

Excercise sheet 2

The topic of this week is the optimization of the atomic positions and lattice vectors in a number of solids.

This can be done rather efficiently in Quantum Espresso by making a few modifications to the input files compared to that of pure calculations of the electronic groundstate. The following picture shows a typical input file for a geometry optimization run:

```

&control
  calculation = 'vc-relax'
  prefix='Cu',
  tprnfor = .true.
  pseudo_dir = './',
  outdir='TMP_DIR/'
  forc_conv_thr = 0.0004
/
&system
 ibrav=0, nat= 4, ntyp= 1,
ecutwfc =90.0, input_dft = 'lda'
/
&electrons
  conv_thr = 1.0d-8
/
&ions
  ion_dynamics = 'bfgs'
/
&cell
  cell_dynamics = 'bfgs',
  press = 0.0,
  press_conv_thr = 0.1,
  cell_factor = 1.1
/
ATOMIC_SPECIES
Cu 63.55 Cu.nc.UPF
ATOMIC_POSITIONS (crystal)
Cu 0.0000000000000000 0.0000000000000000 0.0000000000000000
Cu 0.5000000000000000 0.5000000000000000 0.0000000000000000
Cu 0.0000000000000000 0.5000000000000000 0.5000000000000000
Cu 0.5000000000000000 0.0000000000000000 0.5000000000000000
K_POINTS {automatic}
10 10 10 1 1 1
CELL_PARAMETERS (angstrom)
3.6550000000000000 0.0000000000000000 0.0000000000000000
0.0000000000000000 3.6550000000000000 0.0000000000000000
0.0000000000000000 0.0000000000000000 3.6550000000000000
  
```

Annotations:

- calculation = 'vc-relax'**: ,relax': optimize the atomic positions and leave the lattice vectors (and the volume) fixed
- forc_conv_thr = 0.0004**: ,vc-relax': optimize both atomic positions and the lattice vectors
- forc_conv_thr = 0.0004**: Tolerance for the interatomic forces (in Ry/a₀) for the geometry optimization (optional, default value is 0.001 Ry/a₀)
- &ions** and **ion_dynamics = 'bfgs'**: Block defining the relaxation of the *atomic positions* (non-optional for relax and vc-relax)
- &cell** and its parameters: Block defining the relaxation of the *lattice vectors* (non-optional for vc-relax)
- Starting geometry**: Points to the ATOMIC_POSITIONS and CELL_PARAMETERS blocks.

The input file has all the features of the ones from last week, i.e. parameters that determine the accuracy of the SCF calculations and the atomic positions and lattice vectors of an initial geometry that is supposed to be optimized.

The geometry optimization run is invoked by changing the value of calculation in the &control block:

- **calculation='relax'** tells Quantum Espresso to optimize the atomic positions by moving the atoms around, while keeping the lattice constants fixed. Such a run is useful for the calculation of isolated molecules, which we would do in Quantum Espresso by putting them in a large box of vacuum, i.e. we don't want the lattice constants to change. For solids, it is useful in cases, where one, for whatever reason, wants to use pre-determined lattice constants, for example experimental lattice constants. It is also useful for 'pre-relaxation' of initial geometries that are distorted or 'bad' in other ways and might give trouble when atomic positions and lattice constants are optimized simultaneously.

- `calculation='vc-relax'` tells Quantum Espresso to do a similar run as `'relax'`, but use the calculated stress tensor of the system to optimize the lattice vectors together with the atomic positions. This is typically the standard run for the geometry optimization in solid state systems.

As outlined in the lecture, the geometry optimization run will adjust the atomic positions and perform a subsequent SCF run to minimize the electronic energy for each geometry until a convergence criterion is reached. In Quantum Espresso, this is the case if each component of the interatomic forces is smaller than the force threshold defined by the keyword `force_conv_thr` in the `&control` block. The force threshold is given in units of Ry/a0, which is the unit of the forces given in the Quantum Espresso output. The smaller this threshold is chosen, the better the resulting geometry will be; however, the optimization might converge very slowly (or not at all), if a too small value of `force_conv_thr` is used.

In addition to the changes in the `&control` block, additional blocks have to be defined for geometry optimization runs.

