UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Characterizing water-metal interfaces and machine learning potential energy surfaces

by

Kevin Ryczko

A thesis submitted in partial fulfillment for the degree of Master of Science

in the

Faculty of Science

August 2017

Abstract

In this thesis, we first discuss the fundamentals of *ab initio* electronic structure theory and density functional theory (DFT). We also discuss statistics related to computing thermodynamic averages of molecular dynamics (MD). We then use this theory to analyze and compare the structural, dynamical, and electronic properties of liquid water next to prototypical metals including platinum, graphite, and graphene. Our results are built on Born-Oppenheimer molecular dynamics (BOMD) generated using density functional theory (DFT) which explicitly include van der Waals (vdW) interactions within a first principles approach. All calculations reported use large simulation cells, allowing for an accurate treatment of the water-electrode interfaces. We have included vdW interactions through the use of the optB86b-vdW exchange correlation functional. Comparisons with the Perdew-Burke-Ernzerhof (PBE) exchange correlation functional are also shown. We find an initial peak, due to chemisorption, in the density profile of the liquid water-Pt interface not seen in the liquid water-graphite interface, liquid water-graphene interface, nor interfaces studied previously. To further investigate this chemisorption peak, we also report differences in the electronic structure of single water molecules on both Pt and graphite surfaces. We find that a covalent bond forms between the single water molecule and the platinum surface, but not between the single water molecule and the graphite surface. We also discuss the effects that defects and dopants in the graphite and graphene surfaces have on the structure and dynamics of liquid water. Lastly, we introduce artificial neural networks (ANNs), and demonstrate how they can be used to machine learn electronic

structure calculations. As a proof of principle, we show the success of an ANN potential energy surfaces for a dimer molecule with a Lennard-Jones potential.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Examining the molecular structure of water-solid interfaces is crucial for understanding dynamical processes that occur in both natural and controlled environments. Hydrogen production through the splitting of water at a metal surface is of great interest for solar cell devices [3], and generating hydrogen and oxygen gas has been proposed as a means to store energy [4]. Despite significant efforts both computationally [5–18] and experimentally [16, 19–21] to study electrode-water interfaces, to-date, there has yet to be an explicit treatment of liquid water next to a realistic, catalytic surface computed at the level of accurate, first principles molecular dynamics. Almost 10 years ago, Cicero *et al.* [5] studied several liquid water-graphene interfaces as well as water confined in carbon nanotubes with differing radii using DFT with the PBE exchange correlation functional [22]. Despite the differences of the simulation cell sizes, the interfacial water had a similar structure when comparing the density profiles of the different supercells. Moving away from the graphene at a perpendicular direction, a zero particle density was

found for ≈ 2.5 Å. Further past this volume of zero particle density, away from the surface, an initial spike was seen where the particle density was significantly larger than the density of bulk water in ambient conditions. In Figure 1.1, we show a prototypical plot of the density profile next to a metal. This plot has a similar structure as the density profile of the liquid water-graphene interface. In a subsequent study by Liu *et al.* [23], a liquid water-salt interface was examined using the same theoretical framework. Here, the structure of the interfacial layer was quite different. Despite having a similar zero particle density adjacent to the surface, the first peak in the density was lower, and more broad. Even though graphite has an inferior efficiency for water splitting when compared to platinum, it is commonly used in laboratories to study electrochemical processes because it is inexpensive and abundant. By studying the structural and electronic properties of liquid water-solid interfaces through *ab initio* methods, we can begin to understand and characterize electrodes. This aids in the selection process of a new, abundant, and efficient alternative electrode to superior rare metals.

More recently, another *ab initio* study was done by Velasco *et al.* [18] on a liquid water-Au interface. Here, the computational efforts were exceptional; 80 ps of MD trajectories were generated using DFT with the PBE exchange correlation functional. The molecular density profile of this interface was similar to the liquid water-graphene interface; the only structural difference was an additional second peak ≈ 5 Å past the initial, most prominent peak. It should also be noted that the density of the initial, largest peak reached a value of 4 g/cc; this is approximately twice the value of the largest peak seen for the water-graphene interface. Thus,

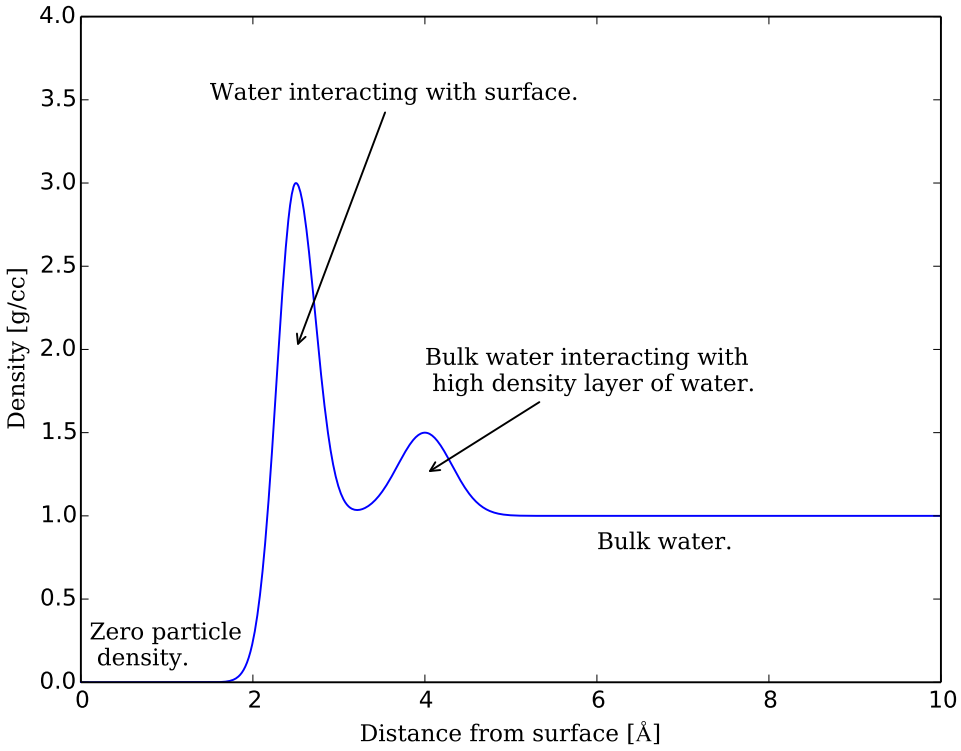

FIGURE 1.1: Prototypical density profile of liquid water next to a metal surface in ambient conditions.

the water shows more attraction to the gold surface than the graphene, but it may not be a key indicator of an efficient catalyst. Gold is known to be chemically inert [24]. When a molecule (H_2 for example) binds to the surface of gold, the energies associated with the anti-bonding molecular orbitals lie below the Fermi level. This results in occupancy of the anti-bonding molecular orbitals, causing repulsion and reducing surface activity.

The liquid water-solid interfaces considered in our work are platinum, graphite, and graphene. platinum is a rare metal, and is known to have a low over potential for water splitting, whereas carbon-based materials are less efficient. The over potential of a metal is a clear indicator of efficiency. To split water, there is a reaction energy barrier to overcome in order for the process to occur. For different metals,

there is an additional energy cost, and this is what is called the over potential of a metal. Due to the efficiency, rarity, and cost of platinum, there is an active area of research focused on finding an alternative, abundant metal/semi-metal with similar properties [23, 25–28]. Many theoretical and experimental studies have been done to further understand the water-platinum interface [29–36]. In particular, Osawa *et al.* [31] performed infrared absorption spectroscopy experiments to examine the structure of water next to an electrified, pristine platinum electrode. They found significant differences in the spectrum when comparing the interfacial water to that of the bulk. They concluded that the differences in the spectrum were due to the orientation of the surface water molecules. At the positively charged electrode, the surface molecules formed a strong hydrogen-bonding network parallel to the surface. At the negatively charged electrode, the interfacial water formed a much weaker hydrogen bonding network and the water molecules preferred an H-down orientation to the surface. The majority of computational studies regarding the water-platinum interface have relied on the use of classical force fields (i.e. empirical fits coupled with molecular dynamics or Monte Carlo engines). They have reported statistical distributions to understand the structure of water next to the surface [29, 33, 37], such as the molecular density profile, orientation probability distributions, and two-dimensional probability distributions outlining adsorption sites. More recent studies have used experimental and *ab initio* techniques to study this interface [38–40], but for smaller systems which only included a surface layer (excluding bulk water). In this report, we compare our results to previous water-solid studies and build on previous classical reports of the liquid water-platinum interface with a quantum description of the forces

which define this interface. A larger system size allows for dynamics in the liquid that would occur naturally in ambient conditions; the natural dynamics we encapsulate in our simulations are absent in previous simulations that only consider the interfacial surface layer of water.

In recent years, better computer hardware/software and the accumulation of data worldwide has allowed for significant progress in machine learning and artificial intelligence. The majority of the success of machine learning cover topics related to computer vision [41–43], or automated speech recognition [44, 45]. In recent years, the electronic structure community has been experimenting with combining traditional methods with machine learning [2, 46–48], and the results look promising. Although DFT is one of the cheapest *ab initio* methods to calculate the electronic structure of some system for atomistic simulations, it gets to be rather computationally taxing when the system size consists of a few hundred atoms. The use of machine learning allows for much faster calculations, and rivals the accuracy of the electronic method it is being trained on. The use of machine learned potentials is relatively new, and requires thorough testing if to be heavily relied on. In the work done by Behler *et al.* [48], they developed a framework to machine learn total energies and forces of atomistic systems. Specifically, they proposed training high dimensional neural network potentials with symmetry functions that represent the local environments of atoms. This allows one to map the output of traditional atomistic simulations to the symmetry functions, which could then be fed in as input to the ANN. Shortly thereafter, Artrith *et al.* [2] implemented this framework to study TiO_2. Here, they discussed the software that was developed

(aenet), their methodology for generating the ANN potential, and the accuracy of the ANN potential. Aenet is software that can produce an ANN potential for a general atomistic system and we leverage this software in Chapter 5. Aenet is incredibly useful, but the accuracy of the ANN potential depends critically on the training set, which requires intuition to generate. In [2], they applied a recursive training technique to optimize their ANN potential. In their preliminary ANN potential, their model was trained from atomic configurations of pristine crystalline phases of TiO_2, compressed and expanded crystalline TiO_2, and removed O atoms creating defects in the crystal. To refine the model, they then used MD to generate new configurations, and self-consistently trained the model on the configurations until the difference between the true energy and the predicted energies was small. After training, they found that the lattice parameters, bulk moduli, and total energies were in excellent agreement with reference values calculated from DFT.

This thesis contains 6 chapters. In Chapter 2, we discuss electronic structure theory related to DFT which is the basis for all calculations. In Chapter 3, we then describe how to compute thermodynamic averages of atomic systems from MD. We also give Python code to explicitly show how to compute these averages. In Chapter 4, we present our main results. This includes analyzing BOMD of liquid water next to prototypical metals, including platinum, and graphitic surfaces with and without defects and dopant materials for insight into water splitting from a nanoscale perspective. This also includes examining the electronic structure of a water monomer on the surface of platinum and graphite. Some of the text in this chapter is in a manuscript that is currently under review. In Chapter 5, we

describe the basic theory of ANNs, and then use ANNs for a simple atomic system which consists of a dimer molecule. Lastly, in Chapter 6, we conclude, and discuss future work with liquid water-metal interfaces and machine learning.

Chapter 2

DFT

2.1 Fundamentals

DFT is one of the most widely used tools to study atomistic systems. It is the modern workhorse for electronic structure calculations. Together, with the ongoing improvements in algorithms and computation, the atomistic system sizes of study has grown significantly. Many of the systems presented in this thesis push the limits of modern computer architecture and software. In the 1960's Hohenberg and Kohn [49], and Kohn and Sham [50] published revolutionary papers that allowed for the exploration of materials using quantum mechanics. Consider a system of n electrons and N ions. The non-relativistic Hamilitonian of the system

is

$$\begin{aligned} H &= -\sum_i^n \frac{\hbar^2}{2m_e}\nabla_i^2 - \sum_k^N \frac{\hbar^2}{2m_k}\nabla_k^2 - \frac{e^2}{4\pi\epsilon_0}\sum_i^n\sum_k^N \frac{Z_k}{|\mathbf{r}_i - \mathbf{R}_k|} \\ &+ \frac{e^2}{8\pi\epsilon_0}\sum_i^n\sum_j^n{}_{i\neq j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \frac{e^2}{8\pi\epsilon_0}\sum_k^N\sum_l^N{}_{k\neq l} \frac{Z_k Z_l}{|\mathbf{R}_k - \mathbf{R}_l|}. \end{aligned} \tag{2.1}$$

The first two terms are the kinetic energies of the electrons and nuclei, respectively. The last three terms are the Coulomb interactions between the electrons and nuclei. We then adopt atomic units, where the units of energy are in Hartree ($E_\mathrm{H} = \frac{\hbar^2}{m_e a_0^2} = \frac{e^2}{4\pi\epsilon_0}$) and distances in Bohr ($a_0 = \frac{4\pi\epsilon_0\hbar^2}{m_e e^2}$). This gives the scaled Hamiltonian

$$\begin{aligned} H &= -\sum_i^n \frac{1}{2}\nabla_i^2 - \sum_k^N \frac{1}{2m_k}\nabla_k^2 - \sum_i^n\sum_k^N \frac{Z_k}{|\mathbf{r}_i - \mathbf{R}_k|} \\ &+ \frac{1}{2}\sum_i^n\sum_j^n{}_{i\neq j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \frac{1}{2}\sum_k^N\sum_l^N{}_{k\neq l} \frac{Z_k Z_l}{|\mathbf{R}_k - \mathbf{R}_l|}. \end{aligned} \tag{2.2}$$

The first approximation we make is the Born-Oppenheimer approximation (or the adiabatic approximation). Due to lower mass, the electrons move adiabatically in the presence of nuclei. This allows us to decouple the electronic and ionic degrees of freedom. With this approximation, we get the electronic Hamiltonian

$$H_\mathrm{electronic} = -\sum_i^n \frac{1}{2}\nabla_i^2 - \sum_i^n\sum_k^N \frac{Z_k}{|\mathbf{r}_i - \mathbf{R}_k|} + \frac{1}{2}\sum_i^n\sum_j^n{}_{i\neq j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}. \tag{2.3}$$

The last term in equation 2.3 is problematic. For a 2 electron system, we can not solve the time independant Schrödinger equation analytically[1]. Computationally, we would need to solve a 6 (3 spatial dimensions per electron) dimensional eigenvalue problem. If we discretized each dimension on a grid with 10 points, we would need to store 10^6 floating point numbers for the wave function, and 10^{12} for our Hamiltonian. One can see that approaching this problem with brute force quickly becomes unfeasible.

In the paper titled, "Inhomogeneous Electron Gas", by Hohenberg and Kohn [49], they approached solving equation 2.3 from another perspective. They suggested to consider the electron density, $\rho(\mathbf{r})$, rather than the wave function $\Psi(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n)$. Using the electron density, they went on to show that the total energy of the system can be expressed in terms of the electron density alone. In addition, they proved the following 2 theorems:

Theorem 2.1. *The external potential, and the total energy are unique functionals of the electron density.*

Theorem 2.2. *The ground state energy can be calculated by finding the global minimizing electron density in the total energy functional.*

To prove the first theorem, consider two external potentials $V_{\text{ext}}^{(1)}(\mathbf{r})$ and $V_{\text{ext}}^{(2)}(\mathbf{r})$ which differ by more than just a constant value and give the same ground state electron density $\rho(\mathbf{r})$. These external potentials lead to Hamiltonians $H^{(1)}$, $H^{(2)}$,

[1] We can find solutions to the time dependent Schrödinger equation by solving the eigenvalue problem $H\psi = E\psi$ and then tacking on the usual time dependance $\phi(t) = e^{-iEt/\hbar}$. This can be done because our Hamiltonian is not time dependant.

and ground state wave functions $\Psi^{(1)}$, $\Psi^{(2)}$. Due to the variational principle,

$$E^{(1)} = \langle \Psi^{(1)} | H^{(1)} | \Psi^{(1)} \rangle < \langle \Psi^{(2)} | H^{(1)} | \Psi^{(2)} \rangle. \tag{2.4}$$

If we write the last term of Equation 2.4 as

$$\langle \Psi^{(2)} | H^{(1)} | \Psi^{(2)} \rangle = \langle \Psi^{(2)} | H^{(2)} | \Psi^{(2)} \rangle + \langle \Psi^{(2)} | H^{(1)} - H^{(2)} | \Psi^{(2)} \rangle, \tag{2.5}$$

then we find

$$\begin{aligned} \langle \Psi^{(2)} | H^{(1)} | \Psi^{(2)} \rangle &= \langle \Psi^{(2)} | H^{(2)} | \Psi^{(2)} \rangle + \langle \Psi^{(2)} | H^{(1)} - H^{(2)} | \Psi^{(2)} \rangle \\ &= E^{(2)} + \int [V_{\text{ext}}^{(1)}(\mathbf{r}) - V_{\text{ext}}^{(2)}(\mathbf{r})] \rho(\mathbf{r}) d\mathbf{r} \\ E^{(1)} &< E^{(2)} + \int [V_{\text{ext}}^{(1)}(\mathbf{r}) - V_{\text{ext}}^{(2)}(\mathbf{r})] \rho(\mathbf{r}) d\mathbf{r}. \end{aligned} \tag{2.6}$$

Here, we have expressed the energy from the external potential as shown in Equation 2.10. If we then repeat this process, with Equation 2.4 as

$$E^{(2)} = \langle \Psi^{(2)} | H^{(2)} | \Psi^{(2)} \rangle < \langle \Psi^{(1)} | H^{(2)} | \Psi^{(1)} \rangle, \tag{2.7}$$

then we arrive at the condition $E^{(1)} + E^{(2)} < E^{(1)} + E^{(2)}$, which can't be true. Therefore there does not exist two external potentials that differ by more than a constant which give the same electron density.

To prove theorem 2, again consider 2 Hamiltonians $H^{(1)}$, $H^{(2)}$, with ground state wave functions $\Psi^{(1)}$, $\Psi^{(2)}$, and electron densities $\rho^{(1)}(\mathbf{r})$, $\rho^{(2)}(\mathbf{r})$. From Equation

2.11,

$$E[\rho^{(1)}(\mathbf{r})] = \langle \Psi^{(1)}|H^{(1)}|\Psi^{(1)}\rangle < \langle \Psi^{(2)}|H^{(1)}|\Psi^{(2)}\rangle = E[\rho^{(2)}(\mathbf{r})], \tag{2.8}$$

again, due to the variational principle. From this, we find that only the true ground state electron density will give the ground state energy. Any other electron density will have greater energy.

To express the total energy as a functional of the electron density, assume we have solved equation 2.3, giving us $\Psi(\mathbf{r}, \mathbf{r}_2, \ldots, \mathbf{r}_n)$. The electron density is found via

$$\begin{aligned}
\rho(\mathbf{r}) &= \int \ldots \int |\Psi(\mathbf{r}, \mathbf{r}_2, \ldots, \mathbf{r}_n)|^2 d\mathbf{r}_2 d\mathbf{r}_3 \ldots d\mathbf{r}_n \\
&\quad + \int \ldots \int |\Psi(\mathbf{r}_1, \mathbf{r}, \ldots, \mathbf{r}_n)|^2 d\mathbf{r}_1 d\mathbf{r}_3 \ldots d\mathbf{r}_n + \ldots \\
&\quad + \int \ldots \int |\Psi(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r})|^2 d\mathbf{r}_1 d\mathbf{r}_2 \ldots d\mathbf{r}_{n-1} \\
&= \int \ldots \int [\delta(\mathbf{r} - \mathbf{r}_1) + \delta(\mathbf{r} - \mathbf{r}_2) + \ldots + \delta(\mathbf{r} - \mathbf{r}_n)] \\
&\quad |\Psi(\mathbf{r_1}, \mathbf{r}_2, \ldots, \mathbf{r}_n)|^2 d\mathbf{r}_1 d\mathbf{r}_2 \ldots d\mathbf{r}_n \\
&= N \int \ldots \int \Psi^*(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n) \delta(\mathbf{r} - \mathbf{r}_1) \Psi(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n) \\
&\quad d\mathbf{r}_1 d\mathbf{r}_2 \ldots d\mathbf{r}_n \\
&= N \int \ldots \int |\Psi(\mathbf{r}, \mathbf{r}_2, \ldots, \mathbf{r}_n)|^2 d\mathbf{r}_2 d\mathbf{r}_3 \ldots d\mathbf{r}_n.
\end{aligned}$$

The factor of N comes from the antisymmetric properties of the wave function, i.e. $\Psi(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n) = -\Psi(\mathbf{r}_2, \mathbf{r}_1, \ldots, \mathbf{r}_n)$. Hohenberg and Kohn state that since $\rho(\mathbf{r})$ is a functional of $\Psi(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n)$, then so too is the kinetic and elecron-electron interaction energy. Explicitly, there is some universal functional F

$$F[\rho(\mathbf{r})] \equiv \langle \Psi|\hat{T} + \hat{U}|\Psi\rangle \tag{2.9}$$

where $\hat{T}$ is the sum over the kinetic energy operators, and $\hat{U}$ the sum over the electron-electron interaction operators. The external potential, can explicitly be written in terms of the electron density,

$$\begin{aligned}
\langle \hat{V} \rangle &= -\sum_{i}^{n}\sum_{k}^{N}\int\ldots\int \Psi^*(\mathbf{r}_1,\mathbf{r}_2,\ldots,\mathbf{r}_n)\frac{Z_k}{|\mathbf{r}_i-\mathbf{R}_k|}\Psi(\mathbf{r}_1,\mathbf{r}_2,\ldots,\mathbf{r}_n) \\
&\quad d\mathbf{r}_1 d\mathbf{r}_2\ldots d\mathbf{r}_n \\
&= \sum_{i}^{n}\int\ldots\int v(\mathbf{r_i})|\Psi(\mathbf{r}_1,\mathbf{r}_2,\ldots,\mathbf{r}_n)|^2 d\mathbf{r}_1 d\mathbf{r}_2\ldots d\mathbf{r}_n \\
&= N\int v(\mathbf{r}_n)\left[\int\ldots\int |\Psi(\mathbf{r}_1,\mathbf{r}_2,\ldots,\mathbf{r}_n)|^2 d\mathbf{r}_1 d\mathbf{r}_2\ldots d\mathbf{r}_{n-1}\right] d\mathbf{r}_n \\
&= \int v(\mathbf{r})\rho(\mathbf{r})d\mathbf{r}, \qquad (2.10)
\end{aligned}$$

which leads to the total energy functional

$$E[\rho(\mathbf{r})] = \int v(\mathbf{r})\rho(\mathbf{r})d\mathbf{r} + F[\rho(\mathbf{r})]. \qquad (2.11)$$

Minimization of this energy functional leads to the ground state charge density and energy of a many body system.

