PC-II Praktikum (HS 2020)

Density functional theory (DFT) calculations of O_2 adsorption and STM images on Pt (111)

Assistants

Xing Wang & Moloud Kaviani
Room: N216
Phone: 031 631 56 25
Email: xing.wang@dcb.unibe.ch, moloud.kaviani@dcb.unibe.ch

The practical will be carried out in room **N212** starting from **9:00**

1. Introduction

In this practical we will compute the adsorption of oxygen on the popular catalytic surface of Pt (111), use this information to interpret an experimental TPD spectrum and also compute STM images of the adsorbed species. In order to understand what one sees in STM, we will make a brief digression into computing STM images for graphene and graphite.

This practical assumes you have never used a Linux computer or a DFT code and we will introduce both these tools after a general introduction to the topics of the practical.

1.1 Surface adsorption

Molecules may adsorb on a catalyst surface either molecularly (i.e. without breaking any intramolecular bonds, also called physisorption) or dissociatively (after breaking some intramolecular bonds and creating new ones with the surface; one of the cases of chemisorption). For each of these adsorption modes, there are multiple sites that depends on the surface structure in question. For the case of a {111} surface of a face-centered cubic metal (such as Ni, Pd and Pt) the "top", "bridging" as well as "hollow" fcc (has no atom in the layer below it) and hcp (has an atom in the layer below) sites are illustrated in figure 1.

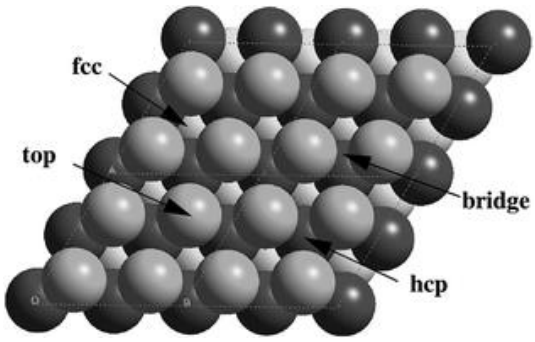


Figure 1: Top view of the adsorption sites on a {111} face-centered cubic surface

Characterizing which molecular or atomic species are adsorbed in what configuration on a catalyst surface is crucial for understanding which reactive pathways may occur. One way to experimentally discriminate between different species and sites is by using the fact that different adsorption configurations have different adsorption energies and desorption is therefore activated at different temperatures. "Temperature programmed desorption" (TPD) monitors the desorption as a function of temperature at a given heating rate. The desorbed species are detected using a mass spectrometer (see figure 2) and monitoring temperature dependent signals of the different m/Z channels allows identifying the desorbed species and also enables them to be isotopically labeled.

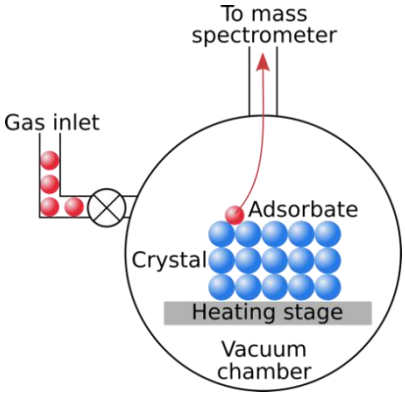


Figure 2: Schematic view of a TPD setup. Adsorbate molecules are dosed on the surface at a given temperature. Then the inlet valve is shut, and the temperature raised at a fixed rate. The desorbing adsorbate molecules are detected using a mass spectrometer.

