helium atoms separated 2 a.u.

where C is the normalization factor

$$C = \left\{ \int d\mathrm{r} \cdot \left(\Psi_1(\mathbf{r}) - \Psi_2(\mathbf{r}) \right) \right\}^2 \tag{6.10}$$

To construct the second orbital, make use of the full molecular density. We do not expect to have this quantity in practical calculations, but since He_2 is so small, we can use this as a proof of concept. Since the second orbital has an even symmetry we construct the second orbital as

$$\phi^{ger}(\mathbf{r}) \approx \left\{ \frac{n(\mathbf{r})}{2} - \phi^{ung}(\mathbf{r})^2 \right\}^{1/2} \tag{6.11}$$

It must be noted that for this construction to work, both fragment densities must belong to the same species, otherwise, we would not have the appropriate symmetry requirements. This approximation should give nice results to systems with 4 electrons such as the He_2 and it is expected to be exact in the limit of infinite separation. Let us now use it to compare it against the exact $T_s^{\mathrm{nad}}[n(\mathbf{r})]$.

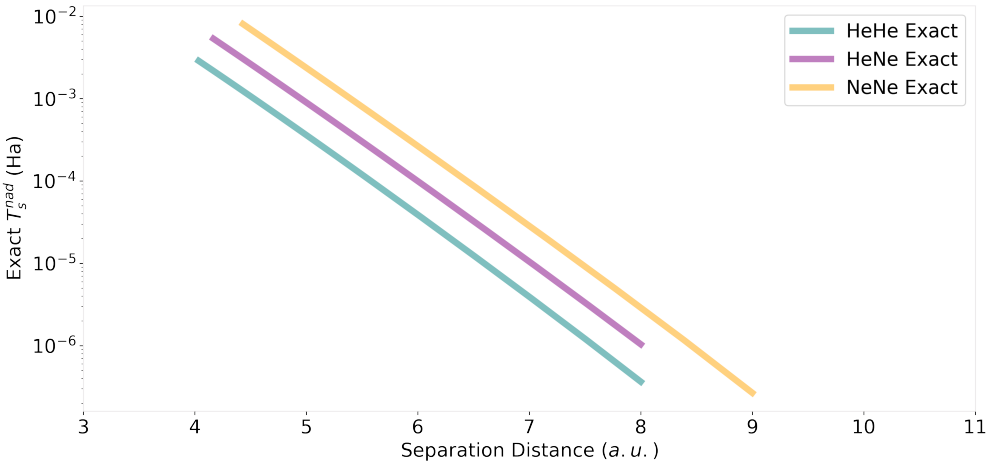


Figure 6.2. Exact non-additive kinetic energies for different diatomic systems at different separations R. Logarithm scale is used to highlight the almost exponential behaviour of the kinetic energy e^{-kR} for different values of the separation distance R.

6.2 Asymptotic Behaviour of Different Diatomic Molecules

To test the approximation, we calculated the $T_s^{nad}[n(\mathbf{r})]$ exactly for a wide range of diatomics: He_2, HeNe, and Ne_2. We used the all-electron real-space code pyCADMium introduced in Section 5. A fine prolate spheroidal grid is used where the nuclei are located at each foci. Using a grid allows the results to be free from basis-set incompleteness errors, providing "exact" results up to an arbitrary accuracy. The LDA minimum was used as the ground state geometry. When we refer to "exact" $T_s^{nad}[n(\mathbf{r})]$ we will be referring to the exact result obtained using the LDA functional. The values of $T_s^{nad}[n(\mathbf{r})]$ are orders of magnitude smaller than the total energy, thus the scf procedure was set to converge only if the changes in energy are lower than 10^{-11}.

Figure 6.2 shows the $T_s^{nad}[n(\mathbf{r})]$ at different internuclear separation distances for different diatomic molecules. One can appreciate that for these systems the $T_s^{nad}[n(\mathbf{r})]$ is always positive. We plot these energies using a logarithm scale, and notice that it behaves almost as an exponential function of the separation distance $T_s^{nad}[n(\mathbf{r})] \sim e^{-kx}$. Additional results showed that even with more sophisticated functionals, such as PBE, the $T_s^{nad}[n(\mathbf{r})]$ does not differ by more than 1%, and thus, we continue with our analysis using the LDA.

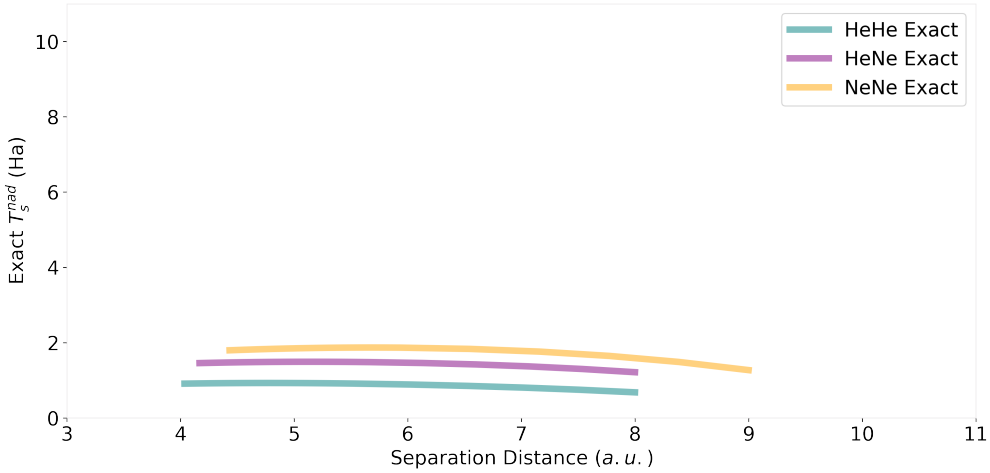


Figure 6.3. Ratio of the exact non-additive kinetic energy and the 2OA kinetic energy, We can appreciate that this function behaves in a linear fashion, suggesting that a system-specific constant is required to accurate reproduce the long range behaviour of the studied systems.

We next compared the exact results with the ones obtained with the 2OA. Figure 6.3 shows the ratio of these two quantities. Each of these functions behaves in a linear way, suggesting that our crude approximation is able to mimic the long-range behaviour of these systems, although only the He_2 is the one closest both in shape and in magnitude, the plot shows that a system specific constant can be added as a factor to the 20A functional to turn into an extremely accurate approximation to the $T_s^{\text{nad}}[n(\mathbf{r})]$. The constants for these systems can be found on [33], but here we focus instead on looking at the orbitals that this approximation produces.

6.3 Analysis of the Shape of Orbitals

Although the orbitals found on Figure 6.1 appear to be a cartoon representation of the Kohn-Sham orbitals, they actually came from a KS-DFT calculation with the LDA approximation. Let us use these orbitals to compare them with the one produced by the 2OA. Since we use the definition of the kinetic energy, the success of our approximation will be dependent on how well we approximate these two functions. To study them further, we again divide the exact Kohn-Sham orbital and the 2OA approximation. The 2OA manages

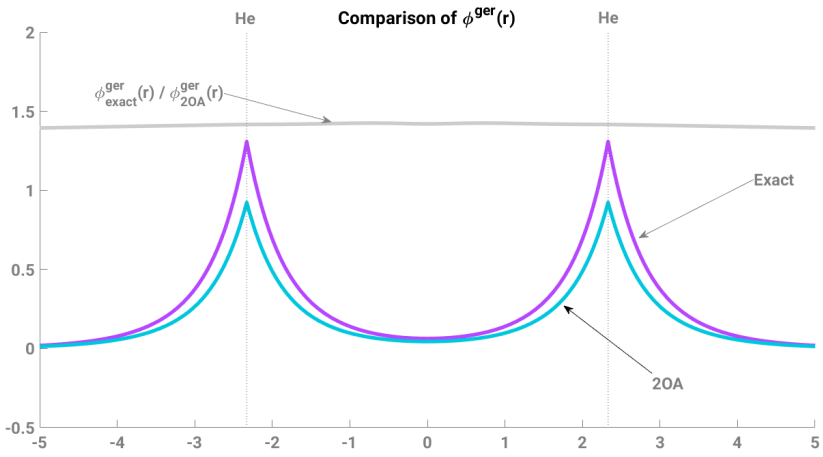


Figure 6.4. Comparison of the exact gerade orbital with the 2OA of the He_2 at the equilibrium geometry. The grey line shows that a system dependent constant may be required to accurately reproduce the exact orbital.

to reproduce the shape of the two orbitals almost perfectly. As in the case with the energies, we can see that for both the gerade in Figure 6.4 and the ungerade in Figure 6.5 reproduce the shape of the exact LDA orbitals almost exactly.

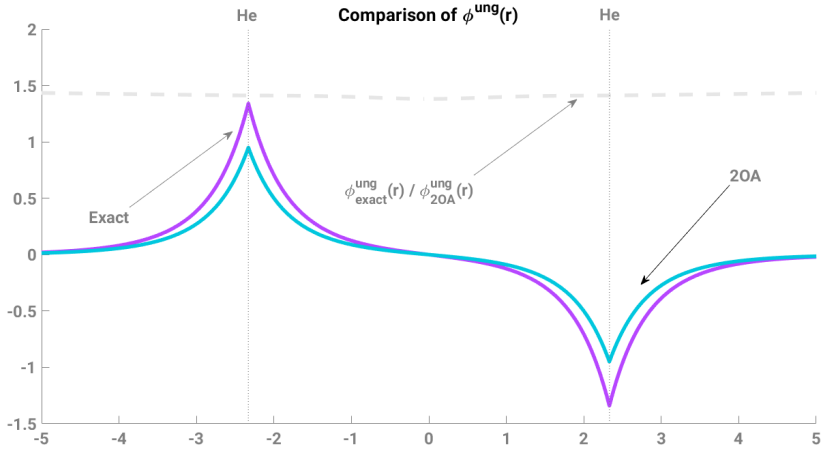
For these two orbitals a constant to approximately 1.5 is required to accurately reproduce the results. This is a great way of analyzing the approximation since the kinetic energy depends entirely on the number and shape of these orbitals. Unfortunately, only systems with up to 2 electrons per fragment can be plotted in such a way.

