Faculty of Science
CHARLES UNIVERSITY IN PRAGUE

Master's thesis

Nonlocal correlation in density functional theory

Jan Hermann

Study program: Chemistry
Adviser: RNDr. Ota Bludský, CSc.

Prague, 2013

I hereby declare that this thesis is a result of my own work and effort and that all my intellectual debts are acknowledged with due reference to literature or otherwise. Neither this thesis nor its significant part has been previously submitted for any degree.

Prohlašuji, že jsem závěrečnou práci zpracoval samostatně a že jsem uvedl všechny použité informační zdroje a literaturu. Tato práce ani její podstatná část nebyla předložena k získání jiného nebo stejného akademického titulu.

In Prague, May 16, 2013

Jan Hermann

To Benjamin

Abstract

The van der Waals (vdW) interactions, or dispersion forces, are crucial in many chemical, physical and biological processes and received much attention from developers of density functional theory (DFT) methods. The most popular non-empirical DFT method for treating vdW interactions is the vdW density functional by Dion *et al.* (vdW-DF). Despite its success, vdW-DF is not accurate enough for many chemical applications. Here, we investigate two possible ways how to improve its accuracy. First, we reoptimize the only weakly specified parameter of vdW-DF for several semi-local functionals. On the S22 benchmark database set, we find that revPBE is the best performer, decreasing the error from 8.8% to 6.3%. Second, a system-specific but very accurate (~ 0.1 kcal/mol) DFT correction scheme is proposed for precise calculations of adsorbent–adsorbate interactions by combining vdW-DF and the empirical DFT/CC correction scheme. The new approach is applied to small molecules (CH_4, CO_2, H_2, H_2O, N_2) interacting with a quartz surface and a lamella of UTL zeolite. The very high accuracy of the new scheme and its relatively easy use and numerical stability compared to the earlier DFT/CC scheme offer a straightforward solution for obtaining reliable predictions of adsorption energies.

~

Van der Waalsovské (vdW) interakce, též disperzní síly, jsou klíčové v mnoha chemických, fyzikálních a biologických procesech a přitahují pozornost mnoha vývojářů metod založených na teorii funkcionálu hustoty (DFT). Nejčastěji používaná neempirická DFT metoda pro popis vdW interakcí je vdW funkcionál hustoty Diona a kol. (vdW-DF). Navzdory jeho úspěchu, vdW-DF neposkytuje dostatečnou přesnost v mnoha chemických aplikacích. V této práci zkoumáme dva možné způsoby jak zlepšit přesnost vdW-DF. Za prvé, optimalizujeme jediný částečně volný parametr vdW-DF pro několik semi-lokálních funkcionálů. Na testovací S22 databázi je revPBE nejlepším kandidátem, s jehož použitím se chyba snižuje z 8.8% na 6.3%. Za druhé, představujeme systémově specifické ale velmi přesné (~ 0.1 kcal/mol) DFT korekční schéma, které lze použít k precizním výpočtům interakcí mezi adsorbentem a adsorbátem. Schéma kombinuje vdW-DF a empirické korekční schéma DFT/CC. Tento nový přístup testujeme na malých molekulách (CH_4, CO_2, H_2, H_2O, N_2) interagující s povrchem křemenu a s lamelou zeolitu UTL. Vysoká přesnost našeho schématu a relativní snadnost jeho použití ve srovnání se starším DFT/CC schématem nabízejí přímočaré řešení pro získání spolehlivých předpovědí adsorpčních energií.

Contents

Preface

This thesis is a result of my two-year encounter with the van der Waals density functional (vdW-DF) developed in 2004. Instead of using available computer codes, I started by implementing vdW-DF in MATLAB in a hope that I will be able to modify and improve it. By the time I completed the code and performed first tests, I realized that the knowledge required for making any serious changes to vdW-DF is scattered over too many highly advanced research papers spanning more than 20 years. At the same time, it is a relatively new topic and there is a lack of not only textbooks but even review articles. So instead, I focused on applying vdW-DF in new areas and on tweaking its only parameter. While the latter proved useful in understanding the behaviour of vdW-DF, it provided no clear way how to improve its accuracy. As this was the primary motivation, I have searched for alternative ways to achieve this goal. In the end, using vdW-DF indirectly in a combination with an older correction scheme for density functional theory proved to be the way and the resulting approach improves the accuracy several times.

As for the structure and language of this thesis, I drew inspiration from Nordic countries, where theses are perceived and considered as short books. The first introductory chapter of this thesis is therefore written in quite broad terms, intended for anyone having at least some background in chemistry and physics. The subsequent chapters are already more traditional. The second and third chapter review the work done by others, while the fourth and fifth present my own work. The thesis was typeset in LaTeX, using a layout inspired by the `classicthesis` package by André Miede.

Finally, I would like to thank my supervisor Ota Bludský for guidance and thoughts. His comments pointed my efforts to the right directions and they were invaluable in development of the vdW-DF/CC correction scheme. Also, his role as an editor was mostly appreciated when my writing got out of hand.

Jan Hermann

Chapter 1

Introduction

Geckos are the ultimate climbers (Fig. 1.1). They can climb vertical surfaces, be they dry or wet, smooth or rough. Put them in a room and they will easily stick to the ceiling. The intricate structure of their footpads has been known for a long time, but it was not clear what kind of interaction could exert such a universal sticking force. It was not before 2002 that Autumn *et al.* rejected all other hypotheses and gave evidence that geckos are in fact utilizing the van der Waals forces [1].

What are these forces then, that do not discriminate between materials, but are attracting in every case? And surely there have to be other macroscopic examples of their effect. The composition of gecko footpads evolved to maximize the contact surface area. The billions of tiny hairs on them perfectly conform to

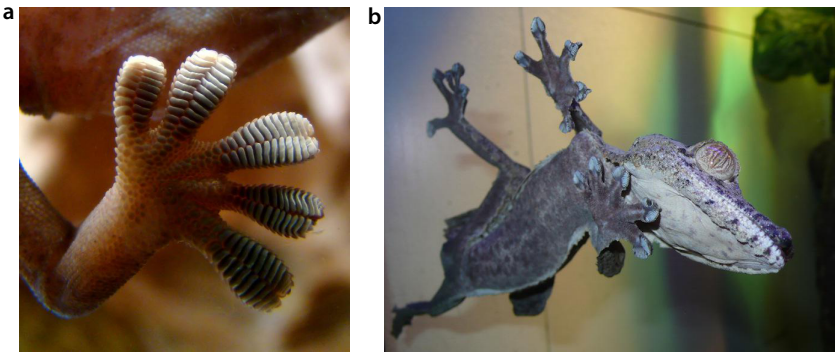

Figure 1.1 | Gecko's climbing ability. Each square millimeter of a gecko footpad (**a**) is equipped with about 14,000 tiny hairs (setae), and each seta contains from 100 to 1,000 spatulae. Each spatula can exert the force of 5 to 25 nN by means of van der Waals forces. This means that at 100% efficiency, footpads of an average gecko could carry more than 100 kg, easily supporting its weight of 70 g while climbing vertical glass (**b**). (Pictures from Wikipedia: GFDL licence, public domain.)

any surface under the gecko. The tips of the hairs contain β-keratin, a protein rich in stacked β-sheets. The sheets are in very close contact with the surface and it is the closeness and amount of surface contact which is the key to van der Waals forces.

Interactions can be classified by many categories. From the mathematical point of view, the important properties are whether they are attractive or repulsive or both, whether they are directional or not, and what is their dependence on distance.[1] In this perspective, we will compare van der Waals interactions, electrostatic interactions of charged and neutral bodies and gravity.

Van der Waals forces and gravity are always attractive. The consequence is that the strength of the interaction is determined by the sheer amount of the interacting objects. Another consequence is that the total effect of these forces is quite simple—they aggregate objects to form bigger objects, which then exert a force roughly equal to the sum of the forces of the aggregated objects. On the other hand, charged bodies can attract or repel each other, based on the sign of their charge. Quite in contrast, this leads either to separation of the bodies, or to aggregation with the consequence of neutralizing the force completely. Or, as in the example of a crystal, to both ends, but on different scales.

All forces from the previous paragraph are non-directional, in contrast to the interactions of neutral bodies. This has the consequence that van der Waals or gravitational forces are additive. In the case of charged bodies, the interactions are also additive in a sense, but as positive and negative interactions of roughly the same magnitude are often added, they tend to cancel out. Compare this to dipole or quadrupole interactions, where the force can oscillate between attractive or repulsive, based on the mutual orientation of the interacting bodies. When such bodies aggregate, these forces tend to average out. There are certain cases where all the aggregated interactions are aligned into one direction and enforced, but this does not happen in a typical chemical system.

All the compared forces decay with power-law dependence on the distance, r^{-n}, but they differ in the exponent of the power. Gravity and interaction of charged bodies decrease as r^{-1}. A mathematical consequence of this power-law is that the shapes of the interacting bodies do not matter much. Indeed, as far as gravity is concerned, two bodies can be well approximated by their centres of mass, and only their weight, a three-dimensional property, is relevant. The same thing holds

[1] The magnitude of the prefactor is crucial as well, but we are not interested in absolute strengths here, only in general properties.

for attraction or repulsion of charged bodies. Thus, ions can be quite accurately treated as point charges in many situations. The decay of interaction between neutral objects depends on the exact charge distributions, being r^{-3} for dipoles or r^{-5} for quadrupoles. The van der Waals potential vanishes very quickly, with the sixth power of distance, r^{-6}. This requirement of close contact leads to the conclusion that only geometry or more specifically the surface of the bodies, a two-dimensional property, is important. This has far-reaching consequences, because while the mass of an object cannot be hidden, a great amount of surface can be concealed in small space by twisting and bending.

The r^{-6} dependence of van der Waals forces is the reason why there is only a few macroscopic evidences of them, geckos being a prominent one. It is just not common for macroscopic objects to be separated by such short distances, and in case of longer separations, the van der Waals forces quickly fade away. Even surfaces which appear to be extremely smooth are quite rough under an electron microscope, and the resulting mean separation between them is much more than what is needed for van der Waals forces to be effective. However, this can change in near future with the advent of nanotechnology, as artificial surfaces similar to that of gecko footpads are created in laboratories. These could be used to produce non-adhesive sticking materials with broad range of applications.

It is important to realize that the notion of surface (or length, or volume) depends on who is asking and how, the notorious example being the length of the coastline of England [2]. Thus the internal surface of a zeolite might be huge for a water molecule, much smaller for propane, and zero for benzene. An enzyme might seem to be just a globule with potato-like surface, but it can provide a lot of "local" surface for bonding to a substrate during the key–lock mechanism. This leads to the conclusion that when considering the potential importance of van der Waals forces, the scale is crucial, and as their range is limited, the particular scale in question is the nanoscale.

A freshly coined term, sparse matter, tries to capture all the materials where van der Waals forces are important [3]. Sparse matter is defined as having significant regions of very low electron density. Examples are proteins, graphite, nanostructures, zeolites, metal-organic frameworks, molecular crystal or polymers. The regions without electrons provide space for any host molecules or other mesoscopic structures, and the boundary between the electron-rich and electron-deficient areas form the surface which serves as a platform for van der Waals bonding. These bonding surfaces can be even visualized and bonding

surfaces can be discriminated from non-bonding ones [4].

But what is the physics behind van der Waals forces? The notion of bonding surfaces from the previous paragraph hints that it is the touching of electron-rich areas, but in fact, it is almost the exact opposite. It is learned in introductory courses to chemical structure, that the electron density overlap is repulsive, and it is this overlap which actually keeps the high-density regions, usually closed-shell molecules or bigger mesoscopic structures, separated and distinct. The attractive part of the interaction, the van der Waals forces, has a completely different mechanism. In the next section, we discuss the nature of chemical bonding, and how does the term nonlocal correlation from the title of this thesis enter into that discussion.

1.1 Electron correlation

In the simplest view, the driving force for chemical bonding is delocalization of electrons. The Heisenberg principle states that the uncertainty in the velocity of a particle is inversely proportional to the uncertainty in its position. This means that when an electron delocalizes, its movement becomes less vigorous, and its kinetic energy lowers. A hydrogen molecule can serve as an example, where both electrons delocalize over two nuclei, forming a covalent bond. But what keeps all electrons from delocalizing over all nuclei? Why some molecules exist while other do not?

The second crucial part of covalent bonding is the Pauli principle which states that no fermions (electrons being fermions) can occupy the same quantum state. A result of this principle is that there is a limit to how densely electrons can be packed. For example, if four electrons tried to delocalize over two helium nuclei, forming He_2, they would have to be too far from the nuclei, and the delocalization energy would be smaller than the energy loss from not utilizing the Coulomb potential of the nuclei. So when two closed-shell molecules meet each other, their electrons cannot delocalize, and the Pauli principle keeps them apart.

Even when the delocalization occurs, the Pauli principle and the negative charge of electrons still force them to at least *locally* avoid each other. Electrons do this by *correlating* their movements, hence local correlation. Now we finally get to the van der Waals forces. Each moving charge emits electromagnetic waves. In this way, electrons in separate molecules can communicate with each other over distance. Again, they can correlate their movements so this mutual radiation is in sync. In

the language of quantum electrodynamics, an electron in one molecule emits a virtual photon, which travels to the other molecule, and there it is absorbed by another electron. It is exactly this *nonlocal* process, which is behind the existence of van der Waals forces, hence nonlocal correlation.

Strictly speaking, the term van der Waals forces comprises three different kinds of interaction and what has been up to now referred to as van der Waals forces is only one of them and should be referred to as London dispersion forces. The other two types van der Waals interactions are electrostatic forces between charge distributions, be it permanent or induced dipoles, quadrupoles, *etc*. However, this convention is not very rigidly adhered to, as can be seen for example in the name of the van der Waals density functional, which in fact describes only dispersion forces. Therefore, we will use these terms more or less interchangeably, in van der Waals forces putting more stress on the actual physical interaction while in dispersion forces referring strictly to its theoretical description. This seems to be a way of some authors.

Several contributions to the total electronic energy of a molecule were mentioned: the kinetic energy, the Coulomb interaction of electrons with themselves and with the nuclei, and their correlation, local or nonlocal. The density functional theory states that all these contributions can be exactly computed from the knowledge of the electron density itself. As the whole discussion above was done using the density, it seems only natural, but it was not before 1964 that Hohenberg and Kohn proved this rigorously.

1.2 Density functional theory

In the last twenty years, density functional theory (DFT) became a major tool for theoretical investigations of chemical systems [5]. For small systems (molecules, dimers, clusters of small molecules), more accurate wavefunction based methods, often called ab-initio, are available. Either one studies these small systems for the sake of themselves, which is a domain of chemical physics, or one uses them as simplified models of larger systems of interest, which is a common technique. On the other range of the spectrum, large systems (proteins, DNA, polymers) are usually described using semi-empirical methods or forcefields. For the whole midrange, which comprises myriads of interesting chemical systems, DFT is most often used.

The hierarchy stated above stems from the fact that the computational cost of

a method grows with the size of the system, and that the cost of more accurate methods usually grows more rapidly than the cost of less accurate ones. Thus it would be unreasonable to use DFT on small systems when one can obtain more accurate results, and it is impossible to use DFT on the big systems because the calculations would take years or more.

DFT will be presented in more detail in the next chapter, but its deficiencies which stand as a motivation for this thesis can be explained simply. While the theory itself guarantees that it can be exact in principle, the exact expression for electron correlation is not known and only approximations are available. There is a plethora of different approximations, called functionals, and for a long time all of them were dealing only with local correlation. The reason for this is that chemistry traditionally deals with covalent (local) bonding. Therefore, the primary goal was to describe local correlation. Only after this stage was complete, and DFT was able to describe covalent bonds correctly in most situations, the van der Waals forces came into focus.

First, various empirical methods were devised which corrected DFT for weak van der Waals forces, but these were not part of the DFT itself. In sparse matter, the equilibrium structure is usually determined by counterbalanced attractive van der Waals (nonlocal) forces and repulsive Pauli (local) forces. The biggest problem with the empirical approaches is that already the local correlation is approximated and adding empirical corrections to approximate numbers can lead to results with an unpredictable error.

In 2004, a fully DFT-integrated approximation for nonlocal correlation was invented, dubbed van der Waals density functional (vdW-DF). At first, its accuracy was inferior compared to the state-of-art empirical schemes, but it was slowly gaining attention in the physical community as is common with first-principles methods. In recent years however, several extensions were made which improved the accuracy and brought more and more chemical applications.

1.3 Overview of the thesis

The second chapter of this thesis reviews the basics of DFT and the derivation of vdW-DF. The focus is on systematic presentation of the various approximations made. The third chapter reviews contributions that have been made to the field of nonlocal correlation functionals since the original publication. This comprises modifications of the original nonlocal functional, development of new nonlocal

functionals, and modifications of the local functionals which supplement the nonlocal one. The fourth chapter presents several short investigations of the properties and behaviours of vdW-DF that we have made, including decomposition of the nonlocal energy to atom pairs, applying vdW-DF to excited states of excimers, and optimizing its only numerical parameter. The fifth chapter presents a novel empirical correction scheme for DFT, which incorporates mainly van der Waals forces, and which uses vdW-DF as a starting point.

Chapter 2

Van der Waals density functional

The development of the vdW density functional (vdW-DF) can be traced back to 1987 when Langreth and Vosko calculated the exact response of an electron gas at high density [6]. They used their response kernel to calculate a long-range attractive interaction between two particles decreasing as r^{-6} with distance. The initial development resulted in a nonlocal density functional which correctly described asymptotic vdW forces between isolated fragments of matter [7]. The remaining problem was to combine the functional with the established local density functionals. This was first achieved for cases with planar symmetry [8] and subsequently even for general geometries in the work of Dion *et al.* in 2004 [9].

vdW-DF comes from the framework of density functional theory (DFT). Therefore, the next section reviews the basics of DFT. A different approach is followed than is traditionally found in textbooks to present only the components of DFT necessary for presentation of vdW-DF. Atomic units are considered if not stated otherwise.

2.1 Exchange–correlation functional

The electrons in a molecule are fully described by an electronic wavefunction $\Psi(\mathbf{r}_1\mathbf{r}_2\cdots)$.[1] This wavefunction can be in principle obtained by finding the eigenvectors of the appropriate electronic Hamiltonian H and all properties concerning electronic structure can be then computed from this knowledge. One of the most often calculated property is the electronic energy, a single number, while the

[1] The spin of electrons is not explicitly considered as a free parameter of the wavefunction, but rather implicitly assumed. This is for brevity and simplicity, and also in correspondence with how practical calculations are usually performed.

wavefunction itself is too complicated object to be understood in some comprehensible manner [10]. The calculation of the wavefunction thus often seems to be an unnecessary but extremely demanding step. Electron density ρ, which is a much simpler quantity, can also be obtained from Ψ,

$$\rho(\mathbf{r}) = N \int \cdots \int \Psi^*(\mathbf{r}_1 \mathbf{r}_2 \cdots \mathbf{r}_N) \Psi(\mathbf{r}_1 \mathbf{r}_2 \cdots \mathbf{r}_N) \, \mathrm{d}\mathbf{r}_2 \cdots \mathbf{r}_N \tag{2.1}$$

where N is the number of electrons. Density functional theory deals with the question of whether it is possible to obtain the electronic energy directly from the electron density, without the superfluous wavefunction.

The electronic Hamiltonian $\hat{H}$ consists of three parts,[2]

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

where $\hat{T}$ is the kinetic energy operator, $\hat{V}_n$ is the operator due to Coulomb potential from nuclei and $\hat{V}_{ee}$ is the operator due to Coulomb forces between electrons.

A class of fictitious molecular systems can be defined where the Coulomb forces between electrons are scaled by the coupling constant $\lambda \in \langle 0, 1 \rangle$ [11]. At the same time, an additional potential is added to V_n to form the effective potential V_{eff} such that the electron densities of the fictitious and of the real system are identical.[3] The Hamiltonian of such a system is

$$\hat{H} = \hat{T} + \hat{V}_{\mathrm{eff}}(\lambda) + \lambda \hat{V}_{ee} \tag{2.3}$$

For $\lambda = 1$, this is clearly the real molecule and $\hat{V}_{\mathrm{eff}}(1) = \hat{V}_n$.

For $\lambda = 0$, we get the so-called Kohn–Sham system of non-interacting electrons with $\hat{V}_{\mathrm{eff}}(0) = \hat{V}_{\mathrm{KS}}$. This system is easily solvable, because there is no direct interaction between electrons, the all-electron eigenproblem is separable, and it

[2] Born-Oppenheimer approximation is implicitly assumed, as we are interested in electronic structure, not in description of the molecule as a whole.

[3] There is a great amount of theoretical research into the question of whether such effective potential exists for all densities. A particular density for which it exists is called non-interacting v-representable. However important for the rigorous formulation of density functional theory, this problem does not seem to be relevant in practical calculations and we will not consider it here, but simply assume that all densities of real chemical systems are non-interacting v-representable.

leads to one-electron eigenproblems with the solutions ϕ_i. The energy is thus

$$E_{\mathrm{KS}} = \sum_i \langle \phi_i | -\tfrac{1}{2}\nabla_i^2 | \phi_i \rangle + \int v_{\mathrm{KS}}(r)\rho(\mathbf{r})\,\mathrm{d}\mathbf{r} = T_{\mathrm{KS}} + V_{\mathrm{KS}} \tag{2.4}$$

The difference in energy of the real (denoted just E) and of the Kohn–Sham system can be expressed as an integral over the coupling constant [12],

$$E - E_{\mathrm{KS}} = \int_0^1 \frac{\mathrm{d}E}{\mathrm{d}\lambda}\,\mathrm{d}\lambda = \int_0^1 \langle \Psi(\lambda) | \frac{\mathrm{d}\hat{H}}{\mathrm{d}\lambda} | \Psi(\lambda) \rangle\,\mathrm{d}\lambda \tag{2.5}$$

where the Hellman–Feynman theorem was used in the last step. Inserting (2.3) and (2.4) into (2.5) and evaluating the matrix elements, one gets

$$E = T_{\mathrm{KS}} + V_n + \int_0^1 \langle \Psi(\lambda) | \hat{V}_{\mathrm{ee}} | \Psi(\lambda) \rangle\,\mathrm{d}\lambda \tag{2.6}$$

Compare this to the expression for energy which comes directly from (2.2),

$$E = T + V_n + \langle \Psi(1) | \hat{V}_{\mathrm{ee}} | \Psi(1) \rangle \tag{2.7}$$

That is, the substitution of the real kinetic energy for the easily computable Kohn–Sham kinetic energy leads to the integration of V_{ee} over the coupling constant.

The next simplification comes from the fact that $\hat{V}_{ee} = 1/|\mathbf{r}_1 - \mathbf{r}_2| = 1/r_{12}$ acts on two electrons only and that Ψ is antisymmetric. The so-called electron pair density can be formed [13],

$$\rho_2(\mathbf{r}_1, \mathbf{r}_2) = N(N-1)\int\cdots\int \Psi^*(\mathbf{r}_1\mathbf{r}_2\mathbf{r}_3\cdots\mathbf{r}_N)\Psi(\mathbf{r}_1\mathbf{r}_2\mathbf{r}_3\cdots\mathbf{r}_N)\,\mathrm{d}\mathbf{r}_3\cdots\mathbf{r}_N \tag{2.8}$$

where N is the number of electrons. The pair density gives the probability of finding any electron pair at $\mathbf{r}_1$ and $\mathbf{r}_2$. Note that while the density ρ is independent of λ (by definition), the pair density ρ_2 depends on λ.