The `&ions` block, in green frame in the figure, controls the movement of the nuclei during the optimization run. The most relevant keyword is `ion_dynamics`, which defines the method that Quantum Espresso should use for the geometry optimization. For `'relax'` or `'vc-relax'` runs, this can (and should be) be left at the default value, which tells Quantum Espresso to use the BFGS method for the optimization. While `ion_dynamics` does not have to be defined, the `&ions` block itself has to be present in the input file (even if it is empty) for geometry optimization runs, otherwise QE will complain.

In a similar way, the `&cell` block (in purple frame) is mandatory for `'vc-relax'` runs, but not for `'relax'` runs. It controls the way the lattice vectors are changed during the geometry optimization. The to-be-used optimization method is defined through the keyword `cell_dynamics`, and has to be `'bfgs'`, if `ion_dynamics='bfgs'`. The keyword `press_conv_thr` defines the convergence threshold for the stress acting on the calculated unit cell, similar as `force_conv_thr` defines the threshold for the interatomic forces. To be honest, I am not entirely sure, what exact unit `press_conv_thr` is given in and what exactly the related convergence criterion is. Usually, it appears to define the largest value that a component of the stress tensor can have for the geometry optimization to be seen as converged, but this is not always the case. Typically, a value of 0.01-0.1 appears to yield good geometries. The keyword `press` can be used to apply an external hydrostatic pressure (in kbar) on the system. This allows to simulate systems under pressure, `press_conv_thr` will then define the convergence of the stress tensor towards `press`.

The input parameters and general format of the input file is described at https://www.quantum-espresso.org/Doc/INPUT_PW.html.

Exercise 1: Geometry optimization of MgO

In this exercise, we will make use of the convergence tests from Exercise Sheet 1 and will see, what lattice constants various exchange-correlation approximations yield.

- (a) In the first step, take the input file *MgO_a.in* from the .zip for this exercise sheet and replace the cutoff energy and the k-point sampling with the converged values from the convergence

tests from last week. You can, of course, choose whether you want to use normconserving or ultrasoft pseudopotentials (in the latter case, remember to define `ecutrho` in the input file). The input file also uses the conventional cell of the fcc lattice, which will make it slightly easier to derive the cubic lattice constants from the output file, but also makes the calculations slower. Again, you can opt to use the primitive cell instead.

Unfortunately, I made a stupid mistake with the atomic positions in last week's input files: I accidentally swapped several Mg and O atoms, so that the atomic positions in the input files did not form the crystal shown in the figure. A good reminder to always check the sanity of the input files, because the calculations might run smoothly even if the input geometry is nonsense. In this sense, please make sure that you use the correct atomic positions for the following calculations. Still, use the converged `ecutwfc` and k-point sampling from last week (unless you want to redo the convergence test).

We will now perform a geometry optimization, where we will let Quantum Espresso try to find the atomic geometry with the lowest total energy by varying both the atomic positions and the lattice vectors. This run type is invoked by `calculation_type='vc-relax'` in the input file. The force tolerance is set to a value of `force_conv_thr=0.0004`, which should yield sufficiently optimized atomic positions. Further, as outlined above, the input file contains two additional blocks, `&ions` and `&cell`, which control the geometry relaxation procedure. We will use the BFGS algorithm for the iterative variation of the atomic structure. We do not use an external pressure (i.e. `press=0.0`) and want the cell stress/pressure to be converged up to a tight value of `cell_conv_thr=0.01`. We will first do the calculations using the local density approximation (LDA) as exchange-correlation functional.

Use the prepared input file to run the geometry optimization calculation and examine the output file. Quantum Espresso first performs an SCF run in order to find the groundstate density and the interatomic forces and stresses that it will use to continue the calculation. First, have a look at the decomposition of the total energy into its contributions that is given after the list of the single-particle energies. The “one-electron contribution” is equal to the non-interacting kinetic energy of the electrons. The “ewald contribution” is the electrostatic energy from Coulomb interaction between the nuclei (approximated as point charges)¹.