The rate of desorption of a species from a surface is given by the Arrhenius equation:

$$R_{des} = -\frac{dN}{dt} = \nu N^x \exp\left(\frac{-E_a}{kT}\right), \tag{1}$$

where ν is the attempt frequency, N is the concentration of species at the surface, x is the kinetic order of desorption (1 for simple desorption, 2 for recombinative desorption), E_a the activation energy for the desorption process and k and T Boltzmann's constant and the temperature respectively. The rate of desorption is given as a function of time. We are however interested in the change as a function of temperature. In typical TPD experiments one uses a linear heating rate

$$\frac{dT}{dt} = \beta. \tag{2}$$

We use this to convert the rate of desorption in equation (1) into a temperature dependent rate:

$$-\frac{dN}{dT} = -\frac{dN}{dt}\frac{dt}{dT} = -\frac{dN}{dt}\frac{1}{\beta} = \frac{\nu N^x}{\beta}\exp\left(\frac{-E_a}{kT}\right) \propto I(T)\ . \tag{3}$$

The intensity $I(T)$ of the measured TPD signal is proportional to the number of species leaving the surface (dN) and is hence proportional to this equation. To find the peak temperature of the TPD signal we differentiate this equation with respect to T

$$\frac{dI}{dT} = \frac{d}{dT}\left[\frac{\nu N^x}{\beta}\exp\left(\frac{-E_a}{kT}\right)\right] = \frac{\nu}{\beta}xN^{x-1}\frac{dN}{dT}\exp\left(\frac{-E_a}{kT}\right) + \frac{\nu N^x}{\beta}\frac{E_a}{kT^2}\exp\left(\frac{-E_a}{kT}\right) \quad (4)$$

where we have a sum of two terms since N also depends on T. We know $\frac{dN}{dT}$ from equation 3 and hence get:

$$\frac{dI}{dT} = \frac{\nu N^x}{\beta}\left[\frac{E_a}{kT^2} - \frac{x\nu N^{x-1}}{\beta}\exp\left(\frac{-E_a}{kT}\right)\right]\exp\left(\frac{-E_a}{kT}\right) = 0, \quad (5)$$

where we have set the derivative equal to zero to find the maximum. The terms outside the square brackets cannot be zero, which means the term in square brackets has to equal zero:

$$\frac{E_a}{kT^2} - \frac{x\nu N^{x-1}}{\beta}\exp\left(\frac{-E_a}{kT}\right) = 0 \quad (6)$$

This equation cannot easily be solved for either E_a or T and you will have to use numerical tools to get T for a given E_a.

TPD therefore gives accurate information about the desorption temperature and hence the adsorption energy of a species but it does not directly allow one to identify the adsorption site. This is where computational approaches can be useful as they yield the adsorption energy of a species at different adsorption sites, which can then be used to associate the TPD peaks with different adsorption sites.

In this practical we will use density functional theory (DFT) calculations to identify the mode and site of adsorption of oxygen on the (111) surface of Platinum. The relevant experimental data is taken from *Jerdev et al., Surface Science, 498, L91-L96, (2002)*. Figure 3 shows the TPD spectrum we will try to explain with our calculations.

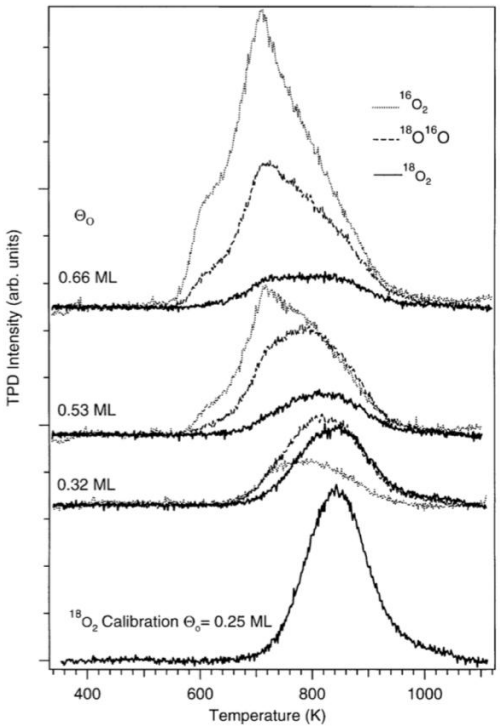


Figure 3: The TPD spectrum we will try to explain. Figure taken from *Jerdev et al., Surface Science, 498, L91-L96, (2002).*