A major part of V_{ee} is the classical Coulomb repulsion, or the Hartree energy,

$$V_{\mathrm{H}} = \frac{1}{2}\iint \frac{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{r_{12}}\,\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2 \tag{2.9}$$

If the electron density was generated by purely classical objects, not of electrons,

this would be the only part of V_{ee}. After separating V_{H} from V_{ee}, we get

$$E = T_{\mathrm{KS}} + V_n + V_{\mathrm{H}} + E_{\mathrm{xc}} \tag{2.10}$$

where E_{xc} is the so-called exchange–correlation (XC) energy and is equal to

$$E_{\mathrm{xc}} = \frac{1}{2} \iint \frac{\rho_{2,\lambda}(\mathbf{r}_1, \mathbf{r}_2) - \rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{r_{12}} \, \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2 \tag{2.11}$$

where $\rho_{2,\lambda}$ is the λ-averaged electron pair density,

$$\rho_{2,\lambda} = \int_0^1 \rho_2(\lambda) \, \mathrm{d}\lambda \tag{2.12}$$

To better understand the meaning of E_{xc}, we can rewrite it as

$$E_{\mathrm{xc}} = \int \rho(\mathbf{r}) \varepsilon_{\mathrm{xc}}(\mathbf{r}) \, \mathrm{d}\mathbf{r} \tag{2.13}$$

where the energy density $\varepsilon_{\mathrm{xc}}$ is

$$\varepsilon_{\mathrm{xc}}(\mathbf{r}_1) = \frac{1}{2} \int \frac{\rho(\mathbf{r}_2)}{r_{12}} \left(\frac{\rho_{2,\lambda}(\mathbf{r}_1, \mathbf{r}_2)}{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)} - 1 \right) \mathrm{d}\mathbf{r}_2 \tag{2.14}$$

These expressions have a clear interpretation. The total XC energy is given simply by its density $\varepsilon_{\mathrm{xc}}$, integrated over the whole space. The energy density itself is an integral over the whole space, which means that the electron at $\mathbf{r}_1$ feels all the electrons at all distances from $\mathbf{r}_1$. This property is usually referred to as nonlocality. As for the integrand, it consists of two parts, the first one telling more about the quantity of $\varepsilon_{\mathrm{xc}}$, the second one about its quality. The first part, $\rho(\mathbf{r}_2)/r_{12}$, simply means that greater densities in $\mathbf{r}_2$ have greater effect on $\mathbf{r}_1$ and that the further $\mathbf{r}_2$ is from $\mathbf{r}_1$, the weaker its effect.

The second part of the integrand, the expression in parentheses called the pair correlation function h_λ, is more complicated and tells us where the XC energy comes from. In a completely uncorrelated system, that is, in a system where the probability of finding an electron in $\mathbf{r}_1$ is independent on the presence of an electron in $\mathbf{r}_2$, the pair density ρ_2 is just $(N-1)/N \times \rho(\mathbf{r}_1)\rho(\mathbf{r}_2)$, the pair correlation function is constant, $h(\mathbf{r}_1, \mathbf{r}_2) = -1/N$ and $E_{\mathrm{xc}} = -V_{\mathrm{H}}/N$, compensating for the so-called self-interaction. In Hartree energy, an electron interacts with all electrons, including itself, which is nonsensical. It is one of the effects treated by the XC

energy, and in case of the uncorrelated system, it is the only one. In a real molecule however, electrons tend to avoid each other, and h is close to -1 for small r_{12}. Due to the $1/r_{12}$ dependence, which gives greatest weight to smal r_{12}, E_{xc} is much greater that in the hypothetical uncorrelated system. It is yet another manifestation that electron correlation stabilizes electronic systems.

The XC functional is often formally divided into the exchange and correlation parts [11]. The exchange part is formally defined by the Hartree–Fock method, comes from the Pauli principle, is present even in the case of uncharged fermions and is the dominant part of the total XC energy. The correlation part is defined as the rest of the XC energy, comes from the Coulomb repulsion of the electrons and is the sole topic of all the post-HF methods. Note however that the division is only formal and though it can be interpreted physically, it cannot be defined physically.

2.2 Hohenberg–Kohn theorems and approximations

Equation 2.14 is an exact expression for the XC energy density $\varepsilon_{\mathrm{xc}}$. It is however not an expression from the realm of DFT, because it contains the λ-averaged pair density $\rho_{2,\lambda}$. The problem with this fact is that while DFT can be easily and efficiently recast into computational form, there is no such way when pair density is involved. The past decades of DFT development has thus been de facto concerned with the problem of expressing $\rho_{2,\lambda}$ as a functional of ρ only.

The motivation for this effort comes from the Hohenberg–Kohn (HK) theorems, which make DFT a real theory and not only a model [13]. The first HK theorem is crucial and simply states that $\rho_{2,\lambda}$ can be in principle fully restored from the knowledge of ρ only. In fact, it claims even more, namely that the whole electronic wavefunction Ψ can be restored. The proof is remarkably simple and is done by showing that there is a one-to-one correspondence between the ground-state densities ρ and the potentials V_n which in turn determine the corresponding wavefunctions. Thus there is a holy grail in DFT and that is the exact XC functional. Sadly, the HK1 theorem does not provide any clues about his functional should be constructed, or in other words, how $\rho_{2,\lambda}$ should be obtained from ρ. Thus all current DFT implementations use only approximations to $\varepsilon_{\mathrm{xc}}$. The number of different approximations is immense. In following paragraphs, some approaches

are briefly reviewed and discussed.[4]

The crudest approximation actually predates the HK theorems and DFT itself [14]. Here, the extremely complicated nonlocal density functional in (2.14) is approximated by a simple local density *function*, hence local density approximation (LDA),

$$\varepsilon_{\mathrm{xc}}[\rho](\mathbf{r}) \approx \varepsilon_{\mathrm{xc}}^{\mathrm{LDA}}\big(\rho(\mathbf{r})\big) \tag{2.15}$$

The question remains about the function $\varepsilon_{\mathrm{xc}}^{\mathrm{LDA}}$. There is no known way how to obtain it directly from the formula for exact $\varepsilon_{\mathrm{xc}}$. An indirect solution is to obtain it from some exactly solvable system. To my knowledge, the only one which got any serious attention in this regard is the homogeneous electron gas (HEG) and this version of LDA is an ingredient present in every DFT calculation today. HEG is defined as a gas of electrons on a homogeneous background of positive charge. Its energy has been solved to arbitrary accuracy by quantum Monte Carlo calculations. It has a uniform electron density and it is this mapping between its density and its energy which parametrizes LDA.

Given the crude approximation to $\varepsilon_{\mathrm{xc}}$ that $\varepsilon_{\mathrm{xc}}^{\mathrm{LDA}}$ is, it is remarkable how well it works. It correctly predicts structural properties of molecules and solids and its accuracy in energies is comparable to that of the Hartree–Fock method or better. At the time when LDA was the state-of-the-art DFT approximation, more accurate wavefunction-based methods were available and LDA was only rarely used by chemists. However, it was a common tool in the solid-state physics. The reason why LDA is so surprisingly good came to be understood only later when different approximations, seemingly better, performed actually worse.

One of the first attempts to go beyond LDA was the gradient expansion approximation (GEA). The idea is to Taylor-expand $\varepsilon_{\mathrm{xc}}$ around $\varepsilon_{\mathrm{xc}}^{\mathrm{LDA}}$ by utilizing the density gradient $\nabla\rho$. Unexpectedly, this expansion turned out to be worse than LDA. The reason behind that is that $\varepsilon_{\mathrm{xc}}^{\mathrm{LDA}}$ is of a real physical system (the uniform electron gas) while $\varepsilon_{\mathrm{xc}}^{\mathrm{GEA}}$ is not. This is manifested by $\varepsilon_{\mathrm{xc}}^{\mathrm{LDA}}$ satisfying several important bounds, limits and integral rules. An example of such a rule is that $\int \rho(\mathbf{r}_2)h(\mathbf{r}_1,\mathbf{r}_2)\,\mathrm{d}\mathbf{r}_2 = -1$. We could see that in the uncorrelated system, $h = -1/N$ and this rule is trivially satisfied. It is more difficult to show that $\varepsilon_{\mathrm{xc}}^{\mathrm{LDA}}$ satisfies it as well but it can be done. On the contrary, $\varepsilon_{\mathrm{xc}}^{\mathrm{GEA}}$ breaks this rule.

[4] The second HK theorem is not relevant for the present discussion and stands behind the above-mentioned fact that DFT can be recast into a computational form. HK2 states that the real ground-state density can be obtained from scratch by minimizing the total energy over all reasonable densities.

A solution to this problem is the so-called generalized gradient approximation (GGA). Here, the dependence on $\nabla\rho$ is retained but instead of doing the Taylor expansion, the functional is constructed in such a way that all known bounds, limits and integral rules are satisfied. The general form of GGA functionals is

$$\varepsilon_{\mathrm{xc}}[\rho](\mathbf{r}) \approx \varepsilon_{\mathrm{xc}}^{\mathrm{GGA}}\big(\rho(\mathbf{r}), \nabla\rho(\mathbf{r})\big) \tag{2.16}$$

The interpretation of the GGA approach is that here the functional knows not only about the point $\mathbf{r}_1$ where the energy density is evaluated but also about its near neighborhood. Therefore, they are called semi-local sometimes, in contrast to LDA.

While LDA is only one, there are dozens of different GGA functionals, caused by the fact that apart from several exact properties which are satisfied by most of them, there is no way how to distinguish the best functional. The only measure is thus their performance and it differs for different types of chemical or solid-state systems. GGA enabled to reach the chemical accuracy of 1 kcal/mol and brought DFT to the attention of chemists. This in turn lead to faster development and more functionals and even new classes of DFT approximations, which we will mention only briefly.

The next natural step beyond GGA is to include more quantities than just the density and its gradient. These approaches are called meta-GGA functionals and two quantities that are most often used is the kinetic energy density and the Laplacian $\nabla^2\rho$. Again, the meta-GGA functionals have more information about the neighborhood than in the pure GGA case.

The most used functional in chemistry today (due to its universal accuracy) is probably the B3LYP functional. It is a hybrid functional and that means that apart from the GGA part, part of the exchange energy is obtained by evaluating exact Hartree–Fock (HF) exchange on the Kohn–Sham orbitals which are used for the kinetic energy. The word "hybrid" refers to the fact that the resulting method is a hybrid between pure DFT and the HF method. There are even some theoretical considerations which justify this approach by considering the λ-integration in the exact XC functional. The reason why the exact exchange is not used in its whole but only partly mixed is that the HF exchange is nonlocal while the DFT correlation is local and therefore they are not fully compatible.

The next extension to hybrid functionals is to include the MP2 correlation, which leads to double hybrid functionals. However, these functionals are not

much used yet due to their unfavourable computational cost (equal to MP2).

2.3 Nonlocal correlation functional

All the pure DFT XC functionals approximations mentioned above (LDA, GGAs, meta-GGAs) are local from a strict mathematical point of view. Indeed, the total energy of two perfectly separated electron systems is equal to the sum of the energies of the individual systems in these approximations. Thus they cannot in any way account for the true dispersion forces, which are the topic of this thesis. In some cases, LDA or some GGAs can bind vdW systems, even overbind them, but this is due to spurious exchange binding, which provides wrong asymptotic interaction between other things [15]. In contrast, it can be shown that vdW forces should be included in the correlation part of the XC energy. Indeed, the HF method having only exchange also does not cover vdW forces. Therefore, even hybrid functionals, which are nonlocal, do not cover vdW forces. Double hybrid functionals cover vdW interaction but not through a density functional but through their MP2 ingredient. Thus to cover vdW forces within DFT, one had to have a fully nonlocal correlation functional. Precisely such functional was devised by Dion *et al.* in 2004 and it is presented in this section. Only physical considerations and approximations are shown here, the often lengthy mathematics is left out.

The departing point for the nonlocal functional is Equation 2.11. The first step is to express ρ_2 in terms of the density response function χ [12, 16], defined as

$$\delta\rho(\omega, \mathbf{r}_1) = \int \chi(\omega, \mathbf{r}_1, \mathbf{r}_2)\varphi_{\text{ext}}(\omega, \mathbf{r}_2)\,\mathrm{d}\mathbf{r}_2 \tag{2.17}$$

The response function relates the perturbing external potential φ_{ext} with frequency ω at point $\mathbf{r}_2$ with the change in electron density at point $\mathbf{r}_1$. The reason why a dynamic (frequency-dependent) quantity is introduced is that it captures a great amount of information about the behaviour of a system. Also note that it is customary in physics to work with complex dynamic quantities because it eases the underlying mathematics. In the end, only their real parts correspond to measured values. Using χ, the pair density can be expressed as [16]

$$\rho_{2,\lambda}(\mathbf{r}_1, \mathbf{r}_2) = \int \frac{\mathrm{d}\omega}{\pi \mathrm{i}} \chi_\lambda(\mathbf{r}_1, \mathbf{r}_2, \omega) + \rho(\mathbf{r}_1)\rho(\mathbf{r}_2) - \delta(r_{12})\rho(\mathbf{r}_1) \tag{2.18}$$

Inserting (2.18) into (2.11), one gets

$$E_{\mathrm{xc}} = \int \frac{\mathrm{d}\omega}{2\pi\mathrm{i}} \iint \chi_\lambda(\omega, \mathbf{r}_1, \mathbf{r}_2) r_{12}^{-1} \,\mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2 - E_{\mathrm{self}} \tag{2.19}$$

where $E_{\mathrm{self}} = \frac{1}{2} \iint \rho(\mathbf{r}_1) r_{12}^{-1} \delta(r_{12}) \,\mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2$ is a divergent quantity called self-energy (not to be confused with the self-interaction), which is cancelled by divergence of the first expression.

The next ingredient is the screened response function $\tilde{\chi}$ (as compared to the bare response χ). It is defined in a similar way as χ in (2.17) but instead of the external potential φ_{ext}, the total (or screened) potential φ is considered. It can be shown that χ and $\tilde{\chi}$ are related by a Dyson-like equation,

$$\chi(\mathbf{r}_1, \mathbf{r}_2) = \tilde{\chi}(\mathbf{r}_1, \mathbf{r}_2) + \iint \tilde{\chi}(\mathbf{r}_1, \mathbf{r}) |\mathbf{r} - \mathbf{r}'| \chi(\mathbf{r}', \mathbf{r}_2) \,\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}' \tag{2.20}$$

In the implicit matrix notation,[5] adopted from now on for brevity, Equation 2.20 becomes

$$\chi = \tilde{\chi} + \tilde{\chi} V \chi \tag{2.21}$$

where V is the Coulomb potential $V(\mathbf{r}, \mathbf{r}') = 1/|\mathbf{r} - \mathbf{r}'|$.

In the DFT context, χ depends on the coupling constant λ and (2.21) holds only for $\chi(\lambda)$, not for the averaged χ_λ. Taking the λ-integration out, utilizing (2.21) and using the matrix notation, (2.19) becomes

$$E_{\mathrm{xc}} = \int \frac{\mathrm{d}\omega}{2\pi\mathrm{i}} \int_0^1 \mathrm{Tr}\left[\frac{\tilde{\chi}(\lambda) V}{1 - \lambda \tilde{\chi}(\lambda) V}\right] \mathrm{d}\lambda - E_{\mathrm{self}} \tag{2.22}$$

This is the last exact expression in this derivation.

The *first* approximation in constructing the nonlocal XC functional is done by approximating $\tilde{\chi}(\lambda)$ by $\tilde{\chi}(1) = \tilde{\chi}$, the full potential approximation (FPA). This is done so that the λ-integration can be carried out at this stage. It can be shown that FPA is asymptotically exact for correlation and exact for exchange. Thus the only region where it can negatively influence accuracy is the vdW bonding distance, which is, on the other hand, usually the most important one. A different

[5] In this notation, the nonlocal (dependent on two positions) quantities are considered to be infinite matrices, the first independent variable being the row index and the second being the column index. The integration $\int f(x_1, y) g(y, x_2) \,\mathrm{d}y$ then becomes a simple matrix multiplication fg, and $\int f(x, x) \,\mathrm{d}x$ is the trace of f, $\mathrm{Tr}\, f$. Likewise $h = g/f = f^{-1} g$ means $fh = g$.

approach, called random phase approximation (RPA), is also often used, and it is defined by replacing $\tilde{\chi}(\lambda)$ by $\chi(0)$ (the bare response). RPA is usually used to treat also other effects than dispersion which are inaccurate in standard DFT. Under FPA, the λ-integration is easy and one obtains

$$E_{\mathrm{xc}} = -\int \frac{\mathrm{d}\omega}{2\pi\mathrm{i}} \mathrm{Tr}[\ln(1-\tilde{\chi}V)] - E_{\mathrm{self}} \tag{2.23}$$

In the next step, a different type of response function is introduced, the dielectric function ϵ,

$$\tilde{\chi} = \nabla \cdot \frac{\epsilon - 1}{4\pi} \nabla \tag{2.24}$$

It is a direct quantum-mechanical equivalent to the classical permittivity in electromagnetism, as is χ to the susceptibility.

While FPA and subsequent approximations are suitable for the long-range vdW forces, they are not so much for the local short-range exchange and correlation. Also, it would be unreasonable not to utilize the vast experience gathered in the semi-local GGA functionals. Therefore, one would like to extract only the long-range part from (2.23). This is done by first calculating the energy of HEG under FPA,

$$E_{\mathrm{xc}}^{0} = -\int \frac{\mathrm{d}\omega}{2\pi\mathrm{i}} \mathrm{Tr}[\ln(\epsilon)] - E_{\mathrm{self}} \tag{2.25}$$

The *second* approximation is then subtracting this from (2.23) and assuming we have subtracted the energy of HEG exactly, which is not true as it in fact was calculated under FPA. Nevertheless, one gets

$$E_{\mathrm{xc}}^{\mathrm{nl}} = -\int \frac{\mathrm{d}\omega}{2\pi\mathrm{i}} \mathrm{Tr}\left[\ln \frac{1-\tilde{\chi}V}{\epsilon}\right] \tag{2.26}$$

The exchange has no long-range component and $E_{\mathrm{xc}}^{\mathrm{nl}}$ covers mainly correlation. Therefore, we will write only $E_{\mathrm{c}}^{\mathrm{nl}}$ from now on and speak of nonlocal *correlation* functional.

The *third* approximation enables carrying out the λ-integration. In trying to perform the λ-integration, one hits the logarithm in the integrand. The way around is to express it in terms of $S = 1 - 1/\epsilon$ and then expand into second order. Thus we arrive at

$$E_{\mathrm{xc}}^{\mathrm{nl}} = -\int \frac{\mathrm{d}\omega}{4\pi\mathrm{i}} \mathrm{Tr}\left[S^2 - \left(\frac{\nabla S \cdot \nabla V}{4\pi}\right)^2\right] \tag{2.27}$$

The *fourth* approximation is connected to expressing S in terms of density. This is done in a manner similar to construction of GGA functionals. Several known exact limits and other behaviours are stated and then a simplest possible analytical form is devised which satisfies these rules. For this step, S is Fourier-transformed into its momentum representation $S(\omega, \mathbf{k}_1, \mathbf{k}_2)$ where $\mathbf{k}_1, \mathbf{k}_2$ are the wave vectors. Four required behaviours are used, which are not presented here as their interpretation would require much deeper inquiry into solid-state physics. The form of S which obeys them is

$$\begin{aligned} S(\omega, \mathbf{k}_1, \mathbf{k}_2) &= \tfrac{1}{2}\left[\tilde{S}(\omega, \mathbf{k}_1, \mathbf{k}_2) + \tilde{S}(\omega, -\mathbf{k}_2, -\mathbf{k}_1)\right] \\ \tilde{S}(\omega, \mathbf{k}_1, \mathbf{k}_2) &= \int \mathrm{e}^{-\mathrm{i}(\mathbf{k}_1 - \mathbf{k}_2)\cdot\mathbf{r}} \frac{4\pi\rho(\mathbf{r})}{\left[\omega + \omega_k(k_1, \mathbf{r})\right]\left[-\omega + \omega_k(k_2, \mathbf{r})\right]} \,\mathrm{d}\mathbf{r} \end{aligned} \tag{2.28}$$

where ω_k is the dispersion function relating frequency to energy.

Still, ω_k is unknown and has to be expressed in terms of density. In this case, the physical motivation for a particular formula is quite simple. In electron gas, there are two types of excitations. The short-wavelength type are the Fermi level excitations where an electron from an occupied orbital is excited to an unoccupied orbital, creating the electron–hole pair. These excitations behave like $\omega_k \propto k^2/2$ for large k. The long-wavelength type are the so-called plasmons, which are a collective coordinate motion of many electrons, and their energy is nearly constant for small k.

The *fifth* approximation is to utilize these two limits and have a form for ω_k which switches between the constant and the quadratic behaviour,

$$\omega_k(k, \mathbf{r}) = \frac{k^2}{2}\left[1 - \exp\left(-\frac{4\pi}{9}\frac{k^2}{k_0(\mathbf{r})}\right)\right]^{-1} \tag{2.29}$$

with k_0 specified such that if (2.25) was used to calculate the energy of HEG with this k_0, it would give the correct energy. This condition leads to

$$k_0 = k_\mathrm{F}\frac{\varepsilon_\mathrm{xc}^0}{\varepsilon_\mathrm{x}^\mathrm{LDA}} \tag{2.30}$$

where $k_\mathrm{F}^3 = 3\pi^2\rho$ is the Fermi wave vector and $\varepsilon_\mathrm{xc}^0$ is the XC energy density which would reproduce the full calculation, $E_\mathrm{xc}^0 = \int \rho(\mathbf{r})\varepsilon_\mathrm{xc}^0(\mathbf{r})\,\mathrm{d}\mathbf{r}$.

The last approximation is thus contained in approximating $\varepsilon_\mathrm{xc}^0$. The only natural thing to do would be to take it from the same GGA functional which will be in

the end used to approximate E_{xc}^{0} and added to E_{c}^{nl}. However, this approach does not work because most modern GGA functionals get saturated in low-density tails, which would lead to unwanted low-density behaviour of ω_k if used here. Therefore, a simple LDA with exchange gradient correction is used,

$$\varepsilon_{xc}^{0} \approx \varepsilon_{xc}^{LDA} - \varepsilon_{x}^{LDA} \frac{Z_{ab}}{9} s^2 \tag{2.31}$$

where $s = |\nabla\rho|/2k_F\rho$ is the reduced gradient and $Z_{ab} = -0.8491$ is taken from gradient expansion for slowly varying electron gas.

2.4 Numerical evaluation

Expressions 2.27, 2.28, 2.29, 2.30 and 2.31 give the complete definition of E_{c}^{nl}. In this raw form however, its computation is numerically unfeasible as it contains a quadruple integral. Thanks to the choice of analytical forms during the derivation, it can be rewritten in the form

$$E_{c}^{nl} = \iint \rho(\mathbf{r}_1)\Phi[\rho](\mathbf{r}_1, \mathbf{r}_2)\rho(\mathbf{r}_2)\,\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2 \tag{2.32}$$

where Φ is the so-called nonlocal kernel, which is a functional of density, namely depending on the values of ρ and $\nabla\rho$ at points $\mathbf{r}_1$, $\mathbf{r}_2$ and on their distance. Better yet, it can be cast as a function of two variables only, defined as

$$D = \tfrac{1}{2} r_{12}\big(k_0(\mathbf{r}_1) + k_0(\mathbf{r}_2)\big), \qquad \delta = \frac{|k_0(\mathbf{r}_1) - k_0(\mathbf{r}_2)|}{k_0(\mathbf{r}_1) + k_0(\mathbf{r}_2)} \tag{2.33}$$

where $0 \leq D < \infty$ is a scaled distance (as if bigger densities were further from each other) and $0 \leq \delta < 1$ is the dimensionless measure of asymmetry between $\mathbf{r}_1$ and $\mathbf{r}_2$. The kernel $\Phi(D, \delta)$ can be easily tabulated for suitable values (D_i, δ_j) and interpolated when evaluating the integral in (2.32).