The code then lists the interatomic forces, as well as the stress tensor in units of Ry/au^3 and kbar. Notice that all interatomic forces are exactly zero. The reason for this is that Quantum Espresso lists the *symmetrized* forces, which vanish due to the cubic symmetry of MgO. The stress, on the other hand, is much larger than our convergence criterion. Correspondingly, Quantum Espresso uses the stress (and the forces) to update the lattice vectors and atomic positions. The Hessian matrix that is used for the update is constructed to satisfy the symmetry

¹ It is negative, because the nuclei are modeled implicitly to be surrounded by a negative homogeneous charge density that keeps the whole system charge neutral. Otherwise, the interaction energy between the nuclei would diverge. A similar treatment is being used for calculation of the Hartree energy, where the Coulomb energy of the electron charge density is calculated implicitly for the presence of a homogeneous background charge (which corresponds to the average charge density of the nuclei). This is equivalent to neglecting the $G=0$ term in the Hartree energy, otherwise, the Hartree energy would diverge. The divergences of the Hartree and the Ewald energy cancel exactly.

operations that the initial geometry exhibits. Correspondingly, the updated atomic positions and lattice will have the same symmetry. On one hand, this is usually beneficial. On the other hand, it can lead to problems, because one imposes a certain symmetry on the final structure in this way, which is not always desirable.

The code then performs a new SCF run for the updated geometry, calculates the forces and stresses and repeats the whole process, until the convergence criteria are satisfied. Notice that Quantum Espresso then performs another, final, SCF run. The reason for this is that the planewave basis set was kept fixed during the geometry optimization run (i.e. fixed N_{PW}). This might have affected the prediction of the interatomic forces and the stresses due to the change in the unit cell volume (and potentially the shape) and led to errors. For this reason, Quantum Espresso generated the correct basis planewave basis set for the relaxed unit cell and `ecutwfc` for the final SCF run. Indeed, you can see at the end of the file that the stress for the correct planewave basis set is calculated to be larger than the convergence threshold.

Replace the initial atomic positions and lattice vectors from the input file by the optimized geometry and rerun the calculation. Continue this process until the stress calculated by the final SCF runs satisfies the convergence criterion or the lattice constants changed by less than 0.001 Angstrom through the rerun.

You can convert the unit cell of the crystal for each geometry step into something that can be visualized by using the *atomsk* package (I will put a version that can be used on the HPC to the .zip file, if you use your home computer, download it at <https://atomsk.univ-lille.fr/>) and running the command

```
PATH_WHERE_ATOMSK_IS_LOCATED/atomsk --one-in-all run.out xsf -remove-property all -u bohr angstrom
```

in the folder where the output of your Quantum Espresso calculations, `run.out` (or whatever you called it) is located. This will write one .xsf file per geometry tried in the geometry optimization run. You can visualize these .xsf files using XCrysDen (on Linux) or VESTA (on Windows and possibly Linux)

If you are interested, you can also compare the validity of the results from the BFGS minimization by calculating a series of total energies where the lattice constants are adjusted manually, and fitting the Birch-Murnaghan equation to the resulting total energy data.

- (b) The interatomic forces usually converge slower with change of the electron density than the total energy, which typically requires tighter convergence thresholds for the SCF runs, `conv_thr`, than one would need if one is simply interested in the total energy. If you look through the output files from one of the previous steps, you will notice that Quantum Espresso actually adjusts the effective convergence threshold below that of `conv_thr` during the geometry optimization run. For example, particularly in the later optimization steps, the SCF runs converge the total energy to an accuracy of 10^{-8} Ry. The maximum reduction of the effective convergence threshold compared to `conv_thr` is decided by a parameter and usually is a factor 1/1000.

Do another optimization run for a reduced `conv_thr=10-10` and see whether this leads to significant changes in the lattice vectors (i.e. changes by more than 0.001 Angstrom) compared to `conv_thr=10-6`.

- (c) As mentioned in the lecture, the interatomic forces usually converge slower with the cutoff energy than the total energy does. For this reason, we also should test the influence of the cutoff energy on the optimized geometry.

To do this, take the optimized atomic positions and lattice vectors from (b) and perform another optimization run for a higher cutoff energy, for example `ecutwfc=100`. Do the lattice constants change by more than 0.001 Angstrom? If so, repeat the calculations for an even higher cutoff.

- (d) Now, optimize the atomic positions and lattice vectors using the PBE and PBEsol XC approximation (on the HPC, I propose running both calculations simultaneously in different folders). For these calculations, you can take the same `ecutwfc` and k-point sampling as for the LDA calculations, as they should yield converged results for all exchange-correlation functionals.

Compare the results with the experimental value of the cubic lattice constant of MgO of $a=4.21$ Angstrom. Which XC functional yields the best results? Was this to be expected from the information given in the lecture?

