UNIVERSIDADE D
COIMBRA

Diogo Alexandre Fernandes de Almeida

Analysis of Non-Covalent Interactions in General Classes of Materials Using a Real-Space Charge Density Approach

Dissertação no âmbito do Mestrado em Física com Especialização em Física da Matéria Condensada sob orientação do Professor Doutor Bruce Forbes Milne e apresentada ao Departamento de Física da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Outubro de 2020

[Page purposely left blank]

Analysis of Non-Covalent Interactions in General Classes of Materials Using a Real-Space Charge Density Approach

Faculty of Sciences and Technologies of the University of Coimbra

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

Diogo Alexandre Fernandes de Almeida

October 2020

Abstract

Non-covalent interactions have long been known to be responsible for significant mechanisms in biological structures, and have historically lacked a satisfying or intuitive quantum mechanical description. The recently developed NCI index method by Johnson *et al* [1] fills a gap that existed within topological methods for the visualization of non-covalent interactions [2]. Even following its implementation by some of its original authors on the NCIPLOT software [3, 4], the method itself has not gained much traction in the Physics community, despite its possible and pertinent use in visualizing and describing non-covalent interactions in solids, including novel materials in which this type of interaction is responsible for structural stabilization. The project described in this document aims to bridge this gap between the method and its potential users by implementing it on a significantly simple, fast, and easy to modify Python script which works for several types of systems and takes as input files of the universally known cube format, which can be output by most major quantum chemistry and electronic structure codes. The resulting code is tested on both simple systems and systems of significant interest.

Resumo

Há algum tempo se sabe que interações não-covalentes são responsáveis por mecanismos significativos em estruturas biológicas e, historicamente, careciam de uma descrição quântica intuitiva ou satisfatória. O método 'NCI index' desenvolvido recentemente por Johnson *et al* [1] preenche uma lacuna que existia em métodos topológicos para a visualização de interações não-covalentes [2]. Mesmo depois da sua implementação por alguns dos seus autores originais no software NCIPLOT [3,4], o método em si não ganhou muita visibilidade na comunidade de Física, apesar do seu uso possível e pertinente na visualização e descrição de interações não-covalentes em sólidos, incluindo novos materiais nos quais este tipo de interação é responsável pela estabilização estrutural. O projeto descrito neste documento visa preencher esta lacuna entre o método e seus potenciais usuários, implementando-o num 'script' Python significativamente simples, rápido e fácil de modificar, que funciona para vários tipos de sistemas e toma como 'input' ficheiros do formato universalmente conhecido 'cube', que pode ser gerado pela maioria dos principais códigos de química quântica e estrutura eletrónica. O código resultante é testado tanto em sistemas simples como sistemas de interesse significativo.

Declaration

This thesis, which is submitted for consideration for the degree of Master, is a record of research carried out in the Department of Physics, University of Coimbra, under the supervision of Dr. Bruce F. Milne. It is believed to be original except where due reference has been made and has not been presented for any other higher degree. Co-orientation was provided by Prof. Fernando Nogueira.

Acknowledgements

The task of thanking the people responsible (in one way or another) for the content of this document is a daunting one, as it is impossible to account for everyone in my life that made it possible, and presenting a list of people is somewhat subjective. With that in mind, I will do my best in the following paragraphs, though I note the order is in no way meant as a measure of greater or minor acknowledgment.

I'd firstly like to thank my parents and the rest of my immediate family for their financial and emotional support. This document is the culmination of the progress of my education thus far, and is thus a product of many years of hard work from their part in ensuring that I have the conditions to pursue something I deem noble and meaningful.

None of this would have been possible without the support and understanding of my wife, Selena, who stood by me through everything and continues to be my role model as a shining example of willpower, determination and endurance. Her native and professional knowledge of the English language was also a major help in writing this document.

I also found I could always count on my colleagues at the Physics Department, who provided thoughtful discussion and encouragement, as well as a stable and caring environment for working through this, something that is often underrated. I'd particularly like to mention Miguel Murça for his inexhaustible help and expertise on computational issues. On this note, I'd also like to thank everyone working in the Physics Department, responsible for making it (and keeping it) what it is today.

I would like to briefly mention Dr. Ask Hjorth-Larsen, who took time off his busy schedule to meet with me and provide thoughtful feedback on the code being developed, at a time in which it was very little optimized and 'polished'.

Dr. Micael Oliveira was responsible for a large portion of the work that lead up to my eventual contribution to the paper on Bismuth carbide clusters.

A big thank you to Prof. Fernando Nogueira, for accepting and welcoming me into the research group some years ago. The weekly group meetings provided necessary structure and thoughtful discussion, as well as an introduction to many major areas of research in Condensed Matter and Chemical Physics, and of course, to the person who would become my supervisor, Dr. Bruce Milne. Prof. Nogueira also reviewed part of the present document and provided valuable help in theoretical aspects of electronic structure calculations. This thank you is extended to everyone

in the Center for Physics of the University of Coimbra.

Last but (by far) not least, I thank Dr. Bruce Forbes Milne for supervising this project, and for giving me the opportunity to work on something useful to the scientific community. His support was invaluable in every possible way, for the literature recommended and personally provided, his words of encouragement, his writing recommendations as an expert in the English language, his scientific and academic expertise in areas of Chemistry (and Physics) I still do not fully understand, and especially for being a trusting friend in a stressful time. I could not recommend a better supervisor.

Molecular graphics were performed with UCSF Chimera, developed by the Resource for Biocomputing, Visualization, and Informatics at the University of California, San Francisco, with support from NIH P41-GM103311.

The author also wishes to acknowledge the Laboratory for Advanced Computing at the University of Coimbra for providing computing resources which have contributed to the research results reported within this document.

Contents

List of Figures

"I'll be honest, we're throwing science at the wall here to see what sticks."

— Cave Johnson

Chapter 1

Introduction

1.1 Motivation

Non-covalent interactions (NCIs) have been a subject of interest and research for many years. Despite being responsible for many bonding mechanisms in biological systems and other important areas such as materials engineering, and the fact that many of these interactions have been empirically categorized into Hydrogen bonds, dipole-dipole interactions, van der Waals interaction, steric crowdings, *etc.*, their non-covalent nature makes it hard to establish a rigorous physical description of them. Unlike covalent interactions, which can now be easily studied from an *ab initio* perspective using *e.g.* the electron localization function (ELF) approach [5], or orbital-based methods, NCIs are known for being considerably difficult to study with simple many-body approaches like density functional theory (DFT) methods, due to the inherent non-locality of the exchange and correlation (XC) effects of the interactions. This is, however, an effect of the failure of local and semi-local XC functionals in describing the energy of the interactions, and which are very sensitive to small changes in Kohn-Sham orbitals and charge density. The charge density itself is not as dependent on the calculation mode [6] and is typically well described at a lower level of theory than the one needed for proper energy convergence. With this in mind, a method for studying NCIs based entirely on charge density, however it may be computed, is a much needed tool in physical chemistry, and a testament to the assertion of the Hohenberg-Kohn theorems [7] that all ground state properties of a quantum mechanical system are functionals of its ground state charge density.

Over the past decade, a new method has been developed by Johnson *et al* [1] for studying NCIs named the NCI index analysis, based entirely on the charge density, and subsequently implemented in the Fortran-based software NCIPLOT [3] which is currently on version 4 and includes a variety of newly developed tools for NCI analysis [4]. For NCI calculations on solids, the method has also been implemented into the Critic software [8, 9]. However, NCIPLOT has two main modes of use, which are reflected in the types of input used for the analysis:

- NCI index from pro-molecular densities, for which the geometry of the system is input in `xyz` format, and the density is built from piece-wise exponential atomic orbitals centered on atomic positions.

- NCI index from self-consistent field (SCF) calculations, for which the quantum

mechanical information of the system is input *via* a `wfn` format file resultant from an SCF calculation.

Although some quantum chemistry codes can output this `wfn` file type, it is certainly not common among the Physics community. Furthermore, since the method of analysis is a real-space method, it seems to make more physical sense to use the universally known Cube format file as input. There have been programs developed which interface NCIPLOT with a cube file input, and these are available in the NCIPLOT website[1]. However, direct analysis of a cube file should be possible without having to use basis sets as an intermediary, and an analysis of this type is, in principle, more physically sound.

With this in mind, a plan was made to re-implement the NCI index method in a Python code which takes cube files as a direct input. Using this language also makes the code more readily available and easy to use, as no installation is required, and has the added benefit of making the code much easier to modify according to the community's needs. Additionally, by making the data generated by the Python implementation be output as data and cube files for use with other software, it also gives the user better control over the results of the analysis. In later stages of the code development a decision was made to have it interface with the Atomic Simulation Environment (ASE) [10] Python package, available on Anaconda, in order to make it easier to analyze densities directly resultant from calculations run with ASE, as well as use some of the already professionally developed ASE capabilities for dealing with unit cells and cube file import and export tools.

This implementation has another apparent advantage over the established NCIPLOT[2] as its real-space nature makes it capable of handling periodic structures with non-orthogonal unit cells.

1.2 Thesis outline

In Chapter 2 a very basic overview of DFT is done in order to clarify the origin of the reduced density gradient (RDG), a key quantity in the method being implemented [1]. Although details are scarce in this chapter, adequate bibliographical references are made. The NCI index method is briefly described and its connections to the Quantum Theory of Atoms in Molecules (QTAIM) by Richard Bader [11], specifically to bond critical points (BCPs) and the physical meaning of density derivatives.

In Chapter 3, an overview is given of the code developed by the author as part of the project described in this document. Complements to this chapter are referenced in the text and present in Appendices A, B and C.

In Chapter 4, Kohn-Sham DFT single-point energy and density calculations are run for several systems of interest in which significant non-covalent interactions are

[1] https://www.lct.jussieu.fr/pagesperso/contrera/nci-oprograms.html

[2] As well as many disadvantages, which are briefly mentioned in the Conclusion of the present document.

present, using different methods and codes, divided into four groups: simple test systems, which serve as a way to test the capabilities of the developed code and to study certain specific aspects of visualization and differences between calculation methods used; Group XII metal dimers, three geometrically simple systems of interest when studying van der Waals dispersion in relatively heavy metals, and for which convergence tests are done before using the developed code for NCI analysis; periodic systems, for which experimental and in some cases optimized structures were used to compute densities and test the code's capabilities in these types of systems; and Bismuth carbide clusters, a potential case of practical application of the developed code in studying the formation of metal carbide clusters through dimerization of simpler systems.

We conclude in Chapter 5 and provide a few interesting ideas for future work to be done both on and with the developed code.

1.3 Atomic units

Throughout this document, mainly in the theoretical introduction, (Hartree) atomic units[3] are implied in every expression unless otherwise specified. The following is a small exposition of this system of units.

The time independent Schrödinger equation for a (neutral) hydrogen atom, considering the proton a static point charge at the origin, is

$$\left[-\frac{\hbar^2}{2m_e}\nabla^2 - \frac{e^2}{4\pi\varepsilon_0}\frac{1}{r}\right]\psi(\mathbf{r}) = E\,\psi(\mathbf{r}) \tag{1.1}$$

in which $\hbar$ is the reduced Plank constant, m_e the electron mass, e the (norm of the) electron charge, ε_0 the vacuum electric permittivity, $\psi(\mathbf{r})$ the one-particle wavefunction which describes the system and E the corresponding energy. Solving this eigenvalue/eigenfunction equation, we can obtain the normalized ground state solution

$$\psi_{1s}(\mathbf{r}) = \frac{1}{\sqrt{\pi}a_0^{3/2}}e^{-r/a_0} \tag{1.2}$$

with

$$a_0 = \frac{4\pi\varepsilon_0}{e^2}\frac{\hbar^2}{m_e} \tag{1.3}$$

the Bohr radius. The square modulus of $\psi_{1s}(\mathbf{r})$ is the ground state electronic density, and by multiplying $|\psi(\mathbf{r})|^2$ and $4\pi r^2$ we get the ground state radial distribution density $u(r)$, which can be shown to peak exactly at a_0, meaning this is the most probable distance between the electron and the proton in the ground state of the hydrogen atom.[4] It thus makes sense to use a system of units such that lengths are measured in a_0. Additionally, one can see how equation (1.1) would be simplified by using m_e as a unit of mass and e as a unit of charge. To complete a coherent system of units of mass, charge, length and time, we simply need to define a unit of

[3]Named after atomic physicist Douglas Hartree [12]

[4]Not to be confused with the average distance between the two particles in the ground state, which is $(3/2)\,a_0$.

time, or find an equivalent definition. The ground state energy which accompanies the solution of the wavefunction found is

$$E_0 = -\frac{1}{2}\frac{m_e}{\hbar^2}\left(\frac{e^2}{4\pi\varepsilon_0}\right)^2 = -\frac{1}{2}\frac{\hbar^2}{m_e}\frac{1}{a_0^2} \tag{1.4}$$

We now have a choice between defining the norm of this quantity as the atomic unit of energy, from which we get $\hbar = \sqrt{2}$ atomic units of action; or simply defining $\hbar$ as the atomic unit of action, getting $E_0 = -1/2$ atomic units of energy as a result. The obvious choice falls on setting $\hbar = 1$, and the resultant atomic unit of energy is typically referred to as an Hartree, for which the symbol E_h is used. Note that this choice also implies, from equation (1.3),

$$4\pi\varepsilon_0 = \frac{m_e e^2}{\hbar^2}a_0 = 1\,\text{a.u. of 'permittivity'} \tag{1.5}$$

although the quantity $4\pi\varepsilon_0$ was never directly set to 1 in this unit system; it is only a consequence of the derived units.

For further characterization of the atomic unit system, let us consider the dimensionless fine structure constant

$$\alpha = \frac{e^2}{4\pi\varepsilon_0}\frac{1}{\hbar c} \tag{1.6}$$

Note that from equations (1.3) and (1.6) we can write

$$a_0 = \frac{1}{\alpha}\frac{\hbar}{m_e c} \tag{1.7}$$

Thus, in atomic units,

$$c = \frac{1}{\alpha}\frac{\hbar}{m_e a_0} \approx 137\,\text{a.u.} \tag{1.8}$$

This allows us to derive an atomic unit of time as

$$\tau = \frac{a_0}{\alpha c} = \frac{m_e}{\hbar}a_0^2 = \frac{m_e}{\hbar}\frac{\hbar^2}{m_e}\frac{1}{E_h} = \frac{\hbar}{E_h} \tag{1.9}$$

which is consistent with the definitions of $\hbar$ as the atomic unit of action and E_h as the atomic unit of energy.

The value in SI units of the four fundamental constants from which all units in the atomic unit system are derived is presented in Table 1.1, as well as the direct SI value for the Hartree energy, given its considerable use in the kind of computations used in this work. For units which can be written in terms of distance and energy directly, a conversion factor is also shown in Table 1.1 between these and their corresponding values in Ångström (Å)[5] and electron-Volt (eV), commonly used in experimental setups and measurements.

The remaining three basic units of the SI are the Kelvin (K), mole (mol) and Candela (Cd), for measuring temperature, amount of substance, and luminous intensity, respectively. The latter has no relevance in the computations in question, and the first two have no reason to be disregarded. As such, they can be included in the atomic system of units, though in practice only the Kelvin is used, since the computations done are based on a relatively low number of particles (10^0 to 10^2).

[5] In honor of Anders Jonas Ångström, 1814 – 1874.

Table 1.1: Common conversion factors for the atomic unit system. [13]

physical quantity	symbol in a.u.	value in SI units	value in exp. units
action	$\hbar$	$1.054\ 571\ 817{\times}10^{-34}$ Js^{-1}	–
charge	e	$1.602\ 176\ 634{\times}10^{-19}$ C	–
length	a_0	$5.291\ 772\ 109\ 03(80){\times}10^{-11}$ m	0.529 177 Å
mass	m_e	$9.109\ 383\ 7015(28){\times}10^{-31}$ kg	–
energy	$E_\mathrm{h} = \frac{\hbar^2}{m_e a_0^2}$	$4.359\ 744\ 722\ 2071(85){\times}10^{-18}$ J	27.211 386 eV
probability density	a_0^{-3}	–	6.748 343 Å^{-3}

Chapter 2

Theoretical background

This chapter concerns the theoretical background leading up to the key quantities which are central to the Non-Covalent Interaction (NCI) index analysis implemented in this project.

2.1 Basics of Density Functional Theory

2.1.1 Hohenberg-Kohn Theorems

The Hohenberg-Kohn Theorems, originally stated and proved by Hohenberg and Kohn [7] in 1964, place the complete description of the ground state in the 3-coordinate dependent (and physical observable) electron density $\rho(\mathbf{r})$, as opposed to the abstract $3N$-component wavefunction, by stating that all ground state properties of a quantum many-body system are functionals of $\rho(\mathbf{r})$.

The first theorem states that the ground state density $\rho(\mathbf{r})$ uniquely determines the external potential $v_{\text{ext}}(\mathbf{r})$ up to an additive constant. The second theorem states that the exact ground state density can be obtained variationally and corresponds to the exact ground state density. In proving these theorems, Hohenberg and Kohn make clear the existence of a universal functional[1] $F[\rho]$ in such a way that the total ground state energy can be written as [14]

$$E[\rho] = F[\rho] + \int d^3\mathbf{r}\, v_{\text{ext}}[\rho](\mathbf{r}) \times \rho(\mathbf{r}) \tag{2.1}$$

The physical meaning of the universal functional is then defined as the part of the energy of the system that is owed to the electrons' kinetic energy as well as their interaction, *i.e.*, the energy that is not owed to the interaction of the particles with the external potential. Unfortunately no exact form of the $F[\rho]$ functional is known to date and needs to be approximated for most systems of interest. However, if $F[\rho]$ was known, the ground state density would be obtained variationally (as stated) by solving

$$\frac{\delta}{\delta\rho}F[\rho] + v_{\text{ext}}(\mathbf{r}) = \mu \tag{2.2}$$

where μ is a Lagrange multiplier which physically corresponds to the chemical potential of the system, and was introduced to impose the constraint that the number

[1] By universal it is meant that the functional is system independent.

of particles remains the same.

$$\int d^3\mathbf{r}\, \rho(\mathbf{r}) = N \tag{2.3}$$

2.1.2 Kohn-Sham DFT

In 1965, Kohn and Sham devised a method [15] of approximating the $F[\rho]$ functional by considering a fictitious system of non-interacting electrons associated with the real system. By construction, this fictitious system, named the Kohn-Sham system, has the exact same ground state density and energy as the real system.

Since the electrons are non-interacting in the Kohn-Sham system, the total energy can be written

$$E[\rho] = T_S[\rho] + \int d^3\mathbf{r}\, \rho(\mathbf{r}) \times v_S[\rho](\mathbf{r}) \tag{2.4}$$

with $T_S[\rho]$ the non-interacting kinetic energy and $v_S[\rho](\mathbf{r})$ the Kohn-Sham potential, which replaces the external potential in the fictitious system by 'simulating' the external potential and the electron-electron interactions. In this system the ground state wavefunction is a simple Slater determinant of the Kohn-Sham orbitals $\phi_i(\mathbf{r})$ which satisfy the Schrödinger-like equation

$$\left[-\frac{1}{2}\nabla^2 + v_S(\mathbf{r})\right] \phi_i(\mathbf{r}) = \epsilon_i\, \phi_i(\mathbf{r}) \tag{2.5}$$

The ground state density is built from the Kohn-Sham orbitals as

$$\rho(\mathbf{r}) = \sum_{i=1}^{N} |\phi_i(\mathbf{r})|^2 \tag{2.6}$$

In order to find $v_S(\mathbf{r})$, we must write

$$F[\rho] = T_S[\rho] + U[\rho] + E_{xc}[\rho] \tag{2.7}$$

where $U[\rho]$ is the Hartree electrostatic interaction energy

$$U[\rho] = \frac{1}{2}\int \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\, d^3\mathbf{r}\, d^3\mathbf{r}' \tag{2.8}$$

and $E_{xc}[\rho]$ is, by definition, the exchange and correlation (XC) energy. This is the part of $F[\rho]$ which needs to be approximated and for which several functionals have been developed over many decades with varying degrees of accuracy [14]. This degree of accuracy is usually dependent on the level of locality of the functional.

From here, and after selecting an adequate functional for $E_{xc}[\rho]$ it is possible to write the set of equations to be solved self-consistently known as the Kohn-Sham equations: equation (2.5) and

$$v_S[\rho](\mathbf{r}) = v_{\text{ext}}(\mathbf{r}) + v_H[\rho](\mathbf{r}) + v_{xc}[\rho](\mathbf{r}) \tag{2.9}$$

with the Hartree and XC potentials given by

$$v_H[\rho](\mathbf{r}) = \frac{\delta}{\delta\rho} U[\rho] \tag{2.10}$$

$$v_{xc}[\rho](\mathbf{r}) = \frac{\delta}{\delta\rho} E_{xc}[\rho] \tag{2.11}$$

2.1.3 Jacob's ladder

Perdew and Schmidt [16] explained the relative accuracy of E_{xc} functionals in terms of their level of locality by constructing a fictional object they coined Jacob's ladder: each step gets us closer to the Heaven of chemical accuracy.[2] Although the ladder as described has five rungs of increasing accuracy (and increasing computational resources needed to use them), we will focus here only in the first three. These are

1. the Local Density Approximation (LDA), in which E_{xc} is only dependent on ρ, *i.e.*, its local value – this functional is exact in systems of spatially constant density, *i.e.*, the homogeneous electron gas;

2. the Generalized Gradient Approximation (GGA), which adds a dependence on the density gradient $\nabla\rho$ to the E_{xc};

3. meta-GGA (or mGGA), which adds to E_{xc} a dependence on $\nabla^2\rho$, or, equivalently, the kinetic energy density $\tau(\mathbf{r})$.

2.1.4 NCI index in DFT

To build GGA functionals, a measure of deviation of the variation of density from the constant density of the homogeneous electron gas is required. To make this measure more universal and respect scaling relations, it should be normalized to the density value. A choice for this measure could then be $|\nabla\rho|/\rho$. However, this measure should be made adimensional for the sake of universality. Noting it has units of length^{-1}, we can do this by choosing a length scale to multiply it by. Since this length scale should also be density-dependent, we choose the Fermi wavelength $1/(2k_F)$ of the homogeneous electron gas of density ρ,

$$s = \frac{|\nabla\rho|}{2k_F\,\rho} \tag{2.12}$$

where k_F is the radius of the Fermi sphere of the electron gas which, in atomic units, has the value

$$k_F = \left(3\pi^2\rho\right)^{1/3} \tag{2.13}$$

Thus, we have, in atomic units, the reduced density gradient (RDG)

$$s = C_s\,\frac{|\nabla\rho|}{\rho^{4/3}}\,, \qquad C_s = \frac{1}{2\left(3\pi^2\right)^{1/3}} \tag{2.14}$$

This measure is an active ingredient in building the exchange part of GGA functionals, as [2]

$$E_x^{(\mathrm{GGA})} - E_x^{(\mathrm{LDA})} = -\int F(s)\,\rho^{4/3}\,d^3\mathbf{r} \tag{2.15}$$

with $F(s)$ a spin-dependent function.

[2] As most if not all other forms of Heaven, it is an arguably unachievable ideal, but one to strive for.

2.2 Basic overview of the NCI index method

A more thorough overview of already established NCI index methods, and of non-covalent interactions in general, is provided in reference [2].

Take a single atom for which a map of the charge density is computed, and subsequently the gradient and reduced density gradient (RDG) from equation (2.14). For a Slater-type orbital (STO) model density, which is accurate enough in density tails, we have

$$\rho(r) = Ae^{-2\alpha r} \tag{2.16}$$

Computing the RDG, we get

$$s = C_s \frac{2\alpha\rho}{\rho^{4/3}} = 2\alpha C_s \, \rho^{-1/3} \tag{2.17}$$

Supposing we can map each point in real-space to a point in a $s(\rho)$ plot, we get the line described in (2.17). Even if there are several atoms, most points will follow an identical trend, as most point are part of the density tails of these atoms. However, in this case, interactions exist between the atoms, and there must be a point in which the density is finite but its gradient is zero, due to contributions to the density from several atoms.[3] This is visible in the mentioned $s(\rho)$ plot as a thorough, or inverted peak, which appears for low densities from the main trend all the way down to near-zero values of the RDG. This is thus a way of identifying interactions, and the strength of the interaction can be gauged by the value of the density at these peaks. By plotting an isosurface in real-space of a low RDG value, we have a way to map these interactions back into real-space and identify them visually in an intuitive manner.

This does not, however, give us any information on whether the interaction is bonding or non-bonding. Most topological methods of this type, including Bader's QTAIM [11], make use of the laplacian of the density at specific points, named bond critical points (BCPs), to distinguish between bonding and non-bonding interactions, as this way one can gauge whether there is an inward or outward density gradient flux, by the divergence theorem.

This does not work for non-covalent interactions, because for inter-nuclear regions, the laplacian is dominated by only the positive principal variation. Some of the underlying principles from QTAIM present a solution for this problem, and are used in NCI index. The laplacian of the density is the trace of the density's Hessian matrix, *i.e.*, the sum of its eigenvalues. Since the Hessian is a real and symmetric matrix, it can be diagonalized if written in terms of its eigenvectors, which are the principal axes of variation. Let $\lambda_1 \leq \lambda_2 \leq \lambda_3$ be the three eigenvalues of the Hessian matrix of the density. In inter-nuclear regions, the trace of the matrix is dominated by the positive λ_3, $\lambda_1 < 0$ and λ_2 can be either positive or negative. The bonding or non-bonding nature of the non-covalent interaction can then be inspected from the sign of the second eigenvalue, that is, the sign of the variation of density (second derivative) along the principal variational axis perpendicular to both the minimum

[3] These points of zero gradient are the bond critical points (BCPs) referenced in QTAIM.

and maximum directional variations. Although this information is hard to accept at face value from the author's explanation, analyses using this method on known systems in section 4.1 hopefully give it the credibility it requires.

Thus, by finding for each point in real space the sign of the median eigenvalue of the Hessian matrix of density, and substituting ρ in the x-axis of the $s(\rho)$ plot by the value $\text{sgn}(\lambda_2) \times \rho$, bonding and non-bonding interactions (or bonding and non-bonding contributions to the same interaction) get separated into different sides of the plot: bonding peaks appear on the negative side and non-bonding peaks appear on the positive side. To complement the isosurface mapping with this new information, we color each point of the represented isosurfaces according to $\text{sgn}(\lambda_2) \times \rho$. This way, bonding and non-bonding interactions / contributions have different colors, and the strength of the interaction can be gauged from the color intensity. Thus, a qualitative but intuitive analysis of NCIs in structures in real-space is made possible.

NCI index as an extension of QTAIM

Implementations of QTAIM in real-space quantum chemistry or electronic structure codes to find and categorize BCPs and separate space into Bader volumes are typically very computationally demanding, as space needs to be thoroughly divided into smaller sections until single points are found. [9] If this is done on spatial grids, interpolation and multigrid methods are required since the specific location of BCPs may not be in the grid points. On the other hand, the NCI index analysis described here simply makes use of the value of the density and the norm of its gradient at every point in the grid, which can be easily computed and which process can even be vectorized. Because of the continuity of these quantities over real space, the points surrounding BCPs are easily identified through the RDG isosurfaces: they have small RDG values and in fact encapsulate a region of space in which the BCP is located. On the other hand, the isosurface coloring according to $\text{sgn}(\lambda_2) \times \rho$ is in itself a way to categorize the BCPs which correspond to non-covalent interactions, albeit not with the full information which is typical of regular QTAIM (exact eigenvalues of the Hessian matrix at the BCP). The NCI index analysis can then be seen as a computationally lighter extension of QTAIM.

Chapter 3

Computational Method

The objective of this chapter is to give a thorough understanding of how the script developed by the author works in the mentioned NCI index analysis of several systems. The code is publicly available[1] under the GPL (version 3.0), and at the time of submission of this document in version 0.3.1. [17] Because of this accessibility, specific computational aspects related with the programming language – such as array slicing – will not be discussed here.

With this intent, the first sections will focus on mathematical details of specific functions of the script, and later sections will include an overview of the analysis steps as well as a flowchart schematic.

3.1 Inputs

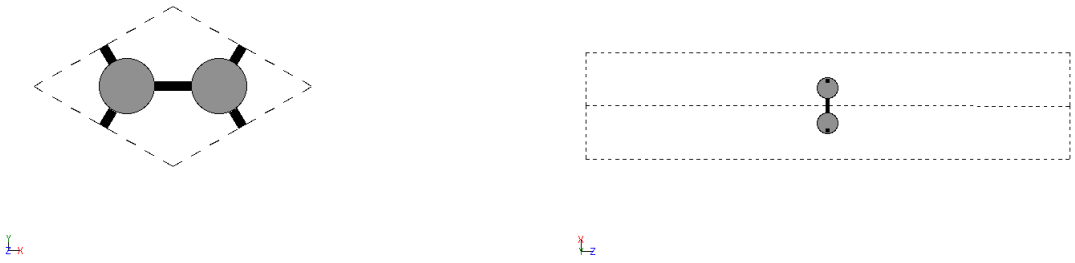
We can categorize the analysis inputs as system-dependent or user-defined. Because of the script's dependence on the Atomic Simulation Environment (ASE) Python package [10], the system dependent inputs can be read from a cube file exported by any quantum chemistry or electronic structure code that exports this type of file; or they can be used directly from ASE if the calculation is done using the ASE interface. The system-dependent inputs are

- density – a 3D array containing the value of the density at every grid point;
- atoms object – the ASE `Atoms` object of the system being analyzed (contains information on the simulation box and system geometry);
- origin – a 1D 3-component array with the origin of the cube file, *i.e.*, the real-space position of the bottom-leftmost grid point (indexed by $0, 0, 0$).

All 3 system-dependent inputs can be read from a cube file using the ASE cube-reading function. The user-defined inputs have to do with the analysis process and not with the system specifically, and will be referenced during the description of the analysis.

It is worth mentioning that since the input density can be read directly from a cube file, it does not matter how exactly it was computed or obtained. Ideally,

[1] https://gitlab.com/diofalmeida/pynci



(a) XY - plane view

(b) ZX - plane view

Figure 3.1: Schematic representation of the unit cell of graphene for density calculation, with cell vectors $\mathbf{a} = \left(2.196\,\hat{i} + 1.268\,\hat{j}\right)$ Å, $\mathbf{b} = \left(2.196\,\hat{i} - 1.268\,\hat{j}\right)$ Å, $\mathbf{c} = -20\,\text{Å}\,\hat{k}$, obtained with the ASE graphical user interface.

the method can be used to analyze real experimental densities obtained *via* X-ray crystallography (XRC) or other methods. This means that although the reduced density gradient arises naturally within the framework of DFT, and the NCI method itself owes its physical significance to the Hohenberg-Kohn theorems,[2] the densities used are by no means constrained to computations using DFT tools.

3.2 Simulation box / unit cell

The systems analyzed usually fall into one of two categories, or might occasionally display characteristics of the two: (i) molecular or atomic systems, for which the density is typically computed on a large cuboid grid with zero-value and zero-gradient border conditions; and (ii) solid systems, for which the density is typically computed on the primitive unit cell – defined by 3 not necessarily orthogonal vectors **a**, **b** and **c** – with completely periodic border conditions. Systems that exhibit characteristics of both these situations include but are not limited to 2-dimensional periodic structures, such as graphene sheets, with zero-value and zero-gradient density on the two borders which are parallel to the structure. Figure 3.1 presents a possible unit cell for the computation of electronic density for a graphene sheet.[3]

The cell parameters of the unit cell are encoded in the atoms object in the form of the cell matrix, Q, a 3×3 matrix whose lines are the real-space components of the unit cell vectors **a**, **b** and **c**.

$$Q = \begin{pmatrix} \mathbf{a}\cdot\hat{i} & \mathbf{a}\cdot\hat{j} & \mathbf{a}\cdot\hat{k} \\ \mathbf{b}\cdot\hat{i} & \mathbf{b}\cdot\hat{j} & \mathbf{b}\cdot\hat{k} \\ \mathbf{c}\cdot\hat{i} & \mathbf{c}\cdot\hat{j} & \mathbf{c}\cdot\hat{k} \end{pmatrix} \tag{3.1}$$

For molecular systems with cuboid simulation boxes, the cell matrix is diagonal. Thus, the simulation box of a molecular system with an orthogonal grid can be seen as an orthogonal unit cell, and will hereby be referenced to as the unit cell of the system even in non-periodic systems. The border conditions, on the other hand, are

[2] See section 2.1.1

[3] It is worth nothing the cell parameters mentioned are not the parameters of the ground state configuration of graphene and serve only to create a schematic representation.

encoded separately into the atoms object as a 1-dimensional 3 component array of Boolean-type values, dictating whether the border conditions are periodic in each of the unit cell vectors' directions. The grid in which space is discretized is also given by the unit cell vectors, meaning it is only an orthogonal grid if the unit cell vectors are orthogonal. If the shape of the density array is $[N_\mathrm{a}\ N_\mathrm{b}\ N_\mathrm{c}]$, the grid point positions in real space are given by

$$\mathbf{r}_{ijk} = \mathbf{r}_{000} + \frac{i}{N_\mathrm{a}}\,\mathbf{a} + \frac{j}{N_\mathrm{b}}\,\mathbf{b} + \frac{k}{N_\mathrm{c}}\,\mathbf{c} \tag{3.2}$$

in which $\mathbf{r}_{000}$ is the origin of the grid.