Equation 2.32 can be interpreted as interaction of two densities at points $\mathbf{r}_1$, $\mathbf{r}_2$ via the nonlocal potential Φ. Figure 2.1 shows the plots of $4\pi D^2\Phi$ vs D for several values of δ. It can be seen that the interaction is repulsive at shorter distances, becomes attracting and has a minimum at mid-range distances, and goes to zero at longer distances. The interaction is strongest between equivalent densities ($\delta = 0$). In case of HEG ($\delta = 0$ everywhere), $\int 4\pi D^2\Phi\,\mathrm{d}D = 0$, which means that

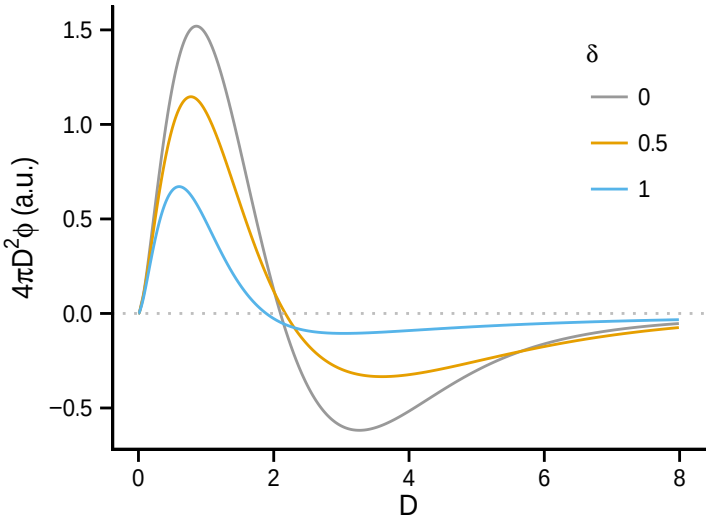

Figure 2.1 | Nonlocal kernel Φ for several values of δ. The $4\pi D^2\Phi$ quantity on the *y*-axis is proportional to the contribution to the XC energy density from the electron density in "distance" *D* in the homogeneous electron gas.

the nonlocal correlation does not change the energy of HEG, as it should.

A final word must be said about approximating the "local" energy E_{xc}^0, which was subtracted during derivation of vdW-DF from the total XC energy,

$$E_{\mathrm{xc}} = E_{\mathrm{xc}}^0 + E_{\mathrm{c}}^{\mathrm{nl}} \tag{2.34}$$

E_{xc}^0 is calculated as the energy of HEG under FPA, and therefore, it is equivalent to LDA in some sense. However, in contrast to LDA, the mathematical form of E_{xc}^0 is not local and it can behave differently. Using LDA in place of E_{xc}^0 would not be the most appropriate thing to do. Rather, Dion *et al.* argue that E_{xc}^0 should be calculated as GGA exchange and LDA correlation. $E_{\mathrm{c}}^{\mathrm{nl}}$ covers mostly correlation, because there is no long-range exchange and E_{xc}^0 contains the majority of exchange. Hence, the use of GGA for exchange is in place. On the other hand, all nonlocal correlation is assumed to be in $E_{\mathrm{c}}^{\mathrm{nl}}$ and using GGA correlation could lead to double counting, hence the use of LDA is advocated.

The question remains which GGA exchange to use. There is no first-principles guidance, and the best GGA flavour can be chosen only based on numerical results. In the original paper from 2004, it was suggested to use the revised

PBE exchange functional. Original PBE is one of the most successful ab-initio functionals due to its versatility, but one of its flaws is that it binds vdW systems by exchange, an unwanted effect. The revised PBE corrects for this deficiency by reparameterizing one of the parameters in PBE.

Chapter 3

Nonlocal functionals since 2004

The vdW-DF method established a new sub-field in the DFT world. Until then, dispersion was treated only empirically, outside of the DFT regime and vdW-DF changed this paradigm. From the beginning, it was clear that while vdW-DF is a promising approach, it has deficiencies. Early tests on database benchmarks and individual cases showed that while it captures the vdW interactions qualitatively well, it does not have sufficient accuracy [3]. It tends to significantly overestimate binding energies and also overestimate interfragment equilibrium distances. This motivated a whole new research which was built around the idea of vdW-DF and whose goal was more accurate description of vdW interactions within the DFT framework. Since than, parallel approaches have been also devised, using different approximations, for example RPA in the work of Tkatchenko *et al.* [17]. However, the following review deals only with works directly descended from the vdW-DF approach.

There are two main branches of efforts to improve the accuracy of vdW-DF. The position and depth of a vdW minimum is given predominantly by counterbalancing the attractive vdW forces (E_c^{nl}) and the repulsive electron exchange (contained in E_{xc}^0. Consequently, one can strive to improve description of the former or of the latter. The next two sections follow this categorization and attempt to chronologically map the vdW DFT research between 2004 and 2012. Inevitably, there is some overlap, but usually the stress is on one of the two categories.

3.1 Changing the nonlocal functional

In 2010, a second version of vdW-DF was introduced by collaborators of the authors of the original vdW-DF, dubbed vdW-DF2 [18]. There are only two minor changes, which however led to better accuracy. The first change is in the

Z_{ab} parameter. In vdW-DF, $Z_{\mathrm{ab}} = -0.8491$ is derived from a simple gradient expansion. This is suitable for slowly varying electron density, but that is not the case in molecules. In vdW-DF2, $Z_{\mathrm{ab}} = -1.887$ is derived from the large-N asymptote. This is a common technique in DFT development, where a series of hypothetical atoms with growing number of electrons N is considered and especially the behaviour when $N \to \infty$ is investigated. The large-N asymptote is believed to be closer to the reality of atoms and molecules than the slowly-varying limit (which is more suitable for solids). The Z_{ab} parameter controls the screening of vdW interaction and its bigger value in vdW-DF2 means that the nonlocal interaction is weaker here. To compensate for this fact, the second change from vdW-DF is made by using a different local functional, namely the revised PW86 for exchange. Tests on several systems including the S22 set show that vdW-DF2 is indeed more accurate for description of vdW interactions between molecules.

The only true alternative to the nonlocal functional by Dion *et al.* was presented in a series of papers by Vydrov and Van Voorhis (VV). They begun in 2009 by slightly modifying vdW-DF into what they called vdW-DF-09 [19]. Instead of Eq. 2.28, they used

$$S(\omega, \mathbf{k}_1, \mathbf{k}_2) = \int \mathrm{e}^{-\mathrm{i}(\mathbf{k}_1 - \mathbf{k}_2)\cdot\mathbf{r}} \frac{4\pi\rho(\mathbf{r})}{\frac{1}{4}\left[\omega_k(k_1, \mathbf{r}) + \omega_k(k_2, \mathbf{r})\right]^2 - \omega^2} \,\mathrm{d}\mathbf{r} \tag{3.1}$$

and instead of Eq. 2.29, they used

$$\omega_k(k, \mathbf{r}) = \tfrac{1}{2}k^2 + \tfrac{1}{3}k_F(\mathbf{r})\left(1 + \lambda s(\mathbf{r})^2\right) \tag{3.2}$$

where λ is fitted to experimental C_6 coefficients. These changes lead to simpler integrals and a second-order gradient expansion of $E_{\mathrm{c}}^{\mathrm{nl}}$ can be computed. This can be shown to behave undesirably, but a simple semi-local gradient correction $E_{\mathrm{GC}} = \int \rho(\mathbf{r})\varepsilon_{\mathrm{GC}}(\mathbf{r})\,\mathrm{d}\mathbf{r}$ can be introduced which cancels this improper behaviour.

In vdW-DF-09, a different local functional is also used—the LC-ωPBE functional. This functional is from a class of range-separated hybrid functionals, where a short-range exchange part is taken from some semi-local XC functional (PBE in this case), while the long-range exchange part comes from the HF method. LC-ωPBE gives usually more repulsive interaction curves than revPBE used for vdW-DF. In contrast to the original vdW-DF, a semi-local correlation is used in combination with the nonlocal correlation functional, but this change is not discussed by the authors. vdW-DF-09 was tested on the argon dimer, carbon

monoxide dimer and benzene–argon complex. It gives somewhat more accurate binding energies compared to vdW-DF, but the equilibrium geometries are not improved.

The next step of VV was to introduce their own nonlocal correlation functional, dubbed VV09. The main message of VV09 was that the nonlocal functional can be simpler than the original vdW-DF [20]. This was achieved by using

$$S(\omega, \mathbf{k}_1, \mathbf{k}_2) = \int \mathrm{e}^{-\mathrm{i}(\mathbf{k}_1-\mathbf{k}_2)\cdot\mathbf{r}} \frac{4\pi\rho}{C\left|\frac{\nabla\rho}{\rho}\right|^4 + \frac{4\pi\rho}{3} - \omega^2} \exp\left(-\frac{\pi}{4}\frac{k_1^2+k_2^2}{k_\mathrm{F}\phi^2}\right) \mathrm{d}\mathbf{r} \tag{3.3}$$

where $\phi = \frac{1}{2}\left[(1+\zeta)^{\frac{2}{3}} + (1-\zeta)^{\frac{2}{3}}\right]$ is the spin-scaling factor ($\zeta = (\rho_\uparrow - \rho_\downarrow)/\rho$) and C is again fitted to reproduce C_6 coefficients. The gradient expansion of E_c^nl with this S is correctly a constant, in contrast to vdW-DF and vdW-DF-09. Different exact constraints were used in construction of this S than in vdW-DF. The main advantage of the form (3.3) is that the $\mathbf{r}$,ω-integration can be performed analytically, leading to expression for E_c^nl containing only a double integral over space.

The VV09 functional initiated an exchange of letters between Langreth and Lundqvist (LL) [21], and Vydrov and Van Voorhis [22]. LL criticize VV09 on the basis that it does not satisfy some exact theorems, such as conservation of electrons or the high-frequency limit. In their reply, VV explain that their formulas need to be understood only in the context of VV09.

The last contribution of VV to the topic came with the VV10 functional [23]. They provide no physically motivated derivation, but rather just define VV10 and then show that it satisfies several limits and constraints. Compared to all previous nonlocal functionals, VV10 is extremely simple. The double-integral form of (2.32) is not a final form here, but rather a starting one. The nonlocal kernel is defined as

$$\Phi[\rho](\mathbf{r}_1, \mathbf{r}_2) = -\frac{3}{2}\frac{1}{g(\mathbf{r}_1)g(\mathbf{r}_2)[g(\mathbf{r}_1)+g(\mathbf{r}_2)]} \tag{3.4}$$

where

$$g = r_{12}^2\sqrt{C\left|\frac{\nabla\rho}{\rho}\right|^4 + \frac{4\pi\rho}{3}} + b\frac{3\pi}{2}\left(\frac{\rho}{9\pi}\right)^{\frac{1}{6}} \tag{3.5}$$

and C, b are two empirical parameters of VV10. C controls the long-range behaviour (C_6 coefficients), while b controls the short-range dumping. The pa-

rameter b can be used to combine VV10 with almost any local functional. In the original paper, VV10 is combined with refitted PW86 exchange and PBE correlation and b is fitted to S22 energies. On all test systems, VV10 performs significantly better than vdW-DF2, reaching a several times better accuracy in binding energies.

3.2 Appropriate exchange functional

The other branch of research aiming at higher accuracy with nonlocal correlation functionals deals with the proper local functional for E_{xc}^0. It has been recognized that the crucial role is played by the exchange part [24]. This is related to the fact that exchange is responsible for the repulsive wall in vdW bonding and its steepness and position determine the resulting vdW minimum.

Pernal *et al.* constructed a semi-local functional which should supposedly contain no short-range dispersion, making it a perfect match for nonlocal correlation functional [25]. They used the form of the M05-2X functional and fitted its 11 parameters to CCSD(T) interaction energies of 9 small dimers from which SAPT(DFT) dispersion terms were subtracted. Regrettably, they combined their dispersion-less density functional (dlDF) with Grimme's semi-empirical dispersion correction scheme, not with a nonlocal correlation functional. Their approach provided very accurate interaction energies for various database benchmarks. It would be interesting to see how dfDL would work with the VV10 functional if its parameter b was optimized for dfDL.

Cooper devised a non-empirical exchange functional, $C09_x$, specifically for vdW-DF [26]. He uses two constrains: (i) in the small-s region (s is the reduced density gradient), he matches his functional to the gradient expansion approximation used in the early days of DFT. (ii) In the large-s region, $C09_x$ approaches the revPBE exchange. It is quite remarkable that this simple construction achieves a great reduction in errors on the S22 set and other test systems. It actually reaches the accuracy of top empirical DFT dispersion schemes.

Klimeš *et al.* systematically investigated the choice of the exchange functional to be combined with vdW-DF [27]. Apart from comparing PBE, B86, B86 with modified gradient correction, PW86 and B88, they also reoptimized the PBE and B88 functionals on the S22 set, with the vdW-DF nonlocal energy added. They show the transferability of their optimizations on water hexamers and adsorption of water on solid NaCl. These reoptimized functionals have also been shown to

perform better than the original vdW-DF for solid-state properties, such as lattice constants or atomization energies [28].

Austin *et al.* came with another local functional which should be void of short-range vdW forces [29]. They suggest a simple linear combination of the B3PW91 (41%) and PBE1PBE (59%) functionals, which mimics the HF exchange well. Again, they devise their own correction scheme for dispersion and do not try its combination with any of the nonlocal functionals.

All the approaches reviewed so far are usually constructed in vdW interactions in mind and they are rarely tested on anything else. This is potentially dangerous because the semi-local GGA functionals have been proven in thousands studies to work very well for covalent and ionic bonding and it is not clear whether the new functionals retain this quality. This problem was tackled by Wellendorf *et al.* who used machine-learning techniques to parametrize from scratch a new GGA exchange and a combination of LDA, PBE and vdW-DF2 correlation, using ten different benchmark databases spanning molecular, solid-state and noncovalent systems and including equilibrium and reaction properties [30]. The optimal correlation mix is found to be 60% LDA, 40% PBE and 100% vdW-DF. They also show that the optimal GGA exchange greatly depends on which database is used for parametrization. This evidences that there is a delicate balance of exchange repulsion and vdW attraction in the vdW equilibrium and that no GGA-type exchange functional can work for all possible systems. A more intricate exchange functional would be needed.

Indeed, it seems that the description of vdW forces within DFT came to the point that its quality supersedes that of the local functionals. The vdW interaction energies are usually in the range of units to tens of kJ/mol. But the GGA functionals have been developed for description of energies which are by an order of magnitude larger. While the percentage difference between different GGA functionals might be several percent for covalent bonding, it may well be tens of percent for vdW interactions. This resulted in the development of new suitable exchange functionals described above, but none of these efforts are systematic because there is no way how to divide the mid-range correlation into the short- and long-range parts in a sound way. What is missing here is a united local and nonlocal functional which would be constructed together.

3.3 Implementation

This final section deals with development concerning the implementation of nonlocal functionals. The simplest implementation of vdW-DF requires evaluating a double (six-dimensional) integral, which scales as N^2 with the size of a system. This is unfavourable because it prohibits its use for bigger system, which are precisely the target systems, being untreatable by ab-initio QM methods and often being determined by vdW interactions.

Román-Pérez and Soler developed a way to evaluate (2.32) which scales as $N \log N$ with the size of a system [31]. If the kernel $\Phi(k_0(\mathbf{r}_1), k_0(\mathbf{r}_2), r_{12})$ was independent on $\mathbf{r}_1$, $\mathbf{r}_2$, the double integral would be a plain convolution and it could be evaluated by Fourier methods. The idea is to expand Φ in terms of $\Phi(k_{0,i}, k_{0,j}, r_{12})$ where $k_{0,l}$ are fixed values. It is equivalent to interpolation. After this expansion, the double integral over position vectors can be Fourier-transformed into a single integral over wave vectors. Using this implementation, the computational cost of vdW-DF is of the order of magnitude of a plain GGA calculation even for systems containing several hundreds of atoms. It was actually this advancement which initiated the heavy use of vdW-DF in computational studies.

The knowledge of a density functional is sufficient for calculation of the energy, but not enough for obtaining the electron density by means of the KS scheme. For that, the XC potential v_{xc} has to be known, which is a functional derivative of the energy functional,

$$v_{\mathrm{c}}^{\mathrm{nl}} = \frac{\delta E_{\mathrm{c}}^{\mathrm{nl}}}{\delta \rho} \tag{3.6}$$

The early implementations of vdW-DF were done non-self-consistently, that is the density was obtained using only the local XC functional and afterwards, the total vdW-DF was evaluated on this density. It has been argued that the vdW-DF interaction is too weak to has any significant impact on the density, but there was no definitive proof. Thonhauser *et al.* derived the potential and illustrated that the effect of performing self-consistent calculation is negligible [32]. The slight change in electron density corresponding the self-consistent use of the nonlocal functional is identified to be a flow of density from the atoms to the areas between the fragments, which maximizes the interaction. Vydrov *et al.* independently derived the matrix elements of the XC potential needed in the self-consistent

calculation [33],

$$\langle \mu | v_{\mathrm{c}}^{\mathrm{nl}} | \nu \rangle = \frac{\mathrm{d}E_{\mathrm{c}}^{\mathrm{nl}}}{\mathrm{d}P_{\mu\nu}} \tag{3.7}$$

where $P_{\mu\nu}$ is the density matrix and μ, ν are the basis set functions.

Chapter 4

Investigations of vdW-DF

This chapter presents several of our own investigations of vdW-DF. They deal with the nature of dispersion interaction, with dispersion interaction in excited states and with optimization of the Z_{ab} parameter in vdW-DF. We use our custom code for evaluation of the vdW-DF nonlocal functional documented in Appendix A.

4.1 Nonlocal energy decomposition

The nonlocal correlation energy is given by (2.32) which reduces the nonlocal six-dimensional kernel Φ to only one number. To better understand the nonlocal energy, one can plot the kernel in terms of D and δ (Figure 2.1), but this still does not shed light on how the nonlocal energy looks when evaluated on a real system. The full six-dimensional kernel cannot be visualized, but it can be half-integrated into the energy density, a three-dimensional quantity, whose visualization is already feasible.

As a model system, we consider the argon dimer, which is a prototypical vdW system. Our implementation of vdW-DF uses Becke's integration scheme, which uses atom-centered quadrature grids. This straightforwardly divides the total electron density ρ on into individual atom densities ρ_i, $\rho = \sum_i \rho_i$. Accordingly, it translates any integral over the whole space into a sum of integrals over atoms. Using this property, we can rewrite the total nonlocal energy of the argon dimer defined in (2.32) as

$$\tfrac{1}{2}\iint \rho\Phi\rho = \tfrac{1}{2}\iint \rho_1\Phi\rho_1 + \tfrac{1}{2}\iint \rho_2\Phi\rho_2 + \iint \rho_1\Phi\rho_2 \tag{4.1}$$

where ρ_1, ρ_2 are the electron densities corresponding to atom 1 and 2 respectively. The first two terms correspond to the nonlocal interaction of the densities of

the individual atoms with themselves, while the last term corresponds to the interaction of the density on atom 1 with the density on atom 2. One is usually interested in the interaction energy which is defined as the total energy of the dimer minus the sum of the total energies of the individual fragments. The total nonlocal energy of the individual non-interacting fragments is

$$\tfrac{1}{2} \iint \rho_1' \Phi \rho_1' + \tfrac{1}{2} \iint \rho_2' \Phi \rho_2' \tag{4.2}$$

where ρ_1', ρ_2' are the electron densities of the noninteracting fragments. Note that while there is almost no density overlap of the fragments in case of dispersion interaction, there is still some depletion of the density from the area between the fragments due to the Pauli principle. Therefore, ρ_i and ρ_i' are very similar but not identical.

The interaction energy thus consists of two principally different parts,

$$\iint \rho_1 \Phi \rho_2 + \iint (\rho_1\rho_1 + \rho_2\rho_2 - \rho_1'\rho_1' - \rho_2'\rho_2')\Phi \tag{4.3}$$

In the case of non-overlapping fragments, the second term is zero and only the first term is relevant. Our tests showed that this holds approximately even for real molecules. On the S22 database set [34], for example, the second term accounts for several percent only. Therefore, we consider only the first term in (4.3) in the following. This leads to smaller computational cost and also easier interpretation of the nonlocal correlation energy.

Figure 4.1 shows the nonlocal energy density of $\iint \rho_1 \Phi \rho_2$, that is energy density on atom 1 generated by electron density on atom 2. The shell structure of the argon atom can be clearly seen. The energy density and even the energy contributions are greatest for the valence shell. This corresponds to the fact that the electrons of the valence shell are most easily polarizable and thus most prone to correlation with the electrons of the other atom. This figure also illustrates that the dispersion interaction is well "localized" on atoms, which justifies all atom-pair dispersion correction methods.

The nonlocal correlation functional is combined with LDA correlation and it is claimed that the gradient correction usually contained within a GGA functional is covered by the nonlocal part. Figure 4.2 puts this assertion into test by comparing the nonlocal correlation energy density of one single atom ($\iint \rho_1 \Phi \rho_1$) with a gradient correction from PBE. While the shell structure is clearly represented

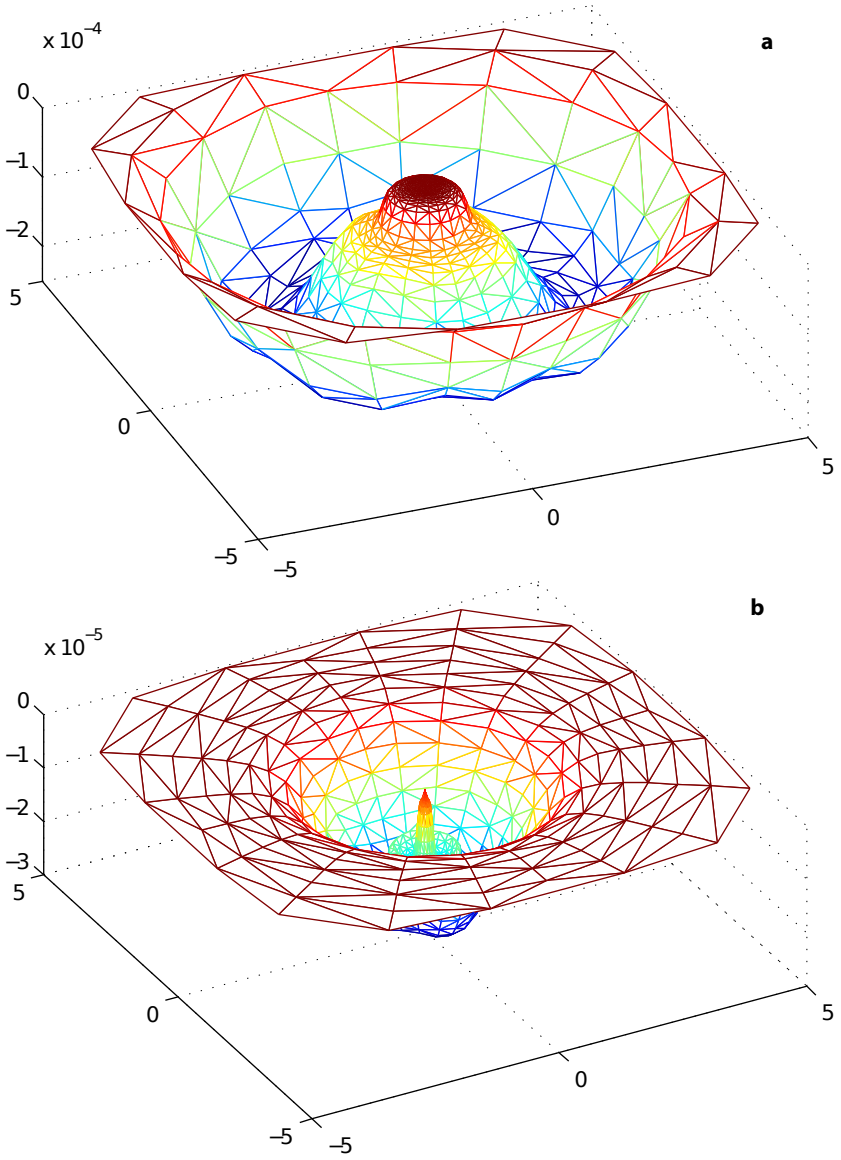

Figure 4.1 | Nonlocal interaction energy. Nonlocal energy of an argon atom generated by the electron density of another argon atom at the distance of 3.5 Å. **a**, The nonlocal energy density $\varepsilon_1^{\mathrm{nl}}(\mathbf{r}_1) = \int \Phi(\mathbf{r}_1, \mathbf{r}_2)\rho_2(\mathbf{r}_2)\,\mathrm{d}\mathbf{r}_2$. **b**, The nonlocal energy contribution $\rho_1(\mathbf{r})\varepsilon_1^{\mathrm{nl}}(\mathbf{r})$.