- (e) The input file *MgO_e.in* contains the geometry of simple cubic MgO with a two-atom basis. First, optimize the geometry of this structure in order to guarantee meaningful predictions of the physical properties of this material. Then, derive the total energy per atom for this structure and compare it with the total energy per atom for the FCC structure that was obtained from the PBEsol calculations in (d). make sure to use the same cutoff energy as in (d).

Which structure of MgO is the most stable, i.e. has the lowest total energy?

Exercise 2: van-der-Waals interaction in hexagonal boron nitride

In this exercise, we will have look at the geometry of hexagonal boron nitride. Similar to graphite, the unit cell of hexagonal boron nitride consists of two flat layers that are bonded together by van-der-Waals interactions. The structure and experimental data for the lattice constants is shown in the figure.

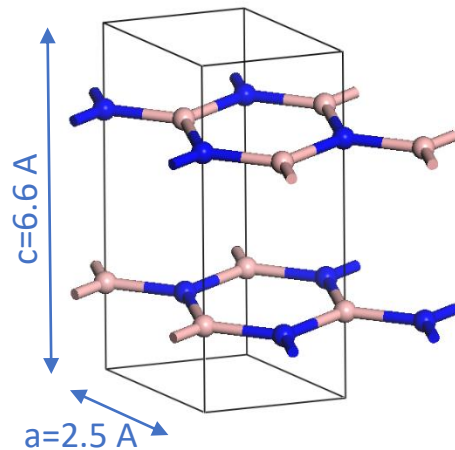
- (a) Use the provided input file *BN_a1.in* and the normconserving pseudopotentials from the .zip file and perform a convergence test for `ecutwfc`. Find an `ecutwfc` that is likely to give good results for the lattice constants and atomic geometries. Keep in mind that the non-covalent forces are quite weak compared to the covalent forces, the prediction of the interatomic forces from DFT hence needs to be sufficiently converged to properly capture them.

A k-point sampling of 12x12x4 points should yield good results for the geometry optimization. If you are doing calculations from home, even a 9x9x3 sampling should suffice.

If you are interested, you can, of course, do an additional convergence test on the k-point sampling as well. Here, it is advisable to keep the k-point density along each direction in reciprocal space roughly the same and first converge the sampling of the two directions that

are parallel to the layers (i.e. the first two numbers) and then the sampling perpendicular to the layers (the third number)

- (b) Now use the parameters obtained from the convergence test to perform a geometry relaxation of bulk BN within the LDA approximation. To ensure an accurate treatment of the interatomic forces, we will use tight convergence criteria for the total energy in the scf procedure of $\text{conv_thr}=10^{-10}$ and for the maximum allowed forces for the resulting geometry, $\text{force_conv_thr}=0.0002$. You can use the input file `BN_b.in` as reference.



- (c) Perform a similar geometry optimization using the PBE functional. As pointed out in the lecture, PBE should perform very badly and increase the volume of the unit cell significantly. You will see that this has a significant effect on the basis set due to the fact that Quantum Espresso keeps the number of plane waves constant throughout the geometry optimization procedure. This will mean that you might to restart the calculations with several times, taking the optimized geometry of the previous run as initial geometry. Do this until the lattice constants change by less than 0.001 Angstrom.

You can also perform another geometry optimization using PBEsol, which performs better than PBE, but does still not give results of satisfactory accuracy.

- (d) Now do another geometry optimization but use the functional `optB88-vdW`, which includes van-der-Waals interaction through a non-local correlation functional. This can be invoked by `input_dft='vdw-df-obk8'`, see the input file `BN_d.in`. In order to run these calculations, you need a file `vdW_kernel_table` in the folder where you run your calculation. This file can be found in the folder of this exercise in the .zip file and should work on both HPC and home computers. You can also generate it yourself by running `generate_vdW_kernel_table.x` or `generate_vdW_kernel_table.exe`, which can be found in the folder where the Quantum Espresso programs are located.

Compare the results to experimental values and the LDA and PBE calculations from (b) and (c).

- (e) Now extract the binding energies of the BN layers for PBE and `opt88-vdW`.

To do this, create a new input file for an isolated layer of BN by removing one of the layers from the input file. Increase the lattice constant in z direction to 20 Angstrom. As the BN layers are flat, this also corresponds to a vacuum layer of 20 Angstrom between the BN layer in our unit cell and the periodic images in the neighboring unit cells in z- direction. You can use the input file `BN_e.in` as reference. Note that we now treat the z-direction as non-periodic, which means that we only sample one k-point in direction of the corresponding reciprocal lattice vector (this is only valid if the vacuum layer is large enough).