What was done in this experiment is that the surface was covered with 0.25 ML (the ratio of adsorbate to surface atoms is ¼) $^{18}O_2$, which desorbed at a temperature of ~840K. Then additional $^{16}O_2$ was dosed on the surface and additional peaks associated with desorption of $^{16}O_2$ and $^{16}O^{18}O$ appeared at lower temperatures of ~700K and ~780K respectively. This indicates that the $^{16}O_2$ adsorbed weaker than the pre-dosed $^{18}O_2$, but how these species adsorb exactly is not extractable from this data alone but can be done with the help of density functional theory calculations that we will carry out in this practical.

1.2 STM images

As you have already seen (or will still see, if you are unlucky enough to do the theory before the experiment) one can experimentally visualize the atomic and electronic structure of surfaces by means of a scanning tunneling microscope (STM). In STM one scans an atomically sharp tip over a surface without touching the surface. From classical mechanics no current should flow, as the sample and tip do not touch. Quantum mechanics however shows that electrons may "tunnel" though the vacuum. This is due to the electronic wavefunction extending a small distance into classically forbidden regions of space, the vacuum region being such a forbidden region. The probability for an electron to cross the vacuum gap decreases exponentially with increasing sample-tip separation (Δz), which means that effectively only the atom(s) at the very tip of the tip (if this expression makes sense), which has the smallest Δz, will participate in tunneling. Tunneling thus enables electrons to cross the vacuum but in order to get a net current, a bias potential has to be applied between the sample and the tip. One can imagine that, depending on the sign of the potential, this bias lowers or increases the energy of the sample with respect to the tip, as shown in Fig. 1 for the case where the

sample energy is increased. The effect is that electrons from the sample can lower their energy by tunneling into the tip, leading to a net current flowing from the sample to the tip. A bias of opposite sign will lead to current flowing the other direction.

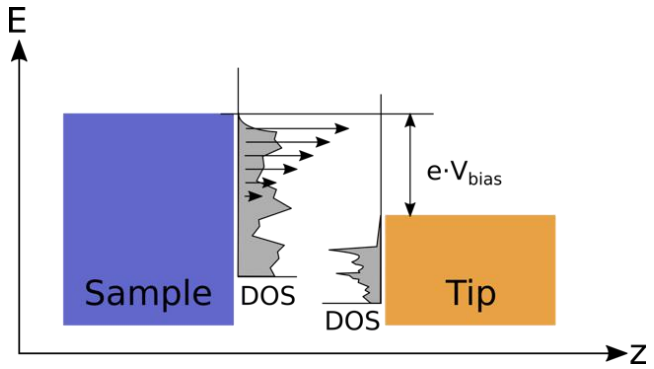


Fig. 4: Schematic view of the effect of the bias potential on the energy levels in a STM. In this example the bias raises the energy of the sample relative to the tip, which allows electrons in the sample in states between E_{fermi}-$e \cdot V_{bias}$ and E_{fermi} to tunnel to the tip (E_{fermi} is the top of the box in both cases). States which are closer to E_{fermi} have a higher probability to tunnel as shown by the different length arrows. Also, of importance for STM are the densities of states (DOS) of both the sample and the tip as they dictate at what energies electrons can exist.

The bias potential allows electrons with energies between the Fermi energy E_{fermi} and $E_{fermi}+V_{bias} \cdot e$ to tunnel through the vacuum. The number of electrons at the various energies in this range is given by the electronic density of states (DOS) also written as ρ, which is an intrinsic property of the materials forming the sample and tip respectively.