6.4 Behaviour of the Exact Functional Derivative for $T_s^{\mathrm{nad}}[n(\mathbf{r})]$

Besides energies, we ought to look at the accuracy of the functional derivative of the approximation. We refer to this quantity as the non-additive kinetic potential v_t^{nad}:

$$v_t^{nad}(\mathbf{r}) = \frac{\delta T_s^{\mathrm{nad}}[\{n_\alpha(\mathbf{r})\}]}{\delta n(\mathbf{r})} \tag{6.12}$$

Let us look at the exact behaviour of the v_t^{nad} for the systems that we have discussed so far. We are interested in two different geometries: 1) The equilibrium distance and 2) The limit of large separation. The latter one depends on the numerical ability of the code to



[h]

Figure 6.5. Comparison of the exact ungerade orbital with the 2OA of the He_2 at the equilibrium geometry. The grey line shows that a system dependent constant may be required to accurately reproduce the exact orbital.

allows us to construct a large discreet space, so most of the molecules presented the longest separation was of 8 bohr. Figure 6.6 shows the exact v_t^{nad} for the He_2, we can see that its most prominent feature is a double peak with a well at the middle of the bond axis, where the overlap between the fragment densities finds its maximum. Since v_t^{nad} is added to every fragment's potential, it is responsible for polarizing each of the fragment densities towards the bonding region. As the separation distance is increased, the peaks decrease in magnitude while the well in between the fragments becomes deeper.

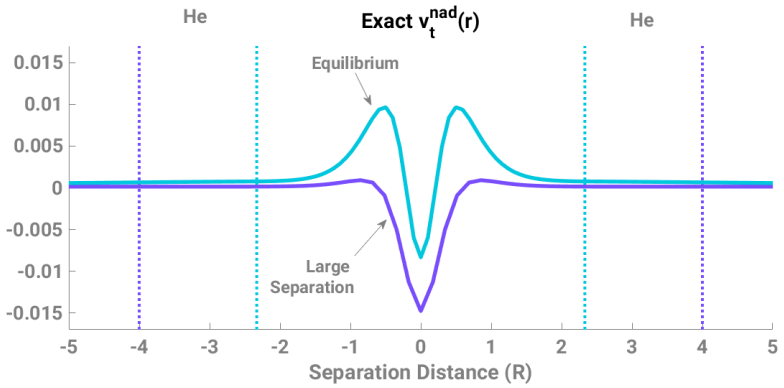


Figure 6.6. Exact v_t^{nad} for the HeHe at two different separation distances: 1) Ground-state separation in Teal and 2) Long separation in Purple.

Figure 6.7 shows a similar plot but for HeNe. The two peaks are again present, with the difference that there is the asymmetry in the peaks that occurs since the system is now a heteronuclear diatomic. Most notably there is a slight plateau in the neighborhood of the Ne atom, a feature that appears commonly in embedding potentials [23] . The edges of the plateau correspond to the transitions between regions where the core orbital contributes the most density and regions where the HOMO contributes the most density. Another relevant feature to notice is that the decay on each side is different, which is also expected from a heteronuclear diatomic [93].

Let us continue by comparing the functional derivative of the 2OA approximation with the exact result that we obtain from inversion. Figure 6.8 shows the two different potentials for the He_2 at equilibrium separation and long distance separation. Although the order of

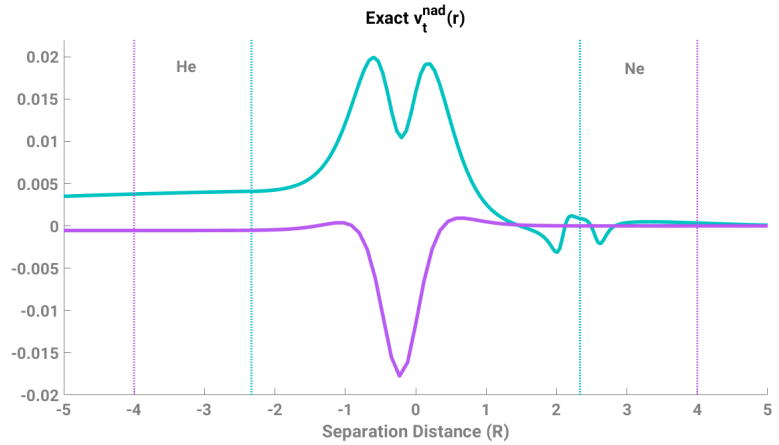


Figure 6.7. Exact v_t^{nad} for the HeNe at two different separation distances: 1) Ground-state separation in Teal and 2) Long separation in Purple

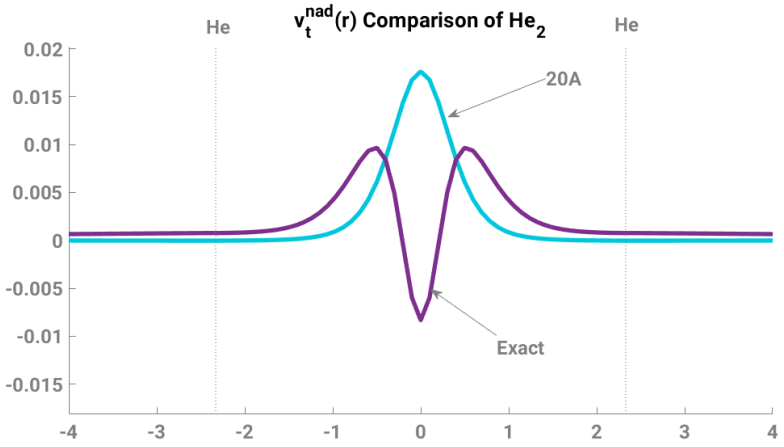


Figure 6.8. v_t^{nad} Comparison between the exact and 2OA potentials for the HeHe molecular at the equilibrium geometry

magnitude for each point of the 2OA seems to be in agreement with the exact potential, we can immediately see that the physics are all incorrect: this potential prevents both fragments from moving towards the bonding region.

Last but not least, we perform a similar analysis but with the HeNe molecule. Surprisingly, we notice here that the approximated 2OA potential now has a magnitude that is considerably larger than the expected potential now over-binding the fragments towards the

bonding region as well as failing to reproduce any other relevant feature in the neighborhood of the nuclei.

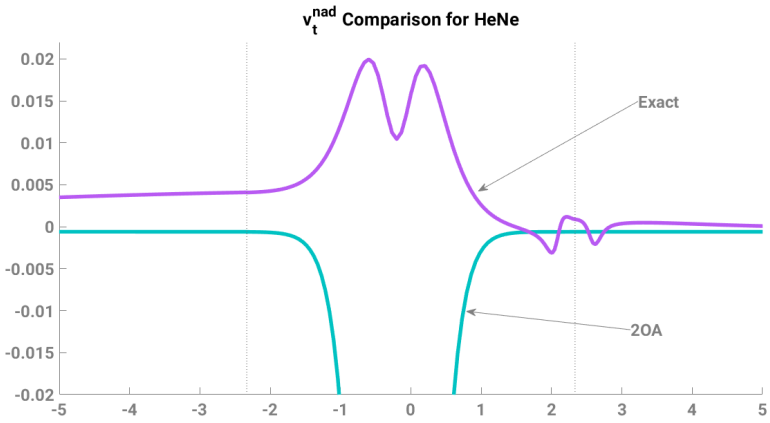


Figure 6.9. v_t^{nad} Comparison between the exact and 2OA potentials for the HeNe molecular at the equilibrium geometry

All-in-all it can be seen that although the 2OA gives very accurate results for the molecule in the asymptotic region, there is still some deficiencies in the approximation: The 2OA completely changes the qualitative behaviour of the potential. We took advantage of the symmetry of the molecule, something that we would not be able to exploit for a non-homonuclear diatomic; and finally we made use of the full molecular density, something that we may not be able to have access to. Let us proceed by modifying some aspects of the approximation and hopefully improve on the negative aspects of the 2OA.

6.5 The modified Orbital Approximation: mOA

Let us now focus on building these two orbitals from a couple of densities $\{ n_1(\mathbf{r}), n_2(\mathbf{r}) \}$ each of them having $\{ N_\alpha, N_\beta \}$ number of electrons respectively (notice that this procedure

could be performed for k orbitals with k fragment densities). From the assumption that each density arises from a single orbital, we can generate each fragment wavefunction as:

$$N_\alpha \cdot \Psi_\alpha(\mathbf{r})^2 = n_\alpha(\mathbf{r}) \rightarrow \sqrt{\frac{n_\alpha(\mathbf{r})}{N_\alpha}} = \Psi_\alpha(\mathbf{r}) \tag{6.13}$$

$$N_\beta \cdot \Psi_2(\mathbf{r})^2 = n_2(\mathbf{r}) \rightarrow \sqrt{\frac{n_2(\mathbf{r})}{N_\beta}} = \Psi_2(\mathbf{r}) \tag{6.14}$$

With an analogous equation for the second fragment. These orbitals are obtained independently and thus, we do not expect them to be orthogonal. We can rotate the orbitals to make them orthogonal as suggested by Lowdin [98].