As an example of a cell matrix, consider the possible unit cell for graphene shown in Figure 3.1. The cell parameters a and b have the same value, the angle between them is $\gamma = \pi/3$ (60°), and the independent c parameter is chosen adequately so that the atoms are far enough away from the non-periodic border. The cell matrix for the represented unit cell is then, in terms of the cell parameters,[4]

$$Q = \begin{pmatrix} \frac{a\sqrt{3}}{2} & \frac{a}{2} & 0 \\ \frac{a\sqrt{3}}{2} & -\frac{a}{2} & 0 \\ 0 & 0 & -c \end{pmatrix}$$

Although border conditions are handled independently of system type in each function of the script, a distinction between cuboid and non-cuboid unit cells still needs to be made for the sake of computing derivatives *via* finite difference methods. For this reason, the script takes into consideration whether the unit cell matrix Q is diagonal or not, and it does this by counting the number of non-zero components of

$$Q - \mathrm{diag}(Q) = \begin{pmatrix} 0 & \mathbf{a}\cdot\hat{j} & \mathbf{a}\cdot\hat{k} \\ \mathbf{b}\cdot\hat{i} & 0 & \mathbf{b}\cdot\hat{k} \\ \mathbf{c}\cdot\hat{i} & \mathbf{c}\cdot\hat{j} & 0 \end{pmatrix} \tag{3.3}$$

If Q is diagonal, the system is considered cuboidal, associated with an orthogonal grid. Then, the spacing in each direction is computed according to

$$\delta x_\alpha = \frac{L_\alpha}{N_\alpha}\,, \quad \alpha = 1,\, 2,\, 3 \equiv \mathrm{a},\, \mathrm{b},\, \mathrm{c} \tag{3.4}$$

with L the length of the unit cell in each direction ($a = ||\mathbf{a}||$, *etc.*) and N the resolution of the grid in each corresponding direction. Thus, rewriting equation (3.2) for orthogonal grids,

$$\mathbf{r}_{ijk} = \mathbf{r}_{000} + i\,\delta x_1\,\hat{i} + j\,\delta x_2\,\hat{j} + k\,\delta x_3\,\hat{k} \tag{3.5}$$

On the other hand, if Q is not diagonal, the system is considered non-cuboidal, and the associated grid non-orthogonal.[5] For these cases, it is common to do calculations in a system of fractional coordinates, as opposed to real space. Each real-space

[4]For this system the cell parameter a can be written from the inter-atomic distance d as $a = d\sqrt{3}$.

[5]The author recognizes the possibility of orthogonal grids associated with non-diagonal cell matrices, as long as the cell vectors are orthogonal. However, in practice, this is ignorable since: (i) the cell matrix is automatically diagonal for orthogonal systems and (ii) the way computations are done yield the same results as if this case was considered.

position $\mathbf{r}$ can be mapped to a corresponding point in the fractional coordinates system $\boldsymbol{\varepsilon} \equiv [\varepsilon_1\, \varepsilon_2\, \varepsilon_3]^T$ (and *vice-versa*) according to its fractional position in relation to each of the cell vectors.[6] The mapping from fractional coordinates to a real space position is given by

$$\mathbf{r} = \mathbf{r}_{000} + \varepsilon_1\, \mathbf{a} + \varepsilon_2\, \mathbf{b} + \varepsilon_3\, \mathbf{c} \tag{3.6}$$

which can be presented in terms of components *via* a matrix relation

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \mathbf{a}\cdot\hat{i} & \mathbf{b}\cdot\hat{i} & \mathbf{c}\cdot\hat{i} \\ \mathbf{a}\cdot\hat{j} & \mathbf{b}\cdot\hat{j} & \mathbf{c}\cdot\hat{j} \\ \mathbf{a}\cdot\hat{k} & \mathbf{b}\cdot\hat{i} & \mathbf{c}\cdot\hat{k} \end{pmatrix} \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{pmatrix} \quad \equiv \quad \mathbf{x} = Q^T \boldsymbol{\varepsilon} \tag{3.7}$$

with $\mathbf{x} = \mathbf{r} - \mathbf{r}_{000}$. While it is not immediate to obtain the coordinates of a real-space position directly from a set of indices in a non-orthogonal grid – see equation (3.2) – by comparing equations (3.2) and (3.6) the fractional coordinates are trivially computed as a fraction of integers, $\varepsilon_\alpha^{(i)} = i/N_\alpha$.

3.3 Derivatives

It is possible to see[7] that derivatives on a grid can be approximated by finite difference expressions through which we compute derivative values in each point using the values of the derivated function in neighboring points. Thus, in an orthogonal grid, the gradient of the charge density (3D array) is computed as a 4D array in which each component is indexed by 4 integers:[8]

$$[\nabla\rho]_\alpha \Big|_{ijk} = \frac{\partial\rho}{\partial x_\alpha}\Big|_{ijk} \approx g_{ijk}^{(\alpha)} \tag{3.8}$$

where $\alpha = 1,\, 2,\, 3$ is the axis along which ρ is being derivated and the indices (ijk) relate to the grid point position, which directly corresponds to a position in real-space; and the spacings used are the ones computed in (3.4).

If the grid is non-orthogonal, the derivatives computed *via* finite difference do not correspond to the real-space gradient components, since the components aren't independent in a non-orthogonal grid, *i.e.*, $x^\alpha_{(ijk)}$ does not depend only on one of the indices.

$$\frac{\partial\rho}{\partial x_\alpha} \neq g_{ijk}^{(\alpha)} \tag{3.9}$$

However, the grid is technically orthogonal in fractional coordinate space, since adding or subtracting to an index corresponds to moving only along one of the fractional coordinates. If we consider ρ as a scalar field in fractional-coordinate-space,

$$\rho(\boldsymbol{\varepsilon}) \equiv \rho(\mathbf{r} = \mathbf{r}_{000} + \varepsilon_1\, \mathbf{a} + \varepsilon_2\, \mathbf{b} + \varepsilon_3\, \mathbf{c}) \tag{3.10}$$

the ρ_{ijk} grid is orthogonal in this space, and the grid spacings are simply

$$\delta\varepsilon_\alpha = \frac{1}{N_\alpha} \tag{3.11}$$

[6] $\varepsilon_i \in [0, 1]\, \forall i$

[7] See Appendix A

[8] One index for the component and three for the grid point position.

Thus, the derivatives computed *via* finite difference – if the spacings defined in (3.11) are used – are derivatives in fractional coordinate space.

$$g_{ijk}^{(\alpha)} \approx \left.\frac{\partial \rho}{\partial \varepsilon_\alpha}\right|_{ijk} \tag{3.12}$$

We require the real-space gradient components in every grid point in order to compute s by equation (2.14). These real-space gradient components can be obtained directly from the fractional-coordinate space gradient, using the relation between real-space and fractional coordinates in (3.7). Let us denote $\nabla\rho$ as the real-space gradient, $\nabla_x \rho$ as the gradient in $\mathbf{x}$-space[9] and $\nabla_\varepsilon \rho$ as the gradient in fractional coordinate space.

Since $\mathbf{r}$ and $\mathbf{x}$ differ only by a constant vector, $\nabla \equiv \nabla_x$. Following a chain-derivation,

$$\frac{\partial \rho}{\partial x_\alpha} = \sum_{\beta=1}^{3} \frac{\partial \varepsilon_\beta}{\partial x_\alpha} \frac{\partial \rho}{\partial \varepsilon_\beta} \tag{3.13}$$

From (3.7),

$$\frac{\partial \varepsilon_\beta}{\partial x_\alpha} = \left[Q^{-T}\right]_{\beta\alpha} = \left[Q^{-1}\right]_{\alpha\beta} \tag{3.14}$$

Then, letting $\chi = Q^{-1}$, we have

$$\nabla\rho = \chi\, \nabla_\varepsilon \rho \tag{3.15}$$

Thus, if during the analysis a non-orthogonal grid is detected, matrix χ is computed from Q and is used to compute the real-space gradient from the fractional-coordinate gradient computed *via* finite-difference.

The same method can be applied to each component of the Hessian matrix after being computed in fractional-coordinate space, and before evaluating the sign of the median eigenvalue. From (3.13),

$$\begin{aligned}
\frac{\partial^2 \rho}{\partial x_\alpha\, \partial x_\beta} &= \frac{\partial}{\partial x_\alpha}\left(\frac{\partial \rho}{\partial x_\beta}\right) \\
&= \sum_{\gamma=1}^{3} \frac{\partial \varepsilon_\gamma}{\partial x_\alpha} \frac{\partial}{\partial \varepsilon_\gamma} \sum_{\theta=1}^{3} \frac{\partial \varepsilon_\theta}{\partial x_\beta} \frac{\partial \rho}{\partial \varepsilon_\theta} \\
&= \sum_{\gamma,\,\theta} \chi_{\alpha\gamma}\, \chi_{\beta\theta}\, \frac{\partial^2 \rho}{\partial \varepsilon_\gamma\, \partial \varepsilon_\theta} \\
H_{\alpha\beta}^{(x)}(\rho) &= \sum_{\gamma,\,\theta} \chi_{\alpha\gamma}\, H_{\gamma\theta}^{(\varepsilon)}(\rho)\, \left[\chi^T\right]_{\theta\beta}
\end{aligned} \tag{3.16}$$

which can be written as a matrix multiplication

$$H^{(x)} = \chi\, H^{(\varepsilon)}\, \chi^T \tag{3.17}$$

[9] A translated real-space given by $\mathbf{x} = \mathbf{r} - \mathbf{r}_{000}$.

Let us check that the Hessian matrix maintains its symmetry even in fractional coordinate space. Applying Q to the left of the equation, and Q^T to the right,

$$\begin{aligned} Q\,H^{(x)}\,Q^T &= Q\,\chi\,H^{(\varepsilon)}\,(Q\,\chi)^T \\ \Rightarrow H^{(\varepsilon)} &= Q\,H^{(x)}\,Q^T \end{aligned} \tag{3.18}$$

In terms of components,

$$H^{(\varepsilon)}_{\alpha\beta} = \sum_{\gamma,\,\theta} Q_{\alpha\gamma}\,Q_{\beta\theta}\,H^{(x)}_{\gamma\theta} \tag{3.19}$$

Since γ and θ are mute summation indices and $H^{(x)}_{\gamma\theta} = H^{(x)}_{\theta\gamma}$ by symmetry of $H^{(x)}$, switching α and β results in the same expression, thus $H^{(\varepsilon)}$ is symmetric. This means even in the case of a non-orthogonal grid, only the 6 independent components of the Hessian matrix need to be computed. Although this is more efficient in terms of memory requirements, it presents a problem – the computation of Hessian components in real space from the Hessian components in fractional-coordinate space cannot be done directly *via* the matrix multiplication in (3.17), and we need to rewrite this process in terms of the 6 independent components only.

Reverting $H^{(\varepsilon)}$ to real space using 6 independent components

Let us separate $H^{(\varepsilon)}$ into diagonal and non-diagonal components.

$$H^{(\varepsilon)} = H^D + H^A \tag{3.20}$$

with

$$H^D_{\alpha\beta} = H^{(\varepsilon)}_{\alpha\alpha}\,\delta_{\alpha\beta} \tag{3.21}$$

and

$$H^A = \begin{pmatrix} 0 & H^{(\varepsilon)}_{12} & H^{(\varepsilon)}_{13} \\ H^{(\varepsilon)}_{12} & 0 & H^{(\varepsilon)}_{23} \\ H^{(\varepsilon)}_{13} & H^{(\varepsilon)}_{23} & 0 \end{pmatrix} \tag{3.22}$$

Note that the three independent components of H^A can be indexed by only one integer ranging from 1 to 3, corresponding to the axis along which no derivatives are being taken. We designate this as the anti-axis corresponding to the matrix element.

$$H^A = \begin{pmatrix} 0 & a_3 & a_2 \\ a_3 & 0 & a_1 \\ a_2 & a_1 & 0 \end{pmatrix} \tag{3.23}$$

In fact, during the analysis, the grids of the 6 independent components of the Hessian matrix are stored in memory in a single 4-dimensional array, indexed by 4 integers (similarly to the gradient), $h^{(\alpha)}_{ijk}$, with α ranging from 1 to 6. The first three α 'components' are the 3D arrays which correspond to the 3 diagonal elements of the Hessian; and the remaining 3 are related to the mentioned anti-axes, following the relation $\alpha = 7 - \gamma$, with γ the anti-axis index. Thus the order in which the components appear is H_{11}, H_{22}, H_{33}, H_{12}, H_{13}, H_{23}. It is then useful to create an artificial mapping between integers 1 to 6 and pairs of integers ranging from 1 to 3, which maps each component's position in the 4D array to the hessian indices. Let us designate this mapping

$$\gamma \quad \longrightarrow \quad p(\gamma) = (\mu,\,\nu) \tag{3.24}$$

Table 3.1: Summarization of $p(\gamma)$ integers mapping.

γ	μ	ν
1	1	1
2	2	2
3	3	3
4	1	2
5	1	3
6	2	3

This $p(\gamma)$ mapping is summarized in Table 3.1.

From (3.17) and (3.20), we have

$$H^{(x)} = \chi\, H^D\, \chi^T + \chi\, H^A\, \chi^T \tag{3.25}$$

Computing explicit components of the first term, using (3.21),

$$\begin{aligned} \chi\, H^D\, \chi^T \Big]_{\alpha\beta} &= \sum_{\eta,\,\mu=1}^{3} \chi_{\alpha\eta}\, \chi_{\beta\mu}\, H^{(\varepsilon)}_{\eta\eta}\, \delta_{\eta\mu} \\ &= \sum_{\eta=1}^{3} \chi_{\alpha\eta}\, \chi_{\beta\eta}\, H^{(\varepsilon)}_{\eta\eta} \end{aligned} \tag{3.26}$$

Doing the same for the second term of (3.25),

$$\chi\, H^A\, \chi^T \Big]_{\alpha\beta} = \sum_{\eta,\,\mu=1}^{3} \chi_{\alpha\eta}\, \chi_{\beta\mu}\, H^{A}_{\eta\mu} \tag{3.27}$$

Note that there are only 6 non-zero terms out of the 9 total. This result can be rewritten evidencing only the non-zero terms by using the mapping of (3.24) as

$$\chi\, H^A\, \chi^T \Big]_{\alpha\beta} = \sum_{\eta=1}^{3} \left(\chi_{\alpha\mu}\, \chi_{\beta\nu} + \chi_{\beta\mu}\, \chi_{\alpha\nu}\right) H^{(\varepsilon)}_{\mu\nu}\,, \quad (\mu,\, \nu) = p(\eta) \tag{3.28}$$

Thus, summing equations (3.26) and (3.28), we obtain the components version of equation (3.25).

$$H^{(x)}_{\alpha\beta} = \sum_{\eta=1}^{3} \left[\chi_{\alpha\eta}\, \chi_{\beta\eta}\, H^{(\varepsilon)}_{\eta\eta} + \left(\chi_{\alpha\mu}\, \chi_{\beta\nu} + \chi_{\beta\mu}\, \chi_{\alpha\nu}\right) H^{(\varepsilon)}_{\mu\nu}\right]\,, \quad (\mu,\, \nu) = p(\eta) \tag{3.29}$$

Finally, considering the ordering of the Hessian components in the 4D array, which is the same as the ordering of the index pairs in the (3.24) mapping, we can write all components of a new array h^* as

$$\begin{aligned} h^{*(\gamma)}_{ijk} &= \sum_{\eta=1}^{3} \left[\chi_{\alpha\eta}\, \chi_{\beta\eta}\, h^{(\eta)}_{ijk} + \left(\chi_{\alpha\mu}\, \chi_{\beta\nu} + \chi_{\beta\mu}\, \chi_{\alpha\nu}\right) h^{(7-\eta)}_{ijk}\right]\,, \\ (\alpha,\, \beta) &= p(\gamma)\,, \\ (\mu,\, \nu) &= p(7-\eta)\,, \quad \forall\, \gamma = 1,\, ...,\, 6 \end{aligned} \tag{3.30}$$

and then do the substitution $h^{(\gamma)}_{ijk} \leftarrow h^{*(\gamma)}_{ijk}$. All of these computations are vectorized in the $\{ijk\}$ indices. Computationally, the reversion to a real-space Hessian is done as follows:

1. Compute $H^{(\varepsilon)}$ 4D-array h through finite-difference derivatives;
2. Define integer mapping from (3.24) and Table 3.1;
3. Initialize empty 4D-array h^*;
4. For $\gamma = 1$ to 6 do:
   (a) $h^{*(\gamma)} = 0$;
   (b) $(\alpha,\, \beta) = p(\gamma)$;
   (c) For $\eta = 1$ to 3 do:
      i. $(\mu,\, \nu) = p(7 - \eta)$;
      ii. $h^{*(\gamma)} \leftarrow h^{*(\gamma)} + \chi_{\alpha\eta}\, \chi_{\beta\eta}\, h^{(\eta)} + \left(\chi_{\alpha\mu}\, \chi_{\beta\nu} + \chi_{\beta\mu}\, \chi_{\alpha\nu}\right) h^{(7-\eta)}$
5. Substitute $h \leftarrow h^*$.

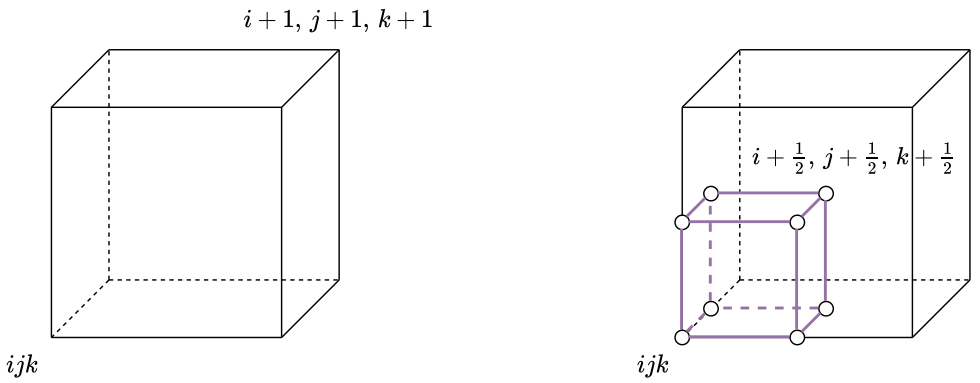
3.4 Interpolation

For its efficiency in the particular case of scalar fields over regular grids, a trilinear interpolation scheme was chosen to artificially increase the resolution of the scatterplot when necessary. This is done by interpolating the density, reduced density gradient and Hessian component arrays before evaluating the median eigenvalue of the Hessian. Typically, one must solve a system of linear equations involving the value of the interpolated scalar field at the eight corners of a cuboid[10] in which the point we wish to interpolate the field to is located, in order to find 8 coefficients of the interpolating function, and then use these to compute the interpolated value at any point inside the cuboid. This process is then repeated for every cuboid in the grid. However, in this particular case the points we wish to interpolate the field to have specific midway positions between the original cuboid corners (which correspond to the points in the original grid). This greatly simplifies the interpolation process and allows it to be vectorized.

Figure 3.2a presents a schematic representation of the cuboid cell in the original grid, the corners of which are consecutive grid points. In this interpolation process, the new grid has half the spacing in each direction, meaning in the refined grid there are 8 grid points for every point in the original grid:[11] 1 point which remains unaltered, 3 points from the edges of the cuboid which intersect at the original point,

[10]The interpolation process described in this section is fully generalizable to non-orthogonal grids, in which case the cell between grid points is not cuboidal. However, instead of dealing with real-space or fractional-coordinate-space, the interpolation is abstractly done in the (ijk) discrete index space, and the points to which the interpolation is done are half-integer index points in the original grid. This means there is no problem in calling the cell a cuboid – in fact, in this space it is always a cube.

[11]Borders are conveniently ignored in this explanation but handled accurately in the derived expressions.



(a) original cuboid

(b) cuboid in the refined grid

Figure 3.2: Schematic representations of the cuboid cell between grid points in the original grid and corresponding new points in the refined grid.

3 points from the faces of the cuboid which intersect at the original point, and 1 point from the center of the cuboid. These are represented in Figure 3.2b, and form a cuboid in the refined grid.

Let $f(\mathbf{r})$ be the general scalar field to be interpolated, and $\{\mathbf{r}_{ijk}\}$ the original grid. Then f_{ijk} is the value of the field at grid point $\mathbf{r}_{ijk}$. Let $\left\{\mathbf{r}^*_{ijk}\right\}$ be the refined grid, and $\left\{f^*_{ijk}\right\}$ be the values of f in the refined grid. We have the following relation between grids:

$$\mathbf{r}_{ijk} = \mathbf{r}^*_{2i,\,2j,\,2k} \tag{3.31}$$

$$f_{ijk} = f^*_{2i,\,2j,\,2k} \tag{3.32}$$

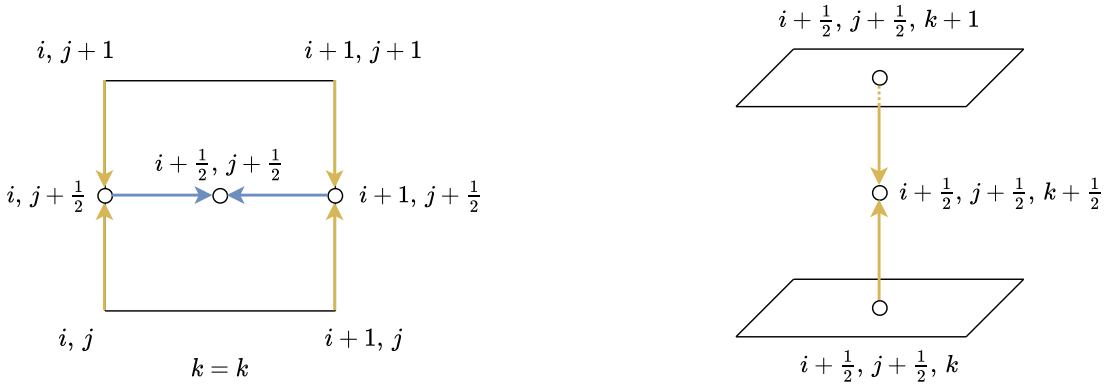
As such, and for simplification, we will be dealing only with f as opposed to using both f and f^*, and the points which are part of the refined grid but not the original grid are characterized by having at least one half-integer index. We will differentiate between these as edge points, face points and center points, as detailed in the discussion leading up to and in Figure 3.2: edge points have only one half-integer index, face points have two half-integer indices and center points have all three half-integer indices.

Edge points

If trilinear interpolation is done in only one direction it is simply linear interpolation. This means for edge points the value of f will be a simple arithmetic mean from the two grid points which share the edge. Letting i be the index related to the direction of the edge in which the point is, we have

$$f^{jk}_{i+\frac{1}{2}} = \frac{1}{2}\left(f^{jk}_{i} + f^{jk}_{i+1}\right) \tag{3.33}$$

where i is under-scripted and j and k are super-scripted to better differentiate the direction of (linear) interpolation. Equations for the other two directions are achieved by simple switch of indices between i, j and k. This process is schematically represented by the gold arrows in Figure 3.3a.



(a) edge and face points

(b) center points

Figure 3.3: Schematic representation of the interpolation to midway points in the refined grid. Arrows symbolize arithmetic averaging.

Face points

Continuing the line of thought, the process that results in the interpolated value for the face points is bilinear interpolation. In this case the value at the center of each face is given by the arithmetic mean of the 4 corners of the face.

$$f^k_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{4}\left(f^k_{ij} + f^k_{i+1,j} + f^k_{i,j+1} + f^k_{i+1,j+1}\right) \tag{3.34}$$

As before, i, j and k can be permuted to achieve the remaining two directional equations, and we maintain the under/super-script notation. From the usual description of bilinear interpolation, and explicitly from equation (3.33), this equation can be written in two other forms:

$$f^k_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2}\left(f^k_{i+\frac{1}{2},j} + f^k_{i+\frac{1}{2},j+1}\right) \tag{3.35}$$

$$= \frac{1}{2}\left(f^k_{i,j+\frac{1}{2}} + f^k_{i+1,j+\frac{1}{2}}\right) \tag{3.36}$$

In each case we choose a direction, linearly interpolate to the edge points in that direction, and then again linearly interpolate to the face point from the edge points in the remaining direction. This process is schematically represented by the blue arrows in Figure 3.3a, in which the first direction chosen was the one pertaining to the j index.

Center points

In this case the interpolated value is given by the arithmetic mean from the 8 corners of the cuboid.

$$f_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = \frac{1}{8}\big(f_{ijk} + f_{i+1,j,k} + f_{i,j+1,k} + f_{i,j,k+1} + \\ + f_{i+1,j+1,k} + f_{i+1,j,k+1} + f_{i,j+1,k+1} + f_{i+1,j+1,k+1}\big) \tag{3.37}$$

Once again, following consecutive linear interpolations in the three directions, and taking equations (3.35) and (3.36) into consideration, this can be rewritten in

three forms:

$$f_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = \frac{1}{2}\left(f_{i+\frac{1}{2},j+\frac{1}{2},k} + f_{i+\frac{1}{2},j+\frac{1}{2},k+1}\right) \tag{3.38}$$

$$= \frac{1}{2}\left(f_{i+\frac{1}{2},j,k+\frac{1}{2}} + f_{i+\frac{1}{2},j+1,k+\frac{1}{2}}\right) \tag{3.39}$$

$$= \frac{1}{2}\left(f_{i,j+\frac{1}{2},k+\frac{1}{2}} + f_{i+1,j+\frac{1}{2},k+\frac{1}{2}}\right) \tag{3.40}$$

of which the form in (3.38) is explicitly represented in Figure 3.3b.

Implementation

There are two notes worth mentioning from the discussion on interpolation so far:

- All the expressions presented are vectorizable in all indices;
- In the expressions that make use of previously interpolated points, the directions of consecutive linear interpolation can be chosen seemingly at random without altering the results.

Thus, to avoid doing the same computation twice (*e.g.* by computing $f^{jk}_{i+\frac{1}{2}}$ again when computing $f^{k}_{i+\frac{1}{2},j+\frac{1}{2}}$ if we do it by equation (3.34)), the following algorithm was implemented. Every step is vectorized *via* `numpy` arrays.

1. Initialize new array f^*;
2. Copy all points from original grid by equation (3.32);
3. Linearly interpolate to edge points by equation (3.33) and other equations derived by index permutation;
4. Linearly interpolate from the obtained edge points to face points by equation (3.36) and other equations derived by index permutation;
5. Linearly interpolate from the obtained face points to center points by equation (3.38).

This grid refinement by trilinear interpolation is done in the analysis as many times as requested by the user *via* an integer input p such that the spacings of the final refined grid are given by

$$\delta x_\alpha^{(\text{new})} = \delta x_\alpha^{(\text{old})} \times 2^{-p} \tag{3.41}$$

In practice, p should not be chosen greater than 2, keeping in mind that higher degrees of refinement do not provide more accurate results. The purpose of the interpolation process is simply to provide a greater point resolution in the 2D and 3D plots that result from the NCI analysis. Border conditions are also taken into account: if in any of the directions the borders are periodic, the original array is augmented one layer in that direction by copying the corresponding border before refinement, and this added layer is removed before outputting the refined array.

3.5 Algorithm

We can now summarize the steps taken in the analysis. For clarification, this description is separated into three parts, each one described schematically by a different section of the flowchart presented in subsection 3.5.4.

3.5.1 Input reading and general derivative calculations

First, the density array is checked for negative values, which have no physical significance and are likely due to floating point errors in either previous computation of the density, or its transcription to a cube file.[12] The negative values are zeroed, meaning any element of the array that was non-positive is identically zero after this step, and all other values remain unaltered. The border conditions in each of the unit cell directions, as well as the cell matrix, are read from the input Atoms object.

The cell matrix is then checked for orthogonality of the unit cell, as detailed in the discussion leading up to equation (3.3). If a non-orthogonal unit cell is detected, the χ matrix is computed as the inverse of the cell matrix, and spacings are computed by equation (3.11). Otherwise (if the cell matrix is diagonal), spacings are computed by (3.4), and the cell parameters – a, b and c – are extracted from the diagonal of the cell matrix.

Using the density array, the computed spacings, and the border conditions, the gradient is computed as a 4D array. In the non-orthogonal cell case, it is then reverted to the real-space gradient using the computed χ matrix *via* equation (3.15). Then, the reduced density gradient prefactor[13] is computed as $C_s\,|\nabla\rho|$. Because of zero and near-zero density values and the divergence of equation (2.14) for $\rho \to 0$, the computation of the reduced density gradient is done using a slight positive shift in density values, and is subsequently (partially) corrected for non-zero values of ρ. This method is thoroughly described in Appendix B.

A 4D array containing the six independent components of the Hessian matrix of the density for each grid point is computed *via* finite-difference, and – for non-orthogonal unit cells – reverted to real-space Hessian components, following the steps at the end of section 3.3.

3.5.2 Optional data treatment

At this point in the analysis, all of the arrays computed so far (density, RDG, Hessian components) can be sliced into smaller arrays pertaining to the real-space grid points nearing the supposed position of the interaction of interest. This slicing is done using the input origin, as well as six additional input values in the form of a 2×3 array. If the cell matrix is diagonal, the first line of this extra array has the real-space coordinates of the new origin, and the second line the real-space coordinates of the opposite corner of a cuboid which is lined up with the unit cell

[12] This is speculation on the author's part, but it seems to be the only explanation for negative density values, considering a non-relativistic scheme.

[13] See equation (2.14)

directions. On the other hand, if the cell matrix is non-diagonal, the elements of the mentioned array should be in fractional coordinates, and the overall shape of the new cell is maintained. The code then determines, using the spacings and the origin, the grid points closer to the input positions, and slices the arrays accordingly. It also updates the origin for posterior exportation into cube files, as well as the cell matrix. Although it is not always possible to know *a priori* which spatial regions will be associated with NCI, this step should be taken whenever possible for finite systems, to reduce the size of the files involved as well as the errors associated with RDG computation at low densities.

The next step is also optional and consists of a grid refinement by trilinear interpolation for all computed arrays. An optional input parameter p determines the degree of interpolation as detailed in section 3.4.

3.5.3 2D and 3D plots

We then arrive at the evaluation of the sign of the median eigenvalue of the Hessian matrix for every grid point. This step warranted an original approach, based on the direct computation of the stationary points of the characteristic polynomial of the Hessian matrix at every grid point, and is thoroughly described in Appendix C. The result is a 3D array containing the sign ($+1$, -1 and possibly 0) of the median eigenvalue everywhere in the grid. It is then multiplied by the density to create a new array which corresponds to the x-axis of the scatterplot.

Because we are only interested in non-covalent interactions, and real-space isosurfaces of the reduced density gradient may include contributions from regions of around 1 a.u. density or higher, *e.g.* from covalent interactions or pseudo-potential artifacts,[14] it becomes necessary to choose two cutoff values for negative and positive x-axis of the scatterplot, before exporting the necessary cube files for 3D visualization. These cutoffs are part of the additional input referenced in section 3.1, and they do not interfere with the 2D scatterplots, they only limit the region of the scatterplot that is visible through isosurfaces of the RDG in the 3D representation. An additional cutoff with the value of 2 is chosen for the RDG to make the exported cube file lighter, since it is only practical to represent isosurfaces of low values of the RDG. In practice, the x-axis and RDG arrays computed thus far are copied, the copies are scanned for values which do not fit the input cutoffs – typically around 10^{-2} a.u – and in both arrays, elements which correspond to grid points for which the criteria is not met in either of the arrays are replaced by a `None` value. The two arrays are then exported as cube files, using the ASE cube writing function, with the origin and cell matrix. These exported cube files are meant to be used by a third-party software (*e.g.*, UCSF Chimera [18], VMD [19]) to plot isosurfaces, along with the system geometry from *e.g.* an `xyz` file created with ASE.