STM can be operated in different modes. In the constant-height mode, one scans the tip over the sample at a given height and monitors the tunneling current. Experimentally this has the drawback of potentially being able to "crash" the tip on surface protrusions. A safer alternative is to operate the microscope in constant-current mode, where the height of the tip is adjusted so as to keep the tunneling current constant and to record the height of the tip at each point. This z=f(x, y) dataset contains the topography coming from both the atoms and the electrons at the surface.

In this practical we want to compute STM images via quantum mechanical density functional theory (DFT) calculations. Simulating the whole system of a surface and a tip with a bias potential would involve many atoms and quickly go beyond the possibilities of DFT calculations. Fortunately, Tersoff and Hamann developed a model that does not rely on having to model the tip since they assume the tip's electronic structure to be the one of a spherically symmetric s-wavefunction, which leads to the following expression for the tunneling current:

$$I(\vec{r}_{tip}, V_{bias}) \propto \int_0^{e \cdot V_{bias}} \rho(\vec{r}_{tip}, E_{fermi} + \varepsilon) d\varepsilon \qquad (7)$$

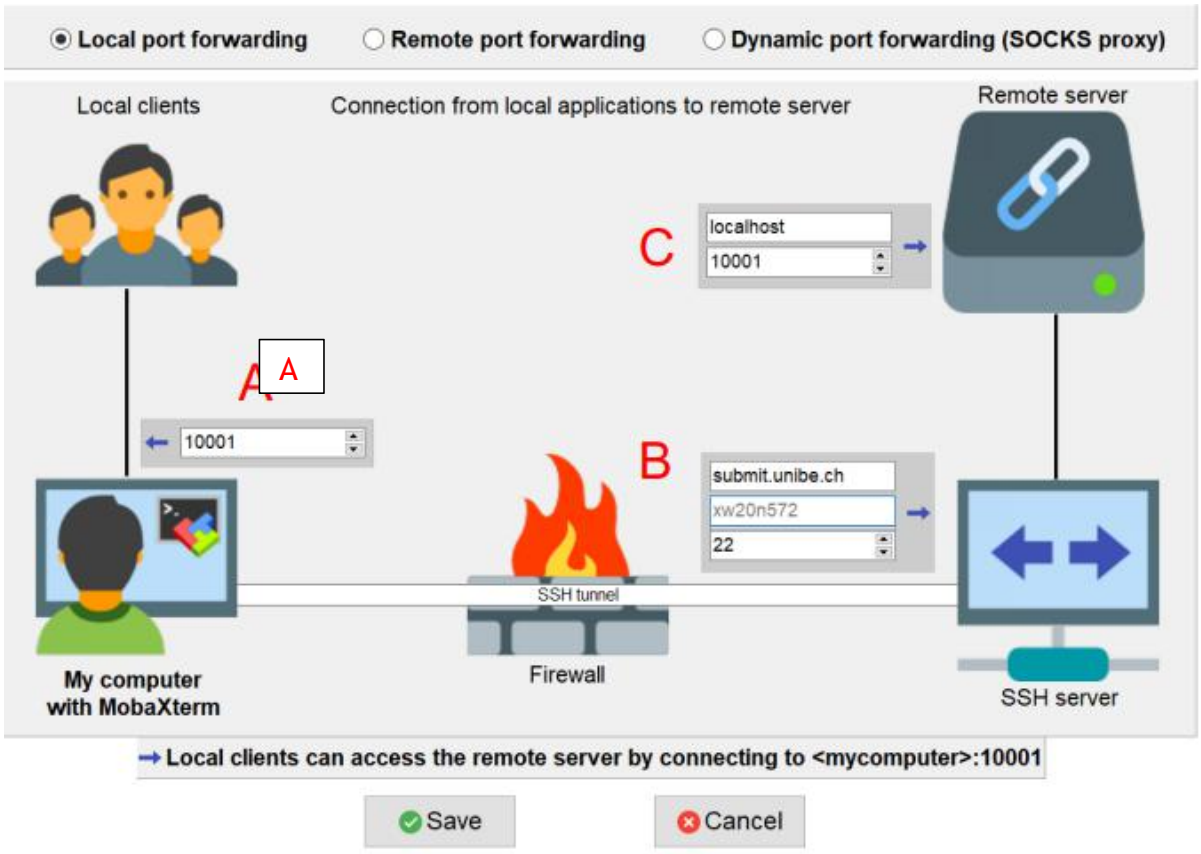
The tunneling current is thus proportional to the integral between E_{fermi} and $E_{fermi}+e \cdot V_{bias}$ of the local density of states of the sample, local meaning at the point