2.2 Kohn-Sham DFT

In the work done by Hohenberg and Kohn [49], they showed that one can solve for the electron density of a many body system by finding the minima of the total energy functional, but did not state the exact form of the universal functional F. The electron-electron interaction makes solving the Schrödinger equation with

the Hamiltonian in Equation 2.3 impossible, and both the kinetic and electron-electron interaction energy can not be explicitly written down in terms of the electron density. Following [49], Kohn and Sham [50] provided a method which overcame these problems in a paper titled: "Self-Consistent Equations Including Exchange and Correlation Effects". Here, they created an auxiliary system of non-interacting electrons, and leveraged the exact density functional formalism. Recall the universal functional F in equation 2.11. This included the problematic terms when dealing with interacting electrons. To overcome these terms, Kohn and Sham defined the total energy as

$$E[\rho(\mathbf{r})] = \int v(\mathbf{r})\rho(\mathbf{r})d\mathbf{r} + \frac{1}{2}\int \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}d\mathbf{r}' + G[\rho(\mathbf{r})], \tag{2.12}$$

where the second term is an averaged interaction energy of the electrons (deemed the Hartee energy), and G is defined as

$$G[\rho(\mathbf{r})] \equiv T_{\mathrm{ni}}[\rho(\mathbf{r})] + E_{\mathrm{xc}}[\rho(\mathbf{r})]. \tag{2.13}$$

Here, T_{ni} is the kinetic energy for a system of non-interacting electrons, and E_{xc} is the exchange and correlation energy of an interacting system of electrons. The one caveat is that the exact form for E_{xc} is unknown. Even though the exact form is unknown, it is expected to hold the effects from correlated electrons, exchange from the antisymmetric properties of the many body wave function, and the energy difference between the true and the non-interacting kinetic energy. Hence the term "exchange correlation" functional.

To minimize the total energy, we use the method of Lagrange multipliers, subject to the constraint that the number of electrons does not change, i.e.

$$n = \int d\mathbf{r}\rho(\mathbf{r}). \tag{2.14}$$

Doing this yields

$$H[\rho(\mathbf{r})] = E[\rho(\mathbf{r})] - \mu\left(\int d\mathbf{r}\rho(\mathbf{r}) - n\right), \tag{2.15}$$

where μ is the Lagrange multiplier. To minimize this expression with respect to the density, we take the functional derivative of H with respect to ρ, and set it equal to 0. This gives,

$$\frac{\delta H}{\delta\rho(\mathbf{r})} = v(r) + \int \frac{\rho(\mathbf{r})}{|\mathbf{r}-\mathbf{r}'|} + \frac{\delta T_{\mathrm{ni}}}{\delta\rho(\mathbf{r})} + \frac{\delta E_{\mathrm{xc}}}{\delta\rho(\mathbf{r})} - \mu = 0. \tag{2.16}$$

Multiplying both sides (from the right) by ψ, we get

$$\left(v(r) + \int \frac{\rho(\mathbf{r})}{|\mathbf{r}-\mathbf{r}'|} + \frac{\delta T_{\mathrm{ni}}}{\delta\rho(\mathbf{r})} + \frac{\delta E_{\mathrm{xc}}}{\delta\rho(\mathbf{r})}\right)\psi = \mu\psi. \tag{2.17}$$

which is just a Schrödinger equation (with $\epsilon \equiv \mu$). As we will see below, the electron density ρ will be built from ψ, and therefore solving for ψ gives ρ, and the total energy. The electron density is constructed by summing over the occupied orbitals

$$\rho(\mathbf{r}) = \sum_{i=1}^{n} |\psi_i(\mathbf{r})|^2 \tag{2.18}$$

where we would have 2 electrons per orbital (to account for spin). To solve for the orbitals, we need to solve the eigenvalue problem,

$$\left(-\frac{1}{2}\nabla^2 - \sum_I^N \frac{Z_I}{|\mathbf{r}-\mathbf{R}_I|} + \int \frac{\rho(\mathbf{r})}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r} + V_{\text{xc}}(\rho(\mathbf{r}))\right)\psi_i(\mathbf{r}) = \epsilon_i\psi_i(\mathbf{r}). \tag{2.19}$$

The first term is the one-electron kinetic energy, the second term is the coulomb potential due to the ions, the third term is the Hartee potential, and the last term is the exchange correlation potential. The Hartee potential is a 'mean field' approximation for the electron-electron interactions. Instead of the electron interacting with every other electron independently, it interacts with the electron density (which includes itself!). The exchange correlation potential is the functional derivative of the exchange correlation energy with respect to the electron density, i.e.

$$V_{\text{xc}}(\rho(\mathbf{r})) = \frac{\delta}{\delta\rho}E_{\text{xc}}[\rho(\mathbf{r})]. \tag{2.20}$$

The total energy of the system is then

$$E = \sum_i^n \epsilon_i - \frac{1}{2}\int\int \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}d\mathbf{r}' + E_{\text{xc}}[\rho(\mathbf{r})] - \int \rho(\mathbf{r})V_{\text{xc}}(\rho(\mathbf{r}))d\mathbf{r}. \tag{2.21}$$

To write down the kinetic energy of the system, we take the sum over the occupied orbitals and subtract off all other potential energies. After doing this, we end up with Equation 2.21.

From equation 2.19, we can see that we need an electron density to solve for the set of electron orbitals. The calculation for the total energy of the system given from equation 2.21 is an iterative process. Once makes an initial "guess" for the

orbitals, constructs the electron density, solves for new orbitals, calculates the total energy based on the new electron density (which is the sum over the new orbitals), and repeats until we reach a self-consistent solution.

2.3 Exchange correlation functionals

Thus far, we have have not defined any functional form for E_{xc}. In [50], Kohn and Sham not only provided a scheme to calculate the ground state electron density, but they also suggested a form for E_{xc} for the case of a slowly varying density. In the case of slowly varying density, we call this the local density approximation (LDA), since the functional of the electron density only needs the electron density at a point $\mathbf{r}$. Explicitly, the functional of the density is

$$E_{\mathrm{xc}}^{\mathrm{LDA}}[\rho(\mathbf{r})] = \int d\mathbf{r} f(n(\mathbf{r})) = \int d\mathbf{r} \rho(\mathbf{r}) \epsilon_{\mathrm{xc}}^{\mathrm{LDA}}(\rho(\mathbf{r})), \tag{2.22}$$

where $\epsilon_{\mathrm{xc}}^{\mathrm{LDA}}$ is the exchange correlation energy per electron for a uniform electron gas. For the exchange correlation energy, we split the functional up such that $E_{\mathrm{xc}}^{\mathrm{LDA}} = E_{\mathrm{x}}^{\mathrm{LDA}} + E_{\mathrm{c}}^{\mathrm{LDA}}$. The exchange part, $E_{\mathrm{x}}^{\mathrm{LDA}}$, can be computed for plane-waves using the exchange energy per electron from Hartee-Fock [51] theory, namely,

$$\epsilon_{\mathrm{x,\,k}}^{\mathrm{HF}} = -\sum_{i}^{n} \int d\mathbf{r}' \frac{\psi_k^*(\mathbf{r})\psi_i^*(\mathbf{r}')\psi_i(\mathbf{r})\psi_k(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}, \tag{2.23}$$

along with the functional form for the exchange

$$E_{\mathrm{x}}^{\mathrm{LDA}}[\rho(\mathbf{r})] = \int d\mathbf{r}\rho(\mathbf{r})\epsilon_{\mathrm{x,\,k}}^{\mathrm{HF}}(\rho(\mathbf{r})). \tag{2.24}$$

For a uniform electron gas, the energies associated with the occupied states form a sphere in momentum space. The sphere has a radius of k_{F}, which corresponds with the energy of the highest occupied state (Fermi energy at 0 K) through the dispersion relation $\epsilon_{\mathrm{F}} = \frac{\hbar^2}{2m}k_{\mathrm{F}}^2$. This energy level has an associated wave vector, $\mathbf{k}_{\mathrm{F}}$, and the electron density can be written such that

$$\mathbf{k}_{\mathrm{F}} = (3\pi^2\rho(\mathbf{r}))^{1/3}. \tag{2.25}$$

Using this relation, [50] showed that the exchange energy per electron (Equation 2.23) is

$$\epsilon_{\mathrm{x,\,k}}^{\mathrm{HF}} = -\frac{3}{2\pi}\left(3\pi^2\rho(\mathbf{r})\right)^{1/3}. \tag{2.26}$$

The correlation energy is much harder to compute, and it is defined as the energy difference between the exact total energy of a system and the Hartee-Fock energy with a complete basis. If we consider the Wigner-Seitz radius of a uniform electron gas

$$r_{\mathrm{s}} = \left(\frac{9\pi}{4}\right)^{1/3}\frac{1}{k_{\mathrm{F}}}, \tag{2.27}$$

then we can consider limits of the electron density. In the LDA, we write the correlation energy with the same form as the exchange energy, namely

$$E_{\mathrm{c}}^{\mathrm{LDA}}[\rho(\mathbf{r})] = \int d\mathbf{r}\rho(\mathbf{r})\epsilon_{\mathrm{c}}(\rho(\mathbf{r})), \tag{2.28}$$

where ϵ_c is the correlation energy per electron. The exact form is not known for the correlation energy, but it has been fit to accurate quantum Monte Carlo simulations [52]. Another way to calculate the correlation energy per electron is to perform an exhaustive calculation for a true 2 electron system. This would require a computer with a large amount of memory, but in principle, it can be done. One would first calculate the ground state wave function, and the total energy associated with that state. Afterwards, one can then subtract off all other energies except for the electron-electron interaction energy from the total energy. This would yield the true electron-electron correlation energy. Typically, in electronic structure theory, the electron correlation energy is the true total energy minus the Hatree-Fock total energy.

In [50], Kohn and Sham realized that their approximation to the exchange correlation energy had limitations and suggested that one should perform a gradient expansion of energy functionals with respect to the density to increase the accuracy. Until the release of generalized gradient approximation (GGA) in the early 90s, the LDA approximation was widely used to perform DFT calculations. One of the most popular GGA functionals in DFT is the PBE functional which was developed by Perdew, Burke, and Ernzerhof [22]. We have used this functional in many cases in Chapter 4 as it is a standard functional to use in the literature. This functional uses the same form of the LDA for correlation effects, but adds additional terms in the exchange which includes a gradient of the electron density. More recently, additional hybrid exchange correlation functionals and non-local functionals have been formulated and used. In particular exchange correlation

functionals which contain van der Waals interactions have been developed [53] and thoroughly tested [54]. These functionals have an additional contribution to the energy,

$$E_{\mathrm{xc}} = E_{\mathrm{x}}^{\mathrm{revPBE}} + E_{\mathrm{c}}^{\mathrm{LDA}} + E_{\mathrm{nl}} \tag{2.29}$$

where E_{nl} is the additional term which is non-local. This term depends on the density at different points in space, which may be long range, which is why it is deemed non-local. The general form for the non-local energy is

$$E_{\mathrm{nl}} = \frac{1}{2} \int d\mathbf{r} d\mathbf{r}' \rho(\mathbf{r}) \phi(\mathbf{r}, \mathbf{r}') \rho(\mathbf{r}') \tag{2.30}$$

where ϕ is some general function (interaction) which depends on $\mathbf{r} - \mathbf{r}'$. In [53], Dion *et al.* approximately derived a function ϕ from the exact non-local expression, where the asymptotic interaction strength of $1/R^6$ was found, thereby giving van der Waals interactions. In Chapter 4, we leverage this exchange correlation functional form for all of our calculations, as graphite, and the interactions of water molecules need van der Waals interactions to be described properly.

2.4 Applying DFT

We now work through the computational aspects of a DFT code. To simply things, we work in one dimension; the extension to higher dimensions is straight forward. We also include Python code, which allows us to explain everything in theory and in a computational sense. The first step is to define the real and momentum space

grids.

```
import numpy as np

xnum = 256

x = np.linspace(-10, 10, 256)

dx = x[1] - x[0]

dk = 2 * np.pi / (dx * xnum)

k = (np.arange(xnum) - (xnum / 2)) * dk
```

This definition of $k = 2\pi/x$ is also needed to work with fast Fourier transforms (FFTs). In DFT, we refer to the plane wave energy cutoff to give the total number of plane waves used in our calculations. As we will see below the kinetic energy operator in Fourier space is $\tilde{T} = \frac{k^2}{2}$. Instead of defining a parameter to define the real space grid, we define an energy cut-off, which yields the maximum value of $k_{\max}$. The real space grid can then be defined after $k_{\max}$ is known. With the grid defined, we can then define the nucleic potential, which is this case, will be a Li atom in the centered at $x = 0$, i.e.

$$V(x) = -\frac{Z}{|x|} \approx \frac{Z}{\sqrt{x^2 + \alpha}} = \frac{-3}{\sqrt{x^2 + 1}}$$

or

```
def V(x):

    return - 3 / np.sqrt(x**2 + 1)
```

which is known as a soft-core atomic potential. The value of Z for Li is 3, since it has 3 protons. The Coulomb operator has a singularity at $x = 0$, and the introduction of the parameter α fixes this problem. We can also define the initial guess of our orbitals and our electron density, which will come from a random function. With $\psi(x)$ defined as our random function, we first normalize, and then sum over $|\psi(x)|^2$ n times, where n is the number of electrons. In the case of Li, we have 3 electrons.

```
psi = np.random.rand(xnum)
psi /= np.sqrt( dx * np.sum(np.absolute(psi)**2))


n_elec = 3
rho = np.zeros(xnum)
for i in range(n_elec):
    rho += np.absolute(psi)**2
```

At every self-consistent step, we must also compute the Hartree potential

$$V_{\mathrm{H}}(x) = \int \frac{\rho(x)}{|x - x'|} dx'$$

which is equivalent to solving the Poisson equation

$$\frac{d^2}{dx^2} V_{\mathrm{H}}(x) = -\frac{\rho(x)}{2}.$$

Solving the Poisson equation is much more stable from a computational perspective and can be done using various iterative solvers. The Poisson equation gives steady state solutions of the diffusion equation, and therefore one can guess a solution, and continue updating the solution until some convergence has been reached. In a very naive approach, we solve for the Hartee potential in the following function:

```
def V_h(rho):
    v_h = np.zeros(xnum) + 1e-16
    v_h_n = np.zeros(xnum)
    etol = 1e-4
    diff = np.ones(xnum)
    dt = dx **2 * 0.001
    while np.max(diff) > etol:
        v_h_n[0] =  v_h[0] + (dt / (dx**2)) *
                    (v_h[-1] + v_h[1] - 2 * v_h[0]) +
                    rho[0] * dt
        v_h_n[-1] = v_h[-1] + (dt / (dx**2)) *
                    (v_h[-2] + v_h[0] - 2 * v_h[-1]) +
                    rho[-1] * dt
        for i in range(1,xnum-1):
            v_h_n[i] =  v_h[i] + (dt / (dx**2)) *
                        (v_h[i-1] + v_h[i+1] - 2 * v_h[i]) +
                        rho[i] * dt
        diff = abs(v_h_n - v_h)
```

```
        v_h = copy.copy(v_h_n)
    return v_h
```

Here, we pass the electron density $\rho(x)$ in, and obtain $V_{\mathrm{H}}(x)$. It should be noted that this function also includes periodic boundary conditions, which is why the end points are excluded from the loop. In the 1D case, the momentum vector associated with the Fermi energy is related to the electron density via

$$k_{\mathrm{F}} = \pi\rho,$$

which means that the exchange energy per electron is

$$\epsilon_{\mathrm{x}}(x) = -\frac{3}{2}\rho(x),$$

and the Wigner-Seitz radius is $r_{\mathrm{s}} = 1/\rho$. The exchange potential per electron is

$$v_{\mathrm{x}}(x) = \frac{d}{d\rho}(\rho\epsilon_{\mathrm{x}}(x)) = -3\rho(x),$$

and we use the fit for the correlation energy per electron found by Chachiyo [52]

$$e_{\mathrm{c}}(x) = a\ln\left(1 + b\rho + b\rho^2\right),$$

where $a = \frac{\ln 2 - 1}{2\pi^2}$ and $b = 20.4562557$. The correlation potential per electron is then

$$v_{\mathrm{c}}(x) = \frac{d}{d\rho}(\rho\epsilon_{\mathrm{c}}) = \epsilon_{\mathrm{c}}(x) + \frac{(a\rho)(b + 2b\rho)}{1 + b\rho + b\rho^2}.$$

With these definitions, we can then define all of the potentials needed in our Hamiltonian:

```
def e_x(rho):
    return - (3./2.) * rho


def e_c(rho):
    r_s = 1 / rho
    return (np.log(2.) - 1.) / (2 * np.pi**2) *
           np.log(1 + 20.4562557 / r_s + 20.4562557 / (r_s**2))


def v_c(rho):
    a = (np.log(2.) - 1.) / (2 * np.pi**2)
    r_s = 1 / rho
    return e_c(rho) + a * rho *
           (1. / (1 + 20.4562557* rho + 20.4562557 * rho**2)) *
           (20.4562557 + 2 * 20.4562557 * rho)


def v_x(rho):
    return - 3 * rho
```

The last piece we need is the kinetic energy operator, which in momentum space is

$$\tilde{T}(k) = \frac{1}{2}(k + k_{\mathrm{p}})^2,$$

where k_p corresponds to the k-point we are interested in. The solutions to our Hamiltonian will be of the Bloch form [55], i.e.

$$\psi(x) = u(x)e^{ik_\mathrm{p}x},$$

where $u(x)$ is the part of the wave function that has the same periodicity of the potential. The values of k_p are directly related to the reciprocal space lattice space vectors. For every unit cell, there is one unique value of k_p. To calculate the total energy per unit cell, we must perform calculations over different values of k_p, and average [56]. This is equivalent to integrating $E(k_\mathrm{p})$ over the first Brillouin zone, which in our case is $-\pi/L \leq k_\mathrm{p} \leq \pi/L$. An example of $E(k_\mathrm{p})$ is shown in Figure 2.1 for the uniform electron gas. To compute this integral computationally, we perform a sum over discrete points in k_p space and average. We can continue increasing the number of k-points until we see a convergence in the total energy.

With the kinetic energy operator defined in momentum space, the Hamiltonian can not be diagonalized directly, as done in a real space finite difference method. To solve for the eigenvalues and eigenvectors iteratively, we need to pass the solver the matrix-vector product operation. Once we have solved for the eigenvalues and eigenvectors, we can construct the new electron density, and the total energy for the specific k-point. Putting all of this together we define a function to return the total energy, eigenvalues, and eigenvectors:

```
def kPointCalc(kp, rho):
```

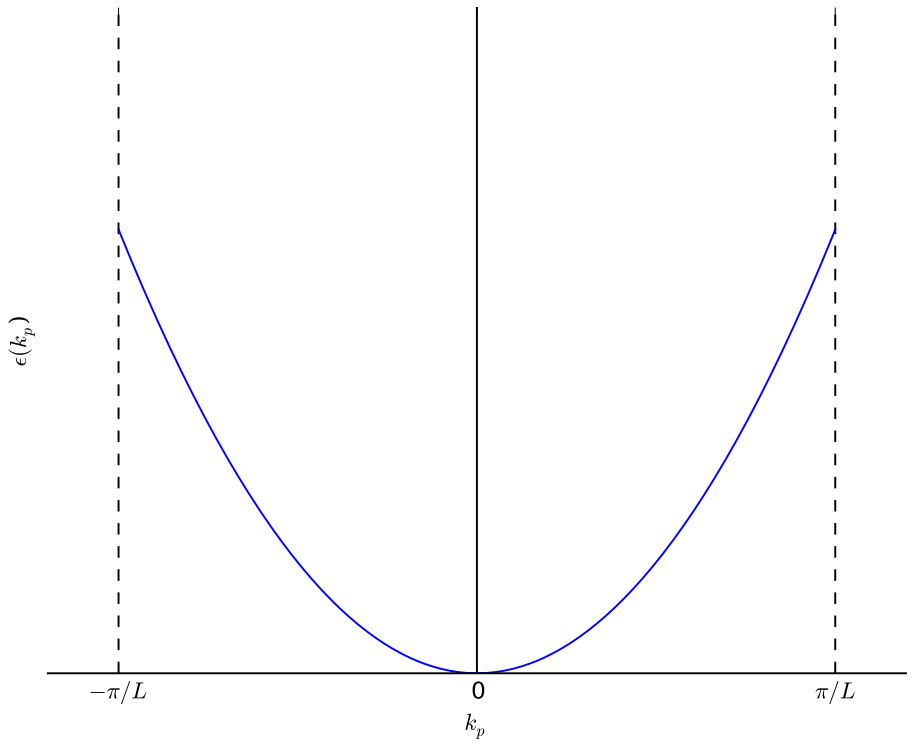

FIGURE 2.1: The dispersion relation for the uniform electron gas. Integrating over k in the Brillouin zone gives us the total energy of the system. In 1 dimension, the Brillouin zone is from $-\pi/L$ to π/L, where L is the real space length.

```
T = 0.5 * (k + kp) ** 2

T = np.fft.fftshift(T)


V_hartree = V_h(rho)

V_x = v_x(rho)

V_c = v_c(rho)


mv = lambda phi: np.fft.ifft(T * np.fft.fft(phi)) +

                 pot_x * phi + V_hartree * phi +

                 (V_c + V_x) * phi

H = LinearOperator((xnum, xnum), matvec=mv )
```

```
    w,v = eigs(H, k=n_elec, which='SR')


    toten = 0.
    for i in range(n_elec):
        orb = (i / 2)
        toten += w[orb]
    toten += -0.5 * np.sum(rho * V_hartree) * dx +
        np.sum(rho * ((e_x(rho) + e_c(rho)) - (V_x + V_c))) * dx


    return np.real(toten), w, v
```

To obtain the total energy, we then define a k_{p}-space grid, and solve for a self-consistent solution at different k-points. We continue until the absolute difference in the total energy of subsequent iterations is smaller than some threshold. This threshold is chosen to be small enough such that the change in the electron density is small, but large enough such that we are reach convergence in a timely fashion. In practice, one would examine the potential energy surface and compare the depths of local minima. This would allow one to calculate an optimal convergence value to allow for a realistic convergence. The pseudocode below provides the self-consistent loop for calculating the total energy:

```
kpoints = np.linspace((-np.pi / a), (np.pi / a), 10)
```

```
rhos = np.zeros((len(kpoints), xnum))
for i in range(len(kpoints)):
    for j in range(n_elec):
        rhos[i] += np.absolute(psi)**2

converged = False
emin_start = 1e6
max_iters = 25
iters = 0
etot = 0.
while not converged:
    dkp = kpoints[1] - kpoints[0]
    etot_next = 0.
    for i, kp in enumerate(kpoints):
        toten, eigvals, eigvecs = kPointCalc(kp, rhos[i])
        rhos[i] = np.zeros(xnum)
        for j in range(n_elec):
            orb = (j / 2)
            rhos[i] += np.absolute(eigvecs[:,orb])**2

        etot_next += toten

    diff = abs(etot_next - etot)
    if diff < 0.001 or iters > max_iters:
        converged = True
```

```
    iters += 1
    etot = copy.copy(etot_next)
    print etot / len(kpoints), iters
```

In this particular case, we have chosen to use a 10 k-point grid. In Figure 2.2 we perform a convergence test as well as plot some of the lowest orbitals for the 1D Li atom. It should be noted that in some cases, the self-consistent cycle will not always converge. There are a few approaches to achieve convergence. Firstly, instead of choosing a random vector as our guess for the orbitals, one can choose a better function as our initial guess. In practice, one would choose atom centered functions that are have similar character to the eigenfunctions of the isolated atoms. Also, once we have found a self-consistent solution, we can use the orbitals and electron density for that solution as initial guesses for other self-consistent calculations. This is done for MD calculations, and in practice, one will use a linear combination of past electron densities. This technique is also used in the self-consistent cycle. If we have 2 orbitals that are close in energy near the fermi level, the occupancy of these orbitals might flip back and forth in the self-consistant cycle. This may lead to vastly different charge densities at each step, leading to lack of convergence. To avoid this, we again take a linear combination of previous electron densities. The rate of convergence is system dependent for a self-consistant cycle. For metals, the eigenvalue spectrum is more broad, which leads to a slower convergence. For insulators and semi-conductors, the width of the eigenvalue spectrum is constant and system size independent, which means that the convergence rate is much faster [57]. If one examines the error versus

iteration for a self-consistent cycle, one should see an exponential decrease in the absolute difference between energies when the calculations are successful. In a MD calculation, one may find the first MD step to be the most costly, and the rate of convergence to be low. Once the first self-consistant solution is found, the self-consistent solutions for the next MD steps are quite rapid. In a self-consistant calculation with a bad initial guess, the absolute difference between energies from step to step may not decrease exponentially initially as it searches for a solution. Once it finds a better solution, the difference between the energies will decrease exponentially.