Finally, all grid points are represented in a 2D plot, with coordinates given by the corresponding elements in the x-axis and RDG arrays, creating a visual representation of the x-axis *v.s.* RDG relations in the system being studied, the scatterplot. Optionally, a pseudo-potential cutoff radius for each element can be specified, so

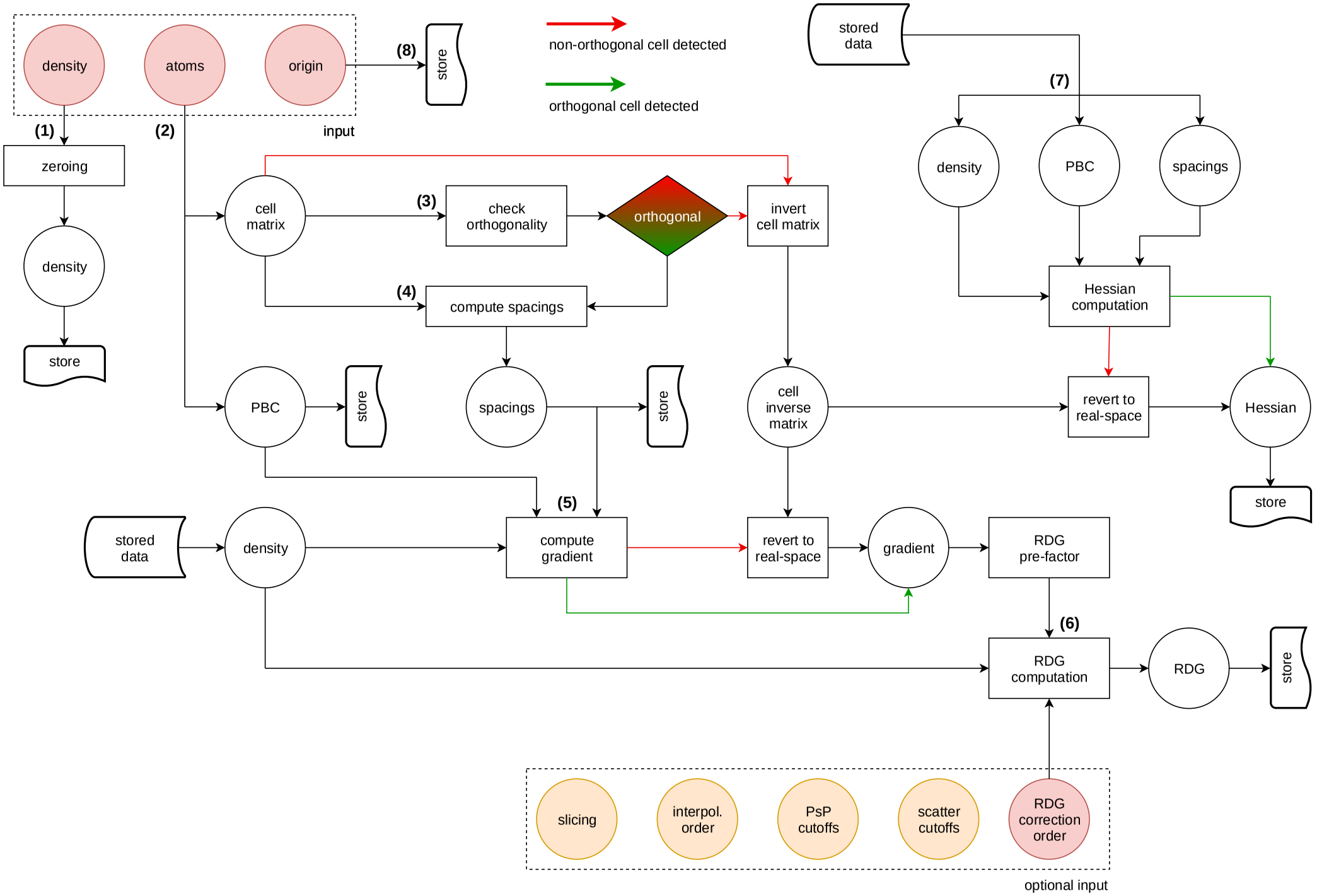
[14] This is visible in subfigure 4.6c.

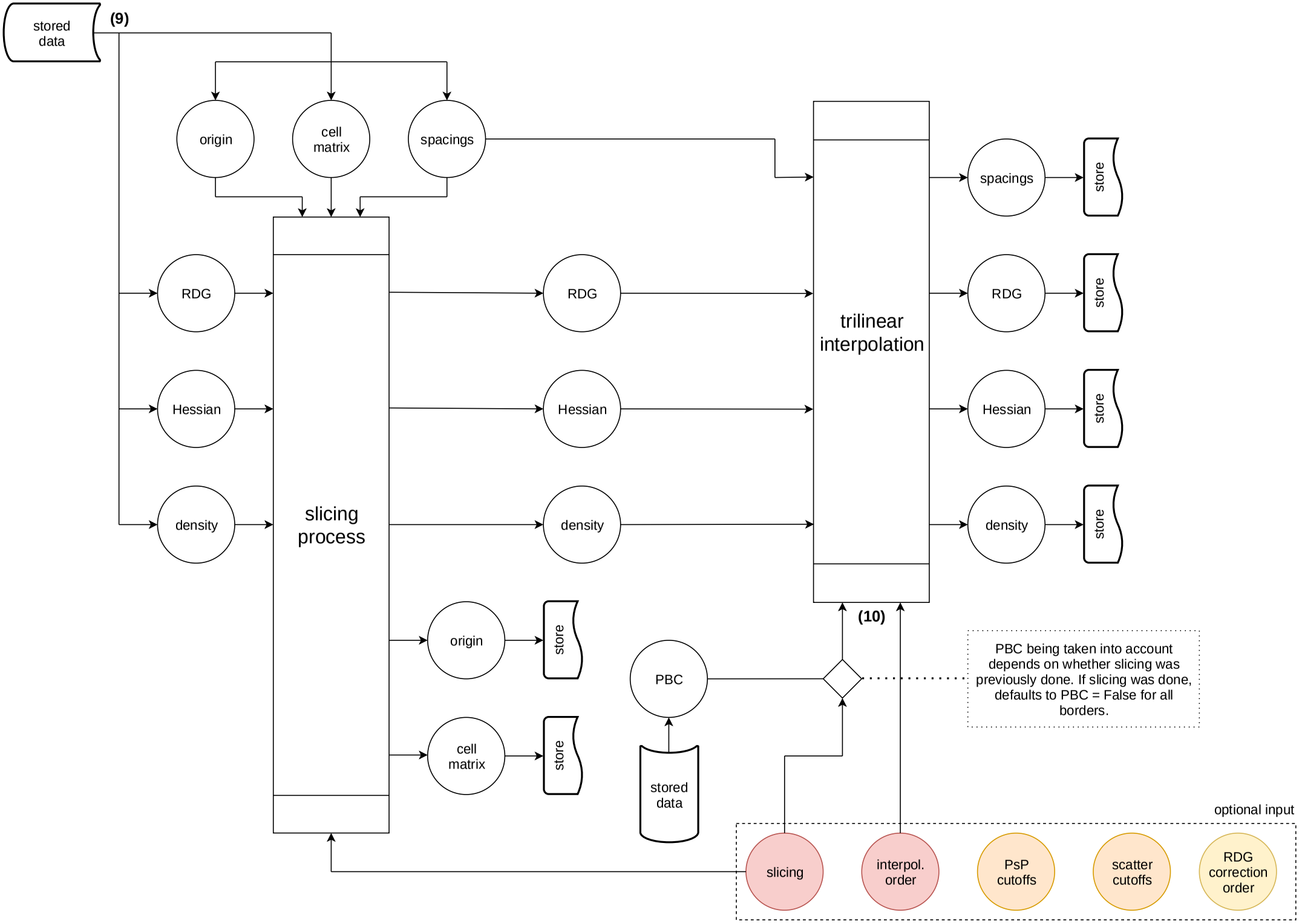
that when dealing with pseudo-densities, only grid points which are not close to nuclei are represented in the scatterplot, eliminating possible pseudo-potential artifacts. An image is exported focusing on the NCI region of these coordinates – densities below 7.5×10^{-2} a.u., and RDG below 2. Additionally, all points in the scatterplot are exported in a text file for posterior use, either with other 2D plotting software (*e.g.* Gnuplot [20]), or for a more quantitative analysis which makes use of these points. Optionally, this text file can be 'trimmed', *i.e.*, points corresponding to densities which are not shown in the mentioned image are not present in the text file. This makes the exported text file much lighter, but is not recommended if the export purpose is a quantitative analysis using the point coordinates. For the two-dimensional plots made within the code, the Matplotlib[15] python package is used.

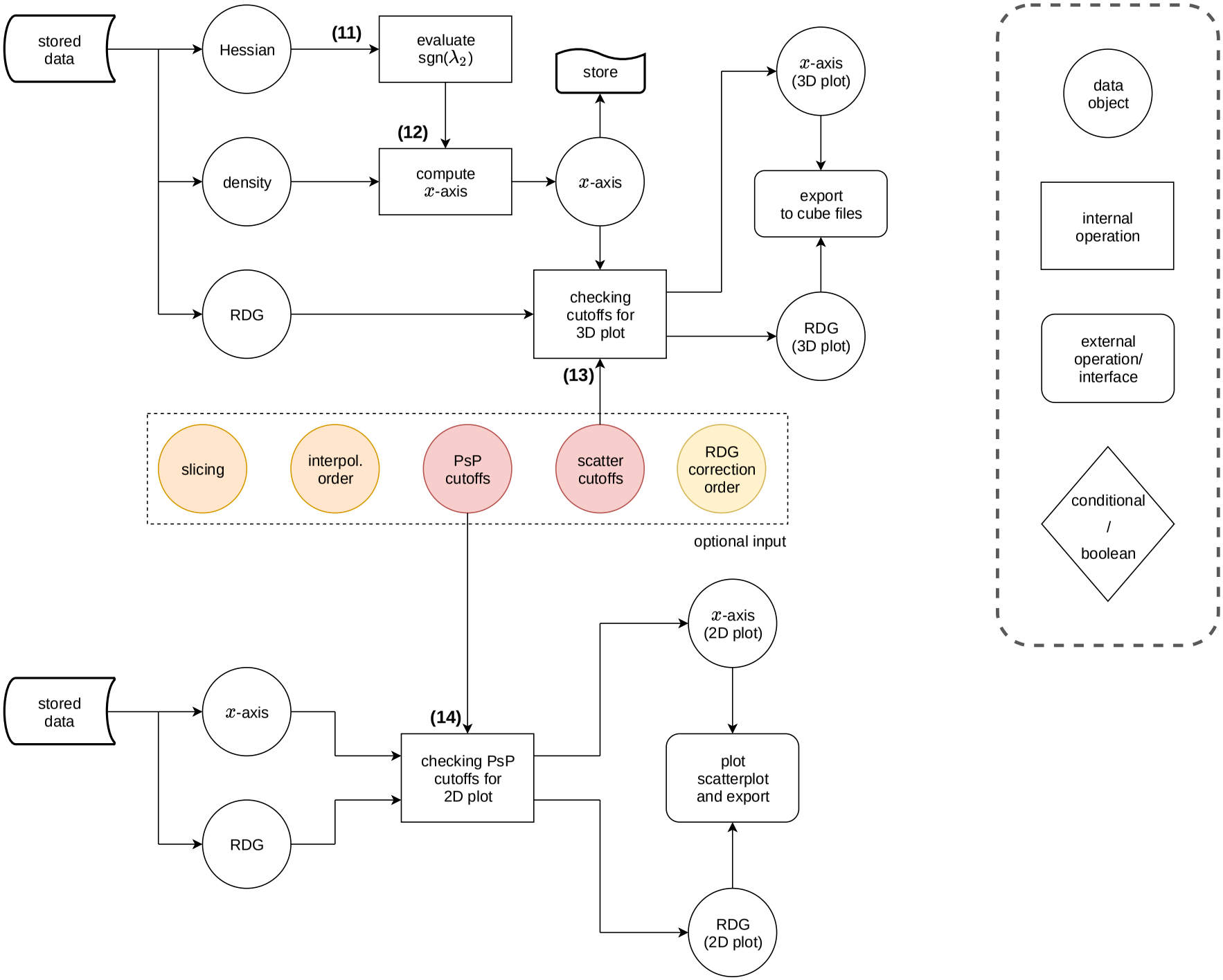
3.5.4 Flowchart

Although all optional processes are represented, they are not necessarily used. When interpreting the flowchart, one can assume the output quantities are unchanged from the input for all optional processes. The numbers represented are not referencing equations, instead specifying the order of operations in the analysis.

[15] https://matplotlib.org/


density
atoms
origin
input
(8)
store
non-orthogonal cell detected
orthogonal cell detected
stored data
(7)
density
PBC
spacings
Hessian computation
(1)
zeroing
density
store
(2)
cell matrix
(3)
check orthogonality
orthogonal
invert cell matrix
(4)
compute spacings
PBC
store
spacings
store
cell inverse matrix
revert to real-space
Hessian
store
stored data
density
(5)
compute gradient
revert to real-space
gradient
RDG pre-factor
(6)
RDG computation
RDG
store
slicing
interpol. order
PsP cutoffs
scatter cutoffs
RDG correction order
optional input


stored data
(9)
origin
cell matrix
spacings
RDG
Hessian
density
slicing process
RDG
Hessian
density
origin
store
cell matrix
store
trilinear interpolation
spacings
store
RDG
store
Hessian
store
density
store
(10)
PBC
stored data
PBC being taken into account depends on whether slicing was previously done. If slicing was done, defaults to PBC = False for all borders.
optional input
slicing
interpol. order
PsP cutoffs
scatter cutoffs
RDG correction order


stored data
Hessian
(11)
evaluate sgn(λ_2)
store
x-axis (3D plot)
(12)
density
compute x-axis
x-axis
export to cube files
RDG
checking cutoffs for 3D plot
RDG (3D plot)
(13)
slicing
interpol. order
PsP cutoffs
scatter cutoffs
RDG correction order
optional input
stored data
x-axis
(14)
checking PsP cutoffs for 2D plot
x-axis (2D plot)
plot scatterplot and export
RDG
RDG (2D plot)
data object
internal operation
external operation/ interface
conditional / boolean

Chapter 4

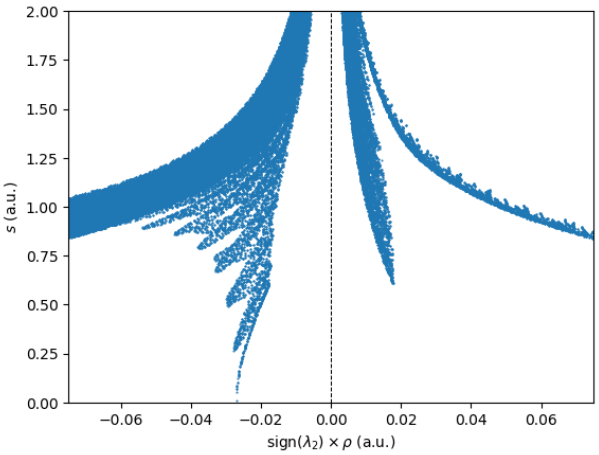
Systems and results

This chapter concerns a selection of systems for which charge densities were computed using different quantum chemistry or electronic structure software, and which were then analyzed using the developed NCI analysis code. For all cases the computational single-point calculation method was Kohn-Sham DFT. All softwares used here make use of the XC functionals as present in LibXC [21,22]. In single-point calculations, relativistic corrections should always be assumed to be non-existent (including Spin-Orbit coupling) except in the case of Group XII metal dimers and Bismuth carbide clusters. All molecular visualization plots which include isosurfaces were done using the UCSF Chimera software [18]. Schematic representations of periodic structures which include unit cells were done using the ASE [10] graphical user interface.

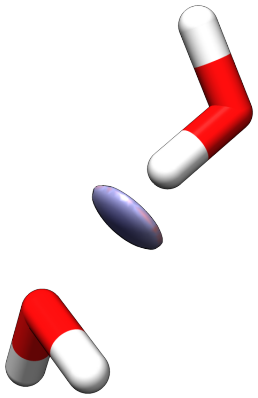
4.1 Simple test systems

The goal of this section is to provide a set of simple test systems for the developed analysis tool and to study its effectiveness with different QC/ES[1] codes and types of density computations. As such, the structures presented here were not optimized by the author, and no convergence tests were done for the density calculations: the spacings, basis sets or cutoff parameters used are based on convergence tests done on similar systems in the course of work presented here. The system geometries are part of the set distributed with version 3 of NCIPLOT [3] as example tests. The author's justification for this is the statement from reference [6] that most of the basis of the NCI index characteristics are owed to the geometry of the system, although there are visible differences after self-consistent field (SCF) relaxation. Since the systems analyzed here are dimers in which the non-covalent interactions present are relatively well known in empirical terms, it will allow us to identify the types of non-covalent interactions in later systems through characteristics of their scatterplot and isosurface representations of the NCI index analysis. Using the same geometries also allows us to reproduce to a certain extent the visual results shown in reference [1]. Lastly, the same analysis is done on a single benzene molecule in order to showcase the difference between all-electron and pseudo densities in terms of the NCI index; and the possibility of representing covalent bonds with this method, as well as a general warning and justification against doing so.

[1] Quantum Chemistry / Electronic Structure



(a) scatterplot



(b) 3D isosurface

Figure 4.1: NCI index analysis of a water dimer: (a) NCI scatterplot; (b) $s = 0.6$ isosurface, colored according to $\text{sgn}(\lambda_2) \times \rho$ following the a blue–white–red color key of minimum -7×10^{-2} a.u. and maximum 4×10^{-2} a.u.

For visual comparison between the isosurface representations of the different systems to be possible, the same value of s and color key of $\text{sgn}(\lambda_2) \times \rho$ were used for all systems, with the exception of the benzene molecule (monomer), which serves a different purpose.

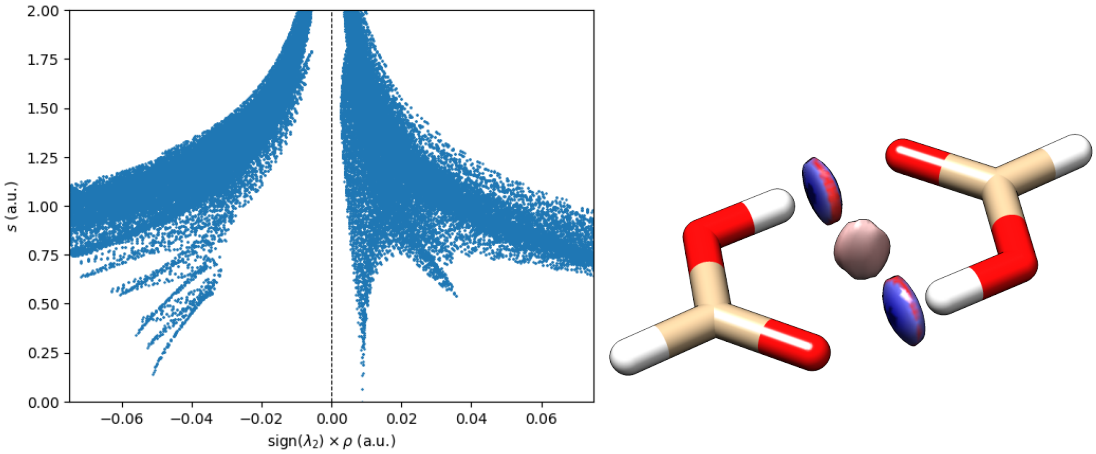
For all single-point calculations in this section, Kohn-Sham DFT with the LDA [23–25] functional as implemented in LibXC was used.

4.1.1 Water dimer

One of the most recognizable types of non-covalent bonding interactions is the Hydrogen bond, or H-bond, the prototypical example of which is the bond between water molecules in the liquid or solid state. For a system of two water molecules, the geometry of which suggests hydrogen bonding between them, a single-point energy calculation was done using GPAW [26,27] in its linear combination of atomic orbitals (LCAO) mode [28], with a DZP basis set, and a grid spacing of 0.1 Å, in a simulation box which provides a vacuum of 4 Å in all directions,[2] with non-periodic borders. The all-electron density was output as a cube file, with a grid refinement of 2, which was read and analyzed by the developed NCI analysis code. The results of this analysis are shown in Figure 4.1.

Two peak-like structures are immediately visible in the scatterplot, both related to the H-bond interaction between the water molecules, as evidenced by the isosurface in subfigure 4.1b. From left to right these peaks relate to the bonding and non-bonding contributions. This allows us to gauge the intensity of H-bonds, with a corresponding density of around 2 — 3×10^{-2} a.u., in comparison with the weaker

[2]By vacuum in GPAW it is meant that the borders of the simulation box are distanced from every atom by at least the distance specified.



(a) scatterplot (b) 3D isosurface

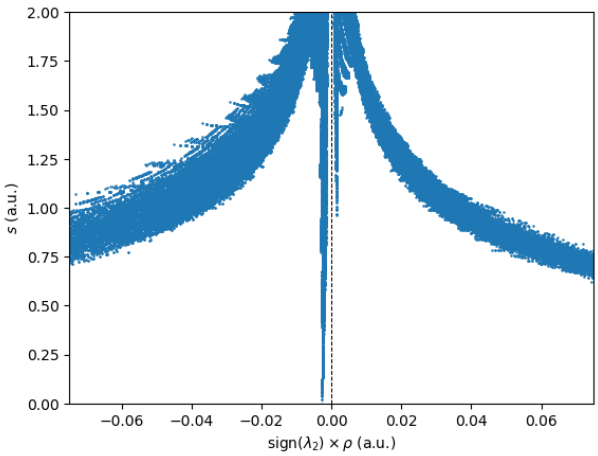
Figure 4.2: NCI index analysis of a formic acid dimer: (a) NCI scatterplot; (b) $s = 0.6$ isosurface, colored according to $\text{sgn}(\lambda_2) \times \rho$ following the a blue–white–red color key of minimum -7×10^{-2} a.u. and maximum 4×10^{-2} a.u.

van der Waals interactions which, as seen in some of the remaining systems, typically correspond to densities of around 5×10^{-3} a.u.

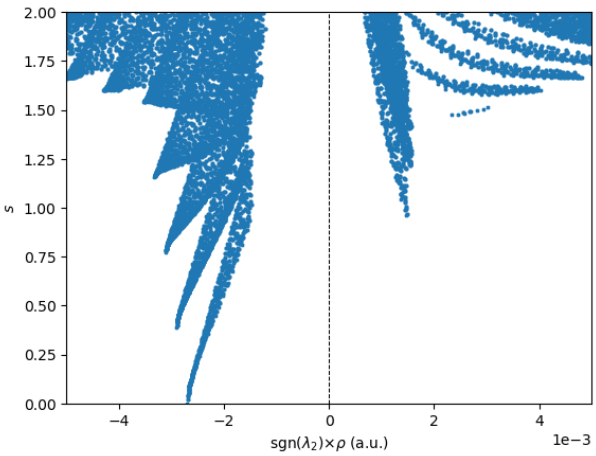
4.1.2 Formic acid dimer

Formic acid is the simplest carboxylic acid. In its vapor phase, as well as when dissolved in hydrocarbons, formic acid molecules tend to form dimers through Hydrogen bonding, and for this reason it does not obey the ideal gas law. [29] For this dimer system, a single-point spin-unpolarized energy calculation was done using Octopus [30], with a grid spacing of 0.2 a_0 in simulation spheres surrounding each nucleus with a radius of 4 Å. This is a real-space code which for ground state calculations solves the Kohn-Sham equations using finite-difference methods in a regular grid, and makes extensive use of pseudo-potentials. The (pseudo) density was output as a cube file, which was read and analyzed by the developed NCI analysis code. The results of this analysis are shown in Figure 4.2.

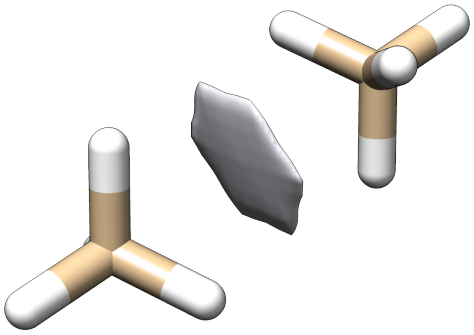
The same H-bond characteristic peak is present, albeit with a higher intensity, *i.e.*, associated with a larger density, along with its non-bonding contribution – on the right side of the plot, nearing 4×10^{-2} a.u. density. The H-bonds are clearly identified in the isosurface representation and, similarly to the water dimer in subfigure 4.1b, it is shown to be an interaction between the Hydrogen atom (donor) of one molecule and the Oxygen atom (acceptor) of the other. Along with the H-bonds, a new strictly non-bonding interaction is shown both in the scatterplot, on 'positive' densities surrounding 10^{-2} a.u., and in the isosurface representation, in the middle of the dimer. This interaction is believed to be between all atoms in both molecules, and is what is commonly referred to as a steric crowding: a repulsive interaction between the elements of the system which is counterbalanced (in this case) by the relatively more intense Hydrogen bonds. It is typically described as a 'non-bonding overlap' [1].



(a) scatterplot



(b) detailed scatterplot



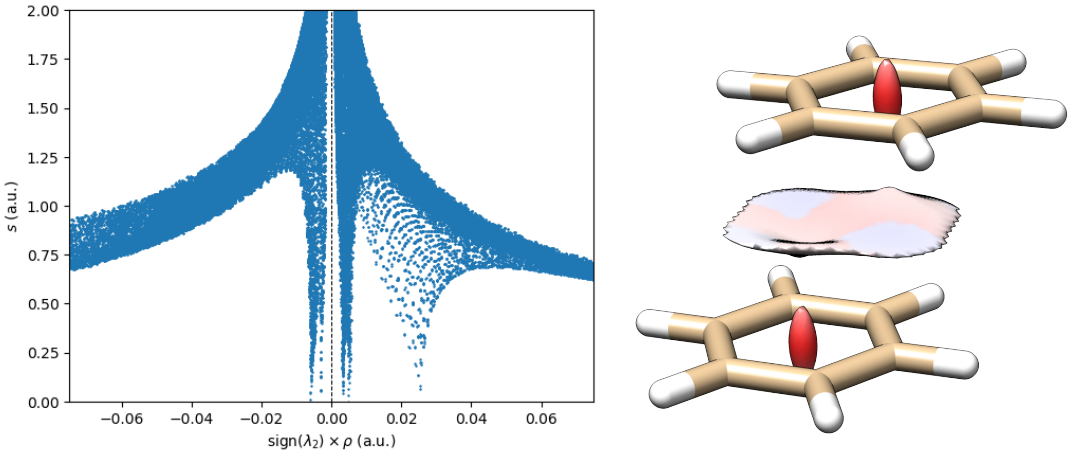
(c) 3D isosurface

Figure 4.3: NCI index analysis of a methane dimer: (a) NCI scatterplot; (b) detailed version of the scatterplot, with zoomed in x axis for densities below $5{\times}10^{-3}$ a.u.; (c) $s = 0.6$ isosurface, colored according to $\mathrm{sgn}(\lambda_2) \times \rho$ following the a blue–white–red color key of minimum -$7{\times}10^{-2}$ a.u. and maximum $4{\times}10^{-2}$ a.u.

4.1.3 Methane dimer

Octopus was again used for this calculation, with a grid spacing of 0.2 Å in simulation spheres surrounding each nucleus with a radius of 4 Å, and no spin polarization. The resulting density was output to a cube file which was read and analyzed by the developed tool. The results of the NCI analysis are shown in Figure 4.3.

Present in the scatterplot are two peaks of very low densities which correspond to a van der Waals interaction between the two molecules, as can be identified in the isosurface representation. Typically, van der Waals interactions like this exhibit very similar bonding and non-bonding contributions in the scatterplots [1], which suggests that for the specified geometry for the methane dimer, the interaction is probably slightly attractive, drawing the molecules closer, at this level of theory.



(a) scatterplot (b) 3D isosurface

Figure 4.4: NCI index analysis of a benzene dimer: (a) NCI scatterplot; (b) $s = 0.6$ isosurface, colored according to $\text{sgn}(\lambda_2) \times \rho$ following the a blue–white–red color key of minimum -7×10^{-2} a.u. and maximum 4×10^{-2} a.u.

4.1.4 Benzene dimer

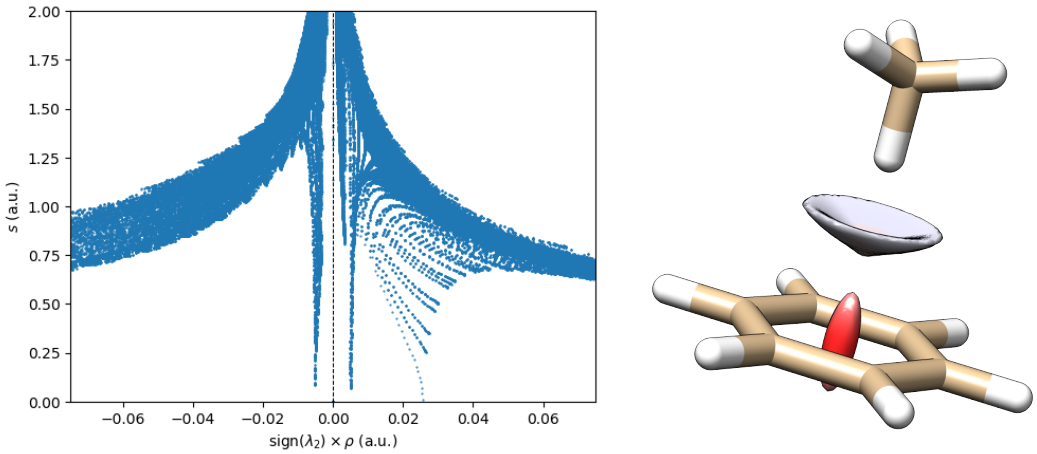
GPAW was used for a single-point calculation on this geometry using its LCAO mode with a DZP basis set, grid spacing 0.15 Å, and a simulation box with a 5 Å vacuum in all directions. The all-electron density was output as a cube file, with a grid refinement of 2, which was read and analyzed by the developed NCI analysis code. The results of this analysis are shown in Figure 4.4.

A strictly non-bonding interaction is visible in the scatterplot, corresponding to densities surrounding 2.25×10^{-2} a.u. It is clear from the isosurface representation that this is the steric crowding mentioned in the discussion for the formic acid dimer, and that it is present in the middle of the ring structures of each benzene molecule. In fact, this type of interaction is very common in ring structures in which covalent bonding is responsible for stabilization. Though it may seem a minor detail, it is worth pointing out the 'pellet' shape of the steric crowding related isosurface.

On the other hand, a van der Waals dispersion interaction can be seen both in the scatterplot for low densities, and in the 3D representation as a surface interaction between the two molecules, presenting both bonding and non-bonding contributions.

4.1.5 Methane – benzene dimer

Although the molecules which make up the dimer studied here are already known from the previous systems, and the type of interaction also already discussed (van der Waals dispersion), the way the interaction manifests itself in the isosurface representation is particularly interesting as it points out both the symmetry of the benzene molecule and the interaction between the two molecules. For this system GPAW was again used for a single-point calculation in a 5 Å vacuum simulation box with a 0.15 Å grid spacing, using its plane waves (PW) mode with an energy cutoff



(a) scatterplot

(b) 3D isosurface

Figure 4.5: NCI index analysis of a methane–benzene dimer: (a) NCI scatterplot; (b) $s = 0.6$ isosurface, colored according to $\text{sgn}(\lambda_2) \times \rho$ following the a blue–white–red color key of minimum -7×10^{-2} a.u. and maximum 4×10^{-2} a.u.

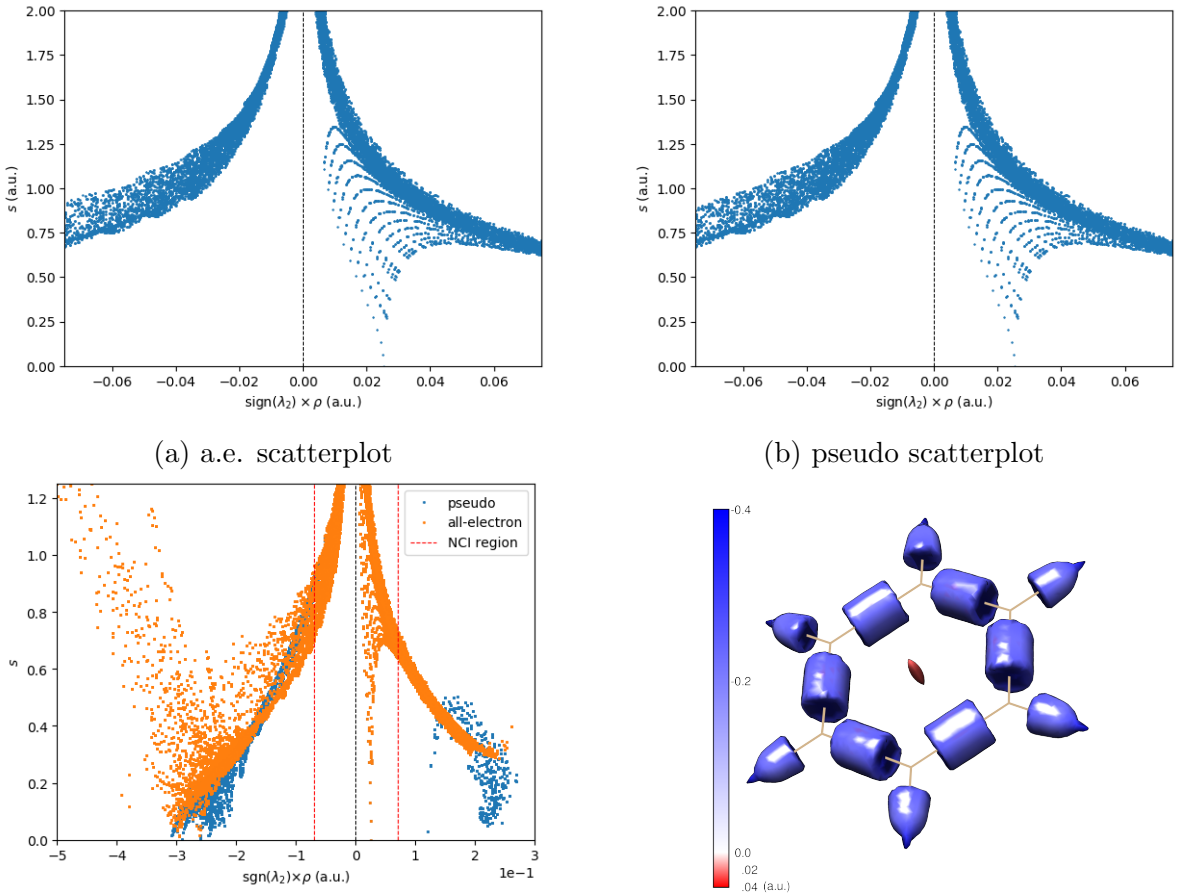
of 500 eV.[3] The all-electron density was output as a cube file with a grid refinement of 2, which was then read by the NCI analysis tool developed. The results of this analysis are shown in Figure 4.5.