Let

$$d = \int \cdot n_\alpha^{\frac{1}{2}}(\mathbf{r}) \cdot n_\beta^{\frac{1}{2}}(\mathbf{r}) \cdot \frac{1}{\sqrt{N_\alpha N_\beta}} \tag{6.15}$$

and

$$S = \begin{bmatrix} 1 & d \\ d & 1 \end{bmatrix} \tag{6.16}$$

We can compute the new orbitals as a linear combination of the old orbitals. Each coefficient is given by the overlap matrix:

$$S^{-\frac{1}{2}} = \frac{1}{2\sqrt{1-d^2}} \begin{bmatrix} \sqrt{1-d} + \sqrt{1+d} & \sqrt{1-d} - \sqrt{1+d} \\ \sqrt{1-d} - \sqrt{1+d} & \sqrt{1-d} + \sqrt{1+d} \end{bmatrix} \tag{6.17}$$

And so we find the orthogonal orbitals

$$\Psi_1^+(\mathbf{r}) = [S^{-\frac{1}{2}}]_{11} \cdot \Psi_1(\mathbf{r}) + [S^{-\frac{1}{2}}]_{12} \cdot \Psi_2(\mathbf{r}) \tag{6.18}$$

$$\Psi_2^+(\mathbf{r}) = [S^{-\frac{1}{2}}]_{21} \cdot \Psi_1(\mathbf{r}) + [S^{-\frac{1}{2}}]_{22} \cdot \Psi_2(\mathbf{r}) \tag{6.19}$$

In this way, we can generate our gerade and ungerade orbitals

$$\phi^{\text{ung}}(\mathbf{r}) \approx \frac{1}{\sqrt{2 - 2\int \Psi_1^+(\mathbf{r})\Psi_2^+(\mathbf{r})}} \cdot (\Psi_1^+(\mathbf{r}) - \Psi_2^+(\mathbf{r})) \tag{6.20}$$

$$\phi^{\text{ger}}(\mathbf{r}) \approx \frac{1}{\sqrt{2 + 2\int \Psi_1^+(\mathbf{r})\Psi_2^+(\mathbf{r})}} \cdot (\Psi_1^+(\mathbf{r}) + \Psi_2^+(\mathbf{r})) \tag{6.21}$$

We can summarize all the transformations that we've performed:

$$\{n_1(\mathbf{r}), n_2(\mathbf{r})\} \xrightarrow{\text{Square root}} \{\Psi_1(\mathbf{r}), \Psi_2(\mathbf{r})\} \xrightarrow{\text{Orthogonalize}} \{\Psi_1^+(\mathbf{r}), \Psi_2^+(\mathbf{r})\} \xrightarrow{\text{Construct}} \phi^{\text{ung}}(\mathbf{r})$$

$$\{n_1(\mathbf{r}), n_2(\mathbf{r})\} \xrightarrow{\text{Square root}} \{\Psi_1(\mathbf{r}), \Psi_2(\mathbf{r})\} \xrightarrow{\text{Orthogonalize}} \{\Psi_1^+(\mathbf{r}), \Psi_2^+(\mathbf{r})\} \xrightarrow{\text{Construct}} \phi^{\text{ger}}(\mathbf{r})$$

Notice that we have performed a multitude of composite functions. This means that we will have to proceed with a chain rule of death. Let us begin by redefining the following variables to make our life a bit more bearable.

$$cg = \sqrt{2 + 2\int d\mathbf{x}\ \Psi_\alpha^+(\mathbf{x})\ \Psi_\beta^+(\mathbf{x})} \quad \& \quad cu = \sqrt{2 - 2\int d\mathbf{x}\ \Psi_\alpha^+(\mathbf{x})\ \Psi_\beta^+(\mathbf{x})} \tag{6.22}$$

$$G(\mathbf{x}) = \frac{1}{cg}\left\{\Psi_\alpha^+(\mathbf{x}) + \Psi_\beta^+(\mathbf{x})\right\} \quad \& \quad U(\mathbf{x}) = \frac{1}{ug}\left\{\Psi_\alpha^+(\mathbf{x}) - \Psi_\beta^+(\mathbf{x})\right\} \tag{6.23}$$

$$\alpha(\mathbf{x}) = n_\alpha(\mathbf{r}) \quad \& \quad \beta(\mathbf{x}) = n_\beta(\mathbf{r}) \tag{6.24}$$

we need to calculate the following colored 8 distinct functional derivatives:

$$\frac{\delta T^G}{\delta\alpha(\mathbf{x})} = -\int d\mathbf{y}\left\{\nabla^2 G(\mathbf{y})\right\}\cdot\int d\mathbf{z}\left\{\frac{\delta G(\mathbf{y})}{\delta\Psi^+_\alpha(\mathbf{z})}\cdot\frac{\delta\Psi^+_\alpha(\mathbf{z})}{\delta\alpha(\mathbf{x})}+\frac{\delta G(\mathbf{y})}{\delta\Psi^+_\beta(\mathbf{z})}\cdot\frac{\delta\Psi^+_\beta(\mathbf{z})}{\delta\alpha(\mathbf{x})}\right\} \tag{6.25}$$

$$\frac{\delta T^G}{\delta\beta(\mathbf{x})} = -\int d\mathbf{y}\left\{\nabla^2 G(\mathbf{y})\right\}\cdot\int d\mathbf{z}\left\{\frac{\delta G(\mathbf{y})}{\delta\Psi^+_\alpha(\mathbf{z})}\cdot\frac{\delta\Psi^+_\alpha(\mathbf{z})}{\delta\beta(\mathbf{x})}+\frac{\delta G(\mathbf{y})}{\delta\Psi^+_\beta(\mathbf{z})}\cdot\frac{\delta\Psi^+_\beta(\mathbf{z})}{\delta\beta(\mathbf{x})}\right\} \tag{6.26}$$

$$\frac{\delta T^U}{\delta\alpha(\mathbf{x})} = -\int d\mathbf{y}\left\{\nabla^2 U(\mathbf{y})\right\}\cdot\int d\mathbf{z}\left\{\frac{\delta U(\mathbf{y})}{\delta\Psi^+_\alpha(\mathbf{z})}\cdot\frac{\delta\Psi^+_\alpha(\mathbf{z})}{\delta\alpha(\mathbf{x})}+\frac{\delta U(\mathbf{y})}{\delta\Psi^+_\beta(\mathbf{z})}\cdot\frac{\delta\Psi^+_\beta(\mathbf{z})}{\delta\alpha(\mathbf{x})}\right\} \tag{6.27}$$

$$\frac{\delta T^U}{\delta\beta(\mathbf{x})} = -\int d\mathbf{y}\left\{\nabla^2 U(\mathbf{y})\right\}\cdot\int d\mathbf{z}\left\{\frac{\delta U(\mathbf{y})}{\delta\Psi^+_\alpha(\mathbf{z})}\cdot\frac{\delta\Psi^+_\alpha(\mathbf{z})}{\delta\beta(\mathbf{x})}+\frac{\delta U(\mathbf{y})}{\delta\Psi^+_\beta(\mathbf{z})}\cdot\frac{\delta\Psi^+_\beta(\mathbf{z})}{\delta\beta(\mathbf{x})}\right\} \tag{6.28}$$

Let us begin with 4 of the easier terms

$$\begin{aligned}\frac{\delta G(\mathbf{y})}{\delta\Psi^+_\alpha(\mathbf{z})} &= \frac{1}{cg^2}\left\{\frac{\delta\Psi^+_\alpha(\mathbf{y})}{\delta\Psi^+_\alpha(\mathbf{z})}-(\Psi^+_\alpha(\mathbf{y})+\Psi^+_\beta(\mathbf{y}))\frac{\delta\ cg}{\delta\Psi^+_\alpha(\mathbf{z})}\right\}\\ &= \frac{1}{cg^2}\left\{cg\ \delta(\mathbf{z}-\mathbf{y})-(\Psi^+_\alpha(\mathbf{y})+\Psi^+_\beta(\mathbf{y}))\frac{1}{2\cdot cg}(2\Psi^+_\beta(\mathbf{z}))\right\}\\ &= \frac{1}{cg^2}\left\{cg\ \delta(\mathbf{z}-\mathbf{y})-G(\mathbf{y})\Psi^+_\beta(\mathbf{z})\right\}\end{aligned} \tag{6.29}$$

In the same way, we can obtain the additional 3 terms:

$$\frac{\delta G(\mathbf{y})}{\delta\Psi^+_\beta(\mathbf{z})} = \frac{1}{cg^2}\left\{cg\ \delta(\mathbf{z}-\mathbf{y})-G(\mathbf{y})\Psi^+_\alpha(\mathbf{z})\right\} \tag{6.30}$$

$$\frac{\delta U(\mathbf{y})}{\delta\Psi^+_\alpha(\mathbf{z})} = \frac{1}{ug^2}\left\{ug\ \delta(\mathbf{z}-\mathbf{y})+U(\mathbf{y})\Psi^+_\beta(\mathbf{z})\right\} \tag{6.31}$$

$$\frac{\delta U(\mathbf{y})}{\delta\Psi^+_\beta(\mathbf{z})} = \frac{1}{ug^2}\left\{ug\ \delta(\mathbf{z}-\mathbf{y})+U(\mathbf{y})\Psi^+_\alpha(\mathbf{z})\right\} \tag{6.32}$$

and proceed with the other 4 terms. We first need to express the $\{\Psi^+_\alpha(\mathbf{r}), \Psi^+_\beta(\mathbf{r})\}$ in terms of the density:

$$\Psi^+_\alpha(\mathbf{z}) = \frac{1}{2}\sqrt{\frac{N}{N_\alpha}}\left\{\frac{\alpha^{\frac{1}{2}}(\mathbf{z})}{1+\xi} + \frac{\alpha^{\frac{1}{2}}(\mathbf{z})}{1-\xi}\right\} + \frac{1}{2}\sqrt{\frac{N}{N_\beta}}\left\{\frac{\beta^{\frac{1}{2}}(\mathbf{z})}{1+\xi} - \frac{\beta^{\frac{1}{2}}(\mathbf{z})}{1-\xi}\right\} \tag{6.33}$$

$$\Psi^+_\beta(\mathbf{z}) = \frac{1}{2}\sqrt{\frac{N}{N_\alpha}}\left\{\frac{\alpha^{\frac{1}{2}}(\mathbf{z})}{1+\xi} - \frac{\alpha^{\frac{1}{2}}(\mathbf{z})}{1-\xi}\right\} + \frac{1}{2}\sqrt{\frac{N}{N_\beta}}\left\{\frac{\beta^{\frac{1}{2}}(\mathbf{z})}{1+\xi} + \frac{\beta^{\frac{1}{2}}(\mathbf{z})}{1-\xi}\right\} \tag{6.34}$$

where

$$\xi = \int d\mathbf{r}\ \alpha^{\frac{1}{2}}(\mathbf{r})\ \beta^{\frac{1}{2}}(\mathbf{r}) \tag{6.35}$$