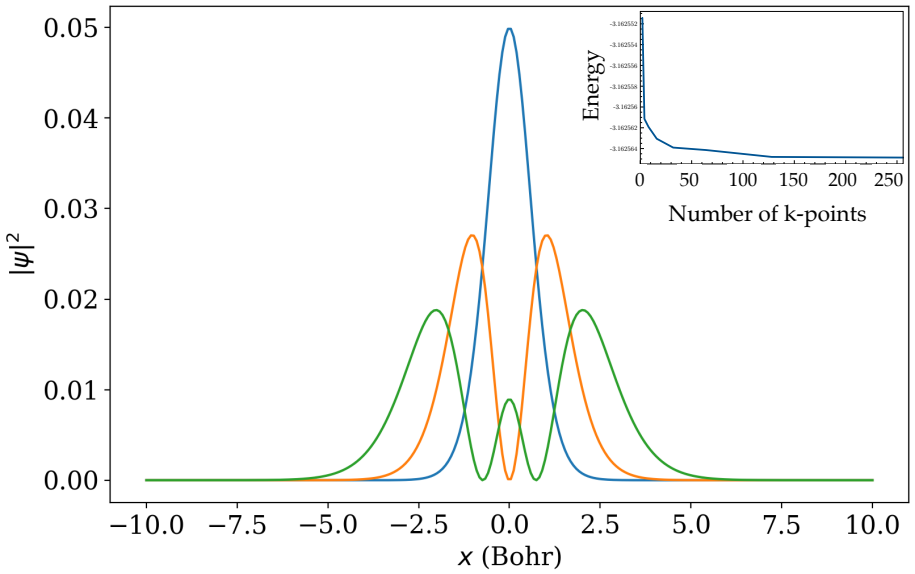

FIGURE 2.2: The ground state, HOMO, and LUMO orbitals for the Li atom. In set: Convergence of the total energy as a function of number of k-points for a Li atom in 1D.

2.5 Computing molecular dynamic trajectories

Molecular dynamics allows one to compute thermodynamic averages, which can then be compared to experimental data. This comparison is extremely important, because it verifies whether our approximations work, or if we need to include more effects to obtain a more accurate answer. Classical molecular dynamics can be paired with DFT, which is described in this section. Once we have minimized the total energy functional, we then have obtained the Kohn-Sham orbitals and their associated energies. Coming back to the nuclei, we now want to find the forces acting on them, so we can integrate Newton's equation of motion. Using the Hellmann-Feynman theorem [58],

$$\frac{\partial E}{\partial \lambda} = \left\langle \psi(\lambda) \left| \frac{\partial H}{\partial \lambda} \right| \psi(\lambda) \right\rangle \tag{2.31}$$

we can compute the force acting on the Ith nuclei via

$$\mathbf{F}_I = -m_I \nabla_{\mathbf{R}_I} E = -m_I \sum_i \langle \psi_i | \nabla_{\mathbf{R}_I} H | \psi_i \rangle , \tag{2.32}$$

where m_I is the atomic mass of the nuclei, and ψ_i are the occupied Kohn-Sham orbitals. In our Hamiltonian H, there are only 2 operators that depend on the nucleic coordinates $\mathbf{R}_I$. These two operators are the external potential of the electronic Hamiltonian, and the coulomb potential for the nuclei. Therefore the

force on nuclei I is

$$\begin{aligned}\mathbf{F}_I(\mathbf{R}_I) &= -m_I \sum_i \int d\mathbf{r}\psi_i^*(\mathbf{r})\nabla_{\mathbf{R}_I}\left[-\frac{Z_I}{|\mathbf{r}-\mathbf{R}_I|}+\sum_{J\neq I}\frac{Z_J Z_I}{|\mathbf{R}_J-\mathbf{R}_I|}\right]\psi_i(\mathbf{r}) \\ &= -m_I\left[\int d\mathbf{r}\rho(\mathbf{r})\frac{Z_I(\mathbf{r}-\mathbf{R}_I)}{(\mathbf{r}-\mathbf{R}_I)^3}+n\sum_{J\neq I}\frac{Z_I Z_J(\mathbf{R}_J-\mathbf{R}_I)}{(\mathbf{R}_J-\mathbf{R}_I)^3}\right]. \qquad (2.33)\end{aligned}$$

After computing the forces, we then use the Verlet algorithm [59] to integrate the equation of motion. The derivation of the Verlet algorithm is straight forward, we consider the Taylor expansion of the position of an atom at an infinitesimal time $\pm\delta t$ centered about t. Doing this we get

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \dot{\mathbf{r}}(t)\delta t + \ddot{\mathbf{r}}(t)\delta t^2 + \dddot{\mathbf{r}}(t)\delta t^3 + \mathcal{O}(\delta t^4) \qquad (2.34)$$

$$\mathbf{r}(t-\delta t) = \mathbf{r}(t) - \dot{\mathbf{r}}(t)\delta t + \ddot{\mathbf{r}}(t)\delta t^2 - \dddot{\mathbf{r}}(t)\delta t^3 + \mathcal{O}(\delta t^4), \qquad (2.35)$$

where $\dot{\mathbf{r}} \equiv d\mathbf{r}/dt$. Summing these together we arrive at

$$\mathbf{r}(t+\delta t) = 2\mathbf{r}(t) - \mathbf{r}(t-\delta t) + \ddot{\mathbf{r}}(t)\delta t^2 + \mathcal{O}(\delta t^4), \qquad (2.36)$$

where $\ddot{\mathbf{r}} = \mathbf{F}/m$ and we can see the errors of the expansion are of order δt^4.

2.6 Simulating the canonical ensemble

In our MD simulations, we would like to describe the system as accurate as possible in comparison with experiments. In experiments, a typical ensemble is the

canonical ensemble, or the NVT ensemble. This ensemble has a constant temperature (T), volume (V), and number of particles (N). There are many different approaches to simulate the NVT ensemble, and in our work we use what is called a Nosé-Hoover thermostat. In the work done by Nosé [60] and Hoover [61], they considered introducing a fictitious force proportional to the velocity in the equation of motion for a particle. Again, for the Ith nuclei, the equation of motion is written as

$$m_I \ddot{\mathbf{r}} = \mathbf{F}(\mathbf{r}) - \zeta m_I \dot{\mathbf{r}} \tag{2.37}$$

$$\dot{\zeta} = \frac{1}{Q}\left[\sum_{I=1}^{N} m_I \frac{\dot{\mathbf{r}}_I^2}{2} - \frac{3N+1}{2} k_B T\right], \tag{2.38}$$

where Q and ζ are the friction parameters, k_B is the Boltzmann constant, and T the desired temperature. In steady state, $\dot{\zeta} \to 0$, and the kinetic energy of the nuclei is equal to $\frac{3}{2}(N+1)k_B T$, as it should due to the equipartition theorem. As $t \to \infty$, the average temperature of the system approaches the desired temperature, which gives the NVT ensemble.

2.7 Pseudopotentials

Pseudopotentials allow us to speed up calculations by only solving for the orbitals of valence electrons. For an atom with many electrons, some are considered as core electrons, and some are valence. The valence electrons are the most interesting because they allow for chemical bonding, whereas the core electrons remain tightly bound to the nuclei. Valence electrons do not see the true nucleic coulomb

potential, but a screened coulomb potential. This potential is much weaker due to the repulsion with the core electrons. If we write down the Hamiltonian for the valence electrons, we can estimate that

$$H = T + V + V_{\mathrm{R}} \tag{2.39}$$

where T is the normal kinetic energy operator, and $V_{\mathrm{P}} \equiv V + V_{\mathrm{R}}$ is a weak screened coulomb potential which can be referred to as the pseudopotential. If we solved for a pseudo wave function of this Hamiltonian, the highly oscillatory part of the original wave function disappears, allowing for use of a much lower resolution grid (which reduces computational cost).

In the 1960's, Ashcroft [62] came up with a simple form for a pseudopotential. Namely,

$$V_{\mathrm{P}}(r) = \begin{cases} 0 & \text{if } r \leq R_{\mathrm{c}}, \\ -\frac{Z}{r} & \text{if } r > R_{\mathrm{c}}. \end{cases} \tag{2.40}$$

Here, R_{c} is some cutoff radius which separates the core from the valence region of the atom. This original form led to other variations done with empirical fits, which were still unphysical, and suffered from discontinuities at R_{c}. Although the idea of orthogonalized plane waves (OPG) [63] was developed much earlier to construct valence wave functions from core wave functions while maintaining the orthogonality condition, they were not used to construct pseudopotentials until the 1970's [64]. Using the formulation of OPG's, we assume that the valence wave function can be written as a sum of a smooth function and a sum over occupied

states,

$$\psi = \phi + \sum_c a_c \phi_c, \tag{2.41}$$

where $a_c = -\langle \phi_c | \phi \rangle$ because $\langle \phi_c | \psi \rangle = 0$. The Schrödinger equation then becomes

$$\begin{aligned} H\psi &= E\psi \\ H\phi - \sum_c \langle \phi_c | \phi \rangle E_c \phi_c &= E\phi - E \sum_c \langle \phi_c | \phi \rangle \phi_c \\ H\phi + \sum_c (E - E_c) \langle \phi_c | \phi \rangle \phi_c &= E\phi \\ (H + V_c)\phi &= E\phi, \end{aligned} \tag{2.42}$$

which looks very similar to the original Hamiltonian, but with a repulsive term, as expected. This formalism allowed one to calculate valence orbitals, but did not satisfy the normality constraint of wave functions. Hamann *et al.* [65] developed "norm-conserving pseudopotentials" which satisfied the following properties:

1. The all electron and pseudo eigenvalues agree.

2. The all electron and pseudo wave functions agree beyond the cut off distance R_c.

3. The electron density agrees between the all electron and pseudo wave functions for $0 \leq R < R_c$, i.e.

$$\int_0^R |\psi_{\mathrm{AE}}|^2 dr = \int_0^R |\psi_{\mathrm{PS}}|^2 dr.$$

4. The logarithmic derivative, and the energy derivative of the logarithmic derivative agree for the all electron and pseudo wave functions, i.e.

$$(r\psi_{\mathrm{AE}})^2 \frac{d}{dE}\frac{d}{dr}\ln\psi_{\mathrm{AE}} = (r\psi_{\mathrm{PS}})^2 \frac{d}{dE}\frac{d}{dr}\ln\psi_{\mathrm{PS}}.$$

The norm conservation comes from item 3, and item 4 allows for the transferability of the pseudopotential in different environments. The logarithmic derivatives must be continuous, as all of the eigenfunctions that are solutions to the Schrödinger equation are continuous and have continuous first derivatives. When generating a pseudopotential, we assume spherical symmetry and therefore only have to deal with the radial coordinate. The first step is to run an all electron calculation, and then decide which eigenfunctions will be core and which ones will be valence. For the valence electrons, we choose some analytic function u_{nl} (n and l are the principal and azimuthal quantum numbers) which can be fit to our all electron orbital. This analytic function must also satisfy the properties mentioned previously. We then invert the radial equation,

$$-\frac{1}{2}\frac{d^2u_{nl}}{dr^2} + \left[V + \frac{1}{2}\frac{l(l+1)}{r^2}\right]u_{nl} = \epsilon u_{nl} \tag{2.43}$$

for V, which yields

$$V(r) = \epsilon - \left[\frac{1}{2}\frac{l(l+1)}{r^2} - \frac{1}{2u_{nl}}\frac{d^2u_{nl}}{dr^2}\right]. \tag{2.44}$$

To replace the ionic potential, we must also subtract off the Hartee and exchange

correlation potential, which gives the pseudopotential for a particular valence electron.

2.7.1 Projector augmented wave method

The projector augmented wave (PAW) method is a generalization of the pseudopotential method, and is used in all of the calculations presented in this thesis. This method uses a linear transformation which maps the pseudo wave functions to the true all electron wave functions found through Kohn-Sham DFT. The pseudo wave functions are computationally much cheaper to calculate, be we must rewrite everything with respect to these pseudo functions. The total energy expression must be transformed as well, leading to a new Hamiltonian to diagonalize. When Blöchl [66] first introduced this method, he proposed that we define some transformation operator $\hat{\mathcal{T}}$ such that

$$\psi_n(\mathbf{r}) = \hat{\mathcal{T}}\tilde{\psi}_n(\mathbf{r}) \Longleftrightarrow \tilde{\psi}_n(\mathbf{r}) = \hat{\mathcal{U}}\psi_n(\mathbf{r}), \tag{2.45}$$

where ψ_n is an eigenfunction of the all electron problem, and $\tilde{\psi}_n$ are the pseudo functions that we want to be well behaved and computationally sound. Here, $\hat{\mathcal{T}} = \hat{\mathcal{U}}^{-1}$. With this definition, we can then express the total energy, a Schrödinger

like equation, and expectation values with respect to the pseudo functions;

$$E = E[\{\psi_n\}] = [\{\hat{\mathcal{T}}\tilde{\psi_n}\}], \tag{2.46}$$

$$H\hat{\mathcal{T}}\tilde{\psi_n} = \epsilon_n\hat{\mathcal{T}}\tilde{\psi_n}, \tag{2.47}$$

$$A = \langle\tilde{\psi_n}|\hat{\mathcal{T}}^\dagger\hat{A}\hat{\mathcal{T}}|\tilde{\psi_n}\rangle, \tag{2.48}$$

where $\hat{\mathcal{T}}^\dagger$ is the adjoint of $\hat{\mathcal{T}}$. To define the transformation operator, we follow a similar procedure as the norm-conserving pseudopotentials. We first solve the all electron problem to obtain a set of valence orbitals $\{\psi_n\}$, choose a cut-off radius R_c, and set the pseudo orbitals $\{\tilde{\psi_n}\}$ such that for $r > R_\mathrm{c}$ the pseudo and true functions are equal, and the pseudo functions also satisfy orthogonality to the core states. We then use the same process as before to invert the radial equation to obtain the pseudopotential for each valence orbital. These all electron orbitals, pseudo orbitals, and pseudopotentials will show up in the transformation operator, and therefore our new Hamiltonian. Writing the transformation operator as

$$\hat{\mathcal{T}} = 1 + \sum_i \hat{\mathcal{S}}_i, \tag{2.49}$$

where the sum goes over all occupied valence orbitals. The transformation then becomes

$$|\psi_n\rangle = |\tilde{\psi_n}\rangle + \hat{\mathcal{S}}_n|\tilde{\psi_n}\rangle \Longleftrightarrow \hat{\mathcal{S}}_n|\tilde{\psi_n}\rangle = |\psi_n\rangle - |\tilde{\psi_n}\rangle, \tag{2.50}$$

which can also be written as

$$\hat{\mathcal{S}}|\tilde{\psi_n}\rangle = \sum_i(|\psi_i\rangle - |\tilde{\psi_i}\rangle)\langle\tilde{p}_i|\tilde{\psi_n}\rangle, \tag{2.51}$$

which means that the transformation operator is

$$\hat{\mathcal{T}} = 1 + \sum_i (|\psi_i\rangle - |\tilde{\psi}_i\rangle)\langle\tilde{p}_i|. \tag{2.52}$$

Here, $\tilde{p}_i$ are localized projector functions which obey the bi-orthogonality condition $\langle\tilde{p}_i|\tilde{\psi}_n\rangle = \delta_{i,n}$. In practice, the projector functions will have either s, p, or d like character to resemble true atomic environments. We then use this transformation and apply it to the Hamiltonian of the Kohn-Sham scheme to obtain a new pseudo Hamiltonian which is then used to self-consistently solve for the density. In this Hamiltonian, we only solve for valence orbitals and the external potential disappears. With that being said, the transformation causes many other terms to appear in the new Hamiltonian which depend on the radial functions and pseudopotentials found through inversion.

Chapter 3

Probing atomic systems

In this chapter, we discuss methods to create, and study interfaces. In this case, the interfaces consist of a crystal lattice (metal) and a liquid (water). In the regime of interest, we need to use quantum mechanics, hence the previous chapter on DFT. DFT is the workhorse that will provide us the forces necessary for computing MD trajectories, which we then analyze.

Within this Chapter, we include Python code to illustrate how the distributions are constructed. Although the Python code is useful for understanding, it can be slow when creating some distributions. It is therefore necessary to use a language where you first compile, and then execute. For all of the distributions we cover, we wrote the equivalent Python code in C++ for computational speed up.

3.1 Simulation details

For most of the distributions we show in this Chapter, we performed the analysis of MD for 100 water molecules in a periodic simulation cell with dimensions $14.41 \times 14.41 \times 14.41$ Å. Within the DFT calculations, we used a $1 \times 1 \times 1$ k-point grid centred about the Γ point, a thermostat of either 330 or 400 K, and the PBE or vdW-optB86b exchange correlation functional. To generate the initial atomic coordinates of the water molecules, we used Packmol [67]. To generate the atomic coordinates of the metal atoms, we used self written scripts. When constructing interfaces, one must be concerned with the exclusion volume of the interface. This volume separates the water molecules from the metal surface and it originates from the electronic structure of the crystal. To obtain an exclusion volume, one must use trial and error until the density of the bulk water (water further away from the surface) is 1 g/cc.

To generate all of the MD in this Chapter (and this thesis), we use the Vienna *ab initio* Simulation Package (VASP) [68]. VASP is a complex simulation package, and we therefore suggest that the reader refer to online documentation provided by VASP or other research groups. To give a brief description, VASP is a command line DFT program written in Fortran meant to be executed on Unix based systems. It has been throughly optimized to run on supercomputers which have thousands of computer cores and is regarded as one of the most accurate and efficient codes in the electronic structure community.

3.2 Structural properties of interfaces

To study the structural properties of interfaces, one must study the geometry of the liquid next to the metal surface. One can first study the liquid on its own through structural distributions, and then begin to understand the effects of the metal by looking at the differences in the distributions. Some of these geometrical distributions we discuss in detail in this chapter. We include both applied (Python) and theoretical approaches when possible. In all cases, we assume we have generated molecular dynamics, and are reading in a file which holds the atomistic information.

3.2.1 Radial distribution function

The radial distribution function (RDF) or pair correlation function (PCF) is commonly used to understand the structure of atomistic system, and is extremely useful in understanding the structure of a liquid. Physically, it is quite intuitive. Imagine a box of water molecules, where the origin is centred about one of the oxygen atoms. To calculate the O-O radial distribution function ($g(r)$), we start moving away radially from the origin by dr, such that the volume slice is

$$V = \frac{4}{3}\pi[(r + dr)^3 - r^3]. \tag{3.1}$$

We then count all of the other oxygen atoms that fall within this volume slice. After moving through the full space, we then move our origin to another oxygen atom, and repeat the process once we have covered all oxygen atoms. In Python

code, a radial distribution function looks like the following.

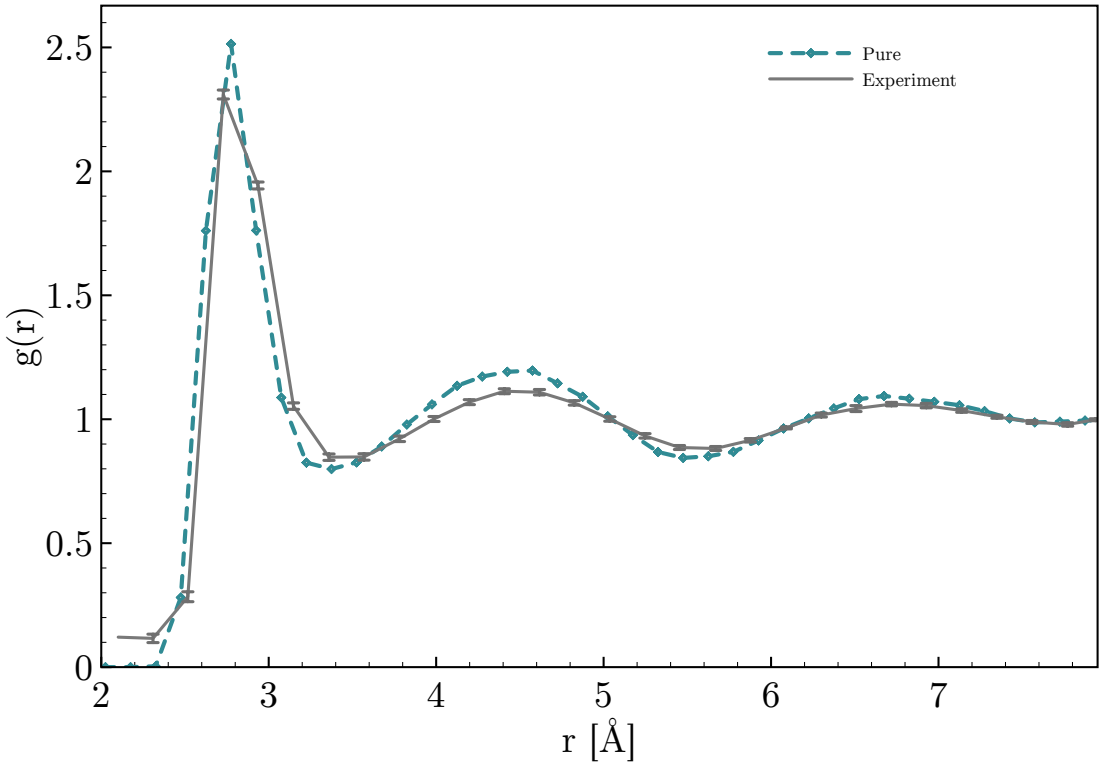

FIGURE 3.1: RDF for a simulation of liquid water (labelled as "Pure") using the PBE exchange correlation functional with a thermostat of 330 K. An RDF from a neutron scattering experiment [1] reported at 298 K is also shown (labelled as "Experiment").

```
import numpy as np

# assuming we know the cell dimensions
V_cell = ax * ay * az
n_particles = 100

# uniform density for normalization
avg_dens = (n_particles - 1) / V_cell

# parameters for the distribution
max_r = 5.0
incr = 0.1
rdf = np.zeros(int(max_r / incr))

# generate the distribution
for frame in frames:
  # get the type and coordinates of atom 1
  for i, (x1, y1, z1) in enumerate(frame):
    # get the type and coordinates of atom 2
```

```
    for j, (x2, y2, z2) in enumerate(frame):
      if i != j:
        dx = x1 - x2
        dy = y1 - y2
        dz = z1 - z2
        r = (dx * dx + dy * dy + dz * dz) ** 0.5
        if r < max_r:
          rdf[int(r / incr)] += 1

# normalize the distribution
for i, val in enumerate(rdf):
    r_0 = i * incr
    r_1 = (i+1) * incr
    V_slice = 4.0 * np.pi * (r_0**3 - r_1**3) / 3.0
    rdf[i] *= avg_dens / V_slice
```

In Figure 3.1, we show the O-O radial distribution function for liquid water. To generate the RDF, we ran 5 ps of MD using the PBE exchange correlation functional at a temperature of 330 K.