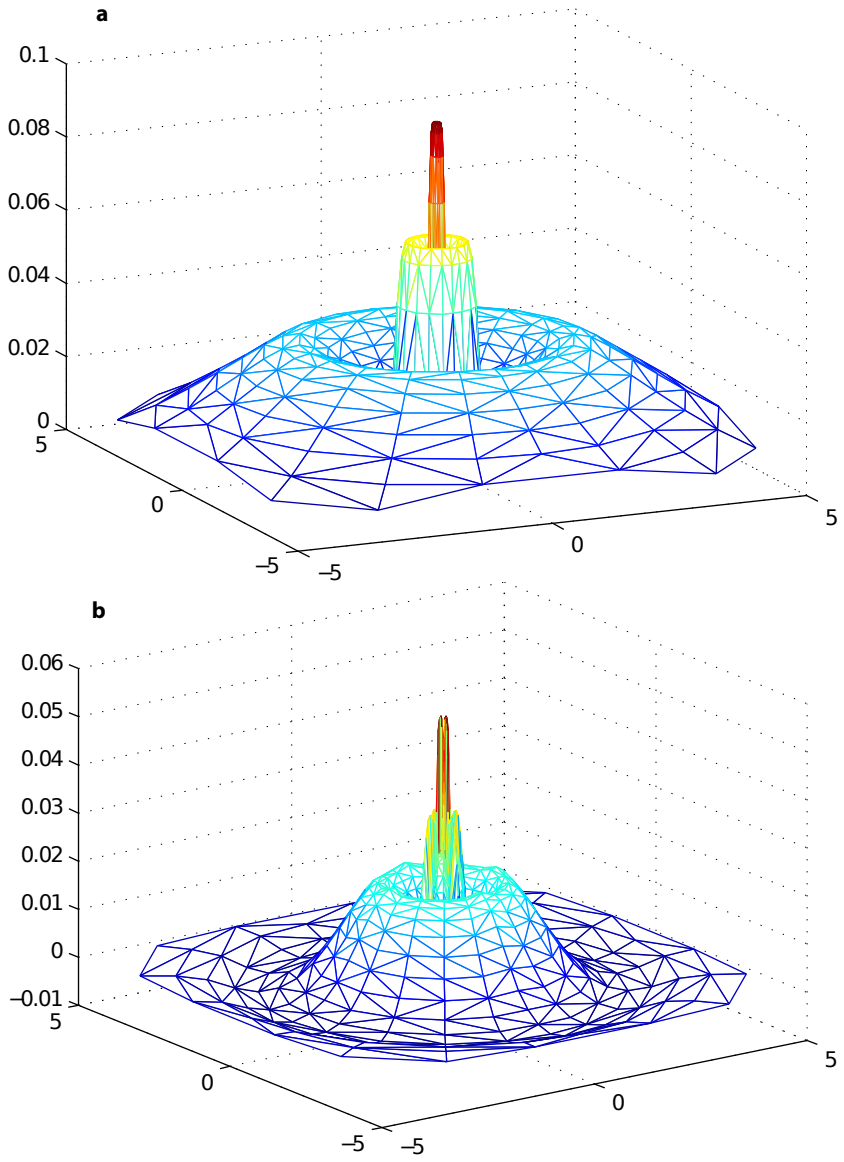

Figure 4.2 | Nonlocal self-energy of an atom. Nonlocal energy of an argon atom generated by its own density. **a**, The nonlocal part of the semi-local PBE correlation energy density $\varepsilon_c^{PBE} - \varepsilon_c^{LDA}$. **b**, The nonlocal energy density $\varepsilon_1^{nl}(\mathbf{r}_1) = \int \Phi(\mathbf{r}_1, \mathbf{r}_2)\rho_1(\mathbf{r}_2)\, d\mathbf{r}_2$.

in both in the same manner, there are some quantitative differences. The PBE gradient correction is positive everywhere. The vdW-DF energy density goes more quickly to zero with the distance from the nucleus, it is generally smaller and it is negative in the density tail areas. The tail areas are far enough from the core of the atom to be in the attractive region of the nonlocal kernel Φ. This figure shows that while part of the semi-local gradient correction might be recovered by the nonlocal functional, some loss of accuracy has to be expected.

This section presented several visualizations of the vdW-DF nonlocal energy. It was shown that nonlocal energy is well "localized" on atoms and that its main contribution comes from the valence electrons. A comparison between the nonlocal energy and the gradient correction of the PBE correlation functional was also presented, illustrating that they share some qualitative aspects but differ quantitatively.

4.2 Excited states with vdW-DF

The wavefunction-based methods which accurately account for dispersion are practically limited by the size of the studied system. This is true for the study of electronic ground state and the limitations are even more severe for the excited states. An equivalent of the HF method for excited states is the CASSCF[1] method. Here, the trial wavefunction consists of more than one Slater determinants, generated by all possible electronic excitations within a specified set of occupied and virtual orbitals (active space). In theory, if the active space contained all the orbitals, CASSCF would give the exact energy (within the basis set), but due to prohibitive computational cost, one usually uses only several orbitals. Therefore, in practical calculations, CASSCF does not account for dispersion, which is computationally covered only by excitations to higher virtual orbitals not present in a typical active space. CASPT2[2] is a method for excited states, which gives results similar to MP2 for systems with single-configuration character, but is able to treat also multi-configuration systems.

Excimers are dimers which are only weakly or not at all bonded in the ground state, but are bonded when one of the monomers is in an excited state. One of the main contributions to the bonding energy are often vdW interactions. There are many excimers which can be calculated by CASSCF, but are too big for CASPT2,

[1] Complete-active-space self-consistent field.
[2] Complete-active-space perturbation theory to second order.

but as CASSCF does not contain dispersion, the interaction energies are often well underestimated.

Here, we illustrate a possible approximate solution to this problem by adding vdW-DF nonlocal energy to CASSCF energies. The reason behind this approach is that the total interaction energy of vdW systems consists mainly of exchange repulsion (exact in CASSCF) and dispersion (added by vdW-DF). The CASSCF+vdW-DF combination should thus describe the main components of the interaction. The original version of vdW-DF is used because it is compatible with the revPBE exchange, which is fitted to the HF exchange, which in turn is equivalent to the CASSCF exchange.

We consider the case of the benzene excimer, which is one of the most studied excimers both theoretically and experimentally [35]. The three lowest lying excited states of the benzene molecule correspond to excitations of electrons from the occupied π-orbitals with one nodal plane (E_{1g} symmetry) to the virtual π-orbitals with two nodal planes (E_{2u}). The transition from E_{1g} to E_{2u} leads to the excited states of symmetries B_{1u}, B_{2u} and E_{1u}. In case of the dimer, its lowest excited states are formed by (anti)symmetrized combinations of one benzene molecule in the ground state and the other one in an excited states. The combined wavefunction can be either symmetric or antisymmetric with respect to inversion (equivalent to bonding and nonbonding orbitals in a hydrogen molecule), resulting in two excited states of the dimer per one excited state of the monomer. From the three lowest lying excited state of the monomer, only B_{2u} and E_{1u} lead to observable transitions, resulting in four excited states which are subject of our study.

All calculations were performed in the Molpro package [36], using the ANO basis set with 4s3p2d basis set functions on carbon atoms and 3s2p on hydrogen atoms. The CASPT2 calculations were carried out with the standard IPEA modification. The active space comprised the six π-orbitals of both benzene molecules, resulting in 12 electrons in 12 orbitals. The basis set superposition error (BSSE) was taken into account by using the counterpoise correction (CP) in the form as presented in Ref. 35. The vdW-DF nonlocal energy was calculated from the CASSCF densities.

Figure 4.3 shows results for the benzene excimer. The addition of nonlocal correlation energy improves the CASSCF interaction curves in all cases. The CASSCF curve for the ground state is purely repulsive due to the missing dispersion. In contrast, the CASSCF+vdW-DF and CASPT2 curves nearly coincide. The agreement is not that perfect for excited states but still promising. For the

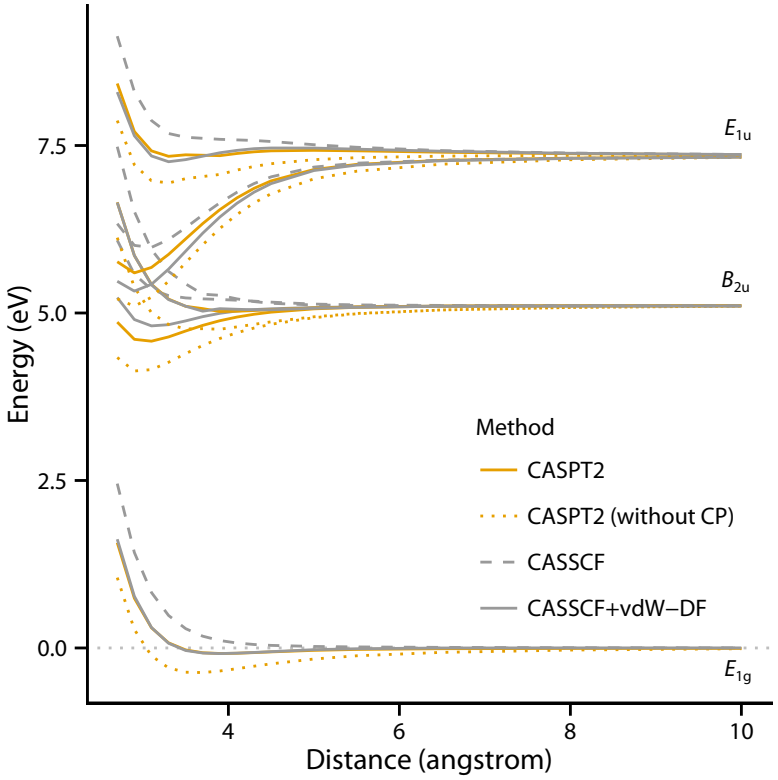

Figure 4.3 | Benzene excimer. Interaction curves of the ground state and lowest lying excited states of benzene dimer. CP stands for counterpoise correction. For comparison, a constant is added to the CASSCF curves so that they have the same dissociation limit as CASPT2 curves. The lower excited state corresponds to the B_{2u} monomer excitation, the higher to the E_{1u} excitation. The CASSCF energies without CP correction are not presented, because BSSE is almost negligible in case of CASSCF.

non-bonding (antisymmetric) states, the vdW-DF correction works very well. The worst case are the bonding (symmetric) states, the interaction in the B_{2u} and E_{1u} case being underestimated and overestimated, respectively, by several tenths of eV. But still, the CASSCF+vdW-DF energies are much better than pure CASSCF energies. On a different note, the BSSE in case of CASPT2 is clearly non-negligible, being of the same magnitude as the dispersion from vdW-DF.

We have also studied the dependence of the dispersion energy on the electronic state of a molecule. For this purpose, we used the benzene–argon system instead of the benzene dimer to have only one fragment which can undergo electronic transitions (the excited states of an argon atom lie much higher than those of a

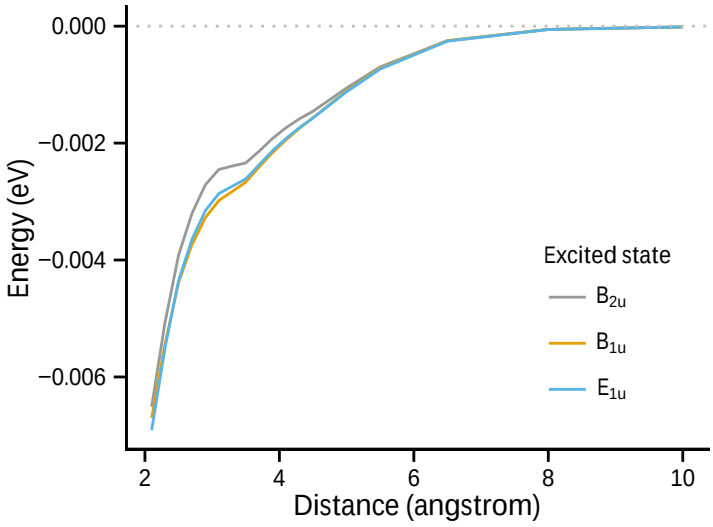

Figure 4.4 | Dispersion in excited states. The difference between the nonlocal energy of a benzene–argon system calculated from the ground state density and from the excited state density.

benzene molecule). Figure 4.4 shows that the change in dispersion interaction coming from the change in electron density due to electronic excitation is insignificant, being less than one hundredth of eV even for short interfragment distances. The difference is even smaller between individual excited states. This independence of dispersion on the electronic state can be understood solely in terms of the number of electrons. Benzene molecule has 30 valence electrons, and the transition of one of them can affect several percent of dispersion at most. Moreover, the transition is of the $\pi \rightarrow \pi$ type so even the electron density coming from the excited electron does not change much. Finally, the transitions are between the same orbitals, differing mainly in symmetry, resulting in the almost identical dispersion for the individual excited states.

This section introduced a potential way how to correct CASSCF interaction energies of excimers for dispersion by simply adding the vdW-DF nonlocal energy. It was shown that this scheme works well for the ground state and nonbonding excited states of the benzene excimer, and it improves also the bonding states albeit less accurately. This discrepancy should be further investigated if this approaches was to be turned into a real method. Further, it was shown here that the nonlocal energy depends on the electronic states only negligibly.

4.3 Optimizing the Z_{ab} parameter

The Z_{ab} parameter in the vdW-DF nonlocal functional controls screening of the electrons and the more negative it is, the weaker the dispersion interaction. In the original version of vdW-DF, Z_{ab} is derived from gradient expansion of density in the slowly varying electron gas. In the second version, vdW-DF2, Z_{ab} is taken from the large-N asymptote, that is the asymptote of hypothetical atoms with N electrons where N goes to infinity. The large-N asymptote has been used in the derivation of some semi-local functionals and it is probably closer to the reality of molecules than the slowly varying electron gas. However, it is clear that neither of these two approaches is exactly similar to real molecules. The idea is thus to optimize the Z_{ab} parameter with respect to some reference benchmark energies.

4.3.1 S22 set

The S22 database set contains seven hydrogen-bonded complexes, eight predominantly dispersion-bonded complexes, and seven mixed complexes [34]. It has become a standard for benchmarking methods intended for vdW interactions [37, 38]. We have used the reference S22 energies to optimize the Z_{ab} parameter in vdW-DF. Taking into account that vdW-DF and vdW-DF2 use also different exchange functionals, we have examined various semi-local functionals to be combined with the vdW-DF nonlocal functionals. Inspired by the VV10 nonlocal functional, we have not limited the search for combinations with LDA correlation, but also considered semi-local correlation functionals. We tested 10 combinations of PBE, revPBE, B88, B86 and PW86 exchange functionals and of LDA, LYP, and PBE correlation functionals.

The optimized values of Z_{ab} are presented in Table 4.1. The best mean absolute percentage error (MAPE) of 6.3% is obtained with the revPBE functional (revPBE exchange and PBE correlation). The semi-local functionals of vdW-DF and vdW-DF2 perform worse, with MAPE of 14.5% and 8.4% and optimized Z_{ab} values of −0.94 (−0.8491 in vdW-DF) and −1.98 (−1.887 in vdW-DF2), respectively. This shows that the Z_{ab} values in the original methods were already quite optimal with respect to the S22 set, but they could be improved. Also, it can be seen that the improvement of vdW-DF2 over vdW-DF stems mainly from the hydrogen-bonded complexes, which corresponds to earlier findings. Two best performing functionals have semi-local correlation, which suggests that while the argument

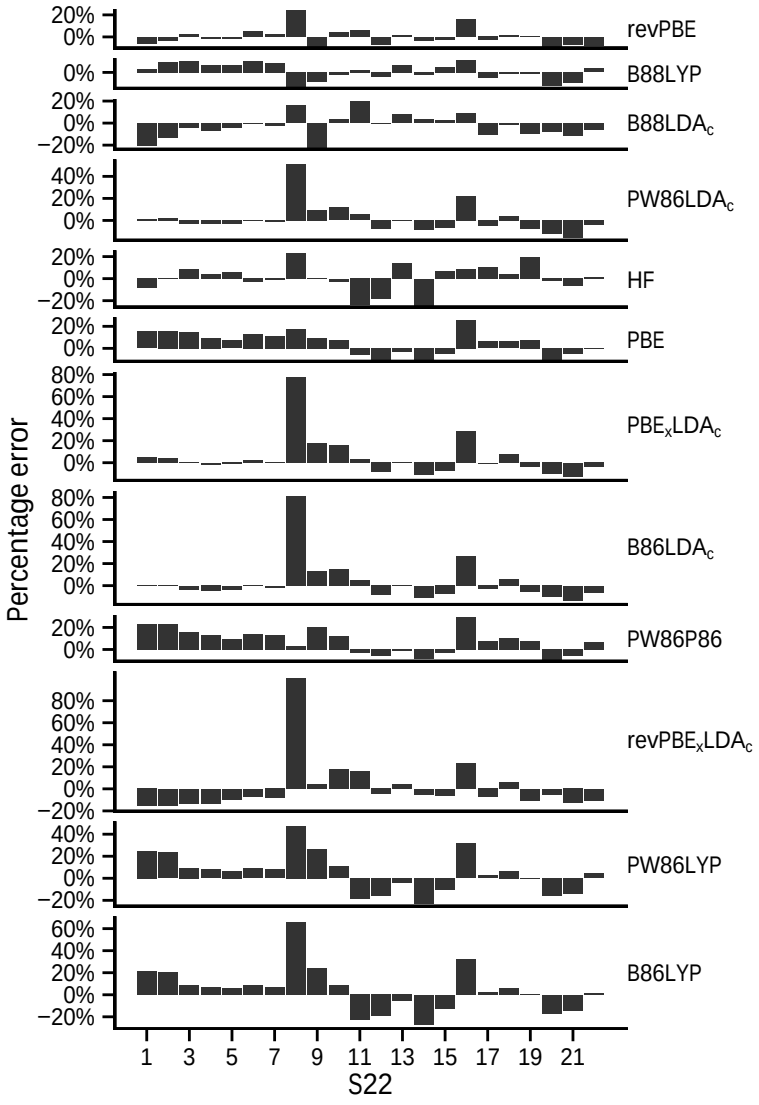

Figure 4.5 | Optimizing Z_{ab} on S22 set. The percentage errors of interaction energies of individual complexes (*x*-axis) from the S22 set as a function of the used semi-local XC functional. Complexes 1–7 are hydrogen-bonded, 8–14 are dispersion-bonded and 15-22 are mixed. The HF method is included for comparison.

Table 4.1 | Values of Z_{ab} for different semi-local functionals optimized on the S22 set.

XC functional	Z_{ab}	MAPE[a]
revPBE	−1.75	6.3%
B88LYP	−1.81	7.2%
B88LDA$_c$	−0.83	8.0%
PW86LDA$_c$[b]	−1.98	8.4%
	−1.887[c]	8.8%
HF[d]	−1.88	9.0%
PBE	−3.79	9.6%
PBE$_x$LDA$_c$	−2.10	10.1%
B86LDA$_c$	−1.77	10.4%
revPBE$_x$LDA$_c$[e]	−0.94	14.5%
	−0.8491[f]	15.2%
PW86LYP	−4.21	14.5%
B86LYP	−3.77	15.3%

[a] The mean absolute percentage error of the interaction energies as calculated with the specified semi-local XC functional and vdW-DF nonlocal functional with given Z_{ab} with respect to the reference S22 energies. MAPE is used as the target quantity in the optimization. [b] The functional used in vdW-DF2. [c] The value of Z_{ab} in the original vdW-DF2 method. [d] The Hartree–Fock method is included only for comparison. [e] The functional used in vdW-DF. [f] The value of Z_{ab} in the original vdW-DF method.

against the use of semi-local correlation (due to potential double counting) might be theoretically justifiable, it does not seem useful in the light of numerical results. Notable is the good performance of the HF method, which suggests that the local correlation is not that important in case of interaction energies of vdW complexes. The final note is on the Z_{ab} values of two worst performing functionals, which are lesser than −3. Such a low Z_{ab} value is unreasonable and suggests that the underlying semi-local functionals are a bad approximation for E_{xc}^0. In contrast, the Z_{ab} values for the top five functionals are all roughly in the range marked by vdW-DF and vdW-DF2 values.

Figure 4.5 sheds further light on the performance of the individual semi-local functionals. Different behaviour for the three groups of the S22 set can be seen, especially for the hydrogen-bonded complexes as compared to the other two groups. The PW86LDA$_c$ functional of vdW-DF2 or even B86LDA$_c$ perform perfectly for the hydrogen-bonded complexes, but lack in the other two groups when compared to the best performing functionals. With the exception of revPBE$_x$LDA$_c$, it can be said that the best functionals for hydrogen-bonded complexes are those

with LDA correlation. The most problematic complex for most of the functionals is number 8, the methane dimer. This can be easily understood as the interaction energy of this complex is lowest in the S22 set (0.5 kcal/mol), making the percentage error large. Indeed, the smallest systems are often the most stringent test for DFT methods, this case being no exception.

This section presented a systematic study of the interplay between choosing the right semi-local functional for approximating E_{xc}^{0} and the value of the Z_{ab} parameter. The semi-local functionals of vdW-DF and vdW-DF2 are included and while the improvement of the second version is apparent, we showed that better choice can be made, at least based on the results on the S22 set. However, the improvement made is only by 25%, and other approaches have to be searched for to improve the accuracy of vdW-DF more significantly.

4.3.2 Noble gas dimers

Dimers of noble gas atoms are prototypical dispersion-bonded systems. Being symmetrical, the electrostatic and induction forces are not present and the attractive forces come from dispersion only. Therefore, they are often used in investigating and testing dispersion methods.

We have used the noble gas dimers to optimize the Z_{ab} values for the PW86LDA$_c$ functional (from vdW-DF2) against CCSD(T)/CBS[3] energies. In contrast to the case of the S22 set however, a distinct Z_{ab} value for each individual dimer and interatomic distance was obtained. This investigation could lead to parametrization of Z_{ab} as a function of the atom type.

Figure 4.6 shows the optimized Z_{ab} values. For all dimers, the optimal Z_{ab} value goes from small negative values to large negative values, passes through a minimum, and then approaches zero with growing distance. All the curves have a similar shape, but they differ quantitatively. The biggest outliers are the helium, neon and He−Ne dimer, that is the smallest dimers.

Figure 4.7 shows the dependence of the nonlocal energy at individual points on the interaction curve. The dependence is much stronger for shorter distance, which is simply due to the dispersion interaction itself being stronger. A more interesting observation is that the shape of the dependence is also different. While

[3] Coupled clusters method with single, double and perturbative triple excitations at the complete basis set limit. We have used a standard method for the basis set extrapolation, CCSD(T)/CBS = CCSD(T)/AVDZ + MP2/CBS − MP2/AVDZ, MP2/AVnZ = MP2/CBS + $A \times n^{-3}$. All energies were calculated in Molpro [36].

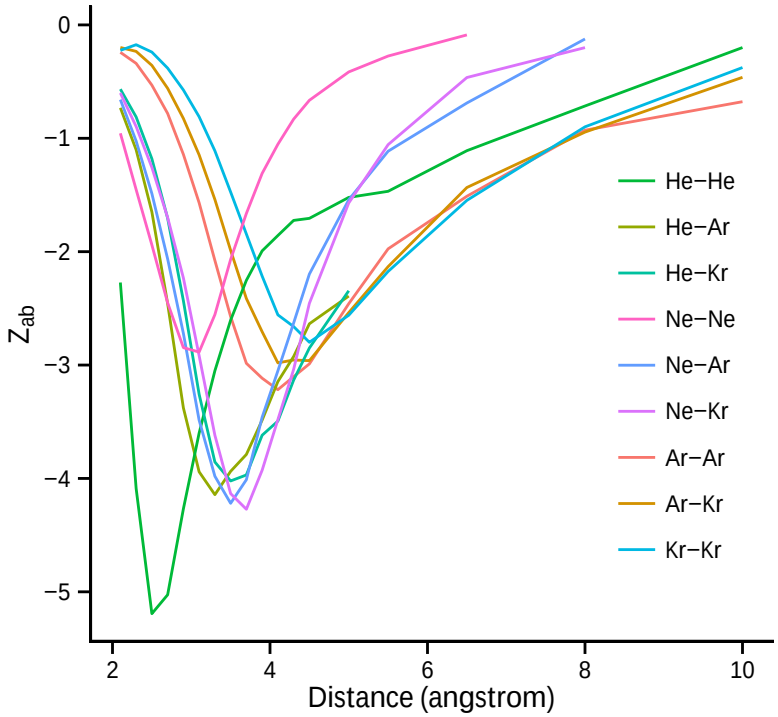

Figure 4.6 | Z_{ab} values for noble gas dimers. Optimal Z_{ab} for PW86LDA$_c$ as a function of the noble gas dimer and interatomic distance. The He−Ne dimer is not present because its optimal Z_{ab} is too negative for a large part of the interatomic distance and not enough points have been obtained for it.

at short distances, the nonlocal energy changes significantly even for very negative values of Z_{ab}, it is nearly constant and changes only for Z_{ab} values close to zero at longer distances. This would suggest to ignore the tail part of the interaction curves when searching for optimal Z_{ab}. However, this approach would invalidate any calculations where asymptotic interactions are important, such as in bulk matter or molecular crystals.