The binding energy that we are interested in is the difference between the total energy of the bulk crystal (calculated in (c) and (d)) with two layers per unit cell and the combined total energy of two isolated layers of BN.

Exercise 3: Distorted graphene nanoribbon

In this exercise, we will further reduce the dimensionality of the considered systems and will do calculations on a simple graphene nanoribbon, which is a system of potential interest for nanotechnology applications. Such nanoribbons can be understood as thin stripes cut out from graphene sheets, which are only periodic in one direction, but finite in the other two spatial directions. Let's assume we want to build a thin nanoribbon and do simulations on it, but don't know the lattice constant or the atomic positions. We can derive those with some geometrical relations from the well-known hexagonal structure of graphene, but let's assume that we were lazy and just drew the atoms into the unit cell using a suitable program (PWGui or BURAI can be used for this purpose, you can find links on the Quantum Espresso webpage under <https://www.quantum-espresso.org/resources/tools>).

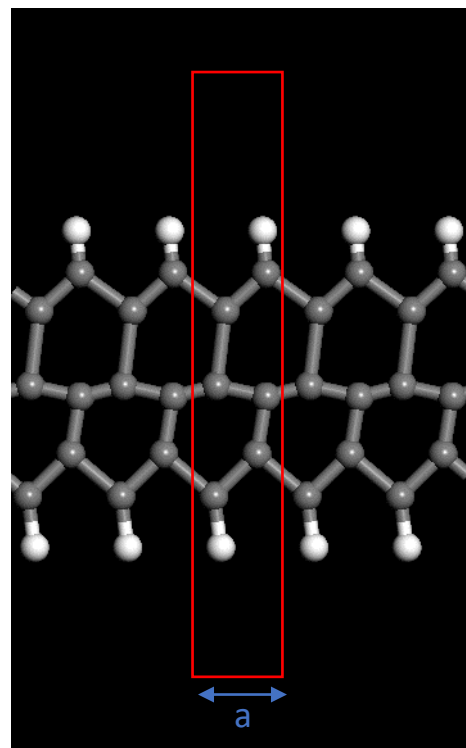
The resulting, distorted geometry is depicted in the figure.

Obviously, this structure won't be very helpful for understanding any physical property of this nanoribbon. We hence need to perform a geometry optimization in order to restore sensible atomic positions and the lattice constant.

- (a) The distortion of the atomic structure is quite bad, which commonly leads to problems, if we want to optimize both lattice vectors and atomic positions simultaneously. Actually, it should be no problem here, but let's follow the usual procedure for the optimization of 'bad' input geometries: We will first perform a pre-relaxation of the atomic positions, while keeping the lattice vectors fixed, i.e. we perform a calculation='relax' run. Our goal is to move the atoms around in order to sufficiently reduce the interatomic forces before we add the additional lattice degrees-of-freedom.

For this purpose, we can use a larger tolerance for interatomic forces than we would use in order to obtain well converged geometries, for example `force_conv_thr=0.001`. As we are not interested in very accurate forces anyway, we can also reduce the convergence threshold for the total energy in the SCF procedure compared to what we used in Exercise 2.

To save some time, you do not need to perform a convergence test for this calculation, you can use the cutoff energy and k-point sampling already included in the input file `C_a.in`, which also contains the distorted initial geometry of the ribbon. Note that the nanoribbon is only periodic along the x-axis, while we include vacuum layer along the y- and z-



axes. This means that the Brillouin zone is nearly one-dimensional and we only need to sample along the reciprocal lattice vector in x-direction and can set the sampling in the other two directions to 1.

Run the calculation. From inspecting the input file, you will notice that the SCF procedure runs into difficulties after a few geometry relaxation steps and does not converge even after 100 SCF steps, causing Quantum Espresso to terminate the calculation with an error. The reason for this is that the band gap of this nanoribbon is very small and the distorted nanoribbon might even be metallic (you can see this from comparing the single-electron energies in the QE output with the value of the Fermi energy). Quantum Espresso noticed this and already included a number of unoccupied bands in the calculation, as you can see from the header of the output file, where the number of bands included in the calculation is larger than $n_{\text{electrons}}/2$. Still, the SCF procedure encountered some charge sloshing. In order to solve the problem, we could try to reduce the mixing weight for the density mixing treatment in the SCF procedure (keyword `mixing_beta` in the `&electrons` block). It is, however, easier to solve the problem by smearing of the band occupations. Let's use a 'cold smearing' with a smearing width of 0.01 Ry. The necessary adjustments can be found in the input file *C_a_smearing.in*.