r_{tip} under the tip. We will use this formulation for this practical and compute the local density of states from DFT.

2. Create an SSH Tunnel to the High Performance Computing Center

This practical is partly formulated as Jupyter Notebooks. In a Jupyter Notebook you are running the calculations on the High Performance Computing Center (Ubelix) of the University of Bern, but the output is displayed in your local browser. This requires some setup, which is described below.

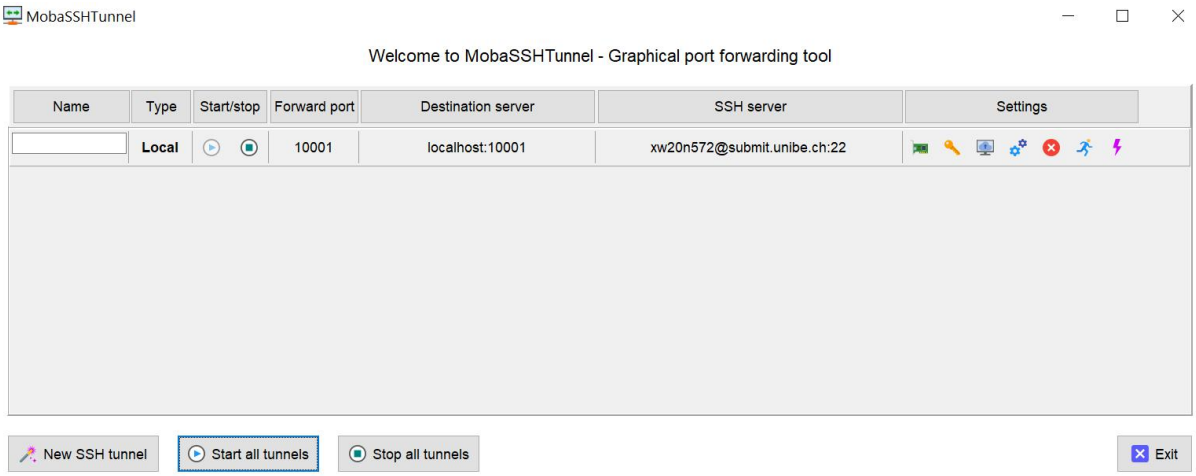
First open the MobaXterm program. In the top of your MobaXterm login window there is a row of buttons. One of them is named Tunneling, press that button. You now get a new window called MobaSSHTunnel, in the lower left corner of the new window you find a button called New SSH Tunnel, press it. A new window opens, as shown here:



In the field marked with a red A on the figure you should write the port number xxx. This port number are numbers between 2000 and 65000. Please do not use the example port 10001. Rember the port number you select, and used it in the future.

In the field marked B you should write the name of the remote computer, in this case use submit.unibe.ch. You also need to enter user name (Your Campus Account, for example xw20n572 in the figure).

In the field marked C you write the name of the remote server (in this case use localhost), and the port number xxx of the Notebook server. Now press the button Save. You will now see a window like the one shown here:



Check that the machine name and port number are correct, then start the tunnel by pressing the small “play” button (with a right-pointing triangle). If you are asked for a password, it is the Ubelix password.

3. Launch the JupyterLab server on UBELIX

In the middle of your MobaXterm login window, click the “Start local terminal” to create a new “Local session”.