This section illustrated that the compatibility between the semi-local and non-local functional is far from perfect. If it was, the Z_{ab} parameter would be independent of the interatomic distance, because it is present in the *local* quantity k_0 from (2.31). Thus Z_{ab} determines the behaviour of the local density, not of the actual nonlocal interaction. Therefore, it cannot be made dependent on the distance, at least not without seriously violating the spirit of vdW-DF. In light of these consideration, it seems that parametrizing Z_{ab} as a function of the atom

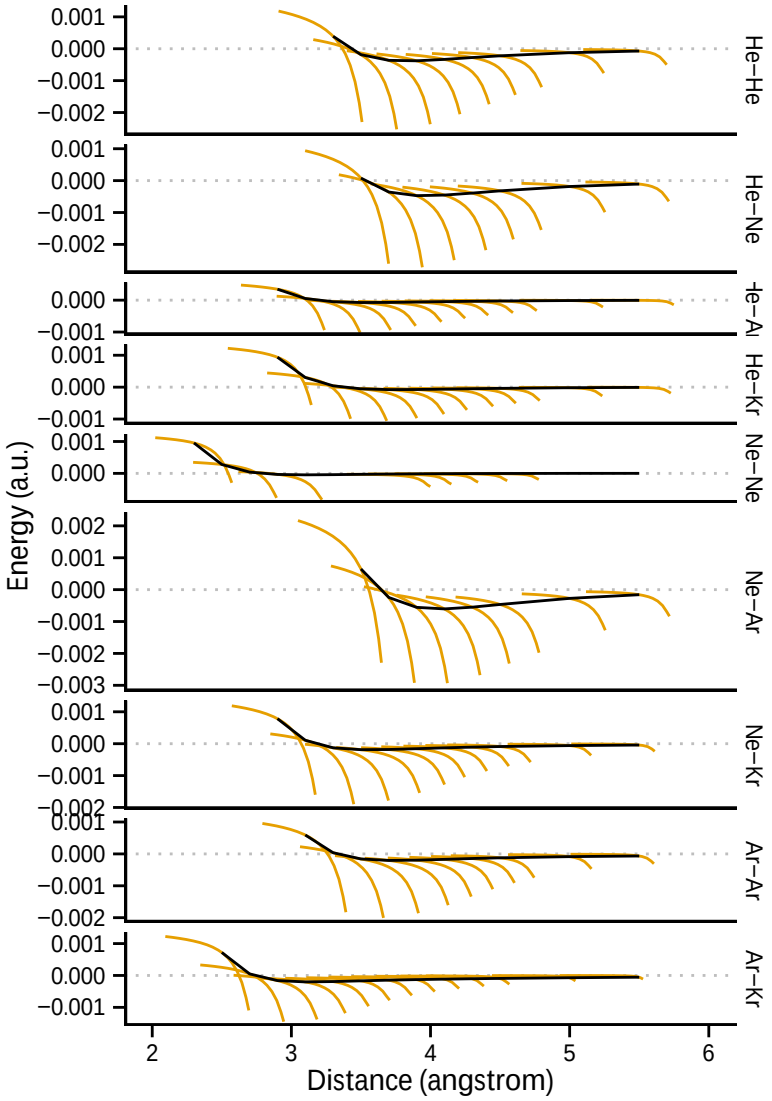

Figure 4.7 | Optimizing Z_{ab} on noble gas dimers. Interaction curves (black) of noble gas dimers obtained from CCSD(T)/CBS. The yellow curves (without displayed axes) denote the dependence of the nonlocal energy on $Z_{ab} \in (-6, 0)$ for interatomic distance at which they intersect the interaction curve. The intersection also denotes the Z_{ab} value with which PW86LDA$_c$+vdW-DF exactly reproduces the CCSD(T)/CBS energy.

type would be superfluous as the underlying mechanism is flawed on a deeper level of the theory.

4.4 Concluding remarks

This chapter presented three studies of the behaviour of vdW-DF, namely (i) the visualization of the vdW-DF nonlocal energy, (ii) a potential use of vdW-DF for treating excimers and (iii) optimizing the Z_{ab} parameter together with selecting the best semi-local functional for vdW-DF. It is apparent from (i) that the nonlocal interaction is mainly between the individual atoms, and that it can be decomposed into a pseudo-sum over atom pairs. At the same time, the results in (iii) showed that optimizing the Z_{ab} parameter is not as promising as previously expect, and that different approaches have to be sought. Progressively, the combination of these two results lead to the development of the correction scheme presented in the next chapter.

Chapter 5

vdW-DF/CC correction scheme[1]

One of the most studied processes where van der Waals interactions, or dispersion forces, play a major role is the physisorption of molecules on various external or internal surfaces of materials [39–41]. DFT is a major tool of computational chemistry for studying such systems (surfaces or crystals). As mentioned earlier, however, traditional local and semi-local DFT functionals either do not cover dispersion at all or simulate it unreliably with non-physical binding by exchange energy [42]. Significant efforts have been made to reliably include dispersion into DFT, and many different methods have been suggested, implemented and tested, ranging from empirical to purely ab-initio approaches [43].

This chapter presents a method for the prediction of the binding energies of physisorption with subchemical accuracy (~ 0.1 kcal/mol) [37]. Such accuracy is necessary for a qualitative comparison of computational results with some experiments, e. g. measurements of adsorption isotherms [44]. Two state-of-the-art generally applicable DFT dispersion methods, DFT-D3 of Grimme [45] and VV10 of Vydrov and Van Voorhis [22], reach the accuracy of several tenths of kcal/mol on the S22 set [34], on which they are partly parametrized, and of about 1 kcal/mol on the noncovalent subset of GMTKN30 [46], which is a predominantly independent benchmark for both [47, 48]. However sufficient for general use, they are not accurate enough for the specific purposes mentioned above.

The method presented here is based on vdW-DF and introduces an empirical scaling derived from accurate CCSD(T) calculations on small models of the adsorption system. The price for this empiricism is system-specificity, but its benefit

[1] This chapter closely follows the paper "Hermann, J. & Bludský, O. A novel correction scheme for DFT: A combined vdW-DF/CCSD(T) approach" submitted to the Journal of Chemical Physics.

is that the method meets the strict criteria for accuracy needed for comparison with adsorption experiments.

5.1 DFT/CC and vdW-DF/CC

The vdW-DF method introduced and investigated in previous chapters is used as a starting point for our correction scheme. vdW-DF has been shown to provide binding energies in good agreement with benchmark methods or experiment [3]. At the same time, it is non-empirical and there is a strong assumption that it is unbiased to any specific type of chemical systems.

The empirical part of our method, which brings in the required accuracy, draws upon the ideas of DFT/CC, an empirical scheme for correcting DFT [49]. Here, the difference ΔE between the accurate CCSD(T) and local DFT interaction energies of two fragments A, B is approximated by a set of correction curves ε_{XY},

$$\Delta E = \sum_{i\in A}\sum_{j\in B} \varepsilon_{T(i)T(j)}(R_{ij}) \tag{5.1}$$

where the summation is over the pairs of atoms i, j, $T(k)$ is the atom type (H, C,…) of atom k, and R_{ij} is the interatomic distance. To parametrize ε_{XY}, one has to devise a series of models where each model serves for parametrizing a new correction curve while all the others have to be known from previous models. For example, for parametrizing $\varepsilon_{\mathrm{HH}}$, $\varepsilon_{\mathrm{CH}}$, $\varepsilon_{\mathrm{CC}}$, one uses $H_2\cdots H_2$, $Bz\cdots H_2$, $Bz\cdots Bz$, respectively [49]. The individual curves are obtained by RKHS interpolation as described in Ref. 49. This approach can provide superb accuracy (< 0.1 kcal/mol), but the parametrization procedure has some deficiencies. It can be tedious in cases with many atom types, and it can also be numerically unstable, e. g. when there are regularly alternating atoms of different types, such as Si and O in silica-based materials.

Our new correction scheme, dubbed vdW-DF/CC, overcomes these two obstacles. It uses only one correction curve ε and the weight functions w_{ij}, obtained from vdW-DF, to differentiate between atom types. The correction energy is thus approximated as

$$\Delta E = \sum_{i\in A}\sum_{j\in B} \varepsilon(R_{ij}) w_{T(i)T(j)}(R_{ij}) \tag{5.2}$$

This scheme needs only a single cluster model to be parametrized for any given

adsorbent–adsorbate system. Moreover, any numerical instabilities of DFT/CC are eradicated, as the distinction between atom types is now made outside of the correction scheme on the vdW-DF level.

It was shown in Chapter 4 that the vdW-DF nonlocal energy is localized on pairs of atoms and that it can be naturally expressed as a sum over the pairs in our implementation,

$$E^{\mathrm{nlc}} = \sum_{i \in A} \sum_{j \in B} E_{ij}^{\mathrm{nlc}}(R_{ij}) \tag{5.3}$$

To turn from E_{ij}^{nlc} for individual atom pairs (ij) to $\epsilon_{XY}(r)$ for atom-type pairs (XY), the RKHS interpolation is used again such that

$$\sum_{i \in X} \sum_{j \in Y} E_{ij}^{\mathrm{nlc}}(R_{ij}) = \sum_{i \in X} \sum_{j \in Y} \epsilon_{XY}(R_{ij}) \tag{5.4}$$

Preliminary tests have shown that at least on a class of similar molecular systems, this representation of vdW-DF by pair curves is well transferable and that the error associated with this approximate representation is significantly smaller than the error of the resulting correction method. Thus, we can evaluate the vdW-DF energy approximately as a simple atom-pair curve method.

The dimensionless weights w_{XY} are then given as

$$w_{XY} = \frac{\epsilon_{XY}}{\epsilon_{X'Y'}} \tag{5.5}$$

where $\epsilon_{X'Y'}$ is an arbitrary reference curve.

Our method can also be interpreted in a different way than as a modified DFT/CC method. After rewriting Eq. 5.2 as

$$\Delta E = \sum_{i \in A} \sum_{j \in B} \frac{\varepsilon(R_{ij})}{\epsilon_{X'Y'}(R_{ij})} \epsilon_{T(i)T(j)}(R_{ij}) \tag{5.6}$$

vdW-DF/CC can be seen as a scaling of vdW-DF by $\varepsilon/\epsilon_{X'Y'}$. Thus, our scheme is a merge of two worlds in a sense. It can be interpreted either as an empirical pair correction curves weighed by non-empirical method, or as a non-empirical nonlocal functional scaled by empirical factors.

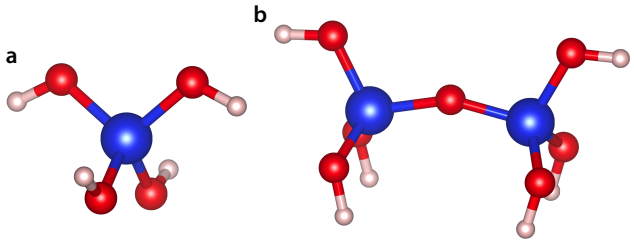

Figure 5.1 | Models of silica. a, The 1T model ($Si(OH)_4$) is the simplest relevant model of silica. The O atoms are terminated by H atoms. **b**, The 2T model ($Si(OH)_3OSi(OH)_3$) is the simplest model of silica containing the Si–O–Si bridge, which is characteristic of silica-based surfaces.

5.2 1T and 2T silica models

We have used our method to calculate the interaction energies of five small molecules (CH_4, CO_2, H_2, H_2O, N_2) with two different silica-based surfaces. Figure 5.1a shows the 1T model of silica utilized to parametrize the correction curves ε and to obtain the vdW-DF curves ϵ_{XY}. Figure 5.1b shows the 2T model used to verify the transferability of the parametrization. Figure 5.2 depicts all the relative orientations of the nT models and molecules used in our study. Geometries with C_{2v} have been used to facilitate the costly CCSD(T) calculations. The most strongly binding orientation of each molecule on the 1T model was used for parametrization. Figures 5.3 and 5.4 show the obtained ϵ_{XY} and ε, respectively. The correction curves ε have all similar shape but they are displaced along the x-axis. On the contrary, a hypothetical correction curve $\varepsilon = \epsilon_{X'Y'}$ which would reproduce the vdW-DF2 energies exactly has a different shape. This suggests that there is a systemic error and a universal correction curve for all molecules could be devised which would still provide better energies than vdW-DF2.

The numerical work uses the vdW-DF2 flavor (PW86 exchange functional, $Z_{ab} = 1.887$) as it provides better interaction energies than the original vdW-DF [18]. The CCSD(T)/CBS energies and DFT electron densities used for the evaluation of vdW-DF were obtained by MOLPRO [36]. The standard procedure for obtaining the CBS limit was used.[1] The PW86+LDA(c)/AVQZ energies were obtained by Gaussian 09 [50]. The geometries and energies of extended systems were obtained by VASP using soft and hard PAW pseudopotentials, respectively [51].

[1] CCSD(T)/CBS = CCSD(T)/AVDZ + MP2/CBS – MP2/AVDZ, MP2/AVnZ = MP2/CBS + $A \times n^{-3}$

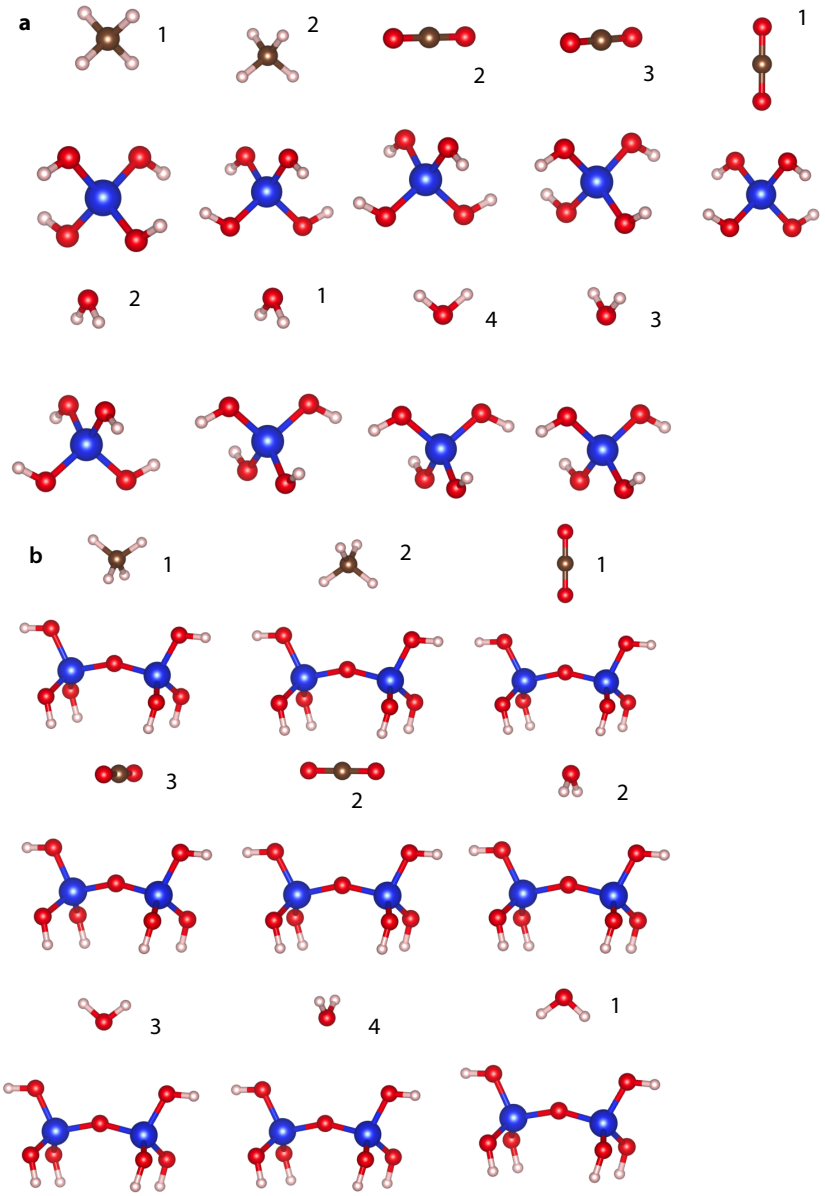

Figure 5.2 | Geometries of model systems. Geometries of model systems with the 1T (**a**) and 2T (**b**) models. The numbers denote the geometry index used throughout this chapter. The geometries of models with H_2 and N_2 are defined equivalently to CO_2.

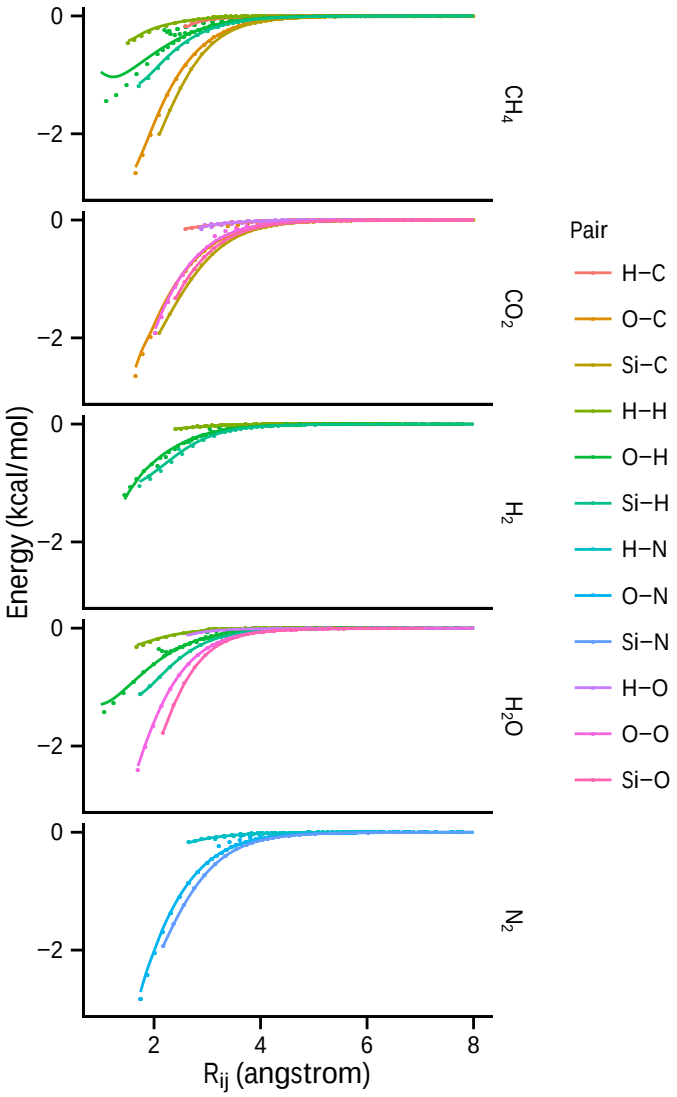

Figure 5.3 | vdW-DF pair curves. The points represent individual vdW-DF atom-pair contributions from all interfragment distances of the most bonding geometry of each molecule with the 1T model as a function of the interatomic distance R_{ij}. The solid curves are obtained from the points by RKHS interpolation.

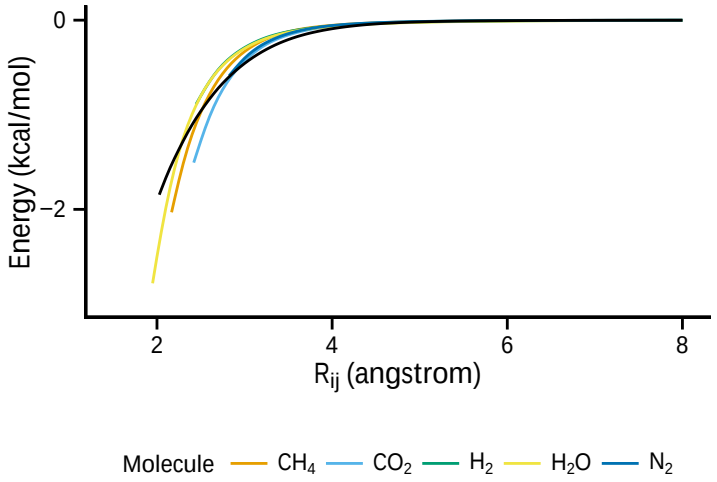

Figure 5.4 | vdW-DF/CC correction curves. Correction curves parametrized on the 1T model as a function of the interatomic distance R_{ij}. The black curve is a hypothetical correction curve that would reproduce vdW-DF2 interaction energies if used.

Figures 5.5 and 5.6 show the calculated interaction curves of the molecules with the 1T and 2T model, respectively. vdW-DF systematically overbinds in all cases. When going from the 1T model, on which our scheme is parametrized, to the 2T model, where it is tested, its accuracy deteriorates somewhat, but it is still several times better than vdW-DF2. The order of binding energies of the individual geometries often differs between the 1T and 2T model. This illustrates the difference between 1T and 2T. The molecules interact predominantly with the terminal O atoms in case of the 1T model, whereas the dominant part of the interaction with the 2T model is with the Si–O–Si bridge. It is notable that vdW-DF/CC works best for the most binding geometries of the 2T model systems even though they are different from the most binding 1T geometries.

Table 5.1 presents the binding energies of the 2T model systems. The root-mean-square error of our method is 0.06 kcal/mol (all molecules and all bonding orientations) while it is 0.45 kcal/mol for vdW-DF2. The mean absolute percentage error is 5% for our method and 51% for vdW-DF2. This shows that our method provides roughly an order of magnitude better binding energies for the systems for which it has been parametrized than vdW-DF2.

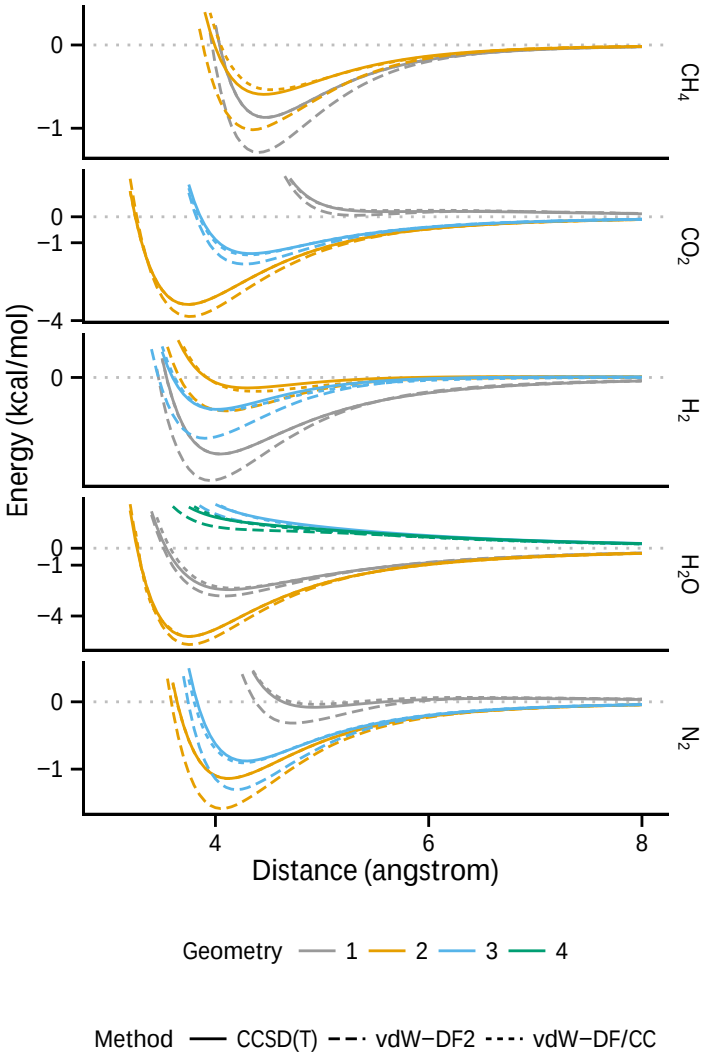

Figure 5.5 | Interaction curves of the 1T model. Geometries correspond to different orientation of the fragments defined in Figure 5.2. vdW-DF/CC is our method parametrized on the most bonding geometry of the 1T model, hence it is coincident with the CCSD(T) curve for that geometry.