As expected, the ring structure steric crowding is present in the benzene molecule. A van der Waals dispersion is also clearly visible in both the scatterplot and the isosurface representation, localized between the molecules. As mentioned, the isosurface is in this case somewhat peculiar as it evidences the symmetry of the system and the interaction. This type of cone-like surface is common in systems of van der Waals dispersion in which the interaction is 'between' one atom of a molecule with several atoms of the other [1].

4.1.6 Benzene molecule

Using GPAW's molecule building tool, a single-point calculation was made for a single benzene molecule with its default GPAW bond lengths (C–C = 1.395 Å; C–H = 1.087 Å) in a non-periodic simulation box which provides a vacuum of 4 Å in each direction, roughly 0.15 Å grid spacing, using GPAW's LCAO mode with a DZP basis set. Although GPAW internally uses pseudo-potentials, it is capable of rebuilding the all-electron density for every calculation. Thus, both the all electron and pseudo densities were output into cube files with a grid refinement of 2. Both density files were then analyzed using the developed NCI analysis tool, and the resulting scatterplots are shown in subfigures 4.6a and 4.6b. To further examine the differences in NCI index results between the types of densities, the range of the scatterplot was increased to density values up to 0.5 a.u., and plotted on the same

[3]There is nothing particular about this system in relation to the others already discussed which would make it a better candidate for a PW calculation. This mode is simply used here as a testament to the versatility of the NCI analysis method and the low dependence of NCI characteristics on the calculation mode.



(a) a.e. scatterplot

(b) pseudo scatterplot

(c) scatterplot comparison

(d) a.e. 3D isosurface

Figure 4.6: NCI index analysis of a single benzene molecule: (a) NCI scatterplot from all-electron density; (b) NCI scatterplot from pseudo density; (c) comparison of scatterplots for both density types at a higher range of density values; (d) $s = 0.25$ isosurface for the all-electron density, colored according to $\text{sgn}(\lambda_2) \times \rho$ following the detailed color key.

graph. This is shown in subfigure 4.6c.

From a visual inspection of the scatterplots for both types of densities, no differences can be detected in what is designated the NCI region, density values below $7{\times}10^{-2}$ a.u. This is an argument in favor of analyzing densities computed using pseudo-potentials, even in cases or codes for which the all-electron density is unavailable or unobtainable, keeping in mind that the correct cutoff values have to be used to avoid pseudo-potential artifacts.

In the scatterplot comparison plot, there are a few relevant aspects to point out:

- In and near the NCI region, there is no visible difference between the all-electron and pseudo density profiles.
- Outside the NCI region, there are at least two artifacts of pseudo-potential use, which are apparent covalent interactions at density values around 0.25 a.u. These correspond to minima in the pseudo-density at nuclear positions caused by the lack of core density.

- For the bonding side of the plot and for the all-electron density, the dispersion of points that starts near the end of the NCI region and continues towards greater densities is an indication that the current implementation of the finite-difference derivatives in the code is not enough to deal with atomic core density values. This is the reason why use of the NCI index for studying covalent interactions is not recommended.[4]

Another argument in favor of using pseudo-densities is their apparent smoothness, which makes them easier to analyze with finite-difference-based methods. However, although pseudo-potential artifacts can be accounted for and sometimes avoided, certain information about the pseudo-potentials used and their generation is required and not always completely available.

It is also clear from subfigure 4.6c that, if the appropriate cutoff values are used, it is possible to use the 3D isosurface representation to localize covalent bonds, as well as NCIs. This was attempted for the benzene molecule studied here, using an isosurface of $s = 0.25$ from the all-electron density, and the results are shown in subfigure 4.6d. As predicted, the covalent bonds are clear in the 3D representation, along with the expected steric crowding 'pellet'. It is worth noting the change in shape of the isosurface between C–H and C–C bonds, and the relative size difference between the isosurfaces relating to the steric crowding interaction and the much more intense covalent bonds.

[4]Besides the fact the electron localization function (ELF) approach [5] is arguably much better for describing covalent bonding, since it takes into account electron pairing.

4.2 Group XII metal dimers

Group XII metals, namely Zinc (Zn), Cadmium (Cd) and Mercury (Hg), are present in a number of structures in which non-covalent interactions involving these atoms, nicknamed 'spodium' bonds, are responsible for its stabilization [31]. These elements are also commonly studied as a natural progression in which relativistic contributions become more important with the growth of atomic number. In this section, the dimer configurations of these elements – two similar atoms interacting – are studied under the light of the NCI index. As the atomic configurations of these are all closed-shell systems[5], the interaction between the atoms is in the form of van der Walls dispersion [32]. Since these are heavy elements, one must also consider relativistic contributions, which were accounted for in the calculations as detailed in the following subsection. Overall, the geometric simplicity of these systems makes them an adequate choice for surveying the capabilities of the developed NCI analysis code.

For all systems, the ORCA [33] software was used for single-point calculations with varying basis sets, and output a cube file with the specified resolution and dimensions. These were computed so that the sides of the cuboid would be spaced 1 Å from the nuclear positions, and the spacing around 0.1 a.u. in all directions. It is worth mentioning this is not the simulation box, we simply output only this region because it is the region of interest. Figure 4.7 contains a schematic representation of the described cuboid.

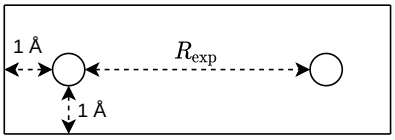


Figure 4.7: Representation of the region in the exported cube file.

4.2.1 Convergence

All single-point energy and density calculations were done using the ORCA software. The family of basis sets Sapporo-DKH3-nZP (n = D, T, Q) [34] was chosen, and was taken from the Basis Set Exchange [35–37][6]. This family of basis sets was specifically built for use with the Douglas-Kroll-Hess method [38] (DKH) for incorporating relativistic contributions by extending the Dirac Hamiltonian to systems of several particles. Additionally, Spin-Orbit coupling was handled using Spin-Orbit Mean Field (SOMF), as it is implemented in ORCA [39]. The DZP, TZP and QZP basis sets are for the purpose of convergence analysis considered separately from their augmented versions aDZP, aTZP and aQZP, which include diffuse functions that in principle more accurately describe the density in regions far away from the nuclei. Experimentally measured inter-nuclear distances from reference [32] were used. Since the property of interest is the NCI index analysis, the convergence tests

[5] Zn: [Ar] $4s^2$ $3d^{10}$; Cd: [Kr] $5s^2$ $3d^{10}$; Hg: [Xe] $6s^2$ $4f^{14}$ $5d^{10}$.

[6] Available online at https://www.basissetexchange.org/

are based on visual analysis of the scatterplots obtained for the densities computed with the different basis sets. No correction was made for superposition of basis functions. The computation method was Kohn-Sham DFT with the meta-GGA strongly constrained and appropriately normed (SCAN) XC functionals [40], for two main reasons:

- Lower level functionals like LDA are known to not accurately describe density tails, creating an artificial damping in inter-nuclear space, and they are also not ideal for describing regions of space in each the density varies very quickly. This could cause problems in the systems in question since the NCIs we wish to analyze are located in a region wish would not be accurately described at this level, and the massive gradients in core density for these heavy elements could cause problems in convergence.

- SCAN specifically has been shown to be more adequate than most functionals of the same level at describing energetics of van der Waals interactions.

Within ORCA, the DFT SCF procedure makes use of an internal grid, which was chosen to be the most refined possible in the software, to eliminate this as a possible source of lack of convergence. It is then believed that the convergence of the results is dependent only on the choice of basis set.

Zinc dimer

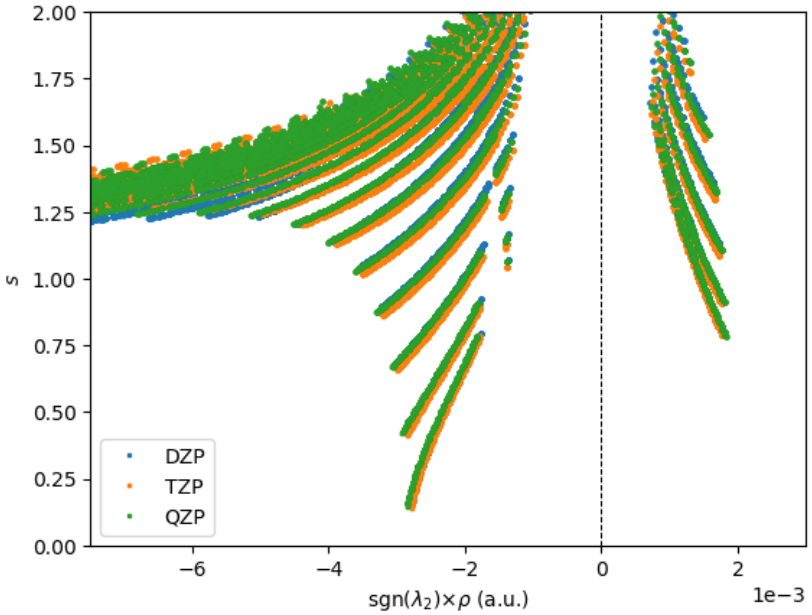
$R_{\text{exp}}^{\text{Zn}} = 4.19$ Å was used as the inter-nuclear distance. Figure 4.8 presents the scatterplot results using densities computed with the basis sets specified, zoomed into the region where non-covalent interactions are represented.

It is possible to see that in each subfamily of basis sets, there is no practical difference in the scatterplot relative to regions of non-covalent interaction, although there may be differences in regions closer to nuclear positions. However, there is a noticeable difference between the inverted peaks predicted by normal and augmented basis sets, evidenced by a slight shift in the density value for the points of minimum reduced density gradient. The shift is from lower to greater densities, when going from normal to augmented basis sets, albeit the relative difference is very small. This phenomenon is partly expected, since the augmented basis sets contain diffuse functions which allow for a greater electronic 'population' of the inter-atomic region, in which the interaction is located. However, upon inspection of the basis sets and results from the remaining dimers, there is reason to believe that this increased population is not physical, and instead an artifact of the augmented basis sets.

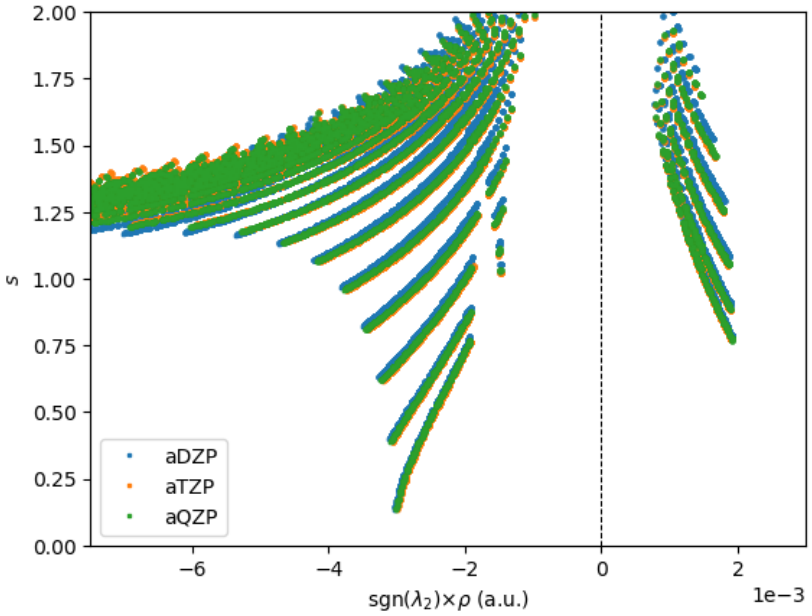
Cadmium dimer

$R_{\text{exp}}^{\text{Cd}} = 4.07$ Å was used as the inter-nuclear distance. Figure 4.9 presents the scatterplot results using densities computed with the basis sets specified, zoomed into the region where non-covalent interactions are represented.

Similarly to the Zinc case, it is visible that in the subfamily of non-augmented basis sets, there is no practical difference in the scatterplot relative to regions of non-covalent interaction, although there may be differences in regions closer to nuclear

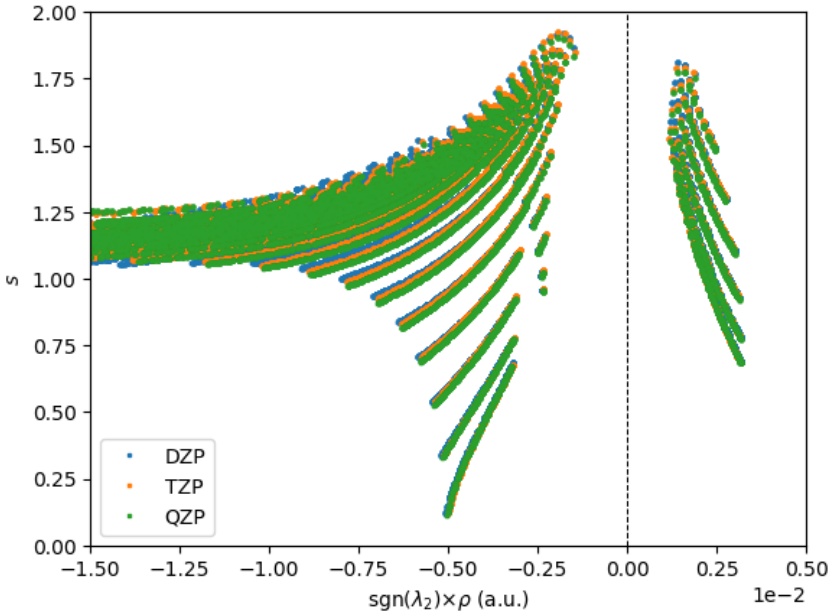


(a) normal basis sets

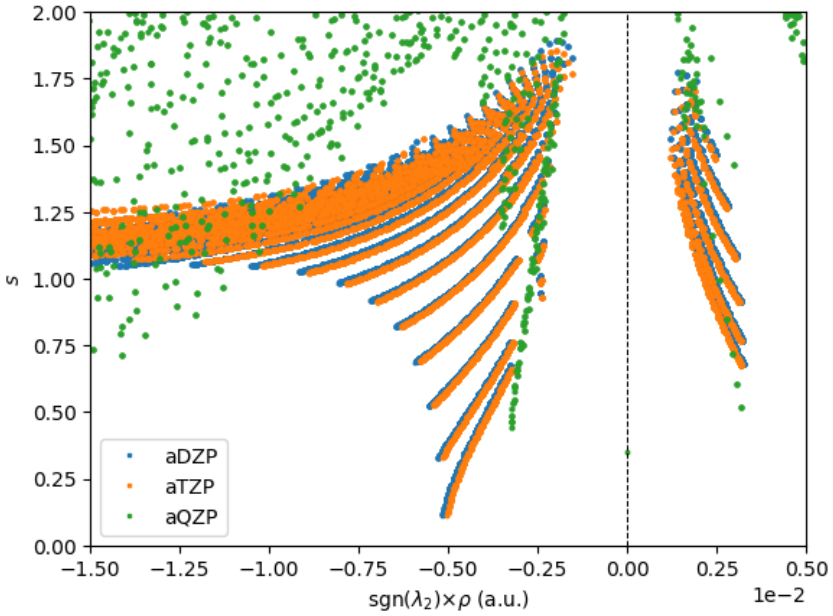


(b) augmented basis sets

Figure 4.8: Scatterplots obtained using the NCI index analysis from densities computed using different basis sets for the Zinc dimer.



(a) normal basis sets



(b) augmented basis sets

Figure 4.9: Scatterplots obtained using the NCI index analysis from densities computed using different basis sets for the Cadmium dimer.

Figure 4.10: 0.2 a.u. density isosurface for an aQZP calculation on the Cadmium dimer, obtained using the UCSF Chimera software.

positions. On the other hand, for the augmented basis sets, there is a slight difference between the results for the aDZP and aTZP basis sets and, more noticeably, a clear indication of failure of convergence with the aQZP density, even though the SCF cycle converged. This is further corroborated by an isosurface based analysis of the aQZP density, for which UCSF Chimera was used: plotting the density isosurface for 0.2 a.u., which is represented in Figure 4.10, one can distinguish several density 'cores',[7] which implies that the density does not decrease monotonically starting from the nuclear positions. Together with the fact that the slight shift in the NCI peak from aDZP to aTZP is towards lower densities, which is contrary to what one would expect for a bonded system, and the fact that that we are dealing with closed shell atoms distanced several Ångström, an argument can be made for dismissing the use of augmented basis sets in this system.

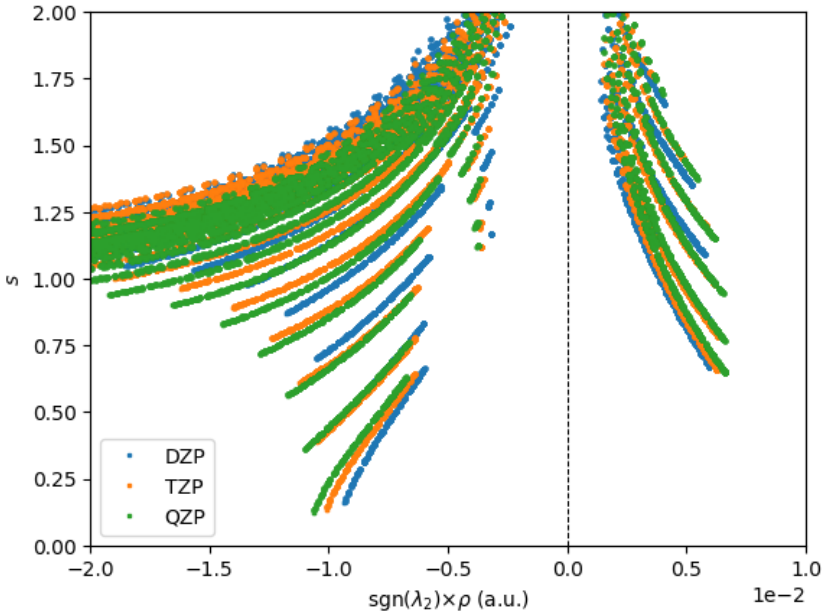
Regarding the difference between the results predicted by normal *vs* augmented basis sets, it is not clear that there is any difference between the prediction of the normal basis sets and of the aDZP basis set.

Mercury dimer

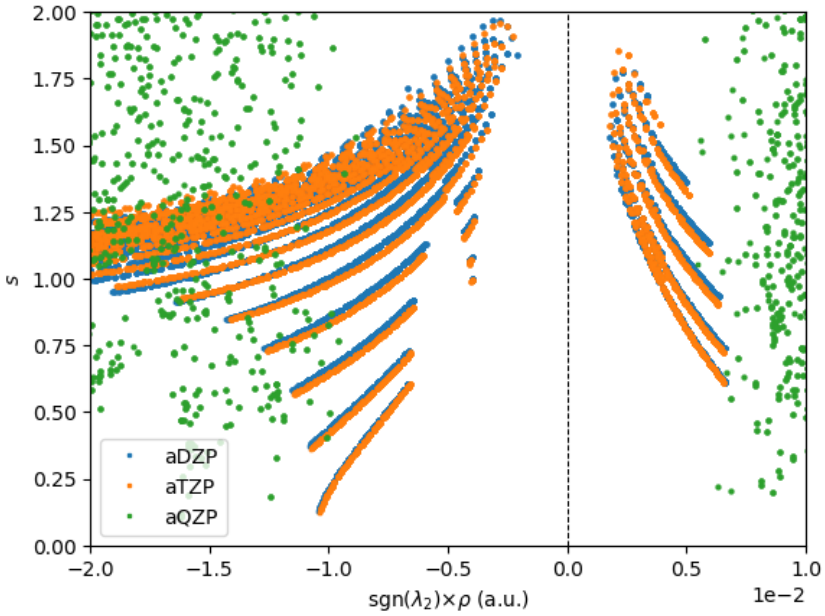
$R_{\text{exp}}^{\text{Hg}} = 3.63$ Å was used as the inter-nuclear distance. Figure 4.11 presents the scatterplot results using densities computed with the basis sets specified, zoomed into the region where non-covalent interactions are represented.

Contrary to the previous two systems, there is a clear difference between the peaks predicted by the DZP, TZP and QZP basis sets, with increasing density values. A possible explanation is the fact that the basis functions of the normal basis sets for Hg extend further since the atom has a greater number of electrons, as well as the considerably shorter inter-nuclear distance than the two previous systems. On the other hand, for the augmented basis sets, the convergence pattern is similar to the one for Cd_2, with very close aDZP and aTZP peaks of decreasing density values, as well as a clearly unconverged density for the aQZP basis set. An inspection of the aQZP density *via* isosurface visualization, similar to the one done for Cd_2, reveals the same non-monotonous density variation: a 0.14 a.u. isosurface is represented in

[7] Such a phenomenon is not visible in the densities for the aDZP or aTZP basis sets.



(a) normal basis sets



(b) augmented basis sets

Figure 4.11: Scatterplots obtained using the NCI index analysis from densities computed using different basis sets for the Mercury dimer.

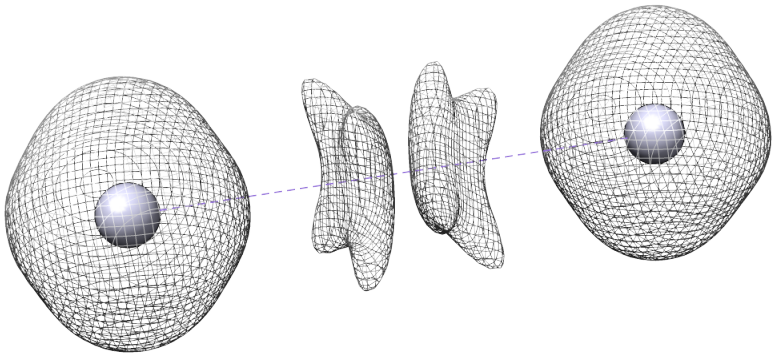


Figure 4.12: 0.14 a.u. density isosurface for an aQZP calculation on the Mercury dimer, obtained using the UCSF Chimera software.

Figure 4.12.

Again, and for the same reasons as discussed for the Cadmium dimer, these results suggest the dismissal of use of augmented basis sets.

Conclusion

For both the Zinc and Cadmium dimers, the NCI index scatterplot seems to be converged in the spatial region where non-covalent interactions are expected already at the aDZP level. Although no problems are apparent for the augmented basis sets in the case of the Zinc dimer, possibly due to the greater inter-nuclear distance and fewer electrons, a decision was made to disregard them in view of choosing the same basis set for all three systems.

A closer inspection of the basis sets suggests a possible reason for the unexpected behavior of the augmented basis sets, particularly aQZP, but is a clear justification for the dismissal of all augmented basis sets. The family of basis sets used, Sapporo-DKH3-nZP, is a contracted gaussian-based basis set which was built and tested specifically for systems in which relativistic effects have a significant contribution to the energetics and electronic structure of the system. As such, these basis sets are in principle extremely flexible due to the possibility of expansion and contraction of orbitals. There is then reason to believe that especially the QZP basis set contains functions which are diffuse enough for the density to be adequately converged. On the other hand, the augmented basis sets are identical to the 'normal' basis sets apart from the addition of diffuse gaussian functions. It is likely that this further addition of diffuse functions to a basis set that is already adequately diffuse leads to a phenomenon of linear dependence of basis set functions.

The Mercury dimer shows a significant difference between the peaks predicted by DZP, TZP and QZP. As such, the QZP basis set was selected to run comparable calculations for all three systems.

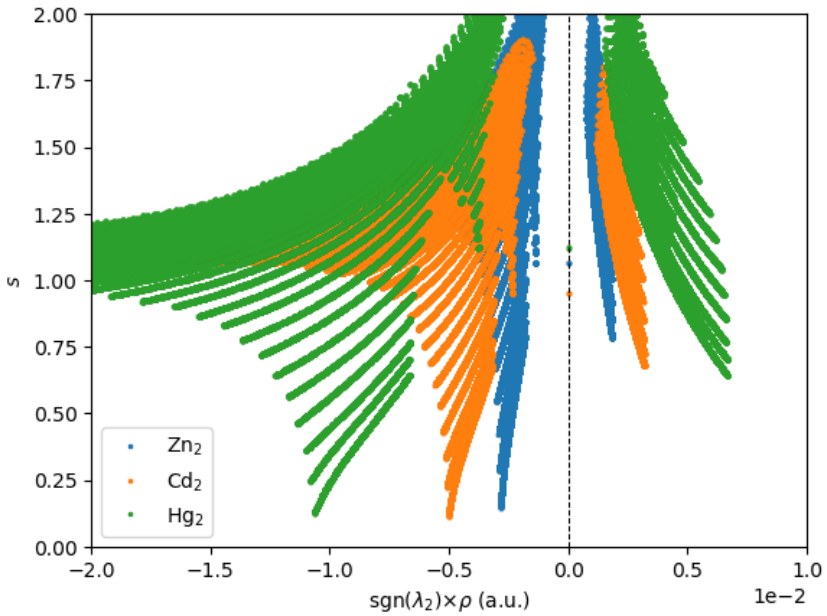


Figure 4.13: Scatterplots obtained with NCI index analysis from QZP densities for the Zinc, Cadmium and Mercury dimers with experimentally measured inter-nuclear distances.

4.2.2 Dimer comparison

Redoing the NCI index analysis on the QZP densities, now with a $p = 1$ grid refinement,[8] for all three dimers, the scatter results are the ones shown in Figure 4.13.

A few things are clear from visualizing the plot:

- For all three dimers, at the corresponding geometries, there is both a bonding and non-bonding contribution, in which the bonding contribution is more significant.
- The value of density at the NCI region, which is related to the intensity of the interaction, increases with the increase in atomic number.
- The increase in NCI region density is greater from Cd_2 to Hg_2 than from Zn_2 to Cd_2. This seems to result from several factors: (i) the increase in number of electrons is nearly double in the Cd $\rightarrow$ Hg case; (ii) the difference in inter-nuclear distance is nearly 4 times greater for Cd $\rightarrow$ Hg; (iii) there could be relativistic contributions to the interaction in the Hg dimer which are not present (to the same extent) in the other two. This type of scatterplot analysis is mostly qualitative and a further study of the atomic and dimer systems on a deeper level is required before drawing any conclusions on what the contributing factors are.

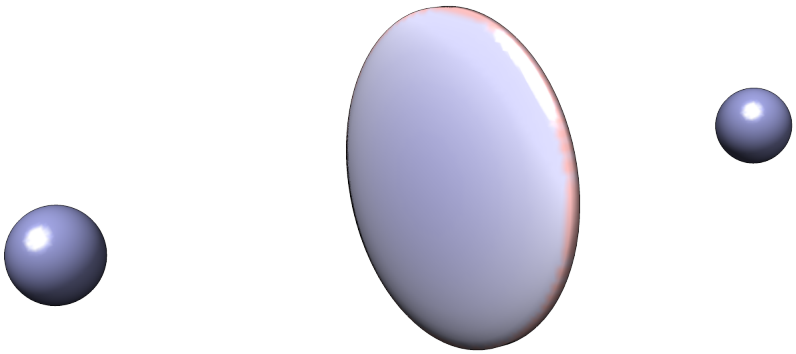
An RDG isosurface of 0.8 was plotted for each of the dimers. The results are shown in Figure 4.14, with the surface points colored according to $\text{sgn}(\lambda_2) \times \rho$,

[8] See section 3.4.

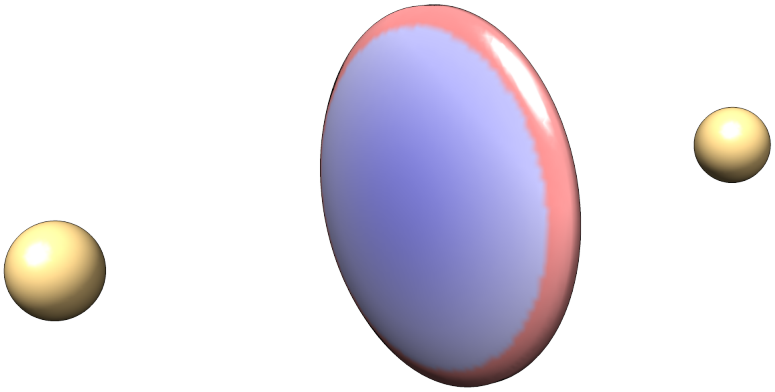
following the color key in subfigure 4.14d. This relatively high RDG value was chosen so that the surfaces include points of positive λ_2 for all three systems.

As was expected, the interaction region is located in the middle of the internuclear region, and presents axial symmetry, and the intensity of both the bonding and non-bonding contributions increased with atomic number. However, there is an unexpected effect: the region limited by the $s = 0.8$ isosurface is roughly similar for Zn_2 and Hg_2, and slightly larger for Cd_2. Several factors can contribute towards this change in isosurface size, some negatively and some positively, such as the increase in number of electrons, as well as more intense electron screening, the increase in atomic number, stronger relativistic effects, *etc.*, as well as those mentioned in the discussion of the scatterplot results.

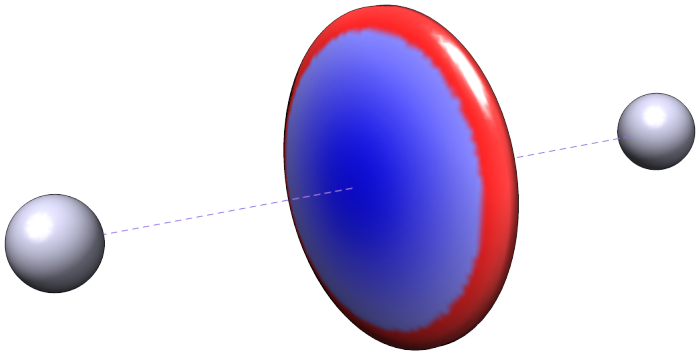
All things considered, the main point we wish to highlight at the end of this section is this: although the atoms present in the three dimers studied here have similar outer-electron configurations, there is a significant change in the way they bond non-covalently in atomic dimers, and these changes are evidenced by the differences in their scatterplot and isosurface NCI index results.



(a) Zinc dimer



(b) Cadmium dimer



(c) Mercury dimer

(d) color key

Figure 4.14: $s = 0.8$ isosurface for all three dimers with QZP densities at experimental inter-nuclear distances, colored according to $\text{sgn}(\lambda_2) \times \rho$. Images obtained using UCSF Chimera.

4.3 Periodic systems

The study of periodic systems is vital to the industry of materials design, and some novel materials have shown NCIs as either responsible for structure stabilization or simply present [41]. The systems chosen to be analyzed in this section have been the object of extensive study for decades: graphene (as a single layer), and graphite in both its thermodynamically stable (Alpha) and unstable (Beta) forms. These materials exhibit some interesting electronic properties owed to the 2D nature of graphene, but which are outside the scope of the present project.

4.3.1 Graphene

Graphene is a two-dimensional structure of Carbon atoms arranged in a honeycomb lattice. The primitive unit cell contains two atoms and can be built and arranged as shown in subfigure 3.1a. Using this unit cell, the cell parameter a can be computed from the inter-nuclear distance d as $a = d\sqrt{3}$. For energy and density calculations, however, a 3D unit cell is required. For this reason, the cell is extended in the direction perpendicular to the 2D structure and non-periodic borders are set distanced from it. In this case, the value of 20 Åwas chosen as the c cell parameter, giving the structure a vacuum of 10 Å. This is represented in subfigure 3.1b. Ideally this vacuum distance should be a target of convergence tests, but it was deemed sufficient for this unit cell considering the light nature of the Carbon atoms.

Convergence

GPAW was used for all periodic calculations, in its plane waves (PW) mode. This software uses Kohn-Sham DFT with an augmented plane waves method which makes use of pseudo-potentials. Despite their use, however, it is capable of rebuilding the all electron density after the calculation. The GGA Perdew-Burke-Ernzerhof (PBE) [42, 43] XC functional was used for all calculations, as present in the LibXC library of functionals. Only the $(0, 0, 0)$ k-point (Γ point) was sampled from the Brillouin zone. Spin-polarized calculations are the default if any of the atoms has an intrinsic magnetic moment, *i.e.*, if there are open-shell atoms. Convergence of results is mostly dependent on two parameters: the plane waves cutoff energy E_{cut}, such that, in atomic units,

$$\frac{|\mathbf{G} + \mathbf{k}|^2}{2} < E_{\text{cut}} \tag{4.1}$$

with $\mathbf{G}$ the reciprocal lattice vectors and $\mathbf{k}$ the plane wave vectors; and the grid spacing h of the internal grid in which the density is represented for finite-difference based methods in certain steps of the SCF cycle.[9]

There is a relation between the two parameters in terms of their convergence effects in a grid, derived by Briggs *et al* [44],

$$h = \frac{\pi}{\sqrt{2\frac{E_{\text{cut}}}{E_{\text{h}}}}}\, a_0 \quad \Leftrightarrow \quad E_{\text{cut}} = \frac{1}{2}\left(\frac{\pi}{h/a_0}\right)^2 E_{\text{h}} \tag{4.2}$$

[9] GPAW makes use of two internal grids, a 'coarse' grid of spacing h and a 'fine' grid of spacing $h/2$.

Table 4.1: Summary of convergence parameters for graphene.