You access UBELIX via ssh (secure shell) by typing the following command in the local session:

```
ssh -Y <username>@submit.unibe.ch
```

where you substitute <username> with your Campus Account.

Type the following commands:

```
module load Anaconda3
```

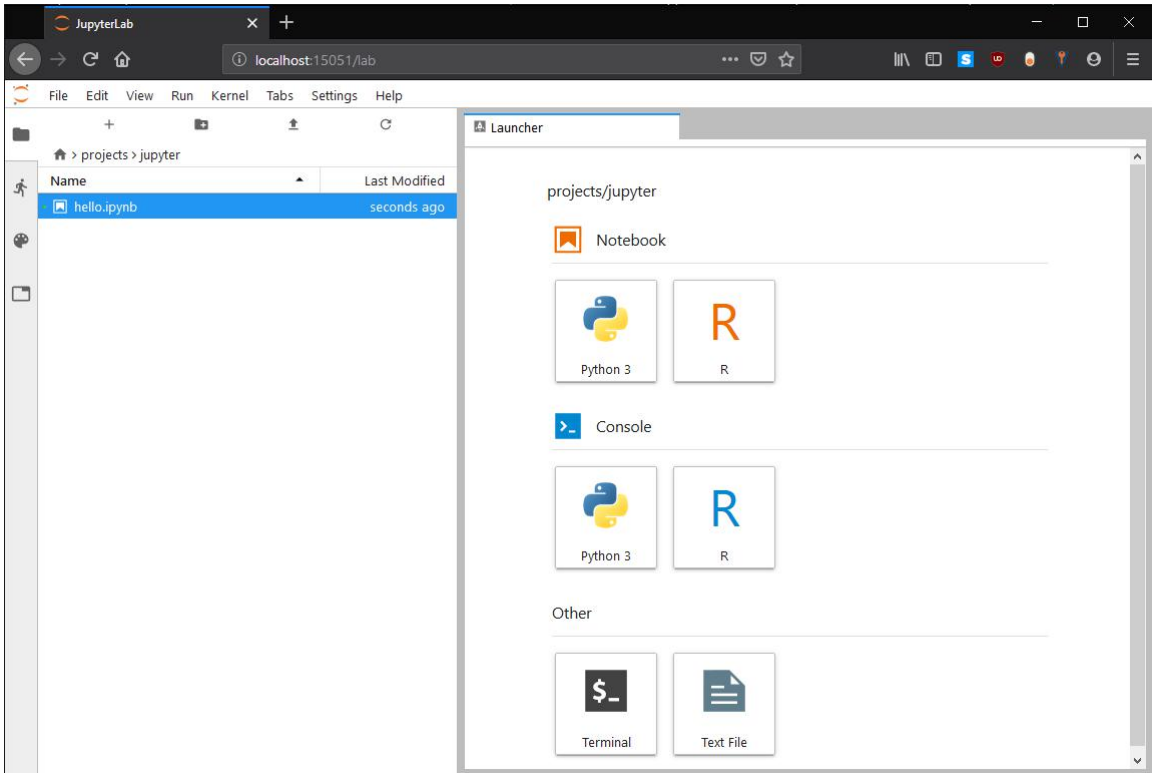
```
jupyter-compute your-port-number --ntasks 1 --cpus-per-task 4 --mem 8G --time=08:00:00 --account=dcb
```