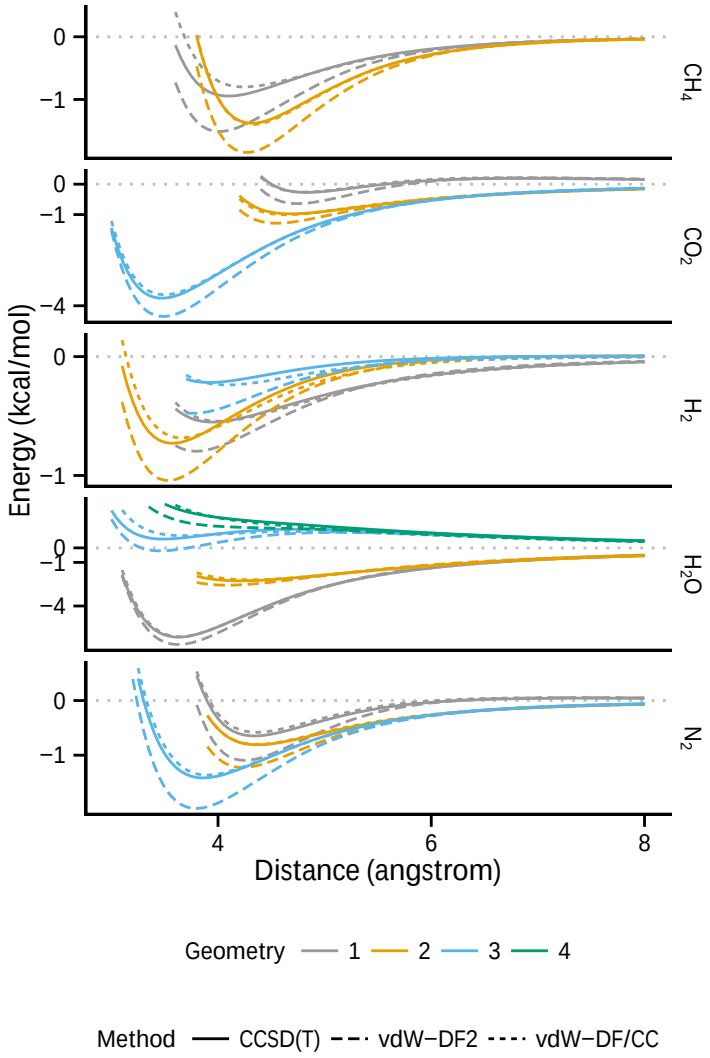

Figure 5.6 | Interaction curves of the 2T model. Geometries correspond to different orientation of the fragments defined in Figure 5.2. vdW-DF/CC is our method parametrized on the most bonding geometry of the 1T model.

Table 5.1 | Binding energies (kcal/mol) of the 2T model systems by different methods.

Molecule	Geometry[a]	vdW-DF2	vdW-DF/CC	CCSD(T)	ΔvdW-DF/CC[b]	ΔCCSD(T)[c]
CH_4	1	-1.5	-0.8	-0.9	0.7	-0.15
	2	-1.8	-1.4	-1.4	0.4	0.02
CO_2	1	-0.6	-0.3	-0.3	0.4	-0.01
	2	-1.3	-1.0	-1.0	0.3	0.03
	3	-4.4	-3.6	-3.8	0.7	-0.12
H_2	1	-0.8	-0.5	-0.6	0.3	-0.01
	2	-1.0	-0.7	-0.7	0.4	-0.05
	3	-0.5	-0.2	-0.2	0.2	0.02
H_2O	1	-6.7	-6.1	-6.1	0.5	-0.03
	2	-2.6	-2.2	-2.3	0.4	-0.07
	3			non-bonding		
	4			non-bonding		
N_2	1	-1.1	-0.6	-0.6	0.5	-0.07
	2	-1.2	-0.8	-0.8	0.4	-0.01
	3	-2.0	-1.4	-1.4	0.6	-0.05

[a] The geometry index defined in Figure 5.2. [b] The difference vdW-DF/CC − vdW-DF2.
[c] The difference CCSD(T) − vdW-DF/CC.

5.3 Tests on quartz surface and UTL lamella

We have tested the 1T model parametrization of vdW-DF/CC on two silica surfaces, namely the quartz surface and the UTL lamella. The perfectly reconstructed α-quartz surface is obtained by cutting the bulk quartz crystal and rearranging the surface atoms such that they fully saturate their covalent bonding [52]. Figure 5.7 depicts the resulting surface. It is possibly the most simple extended silica surface. Two different sites on the quartz surface are considered: site A above a 6-ring and site B above an Si atom.

The equilibrium geometries of the adsorbed molecules were preoptimized using vdW-DF2. This is justified by the fact that while vdW-DF2 generally overbinds on 1T and 2T models, it provides very accurate bonding distances. In the periodic settings, vdW-DF/CC is evaluated by considering all silica–molecule atom pairs which are closer than some threshold. The contribution of pairs distanced 20 to 30 Å is less than one hundredth of kcal/mol, so the threshold of 30 Å has been adopted.

Table 5.2 shows the calculated binding energies. They range from −0.8 kcal/mol

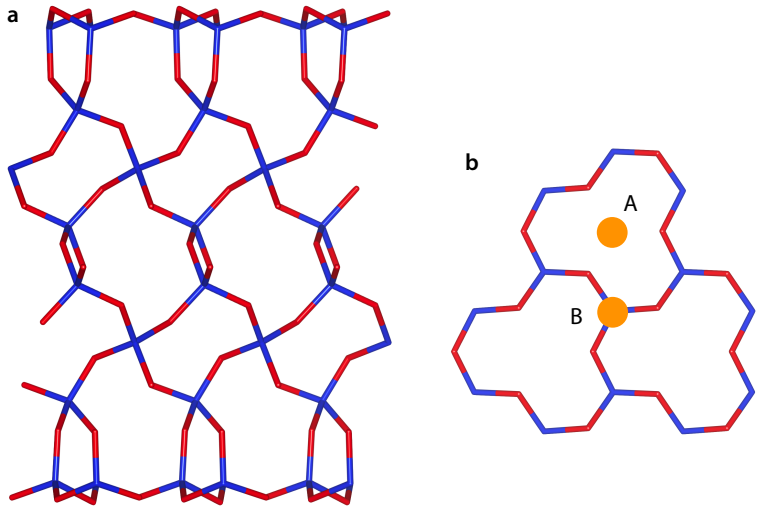

Figure 5.7 | Quartz surface. The side (**a**) and top (**b**) views of the perfectly reconstructed *α*-quartz silica surface slab with sites A and B denotes by orange circles. Site A is on top of the 6-ring, site B is on top of the Si atom.

Table 5.2 | Binding energies (kcal/mol) of molecules with the quartz surface by different methods.

Molecule	Site[a]	GGA[b]	vdW-DF2	vdW-DF/CC	ΔvdW-DF/CC[c]
CH_4	A	1.2	-2.9	-1.9	1.1
	B	0.9	-3.1	-1.8	1.3
CO_2	A	0.5	-5.1	-3.9	1.2
	B	0.9	-4.0	-3.0	1.0
H_2	A	0.2	-1.7	-1.1	0.6
	B	0.3	-1.3	-0.8	0.5
H_2O	A	-0.8	-4.8	-4.3	0.5
	B	-0.1	-3.5	-2.7	0.7
N_2	A	1.0	-3.1	-1.9	1.2
	B	0.8	-2.4	-1.4	1.0

[a] The sites are defined in Figure 5.7. [b] Bare semi-local functional, $PW86LDA_c$.
[c] The difference vdW-DF/CC − vdW-DF2.

for H_2 and site B to −4.3 kcal/mol for H_2O and site A (our method). vdW-DF2 overbinds by 0.5 to 1.3 kcal/mol. The difference between vdW-DF/CC and vdW-DF2 is most pronounced for weakly bonded molecules (CH_4, H_2, N_2), where it can be more than 50%. This also illustrates why physisorption is such a problem

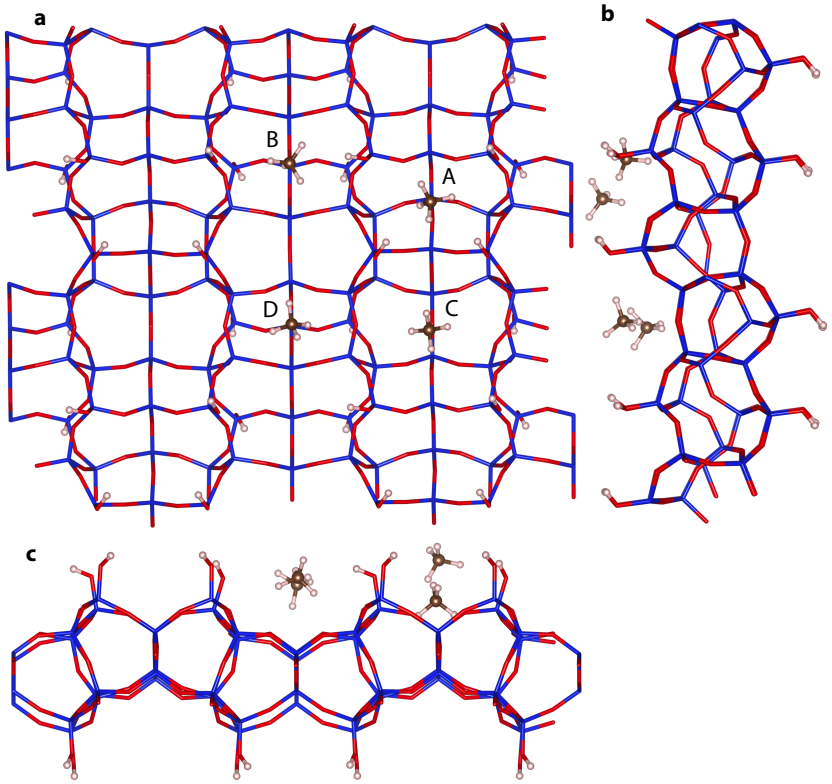

Figure 5.8 | UTL lamella. The front (**a**) and side (**b**, **c**) views of the UTL lamella with a methane molecule bonded to four different sites denotes by letters A–D. Site A is on top of the "hill" formed by four silanol groups, sites B and C are in the valleys between the silanol hills and site D is in the intersection of the valleys.

from the theoretical point of view. Even absolute errors of less than 1 kcal/mol can produce significant relative errors.

The UTL lamella is obtained by removal of the double-4-ring units from UTL [53]. The material made of stacked UTL lamellas is one of the first examples of the so-called 2D zeolites. Properties of these novel materials are yet mostly unraveled, but as their characterisation is often based on adsorption experiments, it is of great importance to know the binding energies of small probe molecules on the surface of an individual lamella. Figure 5.8 shows the computational model of the UTL lamella, along with four different bonding sites which were considered. The choice of these sites is justified by the topological picture of the surface, which is that of two perpendicular systems of valleys separating the silanol hills. The sites are chosen on the hills, in the passes in-between and in the valley intersections. The computational procedure is identical to the case of the quartz-surface.

Table 5.3 | Binding energies (kcal/mol) of molecules with the UTL lamella by different methods.

Molecule	Site[a]	GGA[b]	vdW-DF2	vdW-DF/CC	ΔvdW-DF/CC[c]
CH_4	A	0.8	-5.8	-3.8	2.0
	B	0.5	-6.2	-4.1	2.1
	C	1.1	-5.2	-3.3	1.9
	D	0.9	-3.9	-2.6	1.3
CO_2	A	-3.8	-11.3	-9.8	1.5
	B	1.4	-8.0	-6.1	1.9
	C	-1.8	-7.5	-6.4	1.0
	D	0.0	-6.4	-5.2	1.2
H_2	A	-0.3	-2.9	-2.2	0.8
	B	-0.3	-2.9	-2.2	0.7
	C	0.0	-2.3	-1.7	0.7
	D	0.0	-2.0	-1.5	0.5
H_2O	A	-10.0	-16.3	-15.3	1.0
	B	-7.0	-11.3	-10.8	0.6
	C	-8.5	-12.9	-12.4	0.5
	D	-2.3	-7.0	-6.5	0.5
N_2	A	-1.5	-6.7	-5.1	1.6
	B	-0.3	-5.7	-4.1	1.6
	C	-1.1	-5.6	-4.3	1.2
	D	0.7	-4.2	-2.8	1.4

[a] The sites are defined in Figure 5.8. [b] Bare semi-local functional, $PW86LDA_c$.
[c] The difference vdW-DF/CC − vdW-DF2.

Table 5.3 presents the calculated binding energies. Compared to the quartz surface, the energies are significantly bigger, ranging from −1.5 kcal/mol for H_2 and site D (valleys intersection) to −15.3 kcal/mol for H_2O and site A (silanol hill). The preference for bonding decreases generally in the order A > B ≈ C > D and it should be noted that this is reproduced by both methods. However, as in the case of quartz, the weakly bonded molecules are seriously overbinded. vdW-DF2 overbinds by 0.5 to 2.1 kcal/mol with respect to vdW-DF/CC, CH_4 being the most serious case with 50% overbinding. The difference between ΔvdW-DF of different sites (same molecule) can be as much as 100%, showing that our correction scheme is sensitive to the surrounding of the adsorbing molecule.

5.4 Concluding remarks

In computational chemistry, it is usually aimed at generally applicable methods, which take only the molecular geometry as an input and produce the energy. Here, we presented a different approach, which is rather instructions on how to devise a model for a particular system. This leads to increased efforts when one wants to describe some system for the first time, but enables much firmer grip on the final errors in binding energies. Indeed, the information about accuracy of the generally applicable method is often only statistical, but one does not know how well the method works in a particular case. The earlier DFT/CC scheme is a pioneer of the system-specific approach, but it is quite laborious to use and has some minor complications in certain cases. The vdW-DF/CC is relatively simple to use and is numerically more stable.

The vdW-DF nonlocal functional is at the core of vdW-DF/CC. Our method adds the scaling which effectively dumps or enhances the vdW-DF energy to better match the underlying semi-local functional. Indeed, the DFT/CC can cover even errors in electrostatic interactions coming from the semi-local functional, due to different correction curves for each individual atom types. On the other hand, vdW-DF/CC has one universal correction curve for all atoms, which cannot represent the anisotropic electrostatic bonding. Thus we can suppose that our empirical scheme merely improves the description of dispersion forces.

The results on the 1T and 2T models illustrate that the transferability of vdW-DF/CC is very good. The models differ quite significantly, yet the errors on the 2T model are generally within 0.1 kcal/mol, even though the method is parametrized on the 1T model. The similarity of the 2T model with real silica surfaces (the Si–O–Si bridges) leads to the conclusion that the parametrization should be well transferable even to extended systems. The accuracy of vdW methods for DFT is often measured in absolute errors. However, this can be misleading as the vdW binding energies span more than one order of magnitude. We illustrated on the cases of quartz surface and UTL lamella that the chemical accuracy of 1 kcal/mol is insufficient for some physisorbed molecules, whose binding energy is in the order of several kcal/mol. While the non-empirical vdW-DF2 method described all qualitative aspects correctly, our method provides a higher level of accuracy, which is needed for quantitative comparison with experiments.

Summary

The aim of this thesis was to investigate the nonlocal correlation density functional methods. The van der Waals interactions has become an extensively studied phenomenon in computational chemistry and the nonlocal functionals provide a straightforward way for their description. However, the accuracy of these methods has not reached the quality of state-of-the-art empirical methods. The motivation for this study was to produce new findings about the nonlocal functionals that could bring ideas about the improvement of their accuracy.

We have presented several studies about the behaviour of vdW-DF, the oldest nonlocal functional. We have found out that the nonlocal energy is well localized on the atom pairs and that the main contribution comes from the valence shell electrons. The replacement of the semi-local gradient correction to correlation by the non-local correlation reproduces qualitatively the node structure of the energy density, but is not quantitatively equivalent. We haven shown that the vdW-DF nonlocal energy can greatly improve the binding energies of excimers in excited states calculated by the CASSCF method. The ground and non-bonding excited states are described very well by our combination, but there is a room for improvement for the bonding states. We have extensively studied the Z_{ab} parameter of the vdW-DF functional. We have shown that it can be reoptimized to provide higher accuracy for the S22 benchmark set. The optimal value depends on the semi-local functional and it has been found that the revPBE exchange functional with the revPBE semi-local correlation is the best performer. The optimizations of Z_{ab} on the noble gas dimers showed that the parameter is formally dependent on the distance. However, this dependency is not compatible with the theory and indicates that the PW86LDA$_c$ semi-local functional is not fully compatible with the nonlocal functional.

In the second part, we presented an empirical correction scheme which takes vdW-DF as a starting point. The vdW-DF is decomposed into atom pairs and the resulting pair curves are scaled using the RKHS interpolation. The method is formally very similar to the earlier DFT/CC method, which has been shown

to have a superb accuracy. Our scheme retains much of this quality and adds simplicity and robustness. We have tested our method on silica-based surfaces. The difference between our method and vdW-DF can be as much as 2 kcal/mol or 50%. Such differences can be crucial when comparing theoretical results with experiments.

This thesis represents a broad study of the vdW-DF functional. It gave us the insight into how the method works, where does it work well and where are its flaws. It showed how a non-empirical method can be combined with an empirical scheme to reach better accuracy. We hope that this will enable more studies where experimental results are accompanied by theoretical rationalizations.

References

1. Autumn, K. *et al.* Evidence for van der Waals adhesion in gecko setae. *P. Natl. Acad. Sci. USA* **99**, 12252–12256 (2002).
2. Mandelbrot, B. How long is the coast of Britain? Statistical self-similarity and fractional dimension. *Science* **156**, 636–638 (1967).
3. Langreth, D. C. *et al.* A density functional for sparse matter. *J. Phys.: Condens. Matter* **21**, 084203 (2009).
4. Johnson, E. R. *et al.* Revealing noncovalent interactions. *J. Am. Chem. Soc.* **132**, 6498–6506 (2010).
5. Burke, K. Perspective on density functional theory. *J. Chem. Phys.* **136**, 150901 (2012).
6. Langreth, D. C. & Vosko, S. H. Exact electron-gas response functions at high density. *Phys. Rev. Lett.* **59**, 497–500 (1987).
7. Andersson, Y., Langreth, D. C. & Lundqvist, B. I. Van der Waals interactions in density-functional theory. *Phys. Rev. Lett.* **76**, 102–105 (1996).
8. Langreth, D. *et al.* Van der Waals density functional theory with applications. *Int. J. Quantum Chem.* **101**, 599–610 (2005).
9. Dion, M., Rydberg, H., Schröder, E., Langreth, D. & Lundqvist, B. Van der Waals density functional for general geometries. *Phys. Rev. Lett.* **92**, 246401 (2004).
10. Piela, L. *Ideas of Quantum Chemistry* (Elsevier, 2007).
11. Koch, W. & Holthausen, M. C. *A Chemist's Guide to Density Functional Theory* (Wiley-VCH, 2001).
12. Berland, K. *Bound by long-range interactions: Molecular crystals and benzene on Cu(111)*. Ph.D. thesis, Chalmers University of Technology (2009).
13. Parr, R. G. & Yang, W. *Density-functional theory of atoms and molecules* (Oxford University Press, 1989).
14. Burke, K. & Wagner, L. O. DFT in a nutshell. *Int. J. Quantum Chem.* **113**, 96–101 (2013).
15. Kristyán, S. & Pulay, P. Can (semi)local density functional theory account for

the London dispersion forces? *Chem. Phys. Lett.* **229**, 175–180 (1994).

16. Dion, M. *Van der Waals forces in density functional theory*. Ph.D. thesis, Rutgers University, New Jersey (2004).
17. Tkatchenko, A., Ambrosetti, A. & Robert A. DiStasio, J. Interatomic methods for the dispersion energy derived from the adiabatic connection fluctuation-dissipation theorem. *J. Chem. Phys.* **138**, 074106 (2013).
18. Lee, K., Murray, E. D., Kong, L., Lundqvist, B. I. & Langreth, D. C. Higher-accuracy van der Waals density functional. *Phys. Rev. B* **82**, 081101 (2010).
19. Vydrov, O. A. & Van Voorhis, T. Improving the accuracy of the nonlocal van der Waals density functional with minimal empiricism. *J. Chem. Phys.* **130**, 104105 (2009).
20. Vydrov, O. A. & Van Voorhis, T. Nonlocal van der Waals density functional made simple. *Phys. Rev. Lett.* **103**, 063004 (2009).
21. Langreth, D. C. & Lundqvist, B. I. Comment on "Nonlocal van der Waals density functional made simple". *Phys. Rev. Lett.* **104**, 099303 (2010).
22. Vydrov, O. A. & Van Voorhis, T. Vydrov and Van Voorhis reply. *Phys. Rev. Lett.* **104**, 099304 (2010).
23. Vydrov, O. A. & Van Voorhis, T. Nonlocal van der Waals density functional: The simpler the better. *J. Chem. Phys.* **133**, 244103 (2010).
24. Murray, E. D., Lee, K. & Langreth, D. C. Investigation of exchange energy density functional accuracy for interacting molecules. *J. Chem. Theory Comput.* **5**, 2754–2762 (2009).
25. Pernal, K., Podeszwa, R., Patkowski, K. & Szalewicz, K. Dispersionless density functional theory. *Phys. Rev. Lett.* **103**, 263201 (2009).
26. Cooper, V. R. Van der Waals density functional: An appropriate exchange functional. *Phys. Rev. B* **81**, 161104 (2010).
27. Klimeš, J., Bowler, D. R. & Michaelides, A. Chemical accuracy for the van der Waals density functional. *J. Phys.: Condens. Mattter* **22**, 022201 (2010).
28. Klimeš, J., Bowler, D. R. & Michaelides, A. Van der Waals density functionals applied to solids. *Phys. Rev. B* **83**, 195131 (2011).
29. Austin, A. *et al.* A density functional with spherical atom dispersion terms. *J. Chem. Theory Comput.* **8**, 4989–5007 (2012).
30. Wellendorff, J. *et al.* Density functionals for surface science: Exchange-correlation model development with Bayesian error estimation. *Phys. Rev. B* **85**, 235149 (2012).
31. Román-Peréz, G. & Soler, J. M. Efficient implementation of a van der Waals

density functional: Application to double-wall carbon nanotubes. *Phys. Rev. Lett.* **103**, 096102 (2009).

32. Thonhauser, T. *et al.* Van der Waals density functional: Self-consistent potential and the nature of the van der Waals bond. *Phys. Rev. B* **76**, 125112 (2007).
33. Vydrov, O. A., Wu, Q. & Van Voorhis, T. Self-consistent implementation of a nonlocal van der Waals density functional with a gaussian basis set. *J. Chem. Phys.* **129**, 014106 (2008).
34. Jurečka, P., Šponer, J., Černý, J. & Hobza, P. Benchmark database of accurate (MP2 and CCSD(T) complete basis set limit) interaction energies of small model complexes, DNA base pairs, and amino acid pairs. *Phys. Chem. Chem. Phys.* **8**, 1985–1993 (2006).
35. Rocha-Rinza, T., De Vico, L., Veryazov, V. & Roos, B. O. A theoretical study of singlet low-energy excited states of the benzene dimer. *Chem. Phys. Lett.* **426**, 268–272 (2006).
36. Werner, H.-J. & Knowles, P. J. MOLPRO, version 2010.1 (University College Cardiff Consultants Limited, UK, 2008). Available at http://www.molpro.net/.
37. Riley, K. E., Pitoňák, M., Jurečka, P. & Hobza, P. Stabilization and structure calculations for noncovalent interactions in extended molecular systems based on wave function and density functional theories. *Chem. Rev.* **110**, 5023–5063 (2010).
38. Carter, D. J. & Rohl, A. L. Noncovalent interactions in SIESTA using the vdW-DF functional: S22 benchmark and macrocyclic structures. *J. Chem. Theory Comput.* **8**, 281–289 (2012).
39. Schlapbach, L. & Züttel, A. Hydrogen-storage materials for mobile applications. *Nature* **414**, 353–358 (2001).
40. Morris, R. E. & Wheatley, P. S. Gas storage in nanoporous materials. *Angew. Chem. Int. Edit.* **47**, 4966–4981 (2008).
41. Kruk, M. & Jaroniec, M. Gas adsorption characterization of ordered organic-inorganic nanocomposite materials. *Chem. Mater.* **13**, 3169–3183 (2001).
42. Wu, X., Vargas, M. C., Nayak, S., Lotrich, V. & Scoles, G. Towards extending the applicability of density functional theory to weakly bound systems. *J. Chem. Phys.* **115**, 8748–8757 (2001).
43. Klimeš, J. & Michaelides, A. Perspective: Advances and challenges in treating van der Waals dispersion forces in density functional theory. *J. Chem. Phys.* **137**, 120901 (2012).