So that we can compute

$$\frac{\delta\Psi^+_\alpha(\mathbf{z})}{\delta\alpha(\mathbf{x})} = AI\cdot\alpha^{-\frac{1}{2}}(\mathbf{z})\delta(\mathbf{z}-\mathbf{x}) - AIIP\cdot\alpha^{\frac{1}{2}}(\mathbf{z})\beta^{\frac{1}{2}}(\mathbf{x})\alpha^{-\frac{1}{2}}(\mathbf{x}) - BII\cdot\beta^{\frac{1}{2}}(\mathbf{z})\beta^{\frac{1}{2}}(\mathbf{x})\alpha^{-\frac{1}{2}}(\mathbf{x}) \tag{6.36}$$

$$\frac{\delta\Psi^+_\beta(\mathbf{z})}{\delta\beta(\mathbf{x})} = BI\cdot\beta^{-\frac{1}{2}}(\mathbf{z})\delta(\mathbf{z}-\mathbf{x}) - BIIP\cdot\beta^{\frac{1}{2}}(\mathbf{z})\alpha^{\frac{1}{2}}(\mathbf{x})\beta^{-\frac{1}{2}}(\mathbf{x}) - AII\cdot\alpha^{\frac{1}{2}}(\mathbf{z})\alpha^{\frac{1}{2}}(\mathbf{x})\beta^{-\frac{1}{2}}(\mathbf{x}) \tag{6.37}$$

$$\frac{\delta\Psi^+_\alpha(\mathbf{z})}{\delta\beta(\mathbf{x})} = BIM\cdot\beta^{-\frac{1}{2}}(\mathbf{z})\delta(\mathbf{z}-\mathbf{x}) - BII\cdot\beta^{\frac{1}{2}}(\mathbf{z})\alpha^{\frac{1}{2}}(\mathbf{x})\beta^{-\frac{1}{2}}(\mathbf{x}) + AIIP\cdot\alpha^{\frac{1}{2}}(\mathbf{z})\alpha^{\frac{1}{2}}(\mathbf{x})\beta^{-\frac{1}{2}}(\mathbf{x}) \tag{6.38}$$

$$\frac{\delta\Psi^+_\beta(\mathbf{z})}{\delta\alpha(\mathbf{x})} = AIM\cdot\alpha^{-\frac{1}{2}}(\mathbf{z})\delta(\mathbf{z}-\mathbf{x}) - AII\cdot\alpha^{\frac{1}{2}}(\mathbf{z})\beta^{\frac{1}{2}}(\mathbf{x})\alpha^{-\frac{1}{2}}(\mathbf{x}) + BIIP\cdot\beta^{\frac{1}{2}}(\mathbf{z})\beta^{\frac{1}{2}}(\mathbf{x})\alpha^{-\frac{1}{2}}(\mathbf{x}) \tag{6.39}$$

where each of the coefficients are a constant that is given by the following rule

$$XYZ = \frac{X}{2(1+\xi)^Y} + \frac{X}{2(1-\xi)^Y} \tag{6.40}$$

- $X \in (A, B)$ refers to the coefficient that accompanies each term. $A = \frac{1}{2}\sqrt{\frac{N}{N_\alpha}}$ and $B = \frac{1}{2}\sqrt{\frac{N}{N_\beta}}$.
- $Y \in (1, 2)$ is the exponent of the denominator and it is represented with the roman numerals (I, II).
- Z is an optional parameter $Z \in \{P, M\}$ and it refers to whether or not there is a negative sign accompanying one of the two terms. If $Z \to P$, then the negative sign is on the

term with $1+\xi$ (Positive). If $Z \to M$, then the negative sign is on the term with $1-\xi$ (Negative).

With this expression, we can proceed to calculate the integrals of Equations 6.28. To express them, we add up the functionals associated to the same density. For simplicity, we are going to omit the coefficients of each of the terms,since they do not provide any insight into the performance of this new approximation.

$$\frac{\delta T_s}{\delta n_1(\mathbf{r})} = -\alpha^{-\frac{1}{2}}(\mathbf{x})\left\{\beta^{\frac{1}{2}}(\mathbf{x}) + \nabla^2 G(x) + \nabla^2 U(x) + \Psi^+_\alpha(\mathbf{x}) + \Psi^+_\beta(\mathbf{x})\right\} \tag{6.41}$$

$$\frac{\delta T_s}{\delta n_2(\mathbf{r})} = -\beta^{-\frac{1}{2}}(\mathbf{x})\left\{\alpha^{\frac{1}{2}}(\mathbf{x}) + \nabla^2 G(x) + \nabla^2 U(x) + \Psi^+_\alpha(\mathbf{x}) + \Psi^+_\beta(\mathbf{x})\right\} \tag{6.42}$$

Let us now observe the performance of the mOA. Figure 6.10 shows the resulting potential for the approximation. The potential not only does not show the same behaviour as the exact kinetic potential as in Figure 6.6, but its value at every point is orders of magnitude greater than the exact one. This potential does not converge to a reasonable solution for any fragment's scf calculation. Let us try to understand why the mOA fails.

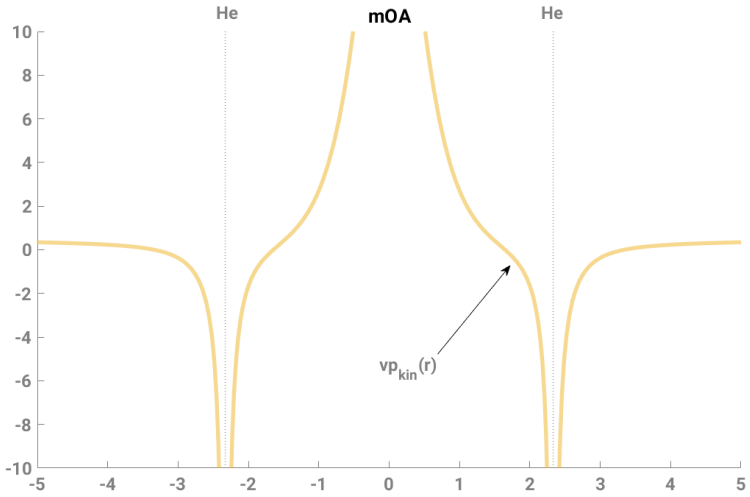


Figure 6.10. v_t^{nad} for the He_2 with the mOA.

6.6 Analysis of the mOA

When we obtain the exact v_t^{nad}, we find a single potential that generates the sum of fragment densities. Nevertheless, here, we associate each orbital to a particular fragment and thus each fragment will have a different potential

$$v_{p,\alpha}^{kin}(\mathbf{r}) = [v_{t,\alpha}[\{n_\alpha(\mathbf{r})\}] - v_t[n_\alpha(\mathbf{r})]] \cdot \frac{n(\mathbf{r})}{n_\alpha(\mathbf{r})} \tag{6.43}$$

Where the factor on the right is the local-Q approximation that acts as soft Heaviside step function that acts as a numerical strategy to ensure that the potential for each fragment converges to the same $v_p(\mathbf{r})$ [99]. Focus on the first term inside the brackets. This is the potential that we obtain from the functional derivative of the 20A and the mOA. Because of this distinction, we cannot simply compare the potential obtained through inversion and the set of potentials obtained for the mOA. On the other hand, we can at least compare the mOA and the 2OA and see if there is any insight from it. Since the He_2 is symmetric, we find that the potentials obtained for the fragment with $n_\alpha(\mathbf{r})$ and the fragment $n_\beta(\mathbf{r})$ are mirrored versions of each other. Thus we only need to analyze one of the fragments to understand their behaviour.

Figure 6.11 shows the two components that make up the potential associated to $n_\alpha(\mathbf{r})$. We can appreciate that around the left He we have a very large peak for both functions. The features around the first Helium contribute to the v_t^{nad} of fragment α. The features that are in the neighborhood of the right He nuclei, on the other hand, are of no relevance to us since anything that is far from the extent of the left density vanishes due to the local-Q function. One thing that is worth mentioning is that these are not singularities, but values that are way too large for the range of values selected.

Figure 6.12 show the same plots but for the mOA. Let us again ignore the features on the neighborhood of the right Helium atom. The main thing that we should notice is that the values near the left Helium are not positive enough. Remember that v_p^{kin} encompasses the difference between the kinetic energy of the full system minus the kinetic energy of the fragments. The fact that this function near the nuclei is not positive enough implies that

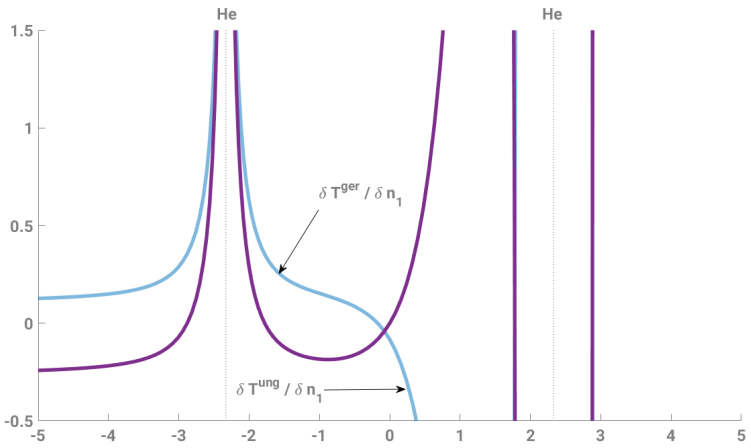


Figure 6.11. $\frac{\delta T^{ger}}{\delta n_\alpha(\mathbf{r})}$ and $\frac{\delta T^{ung}}{\delta n_\alpha(\mathbf{r})}$ derived from the 2OA.

the resulting v_p^{kin} is too negative and results in an nonphysical potential that disrupts the calculation if they are added.