3.2.2 Density profile

Similar to the RDF, the density profile of a liquid is a very common measure. It is fundamental to understanding the structure of liquids next to surfaces, and is almost always used to study the structure of water next to metals. The density profile is normally averaged over a spacial coordinate, but a contour, or 3 dimensional distribution can be calculated as well. In the case of interfaces, the crystal is usually placed next to the liquid along the z-axis, and therefore the density profile with respect to the z-axis is commonly reported. The following Python code gives the density profile for liquid water as a function of the z-axis:

```
import numpy as np

# assuming we know the cell dimensions
```

```
ax = ...
ay = ...
az = ...

n_bins = 100
incr = az / n_bins

# conversion rate to g/cc
conversion = 18.0e-6/(6.023e23*1.0e-30)

# initialize the distribution
dp = np.zeros(n_bins)

# build the distribution
# loop over each frame
for frame in frames:
  # loop over every atom
  for i, (x, y, z) in enumerate(frame):
    dp[int(z / incr)] += 1

# normalize the distribution
for i, val in enumerate(dp):
  dp[i] *= conversion / ( ax * ay * len(frames))
```

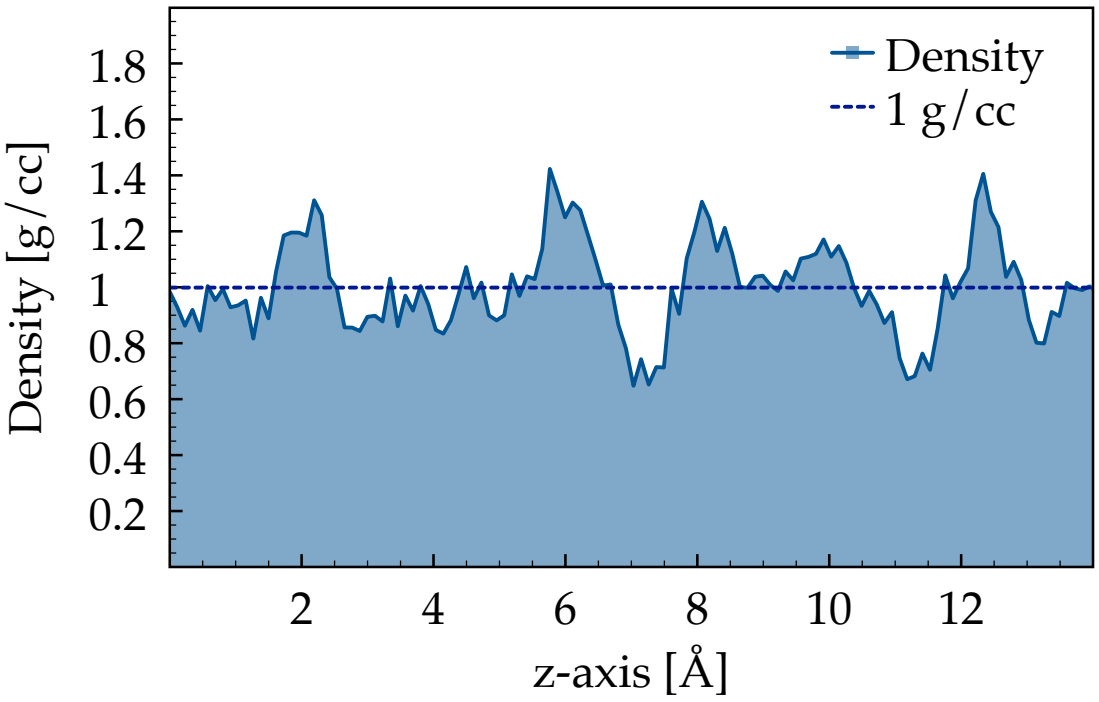

FIGURE 3.2: Density profile as a function of the z-axis for liquid water using the vdW-optB86b exchange correlation functional at a temperature of 400 K. Note that the distribution is not flat. If we simulate for a longer time, the distribution will be more smooth.

In Figure 3.2, we show the density profile for bulk liquid water. To generate the density profile, we ran 10 ps of MD using the vdW-optB86b exchange correlation functional at a temperature of 400 K.

3.2.3 Orientation distributions

When studying liquids that are formed by molecules, it is particularly useful to understand the orientation of the molecules next to a surface. Similar to the density profile, orientation distributions are most useful in either one or two dimensions. In one dimension, one would be concerned with the distribution of angle between some measure of the molecule (i.e. geometric dipole moment) and some axis (i.e. z-axis). To extend this to two dimensions, one would simply add in a dimension like spatial dependence of the angle (like the z-axis for an interface stacked along the z-axis). Below, we show how to obtain a probability distribution of finding a water molecule with a certain angle from the z-axis, based on its geometric dipole. To visualize this angle, see Figure 3.3. To form this distribution, we assume that for each oxygen atom, a pair of intramolecular hydrogens has been grouped using naive radial cut-off distances.

```
import numpy as np

# assuming we know the cell dimensions
ax = ...
ay = ...
az = ...

# initialize the distribution
orientations = np.zeros(180)

# build the distribution
```

```
# loop over every frame
for frame in frames:
  # loop over every molecule
  for id, molecule in enumerate(frame):
    # element wise vector subtraction
    dr1 = molecule.H1_coords - molecule.O_coords
    dr2 = molecule.H2_coords - molecule.O_coords
    dipole = dr1 + dr2
    z_proj = dipole[2] / np.sqrt(dipole[0]*dipole[0]
                       + dipole[1]*dipole[1]
                       + dipole[3]*dipole[3])
    angle = int(np.arccos(z_proj))
    orientations[angle] += 1

# normalize the distribution
for i, val in enumerate(orientations):
  orientations[i] /= np.sum(orientations)
```

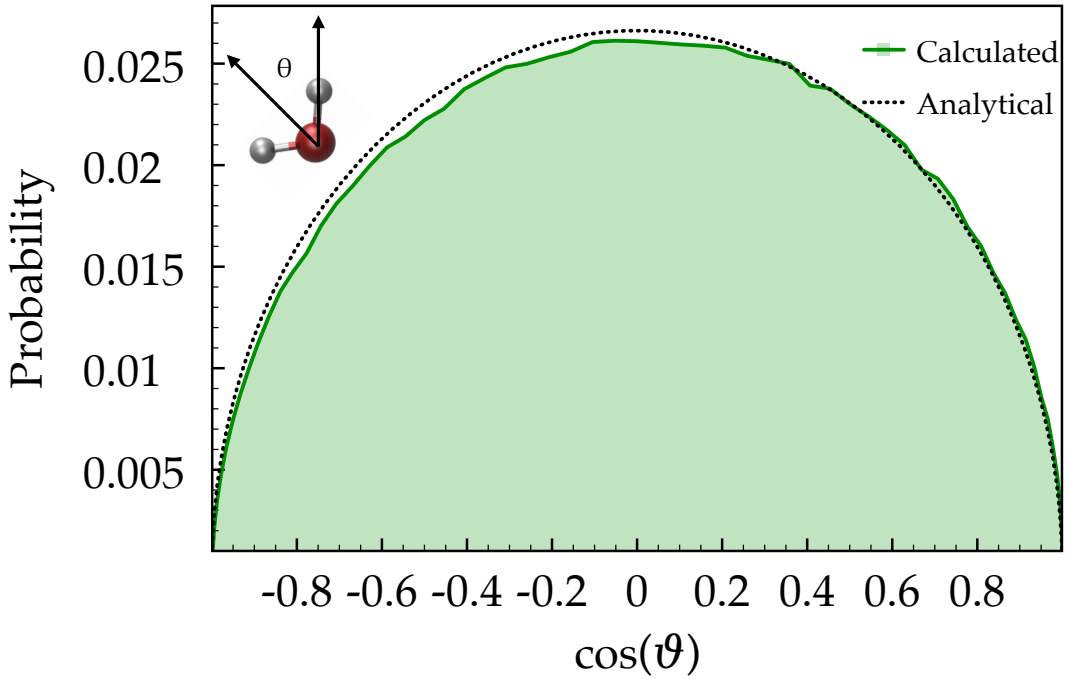

FIGURE 3.3: Probability distribution giving the probability of finding a water molecule with it's geometric dipole at a certain angle from the z-axis.

In Figure 3.3, we show the orientation distribution for a single water molecule that has been randomly oriented a million times. This was done by generating an angle, and then applying three dimensional rotation matrices on the coordinates of the atoms within the water molecule. The shape of the probability distribution

arises due the number of symmetric permutations that can be applied to the water molecule while maintaining the same angle from the z-axis. When $\theta = \pi/2$, we have the maximum number of permutations, and when $\theta = 0$ or $\theta = \pi$, we have a minimum in the number of permutations. For a certain angle θ, the number of permutations is equal to the circumference of the circle where the radius is the projection of the geometric dipole onto the xy-plane. Namely, the circumference is

$$C(\theta) = 2\pi|\mathbf{g}|\sin\theta, \tag{3.2}$$

where $\mathbf{g}$ is the geometric dipole of the water molecule. To calculate the probability of finding a certain orientation, we have to first calculate every permutation we can find (analogous to the partition function). This can be found via

$$Z = 2\pi|\mathbf{g}|\int_0^\pi \sin^2\theta d\theta = \pi^2|\mathbf{g}|. \tag{3.3}$$

The probability is then

$$P(\theta) = C(\theta)/Z = \frac{2}{\pi}\sin\theta. \tag{3.4}$$

We show this along with the calculated result in Figure 3.3.

3.2.4 Mean squared displacement

Apart from looking at geometrical features of liquids, it is also crucial to investigate the dynamics of the molecules. A key observable that describes the dynamics of a liquid is the mean squared displacement (MSD). Experimentally, the self-diffusion

coefficient can be measured, and from Einstein's work on Brownian motion [69], he found that the MSD is related to the self-diffusion coefficient D via

$$\langle \mathbf{r}^2 \rangle = 6Dt. \tag{3.5}$$

Here, $\langle \mathbf{r}^2 \rangle$ is the ensemble average of the mean squared displacement. It tells us to calculate the average across all particles. To compute the MSD, two things should be noted. Firstly, if you have coordinates that are wrapped in their cell (due to the use of periodic boundary conditions), you must unwrap them. Secondly, you must calculate the centre of mass for the system to avoid errors if there is a group velocity. The following Python code gives the MSD for a liquid of one atom type:

```
import numpy as np

# assuming we have a few parameters
time_step = ...
msd_length = ...
n_restarts = ...
n_atoms = ...
block_size = int(msd_length / time_step)

# initialize the distribution
msd = np.zeros(block_size)

# loop over different starting configurations
for i in range(n_restarts):
  start_index = i * block_size
  end_index = (i+1) * block_size
  # loop over the frames in this block
  for step, frame in enumerate(frames[start_index:end_index]):
    # loop over every molecule
    com_diff = center_of_mass(frames[start_index])
                          - center_of_mass(frame)
    for id, molecule in enumerate(frame):
      dx = molecule[0] - frames[step][id][0] - com_diff[0]
      dy = molecule[1] - frames[step][id][1] - com_diff[1]
      dz = molecule[2] - frames[step][id][2] - com_diff[2]
      msd[step - start_index] += dx * dx + dy * dy + dz * dz
```

```
# normalize the distribution
for i, val in enumerate(total_msd):
  msd[i] /= (n_restarts * block_size * n_atoms)
```

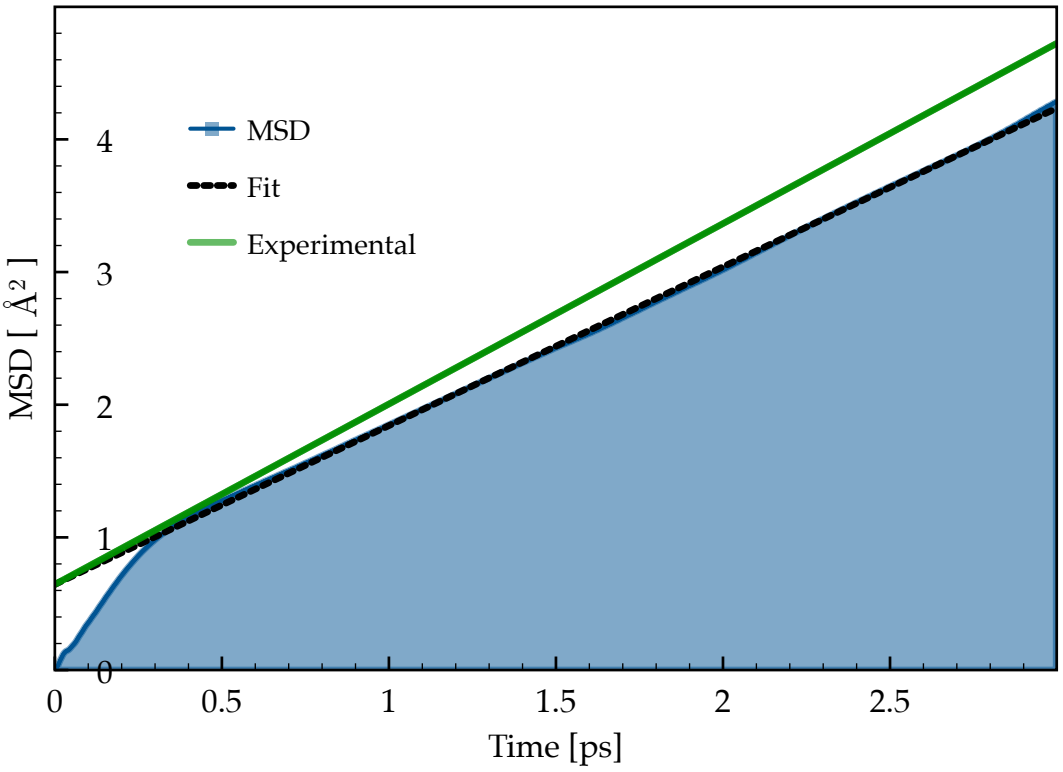

FIGURE 3.4: Mean squared displacement for water using the vdW-optB86b exchange correlation functional at a temperature of 400 K. The dotted line is a fit ($y = mx + b$) to the asymptotic behaviour of the liquid water, where $m = 1.19721$ and $b = 0.6452$. The fit was calculated using gnuplot's fit function. The experimental fit (based on the diffusion coefficient) is also shown.

In Figure 3.4, we show the MSD for bulk liquid water, and overlay the slope of the line, giving the diffusion coefficient. Based on the fit, we calculate a diffusion coefficient of $2.0 \times 10^{-9}\mathrm{m}^2\mathrm{s}^{-1}$. This is less than experimental diffusion coefficient of water found via neutron scattering [70], which gave a value of $2.30\times10^{-9}\mathrm{m}^2\mathrm{s}^{-1}$.

3.2.5 Hydrogen bonding network

Liquid water is a complex liquid, where not only do intramolecular hydrogen bonds form, but intermolecular ones as well. In some of the literature [5, 10], it is common to report the average coordination number of a water molecule as a function of some parameter (i.e. some spatial coordinate). At the interface, the hydrogen bonding network is disrupted from its bulk state, and causes a decrease in the number of hydrogen bonds in the interfacial layer of water. It is therefore useful to report the average number of hydrogen bonds as a function of distance from the surface. Below, we show Python code to calculate such a distribution.

```
import numpy as np

# assuming we know the cell dimensions
ax = ...
ay = ...
az = ...

n_bins = 100
incr = az / n_bins

# initialize the distribution
hb_distro = np.zeros(n_bins)
hb_counts = np.zeros(n_bins)

# build the distribution
# loop over each frame
for frame in frames:
  # loop over every atom
  for atom in frame:
    if atom.type == 'O':
      hb_distro[int(z / incr)] += atom.nearest_neighbour_count
      hb_counts[int(z / incr)] += 1

# average the distribution
for i, val in enumerate(hb_distro):
  hb_distro[i] = float(hb_distro[i] / hb_counts[i])
```

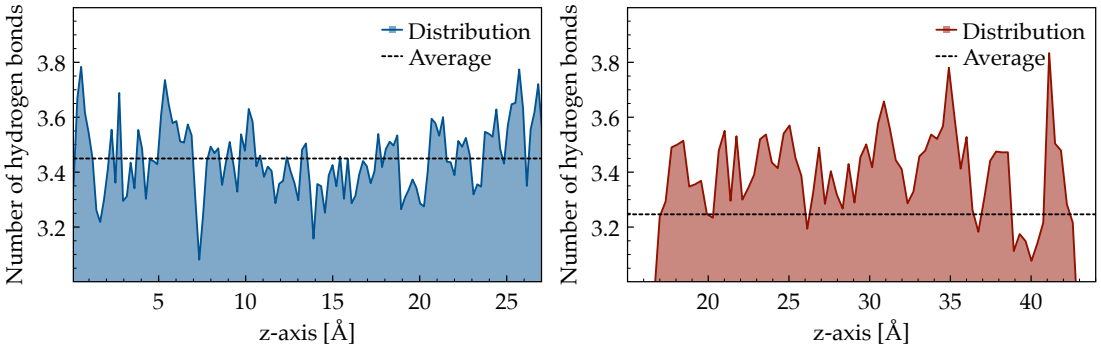

FIGURE 3.5: Average number of hydrogen bonds as a function of the z-axis for bulk water (left), and a liquid water-platinum interface. Both simulations were done with a thermostat of 400 K and used the vdW-optB86b exchange correlation functional.

In Figure 3.5, we show the average number of hydrogen bonds for an interface, and for bulk water. As we can see, the average number of hydrogen bonds has decreased once we introduce the metal slab. As we move away from the metal, the number of hydrogen bonds increases rapidly. Even close to the metal, the hydrogen bonding network has a similar topology to bulk water.

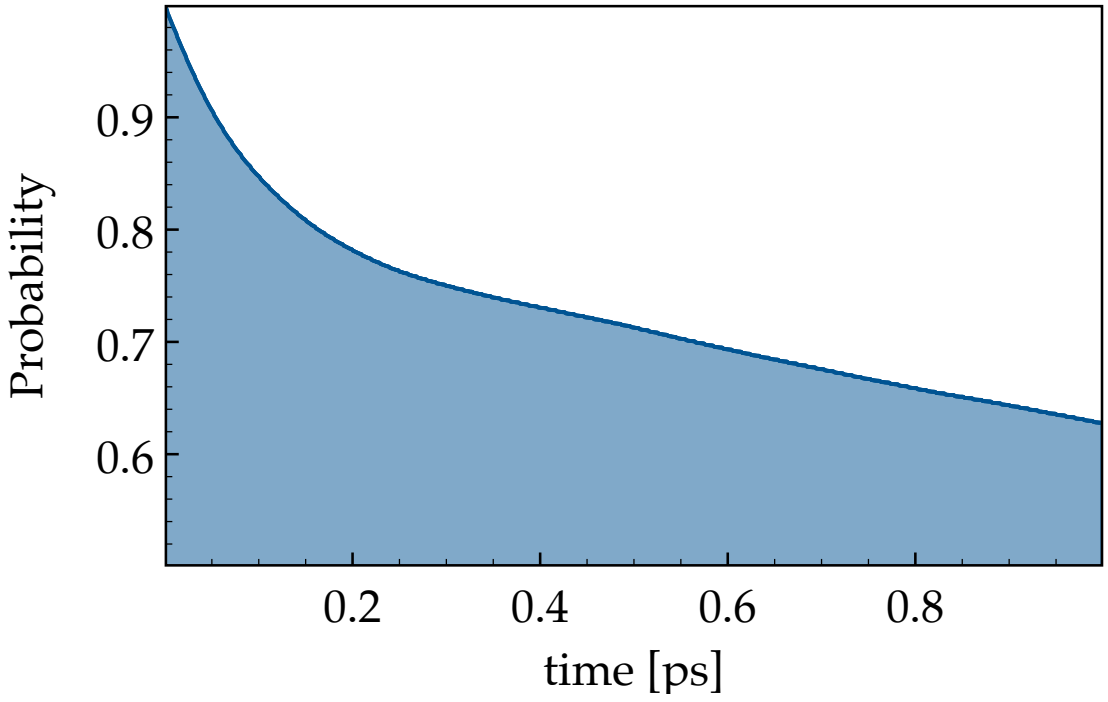

FIGURE 3.6: Probability of finding a new nearest neighbour water molecule for bulk water at a temperature of 400 K using the vdW-optB86b exchange correlation functional.

In addition to the average topology of the hydrogen bonding network, we can also study the dynamics of the intermolecular hydrogen bonds. To do so, we follow the work of [71], and define a probability distribution $\Gamma(t)$, which gives the probability of finding the same nearest neighbours after a certain time interval. When water molecules diffuse, they break intermolecular bonds and form new ones. It is of interest to examine the role a metal has on $\Gamma(t)$. Below, we give Python code for calculating $\Gamma(t)$, again, assuming nearest neighbours have been previously calculated.

```
import numpy as np

# assuming we have a few parameters
time_step = ...
msd_length = ...
n_restarts = ...
n_atoms = ...
block_size = int(nrt_length / time_step)

# initialize the distribution
nrt = np.zeros(block_size)

normalizer = 0

# loop over different starting configurations
for i in range(n_restarts):
  start_index = i * block_size
  end_index = (i+1) * block_size
  # get the total number of old neighbours
  for molecule in frames[start_index]:
    normalizer += molecule.neighbours
  # loop over the frames in this block
  for step, frame in enumerate(frames[start_index:end_index]):
    # loop over every molecule
    for id, molecule in enumerate(frame):
      old_neighbours = frames[step - 1][id].neighbours
      new_neighbours = molecule.neighbours
      # get the count of same neighbours
      nrt[step - start_index] +=
               len(new_neighbors.intersection(old_neighbours))

# normalize the distribution
```

```
for i, val in enumerate(total_msd):
  nrt[i] /= normalizer
```

In figure 3.6, we show $\Gamma(t)$ for pure water. We can see that almost half of the water molecules will have a new nearest neighbour after only 1 ps. In chapter 4, we examine $\Gamma(t)$ in closer detail for liquid water-metal interfaces.

3.3 Electronic properties of interfaces

3.3.1 Partial density of states

In addition to computing structural properties from MD, we also have access to Kohn-Sham orbitals, which contain all of the electronic information. This allows us to compute the density of states (DOS) or the projected density of states (PDOS). The projected density of states is useful, because then we can see contribution of s, p, and d orbitals. One has the ability to construct s, p, and d like orbitals, which we label as $|s\rangle$, $|p\rangle$, and $|d\rangle$. To compute the PDOS for the s orbital, we take the set of Kohn-Sham orbitals we have calculated for our system, and build the distribution using

$$D_s(\epsilon) = \sum_{i=1}^{n} |\langle s|\phi_i\rangle|^2 \delta(\epsilon - \epsilon_i). \tag{3.6}$$

Now, $|s\rangle$ is not necessarily orthogonal to our computed orbitals, and in practice, one must compute all of the inner products $\langle s|\phi_i\rangle$. If the orbital $|s\rangle = \sum_{j=1}^{m} |s_j\rangle$,

then we must compute the overlap matrix elements, often referred to as $S_{ji} = \langle s_j | \phi_i \rangle$.

3.3.2 Adsorption energies

In atomistic surface science, one is often interested in calculating the adsorption energy of molecules on surfaces. If the energy is negative, it means that the molecule is energetically stable on the surface, which would lead to either physisorption or chemisorption. To calculate the adsorption energy, consider a metal slab with a single water molecule structurally relaxed on the surface. The energy associated with this configuration is E_{tot}. We then compare this energy with the energy of a system where the slab is infinitely far away from the water molecule, where the total energy of system is now just the slab E_{slab}, and the energy of just the water molecule $E_{\text{H}_2\text{O}}$. The adsorption energy is the difference between these two regimes,

$$E_{\text{ads}} = E_{\text{tot}} - (E_{\text{slab}} + E_{\text{H}_2\text{O}}). \tag{3.7}$$

Chapter 4

Characterizations of liquid water-metal interfaces

This chapter is split into four sections. In Section 4.1, we discuss our method for generating molecular dynamics trajectories. In Section 4.2, we assess the importance of exchange-correlation functional choice, comparing functionals with (optB86b-vdW [54]) and without (PBE) vdW interactions. In Section 4.3 we compare catalytically active (platinum) and inert (graphite) surfaces (including graphene). Lastly, in Section 4.4, we discuss the effects of adding defects and dopants into graphitic surfaces.