N	h (Å)	E_{cut} (eV)
5	0.491902	77.702456
10	0.245951	310.809825
25	0.098380	1942.561408
50	0.049190	7770.245633

One could think of selecting the grid spacing and using the corresponding cutoff for plane waves using equation (4.2). This is made impossible by the software, since it asserts that the possible values of E_{cut} are limited by the value given by this equation. Indeed, if a very close cutoff is selected, *e.g.* 99% of the computed value, the convergence of the SCF cycle becomes much harder to achieve. However, looking at the default values for these two parameters, we have $h = 0.2\,\text{Å}$ and $E_{\text{cut}} = 340$ eV. The quotient of the default cutoff value and the one computed using equation (4.2) is roughly 0.36. Therefore, a decision was made to link the cutoff and spacing parameters for each calculation according to

$$E_{\text{cut}} = \frac{1}{2} E_{\text{cut}}^{(\text{computed } \textit{via } 4.2)} = \frac{1}{4} \left(\frac{\pi}{h/a_0} \right)^2 E_{\text{h}} \tag{4.3}$$

This way, only one parameter needs to be screened for convergence of results.

To study convergence for graphene, the experimental inter-nuclear distance $d = 1.42\,\text{Å}$ was used [45], which results in a cell parameter $a = 2.46\,\text{Å}$. Four single-point energy calculations were run, each with a different spacing $h = a/N$, where $N = 5$, 10, 25, 50. For each calculation, both the total energy and all-electron density grid were output.[10] Grid spacings and cutoff energies for each calculation are shown in Table 4.1.

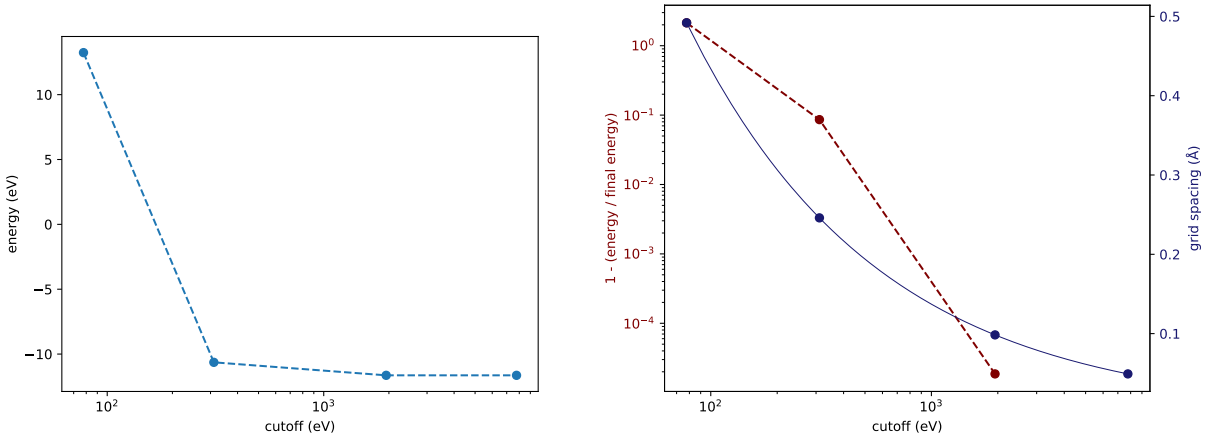
Energy convergence analysis results are presented in Figure 4.15, in terms of the cutoff energy value, in logarithmic scale.

The convergence analysis detailed here suggests that for grid spacings below 0.1 Å, energy is converged down to a relative error of less than 10^{-4}.

To analyze the convergence of density, keeping in mind the property of interest is the NCI index, an NCI index analysis was done for the density from each calculation, and a visual inspection of the resulting scatterplots was conducted, similarly to the Group XII dimers in subsection 4.2.1. The results are shown in Figure 4.16, with each scatterplot being repeated in the subfigure for the following calculation for direct comparison.

These results suggest that for grid spacings around and below 0.1 Å, the density in NCI regions is thoroughly converged.

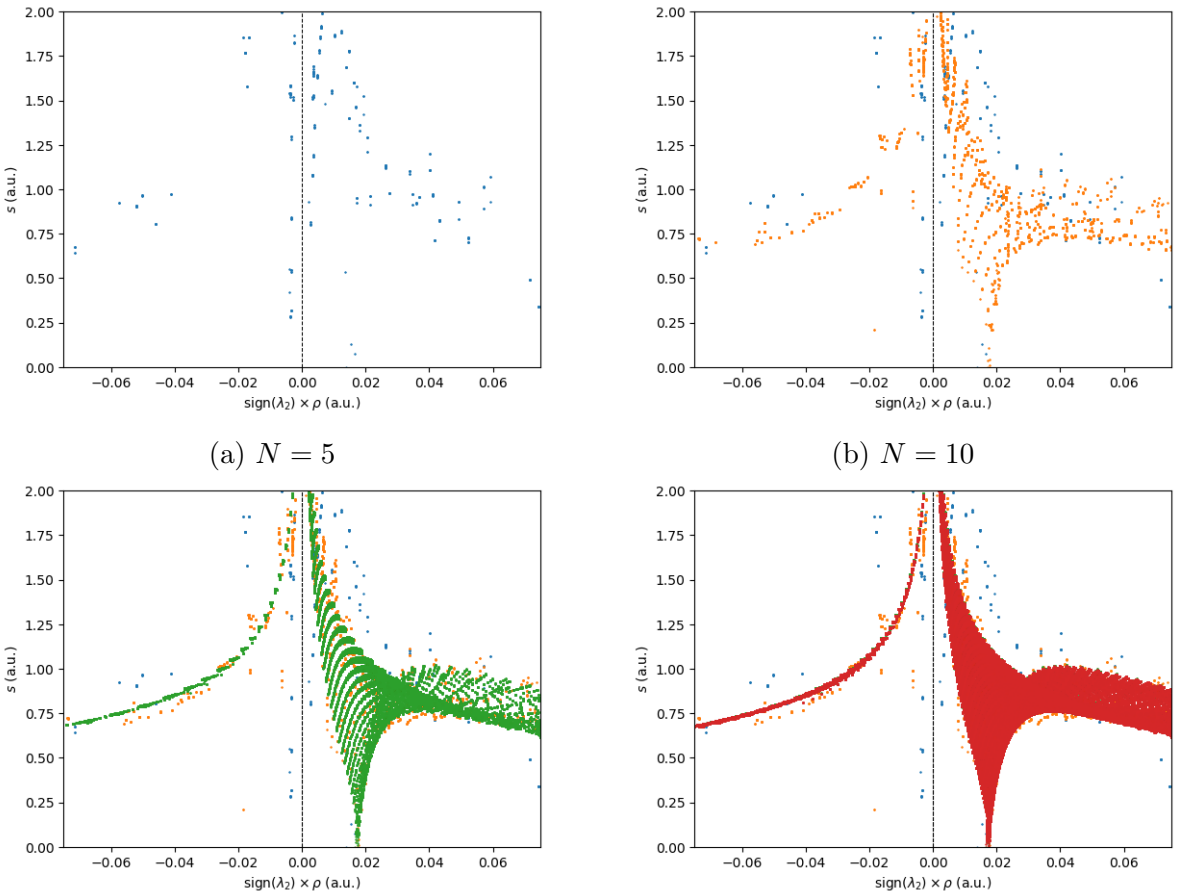
[10]For the density, the GPAW grid refinement parameter was used with value 2, so that the returned grid is the 'fine' grid with spacing $h/2$.



(a) total energy

(b) relative energy difference and grid spacing

Figure 4.15: Energy convergence analysis of graphene: (a) total energy in terms of cutoff energy used; (b) relative difference in energy with respect to last calculation, in logarithmic scale, and grid spacing.



(a) $N = 5$

(b) $N = 10$

(c) $N = 25$

(d) $N = 50$

Figure 4.16: NCI index convergence analysis for graphene.

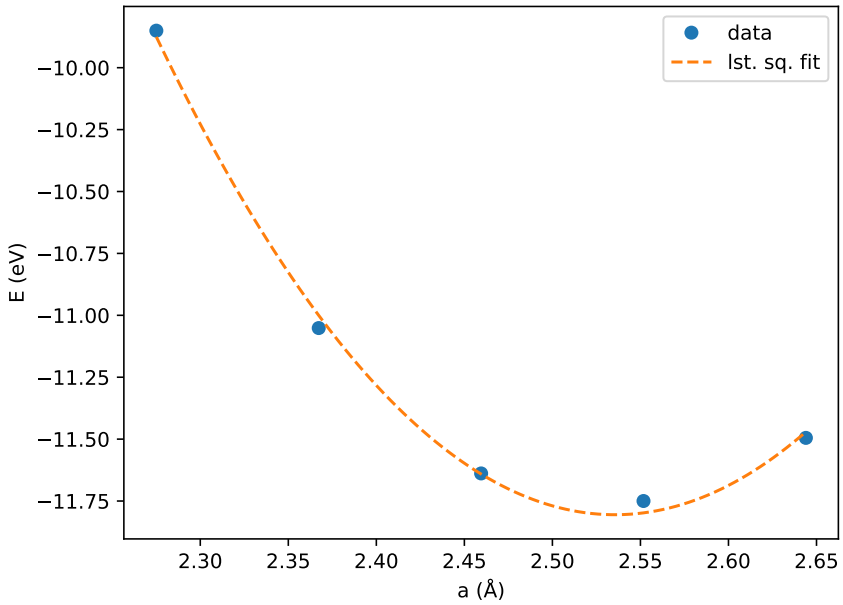


Figure 4.17: Geometry optimization attempt of graphene structure: total energy of the unit cell in terms of cell parameter a; least squares fit to quadratic curve.

Geometry optimization

The structural stability of graphene is mostly dependent on the covalent bonds between the Carbon atoms. The energetics of this covalent bonding are sufficiently described with an XC functional at the level of PBE. Thus, we can attempt to optimize the structure at the PBE level, so that the final NCI analysis of this structure may be made at the actual PBE ground-state of graphene. To do this, five single-point energy calculations were run, each with a different a cell parameter.[11] In each calculation the grid spacing is given by $h = a/25$ and the cutoff energy by equation (4.3) using this spacing. The five different a parameters were selected as

$$a_i = \left(1 - \Delta + i\,\frac{\Delta}{2}\right) a_{\text{exp}}, \qquad i = 0,\ ...,\ 4 \tag{4.4}$$

with $\Delta = 7.5 \times 10^{-2}$ and $a_{\text{exp}} = \sqrt{3} \times 1.42\,\text{Å}$. The resulting energies are shown in Figure 4.17 with respect to the cell parameter.

We assume there is a minimum in the $E(a)$ curve presented, and that the corresponding a value is near the experimental value. As such, if we expand the $E(a)$ in a Taylor series near the minimum, it is expected to have the form[12]

$$E(a) = E^* + \frac{\kappa}{2}\left(\frac{a}{a^*} - 1\right)^2 + \mathcal{O}\left((a - a^*)^3\right) \tag{4.5}$$

with a^* the equilibrium structure cell parameter, $E^* = E(a^*)$ the equilibrium energy, and κ some constant related to the curvature of $E(a)$ at a^*. Thus, ignoring terms of $\mathcal{O}\left((a - a^*)^3\right)$ and performing a least squares fit of the $E(a)$ data to this function, we get the parameters shown in Table 4.2, with a sum of residuals 6.14×10^{-3} eV^2.

[11] c was kept constant so the non-periodic border conditions would remain the same throughout.
[12] Here a^* was used to be distinguishable from the atomic unit of length, a_0.

Table 4.2: Least squares fit result for geometry optimization of graphene at PBE level

a^* (Å)	2.536
κ (eV)	365.738
E^* (eV)	−11.805

This concludes the geometry optimization of graphene at the PBE level.

NCI analysis

Using the optimized cell parameter obtained in the process described above, another single-point energy calculation was done for graphene, with spacing $h = a^*/25$ and E_{cut} given by (4.3) with this spacing. After SCF convergence, the density was output with a GPAW grid refinement of 2, and an NCI analysis was done on it. The resulting scatterplot and isosurface plots are presented in Figure 4.18.

As expected, the interaction is a steric crowding localized exactly at the unit cell edges (vertices of the 2D unit cell), which corresponds to the center of the ring structures in graphene. Although the scatterplot peak corresponding to the steric crowding is similar to that of benzene (see Figure 4.4) as was expected, the shape of the corresponding RDG isosurface is not: benzene presents 'pellet' interactions whereas in graphene the interaction isosurfaces are 'peanut'-shaped. It is unclear whether this change in the isosurface shape is caused by the periodic structure, the change in inter-nuclear distance between Carbon atoms, other factors, or a combinations of several.

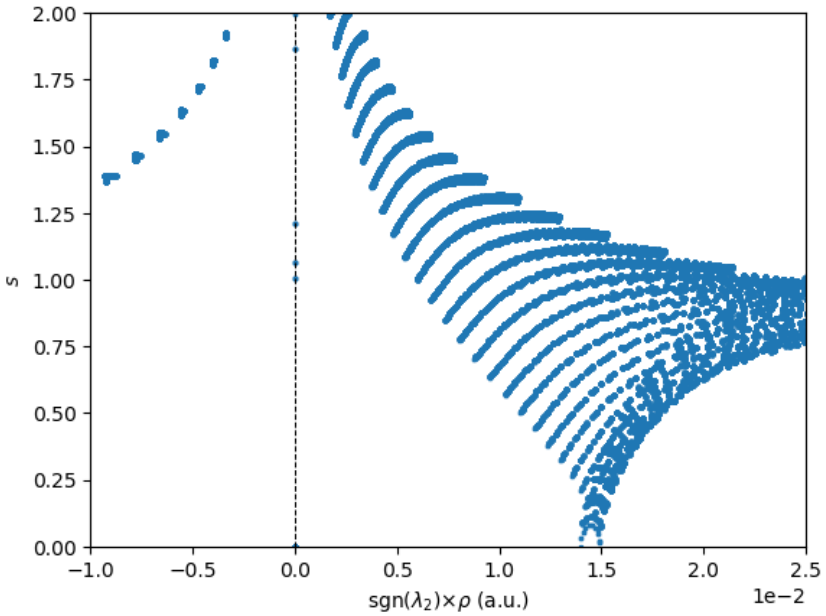
4.3.2 Graphite – Alpha (hexagonal)

Graphite in its Alpha form (hexagonal) can be viewed as a periodic extension of graphene into the direction perpendicular to its 2D structure. A possible way to build the unit cell for α-graphite from the unit cell used for graphene is to copy the two atoms, place them a certain inter-planar distance d_p above the original, move them one C–C bond length in the inter-nuclear direction – or move the original atoms and leave the new ones in the same x and y positions as the original – set the c cell parameter to $2 \times d_p$ and set the positions of the original atoms at $z = 0$ and the new atoms at $z = d_p$. This unit cell is schematically represented in Figure 4.19.

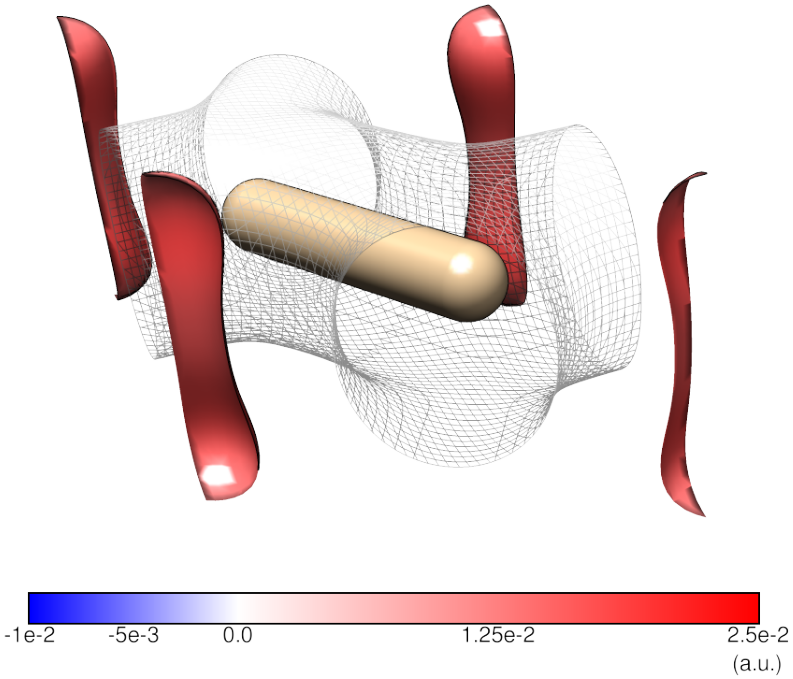
Geometry optimization attempt

An attempt at structure optimization, similar to the one done for graphene, was done with this unit cell for α-graphite. As such, nine single-point energy calculations were done with the same methods as before. The a cell parameter was kept at the experimental value, the spacing used was $h = a/25$, and the cutoff computed with (4.3). Similarly to the graphene optimization, nine values were selected for the c cell parameter, this time according to

$$c_i = \left(1 + i\,\frac{\Delta}{4}\right) c_{\text{exp}}, \quad i = 0, \ldots, 8 \tag{4.6}$$

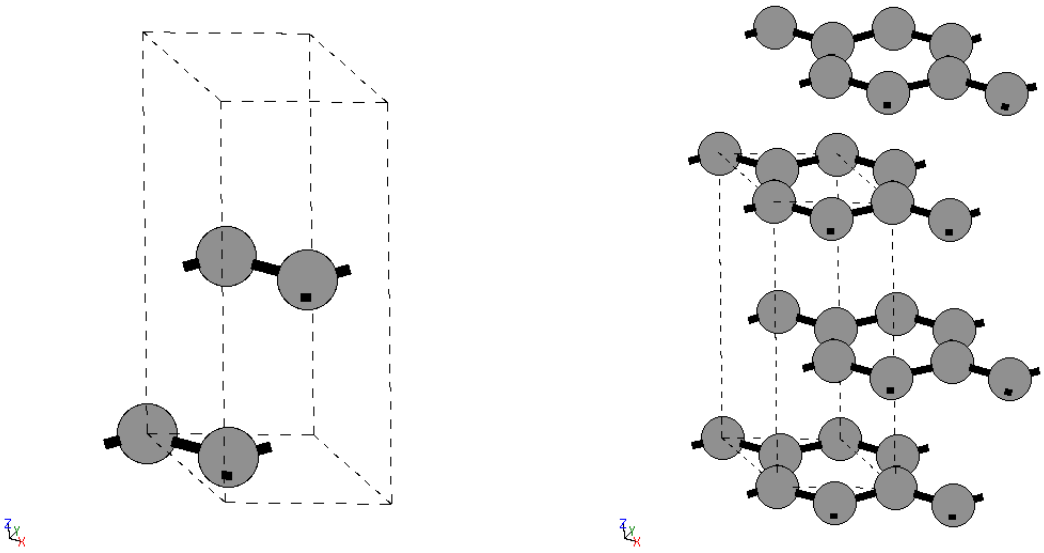


(a) scatterplot



(b) 3D isosurfaces

Figure 4.18: NCI index analysis of graphene at the optimized PBE structure: (a) NCI scatterplot; (b) $s = 0.5$ isosurface for the unit cell, colored according to $\text{sgn}(\lambda_2) \times \rho$ following the detailed color key, along with a $\rho = 0.1$ a.u. isosurface.



(a) primitive cell (b) extended cell

Figure 4.19: Schematic representation of the unit cell used for α-graphite: (a) primitive unit cell and atoms; (b) one cell extended in each direction. Image obtained with the ASE GUI.

with $\Delta = 0.5$, $c_{\text{exp}} = 2\, d_p^{(\text{exp})}$ and $d_p^{(\text{exp})} = 3.35\,\text{Å}$ [45].[13] The resulting energies are shown in Figure 4.20.

These results seem to suggest there is no minimum in the $E(c)$ curve near the experimental value, which in turn would suggest the α-graphite structure is not stable possibly due to electronic repulsion between the graphene layers. This conclusion is clearly wrong, as α-graphite is a common mineral, but was expected. The stabilization of graphite is mainly owed to the weak van der Waals non-covalent interaction between graphene layers [46], the energetics of which are known to not be accurately described at the PBE level [40]. We thus decided to use the experimental structure parameters when computing the density to be used for the NCI index analysis.

NCI analysis

Using the experimental values for the cell parameters referenced above, a single-point energy calculation was done with the same grid spacing and cutoff. After SCF convergence, the density was output with a GPAW grid refinement of 2, and an NCI analysis was done on it. The resulting scatterplot and isosurface plots are presented in Figure 4.21.

As expected, the van der Waals interaction surface spans the entirety of the plane in the middle of each graphene layer, and presents both bonding and non-bonding contributions, which is usually an NCI characteristic of this type of interaction. It seems important to point out that although in the Figure it looks as though the isosurface is represented as two 'sheets' which appear to originate from the graphene layers, they are part of a single isosurface which encapsulates a small volume between

[13] Fractional coordinates of all atoms are kept constant so that the system is scaled accordingly.

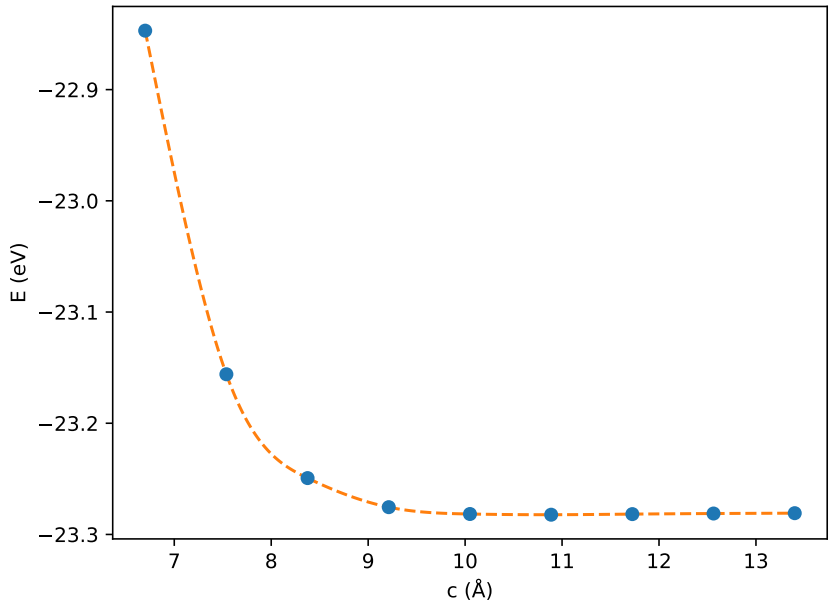


Figure 4.20: Geometry optimization attempt of α-graphite structure: total energy of the unit cell in terms of cell parameter c; natural cubic splines between data points.

each graphene layer and extends periodically in the two dimensions parallel with the graphene layers. The same 'peanut'-shaped steric crowding isosurfaces that were present in graphene as visible here as well.

4.3.3 Graphite – Beta (rhombohedral)

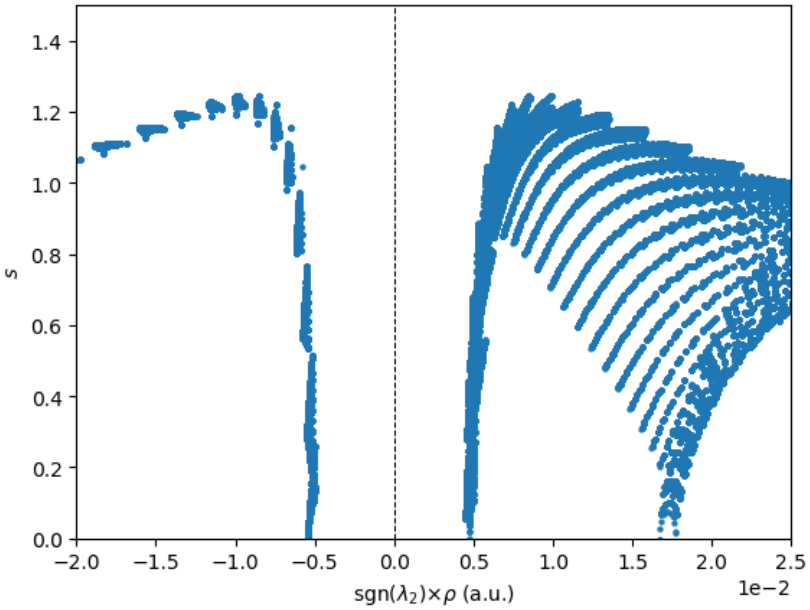
The unit cell for graphite in its Beta form was built from the α-graphite unit cell using the same process as the α-graphite cell from graphene: by copying a pair of Carbon atoms, moving them one bond length, and adjusting the c cell parameter and the z position of the new atoms. A schematic representation of this new unit cell is shown in Figure 4.22.

No attempts were made to optimize its geometry. Instead, the same a parameter and inter-planar distance d_p were used as for α-graphite, which scaled the c parameter by 3/2.

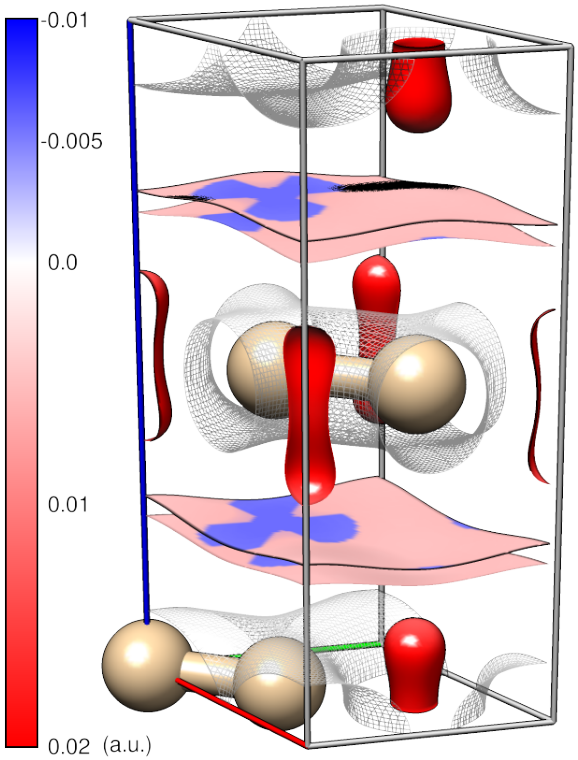
NCI analysis

As before, a single-point energy calculation was done with the same grid spacing and cutoff. After SCF convergence, the density was output with a GPAW grid refinement of 2, and an NCI analysis was performed. The resulting scatterplot and isosurface plots are presented in Figure 4.23.

As expected, the overall shape and scatterplot description of the van der Waals interactions between graphene layers are the same as the ones found for graphite in its Alpha form. There is, however, a noticeable difference in the localization of the bonding and non-bonding contributions to this interaction for each surface. In α-

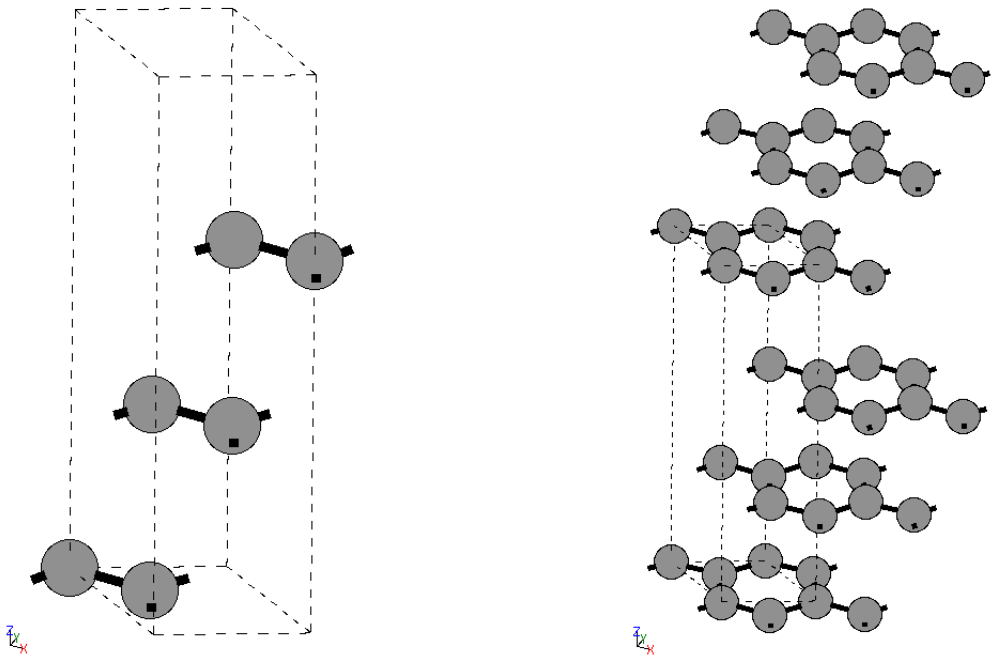


(a) scatterplot



(b) 3D isosurfaces

Figure 4.21: NCI index analysis of α-graphite at the experimental structure: (a) NCI scatterplot; (b) $s = 0.5$ isosurface for the unit cell, colored according to $\text{sgn}(\lambda_2) \times \rho$ following the detailed color key, along with a $\rho = 0.1$ a.u. isosurface.

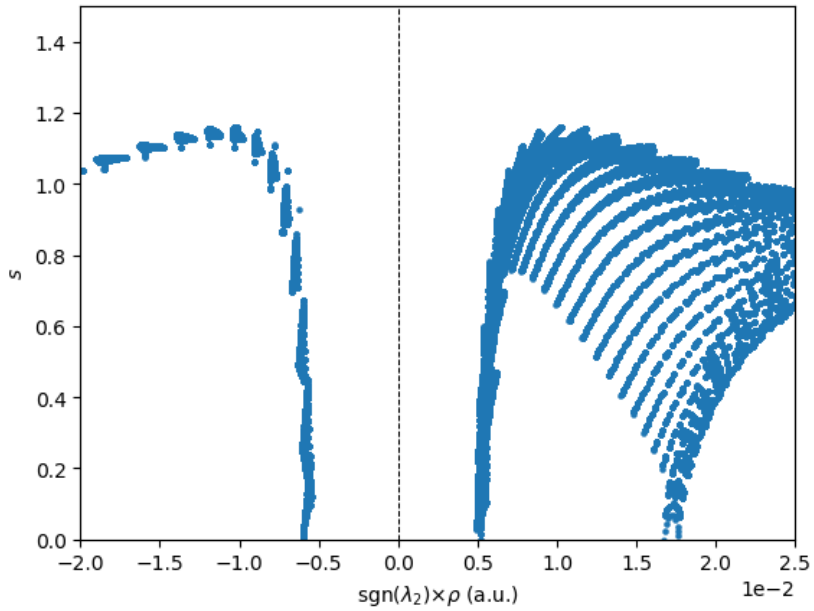


(a) primitive cell (b) extended cell

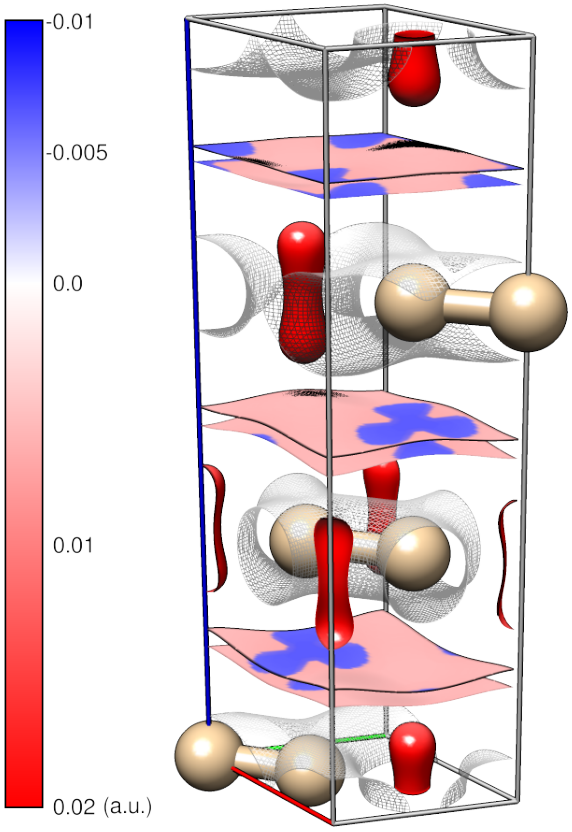
Figure 4.22: Schematic representation of the unit cell used for β-graphite: (a) primitive unit cell and atoms; (b) one cell extended in each direction. Image obtained with the ASE GUI.

graphite, since there are only two different configurations of graphene layers, there is only one type of inter-layer interaction surface: as visible in Figure 4.21b, the localized bonding contribution is always directly between the two atoms which are shown to have the same x and y positions. On the other hand, in β-graphite, since there are three layer configurations – the stacking is of the type ABCABC – there are now three different inter-layer interaction surfaces with bonding contributions localized in different x and y positions in the unit cell, always between two Carbon atoms which are stacked directly on top of each other in subsequent layers. This could be a possible reason, or at least a partly contributing factor, for the rarity of this type of graphite.

The steric crowding interactions, which were not expected to change, do not.