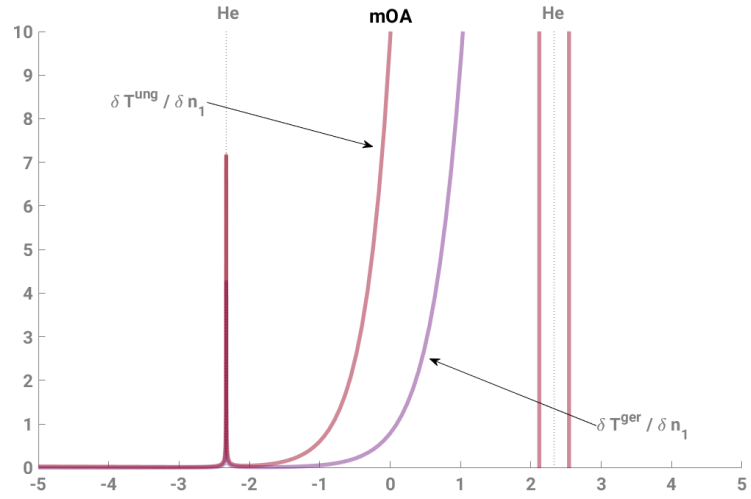


Figure 6.12. $\frac{\delta T^{ger}}{\delta n_\alpha(\mathbf{r})}$ and $\frac{\delta T^{ung}}{\delta n_\alpha(\mathbf{r})}$ derived from the mOA.

6.7 Conclusion

We have proven that the original 2OA was applicable to weakly interacting systems. We showed that generalizing the 2OA to circumvent the need for the full molecular density, as well as to accurately be applicable in heteronuclear systems is it just as challenging as any other kinetic approximation. Work in this path is ongoing, and it is still our believe that approximating orbitals is a more profitable path that approximating the full integrand for the kinetic energy. More improvements in this area seem to remain incredibly challenging.

REFERENCES

[1] M. Born and R. Oppenheimer, "Zur quantentheorie der molekeln," Annalen der physik, vol. 389, no. 20, pp. 457–484, 1927.

[2] E. Schrödinger, "An undulatory theory of the mechanics of atoms and molecules," Physical review, vol. 28, no. 6, p. 1049, 1926.

[3] P. A. M. Dirac, "A new notation for quantum mechanics," in Mathematical Proceedings of the Cambridge Philosophical Society, Cambridge University Press, vol. 35, 1939, pp. 416–418.

[4] A. Szabo and N. S. Ostlund, Modern quantum chemistry: introduction to advanced electronic structure theory. Courier Corporation, 2012.

[5] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," Physical review, vol. 136, no. 3B, B864, 1964.

[6] W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects," Physical review, vol. 140, no. 4A, A1133, 1965.

[7] M. Levy, "Electron densities in search of hamiltonians," Physical Review A, vol. 26, no. 3, p. 1200, 1982.

[8] M. Levy and S. Vuckovic, "The adiabatic connection formula for the exchange–correlation functional," in. Dec. 2017.

[9] V. N. Staroverov, "Density-functional approximations for exchange and correlation," A Matter of Density, pp. 125–156, 2012.

[10] J. Tao, J. P. Perdew, V. N. Staroverov, and G. E. Scuseria, "Climbing the density functional ladder: Nonempirical meta–generalized gradient approximation designed for molecules and solids," Physical Review Letters, vol. 91, no. 14, p. 146 401, 2003.

[11] J. M. Turney, A. C. Simmonett, R. M. Parrish, E. G. Hohenstein, F. A. Evangelista, J. T. Fermann, B. J. Mintz, L. A. Burns, J. J. Wilke, M. L. Abrams, et al., "Psi4: An open-source ab initio electronic structure program," Wiley Interdisciplinary Reviews: Computational Molecular Science, vol. 2, no. 4, pp. 556–565, 2012.

[12] D. G. Smith, L. A. Burns, D. A. Sirianni, D. R. Nascimento, A. Kumar, A. M. James, J. B. Schriber, T. Zhang, B. Zhang, A. S. Abbott, et al., "Psi4numpy: An interactive quantum chemistry programming environment for reference implementations and rapid development," Journal of chemical theory and computation, vol. 14, no. 7, pp. 3504–3511, 2018.

[13] C. R. Jacob and J. Neugebauer, "Subsystem density-functional theory," Wiley Interdisciplinary Reviews-Computational Molecular Science, vol. 4, no. 4, pp. 325–362, 2014, ISSN: 1759-0876. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:000337751100002.

[14] Q. N. Sun and G. K. L. Chan, "Quantum embedding theories," Acc. Chem. Res., vol. 49, no. 12, pp. 2705–2712, 2016, ISSN: 0001-4842. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:000390619500005.

[15] J. Nafziger and A. Wasserman, "Density-based partitioning methods for ground-state molecular calculations," The Journal of Physical Chemistry A, vol. 118, no. 36, pp. 7623–7639, 2014.

[16] S. J. Lee, M. Welborn, F. R. Manby, and T. F. Miller III, "Projection-based wavefunction-in-dft embedding," Accounts of chemical research, vol. 52, no. 5, pp. 1359–1368, 2019.

[17] A. Pribram-Jones, D. A. Gross, and K. Burke, "DFT: A Theory Full of Holes?" Annu. Rev. Phys. Chem., vol. 66, no. 1, pp. 283–304, 2015. DOI: 10.1146/annurev-physchem-040214-121420. (visited on 02/26/2016).

[18] A. Wasserman, J. Nafziger, K. L. Jiang, M. C. Kim, E. Sim, and K. Burke, "The importance of being inconsistent," Annu. Rev. Phys. Chem., vol. 68, pp. 555–581, 2017, ISSN: 0066-426x. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:000401335600025.

[19] A. J. Cohen, P. Mori-Sánchez, and W. Yang, "Insights into current limitations of density functional theory," Science, vol. 321, no. 5890, pp. 792–794, 2008.

[20] T. Schmidt, E. Kraisler, A. Makmal, L. Kronik, and S. Kümmel, "A self-interaction-free local hybrid functional: Accurate binding energies vis-à-vis accurate ionization potentials from Kohn-Sham eigenvalues," J. Chem. Phys., vol. 140, no. 18, 18A510, 2014. (visited on 03/22/2016).

[21] D. N. Komsa and V. N. Staroverov, "Elimination of spurious fractional charges in dissociating molecules by correcting the shape of approximate kohn–sham potentials," Journal of Chemical Theory and Computation, vol. 12, no. 11, pp. 5361–5366, 2016.

[22] J. Tao, J. P. Perdew, V. N. Staroverov, and G. E. Scuseria, "Climbing the density functional ladder: Nonempirical meta–generalized gradient approximation designed for molecules and solids," Physical Review Letters, vol. 91, no. 14, p. 146 401, 2003.

[23] J. Nafziger, "Partition density functional theory," English, Ph.D thesis, Purdue University, 2015. [Online]. Available: https://search.proquest.com/docview/1718546928?accountid=13360.

[24] M. H. Cohen and A. Wasserman, "On Hardness and Electronegativity Equalization in Chemical Reactivity Theory," en, J. Stat. Phys., vol. 125, no. 5-6, pp. 1121–1139, 2006, ISSN: 0022-4715, 1572-9613. DOI: 10.1007/s10955-006-9031-0. (visited on 03/01/2016).

[25] P. Elliott, K. Burke, M. H. Cohen, and A. Wasserman, "Partition density-functional theory," Phys. Rev. A, vol. 82, no. 2, p. 024 501, 2010. DOI: 10.1103/PhysRevA.82.024501. (visited on 11/02/2015).

[26] J. Nafziger and A. Wasserman, "Fragment-based treatment of delocalization and static correlation errors in density-functional theory," The Journal of chemical physics, vol. 143, no. 23, p. 234 105, 2015.

[27] J. Sun, A. Ruzsinszky, and J. P. Perdew, "Strongly constrained and appropriately normed semilocal density functional," Phys. Rev. Lett., vol. 115, no. 3, p. 036 402, 2015.

[28] Y. Zhang, D. A. Kitchaev, J. Yang, T. Chen, S. T. Dacek, R. A. Sarmiento-Pérez, M. A. Marques, H. Peng, G. Ceder, J. P. Perdew, et al., "Efficient first-principles prediction of solid stability: Towards chemical accuracy," NPJ Comput. Mater., vol. 4, no. 1, p. 9, 2018.

[29] N. Mardirossian and M. Head-Gordon, "Thirty years of density functional theory in computational chemistry: An overview and extensive assessment of 200 density functionals," Mol. Phys., vol. 115, no. 19, pp. 2315–2372, 2017, ISSN: 0026-8976. DOI: 10.1080/00268976.2017.1333614. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:000416508000001.

[30] G. Hegde and R. C. Bowen, "Machine-learned approximations to density functional theory hamiltonians," Sci. Rep., vol. 7, p. 42 669, 2017, ISSN: 2045-2322. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:000394253200001.

[31] H. Y. S. Yu, S. H. L. Li, and D. G. Truhlar, "Perspective: Kohn-sham density functional theory descending a staircase," J. Chem. Phys., vol. 145, no. 13, p. 130 901, 2016, ISSN: 0021-9606. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:000386156200001.