44. Chen, L., Grajciar, L., Nachtigall, P. & Düren, T. Accurate prediction of methane adsorption in a metal-organic framework with unsaturated metal sites by direct implementation of an ab-initio derived potential energy surface in GCMC simulation. *J. Phys. Chem. C* **115**, 23074–23080 (2011).
45. Grimme, S., Antony, J., Ehrlich, S. & Krieg, H. A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu. *J. Chem. Phys.* **132**, 154104 (2010).
46. Goerigk, L. & Grimme, S. Efficient and accurate double-hybrid-meta-GGA density functionals: Evaluation with the extended GMTKN30 database for general main group thermochemistry, kinetics, and noncovalent interactions. *J. Chem. Theory Comput.* **7**, 291–309 (2011).
47. Hujo, W. & Grimme, S. Performance of the van der Waals density functional VV10 and (hybrid)GGA variants for thermochemistry and noncovalent interactions. *J. Chem. Theory Comput.* **7**, 3866–3871 (2011).
48. Vydrov, O. A. & Van Voorhis, T. Benchmark assessment of the accuracy of several van der Waals density functionals. *J. Chem. Theory Comput.* **8**, 1929–1934 (2012).
49. Bludský, O., Rubeš, M., Soldán, P. & Nachtigall, P. Investigation of the benzene-dimer potential energy surface: DFT/CCSD(T) correction scheme. *J. Chem. Phys.* **128**, 114102 (2008).
50. Frisch, M. J. *et al.* Gaussian 09, Revision A.02 (Gaussian, Inc., Wallingford, CT, 2009). Available at http://www.gaussian.com/.
51. Hafner, J., Kresse, G., Vogtenhuber, D. & Marsman, M. VASP, version 5.2.12 (University of Vienna, Austria, 2011). Available at http://www.vasp.at/.
52. Chen, Y.-W., Cao, C. & Cheng, H.-P. Finding stable α-quartz (0001) surface structures via simulations. *Appl. Phys. Lett.* **93**, 181911 (2008).
53. Chlubná, P. *et al.* 3D to 2D routes to ultrathin and expanded zeolitic materials. *Chem. Mater.* **25**, 542–547 (2013).
54. Becke, A. D. A multicenter numerical integration scheme for polyatomic molecules. *J. Chem. Phys.* **88**, 2547–2553 (1988).

Appendix A

Evaluation of vdW-DF in Matlab

To be able to tweak vdW-DF and use it in unusual ways, I have implemented it in MATLAB, a high-level programming language that is interpreted on-the-fly rather than compiled into a binary code. It is intended for numerical computation and contains a large number of mathematical functions available as ready-to-use libraries. All source files needed to run my implementation are listed below.

Implementing the evaluation of any energy density functional requires almost the complete DFT machinery, except for the self-consistent cycle. This includes construction of a quadrature grid [54], evaluation of the basis functions, implementation of mathematical formulas defining the functionals and performing the quadrature. The information about the density is obtained from the KS orbitals, which are read from the Molden format outputted by Molpro [36].

Listing A.1 | setparam.m

```
% loads parameters

function param = setparam(file)
    param = defaults();
    if nargin > 0
        param = addparams(param,file);
    end
end

function param = defaults()
    param.fid = 0;
    param.method = 'hf';
    param.maxiter = 64; % SCF iterations
    param.diismax = 10; % number of DIIS matrixes
    param.debug = 0; % no debug info
    param.nonlocal = '';
    param.scfmax = 1e6;
    param.scfrms = 1e8;
    param.scfdelta = 1e6;
```

```
    param.grid = 'fine';
    param.densfit = '';
    param.charge = 0;
    param.mult = 1;
    param.basistype = 'default';
    param.nlcignore = 1e6;
    param.nlccontrib = false;
    param.zab = nan;
end

function param = addparams(param,file)
    fid = fopen(file,'r');
    while ~feof(fid)
        l = fgets(fid);
        if isempty(l) || l(1)=='%' % empty or comment line
            continue
        end
        tok = regexp(l,'(\S+)\s*=(.*)','tokens');
        if isempty(tok)
            continue
        else
            var = tok{1}{1}; % paramter
        end
        tok{1}{2} = strtrim(tok{1}{2});
        if isempty(str2num(tok{1}{2}))
            val = tok{1}{2};
        else
            val = str2num(tok{1}{2});
        end
        param.(var) = val;
    end
    fclose(fid);
end
```

Listing A.2 | dispcorr.m

```
function E = dispcorr(paramarg)
    global param
    param = paramarg;
    param.maxder = 1;
    param.basistype = 'molpro';
    if isnan(param.zab)
        switch param.nonlocal
            case 'vdw', zabs = 0.8491; % vdwdf
            case 'vdw2', zabs = 1.887; % vdwdf2
        end
    else zabs = param.zab;
    end
    [C, occ, bas, geom] = loadmolden(param.molden);
    bas = normalize(bas);
    frag1 = 1:param.frags(1);
    frag2 = (param.frags(1)+1):sum(param.frags);
```

```
    P = C*diag(occ)*C'; % density matrix
    grid = buildgrid(geom, bas);
    E = [];
    nlc = str2func(param.nonlocal);
    for i = 1:length(zabs)
        param.zab = zabs(i);
        j = 1;
        for a = 1:length(frag1)
            for b = 1:length(frag2)
                E(j, i) = 2 * evalnlc(P, nlc, grid(frag1(a)), grid(frag2(b)));
                j = j + 1;
            end
        end
    end
    if ~param.nlccontrib
        E = sum(E,1);
    end
end
```

Listing A.3 | loadmolden.m

```
function [C, occ, bas, geom] = loadmolden(file)
    spherical = false(5, 1);
    f = fopen(file,'r');
    while ~feof(f)
        s = fgets(f);
        if isempty(regexp(s, '\S+', 'once'))
            continue
        end
        section = regexp(s, '\[([\w\s]+)\](.*)?', 'tokens');
        if isempty(section)
            error('Invalid Molden format')
        end
        [section, info] = deal(lower(section{1}{1}), section{1}{2});
        info = regexp(info, '(\w+)', 'tokens');
        if isempty(info)
            info = [];
        else
            info = lower(info{1}{1});
        end
        switch section
            case 'molden format'
            case 'n_atoms'
                fgets(f);
            case 'molpro variables'
                var = molprovariables(f);
            case 'atoms'
                if strfind(info, 'ang')
                    units = 'angs';
                elseif strfind(info, 'au')
                    units = 'au';
                else
```

```
                units = 'angs';
                warning('Units in Molden file not specified, assuming angstroms');
            end
            geom = readgeom(f, units);
        case '5d'
            spherical(2) = true;
        case '7f'
            spherical(3) = true;
        case '9g'
            spherical(4) = true;
        case 'charge'
            for i = 1:length(geom.atoms)
                fgets(f);
            end
        case 'gto'
            bas = readbasis(f, geom);
        case 'mo'
            nbas = 0;
            for i = 1:length(bas)
                l = bas(i).l;
                if l > 0 && spherical(l)
                    type = 's';
                else
                    type = 'c';
                end
                nbas = nbas + nharm(l, type);
            end
            [C, occ, missed] = readorbitals(f, nbas);
        otherwise
            warning('Unknown Molden section')
    end
end
fclose(f);
if ~isempty(missed)
    warning(['Orbitals ' sprintf('%g ', missed) 'were set to zero']);
end
if any(spherical)
    i = 1;
    j = 1;
    for k = 1:length(bas)
        l = bas(k).l;
        if l > 0 && spherical(l)
            B = car2sphr(l, str2func('genns2'), 'real')';
        else
            B = eye(nharm(l, 'c'));
        end
        A(i:i+size(B, 1)1, j:j+size(B, 2)1) = B;
        i = i+size(B, 1);
        j = j+size(B, 2);
    end
    C = A * C;
end
```

```
end

function n = nharm(l, type)
    switch lower(type(1))
        case 'c'
            n = (l+1)*(l+2)/2;
        case 's'
            n = 2*l + 1;
    end
end

function var = molprovariables(f)
    while ~feof(f)
        s = fgets(f);
        switch s(1)
            case '!'
                tok = regexp(s, '!(\w+)=\s*(\S*)', 'tokens');
                val = sscanf(tok{1}{2}, '%f', 1);
                if isempty(val)
                    switch tok{1}{2}
                        case 'TRUE'
                            val = true;
                        case 'FALSE'
                            val = false;
                        otherwise
                            val = [];
                    end
                end
                var.(tok{1}{1}) = val;
            case '['
                fseek(f, length(s), 0);
                break
        end
    end
end

function geom = readgeom(f, units)
    if strcmp(units, 'angs')
        bohr = 1.889725989;
    else
        bohr = 1;
    end
    geom.atoms = [];
    geom.xyz = [];
    geom.frags = [];
    while true
        atom = fscanf(f, '%s', 1);
        if atom(1) == '['
            fseek(f, length(atom), 0);
            break
        end
        elem = regexp(atom, '([azAZ]+)', 'tokens');
```

```
            geom.atoms(end+1,1) = element(elem{1}{1});
            fscanf(f, '%d', 2);
            geom.xyz(end+1, :) = bohr*fscanf(f, '%f', [1 3]);
            if geom.xyz(end, 3) < 0
                frag = 1;
            else
                frag = 2;
            end
            geom.frags(end+1,1) = frag;
        end
    end

    function bas = readbasis(f,geom)
        alphabet = 'spdfghi';
        for i = 1:length(alphabet)
            angular.(alphabet(i)) = i1;
        end
        lbf = 1;
        bas = [];
        for i = 1:length(geom.atoms)
            atom = fscanf(f, '%d', 1); fgets(f);
            while true
                s = fgets(f);
                tok = regexp(s,'\s*([spdfghi])\s+(\d+)','tokens');
                if isempty(tok)
                    break
                end
                l = angular.(tok{1}{1});
                exp = [];
                contr = [];
                nprim = sscanf(tok{1}{2}, '%d');
                for j = 1:nprim
                    s = fgets(f);
                    s = regexprep(s, 'D', 'e');
                    d = sscanf(s, '%f', 2);
                    if d(2) == 0
                        continue
                    end
                    exp(end+1, 1) = d(1);
                    contr(end+1, 1) = d(2);
                end
                R = geom.xyz(atom, :)';
                Nr = radialnorm(exp, l); % radial part of norms
                Na = angularnorm(l); % angluar part of norms
                Nbf = length(Na); % number of cartesian primitives
                bas = [bas; struct(...
                    'R', R, 'l', l, 'exp', exp, 'contr', contr,...
                    'Nr', Nr, 'Na', Na, 'Nbf', Nbf, 'lbf', lbf)];
                lbf = lbf + Nbf;
            end
        end
    end
```

```
function [C, occ, missed] = readorbitals(f, n)
    missed = [];
    C = zeros(n);
    occ = zeros(n, 1);
    i = 1;
    while true
        if i == 411
        end
        for j = 1:4
            s = fgets(f);
        end
        if s == 1
            break % end of file
        end
        tok = regexp(s, '=\s+(\S+)', 'tokens');
        occ(i) = sscanf(tok{1}{1}, '%f');
        s = fscanf(f, '%f', [2 n])';
        fgets(f);
        if size(s, 1) < n
            missed(end+1) = i;
            for j = 1:(n  size(s, 1))
                fgets(f);
            end
            s = zeros(n, 2);
        end
        C(:, i) = s(:, 2);
        i = i+1;
    end
    C(:, i:end) = [];
    occ(i:end) = [];
end

function y = radialnorm(zeta,l)
    y = (2*zeta/pi).^(3/4).*(4*zeta).^(l/2);
end

function y = angularnorm(l)
    ns = genns2(l);
    N = lton(l);
    y = zeros(N,1);
    for i = 1:N
        y(i) = 1/sqrt(dbf(2*ns(i,1)1)*dbf(2*ns(i,2)1)*dbf(2*ns(i,3)1));
    end
end

% double factorial n!!
function y = dbf(n)
    y = prod(1:2:n);
    if y == 0
        y = 1;
    end
```

```
end
```

Listing A.4 | normalize.m

```
% calculates 1electron integrals (overlap S, core H) and
% normalizes contractions in basis set

function bas = normalize(bas)
    genindexing1(10);
    Nsh = length(bas); % number of shells
    Ibf = [bas.Ibf bas(end).Ibf+bas(end).Nbf]; % shell location
    S = zeros(Ibf(end)1);
    for i = 1:Nsh % loop over diagonal
        ri = Ibf(i):Ibf(i+1)1; % shell range
        s = calc2shell(bas([i i])); % calculate the submatrix
        S(ri,ri) = s; % update the big matrix
    end
    N = 1./sqrt(diag(S)); % inverse norms
    for i = 1:Nsh
        bas(i).contr = N(Ibf(i))*bas(i).contr; % normalize contractions
    end
end

% calculates 1electron integrals over contracted shells
function S = calc2shell(sh)
    ls = [sh.l]; % angular momenta
    if ls(1)<ls(2), pi = [2 1]; else
        pi = [1 2]; end
    sh = sh(pi);
    ls = ls(pi);
    S = zeros(sh.Nbf);
    NprA = length(sh(1).exp);
    NprB = length(sh(2).exp);
    factor = (sh(1).contr*sh(2).contr').*(sh(1).Nr*sh(2).Nr');
    code = [10 1]*ls'; % e.g., 20 for ds
    for i = 1:NprA % loop over primitives
        for j = 1:NprB
            sprim = calc2prim(...
                ls,code,sh(1).exp(i),sh(2).exp(j),sh.R);
            S = S + factor(i,j)*sprim;
        end
    end
    anorms = sh(1).Na*sh(2).Na';
    S = anorms.*S;
    S = ipermute(S,pi);
end

% calculates 1electron integrals over primitive gaussians
function int = calc2prim(l,code,za,zb,A,B)
    zeta = za+zb;
    xi = za*zb/zeta;
    P = (za*A+zb*B)/zeta;
```

```
    ss = calcss(A,B,zeta,xi);
    switch code
        case 00
            int = ss;
        otherwise
            int = generalint(l,A,B,P,zeta,ss);
    end
end

function ss = calcss(A,B,zeta,xi)
    AB2 = sum((AB).^2);
    ss = (pi/zeta)^(3/2)*exp(xi*AB2);
end

% calculates 1el integrals for general l

function ints = generalint(l,A,B,P,zeta,ss)
    dPR = (P(:,ones(1,2))[A B])';
    S = preparetrees(l,ss);
    for b = 1:l(2)
        S = step(S,dPR,zeta,1,b,2);
    end
    for a = 1:l(1)
        for b = 1:l(2)+1
            S = step(S,dPR,zeta,a,b,1);
        end
    end
    ints = extracttargets(S);
end

function ints = extracttargets(S)
    ints = S{end};
end

function S = preparetrees(l,ss)
    S = cell(l(1)+1,l(2)+1);
    S{1,1} = ss;
end

function S = step(S,dPR,zeta,a,b,pos)
    global anc des ni
    switch pos
        case 1
            at = a+1;
            bt = b;
        case 2
            at = a;
            bt = b+1;
    end
    A = lton(at1);
    B = lton(bt1);
    % prepare integral matrices
```

```
    S{at,bt} = zeros(A,B);
    % loop over the integral matrix
    l = 1;
    for k = 1:lton(a1)
        for l = 1:lton(b1)
            ind = [k l];
            switch ind(pos)
                case 1
                    i = 1:3;
                case {2,4,7,11,16,22}
                    i = 2:3;
                otherwise
                    i = 3;
            end
            for q = 1:length(i)
                S = increase(a,b,k,l,pos,i(q),S,dPR,zeta,anc,des,ni);
                l = l+1;
            end
        end
    end
end

function S = increase(a,b,k,l,pos,i,S,dPR,zeta,anc,des,ni)
    anck = anc(a,k,i);
    ancl = anc(b,l,i);
    desk = des(a,k,i);
    desl = des(b,l,i);
    nia = ni(a,k,i);
    nib = ni(b,l,i);
    invz = 1/(2*zeta);
    % (a,b) terms
    st = dPR(pos,i)*S{a,b}(k,l);
    % (a1,b) terms
    if anck > 0
        st = st + invz*nia*S{a1,b}(anck,l);
    end
    % (a,b1) terms
    if ancl > 0
        st = st + invz*nib*S{a,b1}(k,ancl);
    end
    % (a+1,b) term
    % update the tree
    switch pos
        case 1
            S{a+1,b}(desk,l) = st;
        case 2
            S{a,b+1}(k,desl) = st;
    end
end
```

Listing A.5 | buildgrid.m

```
% builds grid for all atoms and evalutes basis set on that grid

function grid = buildgrid(geom,bas)
    N = length(geom.atoms);
    grid = [];
    for n = 1:N
        xyz = geom.xyz(n,:);
        atom = geom.atoms(n);
        grid1 = build1grid(xyz,atom);
        grid1.w = grid1.w.*evalwn(n,grid1,geom); % w_n
        grid1 = evalbasis(grid1,bas); % evaluate basis functions
        grid = [grid; grid1];
    end
end
```

Listing A.6 | build1grid.m

```
% builds grid for one atom

function grid = build1grid(R,Z)
    global param
    if Z <= 2, period = 1;
    elseif Z <= 10, period = 2;
    elseif Z <= 18, period = 3;
    elseif Z <= 36, period = 4; else
        error('Don''t have grids for Z>36');
    end
    rad = radius(Z,'grid');
    switch period
        case 1
            part = rad*[0 0.25 0.5 1 4.5 Inf];
        case 2
            part = rad*[0 0.1667 0.5 0.9 3.5 Inf];
        otherwise
            part = rad*[0 0.1 0.4 0.8 2.5 Inf];
    end
    switch param.grid
        case 'nlc'
            nrad = 30;
            nang = [6 26 50 110 50];
        case 'sg1'
            nrad = 50;
            nang = [6 38 86 194 86];
        case 'fine'
            nrad = 70;
            nang = [6 38 110 302 110];
        case 'ultrafine'
            nrad = 75;
            nang = [14 50 146 350 146];
        case 'plot'
            nrad = 70;
```

```
            nang = 302*ones(1,5);
        case 'eml75302'
            nrad = 75;
            nang = 302*ones(1,5);
        otherwise
            error('unknown grid');
    end
    i = 1:nrad;
    r = rad*i.^2./(nrad+1i).^2;
    wr = 2*rad^3*(nrad+1)*i.^5./(nrad+1i).^7;
    [grid.x,grid.y,grid.z,grid.w] = deal([]);
    for i = 1:5
        leb = getLebedevSphere(getdeg(nang(i)));
        ind = and(r>part(i),r<=part(i+1));
        rpart = r(ind);
        wrpart = wr(ind);
        grid.x = [grid.x; reshape(rpart'*leb.x',[],1)+R(1)];
        grid.y = [grid.y; reshape(rpart'*leb.y',[],1)+R(2)];
        grid.z = [grid.z; reshape(rpart'*leb.z',[],1)+R(3)];
        grid.w = [grid.w; reshape(wrpart'*leb.w',[],1)];
    end
end

% gives nearest greater available Lebedev degree
function n = getdeg(n)
    lebdeg = [6 14 26 38 50 74 86 110 146 170 194 230 266 302 ...
            350 434 590 770 974 1202 1454 1730 2030 2354 2702 3074 ...
            3470 3890 4334 4802 5294 5810];
    n = lebdeg(find(lebdeg>=n,1));
end
```

Listing A.7 | evalwn.m

```
% evaluates atomic cell weights

function wn = evalwn(n,grid,geom)
    N = length(geom.atoms);
    p = @(x)((3*xx.^3)/2);
    sx = @(pppx)(.5*(1pppx));
    dist = @(x,y,z,r)(sqrt((xr(1)).^2+(yr(2)).^2+(zr(3)).^2));
    P = cell(1,N);
    r = cell(1,N);
    for i = 1:N
        r{i} = dist(grid.x,grid.y,grid.z,geom.xyz(i,:));
    end
    s = cell(N);
    for i = 1:N
        P{i} = ones(size(grid.x));
        for j = 1:i1
            Rij = sqrt(sum((geom.xyz(j,:)geom.xyz(i,:)).^2));
            muij = (r{i}r{j})/Rij;
            chi = sqrt(radius(geom.atoms(i),'atom')...
```

```
                    /radius(geom.atoms(j),'atom'));
                uij = (chi1)/(chi+1);
                aij = uij/(uij^21);
                aij(aij>1/2) = 1/2;
                aij(aij<1/2) = 1/2;
                nuij = muij+aij*(1muij.^2);
                pppx = p(p(p(nuij)));
                s{i,j} = sx(pppx);
                s{j,i} = sx(pppx);
            end
        end
        for i = 1:N
            P{i} = ones(size(grid.x));
            for j = [1:i1 i+1:N]
                P{i} = P{i}.*s{i,j};
            end
        end
        sumPm = zeros(size(grid.x));
        for m = 1:N
            sumPm = sumPm + P{m};
        end
        wn = P{n}./sumPm;
end
```

Listing A.8 | evalbasis.m

```
% evaluates basis on a grid

function grid = evalbasis(grid,bas)
    global param
    Nsh = length(bas); % number of shells
    Ibf = [bas.Ibf bas(end).Ibf+bas(end).Nbf]; % shell location
    N = Ibf(end)1;
    ngrid = length(grid.w);
    grid.f = zeros(ngrid,N);
    if param.maxder > 0
        [grid.fx,grid.fy,grid.fz] = deal(grid.f);
    end
    if param.maxder > 1
        [grid.fxx,grid.fyy,grid.fzz,grid.fxy,grid.fxz,grid.fyz] = ...
            deal(grid.f);
    end
    for i = 1:Nsh % loop over all shells
        b = bas(i);
        x = grid.xb.R(1);
        y = grid.yb.R(2);
        z = grid.zb.R(3);
        x(x==0) = 1e30;
        y(y==0) = 1e30;
        z(z==0) = 1e30;
        R2 = x.^2+y.^2+z.^2;
        [sum0,sum1,sum2] = deal(zeros(ngrid,1));
```

```
        for j = 1:length(b.exp) % loop over primitives
            t = b.contr(j)*b.Nr(j)*exp(b.exp(j)*R2);
            sum0 = sum0+t;
            sum1 = sum1+b.exp(j)*t;
            sum2 = sum2+b.exp(j)^2*t;
        end
        switch param.basistype
            case 'default', fgenns = 'genns';
            case 'molpro', fgenns = 'genns2';
        end
        fgenns = str2func(fgenns);
        ns = fgenns(b.l);
        for j = 1:size(ns,1) % loop over angular momenta
            k = lbf(i)+j1;
            n = ns(j,:);
            ang = b.Na(j).*x.^n(1).*y.^n(2).*z.^n(3);
            grid.f(:,k) = ang.*sum0;
            if param.maxder == 0, continue, end
            grid.fx(:,k) = dfdx(x,n(1),sum0,sum1,ang);
            grid.fy(:,k) = dfdx(y,n(2),sum0,sum1,ang);
            grid.fz(:,k) = dfdx(z,n(3),sum0,sum1,ang);
            if param.maxder == 1, continue, end
            grid.fxx(:,k) = d2fdx2(x,n(1),sum0,sum1,sum2,ang);
            grid.fyy(:,k) = d2fdx2(y,n(2),sum0,sum1,sum2,ang);
            grid.fzz(:,k) = d2fdx2(z,n(3),sum0,sum1,sum2,ang);
            grid.fxy(:,k) = d2fdxy(x,y,n(1),n(2),sum0,sum1,sum2,ang);
            grid.fxz(:,k) = d2fdxy(x,z,n(1),n(3),sum0,sum1,sum2,ang);
            grid.fyz(:,k) = d2fdxy(y,z,n(2),n(3),sum0,sum1,sum2,ang);
        end
    end
end

function y = dfdx(x,m,sum0,sum1,ang)
    rad = 2*x.*sum1;
    if m > 0, rad = rad+m./x.*sum0; end
    y = ang.*rad;
end

function y = d2fdx2(x,m,sum0,sum1,sum2,ang)
    rad = 2*(2*m+1)*sum1+4*x.^2.*sum2;
    if m > 1, rad = rad+m*(m1)./x.^2.*sum0; end
    y = ang.*rad;
end

function y = d2fdxy(x,y,m,n,sum0,sum1,sum2,ang)
    rad = 4*x.*y.*sum2;
    if m > 0, rad = rad2*m*y./x.*sum1; end
    if n > 0, rad = rad2*n*x./y.*sum1; end
    if m*n > 0, rad = rad+m*n./(x.*y).*sum0; end
    y = ang.*rad;
end
```