4.1 Methods

Here, we provide the preparation steps followed to generate all of the water-solid interfaces used in our analysis. For the liquid water-graphite and liquid water-graphene interfaces, we used supercell dimensions $12.41 \times 12.81 \times 37.36$ Å and $12.41 \times 12.81 \times 23.96$ Å respectively. For the liquid water-graphite interface, we used 300 C atoms to form 5 layers. In the liquid water-graphene interface, we used 60 C atoms. In addition to the pristine surfaces, we have also constructed interfaces with defects or dopant materials. To model defects, we considered two classes. In the first class, we removed 2 carbon atoms from the surface layer(s) of both graphene and graphite. In the second class, we modified the crystalline structure at the surface(s) of graphene and graphite to create Stone-Wales (5-7) defects. To model doping, we replaced a surface carbon atom with either a nitrogen or boron atom in the surface layer(s). For all liquid water-graphitic interfaces, we used 100 D_2O molecules, a $1 \times 1 \times 1$ k-point grid centered about the Γ point, and a plane wave energy cutoff of 500 eV. The use of heavy water allowed for the increase of the time step by 40% (8 a.u. to 11.3 a.u.). We used the exchange-correlation functional optB86b-vdW and a Nosé-Hoover thermostat (T = 300, 330, and 400 K). For the liquid water-platinum interface, we constructed two distinct platinum(111)-D_2O interfaces, both with supercell dimensions of $11.24 \times 9.74 \times 45.69$ Å, 112 platinum atoms, and 100 D_2O molecules. All of these interfaces are visualized in Figure 4.1 For one of the interfaces, we used the PBE exchange correlation functional, and for the other interface, we used the optB86b-vdW exchange correlation functional. We found that both the PBE and optB86b-vdW exchange correlation functionals

give the same lattice constant for bulk platinum. When using the optB86b-vdW functional, we ran molecular dynamics at T = 300, 330, 400, 500, 600, 650, 700, 750, 800, and 1000 K. When using the PBE exchange correlation functional, we ran molecular dynamics at T = 330, 400, and 1000 K. For both interfaces, we used a plane wave energy cutoff of 500 eV as well as a $2 \times 2 \times 1$ k-point grid centered about the Γ point. When constructing the liquid water-platinum interfaces, we chose to use 7 layers of platinum with the 3 middle layers constrained along the z axis. Since we used periodic boundary conditions, the liquid water interacts with two metal faces (assuming no vacuum layer). The two faces should be identical such that the supercell is symmetric. It should also be noted that when placing the water next to the metal surface, an exclusion volume must be included. This is a volume of empty space that occurs naturally and is dependent on the metal. When Cicero *et al.* [5] constructed their liquid water-graphene interface, the thickness (in the z direction) of the exclusion volume was $\sim$ 2 Å. For the liquid water-platinum interfaces, we chose the thickness of the exclusion volume to be $\sim$ 2.35 Å. For the liquid water-graphite and liquid water-graphene interfaces, we chose the thickness of the exclusion volume to be $\sim$2.5 Å. After the supercells were constructed, we then structurally relaxed the systems to eliminate the random configurations of the water molecules. Afterwards, we ran molecular dynamics and monitored the pressure, temperature, and total energy of the systems. Even after the structural relaxations, a few ps of data showed significant fluctuation as the systems approached their equilibrium states. This data was not considered in our analysis. Once the systems reached their equilibrium states, we collected 10 ps of data for analysis. We found that the structure of the density profile converges

at $\approx$ 10 ps, which can be seen in Figure 4.3. To about 5 Å away from the surface, the water has a similar structure. Beyond 5 Å, the water fluctuates about 1.1 g/cc. The same amount of fluctuations can be seen when comparing different MD trajectories of pure water. The convergence of the density profile is much slower than other distributions (i.e. a radial distribution function). In Figure 4.3, we also show the O-H radial distribution function which is well converged through 5 ps. For some of our structural analysis, we have also included error bars which give the standard error of the mean. Subsequent snapshots of molecular dynamics are highly correlated. To remove the correlation, we performed our analysis such that statistical distributions were recursively split and averaged over (block averaging). All of the molecular dynamics reported were done using the canonical (NVT) ensemble, heavy water (D_2O), and Nosé-Hoover thermostats. Some of the NVT calculations were also replicated in the microcanonical (NVE) ensemble to verify that the thermostat did not bias the results. We used VASP [68] for all simulations. In total, over 250 ps of BOMD was generated for this report.

Additional total energy calculations were also run for the water-graphite and water-platinum interfaces where only one water molecule was considered on the surface. The supercells had the same cell dimensions as mentioned above. For these calculations, structural minimizations were done using a $4 \times 4 \times 2$ k-point grid centered about the Γ point and an atomic force convergence of 10^{-4} eV/Å. The exchange-correlation functional used was optB86b-vdW.

In our analysis, we chose a minimum temperature of 400 K for the MD due to earlier studies done on pure liquid water [72, 73]. Schwegler *et al.* [72] found when

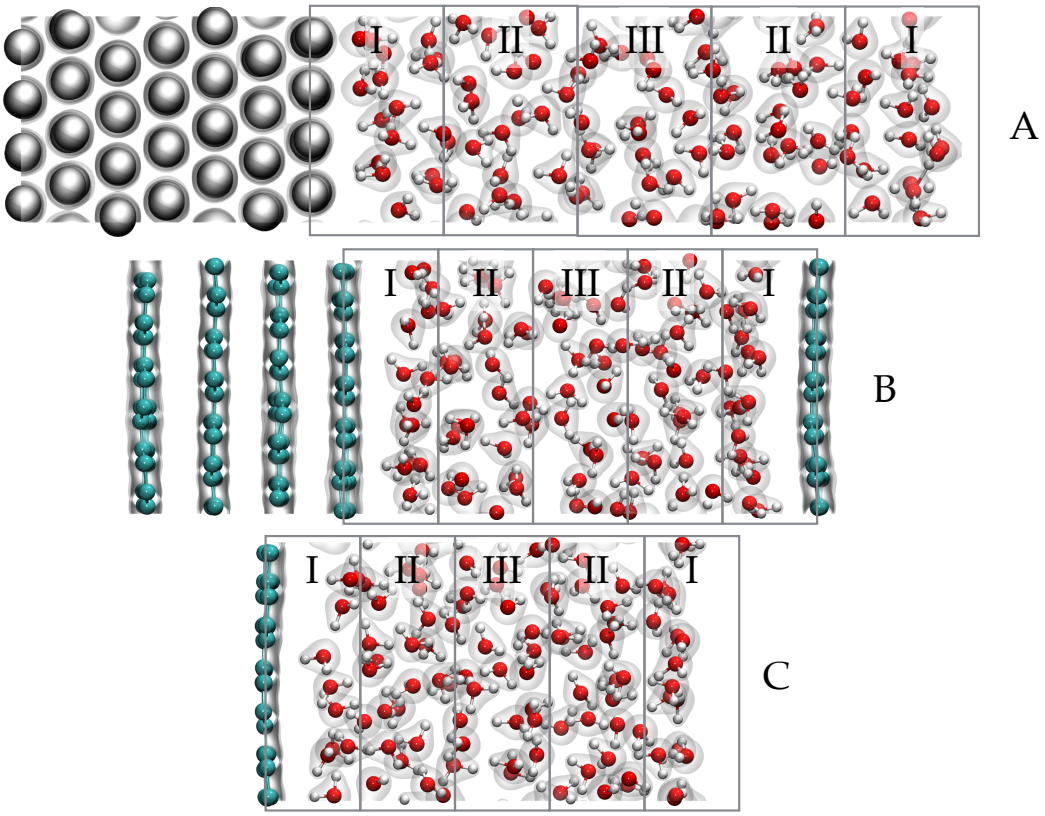

FIGURE 4.1: Visualizations of the liquid water-platinum interface (A), liquid water-graphite interface (B), and liquid water-graphene interface (C) used in the DFT calculations. Here, the transparent surfaces around the atoms are the computed self-consistent charge densities for the geometries seen.

using the PBE exchange correlation functional at 300 K, the water is overstructured after examining radial distribution functions, and the diffusion coefficient is lower than the reported value using experimental techniques. They found that increasing the temperature approximated the inclusion of proton quantum effects in their calculations. In a study done by Morales *et al.* [74], they found that excluding the nuclear quantum effects of ions leads to artificially low displacements at low temperatures. In Figure 4.3, we show the mean squared displacement for liquid water in a cubic cell at 400 K. Here, the diffusion coefficient was calculated to be 2.0×10^{-9} $\mathrm{m^2s^{-1}}$, which is slightly lower than the experimental value of 2.3×10^{-9} $\mathrm{m^2s^{-1}}$ found from neutron scattering [70].

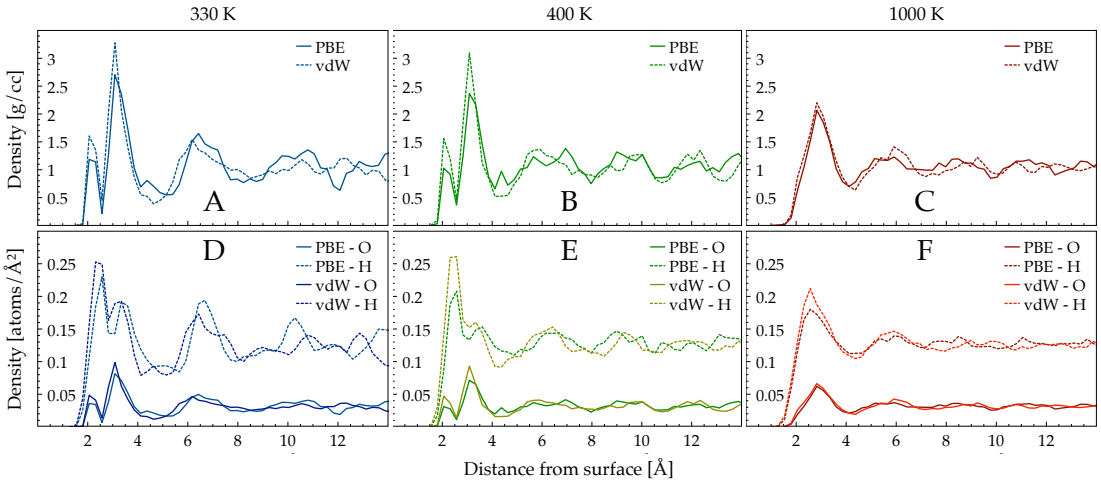

FIGURE 4.2: Molecular (top 3 figures labelled A, B, and C) and atomic (bottom 3 figures labelled as D, E, and F) density profiles for the liquid water-platinum interfaces at 330 K, 400 K, and 1000 K. The top two curves in the molecular density profiles are from two independent MD trajectories, one with vdW interactions (labelled as vdW) and one without (labelled as PBE). The bottom 4 curves are from the same MD trajectories, but have been split up by either O or H atoms to give atomic density profiles.

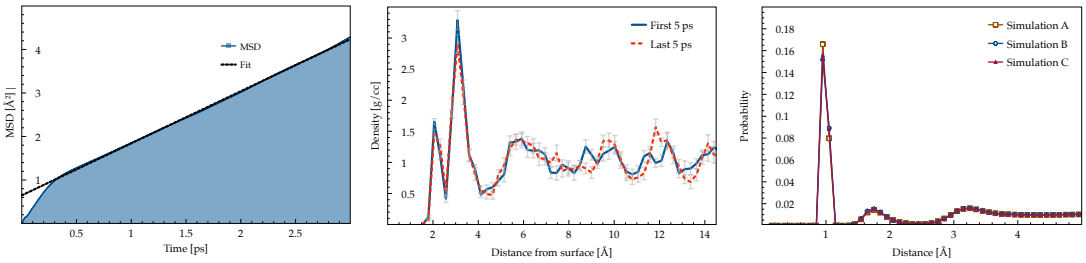

FIGURE 4.3: Left: Mean squared displacement of 100 water molecules in a cubic supercell (14.41^3 Å^3) with a thermostat of 400 K using the vdW-optB86b exchange correlation functional. The fit of the asymptotic behaviour gives a diffusion coefficient of 2.0×10^{-9} m^{2}s^{-1}, which is slightly lower than the experimental value of 2.3×10^{-9} m^{2}s^{-1}. Middle: Molecular density profiles of a 10 ps long simulation (with a thermostat of 400 K) of a liquid water-platinum interface using the vdW-optB86b exchange correlation functional. 10 ps of MD was generated, and then divided in half to give two trajectories, each 5 ps long labelled as 'First 5 ps' or 'Last 5 ps'. The error bars are the standard error found from analyzing the molecular density profile at each time step. Right: O-H radial distribution functions for liquid water-platinum interfaces at 330 K using the PBE functional. 3 independent MD trajectories, each 5 ps long, were generated and are labelled as simulation A, B, and C. Here, it is clear that the O-H radial distribution functions have converged when comparing the different simulations.

4.2 Assessing the importance of van der Waals interactions

Recently, it has been shown that vdW forces have a significant contribution in the interaction between water molecules and metals. Carrasco *et al.* [6] calculated the adsorption energy of a water molecule on a platinum surface and found that it is almost halved when excluding such interactions (-403 meV with vdW interactions (optB88-vdW) and -217 meV without (PBE)). Reports with only one or two layers of water molecules next to a metal surface (ice-metal interfaces) are common in the literature [75–77] due to their smaller system sizes and lower computational cost. A computational study of the magnitude we present is missing. We begin our study by examining the structure of the water molecules next to the platinum surface. In Figure 4.2, we plot the molecular and atomic density profiles as a function of distance from the nearest platinum surface. Immediately adjacent to the surface, we find high density water. This is a common feature of water-electrode interfaces, and has been found using X-ray scattering [20] and *ab initio* methods [5, 18, 23]. The first notable result is the difference between the profiles when comparing the PBE functional with the optB86b-vdW functional. For the optB86b-vdW functional, the largest peaks have heights 15-29% greater than the largest peaks for the PBE functional (depending on temperature). This suggests that there is an increased attraction between the platinum atoms and water molecules when vdW interactions are introduced. This is consistent with the work done by Carraso *et al.* [6]. The second notable result is the appearance

of an initial wetting peak before the largest peak in the density. This peak has not been seen in density profiles of other water-solid interfaces [5, 18, 23], and is absent in the liquid water-graphite and liquid water-graphene interfaces covered in this report. This initial peak is from water molecules chemisorbing to the surface. Past experimental work done using electron energy loss spectroscopy [76], and ultraviolet/X-ray photoemission [78] also found chemisorption at low temperatures. When increasing the temperature of the system, the peak remains at 400 K, but vanishes at 1000 K. This suggests that the energy of the platinum-H_2O bond is somewhere in the range of k_BT where 400 K $< T <$ 1000 K. We ran subsequent molecular dynamic simulations with a thermostat of 500, 600, 650, 700, 750, and 800 K; we find that the initial wetting peak starts to decrease in size around 600 K, and is almost completely gone at 800 K. This is consistent with previous thermal desorption spectroscopy experiments done for oxygen-platinum [79] and hydrogen-platinum [80] interfaces. Here, it was found that hydrogen desorbs around 420 K, but oxygen desorbs at 850 K. At higher temperature (i.e. beyond the boiling point), the system is clearly in a non-equilibrium state, but still gives insight into the nature of the chemisorption layer.

In Figure 4.4, we look at the normalized probability of finding a surface water molecule with a certain orientation. To normalize, we first considered the orientation probability distribution of a randomly oriented water molecule. We then calculated the orientation probability distribution for a particular system and divided by the randomly oriented probability distribution. This is analogous to the normalization procedure in the calculation of a radial distribution function. In

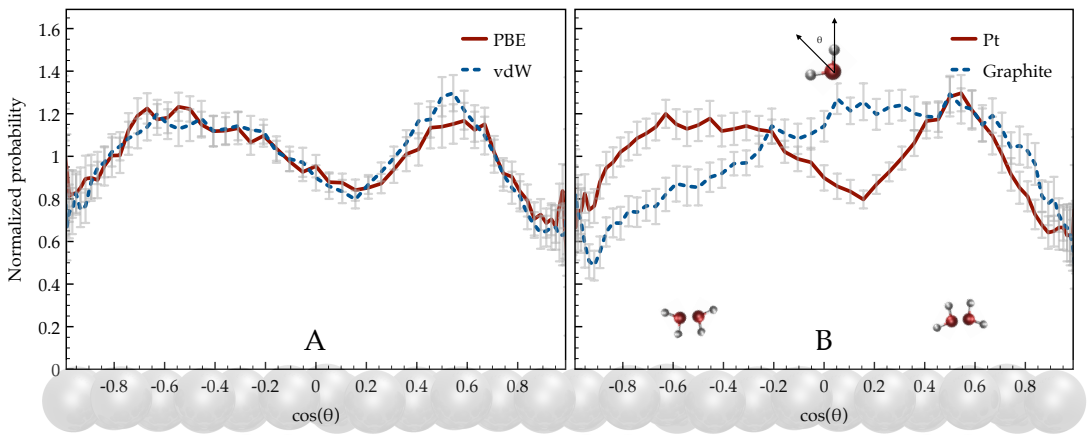

FIGURE 4.4: Normalized probability distributions of a water molecule having a certain orientation. In Figure 4.1, we divided the supercell into regions (I, II, and III) separating bulk from surface water molecules. Here, we only show region I; region II and III show little structure and a more random orientation. Here, θ is the angle between the geometric dipole to the z-normal. In the left figure (labelled as A), we compare 2 independent water-platinum interfaces; one with vdW interactions (labelled as vdW) and one without (labelled as PBE). In the right figure (labelled as B), we compare a water-platinum (labelled as platinum) and water-graphite interface (labelled as Graphite), both with vdW interactions.

Figure 4.1, we outline regions within the supercell to differentiate surface water molecules from bulk ones. For both the PBE and optB86b-vdW simulations, there is a similar trend for surface water molecules. The water molecules orient themselves so that either an oxygen (O-platinum bond) or hydrogen (H-platinum bond) faces the surface. In another study done by Carrasco *et al.* [81], they found that the preferred orientation for a water molecule sitting on 4d metal surfaces (Ru(0001), Rh(111), Pd(111), Ag(111)) is with the geometric dipole almost flat to the surface and the oxygen over the atop site. When bulk water is introduced, surface water molecules bond with other water molecules within the surface layer, or other water molecules further away from the surface (in the bulk). Due to the interaction with the metal surface as well as bulk water, the surface water molecules constantly re-orientate themselves to bind with the surface as well as adjacent water molecules.

The environment in the interfacial layer is highly competitive. Noting the differences between the PBE and the optB86b-vdW simulations, for PBE, there is a much sharper peak where the oxygen atom is facing towards the surface. In Table 4.1, we highlight the probability of finding water molecules (within a distance of 2.5 Å and 5.0 Å from the surface) with an O-down (both hydrogens further away from the surface), O-up (both hydrogens closer to the surface), or mixed orientation. Interestingly, there is very little difference when comparing the functional choice. Closer to the surface ($<$ 2.5 Å away), no water molecule has an O-up orientation. The water molecules will either have an O-down, or mixed orientation with equal probability. These results are similar to the results found by Velasco *et al.* [18] for a liquid water-Au interface. They found that 49% of the time surface water molecules orientate themselves to have an orientation where the geometric dipole of the water molecule is parallel with the surface. In this orientation, the lone-pairs of the oxygen interact with the surface orbitals, and the hydrogens form intermolecular bonds with adjacent molecules. This could be thought of in our framework as the O-down orientation. Velasco *et al.* also found that 49% of the time, water molecules would orientate themselves to have one hydrogen facing the surface, and the other participating in the hydrogen bonding network. This is analogous to our mixed orientation. Along with the theoretical study done by Velasco *et al.*, X-ray absorption spectroscopy revealed that unsaturated hydrogen bonds occur at the surface. The changes in the spectrum are simply due to the reorganization of water molecules at the surface once the electrode is introduced. Looking further away from the surface ($<$ 5.0 Å away), the O-down and mixed probabilities decrease equally, and O-up molecules are seen with a low frequency.

Description	O-down probability $\pm\ \sigma$	O-up probability $\pm\ \sigma$	Mixed probability
platinum (PBE - 2.5 Å)	$0.50 \pm 1.6 \times 10^{-5}$	0.0	$0.50 \pm 1.6 \times 10$
platinum (PBE - 5.0 Å)	$0.45 \pm 2.2 \times 10^{-3}$	$0.10 \pm 4.3 \times 10^{-3}$	$0.45 \pm 2.2 \times 10$
platinum (vdW - 2.5 Å)	$0.50 \pm 7.2 \times 10^{-5}$	0.0	$0.50 \pm 7.6 \times 10$
platinum (vdW - 5.0 Å)	$0.45 \pm 2.2 \times 10^{-3}$	$0.10 \pm 4.5 \times 10^{-3}$	$0.45 \pm 2.2 \times 10$
Graphite (2.5 Å)	$0.50 \pm 6.0 \times 10^{-3}$	0.0	$0.50 \pm 6.1 \times 10$
Graphite (5.0 Å)	$0.44 \pm 1.9 \times 10^{-3}$	$0.12 \pm 3.8 \times 10^{-3}$	$0.44 \pm 1.9 \times 10$
Graphene (2.5 Å)	$0.50 \pm 4.9 \times 10^{-3}$	0.0	$0.50 \pm 4.9 \times 10$
Graphene (5.0 Å)	$0.43 \pm 1.8 \times 10^{-3}$	$0.13 \pm 3.6 \times 10^{-3}$	$0.44 \pm 1.8 \times 10$

TABLE 4.1: The probabilities for finding water molecules with O-down (both hydrogens further away from the surface), O-up (both hydrogens closest to the surface), or other in the surface layers of water next to the metal surfaces. Here, σ is the standard error of the mean.

Next, we examine the dynamics of the water molecules. In Figure 4.5, we plot the mean squared displacement and a probability distribution we call the network reorganization probability function which we label as $\Pi(t)$. $\Pi(t)$ gives the likelihood of finding the same neighbouring water molecules as a function of time. This type of function has been used in previous literature to study pure hydrogen [71], and gives similar distributions to water dipole rotational autocorrelation functions reported by Cicero *et al.* [5]. For the mean squared displacements, there is a clear difference between the optB86b-vdW and PBE simulations. The use of the PBE functional causes the water molecules to be much more mobile, and therefore increases the likelihood of breaking bonds as water molecules diffuse away. For $\Pi(t)$, the temperature difference for both the optB86b-vdW and PBE simulations is identical. In Figure 4.5, we can see that the likelihood of finding the same nearest neighbour decreases as you increase temperature. As the temperature is

increased, more energy is introduced into the system which causes the kinetic energy of the water molecules to be greater than the energy associated with forming an intermolecular hydrogen bond. This explains why the curves are shifted.