[32] M. Chen, X. W. Jiang, H. L. Zhuang, L. W. Wang, and E. A. Carter, "Petascale orbital-free density functional theory enabled by small-box algorithms," J. Chem. Theory Comput., vol. 12, no. 6, pp. 2950–2963, 2016, ISSN: 1549-9618. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:000378016000042.

[33] K. Jiang, J. Nafziger, and A. Wasserman, "Constructing a non-additive non-interacting kinetic energy functional approximation for covalent bonds from exact conditions," The Journal of chemical physics, vol. 149, no. 16, p. 164 112, 2018.

[34] C. F. v. Weizsäcker, "Zur theorie der kernmassen," Zeitschrift für Physik A Hadrons and Nuclei, vol. 96, pp. 431–458, 7 1935, ISSN: 0939-7922. DOI: 10.1007/BF01337700.

[35] L. H. Thomas, "The calculation of atomic fields.," P. Camb. Philos. Soc., vol. 23, pp. 542–548, 1927, ISSN: 0008-1981. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:000200163300064.

[36] E. Fermi, "Statistical method to determine some properties of atoms," Rend. Accad. Naz. Lincei, vol. 6, no. 602-607, p. 5, 1927.

[37] J. P. Perdew, R. G. Parr, M. Levy, and J. L. Balduz, "Density-functional theory for fractional particle number: Derivative discontinuities of the energy," Phys. Rev. Lett., vol. 49, pp. 1691–1694, 23 1982. DOI: 10.1103/PhysRevLett.49.1691. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.49.1691.

[38] R. G. Parr and W. T. Yang, "Density functional-approach to the frontier-electron theory of chemical-reactivity," J. Am. Chem. Soc., vol. 106, no. 14, pp. 4049–4050, 1984, ISSN: 0002-7863. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:A1984SZ32800036.

[39] R. G. Parr, R. A. Donnelly, M. Levy, and W. E. Palke, "Electronegativity - density functional viewpoint," J. Chem. Phys., vol. 68, no. 8, pp. 3801–3807, 1978, ISSN: 0021-9606. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:A1978EZ48000060.

[40] M. S. Gordon, D. G. Fedorov, S. R. Pruitt, and L. V. Slipchenko, "Fragmentation Methods: A Route to Accurate Calculations on Large Systems," Chem. Rev., vol. 112, no. 1, pp. 632–672, 2012, ISSN: 0009-2665. DOI: 10.1021/cr200093j. (visited on 02/26/2016).

[41] C. R. Jacob and J. Neugebauer, "Subsystem density-functional theory," Wiley Interdisciplinary Reviews-Computational Molecular Science, vol. 4, no. 4, pp. 325–362, 2014, ISSN: 1759-0876. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:000337751100002.

[42] Q. N. Sun and G. K. L. Chan, "Quantum embedding theories," Acc. Chem. Res., vol. 49, no. 12, pp. 2705–2712, 2016, ISSN: 0001-4842. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:000390619500005.

[43] S. J. R. Lee, M. Welborn, F. R. Manby, and T. F. Miller, "Projection-based wavefunction-in-dft embedding," Accounts of Chemical Research, vol. 52, no. 5, pp. 1359–1368, 2019, ISSN: 0001-4842. DOI: 10.1021/acs.accounts.8b00672. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:000469304100023.

[44] A. Seidl, A. Gorling, P. Vogl, J. A. Majewski, and M. Levy, "Generalized kohn-sham schemes and the band-gap problem," Phys. Rev. B, vol. 53, no. 7, pp. 3764–3774, 1996, ISSN: 1098-0121. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:A1996TW98100021.

[45] S. Kummel and L. Kronik, "Orbital-dependent density functionals: Theory and applications," Rev. Mod. Phys., vol. 80, no. 1, pp. 3–60, 2008, ISSN: 0034-6861. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:000252119100002.

[46] J. C. Snyder, M. Rupp, K. Hansen, K. R. Muller, and K. Burke, "Finding density functionals with machine learning," Phys. Rev. Lett., vol. 108, no. 25, p. 253 002, 2012, ISSN: 0031-9007. DOI: ARTN253002DOI10.1103/PhysRevLett.108.253002. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:000305568700008.

[47] J. Seino, R. Kageyama, M. Fujinami, Y. Ikabata, and H. Nakai, "Semi-local machine-learned kinetic energy density functional with third-order gradients of electron density," J. Chem. Phys., vol. 148, no. 24, p. 241 705, 2018, ISSN: 0021-9606. [Online]. Available: %3CGo%20to%20ISI%3E://WOS:000437190300008.

[48] R. Van Leeuwen and E. Baerends, "Exchange-correlation potential with correct asymptotic behavior," Physical Review A, vol. 49, no. 4, p. 2421, 1994.

[49] O. V. Gritsenko, R. van Leeuwen, and E. J. Baerends, "Molecular kohn-sham exchange-correlation potential from the correlated ab initio electron density," Physical Review A, vol. 52, no. 3, p. 1870, 1995.

[50] J. D. Goodpaster, N. Ananth, F. R. Manby, and T. F. Miller III, "Exact nonadditive kinetic potentials for embedded density functional theory," The Journal of chemical physics, vol. 133, no. 8, p. 084 103, 2010.

[51] A. A. Kananenka, S. V. Kohut, A. P. Gaiduk, I. G. Ryabinkin, and V. N. Staroverov, "Efficient construction of exchange and correlation potentials by inverting the kohn–sham equations," The Journal of chemical physics, vol. 139, no. 7, p. 074 112, 2013.

[52] P. Schipper, O. Gritsenko, and E. Baerends, "Kohn-sham potentials corresponding to slater and gaussian basis set densities," Theoretical Chemistry Accounts, vol. 98, no. 1, pp. 16–24, 1997.

[53] A. P. Gaiduk, I. G. Ryabinkin, and V. N. Staroverov, "Removal of basis-set artifacts in kohn–sham potentials recovered from electron densities," Journal of chemical theory and computation, vol. 9, no. 9, pp. 3959–3964, 2013.

[54] Q. Zhao, R. C. Morrison, and R. G. Parr, "From electron densities to kohn-sham kinetic energies, orbital energies, exchange-correlation potentials, and exchange-correlation energies," Physical Review A, vol. 50, no. 3, p. 2138, 1994.

[55] A. Kumar and M. K. Harbola, "A general penalty method for density-to-potential inversion," International Journal of Quantum Chemistry, vol. 120, no. 22, e26400, 2020.

[56] P. W. Ayers*, R. C. Morrison, and R. G. Parr, "Fermi-amaldi model for exchange-correlation: Atomic excitation energies from orbital energy differences," Molecular Physics, vol. 103, no. 15-16, pp. 2061–2072, 2005.

[57] R. C. Morrison and Q. Zhao, "Solution to the kohn-sham equations using reference densities from accurate, correlated wave functions for the neutral atoms helium through argon," Physical Review A, vol. 51, no. 3, p. 1980, 1995.

[58] Q. Wu and W. Yang, "A direct optimization method for calculating density functionals and exchange–correlation potentials from electron densities," The Journal of chemical physics, vol. 118, no. 6, pp. 2498–2509, 2003.

[59] Y. Shi and A. Wasserman, "Inverse kohn–sham density functional theory: Progress and challenges," The Journal of Physical Chemistry Letters, vol. 12, pp. 5308–5318, 2021.

[60] D. S. Jensen and A. Wasserman, "Numerical methods for the inverse problem of density functional theory," International Journal of Quantum Chemistry, vol. 118, no. 1, e25425, 2018.

[61] B. Kanungo, P. M. Zimmerman, and V. Gavini, "Exact exchange-correlation potentials from ground-state electron densities," Nature communications, vol. 10, no. 1, pp. 1–9, 2019.

[62] J. Nafziger, K. Jiang, and A. Wasserman, "Accurate reference data for the nonadditive, noninteracting kinetic energy in covalent bonds," Journal of chemical theory and computation, vol. 13, no. 2, pp. 577–586, 2017.

[63] I. G. Ryabinkin, S. V. Kohut, and V. N. Staroverov, "Reduction of electronic wave functions to kohn-sham effective potentials," Physical review letters, vol. 115, no. 8, p. 083 001, 2015.

[64] E. Ospadov, I. G. Ryabinkin, and V. N. Staroverov, "Improved method for generating exchange-correlation potentials from electronic wave functions," The Journal of chemical physics, vol. 146, no. 8, p. 084 103, 2017.

[65] Q. Ou and E. A. Carter, "Potential functional embedding theory with an improved kohn–sham inversion algorithm," Journal of Chemical Theory and Computation, vol. 14, no. 11, pp. 5680–5689, 2018.

[66] A. P. Gaiduk, I. G. Ryabinkin, and V. N. Staroverov, "Removal of basis-set artifacts in kohn–sham potentials recovered from electron densities," Journal of chemical theory and computation, vol. 9, no. 9, pp. 3959–3964, 2013.

[67] A. Kumar, R. Singh, and M. K. Harbola, "Accurate effective potential for density amplitude and the correspondingkohn-sham exchange-correlation potential calculated from approximatewavefunctions," Journal of Physics B: Atomic, Molecular and Optical Physics, 2020.

[68] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, et al., "Array programming with numpy," Nature, vol. 585, no. 7825, pp. 357–362, 2020.

[69] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al., "Scipy 1.0: Fundamental algorithms for scientific computing in python," Nature methods, vol. 17, no. 3, pp. 261–272, 2020.

[70] G. Daniel, J. Gray, et al., "Opt_einsum-a python package for optimizing contraction order for einsum-like expressions," Journal of Open Source Software, vol. 3, no. 26, p. 753, 2018.

[71] S. Lehtola, C. Steigemann, M. J. Oliveira, and M. A. Marques, "Recent developments in libxc—a comprehensive library of functionals for density functional theory," SoftwareX, vol. 7, pp. 1–5, 2018.