Warning: these calculations might run for a while if you are using a home computer.

- (b) Now replace the initial distorted geometry in the input file by the pre-relaxed atomic positions obtained in (a). Starting from this geometry, we will perform the real geometry optimization with tighter convergence criteria and allowing both the atomic positions and the lattice to relax. Modify the input file accordingly, use a force threshold of `force_conv_thr=0.0004` and a threshold for the pressure of `press_conv_thr=0.01`, and reduce `conv_thr` to the value used in Exercise 2. We will continue to use occupation smearing.

Now, we have two lattice vectors along the pseudo-non-periodic direction, i.e. along the y- and z-axis. If we do the relaxation, these will change slightly and might even be strongly reduced, if the vacuum thickness is not large enough, so that there is significant interaction between our nanoribbon and its periodic images in neighbouring unit cells. We can prevent the relaxation of these two lattice vectors by imposing constraints on the geometry optimization. This can be done by adding the keyword `cell_dofree` to the `&cell` block.

`cell_dofree` defines what coordinates of the lattice vectors are allowed to move, it is also possible to constrain the lattice vectors such that the unit cell volume or shape stays constant during the geometry optimization. For instance, we could have used the additional constraint in Exercise 2(e) to only allow the lattice vectors to change in x- or y-direction. The possible values of `cell_dofree` can be found here: https://www.quantum-espresso.org/Doc/INPUT_PW.html#idm1033

For our purpose, we only want the x coordinate of the first lattice vector to change. This corresponds to `cell_dofree='x'`. you can use the input file *C_b.in* as reference.

Run the geometry optimization. This should lead to a nice geometry with small stresses and interatomic forces.

- (c) We will now inspect some properties of the relaxed nanoribbon by making use of the postprocessing capabilities of Quantum Espresso. We will first plot the calculated groundstate electron density of the system. The .zip file contains a suitable input file for this

purpose, *pp.in_density*. First inspect the input file, which is quite short. It contains two blocks:

- `&inputpp` contains information about what quantity we are interested in and where the postprocessing program *pp.x* can find the necessary data from a previous *pw.x* run. For this, we define a `prefix` and an `outdir`, which have to be consistent with the values given in the output file used for the geometry optimization run.
`plot_num` determines the quantity that is supposed to be extracted.
`plot_num=0` extracts the electron density. You can also try `plot_num=9`, which will extract the electron density and subtract the electron density from a superposition of isolated atoms. This will highlight the changes in electron density that arise from interatomic bonding.
- The block `&plot` determines how the extracted data will be plotted. `iflag=3` states that we want a 3D plot of the data that uses the spatial grid used in the *pw.x* calculations. `output_format=5` states that we want the data to be written in XCrysDen format. `fileout` gives the filename of the output file.

The available options that can be used in the input files to *pp.x*, particularly the list of extractable properties can be found at https://www.quantum-espresso.org/Doc/INPUT_PP.html.

Now extract the electron density by running the command

```
PATH_TO_PP/pp.x <pp.in_density
```

This calculation should be finished rather quickly. Plot the output file *C_density.xsf* using VESTA or XCrysDen (in case of Linux). The density around the interatomic bonds is clearly visible in the plot.

- (d) We can also extract the Kohn-Sham potential of the system. Here, we will plot it along the y-axis, which will allow us to easily see, how the potential decays away from the nanoribbon. Notice the comments in the input file *pp.in_pot*. The starting point of the plotting line (keyword `x0`) chosen such that the plotting line passes through the positions of the hydrogen atoms (`x0(1)=0.264*2.46`).

Extract the potential and plot the result in gnuplot or a similar program of your choice. Notice that the potential clearly depicts the location of the atoms, but decays quickly into the vacuum and saturates at a constant value towards the unit cell boundary. We will use this (or a similar) property in a future exercise for the calculation of work functions.

Final remark: if you are using the HPC, it might be beneficial to delete the `TMP_DIR` for the calculations, as they can eat up lots of storage space.