This command will lauch the server on a compute node, and establish the port forwarding. After general output, JupyterLab prints a URL with a unique key and the network port number where the web-server is listening, this should look similar to:

```
Jupyter lab will be started on knode22 and establish port farwarding from submit02.ubelix.unibe.ch on
 port 10001
[I 09:13:41.139 LabApp] JupyterLab extension loaded from /software.el7/software/Anaconda3/2020.02-fos
s-2020a/lib/python3.7/site-packages/jupyterlab
[I 09:13:41.140 LabApp] JupyterLab application directory is /software.el7/software/Anaconda3/2020.02-
foss-2020a/share/jupyter/lab
[I 09:13:41.428 LabApp] Serving notebooks from local directory: /gpfs/homefs/dcb/xw20n572
[I 09:13:41.428 LabApp] The Jupyter Notebook is running at:
[I 09:13:41.428 LabApp] http://knode22:10001/?token=7de2dfd0ca8fe0c39e68933182645f6cf67a5b66d4034514
[I 09:13:41.428 LabApp]  or http://127.0.0.1:10001/?token=7de2dfd0ca8fe0c39e68933182645f6cf67a5b66d40
34514
[I 09:13:41.428 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip co
nfirmation).
[C 09:13:41.445 LabApp]

    To access the notebook, open this file in a browser:
        file:///gpfs/homefs/dcb/xw20n572/.local/share/jupyter/runtime/nbserver-74376-open.html
    Or copy and paste one of these URLs:
        http://knode22:10001/?token=7de2dfd0ca8fe0c39e68933182645f6cf67a5b66d4034514
     or http://127.0.0.1:10001/?token=7de2dfd0ca8fe0c39e68933182645f6cf67a5b66d4034514
```

The full address on the last line (starting with the 127.0.0.1) including the token needs to be copied into your browser (Chrome and Firefox) on your local machine. After initializing Jupyter Lab you should see a page similar to:



Click Terminal on the browser to create a terminal. Type the following command:

```
cp /home/ubelix/dcb/aschauer/practical.tar.gz .
```

Where “cp” stands for copy. This copies the compressed archive (.tar.gz) containing the files needed for this practical to your home directory. Decompress it with the following command:

```
tar -xzf practical.tar.gz
```

This will generate a directory with the required files. You can see a “practical” folder in the right side of your browser.

Open the “practical” folder in your browser, you are now ready to open one of the notebooks (file ended with “.ipynb”), and run the exercises.

4. Density functional theory – the absolute basics

4.1 Basic formulation

In this practical we will use density functional theory (DFT), which is a popular quantum mechanical method to compute energies, electronic properties and structures of molecular and crystalline systems. The many-particle wave function ψ of the Schrödinger equation with Hamiltonian operator $\widehat{H}$ and total energy *E* that we seek to solve

$$\widehat{H}\psi(\vec{r}_1, \vec{r}_2, ..., \vec{r}_n) = E\psi(\vec{r}_1, \vec{r}_2, ..., \vec{r}_n) \tag{8}$$

is a function of the 3D vectors of the positions $\vec{r}_i$ of all *n* electrons and hence depends on 3*n* variables. The Hamiltonian is defined as the sum of the electron kinetic ($\widehat{T}$), electron-core ($\widehat{V}$) and electron-electron ($\widehat{U}$) interactions:

$$\widehat{H} = \widehat{T} + \widehat{V} + \widehat{U} = -\frac{\hbar^2}{2m_e}\sum_i \nabla_i^2 + \sum_i V(\vec{r}_i) + \sum_{i\neq j}\frac{e^2}{|\vec{r}_i - \vec{r}_j|} \tag{9}$$

For a large number of electrons, this quickly becomes impractical to solve, especially due to the computationally expensive electron-electron interaction term. The popularity of density functional theory stems from the fact that it reformulates the many-body problem into a combination of *n* independent single-particle equations with single-particle wave functions Φ_i

$$\left[-\frac{\hbar^2}{2m_e}\nabla_i^2 + V(\vec{r}_i) + \int\frac{e^2 n(r')}{|\vec{r}_i - \vec{r}'|}d\vec{r}' + V_{XC}[n(\vec{r}_i)]\right]\Phi_i(\vec{r}) = \varepsilon_i\Phi_i(\vec{r}_i), \tag{10}$$

in which the previously expensive terms of the Hamiltonian depend not on all *n* 3D vectors of the electron positions, but on the electron density, which is a function of only the 3 spatial coordinates and is defined as

$$n(\vec{r}) = \sum_i |\Phi_i(\vec{r}_i)|^2. \tag{11}$$

The full quantum-mechanical electron-electron interaction contained in (8) is described by the sum of the classical coulomb interaction of an electron with the electron density and a functional of the electron density that captures all the many-body interactions and is known as the exchange-correlation potential V_{XC}.