[72] J. D. Hunter, "Matplotlib: A 2d graphics environment," IEEE Annals of the History of Computing, vol. 9, no. 03, pp. 90–95, 2007.

[73] R. Lehmann, "The sphinx project," Universität Potsdam, Project Documentation, 2011.

[74] J. Nash, D. Altarawy, T. Barnes, S. Ellis, E. Marin Rimoldi, B. Pritchard, and D. Smith, Best practices in python package development, en, 2021/03/10/15:38:40 2018. [Online]. Available: http://education.molssi.org/python-package-best-practices/.

[75] A. Krylov, T. L. Windus, T. Barnes, E. Marin-Rimoldi, J. A. Nash, B. Pritchard, D. G. Smith, D. Altarawy, P. Saxe, C. Clementi, et al., "Perspective: Computational chemistry software and its advancement as illustrated through three grand challenge cases for molecular science," The Journal of chemical physics, vol. 149, no. 18, p. 180 901, 2018.

[76] T. A. Barnes, E. Marin-Rimoldi, S. Ellis, and T. D. Crawford, "The molssi driver interface project: A framework for standardized, on-the-fly interoperability between computational molecular sciences codes," Computer Physics Communications, vol. 261, p. 107 688, 2021.

[77] D. G. Smith, A. T. Lolinco, Z. L. Glick, J. Lee, A. Alenaizan, T. A. Barnes, C. H. Borca, R. Di Remigio, D. L. Dotson, S. Ehlert, et al., "Quantum chemistry common driver and databases (qcdb) and quantum chemistry engine (qcengine): Automation and interoperability among computational chemistry programs," The Journal of Chemical Physics, vol. 155, no. 20, p. 204 801, 2021.

[78] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, et al., "The atomic simulation environment—a python library for working with atoms," Journal of Physics: Condensed Matter, vol. 29, no. 27, p. 273 002, 2017.

[79] X. Shao, K. Jiang, W. Mi, A. Genova, and M. Pavanello, "Dftpy: An efficient and object-oriented platform for orbital-free dft simulations," Wiley Interdisciplinary Reviews: Computational Molecular Science, vol. 11, no. 1, e1482, 2021.

[80] T. Verstraelen, W. Adams, L. Pujal, A. Tehrani, B. D. Kelly, L. Macaya, F. Meng, M. Richer, R. Hernández-Esparza, X. D. Yang, et al., "Iodata: A python library for reading, writing, and converting computational chemistry file formats and generating input files," Journal of Computational Chemistry, vol. 42, no. 6, pp. 458–464, 2021.

[81] C. H. Borca, B. W. Bakr, L. A. Burns, and C. D. Sherrill, "Crystalatte: Automated computation of lattice energies of organic crystals exploiting the many-body expansion to achieve dual-level parallelism," The Journal of Chemical Physics, vol. 151, no. 14, p. 144 103, 2019.

[82] D. Poole, J. L. Galvez Vallejo, and M. S. Gordon, "A new kid on the block: Application of julia to hartree–fock calculations," Journal of Chemical Theory and Computation, vol. 16, no. 8, pp. 5006–5013, 2020.

[83] M. F. Herbst, A. Levitt, and E. Cancès, "Dftk: A julian approach for simulating electrons in solids," in Proceedings of the JuliaCon Conferences, vol. 3, 2021, p. 69.

[84] D. B. Magers, V. H. Chávez, B. G. Peyton, D. A. Sirianni, R. C. Fortenberry, and A. Ringer McDonald, "Psi4education: Free and open-source programing activities for chemical education with free and open-source software," in Teaching Programming across the Chemistry Curriculum, ACS Publications, 2021, pp. 107–122.

[85] M. Govoni, J. Whitmer, J. de Pablo, F. Gygi, and G. Galli, "Code interoperability extends the scope of quantum simulations," npj Computational Materials, vol. 7, no. 1, pp. 1–10, 2021.

[86] J. M. Turney, A. C. Simmonett, R. M. Parrish, E. G. Hohenstein, F. A. Evangelista, J. T. Fermann, B. J. Mintz, L. A. Burns, J. J. Wilke, M. L. Abrams, et al., "Psi4: An open-source ab initio electronic structure program," Wiley Interdisciplinary Reviews: Computational Molecular Science, vol. 2, no. 4, pp. 556–565, 2012.

[87] R. M. Parrish, L. A. Burns, D. G. Smith, A. C. Simmonett, A. E. DePrince III, E. G. Hohenstein, U. Bozkaya, A. Y. Sokolov, R. Di Remigio, R. M. Richard, et al., "Psi4 1.1: An open-source electronic structure program emphasizing automation, advanced libraries, and interoperability," Journal of chemical theory and computation, vol. 13, no. 7, pp. 3185–3197, 2017.

[88] Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, S. Sharma, et al., "Pyscf: The python-based simulations of chemistry framework," Wiley Interdisciplinary Reviews: Computational Molecular Science, vol. 8, no. 1, e1340, 2018.

[89] Q. Sun, X. Zhang, S. Banerjee, P. Bao, M. Barbry, N. S. Blunt, N. A. Bogdanov, G. H. Booth, J. Chen, Z.-H. Cui, et al., "Recent developments in the pyscf program package," The Journal of chemical physics, vol. 153, no. 2, p. 024 109, 2020.

[90] A. Szabo and N. S. Ostlund, Modern quantum chemistry: introduction to advanced electronic structure theory. Courier Corporation, 2012.

[91] J. G. Hill, "Gaussian basis sets for molecular applications," International Journal of Quantum Chemistry, vol. 113, no. 1, pp. 21–34, 2013.

[92] X. Andrade, D. Strubbe, U. De Giovannini, A. H. Larsen, M. J. Oliveira, J. Alberdi-Rodriguez, A. Varas, I. Theophilou, N. Helbig, M. J. Verstraete, et al., "Real-space grids and the octopus code as tools for the development of new simulation approaches for electronic systems," Physical Chemistry Chemical Physics, vol. 17, no. 47, pp. 31 371–31 396, 2015.

[93] I. G. Ryabinkin, E. Ospadov, and V. N. Staroverov, "Exact exchange-correlation potentials of singlet two-electron systems," The Journal of Chemical Physics, vol. 147, no. 16, p. 164 117, 2017.

[94] G. B. Arfken and H. J. Weber, Mathematical methods for physicists, 1999.

[95] J. Kobus, L. Laaksonen, and D. Sundholm, "A numerical hartree-fock program for diatomic molecules," Computer physics communications, vol. 98, no. 3, pp. 346–358, 1996.

[96] M. A. Marques, M. J. Oliveira, and T. Burnus, "Libxc: A library of exchange and correlation functionals for density functional theory," Computer physics communications, vol. 183, no. 10, pp. 2272–2281, 2012.

[97] V. V. Karasiev, R. S. Jones, S. B. Trickey, and F. E. Harris, "Recent advances in developing orbital-free kinetic energy functionals," New developments in quantum chemistry, pp. 25–54, 2009.

[98] P.-O. Löwdin, "On the non-orthogonality problem connected with the use of atomic wave functions in the theory of molecules and crystals," The Journal of Chemical Physics, vol. 18, no. 3, pp. 365–375, 1950.

[99] M. A. Mosquera and A. Wasserman, "Partition density functional theory and its extension to the spin-polarized case," Molecular Physics, vol. 111, no. 4, pp. 505–515, 2013.

[100] Z. Romanowski and S. Krukowski, "Derivation of von weizsäcker equation based on green-gauss theorem," Acta Phys. Pol. A, vol. 115, pp. 653–655, 2009.

A. CRASH COURSE IN FUNCTIONAL DERIVATIVES

We briefly review some aspects of functionals and its derivatives. These are thoroughly used in Chapter 6. A functional F is a mapping that takes a function and returns a number, i.e. $F[f] \to \mathscr{C}, \quad \mathscr{C} \in \mathscr{R}$. The differential of a functional is given by the difference of a functional and the same functional with an infinitesimal variation of the function:

$$\delta F[f] = F[f + \delta f] = F[f] \tag{A.1}$$

This variation $\delta F[f]$ is linear in $\delta f(x)$ at any point of its domain

$$F[f] = \int \frac{\delta F[f]}{\delta f(x)} \delta f(x) dx \tag{A.2}$$

$\delta F[f]/\delta f(x)$ defines the functional derivative of $F[f]$ with respect to $f(x)$.

In practice, many of the properties of the derivatives in $f : \mathscr{R} \to \mathscr{R}$ translate to the functional derivatives. Let F and G be two functionals of f, and $c_1, c_2 \in \mathscr{R}$, then

$$\frac{\delta}{\delta f(x)} \Big\{ c_1 F[f] + c_2 G[f] \Big\} = c_1 \frac{\delta F[f]}{\delta f(x)} + c_2 \frac{\delta G[f]}{\delta f(x)} \tag{A.3}$$

$$\frac{\delta}{\delta f(x)} \Big\{ F[f] \ * \ G[f] \Big\} = \frac{\delta F[f]}{\delta f(x)} G[f] + F[f] \frac{\delta G[f]}{\delta f(x)} \tag{A.4}$$

$$\frac{\delta}{\delta f(x)} \Big\{ F[f] \ / \ G[f] \Big\} = \frac{1}{G^2[f]} \Big\{ G[f] \frac{\delta F[f]}{\delta f(x)} - F[f] \frac{\delta G[f]}{\delta f(x)} \Big\} \tag{A.5}$$

Let us now assume that a functional $F[f]$ is a functional of $f[g](x)$ which is in itself also a functional of the function $g(x)$, then the functional derivative $\frac{\delta F[f]}{\delta g(x)}$ is found through the analogous of the chain rule

$$\frac{\delta F[f]}{\delta g(x)} = \int \frac{\delta F[f]}{\delta f(y)} \cdot \frac{\delta f(y)}{\delta g(x)} dy \tag{A.6}$$

We may be interested in computing the functional derivative of $F[f(x)]$ with respect to $f(y)$. In that case

$$\frac{\delta F[f(x)]}{\delta f(y)} = \frac{dF[f]}{df} \cdot \delta(x - y) \tag{A.7}$$

Derivatives of higher orders can also be performed. In addition, there is an analogous of Fubini's theorem for functional derivatives

$$\frac{\delta^2 F[f]}{\delta f(x)\delta f(y)} = \frac{\delta^2 F[f]}{\delta f(y)\delta f(x)} \tag{A.8}$$

Given a functional $F[f]$ that is dependent of f and any arbitrary number (N) of its derivatives, i.e.

$$F[f] = \int f(x, f(x), \frac{d}{dx}f(x), \frac{d^2}{dx^2}f(x), ..., \frac{d^N}{dx^N}f(x)) \tag{A.9}$$

then

$$\frac{\delta F[f]}{\delta f} = \frac{\delta F[f]}{\delta f(x)} - \frac{d}{dx}\cdot\frac{\delta F[f]}{\delta \frac{df}{dx}} + \frac{d^2}{dx^2}\cdot\frac{\delta F[f]}{\delta \frac{d^2 f}{dx^2}} + ... + (-1)^N \frac{d^N F[f]}{dx^N} \tag{A.10}$$

The same applies for a functional $\mathfrak{F}$ that depends on $\mathfrak{f} : \mathscr{R}^3 \to \mathscr{R}$, simply replace the total derivative with the ∇ operator.