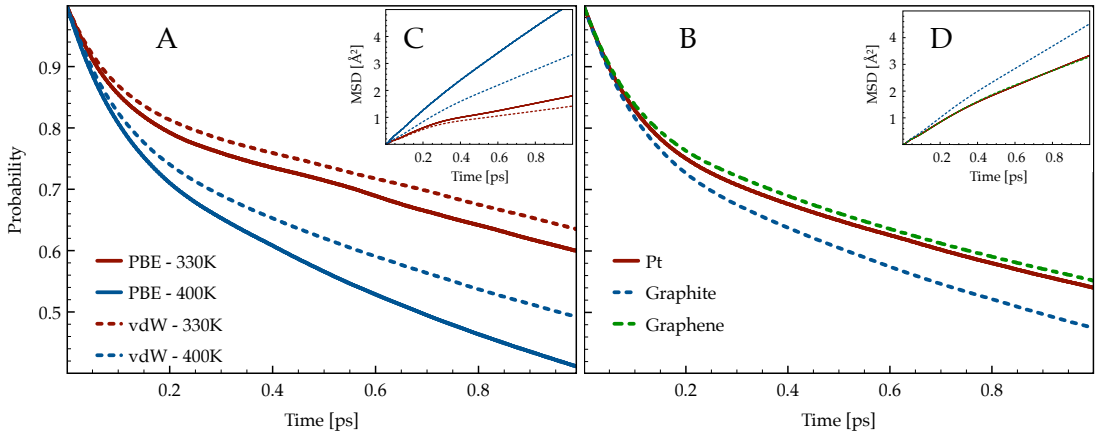

FIGURE 4.5: Network reorganization probability functions ($\Pi(t)$) giving the likelihood of finding the same neighbouring water molecules as a function of time (labelled A and B), and the mean squared displacements (subfigures) for selected simulations (labelled C and D). In the left plot, we compare the liquid water-platinum interfaces with vdW interactions (labelled as vdW) and without (labelled as PBE) as well as temperature effects (330 and 400 K). In the right plot we compare the liquid water-platinum (labelled as platinum), liquid water-graphite (labelled as Graphite), and liquid water-graphene (labelled as Graphene) interfaces, all with a thermostat of 330 K using the optB86b-vdW exchange correlation functional.

4.3 Comparing common electrodes

In this section, we compare the liquid water-platinum interface with the liquid water-graphite and liquid water-graphene interfaces (using the optB86b-vdW exchange correlation functional). Graphene has shown extraordinary structural and electronic properties due to its massless Dirac fermions [82]. As mentioned previously, an exceptional larger scale water-graphene theoretical study was done by Cicero *et al.* [5] using a PBE functional (without vdW interactions) to examine the overall structure and hydrogen-bonding network in closer detail. In another

report, Li *et al.* [8] studied how the adsorption of a single water molecule effects the electronic structure of metal-supported graphene. Here, they used DFT with vdW interactions (using optB86b-vdW) and found that the π and π^* bands of graphene are not strongly perturbed by water adsorption in the case of a strong graphene-metal contact. Due to the stacked layers of graphite interacting solely through vdW forces, it can only be described correctly using non-local exchange correlation functionals. It is only the recent development of vdW exchange correlation functionals that allows for an accurate description of graphite in the DFT framework.

The first notable difference between the platinum and the graphite/graphene density profiles (Figure 4.6) is the initial peak next to the platinum surface ($\approx$ 2 Å from the surface). This peak is absent for both the graphite and graphene interfaces. Given the adsorption geometries and band structures of single water molecules adsorbed to the surfaces of Ru(0001), Rh(111), and Pd(111) [81], we suspect a chemisorption peak to be present for these surfaces. For platinum, water is well known to chemisorb to the surface, and the subsequent structural minimization calculation we performed for a single water molecule next to the platinum surface confirmed that a covalent bond forms on the atop site. For graphite, water does not chemisorb. When we structurally minimized the system with a single water molecule on the graphite surface, the water molecule is approximately 1 Å further away from the surface than the single water molecule sitting on a platinum slab, and the preferred adsorption site is the hollow site. This is consistent with the work done by Ambrosetti *et al.* [83]. They found that the optimal bond

length for a single water molecule on a graphite surface is greater than 3.2 Å, with the water molecule sitting on the hollow site. When calculating the adsorption energy of water on graphite, Ambrosetti *et al.* found very weak adsorption values ($63.5 \leq |E_{\text{ads}}| \leq 143.8$ meV depending on the exchange-correlation functional). In the case of platinum, binding energies on the platinum surface (-403 meV [6]) are almost 3 times the binding energy on graphite when using a vdW exchange correlation functional (-143.8 meV [83]). In the case of gold, the adsorption energy of a single water monomer was also significantly less than platinum (-281 meV [6]). When examining the PDOS of the p orbitals on the O atom, and the p_z band of the graphite surface atoms, it looks like a bond would be favourable given the overlap of energies of the $1b_1$ (highest occupied p orbital) molecular orbital and the p_z band (Figure 4.6). The platinum d band and the $1b_1$ orbital of water also overlap. The $1b_1$ orbital couples with the d band causing the peak to broaden in the distribution. Although the energies of the p_z orbitals and the $1b_1$ orbital coincide for the graphite slab, there is no coupling between the states (no broadening in the distribution for the $1b_1$ orbital). For the liquid water-graphite and liquid water-graphene density profiles, the largest peaks occur at about 3 Å away from the surface, consistent with the density profiles reported by Cicero *et al.* for bulk water next to a graphene slab at 330 K [5]. The peaks are slightly shifted in comparison to the density profile of the liquid water-platinum interface, consistent with the adsorption geometries of the single water molecules next to the surfaces. Comparing the density profile of the liquid water-graphite interface with the liquid water-graphene interface, the second largest peak in the density profile

of the water-graphite interface is slightly larger than the initial peak of the water-graphene interface. This indicates that the extra layers in graphite increase the attraction of water molecules to the surface. This is consistent with Ambrosetti *et al.* [83], where they considered a single water molecule next to graphite and graphene and found the water molecule sits closer to the graphite surface.

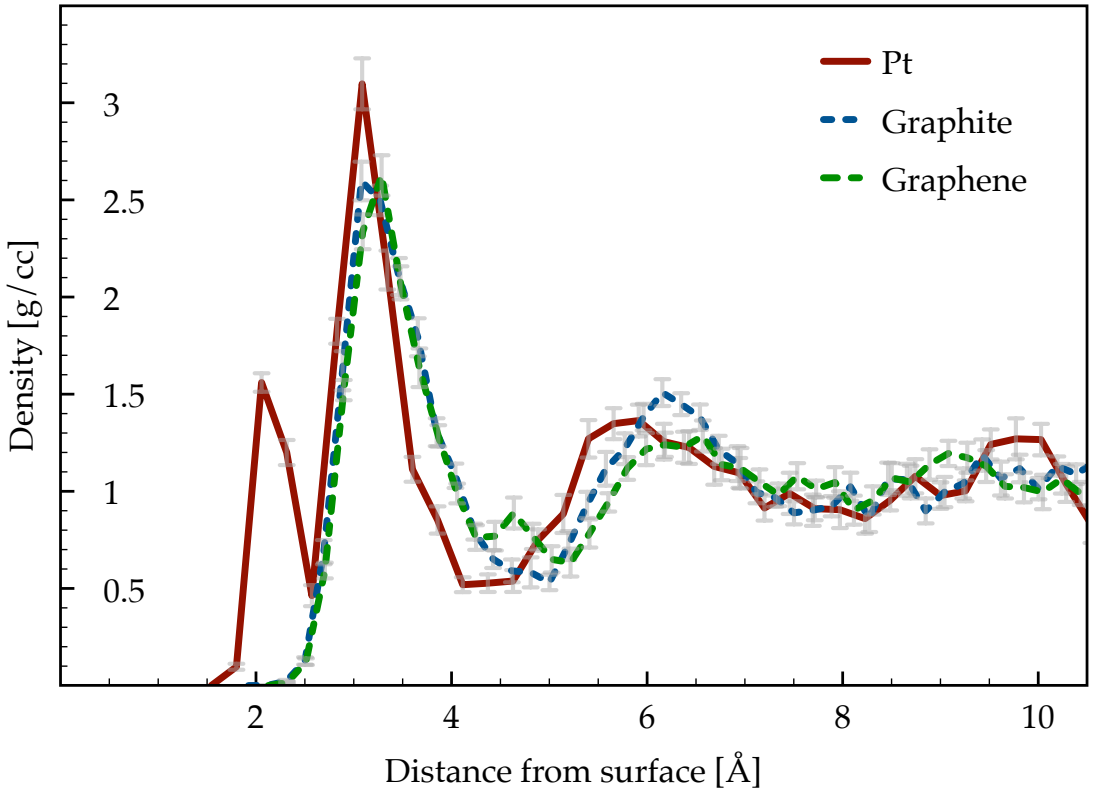

FIGURE 4.6: Molecular density profiles for the liquid water-platinum (labelled at platinum), liquid water-graphite (labelled at Graphite), and liquid water-graphene (labelled as Graphene) interfaces. For all three interfaces, the simulations were run with a thermostat of 400 K, and the optB86b-vdW exchange-correlation functional was used.

When comparing the orientation of the water molecules (Figure 4.4), the water molecules next to the graphite surface show little preferred orientation (this was identical for the graphene interface). The maximum of the distribution gives an O-down orientation but the absence of peaks or valleys in the distribution suggests no preferred orientation. When Cicero *et al.* [5] examined the orientation of bulk

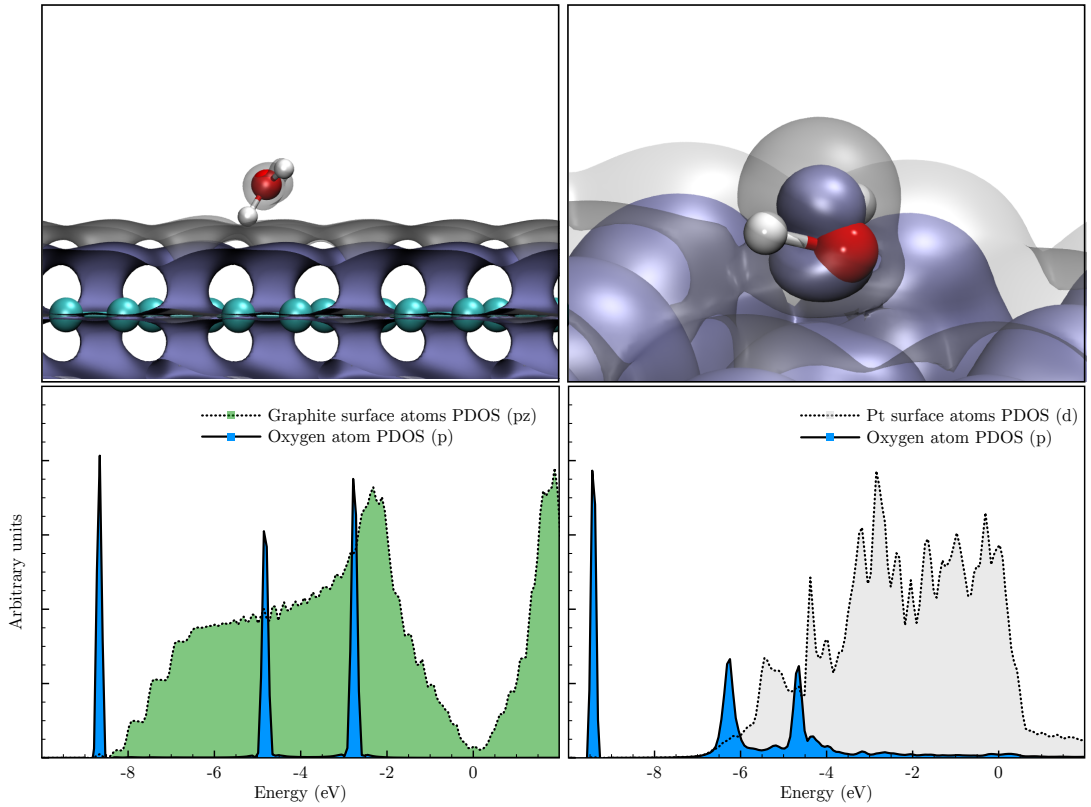

FIGURE 4.7: (Colour online) Visualizations of the HOMO orbital (top) for the graphite (left) and platinum (right) surface as well as the PDOS (bottom) for the graphite p_z orbitals, platinum d orbitals, and O p orbitals. The opaque purple surface (1.1e-5) and the transparent black surface (8.2e-5) are at different isovalues.

water next to graphene, they found that the preferred orientation was one hydrogen pointing down towards the surface, with the other hydrogen contributing to the local hydrogen bonding network in the surface layer. This is consistent with the geometry of the single water molecule sitting on the graphite surface, where we found this orientation energetically preferable. Looking to Table 4.1, the probabilities of finding O-down, O-up, and mixed orientations of water molecules in the surface layer have the same trend when next to the platinum slab. The maximum of the orientation distribution most likely arises due to the water molecules closest to the surface, where 50% of the time the oxygen faces the surface with both H atoms participating in the hydrogen bonding network.

Next, we compare the dynamics of the water molecules. Looking to Figure 4.5, the mean squared displacements are similar, but the diffusion of water molecules is greatest for the graphite interface, and almost identical for the liquid water-platinum and liquid water-graphene interfaces. As in the case with the mean squared displacements, the probability functions, $\Pi(t)$, produces similar curves for all 3 interfaces. The hydrogen bonding network is most active for the liquid water-graphite interface and again almost identical for the liquid water-platinum and liquid water-graphene interfaces. In this case, a more active hydrogen bonding network implies that intermolecular hydrogen bonds break and new bonds form (with other molecules) at a faster rate. In the work done by Cicero *et al.* [5], they computed a water dipole rotational autocorrelation function which dropped off exponentially as a function of time. The faster the decay of the distribution indicates more mobility and reorientation of the water molecules. Interestingly, these dipole rotational autocorrelation functions are remarkably similar to $\Pi(t)$. This indicates a strong correlation between nearest neighbour dynamics and the rotational autocorrelation function of the dipole vector of a water molecule. Physically, this is intuitive as the orientation of a water molecule is highly dependent on the neighbouring water molecules. When the orientation of a water molecule drastically changes due to thermal fluctuations, it is possible that a new hydrogen bond will form with a different neighbouring water molecule. This would cause local restructuring in the hydrogen bonding network, in turn causing the dipole vectors of the local water molecules to reorient themselves.

4.4 The effects of defects and dopants in graphitic surfaces

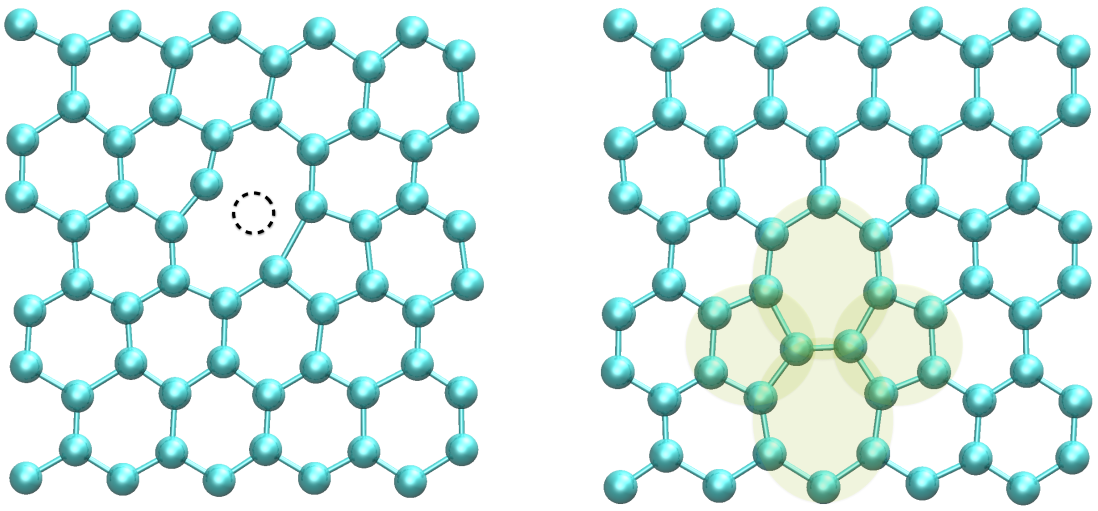

FIGURE 4.8: Left: Graphene surface with a single point defect (labelled with the dotted circle). Right: Graphene surface with a Stone-Wales defect (labelled with a yellow transparent colour.)

Current experimental growth mechanisms of graphitic surfaces sometimes lead to concentrations of defects [84]. These sites break symmetry in the lattice, leading to extra electrons or holes, which may lead to an increase in surface activity. In addition to increasing surface activity of inert materials via defects, many studies involving graphitic surfaces and dopant materials have been done to try to improve the catalytic performance of the electrode [85–88]. In particular, nitrogen and boron doped graphene has been shown to exhibit p-type and n-type semiconducting behaviour [85]. To explore both of these scenarios, we have completed a preliminary study of liquid water-graphene and liquid water-graphite surfaces where single point defects, Stone-Wales defects, and dopant materials (B and N) have been introduced at the surface. Examples of single point defects and Stone-Wales defects are shown in Figure 4.8.

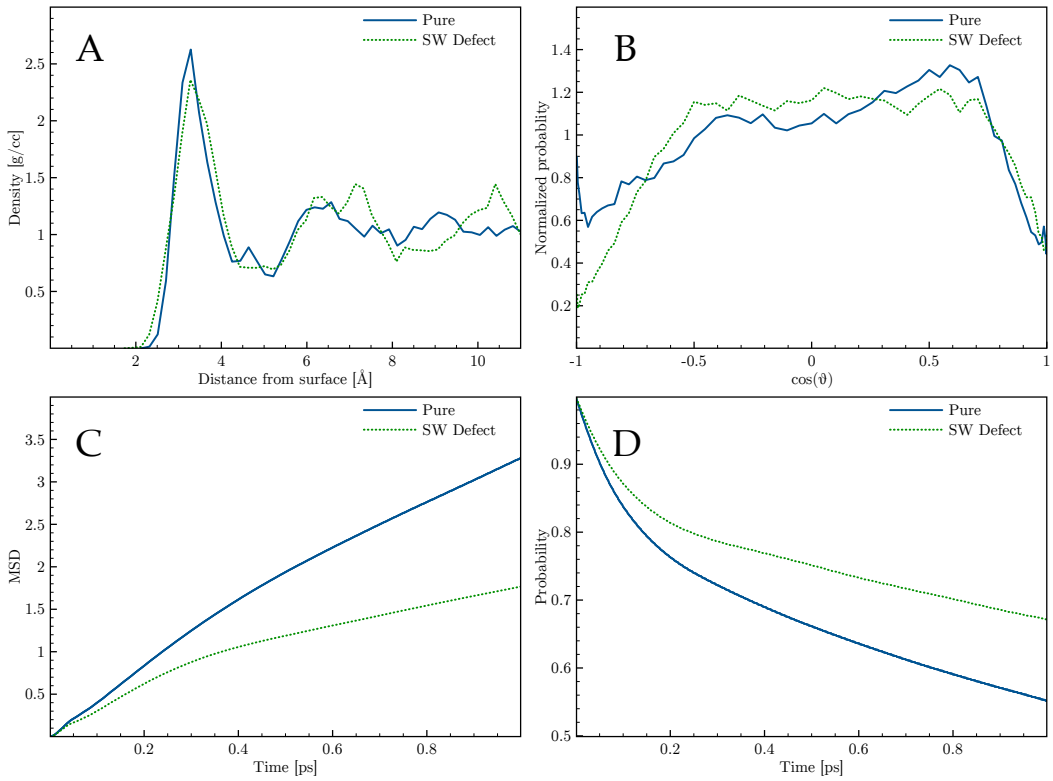

FIGURE 4.9: Comparing selected distributions for the liquid water-graphene interfaces with and without Stone-Wales defects at a temperature of 400 K using the vdW-optB86b exchange correlation functional. In the top left (labelled as A) we show the density profiles. In the top right (labelled as B) we show the normalized orientation distributions. In the bottom left (labelled as C) we show the mean squared displacements. In the bottom right (labelled as D) we show the network reorientation probability functions ($\Gamma(t)$).

At a temperature of 400 K, we found that for most of the simulations, the structure and dynamics of the liquid water subtly change when the defects/dopants are introduced. Interestingly, for the Stone-Wales defects simulations of the liquid water-graphene interface, we found a substantial decrease in the mobility of the water molecules as well as an almost random orientation of water molecules at the surface in comparison to all other pristine graphitic surfaces analyzed (see Figure 4.9). As presented previously, the orientation of water molecules next to the pristine graphitic surfaces is mostly random, with a slight preference for having one oxygen facing towards the surface. When the Stone-Wales defect is

introduced into the graphene surface, this preferred orientation disappears, leading to a mostly random orientation of water molecules at the surface. The absence of a preferred orientation indicates the lack of surface activity, or even repulsion at the surface. This repulsion pushes the surface water molecules towards the bulk, and therefore confines the liquid water leading to the decrease of the mobility of the water molecules. This is in agreement with [89], where they found that Stone-Wales defects in single-wall carbon nanotubes are less reactive than the pristine counterpart. The Stone-Wales defect in the graphite surface did not produce the same effects as the graphene surface. The structure and dynamics of the water molecules is almost identical with pristine graphite.

Chapter 5

Machine learning potential energy surfaces

5.1 Motivation

Although DFT is one of the cheapest electronic structure methods, the computational cost of the eigenvalue problem still scales as $\mathcal{O}(N^3)$, where N is the number of grid points in real space (assuming a $N \times N \times N$ grid). When we start investigating large-scale systems with thousands of electrons, finding a self-consistent solution at every time step becomes computationally demanding. This is where machine learning can provide a solution. In machine learning, the computational cost scales as $\mathcal{O}(N)$, where N is the number of weights in the network. This decrease in complexity makes large-scale calculations feasible and the trained networks will rival the accuracy of the method they are trained on. The caveat of

machine learning is that you must provide a training set before a machine learned model can be used. If the training set consists of DFT calculations, then you must deal with the computational cost of DFT before training the machine learning model. Once trained, the machine learned models will be very cheap to calculate some property of an atomic system. As an example, a machine learned model can be paired with a Monte Carlo method to provide accurate total energy calculations for a large-scale atomic system. We give an example of this in the following Chapter.

5.2 Some basics of ANNs

ANNs or multilayer perceptrons, are one of, if not the most topologically simple form of a neural network. They consist of an input layer, usually a few hidden layers, and an output layer. When training an ANN, one wishes to map a set of input vectors $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n\}$ to a set of output vectors (or scalars) $\{\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_n\}$ (or $\{y_1, y_2, \ldots, y_n\}$). The input and output layers of the ANN are just our input and output vectors, where a bias value may be added in for the input layer. Therefore the dimensions of these layers are simply the dimensions of the vectors. The hidden layers is where all of the learning is done. The dimensionality of the hidden layers is arbitrary, and it gives the number of neurons/weights that connect to the previous layer. The dimensionality of the hidden layers requires trial and error to minimize the error. Within each neuron is an activation function. The activation function takes the inner product of the weights and outputs from the previous

layer, and returns a scalar, which can then be used to compute other inner products in the next layer. To give a visual representation, consider Figure 5.1. Here, we have an ANN with 2 hidden layers, each with 3 neurons. To calculate the activation of the neuron, we perform the inner product of the associated weights with the activations of the previous layer. The result of the inner product is then passed into the logistic function. The logistic function is one of many choices for an activation function.