Listing A.9 | evalnlc.m

```
% calculates nonlocal energy term

function Enlc = evalnlc(P,phi,grid1,grid2)
    global param
    if nargin < 4
        grid2 = grid1;
    end
    thre = param.nlcignore;
    n1 = evaldensity(grid1,P);
    n2 = evaldensity(grid2,P);
    n1w = n1.f.*grid1.w;
    n2w = n2.f.*grid2.w;
    pass1 = n1w > thre;
    pass2 = n2w > thre;
    [n1,grid1] = reducegrid(n1,grid1,pass1);
    [n2,grid2] = reducegrid(n2,grid2,pass2);
    R = distance(grid1,grid2);
    [kern,shift] = phi(n1,n2,R);
    Enlc = n1w(pass1)'*(kern*n2w(pass2)/2+shift);
    end

function R = distance(g1,g2)
    n1 = length(g1.x);
    n2 = length(g2.x);
    ind1 = 1:n1;
    ind1 = ind1(ones(1,n2),:)';
    ind2 = 1:n2;
    ind2 = ind2(ones(1,n1),:);
    R = sqrt((g1.x(ind1)g2.x(ind2)).^2 ...
        +(g1.y(ind1)g2.y(ind2)).^2 ...
        +(g1.z(ind1)g2.z(ind2)).^2);
%  [a,b] = find(R==0);
%  for i = 1:length(a)
%   R(a(i),b(i)) = 1e100;%(g1.w(a(i))*g2.w(b(i)))^(1/6);
%  end
end

function [n,grid] = reducegrid(n,grid,pass)
    global param
    n.f = n.f(pass);
    n.g = n.g(pass);
    if param.maxder > 1
        n.lap = n.lap(pass);
        n.gHg = n.gHg(pass);
    end
    grid.x = grid.x(pass);
    grid.y = grid.y(pass);
    grid.z = grid.z(pass);
    grid.w = grid.w(pass);
    grid.f = grid.f(pass,:);
```

```
    grid.fx = grid.fx(pass,:);
    grid.fy = grid.fy(pass,:);
    grid.fz = grid.fz(pass,:);
end
```

Listing A.10 | preparekernel.m

```
function kernel = preparekernel(dD,dd)
    D = 3*(1./(1(0:dD:1))1);
    d = 0:dd:1;
    nD = length(D);
    nd = length(d);
    kern = zeros(nD,nd);
    for i = 1:nD
        for j = 1:nd
            kern(i,j) = evalkernel(D(i),d(j));
        end
    end
    kernel.D = D;
    kernel.d = d;
    kernel.kern = kern;
    save vdw.mat kernel v6
end

function phi = evalkernel(D,d)
    del = 0.1;
    M = 20;
    b = del/2:del:M;
    a = b';
    d1 = D*(1+d);
    d2 = D*(1d);
    int = 2/pi^2*a.^2*b.^2.*W(a,b)...
        .*T(nu(a,d1),nu(b,d1),nu(a,d2),nu(b,d2));
    int(isnan(int)) = 0;
    phi = sum(sum(int))*del^2;
end

function w = W(a,b)
    a2 = a.^2;
    b2 = b.^2;
    sa = sin(a);
    sb = sin(b);
    ca = cos(a);
    cb = cos(b);
    a2b2 = a2(:,ones(size(b)))+b2(ones(size(a)),:);
    w = 2*(((3a2).*sa)*(b.*cb)+(a.*ca)*((3b2).*sb)...
        +(a2b23).*(sa*sb)3*a.*ca*(b.*cb))./(a*b).^3;
end

function t = T(w,x,y,z)
    a = ones(size(w));
    b = ones(size(x));
```

```
    t = 1/2*(1./(w(:,b)+x(a,:))+1./(y(:,b)+z(a,:)))...
        .*(1./((w+y)*(x+z))+1./((w(:,b)+z(a,:)).*(y(:,b)+x(a,:))));
end

function z = nu(y,d)
    z = y.^2./(2*h(y/d));
end

function z = h(y)
    z = 1exp(4*pi/9*y.^2);
end
```

Listing A.11 | evaldensity.m

```
% evaluates density on the grid

function rho = evaldensity(grid,P)
    global param
    if isstruct(P)
        [rho.a,d.a] = evalsingledensity(grid,P.a);
        [rho.b,d.b] = evalsingledensity(grid,P.b);
        if param.maxder == 0, return, end
        rho.gab = d.a.x.*d.b.x+d.a.y.*d.b.y+d.a.z.*d.b.z;
        rho.gaa = rho.a.g;
        rho.gbb = rho.b.g;
        d = maketot(d);
        rho.g = d.x.^2+d.y.^2+d.z.^2;
        if param.maxder == 1, return, end
        rho.lap = d.xx+d.yy+d.zz;
        rho.gHg = d.xx.*d.x.^2+d.yy.*d.y.^2+d.zz.*d.z.^2 ...
            +2*(d.xy.*d.x.*d.y+d.xz.*d.x.*d.z+d.yz.*d.y.*d.z);
    else
        rho = evalsingledensity(grid,P);
    end
end

function tot = maketot(d)
    a = d.a;
    b = d.b;
    f = fieldnames(d.a);
    for i = 1:length(f)
        tot.(f{i}) = d.a.(f{i})+d.b.(f{i});
    end
end

function [rho,d] = evalsingledensity(grid,P)
    global param
    phiP = grid.f*P;
    rho.f = sum(phiP.*grid.f,2);
    if param.maxder == 0, d = []; return, end
    d.x = 2*sum(phiP.*grid.fx,2);
    d.y = 2*sum(phiP.*grid.fy,2);
```

```
    d.z = 2*sum(phiP.*grid.fz,2);
    rho.g = d.x.^2+d.y.^2+d.z.^2;
    if param.maxder == 1, return, end
    d.xx = 2*sum((grid.fx*P).*grid.fx,2)+2*sum(phiP.*grid.fxx,2);
    d.yy = 2*sum((grid.fy*P).*grid.fy,2)+2*sum(phiP.*grid.fyy,2);
    d.zz = 2*sum((grid.fz*P).*grid.fz,2)+2*sum(phiP.*grid.fzz,2);
    rho.lap = d.xx+d.yy+d.zz;
    d.xy = 2*sum((grid.fx*P).*grid.fy,2)+2*sum(phiP.*grid.fxy,2);
    d.xz = 2*sum((grid.fx*P).*grid.fz,2)+2*sum(phiP.*grid.fxz,2);
    d.yz = 2*sum((grid.fy*P).*grid.fz,2)+2*sum(phiP.*grid.fyz,2);
    rho.gHg = d.xx.*d.x.^2+d.yy.*d.y.^2+d.zz.*d.z.^2 ...
        +2*(d.xy.*d.x.*d.y+d.xz.*d.x.*d.z+d.yz.*d.y.*d.z);
end
```

Listing A.12 | vdw.m

```
% vdW kernel

function [kern,shift] = vdw(n1,n2,R)
    global kernel
    t = regexp(mfilename('fullpath'),'(.*)[\\/][^\\/]*','tokens');
    load([t{1}{1} filesep 'vdw.mat'],'kernel');
    q1 = calcq(n1);
    q2 = calcq(n2);
    q12D = repmat(q1,1,length(q2));
    q22D = repmat(q2',length(q1),1);
    q12 = q12D+q22D;
    D = R.*q12/2; D(isnan(D)) = 0;
    D(D<0.1) = 0.1;
    d = abs(q12Dq22D)./q12;
    d(q12==0) = 0;
    d(isnan(d)) = 1;
    clear q12D q22D q12
    kern = corekern(D,d);
    shift = 0;
end

function q = calcq(n)
    global param
    kF = (3*pi^2*n.f).^(1/3);
    gamma = param.zab/9;
    s2 = n.g./(2*kF.*n.f).^2;
    q = 4*pi/3*(pw92c(n)+dirac(n).*(1+gamma*s2));
end

function kern = corekern(D,d)
    global kernel
    D(isinf(D)) = 1e10;
    kernel.D(isinf(kernel.D)) = 1e100;
    id = d/0.05+1;
    iD = (11./(D/3+1))/0.001+1;
    kern = interp2(kernel.kern,id,iD);
```

```
end

function ind = sub2ind(siz,s1,s2)
    ind = (s21)*siz(1)+s1;
end
```

Listing A.13 | dirac.m

```
% Dirac exchange functional

function [eX,vX] = dirac(rho)
    vX = (3/pi)^(1/3)*rho.f.^(1/3);
    eX = 3/4*vX;
end
```

Listing A.14 | pw92c.m

```
% PerdewWang '92 correlation functional

function [eC,vC] = pw92c(rho)
    if isfield(rho,'a')
        n = rho.a.f+rho.b.f;
        rs = (3/(4*pi)./n).^(1/3);
        zeta = (rho.a.frho.b.f)./n;
        zeta(isnan(zeta)) = 1;
        [eC,vC] = eCunif(rs,zeta);
    else
        rs = (3/(4*pi)./rho.f).^(1/3);
        [eC,vC] = eCunif1(rs);
    end
end

function [eC,vC] = eCunif(rs,zeta)
    [ecrs0,decrs0] = G(rs,...
        0.031091,0.21370,7.5957,3.5876,1.6382,0.49294,1);
    [ecrs1,decrs1] = G(rs,...
        0.015545,0.20548,14.1189,6.1977,3.3662,0.62517,1);
    [acrs,dacrs] = G(rs,...
        0.016887,0.11125,10.357,3.6231,0.88026,0.49671,1);
    acrs = acrs;
    dacrs = dacrs;
    fpp0 = 1.709921;
    zeta1p = (1+zeta).^(1/3);
    zeta1m = (1zeta).^(1/3);
    fzeta = (zeta1p.*(1+zeta)+zeta1m.*(1zeta)2)/(2^(4/3)2);
    fpzeta = (4/3*(zeta1pzeta1m)/(2^(4/3)2));
    zeta4 = zeta.^4;
    eC = ecrs0+acrs.*fzeta/fpp0.*(1zeta4)+(ecrs1ecrs0).*fzeta.*zeta4;
    drsec = decrs0.*(1fzeta.*zeta4)+decrs1.*fzeta.*zeta4 ...
        +dacrs.*fzeta/fpp0.*(1zeta4);
    dzetaec = 4*zeta.^3.*fzeta.*(ecrs1ecrs0acrs/fpp0)...
        +fpzeta.*zeta4.*(ecrs1ecrs0+(1zeta4).*acrs/fpp0);
```

```
    tmp = eCrs/3.*drseczeta.*dzetaec;
    tmp(isnan(tmp)) = 0;
    vC.a = tmp+dzetaec;
    vC.b = tmpdzetaec;
end

function [eC,vC] = eCunif1(rs)
    [eC,drsec] = G(rs,...
        0.031091,0.21370,7.5957,3.5876,1.6382,0.49294,1);
    vC = eCrs/3.*drsec;
    vC(isnan(vC)) = 0;
end

function [y,dy] = G(rs,A,a1,b1,b2,b3,b4,p)
    sqrs = sqrt(rs);
    Q0 = 2*A*(1+a1*rs);
    Q1 = 2*A*(b1*sqrs+b2*rs+b3*sqrs.^3+b4*rs.^(p+1));
    Q1p = A*(b1./sqrs+2*b2+3*b3*sqrs+2*(p+1)*b4*rs.^p);
    log1p1oQ1 = log1p(1./Q1);
    y = Q0.*log1p1oQ1;
    y(isnan(y)) = 0;
    dy = 2*A*a1*log1p1oQ1Q0.*Q1p./(Q1.^2+Q1);
    dy(isnan(dy)) = 0;
end
```

Listing A.15 | pbec.m

```
% PBE correlation functional

function [eC,vC] = pbec(rho)
    if ~isfield(rho,'a')
        n = rho.f;
        phi = 1;
    else
        n = rho.a.f+rho.b.f;
        zeta = (rho.a.frho.b.f)./n;
        phi = ((1+zeta).^(2/3)+(1zeta).^(2/3))/2;
    end
    kF = (3*pi^2*n).^(1/3);
    ks = sqrt(4*kF/pi);
    t2 = rho.g./(2*phi.*ks.*n).^2;

    %gamma = (1log(2))/pi^2; % this is PBE
    %beta = 0.066725;
    gamma = 0.031092193703936; % this is Molpro
    beta = 2.146079601698033*gamma;

    [eCunif,vCunif] = pw92c(rho);

    gf3 = gamma*phi.^3;
    bg = beta/gamma;
    S = exp(eCunif./gf3);
```

```
    A = bg./(S1);
    At2 = A.*t2;
    At2quad = 1+At2+At2.^2;
    Q = (1+At2)./At2quad;
    bgt2Q = bg*t2.*Q;
    H = gf3.*log(1+bgt2Q);
    eC = eCunif+H;
    eC(isnan(eC)|isinf(eC)) = 0;

    g = rho.g;
    R = At2.*(2+At2)./At2quad.^2;
    RAt2Q = R.*At2./Q;
    x1xbgt2Q = bgt2Q./(1+bgt2Q);
    V = (At2.^3+3*At2.^23*At24)./At2quad.^3;
    W = At2.*(At2.*VR)./(At2.*RQ);
    dgH = gf3.*x1xbgt2Q./g.*(1RAt2Q);
    dgdgH = dgH./g.*(x1xbgt2Q.*(RAt2Q1)W);
    if ~isfield(rho,'a')
        SS173 = S./(S1).*(vCunifeCunif)./gf37/3;
        dnH = gf3.*x1xbgt2Q./n.*(7/3+RAt2Q.*SS173);
        dndgH = dgH./n.*(x1xbgt2Q.*(7/3+RAt2Q.*SS173)7/3W.*SS173);
        vC = vCunif+(H2*dgH.*g)+n.*(dnH2*dndgH.*g...
            2*dgdgH.*(2*rho.gHg)2*dgH.*rho.lap);
        vC(isnan(vC)|isinf(vC)) = 0;
    else
        f1fz.a = ((1zeta).*(1+zeta).^(1/3)(1zeta).^(2/3))./phi/3;
        f1fz.b = ((1+zeta).*(1zeta).^(1/3)(1+zeta).^(2/3))./phi/3;
        f1fz.a(zeta==1) = 0;
        f1fz.b(zeta==1) = 0;
        f73.a = 7/3+2*f1fz.a;
        f73.b = 7/3+2*f1fz.b;
        SS173.a = S./(S1).*(vCunif.aeCunif.*(1+3*f1fz.a))./gf3f73.a;
        SS173.b = S./(S1).*(vCunif.beCunif.*(1+3*f1fz.b))./gf3f73.b;
        dnaH = 3*f1fz.a./n.*H...
            gf3.*x1xbgt2Q./n.*(f73.a+RAt2Q.*SS173.a);
        dnbH = 3*f1fz.b./n.*H...
            gf3.*x1xbgt2Q./n.*(f73.b+RAt2Q.*SS173.b);
        dnadgH = dgH./n...
            .*(3*f1fz.a+x1xbgt2Q.*(f73.a+RAt2Q.*SS173.a)f73.aW.*SS173.a);
        dnbdgH = dgH./n...
            .*(3*f1fz.b+x1xbgt2Q.*(f73.b+RAt2Q.*SS173.b)f73.bW.*SS173.b);
        tmp = (H2*dgH.*g)+n.*(2*dgH.*rho.lap2*dgdgH.*(2*rho.gHg)...
            2*dnadgH.*(rho.gaa+rho.gab)2*dnbdgH.*(rho.gbb+rho.gab));
        vC.a = vCunif.a+n.*dnaH+tmp;
        vC.b = vCunif.b+n.*dnbH+tmp;
        vC.a(isnan(vC.a)|isinf(vC.a)) = 0;
        vC.b(isnan(vC.b)|isinf(vC.b)) = 0;
    end
end
```

Listing A.16 | genindexing1.m

```
% generates indexing arrays for 1elintegral recursion formulas
% [l+1,anc(l,k,i)] is (l,k)+1_i
% [l1,des(l,k,i)] is (l,k)1_i
% [ni(l,k,1),ni(l,k,2),ni(l,k,3)] is (l,k)

function genindexing1(n)
    global anc des ni
    one = eye(3);
    for i = 1:n
        ns{i} = genns(i1);
    end
    anc = zeros(n,size(ns{end},1),3);
    des = zeros(n1,size(ns{end1},1),3);
    ni = zeros(n,size(ns{end},1),3);
    anc(1,1,:) = zeros(1,3);
    des(1,1,:) = 1:3;
    for i = 2:n
        for j = 1:size(ns{i},1)
            for k = 1:3
                s = find(sum(...
                    repmat(ns{i}(j,:)one(k,:),size(ns{i1},1),1)==ns{i1}...
                    ,2)==3);
                if isempty(s)
                    s = 0;
                end
                anc(i,j,k) = s;
                if i < n
                    s = find(sum(...
                        repmat(ns{i}(j,:)+one(k,:),size(ns{i+1},1),1)==ns{i+1}...
                        ,2)==3);
                    des(i,j,k) = s;
                end
                ni(i,j,k) = ns{i}(j,k);
            end
        end
    end
end
```

Listing A.17 | genns.m

```
% generates list of cartesian polynomials for given l

function ns = genns(l)
    N = lton(l);
    ns = zeros(N,3);
    n = [l 0 0];
    for j = 1:N
        ns(j,:) = n;
        if n(2) > 0
            n(2:3) = n(2:3)+[1 1];
        else
```

```
            n = [n(1)1 ln(1)+1 0];
        end
    end
end
```

Listing A.18 | genns2.m

```
function n = genns2(l)
    switch l
        case 0
            n = zeros(1,3);
        case 1
            n = eye(3);
        case 2
            n = extract({'xx' 'yy' 'zz' 'xy' 'xz' 'yz'});
        case 3
            n = extract({'xxx' 'yyy' 'zzz' 'xyy' 'xxy' 'xxz'...
        'xzz' 'yzz' 'yyz' 'xyz'});
        case 4
            n = extract({'xxxx' 'yyyy' 'zzzz' 'xxxy' 'xxxz'...
                'yyyx' 'yyyz' 'zzzx' 'zzzy' 'xxyy' 'xxzz'...
                'yyzz' 'xxyz' 'yyxz' 'zzxy'});
        otherwise
            error('Molden doesn''t support l > 4');
    end
end

function n = extract(s)
    n = [];
    t = 'xyz';
    for i = 1:length(s)
        m = zeros(1,3);
        for j = 1:3
            m(j) = sum(ismember(s{i},t(j)));
        end
        n = [n; m];
    end
end
```

Listing A.19 | car2sphr.m

```
function coeff = car2sphr(l, fgenns, type)
    switch l
        case 0
            coeff = 1;
        case 1
            coeff = eye(3);
        otherwise
            ls = fgenns(l);
            coeff = zeros(2*l+1, size(ls, 1));
            for i = 1:size(coeff, 1)
                m = i1l;
```

```
                for j = 1:size(coeff, 2)
                    coeff(i, j) = c(l, m, ls(j, 1), ls(j, 2), ls(j, 3));
                end
            end
            if nargin > 2 && strcmp(type, 'real')
                A = zeros(2*l+1);
                A(1, l+1) = 1;
                for i = 1:l
                    A(2*i, l+1i) = 1/sqrt(2);
                    A(2*i, l+1+i) = 1/sqrt(2);
                    A(2*i+1, l+1i) = 1/sqrt(2);
                    A(2*i+1, l+1+i) = 1/sqrt(2);
                end
                coeff = A * coeff;
                coeff = real(coeff) + imag(coeff);
            end
    end
end

function coeff = c(l, m, lx, ly, lz)
    j = (lx+lyabs(m)) / 2;
    if j ~= round(j)
        coeff = 0;
        return
    end
    if m < 0
        pm = 1;
    else
        pm = 1;
    end
    A = factorial(2*lx) * factorial(2*ly) * factorial(2*lz)...
            * factorial(l) * factorial(labs(m))...
        / (factorial(2*l) * factorial(lx) * factorial(ly)...
            * factorial(lz) * factorial(l+abs(m)));
    B = 0;
    for i = 0:(labs(m))/2
        C = binom(l, i)*binom(i, j) * (1)^i...
            * factorial(2*l2*i) / factorial(labs(m)2*i);
        D = 0;
        for k = 0:j
            E = binom(j, k) * binom(abs(m), lx2*k)...
                * (1)^(pm*(abs(m)lx+2*k)/2);
            D = D + E;
        end
        B = B + C * D;
    end
    coeff = sqrt(A) * 1/(2^l*factorial(l)) * B;
end

function x = binom(n, k)
    if k < 0 || k > n
        x = 0;
```

```
        return
    else
        x = nchoosek(n, k);
    end
end
```

Listing A.20 | element.m

```
% converts element symbol on atomic number or vice versa. 12/04/07

function n = element(s)
    table = {'H' 'He' 'Li' 'Be' 'B' 'C' 'N' 'O' 'F' 'Ne'...
        'Na' 'Mg' 'Al' 'Si' 'P' 'S' 'Cl' 'Ar' 'K'...
        'Ca' 'Sc' 'Ti' 'V' 'Cr' 'Mn' 'Fe' 'Co' 'Ni'...
        'Cu' 'Zn' 'Ga' 'Ge' 'As' 'Se' 'Br' 'Kr' 'Rb'...
        'Sr' 'Y' 'Zr' 'Nb' 'Mo' 'Tc' 'Ru' 'Rh' 'Pd'...
        'Ag' 'Cd' 'In' 'Sn' 'Sb' 'Te' 'I' 'Xe' 'Cs'...
        'Ba' 'La' 'Ce' 'Pr' 'Nd' 'Pm' 'Sm' 'Eu' 'Gd'...
        'Tb' 'Dy' 'Ho' 'Er' 'Tm' 'Yb' 'Lu' 'Hf' 'Ta' 'W'...
        'Re' 'Os' 'Ir' 'Pt' 'Au' 'Hg' 'Tl' 'Pb' 'Bi'...
        'Po' 'At' 'Rn' 'Fr' 'Ra' 'Ac' 'Th' 'Pa' 'U'};
    if isnumeric(s)
        n = table{s};
    else
        for n = 1:length(table)
            if strcmpi(s,table{n}), return, end
        end
        error('Unknown element symbol');
    end
end
```

Listing A.21 | lton.m

```
% gives number of cartesian spherical harmonics

function N = lton(l)
    N = (l+1).*(l+2)/2;
end
```

Listing A.22 | radius.m

```
% returns various atomic radii

function r = radius(n,type)
    bohr = 1.889725989;
    switch type
        case 'atom'
            data = bohr*[0.31 0.28 1.28 0.96 0.84 0.74 ...
                0.71 0.66 0.57 0.58];
            data(14) = bohr*1.0;
            data(18) = bohr*0.85;
            data(36) = bohr*1.15;
```

```
        case 'grid'
            data = [1 0.5882 3.0769 2.0513 1.5385 1.2308 ...
        1.0256 0.8791 0.7692 0.6838];
            data(14) = 1.6;
            data(18) = 1.3333;
            data(36) = 1.9;
    end
    nmax = length(data);
    if n <= nmax
        r = data(n);
        if r > 0, return, end
    end
    error('Radius "%s" not defined for Z=%i',type,n);
end
```