B. QUANTUM CHEMISTRY ON A BASIS-SET

Most practical calculations for atomic systems done by expressing the quantities of interest in a basis of M atomic functions $\{\chi_\nu(\mathbf{r})\}$, usually these functions Gaussian-type functions centered around each atom. Each of the orbitals is expanded as:

$$\phi_{\mathrm{i}}(\mathbf{r}) = \sum_{\mu=1}^{M} c_{\nu\mathrm{i}} \cdot \chi_\nu(\mathbf{r}) \tag{B.1}$$

We can plug in the previous expression into the Kohn-Sham equations (2.20) to find

$$h(\mathbf{r})\phi_{\mathrm{i}}(\mathbf{r}) = \varepsilon\phi_{\mathrm{i}}(\mathbf{r}) \tag{B.2}$$

that after multiplying by $\chi^*_\nu(\mathbf{r})$ and integrating over all space becomes

$$\sum_{\nu}^{M} F_{\mu\nu}c_{\nu\mathrm{i}} = \varepsilon_{\mathrm{i}} \sum_{\nu}^{M} S_{\mu\nu}c_{\nu\mathrm{i}} \tag{B.3}$$

where

$$F_{\mu\nu} = \int \chi^*_\nu(\mathbf{r})h(\mathbf{r})\chi_\nu(\mathbf{r})d\mathbf{r} \tag{B.4}$$

$$S_{\mu\nu} = \int \chi^*_\nu(\mathbf{r})\chi_\nu(\mathbf{r})d \tag{B.5}$$

are the elements of the "Fock-like" KS matrix and the elements of the overlap matrix of the basis functions, respectively. The KS matrix is composed of the matrix elements[1]

$$F_{\mu\nu} = h_{\mu\nu} + J_{\mu\nu} + V_{xc,\mu\nu} \tag{B.6}$$

where $h_{\mu\nu}$ is known as the core matrix

$$h_{\mu\nu} = \int \chi^*_\nu(\mathbf{r}) \left\{\frac{1}{2}\nabla^2 + v_{ext}(\mathbf{r})\right\} \chi_\nu(\mathbf{r})d\mathbf{r} \tag{B.7}$$

[1] ↑Here we exclude the exact exchange.

$J_{\mu\nu}$ is the Hartree potential

$$J_{\mu\nu} = \int \chi_{\nu}^{*}(\mathbf{r})\ v_{\mathrm{H}}(\mathbf{r})\ \chi_{\nu}(\mathbf{r}) d\mathbf{r} \tag{B.8}$$

$V_{xc,\mu\nu}$ is the exchange-correlation potential

$$V_{xc,\mu\nu} = \int \chi_{\nu}^{*}(\mathbf{r})\ v_{\mathrm{xc}}(\mathbf{r})\ \chi_{\nu}(\mathbf{r}) d\mathbf{r} \tag{B.9}$$

To express the electronic density in terms of the basis-set we write

$$P_{\mu\nu} = \sum_{\mathrm{i}=1}^{N} c_{\mu\mathrm{i}} \cdot c_{\nu\mathrm{i}}^{*} \tag{B.10}$$

Since the exchange-correlation potential is obtained through a functional derivative of the exchange-correlation energy, the density needs to be available on the grid, to retrieve $n(\mathbf{r})$ from the basis set we use

$$n(\mathbf{r}) = \sum_{\mu=1}^{N} \sum_{\nu=1}^{N} \chi_{\mu}(\mathbf{r})\ P_{\mu\nu}\ \chi_{\mu}^{*}(\mathbf{r}) \tag{B.11}$$

C. DERIVATION OF THE VON WEIZSÄCKER EQUATION

This section is inspired by Reference [100] to which the reader is referred for more details.

The von Weizsäcker model is a kinetic energy functional that is exact for one-electron systems. The derivation is based on the definition of the kinetic energy

$$T = -\frac{1}{2}\int \Psi(\mathbf{r}) \cdot \nabla^2\Psi(\mathbf{r})d\mathbf{r} \tag{C.1}$$

and the electron density is defined as:

$$n(\mathbf{r}) = \Psi^2(\mathbf{r}) \tag{C.2}$$

Let us apply the laplacian on both sides of Equation (C.2)

$$\nabla^2 n(\mathbf{r}) \equiv \nabla^2\Psi(\mathbf{r}) \tag{C.3}$$

$$= 2 \mid \nabla\Psi(\mathbf{r}) \mid^2 + 2\Psi(\mathbf{r})\nabla^2\Psi(\mathbf{r}) \tag{C.4}$$

where the second line comes from the identity from vector calculus

$$\nabla^2(fg) = f\nabla^2 g + 2\nabla f \cdot \nabla g + g\nabla^2 f$$

Let us now integrate Equation (B.4) in all of space and proceed to substitute (C.1)

$$\int n(\mathbf{r})d\mathbf{r} = 2\int \mid \nabla\Psi(\mathbf{r}) \mid^2 d\mathbf{r} - 4T \tag{C.5}$$

and solve for T

$$T = \frac{1}{2}\int \mid \nabla\Psi(\mathbf{r}) \mid^2 d\mathbf{r} - \frac{1}{4}\int \nabla^2 n(\mathbf{r})d\mathbf{r} \tag{C.6}$$

Save that equation for later. In the meantime, let us apply ∇ to Equation (C.2)

$$n(\mathbf{r}) \equiv \nabla\Psi^2(\mathbf{r}) = 2\Psi(\mathbf{r}) \cdot \nabla\Psi(\mathbf{r}) \tag{C.7}$$

Let us square it and substitute Equation (C.2) once more

$$| \nabla n(\mathbf{r}) |^2 = 4 \cdot \Psi^2(\mathbf{r}) | \nabla \Psi(\mathbf{r}) | 2 \tag{C.8}$$

$$= 4 \cdot n(\mathbf{r}) | \nabla \Psi(\mathbf{r}) |^2 \tag{C.9}$$

Solve for the second factor on the right hand side

$$| \nabla \Psi(\mathbf{r}) |^2 = \frac{| \nabla n(\mathbf{r}) |^2}{n(\mathbf{r})} d\mathbf{r} \tag{C.10}$$

As a last step, substitute the last equation into Equation (C.6)

$$T = \frac{1}{8} \int \frac{| \nabla n(\mathbf{r}) |^2}{n(\mathbf{r})} d\mathbf{r} - \frac{1}{4} \int \nabla^2 n(\mathbf{r}) d\mathbf{r} \tag{C.11}$$

From [100], we know that the second term is equal to zero. This is how we find the von Weizsäcker kinetic energy functional

$$T = \frac{1}{8} \int \frac{| \nabla n(\mathbf{r}) |^2}{n(\mathbf{r})} d\mathbf{r} \tag{C.12}$$

D. NEWTON-COTES WEIGHTS FOR INTEGRATION

Consider a function in 1D. Our goal is to compute the integral

$$\int_a^b f(x)dx \tag{D.1}$$

where f is a function that is defined on a discreet space of x_j points. Each of these points is given by:

$$x_\mathrm{j} = a + \mathrm{j}h \tag{D.2}$$

$$h = \frac{b-a}{n} \tag{D.3}$$

$$\mathrm{j} = 0, 1, 2, ...n \tag{D.4}$$

Proceed to compute the Lagrange interpolation polynomial of order n

$$p_n(x) = \sum_{k=0}^{n} L_k(x) f(x_k) \tag{D.5}$$

which we are hoping will mimic our function (this is not always true)

$$p_n(x) \approx f(x), \quad x \in [a, b] \tag{D.6}$$

So that we focus on integrating the polynomials

$$\int_a^b f(x)dx \approx \int_a^b p_n(x)dx = \sum_{k=0}^{n} f(s_k) \int_a^b L_k(x)dx \tag{D.7}$$

that are the weights that we are looking for:

$$w_k = \int_a^b L_k(x)\ dx, \quad k = 0, 1, 2, ...n \tag{D.8}$$

that are computed as

$$w_k = \int_a^b \prod_{\mathrm{j}=0,\mathrm{j}\neq\mathrm{i}}^{n} \frac{x - x_\mathrm{j}}{x_\mathrm{i} - x_\mathrm{j}} = (b-a)\frac{1}{n+2}\int_x^n \prod_{\mathrm{j}=0,\mathrm{j}\neq\mathrm{i}} \frac{s-\mathrm{j}}{\mathrm{i}-\mathrm{j}} ds \tag{D.9}$$

where a = a + sh.