Once we have propagated through the network for one input vector (forward

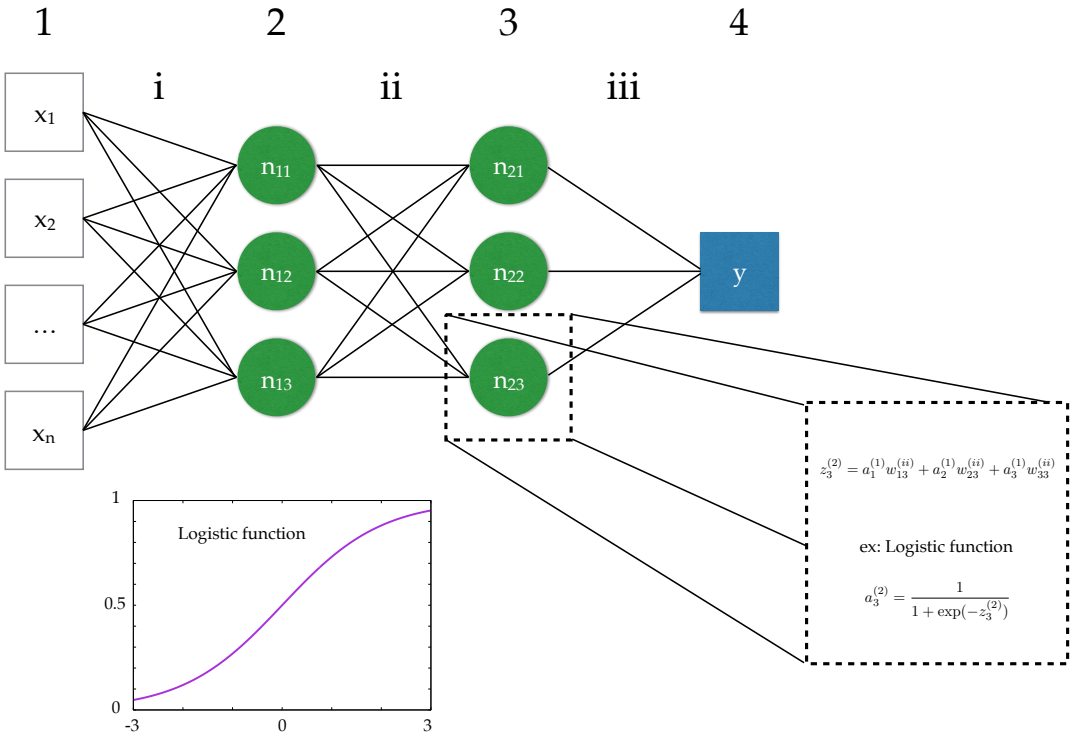

FIGURE 5.1: Schematic representation of an ANN. The layers of the ANN are labelled by 1,2,3, and 4. The weight matrices are labelled by i, ii, and iii. The dotted square shows how the inner product is constructed, and then activated by the logistic function.

pass), we then obtain either some vector or value. To train the weights on the

hidden layers, we first look at the error of the output

$$E = \frac{1}{2}|\mathbf{Y} - \mathbf{Y}_{\text{true}}|^2, \tag{5.1}$$

which we want to minimize. To do so, we rely on optimization methods like gradient descent where the gradient of the error with respect to the weights must be calculated. Consider the weight highlighted in Figure 5.1, we want to compute derivative of the error with respect to that weight, and follow the negative gradient to find a minima in the weight space. The last set of weights (labelled as 'iii' in Figure 5.1) have the most clean gradients, and in vectorized form, they are

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}^{(iii)}} &= \frac{\partial E}{\partial \mathbf{a}_j^{(3)}} \frac{\partial \mathbf{a}_j^{(3)}}{\partial \mathbf{z}_j^{(3)}} \frac{\partial \mathbf{z}_i^{(3)}}{\partial w_{ij}^{(iii)}} \\ &= \delta_j^{(3)} \mathbf{a}_j^{(2)} \end{aligned} \tag{5.2}$$

where, in this case $\mathbf{a}^{(3)} \equiv \mathbf{Y}$, and we have set

$$\delta_j^{(3)} \equiv \frac{\partial E}{\partial \mathbf{a}_j^{(3)}} \frac{\partial \mathbf{a}_j^{(3)}}{\partial \mathbf{z}_j^{(3)}} = |\mathbf{a}^{(3)} - \mathbf{Y}_{\text{true}}| f(\mathbf{z}_j^{(3)})(1 - f(\mathbf{z}_j^{(3)}), \tag{5.3}$$

where $f(\mathbf{z})$ is the logistic function shown in Figure 5.1. For the middle set of weights (labelled as 'ii' in Figure 5.1), the gradients are

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}^{(ii)}} &= \frac{\partial E}{\partial \mathbf{a}_j^{(3)}} \frac{\partial \mathbf{a}_j^{(3)}}{\partial \mathbf{a}_j^{(2)}} \frac{\partial \mathbf{a}_j^{(2)}}{\partial \mathbf{z}_j^{(2)}} \frac{\partial \mathbf{z}_j^{(2)}}{\partial w_{ij}^{(ii)}} \\ &= \frac{\partial E}{\partial \mathbf{a}_j^{(3)}} \frac{\partial \mathbf{a}_j^{(3)}}{\partial \mathbf{z}_j^{(3)}} \frac{\partial \mathbf{z}_j^{(3)}}{\partial \mathbf{a}_j^{(2)}} \frac{\partial \mathbf{a}_i^{(2)}}{\partial \mathbf{z}_j^{(2)}} \frac{\partial \mathbf{z}_j^{(2)}}{\partial w_{ij}^{(ii)}} \\ &= \delta_j^{(3)} \zeta_j^{(2)} \mathbf{a}_j^{(1)} \end{aligned} \tag{5.4}$$

where we define

$$\zeta_j^{(2)} \equiv \sum_i w_{ij}^{(iii)} f(\mathbf{z}_j^{(2)})(1 - f(\mathbf{z}_j^{(2)}). \tag{5.5}$$

Computing the gradients for the last set of weights (labelled as 'i' in Figure 5.1), we get

$$\begin{aligned}
\frac{\partial E}{\partial w_{ij}^{(i)}} &= \frac{\partial E}{\partial \mathbf{a}_j^{(3)}} \frac{\partial \mathbf{a}_j^{(3)}}{\partial \mathbf{a}_j^{(2)}} \frac{\partial \mathbf{a}_j^{(2)}}{\partial \mathbf{z}_j^{(2)}} \frac{\partial \mathbf{z}_j^{(2)}}{\partial \mathbf{a}_j^{(1)}} \frac{\partial \mathbf{a}_j^{(1)}}{\partial \mathbf{z}_j^{(1)}} \frac{\partial \mathbf{z}_j^{(1)}}{\partial w_{ij}^{(i)}} \\
&= \frac{\partial E}{\partial \mathbf{a}_j^{(3)}} \frac{\partial \mathbf{a}_j^{(3)}}{\partial \mathbf{z}_j^{(3)}} \frac{\partial \mathbf{z}_j^{(3)}}{\partial \mathbf{a}_j^{(2)}} \frac{\partial \mathbf{a}_j^{(2)}}{\partial \mathbf{z}_j^{(2)}} \frac{\partial \mathbf{z}_j^{(2)}}{\partial \mathbf{a}_j^{(1)}} \frac{\partial \mathbf{a}_j^{(1)}}{\partial \mathbf{z}_j^{(1)}} \frac{\partial \mathbf{z}_j^{(1)}}{\partial w_{ij}^{(i)}} \\
&= \delta_j^{(3)} \zeta_j^{(2)} \zeta_j^{(1)} \mathbf{a}_j^{(0)}
\end{aligned} \tag{5.6}$$

where we similarly define

$$\zeta_j^{(1)} \equiv \sum_i w_{ij}^{(ii)} f(\mathbf{z}_j^{(1)})(1 - f(\mathbf{z}_j^{(1)}). \tag{5.7}$$

We can see, that by adding more and more hidden layers, we simply need to keep adding on additional ζ terms to update our weights. We forward pass the data through the network, compute the error, and move backwards layer by layer to update the weights. This method is called back propagation.

Throughout this derivation, I have assumed that the length of each layer is the same for brevity. This is normally not true in practice, and you must keep track of all the indices correctly in the partial derivatives.

5.3 Constructing neural network potentials

In atomic systems, we either want to know the total energy, or the gradient of the total energy with respect to a spacial coordinate to determine forces. Since we anticipate to use Metropolis MC, we only need to predict energies. Therefore, if we have a configuration of atoms, we would like to be able to predict the total energy for that configuration. To train, we need to generate hundreds, if not thousands of configurations for our particular atomic system as well as the energies associated for those configurations. To use an ANN, we firstly need to map the atomic configurations to a feature vector. We follow the formalism developed by Behler *et al.* [90], where the total energy is a sum over single atom energies,

$$E = \sum_i \epsilon_i. \tag{5.8}$$

This means that we will have a different ANN potential for every atomic species in our configuration. Now, we need a feature vector for every atom in our atomic configuration. This feature vector represents the local environment for an atom, and it should inherit properties of atomic systems. In an atomic system, if a particular configuration is translated or rotated, the energy does not change. Also, if two atoms of the same species are swapped, we should also get the same energy. Again, following the work of [90], our feature vector comes from discretizing 2 functions that satisfy the aforementioned requirements. With a cut off function

defined as

$$f_c(r_{ij}) = \left\{ \begin{array}{cc} \frac{1}{2}\left[\cos\left(\frac{\pi r_{ij}}{r_c}\right) + 1\right] & \text{if } r_{ij} \leq r_c \\ 0 & \text{if } r_{ij} > r_c \end{array} \right\} \tag{5.9}$$

our feature vector is built from the functions

$$G_i^{(1)} = \sum_{j \neq i} \exp[-\eta(r_{ij} - r_s)^2] f_c(r_{ij}), \tag{5.10}$$

and

$$\begin{aligned} G_i^{(2)} &= 2^{1-\zeta} \sum_{j \neq i} \sum_{k \neq i,j} (1 + \lambda \cos\theta_{ijk})^\zeta \\ &\quad \times \exp[-\eta(r_{ij}^2 + r_{ik}^2 + r_{jk}^2)] f_c(r_{ij}) f_c(r_{ik}) f_c(r_{jk}). \end{aligned} \tag{5.11}$$

Here, r_c is a cutoff distance, η, r_s, ζ, and λ are all tuneable parameters, $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|^2$, and $\theta_{ijk} = \frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{ik}}{r_{ij} r_{ik}}$. To discretize these functions and use them as our feature vectors, one chooses a discrete set of values for each of the tuneable parameters.

This exact formalism has been implemented in aenet (atomic energy network) [2], which we leverage in the work completed for Subsection 5.5. Aenet generates high dimensional neural network potentials based on a set of atomic configurations and associated energies.

5.4 Metropolis Monte Carlo

As mentioned previously, to calculate the thermodynamic average of some observable of an atomic system, one can either use MD or MC. MC methods have been used steadily since it's introduction at the end of the second world war. The term "Monte Carlo" was coined by Metropolis in his work at Los Alamos, due to the extensive use of random numbers [91]. One of the more well known uses of MC is MC integration, and the popular example of calculating the value of π (see Figure 5.2). To calculate π, we calculate the euclidian distance $r = \sqrt{x^2 + y^2}$, where x and y are generated randomly on the interval $[0, 1]$, and count the number of times r falls within the unit circle. With C_{in} defined as the number of times $r \leq 1$, and C_{out} the number of times $r > 1$, we can see that $\pi = 4\frac{C_{\text{in}}}{C_{\text{out}}}$ based on the area of a circle.

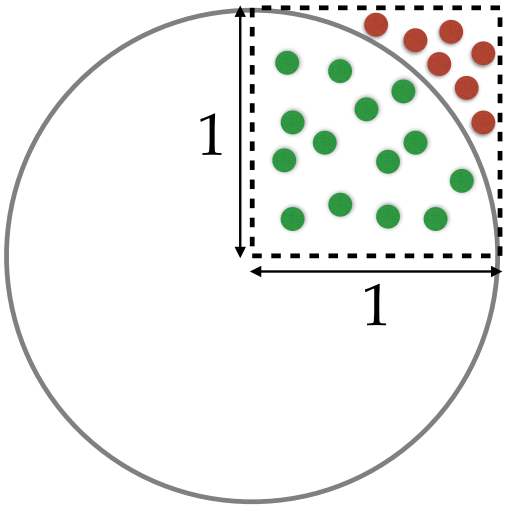

FIGURE 5.2: A schematic view of how to calculate π from generating random numbers. The green dots inside the circle count towards C_{in}, and the red dots that fall outside the circle count towards C_{out}.

In Metropolis MC applied to atomic systems, we wish to sample configuration

space subject to some Hamiltonian. To do so, we evaluate the Hamiltonian based on an atomic configuration to obtain the energy for that system. We then consider Boltzmann factors of the energy, and make a decision whether or not the atomic configuration is favourable or not (see Figure 5.3).

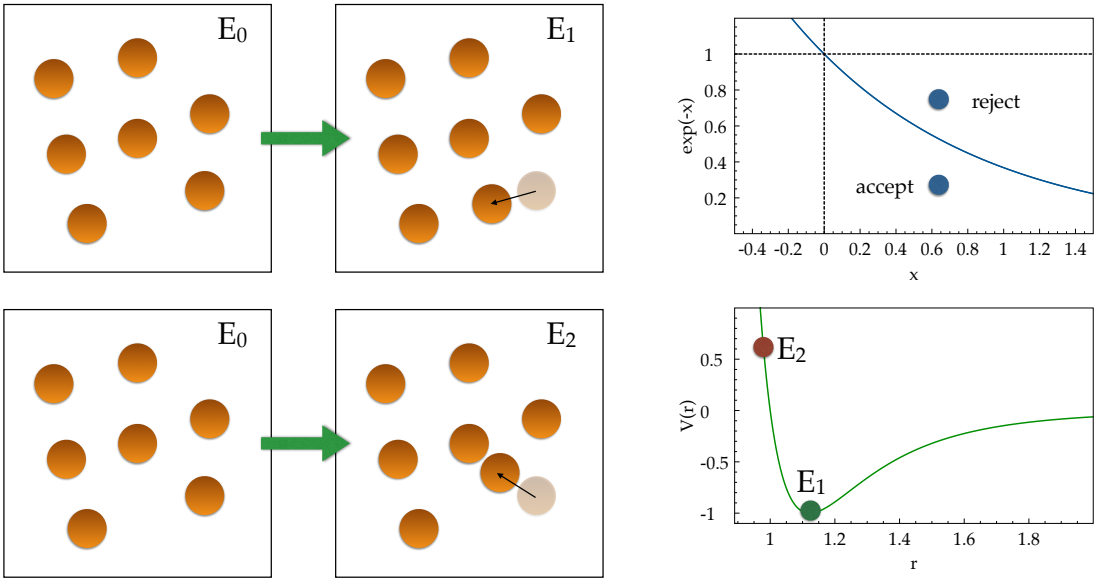

FIGURE 5.3: Left: 2 dimensional atomic system where a single particle move is accepted (E_1) and rejected (E_2). Since the proposed particle move has such a large positive value, it is highly likely to be rejected. Right: Exponential function that shows how the acceptance probability changes as a function difference in energy (x). The LJ potential is also shown where E_1 is close to the bottom of the well, and E_2 climbs up the well to a positive energy, indicating repulsion.

Consider the 2-particle Lennard-Jones (LJ) potential seen in Figure 5.3

$$v(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right] \tag{5.12}$$

where ϵ and σ are parameters that can change based on the atomic species. Here, r is the interatomic distance between 2 atoms. For a LJ system, the interaction

energy of the n-particle system can be written as

$$E = \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} v(r_{ij}) \tag{5.13}$$

where r_{ij} is the interatomic distance between the ith and jth particle. Now consider an atomic configuration with an interaction energy E_0. We then randomly move one of the particles in the original atomic configuration, which gives us a new interaction energy E_1. If the move is downhill in energy, i.e. $E_1 < E_0$, then this configuration is accepted. If not, we consider accepting the configuration with probability p_1/p_0, where

$$\frac{p_1}{p_0} = \frac{Z^{-1} \exp(-\beta E_1)}{Z^{-1} \exp(-\beta E_0)} = \exp(-\beta[E_0 - E_1]). \tag{5.14}$$

Here, Z is the partition function, and $\beta = (k_B T)^{-1}$, with k_B defined as the Boltzmann constant, and T, the temperature. We can see from equation 5.14, that at a higher temperature, we will have an increased probability of moving uphill in energy. To do this programmatically, consider the following Python code snippet.

```
import random

r_move = 0.1
beta = 1.0
n_trials = 1e6
n_particles = 100
particles = []

# calculate the total energy
total_E = E(particles)

for i in range(n_trials):
  # select a random particle
  atom = random.uniform(n_particles)
```

```
# displace particle randomly
particles[atom][0] += (2.0 * random.random() - 1.0) * r_move
particles[atom][1] += (2.0 * random.random() - 1.0) * r_move
particles[atom][2] += (2.0 * random.random() - 1.0) * r_move

# get new energy and difference
trial_E = E(particles)
delta_E = trial_E - total_E

if delta_E < 0.0:
  # accept
  total_E = trial_E
elif exp(-beta * delta_e) > random.random():
  # accept
  total_E = trial_E
else:
  # revert coordinates and try again
  revert(particles)
```

Surprisingly, this is all you need to do Metropolis MC for atomic systems. The complexity is hidden in the total energy calculation. For the 2-body LJ potential, it is efficient to compute due to the form, and we only need to consider the energies for the pairs involving the particle that has been moved. If the total energy comes from a DFT calculation, then using Metropolis MC is unfeasible to calculate valid thermodynamic averages. We simply could not compute enough atomic configurations.

5.5 A toy example

As a proof of principle, we present a toy problem where we can leverage a machine learned potential energy surface. Consider a dimer molecule which interacts through the LJ potential

$$v(r) = \frac{4}{r^{12}} - \frac{4}{r^6}. \tag{5.15}$$

type	η	λ	ζ	r_s	r_c
$G^{(1)}$	0.003214	-	-	0.0	5.0
$G^{(1)}$	0.035711	-	-	0.0	5.0
$G^{(1)}$	0.071421	-	-	0.0	5.0
$G^{(1)}$	0.124987	-	-	0.0	5.0
$G^{(1)}$	0.214264	-	-	0.0	5.0
$G^{(1)}$	0.357106	-	-	0.0	5.0
$G^{(1)}$	0.714213	-	-	0.0	5.0
$G^{(1)}$	1.428426	-	-	0.0	5.0
$G^{(2)}$	0.028569	1	1.0	-	5.0
$G^{(2)}$	0.089277	1	1.0	-	5.0
$G^{(2)}$	0.000357	-1	1.0	-	5.0
$G^{(2)}$	0.028569	-1	1.0	-	5.0
$G^{(2)}$	0.089277	-1	1.0	-	5.0
$G^{(2)}$	0.000357	1	2.0	-	5.0
$G^{(2)}$	0.028569	1	2.0	-	5.0
$G^{(2)}$	0.089277	1	2.0	-	5.0
$G^{(2)}$	0.000357	-1	2.0	-	5.0
$G^{(2)}$	0.028569	-1	2.0	-	5.0
$G^{(2)}$	0.089277	-1	2.0	-	5.0
$G^{(2)}$	0.000357	1	4.0	-	5.0
$G^{(2)}$	0.028569	1	4.0	-	5.0
$G^{(2)}$	0.089277	1	4.0	-	5.0
$G^{(2)}$	0.000357	-1	4.0	-	5.0
$G^{(2)}$	0.028569	-1	4.0	-	5.0
$G^{(2)}$	0.089277	-1	4.0	-	5.0

TABLE 5.1: Table of parameters used to discretize the atomic fingerprint functions. These parameters were also used in [2] to describe the local environments of TiO_2.

This is equivalent to Equation 5.12, with $\epsilon = \sigma = 1$. Using this potential, we generate ten thousand random dimer configurations with r in the domain $[0.5, 5.0]$ Å. Using these dimer configurations, we then use the atomic fingerprint functions from Equations 5.10 and 5.11 to generate our input vectors for the ANN. We have shown all of the parameters used when constructing the fingerprint functions in Table 5.1.

Once the fingerprint functions were generated, we then proceeded to train the ANN potential. The topology of our ANN consisted of an input vector of length 26, 2 hidden layers with 10 neurons, and an output layer with one neuron. The activation functions used in the hidden layers were scaled hyperbolic functions with linear twisting, i.e.

$$f(x) = 1.7158 \tanh\left(\frac{2}{3}x\right) + x, \tag{5.16}$$

and the activation function of the output layer was linear. In the optimization process, we used the Levenberg-Marquardt optimization scheme [92] to update the weights. While training, we split the data set up such that 70% of the data was for training, and 30% for testing. In Figure 5.4, we show the mean squared error (MSE), and the root mean squared error (RMSE), which clearly exhibits convergence after 10000 epochs. We can also see that the MSE and RMSE of the test set does not diverge, indicating no over fitting of the training set. We then use this ANN potential to generate atomic trajectories using both Metropolis MC, and MD generated using the canonical ensemble (Langevin method). After generating these trajectories, we then plot the RDF for both MC and MD, seen in Figure 5.5. We can clearly see that the machine learned potential energy surface does remarkably well compared to using the exact potential. The true and predicted energy values as a function of time are almost identical, and the RDFs are very similar. Future work includes investigating much more complex atomic systems, and using convolutional neural networks (CNNs). This means we must replace the input vector with a 2 dimensional image of our atomic system. We will rely

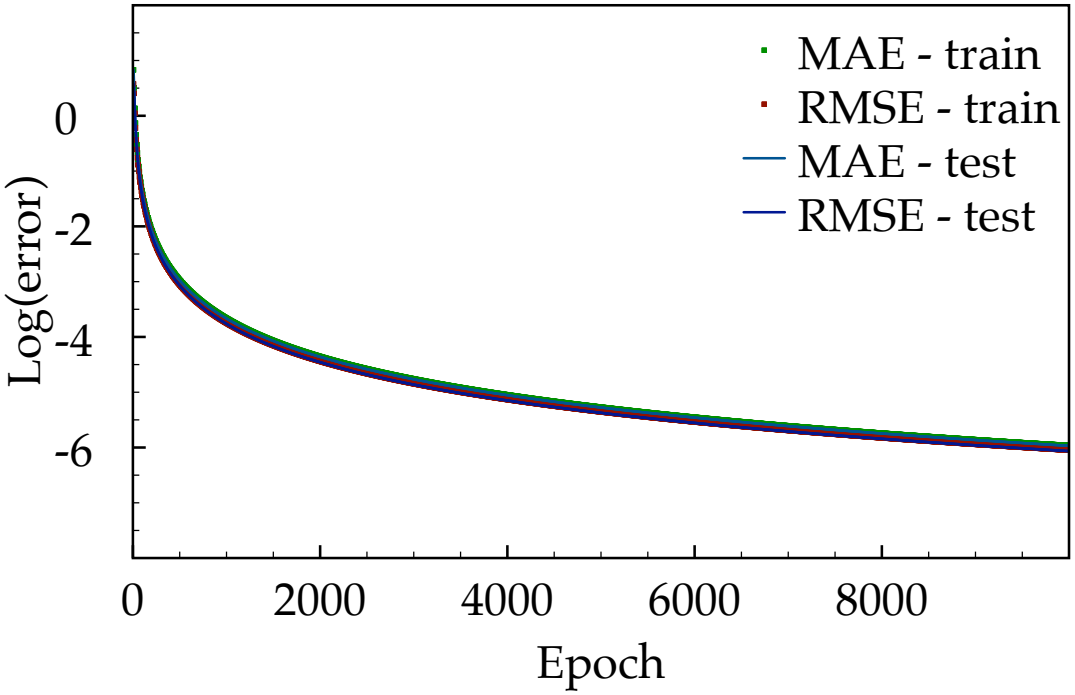

FIGURE 5.4: The mean average error as a function of epoch when training the dimer molecules for the training and testing sets. Since the testing loss curve does not diverge, we do not see overfitting. All of the curves are well converged at 10 thousand epochs.