4.2 Exchange-correlation functionals

Despite the fact that we do not formally show it here, density functional is up to this point an exact theory. A practical issue is however that the exact exchange-correlation functional is not known and one has to make approximations for it. The simplest approximation is the so-called "local density approximation" (LDA), in which the exchange-correlation potential is taken as the one in a homogeneous electron gas of the same density. One step up in complexity are the "gradient corrected approximation" (GGA) functionals in which not only the local density but also its gradient determines the exchange correlation energy. Such functionals

typically do much better for molecular systems and we chose one of the most popular, the Perdew Burke Ernzerhof (PBE) functional for this practical. Over the years more advanced functionals have been developed, which better capture the many-body effects, however also at a significantly increased computational cost.

4.3 Self consistency

A further practical issue is that in equation (9) the classical electrostatic interaction and the exchange-correlation potential depend on the electron density, which according to equation (10) is determined by the wave-functions. We thus need the density to get the wave-function but also need the wave-function to get the density, which requires solving the problem self-consistently starting from a guess of the electron density and iterating until the density determines the potential well enough for our purposes.

5. Practical aspects of planewave-pseudopotential DFT

5.1 Reciprocal space integration (k-points)

The wavefunction in a periodic crystalline solid has the form of a Bloch wave

$$\Phi_{i,\vec{k}}(\vec{r}) = e^{i\vec{k}\vec{r}} u_{i,\vec{k}}(\vec{r}) \tag{12}$$

Where $u_k(\vec{r})$ is periodic within the unit cell and it is modulated by a planewave $e^{i\vec{k}\vec{r}}$. For macroscopic crystals, $\vec{k}$ is a continuous variable and properties have to be obtained by integrating over all values of $\vec{k}$. In practice, we typically sum over a discrete and finite set of so-called k-points, dense enough to yield converged properties.

Finding the required number of k-points is one of the tasks you will have to perform.

5.2 Plane wave basis set

The periodic part $u_k(\vec{r})$ is in practice represented by expansion in a given set of basis functions. In a solid, plane-waves are a natural choice as by construction they have the periodicity of the lattice. The expansion in this basis becomes

$$u_{i,\vec{k}}(\vec{r}) = \frac{1}{V_{cell}} \sum_{m}^{\left(\vec{k}+\vec{G}_m\right)^2 < 2E_{cutoff}} c_{i,m}(\vec{k}) e^{i\vec{G}_m\vec{r}} \tag{13}$$

Where we include plane waves ($e^{i\vec{G}_m\vec{r}}$) with kinetic energy determined by $\vec{k}$ and $\vec{G}_m$up to a cutoff energy E_{cutoff} in the expansion. As we increase this cutoff energy, the basis becomes more and more complete and in practice we choose this value high enough to obtain well-converged properties.

Finding the required cutoff energy is one of the tasks you will have to perform.

5.3 Pseudopotentials

Core electrons have a negligible influence on the chemistry but their localized character requires high plane-wave cutoffs in the expansion of the wavefunction. In order to solve this problem and simultaneously cut down the number of electrons to be computed, we use so-called pseudopotentials to describe the potential that an electron feels due to the combined presence of the core and the core electrons. This potential is precomputed and we either choose or generate one that is compatible with our exchange-correlation functional and reproduces the properties we would like to study.

This step has been done for you and we will provide you with suitable pseudopotentials.