(a) scatterplot



(b) 3D isosurfaces

Figure 4.23: NCI index analysis of β-graphite at the experimental structure: (a) NCI scatterplot; (b) $s = 0.5$ isosurface for the unit cell, colored according to $\text{sgn}(\lambda_2) \times \rho$ following the detailed color key, along with a $\rho = 0.1$ a.u. isosurface.

4.4 Bismuth carbide clusters

The NCI analysis presented here was done by the author as a contribution to an article which at the time of writing is being submitted to the Journal of Organometallic Chemistry, in conjunction with Dr. Bruce F. Milne and Dr. Micael J. T. Oliveira, who were also responsible for the non-NCI-related work presented in the article.

4.4.1 Introduction

Although transition metal cluster-type structures have been subject of some research for a few years [47–50], specifically those of transition metal carbides, being investigated for applications in areas such as catalysis in chemical production and fuel cell design [51–53], in recent years interest has turned to the later p-block metallic elements[14] [54,55]. Not only do these elements participate in reactions very differently from the transition metals, they can also be expected to display unique properties due to strong relativistic effects owed to the presence of very heavy nuclei.

Relatively early examples of this class of main group metal carbides are the range of cationic cage-like Bismuth carbide metalcarbohedrons $Bi_nC_{2n}{}^+$ which were synthesised using a gas-aggregation source by Yamada and Nakagawa (2009) [56]. The creation of metal carbides containing such a heavy element is of considerable interest as it provides an opportunity to study the properties of (p-block) metal carbides taken to the extremes of difference in masses between their metal and carbon constituents. A characterization of the Bismuth carbides thus produced involved mass spectroscopic analysis of the resulting gas stream combined with DFT calculations of the relative energies of possible structural isomers. While this permits a tentative prediction of the species actually produced by the gas-aggregation method, a more complete spectroscopic analysis would be required in order to positively identify the cluster/cage geometries.

In the article, we present theoretical calculations of vibrational and optical absorption spectra for the various structural isomers posited by Yamada and Nakagawa. Spectroscopic features that will permit clear characterization of these cluster types were identified and suggest fundamental differences in the bonding displayed by these clusters and, in particular, within C_2 dicarbon units. Furthermore, analysis of the properties of the charge density of these clusters support the spectroscopic observations regarding bonding and help to rationalize the observed relative yield of clusters of differing masses.

4.4.2 Systems analyzed

The experimental study by Yamada and Nakagawa [56] was done on a group of synthesized cationic metalcarbohedrons of the form $Bi_nC_{2n}{}^+$, with n ranging from 3 to 9. The authors performed DFT calculations to predict low energy structures corresponding to each of the cluster masses experimentally observed. The geometry of these systems is shown in Figure 4.24.

[14] Metallic elements in Groups XIII to XVIII of the periodic table.

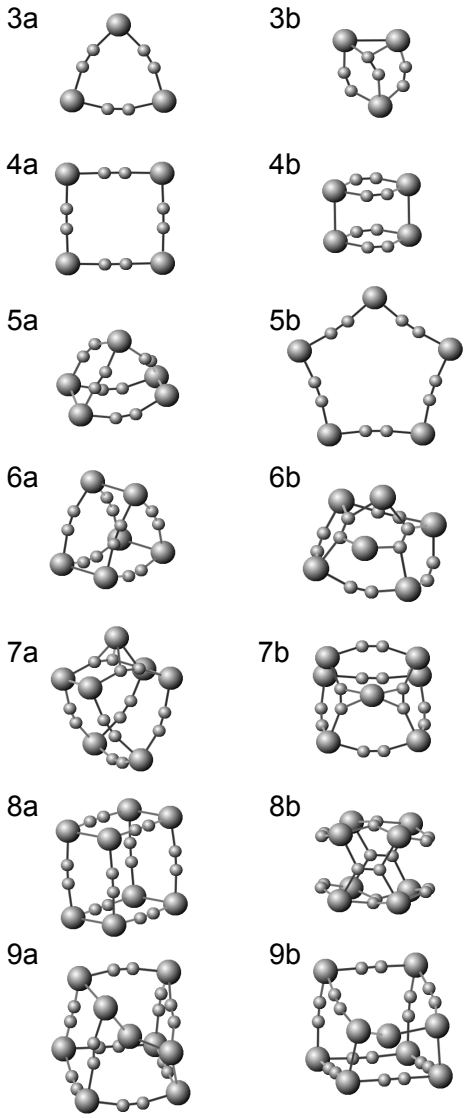


Figure 4.24: Bismuth carbide cations investigated in this work. Large spheres = Bismuth, small spheres = Carbon. Molecular graphics created using Marvin (www.chemaxon.com)

The geometries presented were optimized by Bruce Milne and Micael Oliveira. Optimizations and vibrational analyses/calculations of infra-red (IR) spectra were performed at the DFT/B3LYP level [57, 58] using GAMESS-US (11 August 2011 (R1) release) [59]. The polarized triple-ζ Def2-TZVP basis set [60] was used in all B3LYP calculations. Scalar relativistic effects were included in the B3LYP/Def2-TZVP calculations through the use of effective core potentials (ECP)[15] for Bi atoms [61]. Spin multiplicities used are the same as those in the article by Yamada and Nakagawa [56]. Relative energies for the 'a' and 'b' forms of each $Bi_nC_{2n}{}^+$ cluster were calculated including a correction for zero-point energy (ZPE). The B3LYP/Def2-TZVP ZPE was scaled by 0.965 in line with the data contained in the NIST CCCBDB database (http://cccbdb.nist.gov/), in order to produce accurate energetics for the various candidate structures. IR spectra were processed using the Gabedit software package in order to visualise the calculated vibrational modes and to fit Lorentzian curves to the energies and intensities calculated with GAMESS-US [62]. The optical absorption cross-sections were calculated using time dependent density functional theory (TDDFT), [63, 64], in particular, the real-time method as implemented in the Octopus code [65–67] using the PBE XC functional.

For bonding analysis, including NCI index, only four representative structures of the 14 presented were considered, namely the ones indexed by '3a', '4a', '6a' and '8a'. For these systems, spectroscopic analyses indicated several similarities in the spectra of $Bi_nC_{2n}{}^+$ and $Bi_{2n}C_{4n}{}^+$ ($n = 3, 4$).[16] The similarities are also apparent in the geometry of the systems, as each a-$2n$ conformation appears to be 'made up' of two a-n conformations stacked together. All-electron densities were then calculated with the SCAN meta-GGA functional [40] using the B3LYP/Def2-TZVP geometries. Scalar relativistic effects were incorporated with the second-order Douglas-Kroll-Hess (DKH2) method [68–72] and the spin-orbit contribution was calculated using the one-centre spin-orbit mean field (SOMF(1X)) [39] approach, similarly to the previously mentioned Group XII metal dimers in section 4.2. Basis sets re-contracted for use in relativistic calculations were employed (DKH-Def2-TZVP [73] for Carbon and SARC-DKH-TZVP [74] for Bismuth). Coulomb integrals were approximated with the resolution of the identity approach using the Def2/J and SARC/J auxiliary basis sets. [75, 76] Electronic charge densities were output in Gaussian cube format for analysis with the PyNCI program. All SCAN calculations were performed with the ORCA software package version 4.2.1. [33].

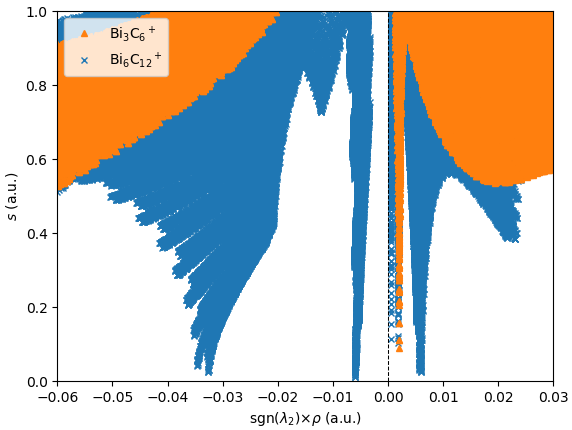
Covalent bond analysis using the electron localization function (ELF) [5] evidenced the strong covalent bonding between Bi and C atoms, and between C atoms in the same ring structure, but no covalent bonding was detected between any atoms of one ring with atoms of the other. As such it can be concluded that the interaction that leads to stabilization of the double ring structures is non-covalent in nature.

4.4.3 NCI analysis

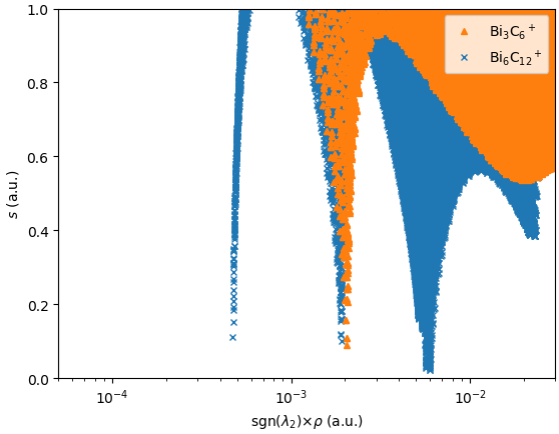
The computed all-electron densities were analyzed with the developed implementation of the Johnson *et al* [1] method, the results of which are shown in Figures

[15] Pseudo-potentials are generally named effective core potentials in the Chemistry community.

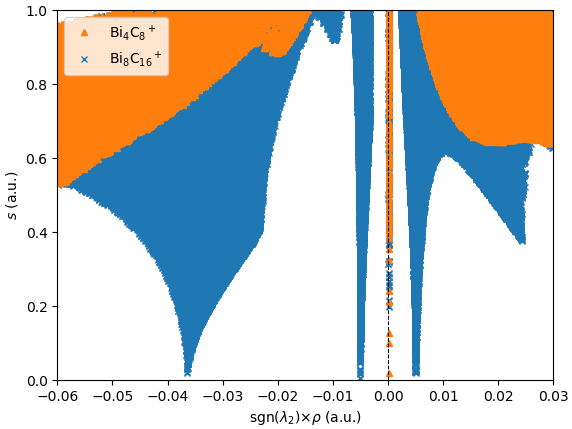
[16] Both the IR and the optical absorption spectra referenced are presented in Appendix D.



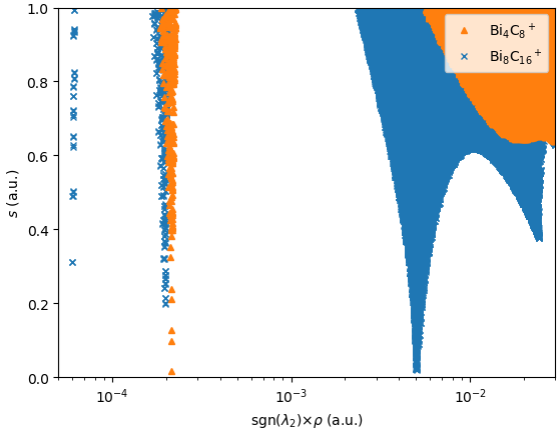
(a) $n = 3$, full



(b) $n = 3$, right



(c) $n = 4$, full



(d) $n = 4$, right

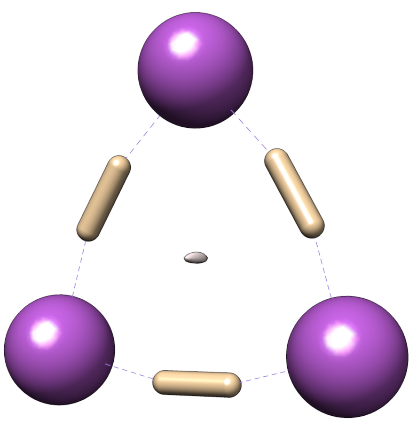
Figure 4.25: NCI index analysis of Bismuth carbide clusters: (a,c) scatterplots comparing $Bi_nC_{2n}{}^+$ and $Bi_{2n}C_{4n}{}^+$ systems and (b,d) the right side (non-bonding) of the scatterplots in logarithmic scale for $\text{sgn}(\lambda_2) \times \rho$.

4.25 and 4.26. In Figure 4.25, the non-bonding (positive $\text{sgn}(\lambda_2) \times \rho$) side of the scatterplots is shown again separately in logarithmic scale in order to emphasize the details of peaks pertaining to steric crowding in the center of ring structures.

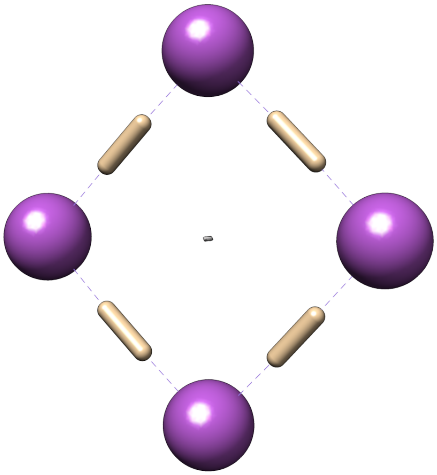
From both NCI index representations, several types of NCIs are visible:

- Strong non-covalent interactions between the Bi atoms of each ring structure. These are likely the largest contribution to the stability of the structure.
- Weak van der Waals dispersion with similar bonding and non-bonding contributions between the bicarbon units of each ring structure.
- Non-bonding interactions resulting from steric crowding at the center of each ring structure, as well as at the center of the double ring structures. We note that these are significantly less intense than those seen for example in Benzene[17], and are even weaker than the van der Waals interactions seen in the same structure.

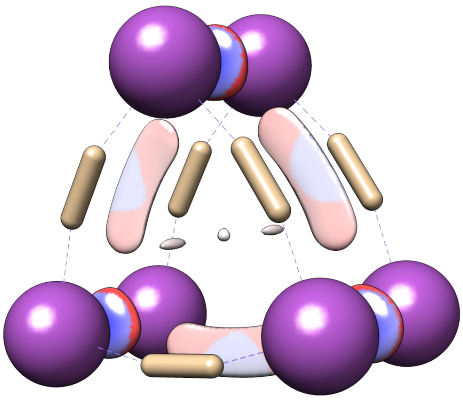
[17] See section 4.1



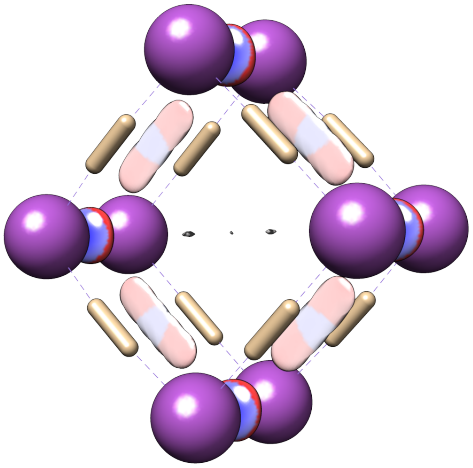
(a) $Bi_3C_6^+$



(b) $Bi_34C_8^+$



(c) $Bi_6C_{12}^+$



(d) $Bi_8C_{16}^+$

Figure 4.26: NCI index analysis of Bismuth carbide clusters: $s = 0.45$ isosurface for several systems, colored according to $\text{sgn}(\lambda_2) \times \rho$, following a blue-white-red color key of minimum -$6{\times}10^{-2}$ a.u. and maximum of $3{\times}10^{-2}$ a.u.

4.4.4 Discussion

The data obtained in the current work help to rationalize the similarities in the calculated spectra of certain pairs of complexes such as $Bi_3C_6^+$(a) and $Bi_6C_{12}^+$(a) or $Bi_4C_8^+$(a) and $Bi_8C_{16}^+$(a) by showing that in fact the heavier complexes are essentially non-covalent dimers of their lighter counterparts and might therefore be expected to behave very similarly from the point of view of spectroscopic analysis. This might also go some way to explain the lack of these heavier complexes in the product mixture obtained from the gas-aggregation experiments; if these are indeed dimers bound by non-covalent interactions then it is not unreasonable to expect them to be less abundant than the corresponding monomer units.

Using conventional approaches for the description of bonding processes in these systems would be considerably difficult, given that some of the monomer and dimer configurations have different spin states. However, the conjoint use of ELF and NCI index for covalent and non-covalent interactions respectively allows us to visually rationalize the dimerization process through which the more complex structures are formed in a more intuitive manner.

Chapter 5

Conclusion

We have succeeded in building an easily usable and fast Python implementation of Johnson *et al*'s NCI index analysis [1]. Its visualization aspects, both two- and three-dimensional, were successfully tested in several types of systems – molecular dimers, atomic dimers and periodic structures – with DFT densities computed using different methods and codes, only from the cube files exported at the end of SCF calculations. Within the project, several specific problems were found and solved accordingly, which include, but are likely not limited to: pseudo-potential artifacts from using pseudo-densities, computation of finite-difference derivatives in non-orthogonal grids, lack of resolution in scatterplots and isosurface representations (solved with trilinear interpolation), computation of the RDG in arrays which include non-positive density values, and computational efficiency when evaluating the sign of the median eigenvalue of the Hessian matrix of density for every point in a grid. Furthermore, some of the results obtained with this implementation have aided the study of metal carbide clusters by supporting the hypothesis proposed by TDDFT theoretical spectroscopy analyses of a dimerization process through which more complex structures are built from similar but simpler systems.

Future work

The most recent version of NCIPLOT [4] (2020) includes thoroughly tested tools for quantitative analysis of non-covalent interactions built upon the visualization aspects similar to those developed here, by making use of multi-grid methods and defining NCI volume regions in dimers by splining the scatterplot for monomer versions of the same systems. It was attempted in this project to extend our analysis to similar quantitative means, namely by using the Atomic Pseudopotentials Engine [77] for atomic calculations from which, in principle, a $s(\rho)$ injective relation could be derived, to be used in the analysis of atomic dimers. At the time, this effort has not yielded any usable results. However, the groundwork is set for the development of these quantitative tools in this Python implementation. Then, for a more quantitative analysis, interaction energy calculations based solely on the density maps, *e.g.* considering bonding and non-bonding contributions, are an achievable possibility. A similar attempt was made before by some of the development team of NCIPLOT [78] and was likely the genesis of the more recent quantitative analysis. This could in turn facilitate molecular dynamics simulations, as well as geometry

optimizations in systems for which weak non-covalent interactions are responsible for stabilization. Examples of this last class include processes of adsorption of simple molecules onto surfaces, such as graphene.

Although all NCI analyses done here were performed on ground state densities, in principle there is no reason why they cannot be performed on 'snapshot' densities of time-dependent systems, from *e.g.* TDDFT calculations or molecular dynamics. This would allow us, for instance, to study thermal effects on NCIs, or how NCIs are responsible for long range bonding in large biological systems which exhibit mechanical folding and unfolding, yielding different conformations. However, for 3D visual representations of these processes using the NCI index analysis, it is recommended that a process for representing isosurfaces is automated.

Appendix A

Finite Difference Derivatives

A.1 Introduction

To compute spatial derivatives of the charge density based on its grid values obtained from previous self-consistent calculations, finite-difference based approximations were used. The expressions used are shown here for first order (gradient) and second order (Hessian matrix) derivatives, and the first terms of the approximation errors are explicitly computed.

A.2 First order derivatives

Let $f(x)$ be a scalar field, and suppose the x coordinate of which it is a function is discretized in some interval according to

$$x_i = x_0 + i\,\delta x \tag{A.1}$$

with $\delta x \ll 1$. We then have a discretization of the values of f given by $f_i = f(x_i)$. Expanding $f(x)$ in a Taylor series around $x = x_i$,[1]

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_i)}{n!}(x - x_i)^n \tag{A.2}$$

Let us define $g_i = g(x_i)$ as

$$g_i = \frac{f_{i+1} - f_{i-1}}{2\,\delta x} \tag{A.3}$$

Expanding this definition using (A.2),

$$\begin{aligned} g_i &= \frac{1}{2\,\delta x}\left(f(x_i + \delta x) - f(x_i - \delta x)\right) \\ &= \frac{1}{2\,\delta x}\sum_{n=0}^{\infty} \frac{f^{(n)}(x_i)}{n!}\left[\delta x^n - (-\delta x)^n\right] \end{aligned} \tag{A.4}$$

[1] $f^{(n)}(x_i) \equiv \left.\frac{d^n f}{dx^n}\right|_{x=x_i}$

The even terms[2] of this expansion are null since $(-\delta x)^n = \delta x^n$. On the other hand, for odd terms $(-\delta x)^n = -\delta x^n$, thus we can write

$$g_i = \frac{1}{2\,\delta x} \sum_{n=0}^{\infty} \frac{f^{(2n+1)}(x_i)}{(2n+1)!} \cdot 2\,\delta x^{(2n+1)} = \sum_{n=0}^{\infty} \frac{f^{(2n+1)}(x_i)}{(2n+1)!} \cdot \delta x^{2n} \tag{A.5}$$

Explicitly writing the first terms of this expansion,

$$g_i = f'(x_i) + \frac{f^{(3)}(x_i)}{6}\,\delta x^2 + \mathcal{O}\left(\frac{\delta x^4}{120}\right) \tag{A.6}$$

Thus, equation (A.3) provides an accurate approximation for the first derivative in x of the scalar field f at grid point x_i, using only the value of f at the neighboring grid points. More accurate finite-difference derivatives can be constructed which make use of a greater number of neighboring points, but this form was found to be accurate enough for the purpose of this project, both because of relatively low spacings and spatial third order derivatives of the charge density near non-covalent interaction regions.

This one-dimensional first derivative can be generalized into partial derivatives of n independent coordinates. Let f be a scalar field on an n-dimensional space

$$f = f(\mathbf{x})\,, \quad \mathbf{x} \in \mathbb{R}^n \tag{A.7}$$

Discretizing this n-space similarly to (A.1),

$$x_\alpha^{(i)} = x_\alpha^{(0)} + i\,\delta x_\alpha \tag{A.8}$$

with $\alpha = 1, ..., n$ the component index of $\mathbf{x}$. The discretization of f follows and its values on the grid are indexed by n indices i, j, k, *etc.* Let i_α be the index pertaining to component α, *e.g.* $i_1 \equiv i$, $i_2 \equiv j$, $i_3 \equiv k$, *etc.* Then we have that the partial derivative of $f(\mathbf{x})$ with respect to component x_α at the point indexed by $(ijk...)$ can be approximated by

$$\left.\frac{\partial f}{\partial x_\alpha}\right|_{ijk...} \approx g^\alpha_{ijk...} = \frac{f_{i_1\,...,\,i_\alpha+1,\,...,\,i_n} - f_{i_1,\,...,\,i_\alpha-1,\,...,\,i_n}}{2\,\delta x_\alpha} \tag{A.9}$$

Taking $n = 3$ and f the charge density, each grid value of the three spatial first derivatives was computed thusly, in a vectorized manner.

A.3 Second order derivatives (Hessian matrix)

For simplification of notation, let us consider the $n = 3$ case from the start. The Hessian matrix of scalar field f, $H(f)$, is a 9 component[3] field indexed by 2 integers and given by

$$H_{\alpha\beta}\left[f(\mathbf{x})\right] = \frac{\partial^2 f}{\partial x_\alpha \partial x_\beta} \tag{A.10}$$

[2] $n = 2l$ with $l \in \mathbb{N}$

[3] Only 6 of which are independent due to its symmetry

In practice, since a second derivative can be computed as the derivative of a derivative of f, we can use the expression in (A.9) to compute these from the previously computed gradient components. There are, however, a couple of reasons as to why it would be more efficient to derive and use a second derivative expression rather than computing the gradient of each gradient component:

- Such an approach would compute arrays for 9 components, rather than the required 6 independent components, needlessly sacrificing memory in the analysis;

- For specific derivatives, this approach leads to an approximation with greater associated errors.

For these reasons we will focus on diagonal ($\alpha = \beta$) and non-diagonal Hessian components separately.

A.3.1 Diagonal components

Diagonal components of $H(f)$ are second derivatives in the same spatial coordinate, so we can deal with the one-dimensional case as in the previous section, and then generalize it to $n = 3$. Thus, consider f and g discretized in grids as shown in (A.1) and (A.3).

Since $f''(x_i)$ is simply the derivative of $f'(x_i)$, a possible way to approximate it would be

$$\tilde{h}_i = \frac{g_{i+1} - g_{i-1}}{2\,\delta x} \tag{A.11}$$

Replacing the values of g according to (A.3),

$$\begin{aligned} \tilde{h}_i &= \frac{1}{2\,\delta x}\left(\frac{f_{i+2} - f_i}{2\,\delta x} - \frac{f_i - f_{i-2}}{2\,\delta x}\right) \\ &= \frac{f_{i+2} - 2\,f_i + f_{i-2}}{(2\,\delta x)^2} \end{aligned} \tag{A.12}$$

As shall become apparent, this would correspond to the computation of $f''(x_i)$ in a grid with the spacing doubled (half the resolution), and thus has a greater error than if the first neighbors of f_i were used instead of $f_{i\pm2}$. A better expression can be derived in one of two ways which are equivalent:

- by considering the first derivatives in virtual midpoints $g_{i\pm1/2}$ – even though these grid points are not defined this is not a problem because the final expression is independent of them;

- by considering another approximation to the first derivatives – side point – which have greater associated errors than the central point expression, but which ultimately cancel out when computing the second derivative.

The reason these two are equivalent is that the different approximations mentioned for the first derivatives on the first neighbors are actually accurate central

point approximations for the virtual midpoints. We will, thus, formally consider the first alternative. Computing the midpoint first derivatives,

$$g_{i+1/2} = \frac{f_{i+1} - f_i}{\delta x} \qquad \wedge \qquad g_{i-1/2} = \frac{f_i - f_{i-1}}{\delta x} \tag{A.13}$$

Considering the central point derivative of these two points,

$$h_i = \frac{g_{i+1/2} - g_{i-1/2}}{\delta x} = \frac{f_{i+1} - 2\,f_i + f_{i-1}}{\delta x^2} \tag{A.14}$$

which confirms the statement after equation (A.12) about a grid with half the resolution.

For a deeper understanding of the associated errors, we can compute the Taylor series expansion of h_i based on the expansions of f around x_i in (A.2), similarly to the expansion for g_i in (A.6).

$$\begin{aligned} h_i &= \frac{1}{\delta x^2}\left[f(x_i + \delta x) - 2\,f(x_i) + f(x_i - \delta x)\right] \\ &= \frac{1}{\delta x^2}\left[f(x_i) - 2\,f(x_i) + f(x_i) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_i)}{n!}\,\left(\delta x^n + (-\delta x)^n\right)\right] \end{aligned} \tag{A.15}$$

It is trivial to see the even terms survive in this case, as opposed to the odd terms in the gradient computation. Thus,

$$\begin{aligned} h_i &= \frac{1}{\delta x^2} \sum_{n=1}^{\infty} \frac{f^{(2n)}(x_i)}{(2n)!} \cdot 2\,\delta x^{2n} \\ &= \sum_{n=1}^{\infty} \frac{f^{(2n)}(x_i)}{(2n)!/2}\,\delta x^{2(n-1)} \end{aligned} \tag{A.16}$$

Explicitly writing the first terms,

$$h_i = f''(x_i) + \frac{f^{(4)}(x_i)}{12}\,\delta x^2 + \mathcal{O}\left(\frac{\delta x^4}{360}\right) \tag{A.17}$$

Thus, equation (A.14) provides an accurate approximation for the second derivative in x of the scalar field f at grid point x_i, using only the value of f at the neighboring grid points. It is once again worth mentioning more accurate finite-difference derivatives can be constructed which make use of a greater number of neighboring points, but that this form was found to be accurate enough for the purpose of evaluating the sign of the median eigenvalue of the Hessian matrix at every grid point.

The generalization to 3-dimensional spatial derivatives is done trivially in a similar manner to the gradient in (A.9). Let f be a scalar field in 3-dimensional space, let i be the index pertaining to component α, and let j and k be the indices pertaining to the remaining components. At the spatial point indexed by ijk we have[4]

$$\left.\frac{\partial^2 f}{\partial x_\alpha^2}\right|_{ijk} \approx h_{ijk}^{(\alpha\alpha)} = \frac{f_{i+1}^{jk} - 2\,f_i^{jk} + f_{i-1}^{jk}}{\delta x_\alpha^2} \tag{A.18}$$

[4]For a clearer notation, the index related to the derivative component is subscripted, and the indices related to the remaining components are superscripted, in f.

A.3.2 Non-diagonal components

In this case, because the derivatives are done in different directions, *i.e.*, along dependencies on different components, we do not run the risk of originating an expression which depends on not-first neighbors of f_{ijk}. We can thus use the central-point derivative of the previously computed gradient components, although being careful only to compute the necessary independent ones to avoid needless memory requirements. It is, however, still useful to compute the overall central point expression for this case, to prove that it maintains the symmetry of the Hessian matrix, and to better understand the associated errors with such an approximation.

With this in mind, consider i and j the indices pertaining to the components α and β along which f is being derivated ($\alpha \neq \beta$), and k the remaining component. Following (A.3), we have

$$g^{(\alpha)}_{ijk} = \frac{f^k_{i+1,\,j} - f^k_{i-1,\,j}}{2\,\delta x_\alpha} \qquad \wedge \qquad g^{(\beta)}_{ijk} = \frac{f^k_{i,\,j+1} - f^k_{i,\,j-1}}{2\,\delta x_\beta} \tag{A.19}$$

Since

$$H_{\alpha\beta}(f) = \frac{\partial}{\partial x_\alpha}\left(\frac{\partial f}{\partial x_\beta}\right) \tag{A.20}$$

we have

$$\begin{aligned}
h^{(\alpha\beta)}_{ijk} &= \frac{g^{(\beta)}_{i+1,\,jk} - g^{(\beta)}_{i-1,\,jk}}{2\,\delta x_\alpha} \\
&= \frac{1}{2\,\delta x_\alpha}\left[\frac{f^k_{i+1,\,j+1} - f^k_{i+1,\,j-1}}{2\,\delta x_\beta} - \frac{f^k_{i-1,\,j+1} - f^k_{i-1,\,j-1}}{2\,\delta x_\beta}\right] \\
h^{(\alpha\beta)}_{ijk} &= \frac{f^k_{i+1,\,j+1} - f^k_{i+1,\,j-1} - f^k_{i-1,\,j+1} + f^k_{i-1,\,j-1}}{4\,\delta x_\alpha\,\delta x_\beta}
\end{aligned} \tag{A.21}$$

From (A.20), switching α and β,

$$\begin{aligned}
h^{(\beta\alpha)}_{ijk} &= \frac{g^{(\alpha)}_{i,\,j+1} - g^{(\alpha)}_{i,\,j-1}}{2\,\delta x_\beta} \\
&= \frac{1}{2\,\delta x_\beta}\left[\frac{f^k_{i+1,\,j+1} - f^k_{i-1,\,j+1}}{2\,\delta x_\alpha} - \frac{f^k_{i+1,\,j-1} - f^k_{i-1,\,j-1}}{2\,\delta x_\alpha}\right] \\
h^{(\beta\alpha)}_{ijk} &= \frac{f^k_{i+1,\,j+1} - f^k_{i-1,\,j+1} - f^k_{i+1,\,j-1} + f^k_{i-1,\,j-1}}{4\,\delta x_\beta\,\delta x_\alpha} = h^{(\alpha\beta)}_{ijk}
\end{aligned} \tag{A.22}$$

Thus this expression maintains the symmetry of the Hessian matrix, and the order of α and β are irrelevant.

Error analysis of approximation for non-diagonal components

Since there is more than one dimension involved in the derivative, it is not possible to use the conventional approach seen in (A.6) and (A.17) of taking the next terms in the Taylor series expansion of h. This is because, although it is possible to write a Taylor-series-like expansion of f for multiple coordinates, it does not have a

(trivial) closed form like for the one-dimensional case, and each group of terms has to be analyzed individually. Let n be the number of independent coordinates, *i.e.*, $\mathbf{x} \in \mathbb{R}^n$,

$$f(\mathbf{x}_0 + \delta\mathbf{x}) = f(\mathbf{x}_0) + \sum_{\alpha=1}^{n} \frac{\partial f}{\partial x_\alpha}\,\delta x_\alpha + \frac{1}{2}\sum_{\alpha,\beta=1}^{n} \frac{\partial^2 f}{\partial x_\alpha\,\partial x_\beta}\delta x_\alpha\,\delta x_\beta + \\ + \frac{1}{6}\sum_{\alpha,\beta,\gamma=1}^{n} \frac{\partial^3 f}{\partial x_\alpha\,\partial x_\beta\,\partial x_\gamma}\,\delta x_\alpha\,\delta x_\beta \delta x_\gamma \quad + \quad ... \tag{A.23}$$

We can, however, still find a closed form expansion for f in the neighborhood of some point $\mathbf{x}_0$. To simplify calculations we will first consider the lowest-dimensional case that can be considered – similarly to the one-dimensional case for first derivatives and diagonal-component second derivatives: that f is dependent only on the two independent coordinates along which it is being derivated (x_α and x_β). Designate these coordinates x and y, and take[5]

$$f^z(x,y) = f(\mathbf{x}^*)\,, \quad \mathbf{x}^* : \, x_\gamma = z,\, \gamma \neq \alpha, \beta \tag{A.24}$$

In other words, this procedure comes down to fixing the coordinate along which no derivatives are being taken, and considering f as a function of only x and y along that plane. For simpler notation, f and f^z are used interchangeably for as long as it is possible to differentiate between them given the context.