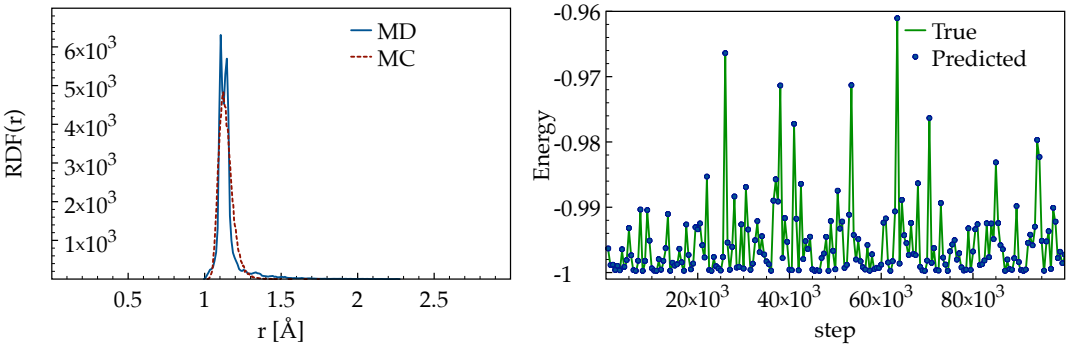

FIGURE 5.5: Left: The radial distribution functions for MD produced from the Langevin method, and for MC produced from the ANN potential. Right: True energy versus predicted energy for 100 frames of MD produced from the Langevin method.

on the CNNs to learn the features in the training process. CNNs have operations which are non-local, and therefore inherently include spatial dependencies. CNNs are very promising for studying atomic systems.

Chapter 6

Conclusion

In this thesis, we first described the fundamentals of electronic structure theory related to DFT, and then gave a 1D step-by-step guide for implementing a self-consistent DFT code. Afterwards, we describe how one can obtain molecular dynamics from DFT, and then show how structural and electronic observables can be calculated from molecular dynamics trajectories.

We then carried out large-scale molecular dynamic simulations for bulk water next to platinum, and doped/defect graphitic surfaces in ambient and high temperature conditions. After analysis of the density profile, we find water next to a platinum slab forms a chemisorption layer before the largest density peak. This feature was absent for the liquid water-graphitic interfaces, and has not been discovered in other studies of other water-solid interfaces (including a liquid water-Au interface). Noticing the disappearance of this chemisorption peak at 1000 K for the liquid-water platinum interface, subsequent non-equilibrium calculations were

run at temperatures in the range of 500-800 K. It was concluded that at 650 K, the peak begins to disappear. At 800 K, the peak has almost fully disappeared. When comparing the exchange correlation functional choice (i.e. with and without vdW interactions), there are differences. In the density profile, the highest peak can be 15-29% larger when the vdW interactions are included (depending on the temperature). The orientation of water molecules next to the surface have subtle differences but show similar structure, and the dynamics of the water molecules indicate a further attraction to the surface and less mobility when the vdW interactions are included.

When comparing the orientation of a single water molecule next to the platinum surface and the orientation of surface molecules in bulk water next to the platinum slab, there is a significant difference. For a single water molecule, the oxygen faces towards the atop site, with the geometric dipole of the molecule almost parallel to the surface. When analyzing the orientation of surface molecules in the bulk liquid, this is not always the case. The surface molecules interact with the metal atoms, as well as with other water molecules in their local environment. This, along with thermal fluctuations, cause the surface water molecules to constantly reorientate themselves in the competitive force field. For a single water molecule next to the graphite/graphene surface, there is little attraction to the surface. The optimal structure has a single hydrogen pointing down toward the surface and the distance between the water molecule and the surface is almost 1 Å greater than the distance between a single water molecule next to a platinum slab. The surface molecules in the bulk liquid next to graphite/graphene show little preference for

orientation, confirming the weak interaction between water molecules and the graphite/graphene surface. Although the PDOS of the p_z band for the surface C atoms overlaps with the p orbitals of O, there is no coupling between them. The PDOS of the d band for the surface atoms of platinum also overlaps with the p orbitals of O, and the states do couple. The p orbital of O, which coincides with the $1b_1$ orbital of water, broadens when next to the platinum surface.

When introducing defects and dopants into the liquid water-graphitic surfaces, we find that there are almost no changes in the structure and dynamics of the water molecules. The only notable difference came from Stone-Wales defects in the graphene surface, where we found a decrease in the mobility of water molecules. We propose this mobility decrease coincides with the decrease of surface activity, inevitably causing the surface water molecules to repel from the surface. Future work includes investigating other materials and optimizing the search for observables indicative of an efficient catalyst. When considering a new metal for water electrolysis, one should expect to observe chemisorption as well broadening in the peaks of the p orbitals in the PDOS of liquid water as an indication of an efficient catalyst.

Lastly, we described how to use ANNs to learn a potential energy surface and carry out a preliminary study of machine learning a simple potential energy surface with ANNs. We show that, for a dimer molecule with a Lennard-Jones potential, an accurate potential energy surface can be learned using ANNs. We then show that the RDF is similar between MD using the Langevin method, and MC with the ANN potential, again confirming the validity of the ANN potential. Future

work includes investigating more complex systems, and different neural network architectures.

Bibliography

[1] AK Soper. The radial distribution functions of water and ice from 220 to 673 k and at pressures up to 400 mpa. *Chemical Physics*, 258(2):121–137, 2000.

[2] Nongnuch Artrith and Alexander Urban. An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for tio2. *Computational Materials Science*, 114:135 – 150, 2016. ISSN 0927-0256. doi: https://doi.org/10.1016/j.commatsci.2015.11.047. URL http://www.sciencedirect.com/science/article/pii/S0927025615007806.

[3] Michael G Walter, Emily L Warren, James R McKone, Shannon W Boettcher, Qixi Mi, Elizabeth A Santori, and Nathan S Lewis. Solar water splitting cells. *Chemical reviews*, 110(11):6446–6473, 2010.

[4] Nathan S Lewis and Daniel G Nocera. Powering the planet: Chemical challenges in solar energy utilization. *Proceedings of the National Academy of Sciences*, 103(43):15729–15735, 2006.

[5] Giancarlo Cicero, Jeffrey C Grossman, Eric Schwegler, Francois Gygi, and Giulia Galli. Water confined in nanotubes and between graphene sheets: A

first principle study. *Journal of the American Chemical Society*, 130(6):1871–1878, 2008.

[6] Javier Carrasco, Jiří Klimeš, and Angelos Michaelides. The role of van der waals forces in water adsorption on metals. *The Journal of Chemical Physics*, 138(2):024708, 2013.

[7] Axel Groß, Florian Gossenberger, Xiaohang Lin, Maryam Naderian, Sung Sakong, and Tanglaw Roman. Water structures at metal electrodes studied by ab initio molecular dynamics simulations. *Journal of The Electrochemical Society*, 161(8):E3015–E3020, 2014.

[8] Xiao Li, Ji Feng, Enge Wang, Sheng Meng, Jiří Klimeš, and Angelos Michaelides. Influence of water on the electronic structure of metal-supported graphene: Insights from van der waals density functional theory. *Physical Review B*, 85(8):085425, 2012.

[9] CS Lin, RQ Zhang, ST Lee, M Elstner, Th Frauenheim, and LJ Wan. Simulation of water cluster assembly on a graphite surface. *The Journal of Physical Chemistry B*, 109(29):14183–14188, 2005.

[10] Limin Liu, Matthias Krack, and Angelos Michaelides. Density oscillations in a nanoscale water film on salt: insight from ab initio molecular dynamics. *Journal of the American Chemical Society*, 130(27):8572–8573, 2008.

[11] Li-Min Liu, Alessandro Laio, and Angelos Michaelides. Initial stages of salt crystal dissolution determined with ab initio molecular dynamics. *Phys. Chem. Chem. Phys.*, 13(29):13162–13166, 2011.

[12] Li-Min Liu, Changjun Zhang, Geoff Thornton, and Angelos Michaelides. Structure and dynamics of liquid water on rutile tio 2 (110). *Physical Review B*, 82(16):161415, 2010.

[13] Jie Ma, Angelos Michaelides, Dario Alfè, Laurids Schimka, Georg Kresse, and Enge Wang. Adsorption and diffusion of water on graphene from first principles. *Physical Review B*, 84(3):033402, 2011.

[14] Elizabeth C Beret, Rafael R Pappalardo, Nikos L Doltsinis, Dominik Marx, and Enrique Sánchez Marcos. Aqueous pdii and ptii: Anionic hydration revealed by car–parrinello simulations. *ChemPhysChem*, 9(2):237–240, 2008.

[15] Angelos Michaelides. Density functional theory simulations of water–metal interfaces: Waltzing waters, a novel 2d ice phase, and more. *Applied Physics A*, 85(4):415–425, 2006.

[16] Wei-Fu Chen, Kotaro Sasaki, Chao Ma, Anatoly I Frenkel, Nebojsa Marinkovic, James T Muckerman, Yimei Zhu, and Radoslav R Adzic. Hydrogen-evolution catalysts based on non-noble metal nickel–molybdenum nitride nanosheets. *Angewandte Chemie International Edition*, 51(25):6131–6135, 2012.

[17] Xiongce Zhao. Wetting transition of water on graphite: Monte carlo simulations. *Physical Review B*, 76(4):041402, 2007.

[18] Juan-Jesus Velasco-Velez, Cheng Hao Wu, Tod A Pascal, Liwen F Wan, Jinghua Guo, David Prendergast, and Miquel Salmeron. The structure of

interfacial water on gold electrodes studied by x-ray absorption spectroscopy. *Science*, 346(6211):831–834, 2014.

[19] Ying Jiang. Real-space imaging of interfacial water with submolecular resolution. *Bulletin of the American Physical Society*, 59, 2014.

[20] Michael F Toney, Jason N Howard, Jocelyn Richer, Gary L Borges, Joseph G Gordon, Owen R Melroy, David G Wiesler, Dennis Yee, and Larry B Sorensen. Voltage-dependent ordering of water molecules at an electrode–electrolyte interface. *Nature*, 368(6470):444–446, 1994.

[21] Adam P Willard, Stewart K Reed, Paul A Madden, and David Chandler. Water at an electrochemical interfacea simulation study. *Faraday Discussions*, 141:423–441, 2009.

[22] John P Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Physical Review Letters*, 77(18):3865, 1996.

[23] Rui Liu, Yongjing Lin, Lien-Yang Chou, Stafford W Sheehan, Wangshu He, Fan Zhang, Harvey JM Hou, and Dunwei Wang. Water splitting by tungsten oxide prepared by atomic layer deposition and decorated with an oxygen-evolving catalyst. *Angewandte Chemie*, 123(2):519–522, 2011.

[24] B Hammer and JK Norskov. Why gold is the noblest of all the metals. *Nature*, 376(6537):238–240, 1995.

[25] Steven Y Reece, Jonathan A Hamel, Kimberly Sung, Thomas D Jarvi, Arthur J Esswein, Joep JH Pijpers, and Daniel G Nocera. Wireless solar

water splitting using silicon-based semiconductors and earth-abundant catalysts. *Science*, 334(6056):645–648, 2011.

[26] Michele Orlandi, Roberto Argazzi, Andrea Sartorel, Mauro Carraro, Gianfranco Scorrano, Marcella Bonchio, and Franco Scandola. Ruthenium polyoxometalate water splitting catalyst: very fast hole scavenging from photogenerated oxidants. *Chemical Communications*, 46(18):3152–3154, 2010.

[27] Pingwu Du, Oleksandr Kokhan, Karena W Chapman, Peter J Chupas, and David M Tiede. Elucidating the domain structure of the cobalt oxide water splitting catalyst by x-ray pair distribution function analysis. *Journal of the American Chemical Society*, 134(27):11096–11099, 2012.

[28] Pingwu Du and Richard Eisenberg. Catalysts made of earth-abundant elements (co, ni, fe) for water splitting: recent progress and future challenges. *Energy & Environmental Science*, 5(3):6012–6021, 2012.

[29] Xinfu Xia and Max L Berkowitz. Electric-field induced restructuring of water at a platinum-water interface: A molecular dynamics computer simulation. *Physical Review Letters*, 74(16):3193, 1995.

[30] Adam P Willard, David T Limmer, Paul A Madden, and David Chandler. Characterizing heterogeneous dynamics at hydrated electrode surfaces. *The Journal of Chemical Physics*, 138(18):184702, 2013.

[31] Masatoshi Osawa, Minoru Tsushima, Hirokazu Mogami, Gabor Samjeské, and Akira Yamakata. Structure of water at the electrified platinum-water

interface: A study by surface-enhanced infrared absorption spectroscopy. *The Journal of Physical Chemistry C*, 112(11):4248–4256, 2008.

[32] Christopher P Smith, Mayumi Maeda, Ljiljana Atanasoska, Henry S White, and DJ McClure. Ultrathin platinum films on mica and the measurement of forces at the platinum/water interface. *The Journal of Physical Chemistry*, 92(1):199–205, 1988.

[33] K Heinzinger. Molecular dynamics studies of platinum/water interfaces. *Pure and Applied Chemistry*, 63(12):1733–1742, 1991.

[34] H Ogasawara, B Brena, D Nordlund, M Nyberg, A Pelmenschikov, LGM Pettersson, and A Nilsson. Structure and bonding of water on pt (111). *Physical Review Letters*, 89(27):276102, 2002.

[35] David T Limmer, Adam P Willard, Paul A Madden, and David Chandler. Water exchange at a hydrated platinum electrode is rare and collective. *The Journal of Physical Chemistry C*, 119(42):24016–24024, 2015.

[36] Minoru Otani, Ikutaro Hamada, Osamu Sugino, Yoshitada Morikawa, Yasuharu Okamoto, and Tamio Ikeshoji. Structure of the water/platinum interface—-a first principles simulation under bias potential. *Physical Chemistry Chemical Physics*, 10(25):3609–3612, 2008.

[37] G Nagy and K Heinzinger. A molecular dynamics simulation of electrified platinum/water interfaces. *Journal of Electroanalytical Chemistry and Interfacial Electrochemistry*, 296(2):549–558, 1990.

[38] Jacqueline Berges, Isabelle Fourre, Julien Pilme, and Jiri Kozelka. Quantum chemical topology study of the water-platinum (ii) interaction. *Inorganic Chemistry*, 52(3):1217–1227, 2013.

[39] Anna V Sberegaeva, Wei-Guang Liu, Robert J Nielsen, William A Goddard III, and Andrei N Vedernikov. Mechanistic study of the oxidation of a methyl platinum (ii) complex with o2 in water: Ptiime-to-ptivme and ptiime-to-ptivme2 reactivity. *Journal of the American Chemical Society*, 136(12): 4761–4768, 2014.

[40] Kuan-Yu Yeh, Michael J Janik, and Janna K Maranas. Molecular dynamics simulations of an electrified water/pt (111) interface using point charge dissociative water. *Electrochimica Acta*, 101:308–325, 2013.

[41] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library.* " O'Reilly Media, Inc.", 2008.

[42] Nuria M Oliver, Barbara Rosario, and Alex P Pentland. A bayesian computer vision system for modeling human interactions. *IEEE transactions on pattern analysis and machine intelligence*, 22(8):831–843, 2000.

[43] Andrea Vedaldi and Brian Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1469–1472. ACM, 2010.

[44] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech

recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[45] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012.

[46] John C Snyder, Matthias Rupp, Katja Hansen, Klaus-Robert Müller, and Kieron Burke. Finding density functionals with machine learning. *Physical review letters*, 108(25):253002, 2012.

[47] K Mills, M Spanner, and I Tamblyn. Deep learning and the schr\" odinger equation. *arXiv preprint arXiv:1702.01361*, 2017.

[48] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98: 146401, Apr 2007. doi: 10.1103/PhysRevLett.98.146401. URL https://link.aps.org/doi/10.1103/PhysRevLett.98.146401.

[49] Pierre Hohenberg and Walter Kohn. Inhomogeneous electron gas. *Physical review*, 136(3B):B864, 1964.

[50] Walter Kohn and Lu Jeu Sham. Self-consistent equations including exchange and correlation effects. *Physical review*, 140(4A):A1133, 1965.

[51] John C Slater. The theory of complex spectra. *Physical Review*, 34(10):1293, 1929.

[52] Teepanis Chachiyo. Communication: Simple and accurate uniform electron gas correlation energy for the full range of densities, 2016.

[53] Max Dion, Henrik Rydberg, Elsebeth Schröder, David C Langreth, and Bengt I Lundqvist. Van der waals density functional for general geometries. *Physical review letters*, 92(24):246401, 2004.

[54] Jiří Klimeš, David R Bowler, and Angelos Michaelides. Van der waals density functionals applied to solids. *Physical Review B*, 83(19):195131, 2011.

[55] Felix Bloch. Über die quantenmechanik der elektronen in kristallgittern. *Zeitschrift für Physik*, 52(7):555–600, Jul 1929. ISSN 0044-3328. doi: 10.1007/BF01339455. URL https://doi.org/10.1007/BF01339455.

[56] Richard M Martin. *Electronic structure: basic theory and practical methods*. Cambridge university press, 2004.

[57] Georg Kresse and Jürgen Furthmüller. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Physical review B*, 54 (16):11169, 1996.

[58] R. P. Feynman. Forces in molecules. *Phys. Rev.*, 56:340–343, Aug 1939. doi: 10.1103/PhysRev.56.340. URL https://link.aps.org/doi/10.1103/PhysRev.56.340.

[59] Loup Verlet. Computer" experiments" on classical fluids. ii. equilibrium correlation functions. *Physical Review*, 165(1):201, 1968.

[60] Shuichi Nosé. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of chemical physics*, 81(1):511–519, 1984.

[61] William G Hoover. Canonical dynamics: equilibrium phase-space distributions. *Physical review A*, 31(3):1695, 1985.

[62] NW Ashcroft. Electron-ion pseudopotentials in metals. *Physics Letters*, 23 (1):48–50, 1966.

[63] Conyers Herring. A new method for calculating wave functions in crystals. *Physical Review*, 57(12):1169, 1940.

[64] Leonard Kleinman and DM Bylander. Efficacious form for model pseudopotentials. *Physical Review Letters*, 48(20):1425, 1982.

[65] DR Hamann, M Schlüter, and C Chiang. Norm-conserving pseudopotentials. *Physical Review Letters*, 43(20):1494, 1979.

[66] Peter E Blöchl. Projector augmented-wave method. *Physical review B*, 50 (24):17953, 1994.

[67] Leandro Martínez, Ricardo Andrade, Ernesto G Birgin, and José Mario Martínez. Packmol: a package for building initial configurations for molecular dynamics simulations. *Journal of computational chemistry*, 30(13):2157–2164, 2009.

[68] G Kresse and J Furthmüller. Software vasp, vienna (1999). *Phys. Rev. B*, 54 (11):169, 1996.

[69] Albert Einstein. *Investigations on the Theory of the Brownian Movement*. Courier Corporation, 1956.

[70] MARIA SPYROU. *The Diffusion Coefficient of Water*. PhD thesis, University of Surrey.

[71] Isaac Tamblyn and Stanimir A. Bonev. Structure and phase boundaries of compressed liquid hydrogen. *Phys. Rev. Lett.*, 104:065702, Feb 2010. doi: 10.1103/PhysRevLett.104.065702. URL http://link.aps.org/doi/10.1103/PhysRevLett.104.065702.

[72] Eric Schwegler, Jeffrey C Grossman, François Gygi, and Giulia Galli. Towards an assessment of the accuracy of density functional theory for first principles simulations of water. ii. *The Journal of Chemical Physics*, 121(11):5400–5409, 2004.

[73] Tuan Anh Pham, Tadashi Ogitsu, Edmond Y Lau, and Eric Schwegler. Structure and dynamics of aqueous solutions from pbe-based first-principles molecular dynamics simulations. *The Journal of Chemical Physics*, 145(15):154501, 2016.

[74] Miguel A Morales, Jeffrey M McMahon, Carlo Pierleoni, and David M Ceperley. Towards a predictive first-principles description of solid molecular hydrogen with density functional theory. *Physical Review B*, 87(18):184107, 2013.

[75] Ralf Ludwig. How does water bind to metal surfaces: hydrogen atoms up or hydrogen atoms down? *Angewandte Chemie International Edition*, 42(30): 3458–3460, 2003.

[76] Brett A Sexton. Vibrational spectra of water chemisorbed on platinum (111). *Surface Science*, 94(2-3):435–445, 1980.

[77] A Michaelides and P Hu. Catalytic water formation on platinum: A first-principles study. *Journal of the American Chemical Society*, 123(18):4235–4242, 2001.

[78] Galen B Fisher and John L Gland. The interaction of water with the pt (111) surface. *Surface Science*, 94(2-3):446–455, 1980.

[79] JL Gland and VN Korchak. The adsorption of oxygen on a stepped platinum single crystal surface. *Surface Science*, 75(4):733–750, 1978.

[80] K Christmann, G Ertl, and T Pignet. Adsorption of hydrogen on a pt (111) surface. *Surface Science*, 54(2):365–392, 1976.

[81] Javier Carrasco, Angelos Michaelides, and Matthias Scheffler. Insight from first principles into the nature of the bonding between water molecules and 4d metal surfaces. *The Journal of Chemical Physics*, 130(18):184707, 2009.

[82] KSA Novoselov, Andre K Geim, SVb Morozov, Da Jiang, MIc Katsnelson, IVa Grigorieva, SVb Dubonos, and AAb Firsov. Two-dimensional gas of massless dirac fermions in graphene. *Nature*, 438(7065):197–200, 2005.

[83] A Ambrosetti and PL Silvestrelli. Adsorption of rare-gas atoms and water on graphite and graphene by van der waals-corrected density functional theory. *The Journal of Physical Chemistry C*, 115(9):3695–3702, 2011.

[84] Florian Banhart, Jani Kotakoski, and Arkady V Krasheninnikov. Structural defects in graphene. *ACS nano*, 5(1):26–41, 2010.

[85] LS Panchakarla, KS Subrahmanyam, SK Saha, Achutharao Govindaraj, HR Krishnamurthy, UV Waghmare, and CNR Rao. Synthesis, structure and properties of boron and nitrogen doped graphene. *arXiv preprint arXiv:0902.3077*, 2009.

[86] Yao Zheng, Yan Jiao, Lei Ge, Mietek Jaroniec, and Shi Zhang Qiao. Two-step boron and nitrogen doping in graphene for enhanced synergistic catalysis. *Angewandte Chemie*, 125(11):3192–3198, 2013.

[87] Arava Leela Mohana Reddy, Anchal Srivastava, Sanketh R Gowda, Hemtej Gullapalli, Madan Dubey, and Pulickel M Ajayan. Synthesis of nitrogen-doped graphene films for lithium battery application. *ACS nano*, 4(11):6337–6342, 2010.

[88] LS Panchakarla, A Govindaraj, and CNR Rao. Boron-and nitrogen-doped carbon nanotubes and graphene. *Inorganica Chimica Acta*, 363(15):4163–4174, 2010.

[89] Xin Lu, Zhongfang Chen, and Paul v R Schleyer. Are stone- wales defect sites always more reactive than perfect sites in the sidewalls of single-wall carbon nanotubes? *Journal of the American Chemical Society*, 127(1):20–21, 2005.

[90] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98 (14):146401, 2007.

[91] Mike P Allen and Dominic J Tildesley. *Computer simulation of liquids*. Oxford university press, 1989.

[92] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.