The assumption that f is composed of separable functions

$$f(x,y) = \Theta(x)\,\Phi(y) \tag{A.25}$$

is not general enough. However, if we consider a (possibly infinite) sum of separable functions

$$f(x,y) = \sum_{p=1}^{P} \Theta_p(x)\,\Phi_p(y) \tag{A.26}$$

this form is accurate enough that for any 'well-behaved' f we can consider it to be true – after all, this is technically what the expansion shown in (A.23) is.

From (A.26) we have the general derivative form

$$\frac{\partial^{k+l} f}{\partial x^k\,\partial y^l} = \sum_{p=1}^{P} \Theta_p^{(k)}(x)\,\Phi_p^{(l)}(y) \tag{A.27}$$

Since they're single-coordinate dependent, each of the Θ_p and Φ_p functions can be expanded in a Taylor series near x^0 and y^0 by (A.2). Thus, we can expand $f(x,y)$ from (A.26) as

$$f(x^0 + \delta x,\, y^0 + \delta y) = \sum_{p=1}^{P}\sum_{k=0}^{\infty}\sum_{l=0}^{\infty} \frac{\Theta_p^{(k)}(x^0)}{k!}\,\frac{\Phi_p^{(l)}(y^0)}{l!}\,\delta x^k\,\delta y^l \tag{A.28}$$

[5]For $n = 3$, z is the value that the third coordinate (along which no derivatives are done) takes. We can still consider this definition for general n, in which case z is a set of values that the remaining coordinates take.

The sums can be rearranged such that the right side of equation (A.27) is visible as the sum in index p, then replaced by the left side of the equation.

$$\begin{aligned} f(x^0+\delta x,\, y^0+\delta y) &= \sum_{k=0}^{\infty}\frac{\delta x^k}{k!}\sum_{l=0}^{\infty}\frac{\delta x^l}{l!}\sum_{p=1}^{P}\Theta_p^{(k)}(x^0)\,\Phi_p^{(l)}(y^0) \\ &= \sum_{k,l=0}^{\infty}\frac{\delta x^k\,\delta y^l}{k!\,l!}\left.\frac{\partial^{k+l}f}{\partial x^k\,\partial x^l}\right|_{x^0,\,y^0} \end{aligned} \tag{A.29}$$

We have thus found a closed form of (A.23) for $n = 2$.[6] The next step is to redefine the sums presented so that the terms are ordered in powers of δx or δy: terms of order m are proportional to products of the type $\delta x^k\,\delta y^l$ if $k + l = m$. We can achieve this by switching one of the indices k or l to $m = k + l$. Replacing l by $m - k$, we rewrite (A.29) as

$$f(x^0+\delta x,\, y^0+\delta y) = \sum_{m=0}^{\infty}\sum_{k=0}^{m}\frac{\delta x^k\,\delta y^{m-k}}{k!\,(m-k)!}\left.\frac{\partial^m f}{\partial x^k\,\partial x^{m-k}}\right|_{x^0,\,y^0} \tag{A.31}$$

This way the summed terms have a clear ordering, and it is visible that each term of order m is composed of a sum of $m + 1$ 'sub-terms'.

We are now ready to take (A.21) and expand the f terms in a Taylor-like series, keeping in mind that

$$f_{i\pm1,\,j\pm1} = f(x_\alpha^{(i)} \pm \delta x_\alpha,\, x_\beta^{(j)} \pm \delta x_\beta) \tag{A.32}$$

Thus, from (A.21) and (A.31),

$$h_{ijk}^{(\alpha\beta)} = \frac{1}{4\,\delta x_\alpha\,\delta x_\beta}\sum_{m=0}^{\infty}\sum_{k=0}^{m}\frac{1}{k!\,(m-k)!}\left.\frac{\partial^m f}{\partial x_\alpha^k\,\partial x_\beta^{m-k}}\right|_{ijk}\delta x_\alpha^k\,\delta x_\beta^{m-k}\,\sigma_{mk} \tag{A.33}$$

with

$$\sigma_{mk} \equiv 1 - (-1)^k - (-1)^{m-k} + (-1)^m \tag{A.34}$$

The value of σ_{mk} depends only on the parity of m and k. Looking at the four individual cases,

- with m odd and k odd, $m - k$ is even and we have $\sigma_{mk} = 1 + 1 - 1 - 1 = 0$;
- with m odd and k even, $m - k$ is odd and we have $\sigma_{mk} = 1 - 1 + 1 - 1 = 0$;
- with m even and k odd, $m - k$ is odd and we have $\sigma_{mk} = 1 + 1 + 1 + 1 = 4$;
- with m even and k even, $m - k$ is even and we have $\sigma_{mk} = 1 - 1 - 1 + 1 = 0$.

[6] This form appears to be generalizable to n dimensions as

$$f(\mathbf{x}_0+\delta\mathbf{x}) = \sum_{\{k_\alpha\}}^{\infty}\left[\prod_{\alpha=1}^{n}\frac{\delta x_\alpha^{k_\alpha}}{k_\alpha!}\right]\left.\frac{\partial^{\left(\sum_{\alpha=1}^{n}k_\alpha\right)}f}{\prod_{\alpha=1}^{n}\partial x_\alpha^{k_\alpha}}\right|_{\mathbf{x}_0} \tag{A.30}$$

where k_α is the index related to component α and each k_α is summed from 0 to infinity.

Thus, the only surviving terms are for even m and odd k, so we can replace σ_{mk} by 4 if we replace the summing indices according to $m = 2a$ and $k = 2b + 1$, obtaining

$$h_{ijk}^{(\alpha\beta)} = \sum_{a=1}^{\infty} \sum_{b=0}^{a-1} \frac{1}{(2b+1)!\,(2(a-b)-1)!} \left. \frac{\partial^{2a} f}{\partial x_\alpha^{2b+1}\, \partial x_\beta^{2(a-b)-1}} \right|_{ijk} \delta x_\alpha^{2b}\, \delta x_\beta^{2(a-b-1)} \quad \text{(A.35)}$$

The notation can be simplified if we designate $m_a = 2a$, $k_b = 2b + 1$ and $l_{ab} = m_a - k_b = 2(a - b) - 1$ (regaining these indices and their intuitive meaning from (A.29)), and writing

$$h_{ijk}^{(\alpha\beta)} = \sum_{a=1}^{\infty} \sum_{b=0}^{a-1} \frac{1}{k_b!\, l_{ab}!} \left. \frac{\partial^{m_a} f}{\partial x_\alpha^{k_b}\, \partial x_\beta^{l_{ab}}} \right|_{ijk} \delta x_\alpha^{k_b - 1}\, \delta x_\beta^{l_{ab} - 1} \quad \text{(A.36)}$$

Finally, explicitly writing the first terms of (A.35),

$$h_{ijk}^{(\alpha\beta)} = \left. \frac{\partial^2 f}{\partial x_\alpha\, \partial x_\beta} \right|_{ijk} + \frac{1}{3!} \left[\left. \frac{\partial^4 f}{\partial x_\alpha\, \partial x_\beta^3} \right|_{ijk} \delta x_\beta^2 + \left. \frac{\partial^4 f}{\partial x_\alpha^3\, \partial x_\beta} \right|_{ijk} \delta x_\alpha^2 \right] + \mathcal{O}\left(\frac{\delta x^4}{36} \right) \quad \text{(A.37)}$$

Thus, equation (A.21) provides an accurate approximation for the second derivative in x_α and x_β of the scalar field f at the grid point indexed by (ijk), using only the value of f at the neighboring grid points. Once more, it is worth mentioning more accurate finite-difference derivatives can be constructed which make use of a greater number of neighboring points, but this form was found to be accurate enough for the purpose of evaluating the sign of the median eigenvalue of the Hessian matrix at every grid point.

A.4 Final note on border conditions

All of the expressions discussed in this appendix conveniently ignore the fact that border points of the grid do not have all nearest neighbors defined. This problem is addressed by temporarily expanding the grid depending on the border conditions for each specific border, prior to derivative computations. If the border is periodic, the grid is expanded one plane across said border by copying the values of the parallel border.

If, on the other hand, there is no information about the border conditions other than the fact that it is not periodic, *i.e.*, `pbc = False`, the grid is expanded such that the first derivative is approximated by a side-point expression as opposed to central point. Let us take, as a particular case, the border defined by $i = 0$. According to the central-point expression in (A.3), the derivative along component $\alpha = 1$ would be computed as

$$g_{0jk}^{(1)} = \frac{f_1^{jk} - f_{-1}^{jk}}{2\,\delta x} \quad \text{(A.38)}$$

Since we have no information on f_{-1}^{jk} we wish to compute the derivative as

$$g_{0jk}^{(1)} = \frac{f_1^{jk} - f_0^{jk}}{\delta x} \quad \text{(A.39)}$$

which is a central-point approximation for the derivative at a virtual midpoint indexed by $(1/2, j, k)$. We wish to compute all derivatives in the grid in a vectorized manner, so it becomes important to use the same expression for all points, instead of a different expression for borders. We can, however, from equations (A.38) and (A.39), extrapolate what would have to be the value of f_{-1}^{jk} for both expressions to yield the same result.

$$\frac{f_1^{jk} - f_{-1}^{jk}}{2\,\delta x} = \frac{f_1^{jk} - f_0^{jk}}{\delta x} \quad \Rightarrow \quad f_{-1}^{jk} = 2f_0^{jk} - f_1^{jk} \tag{A.40}$$

The grid is thus expanded so that f_{-1}^{jk} has this value, and then the same central-point expression is used for all points. This method is used for all non-periodic borders.

Appendix B

Discussion on RDG calculation errors

B.1 Introduction

During the NCI index analysis it is necessary to compute for each grid point the reduced density gradient s, which is proportional to the norm of the density gradient and inversely proportional to a positive power of the density ρ. This is a possible problem *e.g.* in the following two situations:

- the imported cell includes simulation box borders in which the density is 0;
- occasional floating point errors in the prior computation of the density – or in its transcription to a cube file – may result in sporadic negative density values, which need to be zeroed before the NCI index analysis is carried out.

In both situations and in others like these, it becomes necessary to deal with possible infinite values of s at certain grid points, since $s \xrightarrow{\rho \to 0} \infty$. A simple solution would be to carry out the evaluation of s over a grid point cycle, computing s only for points with non-zero density. This results in problems of its own, namely,

- this grid cycle evaluation is much slower than the vectorized direct computation of s at every grid point;
- `Nan` values of s in the grid later result in complications during other evaluations done in the code, which would need to be corrected using a similar grid point cycle, thus propagating the slowness of the method into further parts of the analysis.

The simplest solution found to solve this issue without creating more problems was to shift the density at every grid point by a very small positive amount $\delta\rho$. Note that this is done only after computing the density gradient, so it does not affect the s result in any way other than through its dependency on the density. In practice it creates an artificial bound for large values of s, never reaching infinity but being large enough for all practical purposes.

The error introduced in s by using this method can be quantified and controlled.

B.2 Introduced error

Let

$$\rho' = \rho + \delta\rho \tag{B.1}$$

with $\delta\rho$ a small positive value. For constant $|\nabla\rho|$[1], the relative error made in computing s with ρ' instead of ρ is given by

$$\varepsilon = \left| \frac{s(\rho') - s(\rho)}{s(\rho)} \right| = 1 - \frac{s(\rho')}{s(\rho)} \tag{B.2}$$

since $s(\rho) \geq s(\rho')$ if $\rho' \geq \rho$. Substituting the expression for $s(\rho)$ from equation (2.14), we eventually arrive at

$$\frac{\delta\rho}{\rho} = (1-\varepsilon)^{-3/4} - 1 \tag{B.3}$$

If $\varepsilon \ll 1$ we can expand

$$(1-\varepsilon)^{-3/4} = 1 + \frac{3}{4}\varepsilon + \mathcal{O}\left(\varepsilon^2\right) \tag{B.4}$$

and finally obtain (ignoring second order terms in ε),

$$\frac{\delta\rho}{\rho} \simeq \frac{3}{4}\varepsilon \tag{B.5}$$

Expanding the expression in logarithmic form, assuming atomic units for the density,

$$\log\varepsilon \simeq \log\delta\rho - \log\frac{3}{4} - \log\rho \tag{B.6}$$

To reiterate, this relates the relative error ε in computing s when the density ρ is shifted slightly by a positive amount $\delta\rho$, but $|\nabla\rho|$ is kept constant. Suppose $\delta\rho = 10^{-16}$ a.u.. If we choose a certain density value ρ_0 and compute the error ε_0 associated with it we have

- for $\rho < \rho_0$, $\varepsilon > \varepsilon_0$;
- for $\rho > \rho_0$, $\varepsilon < \varepsilon_0$.

Most weak interactions analyzed in the current document have a minimum associated density greater than $\sim 10^{-6}$ a.u.. Choosing $\rho_0 = 10^{-6}$ we have

$$\log\varepsilon_0 \simeq -16 - \log\frac{3}{4} + 6 \sim -10$$

Thus for all points with densities greater than $\rho_0 = 10^{-6}$ a.u. the relative error in s is lower than $\varepsilon_0 \sim 10^{-10}$. Figure B.1 contains a graphical representation of the relation between the relative error and the density value (in atomic units) for several values of $\delta\rho$, from both equations (B.3) and (B.5).[2]

Obviously closer to $\rho = 0$ the relative error tends to 1, since (for constant $|\nabla\rho|$), $s \xrightarrow{\rho\to 0} \infty$.

[1] *i.e.*, considering s as a function of (only) ρ.

[2] -10 was used here as a base value instead of the aforementioned -16 to circumvent floating point errors that would result in divergences in the high ρ limit of the graph.

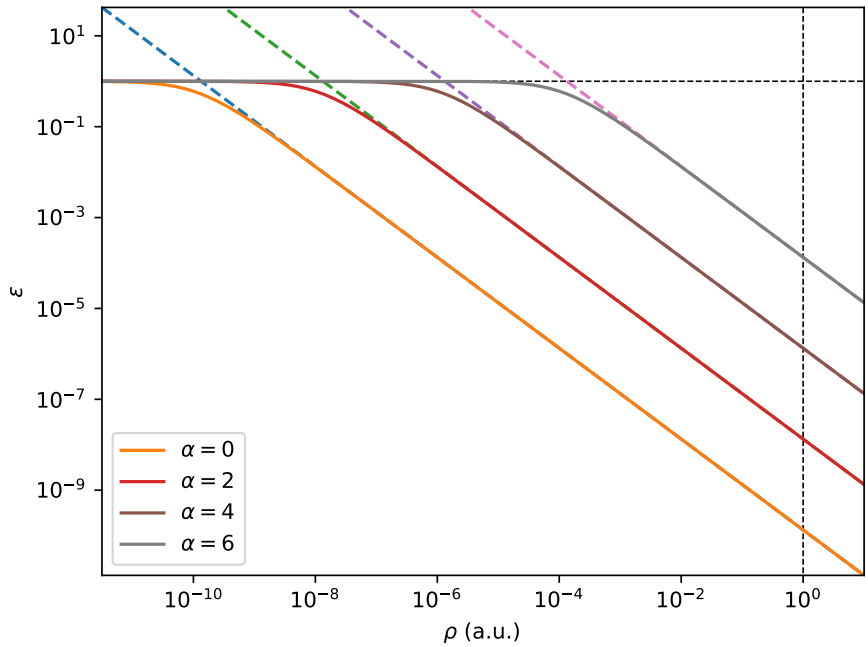


Figure B.1: Relative error ε as a function of density ρ for several values of $\delta\rho = 10^{-10+\alpha}$ a.u., in logarithmic scale. Dashed lines represent the linear relation given by (B.5), and solid lines represent the real relation as seen in (B.3).

B.3 Possible partial correction of s for finite ρ

B.3.1 Derivatives

For any function $f(x)$ which has the form $f(x) = Ax^{\alpha}$, we have

$$f^{(n)}(x) = \left[\prod_{k=0}^{n-1}(\alpha - k)\right] \frac{f(x)}{x^n} \tag{B.7}$$

Proof

The result is trivial for $n = 1$. For $n + 1$ given n,

$$\begin{aligned} f^{(n+1)}(x) &= \frac{d}{dx} f^{(n)}(x) \\ &= \left[\prod_{k=0}^{n-1}(\alpha - k)\right] \left(\frac{f'(x)}{x^n} - n\frac{f(x)}{x^{n+1}}\right) \\ &= \left[\prod_{k=0}^{n-1}(\alpha - k)\right] (\alpha - n)\frac{f(x)}{x^{n+1}} \\ f^{(n+1)}(x) &= \left[\prod_{k=0}^{n}(\alpha - k)\right] \frac{f(x)}{x^{n+1}} \end{aligned} \tag{B.8}$$

Thus a proof by induction is complete.

B.3.2 Taylor series expansion

Considering s a function of ρ,

$$s^{(n)}(\rho) = \frac{s(\rho)}{\rho^n} \prod_{k=0}^{n-1} (\alpha - k) \ , \qquad \alpha = -\frac{4}{3} \tag{B.9}$$

Expanding $s(\rho)$ in a Taylor series around $\rho' = \rho + \delta\rho$,

$$s(\rho' - \delta\rho) = \sum_{n=0}^{\infty} \frac{s^{(n)}(\rho')}{n!} (-\delta\rho)^n \tag{B.10}$$

Keeping in mind that if $\alpha < 0$,

$$\alpha - k = -\left(|\alpha| + k\right) \quad \Rightarrow \quad \prod_{k=0}^{n-1} (\alpha - k) = (-1)^n \prod_{k=0}^{n-1} (|\alpha| + k) \tag{B.11}$$

then from equations (B.9) and (B.10) we have

$$\begin{aligned} s(\rho) &= \sum_{n=0}^{\infty} \frac{1}{n!} \frac{s(\rho')}{\rho'^n} \, (-1)^n \, \delta\rho^n \, (-1)^n \prod_{k=0}^{n-1} \left(\frac{4}{3} + k\right) \\ &= s(\rho') \sum_{n=0}^{\infty} \frac{1}{n!} \left(\frac{\delta\rho}{\rho'}\right)^n \prod_{k=0}^{n-1} \left(\frac{4}{3} + k\right) \end{aligned} \tag{B.12}$$

For simple visualization of correction factors, we can rewrite (B.12) as

$$s(\rho) = s(\rho') \left\{ 1 + \sum_{n=1}^{\infty} \gamma_n \, \eta^n \right\} \tag{B.13}$$

with

$$\eta = \frac{\delta\rho}{\rho'} = \left(1 + \frac{\rho}{\delta\rho}\right)^{-1} \leq 1 \tag{B.14}$$

and

$$\gamma_n = \frac{1}{n!} \prod_{k=0}^{n-1} \left(\frac{4}{3} + k\right) \tag{B.15}$$

We can set $\gamma_0 = 1$ and obtain γ_{n+1} from γ_n as

$$\frac{\gamma_{n+1}}{\gamma_n} = \frac{n!}{(n+1)!} \frac{\prod_{k=0}^{n} \left(\frac{4}{3} + k\right)}{\prod_{k=0}^{n-1} \left(\frac{4}{3} + k\right)} = \frac{n + 4/3}{n+1} \tag{B.16}$$

Convergence

To analyze the convergence of the series presented in equation (B.13), let $a_n = \gamma_n \, \eta^n$,

$$\begin{aligned} \frac{a_{n+1}}{a_n} &= \frac{\gamma_{n+1}}{\gamma_n} \frac{\eta^{n+1}}{\eta^n} \\ &= \frac{n + 4/3}{n+1} \, \eta \quad \xrightarrow{n \to \infty} \quad \eta^+ \end{aligned} \tag{B.17}$$

Thus, the series converges for $\eta < 1$, *i.e.*, for finite ρ; and diverges for $\eta = 1$, which was expected from the divergence of s in this case.

B.3.3 Correction algorithm

In practice, a simple algorithm for the partial correction of s for finite ρ would go as follows:

- choose correction order n;
- compute $\eta = \frac{\delta\rho}{\rho'}$ and define $\gamma, s_f = 1$;
- for $k = 0$ to $n - 1$ do

$$\begin{aligned} \gamma &\leftarrow \gamma \cdot \frac{k + 4/3}{k + 1} \\ s_f &\leftarrow s_f + \gamma \cdot \eta^{k+1} \end{aligned}$$

- correct s with

$$s \leftarrow s \cdot s_f$$

B.3.4 Final remarks

It is important to note that the partial correction described here has no effect on the visualization aspect of the analysis – be it in the 2D or 3D plots – but may be necessary for more accurate quantified results which make use of the scatterplot points. Specifically, following the analysis in section B.2, we can see that although the charge density in the NCI regions is very low – typically around 10^{-6} to 10^{-2} a.u. – it is still several orders of magnitude above the density shift used, in the order of 10^{-16}, and as a consequence the introduced error is minimal in the NCI regions.

Alternative shifts and correction

The $\delta\rho$ shift described here is constant in space, being the same for all points in the grid. However, since we need only shift the density of points with zero density, it is possible to make the shift dependent on density – though it is recommended that such a dependence is relatively continuous and smooth – *e.g.* with

$$\delta\rho = A \exp\left\{ -\left(\frac{\rho}{D}\right)^2 \right\} \tag{B.18}$$

with A and D shift parameters.[3] Since the derivation of the correction algorithm done in this appendix is for every grid point independently, there is no change in the algorithm when dealing with variable shifts, as long as the appropriate η factor is computed and used. In practice, the smooth dependence of $\delta\rho$ in ρ guarantees the smoothness of the correction factor as well.

[3] A defines the order of magnitude of the shift, and D defines the order of magnitude below which the densities are shifted. In the most recent versions of the code, this was the correction scheme chosen, and the value 10^{-16} was used for both shift parameters A and D.

Appendix C

Evaluating the sign of the median eigenvalue of a symmetric 3×3 matrix

C.1 Introduction

One of the steps of a real-space-based NCI index analysis is to compute, for every grid point, the sign of the median eigenvalue (λ_2) of the Hessian matrix of the electronic density. The 6 independent components of the Hessian matrix can be computed for each point via finite-difference derivatives. A possible conventional approach would then be to explicitly build the 3×3 matrix for each grid point and diagonalize it in order to obtain the three eigenvalues, from which λ_2 (and its sign) can be extracted.[1] However, this evaluation becomes too computationally exhausting for a relatively dense grid, especially considering that the method involves computing values which are irrelevant to the analysis in question.

We thus present a more efficient way to obtain the sign of λ_2 without explicitly computing its value (or any of the remaining eigenvalues), using some of the properties of the characteristic polynomial of 3×3 symmetric matrices, as is the case for the Hessian matrix.

C.2 Characteristic polynomial

C.2.1 Polynomial coefficients

Let A be a 3×3 symmetric matrix

$$A = \begin{pmatrix} a_1 & b & c \\ b & a_2 & d \\ c & d & a_3 \end{pmatrix} \tag{C.1}$$

[1] It is worth noting that by conventional the author does not mean standard/common. In fact, this is not the approach taken by Johnson *et al* in [3], relying instead on derivatives *via* Fourier transforms, obtaining coefficients which directly relate to the eigenvalues and eigenvectors discussed here. [41]

with eigenvalues $\lambda_1 \leq \lambda_2 \leq \lambda_3$. We can define the characteristic polynomial of the matrix as

$$P(\lambda) \equiv \det\left(A - \lambda \mathbb{1}\right) \tag{C.2}$$

Computing it explicitly in terms of the 6 independent parameters detailed in (C.1), and grouping the resulting terms in powers of λ,

$$\begin{aligned} P(\lambda) &= -\lambda^3 + \lambda^2 (a_1 + a_2 + a_3) - \lambda \left(a_1 a_2 + a_1 a_3 + a_2 a_3 - b^2 - c^2 - d^2\right) \\ &\quad + a_1 a_2 a_3 + 2bcd - \left(a_1 d^2 + a_2 c^2 + a_3 b^2\right) \end{aligned} \tag{C.3}$$

Two terms are immediately recognizable in this form – the trace of matrix A

$$\tau = \operatorname{tr} A = a_1 + a_2 + a_3 \tag{C.4}$$

and its determinant

$$\Delta = \det A = a_1 a_2 a_3 + 2bcd - \left(a_1 d^2 + a_2 c^2 + a_3 b^2\right) \tag{C.5}$$

In the same fashion, let us designate the term multiplied by $-\lambda$ in equation (C.3) by β:

$$\beta \equiv a_1 a_2 + a_1 a_3 + a_2 a_3 - b^2 - c^2 - d^2 \tag{C.6}$$

We can now rewrite (C.3) as

$$P(\lambda) = -\lambda^3 + \tau \lambda^2 - \beta \lambda + \Delta \tag{C.7}$$

C.2.2 Relation to eigenvalues

The eigenvalues of A are the roots of $P(\lambda)$:

$$P(\lambda_i) = 0 \quad i = 1, 2, 3 \tag{C.8}$$

We can thus write

$$P(\lambda) = a (\lambda - \lambda_1)(\lambda - \lambda_2)(\lambda - \lambda_3) \tag{C.9}$$

Expanding in powers of λ,

$$P(\lambda) = a \left[\lambda^3 - (\lambda_1 + \lambda_2 + \lambda_3) \lambda^2 + (\lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_2 \lambda_3) \lambda - \lambda_1 \lambda_2 \lambda_3\right] \tag{C.10}$$

Comparing equations (C.7) and (C.10) we arrive at

$$\begin{aligned} a &= -1 \\ \tau &= \lambda_1 + \lambda_2 + \lambda_3 \\ \beta &= \lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_2 \lambda_3 \\ \Delta &= \lambda_1 \lambda_2 \lambda_3 \end{aligned} \tag{C.11}$$

which was expected for τ and Δ, since the trace and determinant of a symmetric matrix are invariant under rotation transformations, including the rotation that diagonalizes it. This result suggests that the quantity β is also invariant under such unitary transformations, since its form is the same as in (C.6).

C.2.3 Monotony of $P(\lambda)$ and stationary points

Taking the derivative of equation (C.7),

$$P'(\lambda) = -3\lambda^2 + 2\tau\lambda - \beta \tag{C.12}$$

Solving for the roots of $P'(\lambda)$, we expect to obtain the stationary points of the $P(\lambda)$ curve. Since it is a quadratic function, the number of (real) solutions can be 0, 1, or 2. As shall become apparent in the following subsections, due to the symmetry of matrix A and consequence that all eigenvalues are real, the number of stationary points will be 2 in the most general case, and 1 in the specific case of total eigenvalue degeneracy.

Solving for the roots of (C.12),

$$P'(x) = 0 \Rightarrow x = \frac{\tau}{3} \pm \sqrt{\left(\frac{\tau}{3}\right)^2 - \frac{\beta}{3}} \tag{C.13}$$

Note that both solutions presented are directly computable from the τ, β and Δ coefficients, which are in turn directly computable from the 6 independent parameters that define matrix A.

No degeneracy

Take the case for which $\lambda_1 < \lambda_2 < \lambda_3$, *i.e.* all eigenvalues are different. Since the dominant term in (C.7) for large $|\lambda|$ is $-\lambda^3$, we know that

- $P(\lambda) > 0$, for $\lambda < \lambda_1$;
- $P(\lambda) < 0$, for $\lambda > \lambda_3$.

Taking into account that there is an additional (different) root of $P(\lambda)$ between λ_1 and λ_3, we can infer that

- $P(\lambda) < 0$, for $\lambda_1 < \lambda < \lambda_2$;
- $P(\lambda) > 0$, for $\lambda_2 < \lambda < \lambda_3$.

which in turn means we must find two stationary points in the $]\lambda_1, \lambda_3[$ interval: the first one is a minimum $P(x_1) < 0$ with $x_1 \in]\lambda_1, \lambda_2[$ the negative sign solution to equation (C.13); and the second one is a maximum $P(x_2) > 0$ with $x_2 \in]\lambda_2, \lambda_3[$ the positive sign solution to equation (C.13). Thus, it is possible to compute both solutions to equation (C.13) after which we know $\lambda_2 \in]x_1, x_2[$ and is the only root in this interval.

A graphical example of this case is presented in Figure C.1. We can extend the ordering of eigenvalues and stationary points to the general case by changing the inequality signs to include possible equalities:

$$\lambda_1 \leq x_1 \leq \lambda_2 \leq x_2 \leq \lambda_3 \tag{C.14}$$

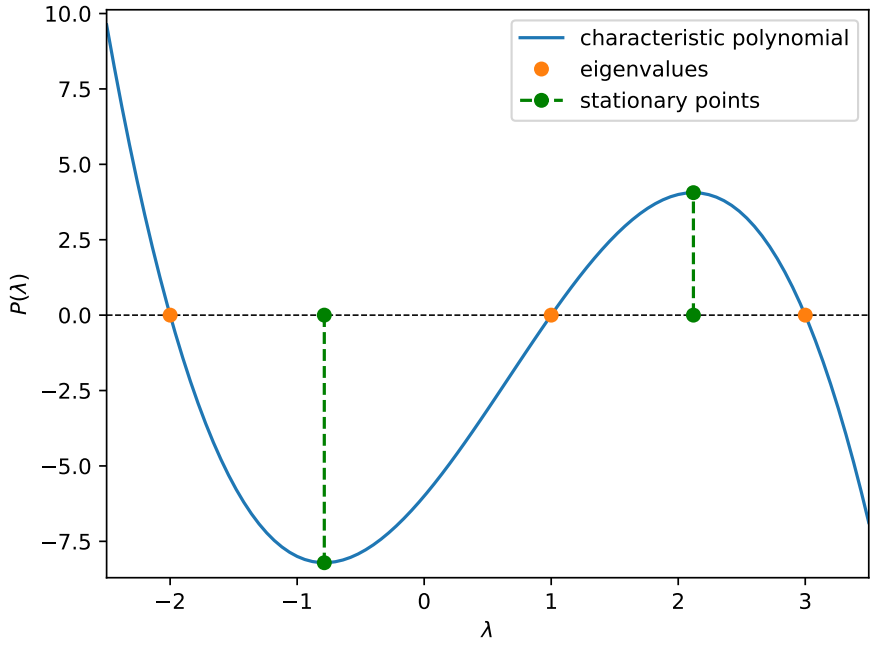


Figure C.1: Characteristic polynomial of a symmetric matrix with eigenvalues -2, 1, and 3, which result in the parameters $\tau = 2$, $\beta = -5$, and $\Delta = -6$.

Partial degeneracy

Take the case for which two of the eigenvalues are equal to ω and the remaining one equal to $\kappa \neq \omega$. From (C.11),

$$\tau = 2\omega + \kappa \qquad \wedge \qquad \beta = \omega(\omega + 2\kappa) \tag{C.15}$$

Substituting these into equation (C.13),

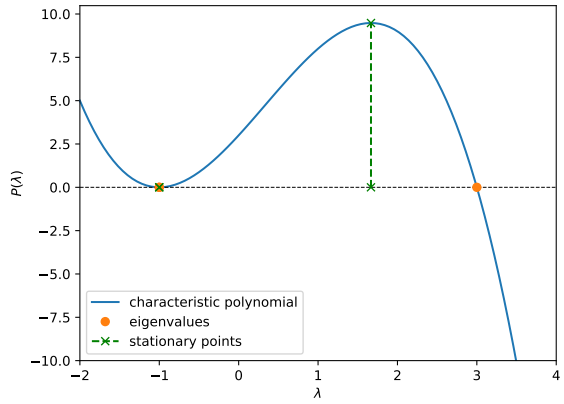
$$x = \frac{2\omega + \kappa \pm |\omega - \kappa|}{3} \tag{C.16}$$

There are still two possible sets of solutions:

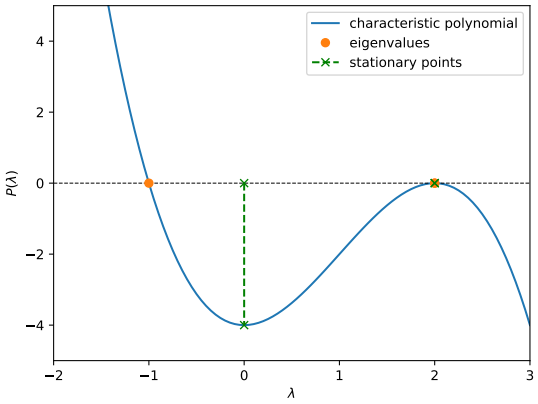
- If $\lambda_1 = \lambda_2 = \omega$ and $\lambda_3 = \kappa$, then $|\omega - \kappa| = \kappa - \omega$ and the negative sign solution is $x_1 = \omega = \lambda_1 = \lambda_2$;
- Otherwise $\lambda_1 = \kappa$ and $\lambda_2 = \lambda_3 = \omega$, in which case $|\omega - \kappa| = \omega - \kappa$ and the positive sign solution is $x_2 = \omega = \lambda_2 = \lambda_3$.

In conclusion, if two eigenvalues are equal, they are also equal to the stationary value[2] that would otherwise be between them. This means said stationary value is a root of $P(\lambda)$, which is easily verifiable, and that λ_2 is equal to any stationary value that is also a root of $P(\lambda)$. Two graphical examples of this case are presented in Figure C.2 – one for each ω, κ distribution of the eigenvalues.

[2]We here define *stationary value* as x such that the point $(x, P(x))$ is a stationary point of the $P(\lambda)$ curve, *i.e.*, $P'(x) = 0$.



(a) Eigenvalues -1 (double) and 3, which result in $\tau = 1$, $\beta = -5$, and $\Delta = 3$.



(b) Eigenvalues -1 and 2 (double), which result in $\tau = 3$, $\beta = 0$, and $\Delta = 4$.

Figure C.2: Characteristic polynomials of symmetric matrices with the detailed degenerate eigenvalues and resulting parameters.

Total degeneracy

Take the case for which all eigenvalues are equal to ω. From (C.11),

$$\tau = 3\omega \qquad \wedge \qquad \beta = 3\omega^2 \tag{C.17}$$

This means

$$\left(\frac{\tau}{3}\right)^2 - \frac{\beta}{3} = \omega^2 - \omega^2 = 0 \tag{C.18}$$

From (C.13), one can see that the stationary values are degenerate in this case, and equal to the eigenvalues. Thus, it is possible to verify after computing τ and β that if $\tau^2/3 - \beta = 0$, all eigenvalues are degenerate and simply equal to $\tau/3$.[3] It is worth noting this case is included here out of academic interest, and for the sake of completeness of the method, since it has no practical application to the analysis at hand as of yet[4] – a totally degenerate Hessian matrix would correspond to a spherically symmetric density from the point of view of the spatial point in which the matrix is computed, which is only possible at the position of the nucleus of a single atom system, in which case there are no interactions (noncovalent or otherwise) to analyze.

We have now verified that equation (C.14) always holds.

C.3 Evaluating the sign of λ_2

A priori we have no knowledge of the eigenvalue degeneracy of matrix A. We can still, however, based on the analysis of the previous section, carefully examine the obtained coefficients – from equations (C.4) to (C.6) – and evaluate the sign of the median eigenvalue. The first step is to compute both solutions to equation (C.13). We know $\lambda_2 \in [x_1, x_2]$, thus,

[3] The curve thus has the form $P(\lambda) = -(\lambda - \tau/3)^3$.

[4] When considering only atomic or molecular systems.

- if $x_1 > 0$, then $\text{sgn}(\lambda_2) = 1$;
- else if $x_2 < 0$, then $\text{sgn}(\lambda_2) = -1$.

Otherwise $x_1 \leq 0 \leq x_2$, in which case we need to compute the value of $P(x_1)$ and $P(x_2)$, using equation (C.7). If any of the stationary values are roots of $P(\lambda)$, then λ_2 is equal to said value, thus

- if $P(x_1) = 0$, then $\text{sgn}(\lambda_2) = \text{sgn}(x_1)$;
- else if $P(x_2) = 0$, then $\text{sgn}(\lambda_2) = \text{sgn}(x_2)$.

The sign evaluations are done this way in order to include the possibility of x_1 or x_2 equal to zero (in which case $\text{sgn}(\lambda_2) = 0$).

If $\text{sgn}(\lambda_2)$ has not been found so far, we arrive at the non-degenerate case with $x_1 \leq 0 \leq x_2$ and $x_1 < \lambda_2 < x_2$. Recall that in this case $P(x_1) < 0$ is a minimum, $P(x_2) > 0$ is a maximum, and that λ_2 is the only root in this interval. Thus,

- if $P(0) > 0$, then the $P(\lambda)$ curve crosses zero before $\lambda = 0$, thus $\lambda_2 < 0$;
- if $P(0) < 0$, then $P(\lambda)$ only crosses zero after $\lambda = 0$, thus $\lambda_2 > 0$;
- if $P(0) = 0$, then $\lambda = 0$ is the root, and $\lambda_2 = 0$.

From (C.7) we have $P(0) = \Delta$. Thus, the conditions stated here can be summarized as $\text{sgn}(\lambda_2) = -\text{sgn}(\Delta)$.

The total degeneracy case was ignored here for the practical reasons detailed in the previous section. If, however, it becomes useful to include this case in the presented analysis, the only recommended addition to the start of the algorithm is an evaluation of whether $\tau^2/3 - \beta = 0$, in which case $\lambda_2 = \tau/3$, thus $\text{sgn}(\lambda_2) = \text{sgn}(\tau)$. Nevertheless, note that such an addition is not required, since this case is covered in the evaluation of $P(x_1)$.

The algorithm detailed here is summarized in flowchart form in Figure C.3.

C.4 Conclusion and remarks

We are thus able to evaluate the sign of the median eigenvalue of a symmetric 3×3 matrix in every case, without explicitly building the matrix – *i.e.*, by only having access to its 6 independent components – or computing any of the eigenvalues directly. This greatly reduces the computational cost of evaluating $\text{sgn}(\lambda_2)$ at every point in the grid. It is also worth noting that, should the explicit computation of λ_2 become necessary in future versions of the NCI index analysis, this algorithm provides a simple way to do so without computing any of the other eigenvalues: even in the case of total non-degeneracy, λ_2 is easily computable *via* a closed interval root search between x_1 and x_2, such as *regula falsi*.

Relating this algorithm explicitly to the real-space-based NCI index, note that before this step in the analysis, we have computed 6 arrays of the same shape as

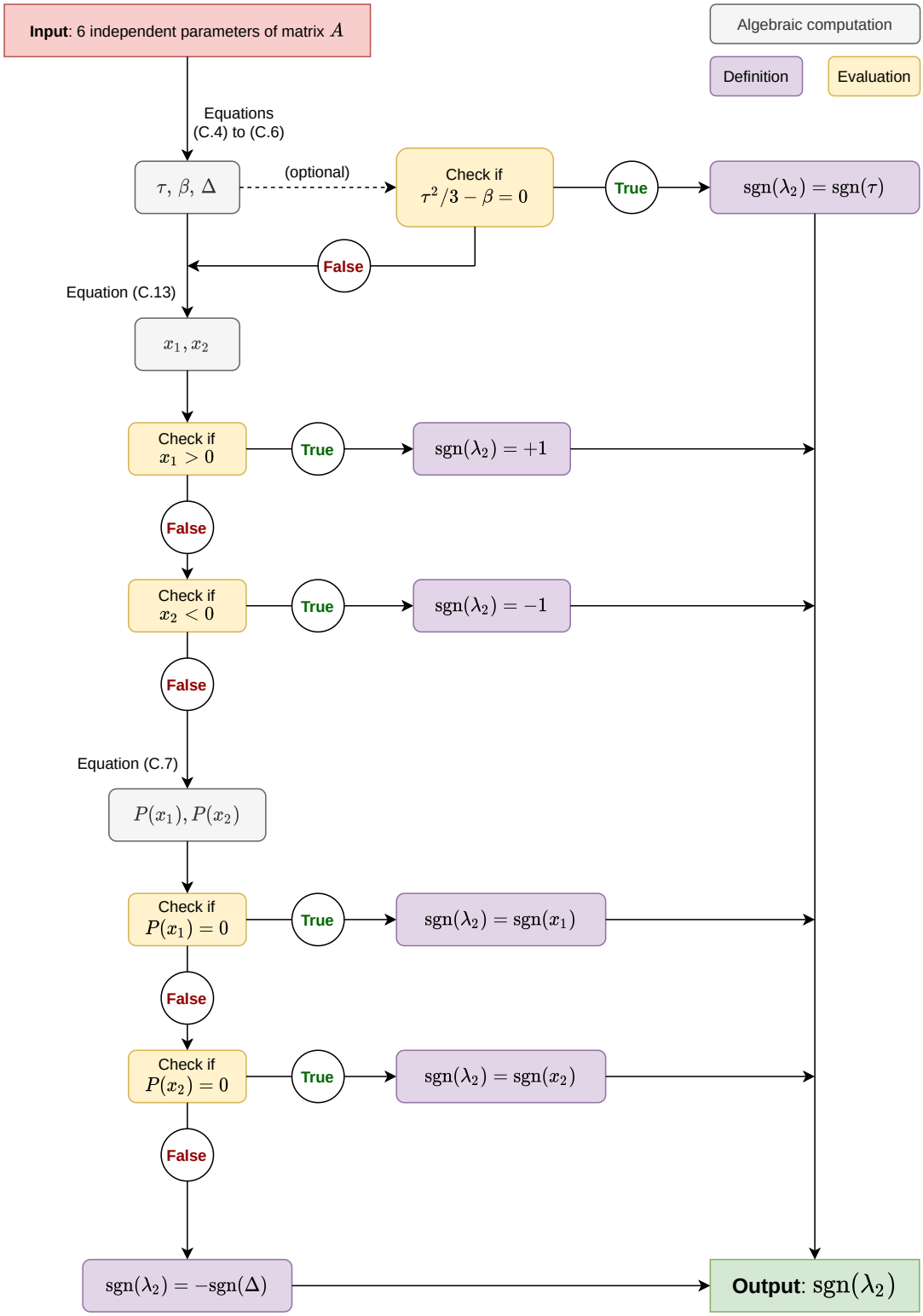


Figure C.3: Flowchart schematic of the algorithm detailed in the present appendix to evaluate the sign of the median eigenvalue of a 3×3 symmetric matrix.

the input density – related to the 6 independent components of the Hessian at every grid point – and can easily compute the τ, β and Δ coefficients at every point by the definitions presented in equations (C.4) to (C.6), or directly in terms of Hessian components:

$$\tau = H_{11} + H_{22} + H_{33} \tag{C.19}$$

$$\beta = H_{11}H_{22} + H_{11}H_{33} + H_{22}H_{33} - \left(H_{12}^2 + H_{13}^2 + H_{23}^2\right) \tag{C.20}$$

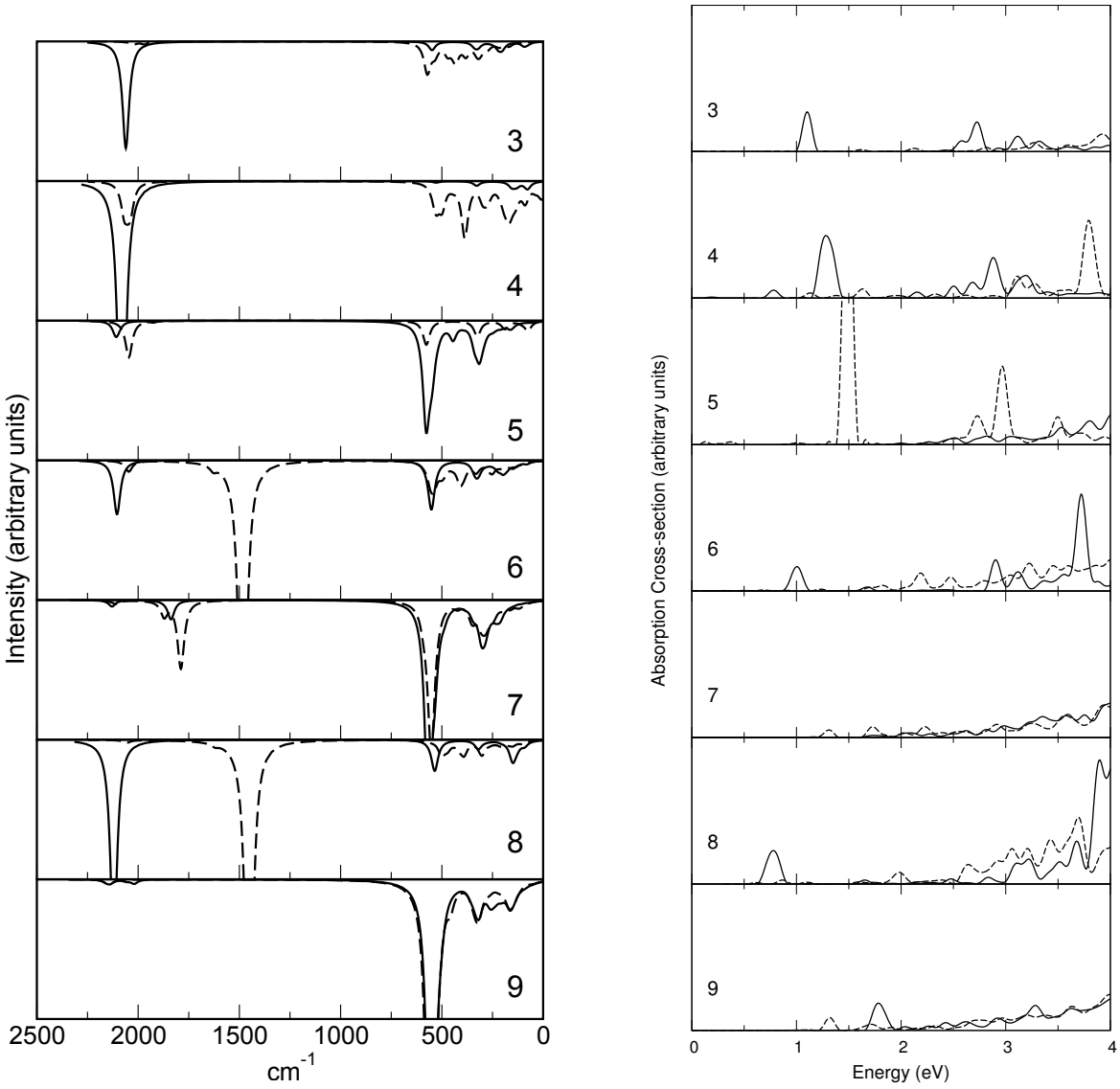
$$\Delta = H_{11}H_{22}H_{33} + 2\,H_{12}H_{13}H_{23} - \left(H_{11}H_{23}^2 + H_{22}H_{13}^2 + H_{33}H_{12}^2\right) \tag{C.21}$$

These calculations are simple algebraic computations and so can be easily done for every grid point in a vectorized manner, *e.g.* using `numpy` arrays. This means the point-by-point evaluation for each grid point only needs to be done after most of the calculations, making the process even more efficient. In the current implementation of this algorithm, we make further use of said vectorization by also computing the arrays for x_1, x_2, $P(x_1)$ and $P(x_2)$ before starting the grid point cycle – even though some of the components of these arrays may not be necessary, we found that doing these vectorized pre-calculations is more efficient than computing only the necessary values during the grid point cycle. Complementing this line of thought, we also pre-compute four auxiliary Boolean-datatype arrays: two for checking the x_1 and x_2 relations with 0 for every grid point (if $x_1 > 0$ and if $x_2 < 0$), and two for checking if $P(x_i) = 0$, $i = 1, 2$ – thus during the grid point cycle we need only check the value of the grid point component of these arrays instead of explicitly doing the evaluation, when required.

Appendix D

Bismuth carbide clusters theoretical spectroscopic results

Presented here are the infra-red and optical absorption spectra of the Bismuth carbide clusters discussed in section 4.4 obtained by Bruce Milne and Micael Oliveira.



(a) IR spectra

(b) Optical absorption spectra

Figure D.1: Theoretical spectroscopic results for Bismuth carbide clusters. Solid and dashed spectra correspond to 'a' and 'b' forms respectively

Bibliography

[1] Erin R. Johnson, Shahar Keinan, Paula Mori-Sánchez, Julia Contreras-García, Aron J. Cohen, and Weitao Yang. Revealing noncovalent interactions. *Journal of the American Chemical Society*, 132(18):6498–6506, may 2010.

[2] Christophe Narth, Zeina Maroun, Roberto A. Boto, Robin Chaudret, Marie Laure Bonnet, Jean Philip Piquemal, and Julia Contreras-García. A Complete NCI Perspective: From New Bonds to Reactivity. In *Challenges and Advances in Computational Chemistry and Physics*, volume 22, pages 491–527. 2016.

[3] Julia Contreras-García, Erin R Johnson, Shahar Keinan, Robin Chaudret, Jean-Philip Piquemal, David N Beratan, and Weitao Yang. NCIPLOT: A Program for Plotting Noncovalent Interaction Regions. *J. Chem. Theory Comput*, 7:625–632, 2011.

[4] Roberto A Boto, Francesca Peccati, Chaoyu Quan, Alessandra Carbone, Jean-Philip Piquemal, Yvon Maday, and Julia Contreras-García. NCIPLOT4: Fast, Robust, and Quantitative Analysis of Noncovalent Interactions. *J. Chem. Theory Comput*, 16:0, 2020.

[5] A. D. Becke and K. E. Edgecombe. A simple measure of electron localization in atomic and molecular systems. *The Journal of Chemical Physics*, 92(9):5397–5403, 1990.

[6] Julia Contreras-García, Roberto A. Boto, Fernando Izquierdo-Ruiz, Igor Reva, Tatiana Woller, and Mercedes Alonso. A benchmark for the non-covalent interaction (NCI) index or... is it really all in the geometry? *Theoretical Chemistry Accounts*, 135(10):1–14, 2016.

[7] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Physical Review B*, 136, 1964.

[8] A. Otero de-la Roza, M.A. Blanco, A. Martín Pendás, and Víctor Luaña. Critic: a new program for the topological analysis of solid-state electron densities. *Computer Physics Communications*, 180(1):157 – 166, 2009.

[9] A. Otero de-la Roza, Erin R. Johnson, and Víctor Luaña. Critic2: A program for real-space analysis of quantum chemical interactions in solids. *Computer Physics Communications*, 185(3):1007 – 1018, 2014.

[10] Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, Eric D Hermes, Paul C Jennings, Peter Bjerre Jensen, James

Kermode, John R Kitchin, Esben Leonhard Kolsbjerg, Joseph Kubal, Kristen Kaasbjerg, Steen Lysgaard, Jón Bergmann Maronsson, Tristan Maxson, Thomas Olsen, Lars Pastewka, Andrew Peterson, Carsten Rostgaard, Jakob Schiøtz, Ole Schütt, Mikkel Strange, Kristian S Thygesen, Tejs Vegge, Lasse Vilhelmsen, Michael Walter, Zhenhua Zeng, and Karsten W Jacobsen. The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, 2017.

[11] Richard Bader. *Atoms in molecules : a quantum theory*. Clarendon Press, Oxford, 1990.

[12] D. R. Hartree. The Wave Mechanics of an Atom with a Non-Coulomb Central Field Part I Theory and Methods. *Mathematical Proceedings of the Cambridge Philosophical Society*, 24(1):89–110, jan 1928.

[13] The 2018 codata recommended values of the fundamental physical constants.

[14] Carlos Fiolhais, Fernando Nogueira, and Miguel A. L. Marques, editors. *A Primer in Density Functional Theory*. Springer Berlin Heidelberg, 2003.

[15] W. Kohn and L.J. Sham. Self-Consistent Equations Including Exchange and Correlation Effects. *Physical Review A*, 140, 1965.

[16] V.E. Van Doren, C. Van Alsenoy, P. Geerlings, and American Institute of Physics. *Density Functional Theory and Its Application to Materials: Antwerp, Belgium, 8-10 June 2000*. AIP conference proceedings. American Institute of Physics, 2001.

[17] Diogo A. F. Almeida. Pynci, September 2020. Link to GitLab repository for more recent versions of the code: https://gitlab.com/diofalmeida/pynci.

[18] Eric F. Pettersen, Thomas D. Goddard, Conrad C. Huang, Gregory S. Couch, Daniel M. Greenblatt, Elaine C. Meng, and Thomas E. Ferrin. Ucsf chimera—a visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 25(13):1605–1612, 2004.

[19] William Humphrey, Andrew Dalke, and Klaus Schulten. VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics*, 14:33–38, 1996.

[20] Thomas Williams, Colin Kelley, and many others. Gnuplot 5.2: an interactive plotting program, Dec. 2019.

[21] Miguel A.L. Marques, Micael J.T. Oliveira, and Tobias Burnus. Libxc: A library of exchange and correlation functionals for density functional theory. *Computer Physics Communications*, 183(10):2272 – 2281, 2012.

[22] Susi Lehtola, Conrad Steigemann, Micael J.T. Oliveira, and Miguel A.L. Marques. Recent developments in libxc — a comprehensive library of functionals for density functional theory. *SoftwareX*, 7:1 – 5, 2018.

[23] P. A. M. Dirac. Note on exchange phenomena in the thomas atom. *Mathematical Proceedings of the Cambridge Philosophical Society*, 26(3):376–385, 1930.

[24] F. Bloch. Bemerkung zur elektronentheorie des ferromagnetismus und der elektrischen leitfähigkeit. *Zeitschrift für Physik*, 57(7-8):545–555, July 1929.

[25] J. P. Perdew and Alex Zunger. Self-interaction correction to density-functional approximations for many-electron systems. *Phys. Rev. B*, 23:5048–5079, May 1981.

[26] J. J. Mortensen, L. B. Hansen, and K. W. Jacobsen. Real-space grid implementation of the projector augmented wave method. *Phys. Rev. B*, 71:035109, Jan 2005.

[27] J Enkovaara, C Rostgaard, J J Mortensen, J Chen, M Dułak, L Ferrighi, J Gavnholt, C Glinsvad, V Haikola, H A Hansen, H H Kristoffersen, M Kuisma, A H Larsen, L Lehtovaara, M Ljungberg, O Lopez-Acevedo, P G Moses, J Ojanen, T Olsen, V Petzold, N A Romero, J Stausholm-Møller, M Strange, G A Tritsaris, M Vanin, M Walter, B Hammer, H Häkkinen, G K H Madsen, R M Nieminen, J K Nørskov, M Puska, T T Rantala, J Schiøtz, K S Thygesen, and K W Jacobsen. Electronic structure calculations with GPAW: a real-space implementation of the projector augmented-wave method. *Journal of Physics: Condensed Matter*, 22(25):253202, jun 2010.

[28] A. H. Larsen, M. Vanin, J. J. Mortensen, K. S. Thygesen, and K. W. Jacobsen. Localized atomic basis set in the projector augmented wave method. *Phys. Rev. B*, 80:195112, Nov 2009.

[29] Roman M. Balabin. Polar (acyclic) isomer of formic acid dimer: Gas-phase raman spectroscopy study and thermodynamic parameters. *The Journal of Physical Chemistry A*, 113(17):4910–4918, 2009. PMID: 19344174.

[30] Nicolas Tancogne-Dejean, Micael J. T. Oliveira, Xavier Andrade, Heiko Appel, Carlos H. Borca, Guillaume Le Breton, Florian Buchholz, Alberto Castro, Stefano Corni, Alfredo A. Correa, Umberto De Giovannini, Alain Delgado, Florian G. Eich, Johannes Flick, Gabriel Gil, Adrián Gomez, Nicole Helbig, Hannes Hübener, René Jestädt, Joaquim Jornet-Somoza, Ask H. Larsen, Irina V. Lebedeva, Martin Lüders, Miguel A. L. Marques, Sebastian T. Ohlmann, Silvio Pipolo, Markus Rampp, Carlo A. Rozzi, David A. Strubbe, Shunsuke A. Sato, Christian Schäfer, Iris Theophilou, Alicia Welden, and Angel Rubio. Octopus, a computational framework for exploring light-driven phenomena and quantum dynamics in extended and finite systems. *The Journal of Chemical Physics*, 152(12):124119, 2020.

[31] Antonio Bauzá, Ibon Alkorta, José Elguero, Tiddo J. Mooibroek, and Antonio Frontera. Spodium Bonds: Noncovalent Interactions Involving Group 12 Elements. *Angewandte Chemie - International Edition*, pages 1–7, 2020.

[32] Joseph L. Jules and John R. Lombardi. Transition metal dimer internuclear distances from measured force constants. *Journal of Physical Chemistry A*, 107(9):1268–1273, 2003.

[33] Frank Neese. The orca program system. *WIREs Computational Molecular Science*, 2(1):73–78, 2012.

[34] Takeshi Noro, Masahiro Sekiya, and Toshikatsu Koga. Segmented contracted basis sets for atoms h through xe: Sapporo-(dk)-nzp sets (n = d, t, q). *Theoretical Chemistry Accounts*, 131, 2012.

[35] Benjamin P. Pritchard, Doaa Altarawy, Brett Didier, Tara D. Gibsom, and Theresa L. Windus. A new basis set exchange: An open, up-to-date resource for the molecular sciences community. *J. Chem. Inf. Model.*, 59, 2019.

[36] David Feller. The role of databases in support of computational chemistry calculations. *J. Comput. Chem.*, 17, 1996.

[37] Karen L. Schuchardt, Brett T. Didier, Todd Elsethagen, Lisong Sun, Vidhya Gurumoorthi, Jared Chase, Jun Li, and Theresa L. Windus. Basis set exchange: A community database for computational sciences. *J. Chem. Inf. Model.*, 47, 2007.

[38] Takahito Nakajima and Kimihiko Hirao. The douglas–kroll–hess approach. *Chemical Reviews*, 112(1):385–402, 2012. PMID: 21678899.

[39] Frank Neese. Efficient and accurate approximations to the molecular spin-orbit coupling operator and their use in molecular g-tensor calculations. *J. Chem. Phys.*, 122(3):034107, jan 2005.

[40] Jianwei Sun, Adrienn Ruzsinszky, and John P. Perdew. Strongly constrained and appropriately normed semilocal density functional. *Phys. Rev. Lett.*, 115:036402, Jul 2015.

[41] Alberto Otero-De-La-Roza, Erin R. Johnson, and Julia Contreras-García. Revealing non-covalent interactions in solids: NCI plots revisited. *Physical Chemistry Chemical Physics*, 14(35):12165–12172, 2012.

[42] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865–3868, Oct 1996.

[43] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple [phys. rev. lett. 77, 3865 (1996)]. *Phys. Rev. Lett.*, 78:1396–1396, Feb 1997.

[44] E. Briggs, D. Sullivan, and J. Bernholc. Real-space multigrid-based approach to large-scale electronic structure calculations. *Physical Review B - Condensed Matter and Materials Physics*, 54(20):14362–14375, 1996.

[45] Daniel R Cooper, B. D'Anjou, Nageswara R Ghattamaneni, Benjamin Harack, M. Hilke, Alexandre Horth, Norberto Majlis, M. Massicotte, L. Vandsburger, E. Whiteway, and V. Yu. Experimental review of graphene. *International Scholarly Research Notices*, 2012:1–56, 2011.

[46] D. D. L. Chung. *Journal of Materials Science*, 37(8):1475–1489, 2002.

[47] Shigeo Ted Oyama and S T Gyama. Preparation and Catalytic Properties of Transition Metal Carbides and Nitriles Hydrodeoxygenation of Bio-oil Model Compounds View project Selective oxidation using molecular oxygen as an oxidant View project Preparation and Catalytic Properties of Transit. *Catal. Today*, 15:179–200, 1992.

[48] S. T. Oyama. Introduction to the chemistry of transition metal carbides and nitrides. In *The Chemistry of Transition Metal Carbides and Nitrides*, pages 1–27. Springer Netherlands, Dordrecht, 1996.

[49] Jonathan Bernstein, Eran Armon, Erez Zemel, and Eli Kolodney. Formation of indium carbide cluster ions: Experimental and computational study. *J. Phys. Chem. A*, 117(46):11856–11865, nov 2013.

[50] Jin-Yun Yuan, Hong-Guang Xu, and Wei-Jun Zheng. Photoelectron spectroscopy and density functional study of $co_nc_2^-$ (n = 1-5) clusters. *Phys. Chem. Chem. Phys.*, 2014.

[51] L Delannoy, J.-M Giraudon, P Granger, L Leclercq, and G Leclercq. Group {VI} transition metal carbides as alternatives in the hydrodechlorination of chlorofluorocarbons. *Catal. Today*, 59(3–4):231 – 240, 2000.

[52] Raman Ganesan and Jae Sung Lee. Tungsten carbide microspheres as a noble-metal-economic electrocatalyst for methanol oxidation. *Angew. Chem. Int. Ed.*, 44(40):6557–6560, 2005.

[53] Ying Liu, Thomas G. Kelly, Jingguang G. Chen, and William E. Mustain. Metal carbides as alternative electrocatalyst supports. *ACS Catalysis*, 3(6):1184–1194, 2013.

[54] J. Bernstein, A. Landau, E. Zemel, and E. Kolodney. Tin-carbon clusters and the onset of microscopic level immiscibility: Experimental and computational study. *J. Chem. Phys.*, 143(11):114307, sep 2015.

[55] Takahiro Tamura, Takuya Takami, Sachio Kobayashi, Taro Nagahama, Takashi Yanase, and Toshihiro Shimada. Formation of bismuth-core-carbon-shell nanoparticles by bismuth immersion during plasma CVD synthesis of thin diamond films. *Diam. Relat. Mater.*, 69:127–132, oct 2016.

[56] Y. Yamada and T. Nakagawa. Bismuth carbide cluster ions produced by a gas aggregation source. *Int. J. Mass. Spec.*, 282(3):123 – 127, 2009.

[57] Axel D. Becke. Density-functional thermochemistry. iii. the role of exact exchange. *J. Chem. Phys.*, 98(7):5648–5652, 1993.

[58] P. J. Stephens, F. J. Devlin, C. F. Chabalowski, and M. J. Frisch. Ab initio calculation of vibrational absorption and circular dichroism spectra using density functional force fields. *J. Phys. Chem.*, 98(45):11623–11627, November 1994.

[59] M. W. Schmidt, K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. J. Su, T. L. Windus, M. Dupuis, and J. A. Montgomery. General atomic and molecular Electronic-Structure system. *J. Comput. Chem.*, 14(11):1347–1363, 1993.

[60] Florian Weigend and Reinhart Ahlrichs. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.*, 7:3297–3305, 2005.

[61] Bernhard Metz, Hermann Stoll, and Michael Dolg. Small-core multiconfiguration-dirac–hartree–fock-adjusted pseudopotentials for post-[bold d] main group elements: Application to pbh and pbo. *J. Chem. Phys.*, 113(7):2563–2569, 2000.

[62] Abdul-Rahman Allouche. Gabedit—a graphical user interface for computational chemistry softwares. *J. Comput. Chem.*, 32(1):174–182, 2011.

[63] E. Runge and E. K. U. Gross. Density-functional theory for time-dependent systems. *Phys. Rev. Lett.*, 52:997–1000, 1984.

[64] Miguel A. L. Marques, Neepa T. Maitra, Fernando M. S. Nogueira, Eberhard K. U. Gross, and Angel Rubio, editors. *Fundamentals of Time-Dependent Density Functional Theory*, volume 837 of *Lecture Notes in Physics*. Springer Verlag, Berlin, 2012.

[65] Alberto Castro, Heiko Appel, Micael Oliveira, Carlo A. Rozzi, Xavier Andrade, Florian Lorenzen, Miguel A. L. Marques, E. K. U. Gross, and Angel Rubio. Octopus: a tool for the application of time-dependent density functional theory. *Phys. Stat. Sol. B*, 243:2465–2488, 2006.

[66] Xavier Andrade, Joseba Alberdi-Rodriguez, David A Strubbe, Micael J T Oliveira, Fernando Nogueira, Alberto Castro, Javier Muguerza, Agustin Arruabarrena, Steven G Louie, Alán Aspuru-Guzik, Angel Rubio, and Miguel A L Marques. Time-dependent density-functional theory in massively parallel computer architectures: the octopus project. *J. Phys.: Condens. Matter*, 24:233202, 2012.

[67] Xavier Andrade, David Strubbe, Umberto De Giovannini, Ask Hjorth Larsen, Micael J. T. Oliveira, Joseba Alberdi-Rodriguez, Alejandro Varas, Iris Theophilou, Nicole Helbig, Matthieu J. Verstraete, Lorenzo Stella, Fernando Nogueira, Alan Aspuru-Guzik, Alberto Castro, Miguel A. L. Marques, and Angel Rubio. Real-space grids and the octopus code as tools for the development of new simulation approaches for electronic systems. *Phys. Chem. Chem. Phys.*, DOI: 10.1039/C5CP00351B:, 2015.

[68] Marvin Douglas and Norman M. Kroll. Quantum electrodynamical corrections to the fine structure of helium. *Ann. Phys.*, 82(1):89–155, 1974.

[69] Bernd A. Hess. Applicability of the no-pair equation with free-particle projection operators to atomic and molecular structure calculations. *Phys. Rev. A*, 32(2):756–763, 1985.

[70] Bernd A. Hess. Relativistic electronic-structure calculations employing a two-component no-pair formalism with external-field projection operators. *Phys. Rev. A*, 33(6):3742–3748, 1986.

[71] Georg Jansen and Bernd A. Hess. Revision of the Douglas-Kroll transformation. *Phys. Rev. A*, 39(11):6016–6017, 1989.

[72] Alexander Wolf, Markus Reiher, and Bernd Artur Hess. The generalized Douglas-Kroll transformation. *J. Chem. Phys.*, 117(20):9215–9226, nov 2002.

[73] Michael Bühl, Christoph Reimann, Dimitrios A. Pantazis, Thomas Bredow, and Frank Neese. Geometries of third-row transition-metal complexes from density-functional theory. *J. Chem. Theor. Comput.*, 4(9):1449–1459, sep 2008.

[74] Dimitrios A. Pantazis and Frank Neese. All-electron scalar relativistic basis sets for the 6p elements. *Theor. Chem. Acc.*, 131(11):1–7, oct 2012.

[75] Frank Neese. An improvement of the resolution of the identity approximation for the formation of the Coulomb matrix. *J. Comput. Chem.*, 24(14):1740–1747, nov 2003.

[76] Florian Weigend. Accurate Coulomb-fitting basis sets for H to Rn. *Phys. Chem. Chem. Phys.*, 8(9):1057–1065, feb 2006.

[77] Micael J.T. Oliveira and Fernando Nogueira. Generating relativistic pseudo-potentials with explicit incorporation of semi-core states using ape, the atomic pseudo-potentials engine. *Computer Physics Communications*, 178(7):524 – 534, 2008.

[78] Julia Contreras-García, Weitao Yang, and Erin R. Johnson. Analysis of hydrogen-bond interaction potentials from the electron density: Integration of noncovalent interaction regions. *Journal of Physical Chemistry A*, 115(45):12983–12990, 2011.