

Exercises for

Computational Modeling of Quantum Materials

Dr. Dominik Juraschek

Fall semester 2023/24

Exercise 1: Monday, January 15th, 2024

1. Running a basic VASP calculation

Log into the workstation and go to the Calculations folder in your home directory. From the /home/vasp_input directory, copy the Si_input_files folder to your Calculations folder. You will find four files in that folder:

```
$ ls
INCAR KPOINTS POSCAR POTCAR
```

These are the basic input files for a VASP calculation. Anything and everything about them can be found on the [VASP Wiki](#). Here are the relevant parts for our calculation.

1.1 POSCAR file

The POSCAR contains the unit cell (primitive or nonprimitive) of the material that we want to compute properties for. In our example, it reads:

```
Silicon
5.5
  0.0  0.5  0.5
  0.5  0.0  0.5
  0.5  0.5  0.0
Si
2
Direct
  0      0      0
  0.25  0.25  0.25
```

The first line is a comment, the second line marks the lattice-vector scaling, and lines 3-5 contain the three direct lattice vectors, with (a_{1x}, a_{1y}, a_{1z}) components in each row, that span the unit cell. Line 6 specifies the types of atoms and line 7 the number of atoms in the unit cell. Line 8 specifies that the atomic coordinates are given in terms of the lattice vectors and lines 9 and 10 mark the atomic coordinates in the unit cell, again as (x, y, z) components.

To visualize the structure file, you can install VESTA on your laptop, a crystallography tool. You can find it at the [VESTA webpage](#) and it's an easy installation.

1.2 KPOINTS file

The KPOINTS file specifies how densely the Brillouin zone will be sampled. Many quantities are calculated as a sum over all occupied single-particle Kohn-Sham states,

$$n(\mathbf{r}) = \sum_{n,\text{occ.}} \phi_n^*(\mathbf{r})\phi_n(\mathbf{r}), \quad E_{\text{KS}} = \sum_{n,\text{occ.}} \varepsilon_n. \quad (1)$$

In periodic systems, the summation can be written as an integration in reciprocal space,

$$\sum_{n,\text{occ.}} \rightarrow \frac{1}{(2\pi)^3} \sum_i \int_{\text{1.BZ}} d\mathbf{k} \theta(\varepsilon_F - \varepsilon_{\mathbf{k},i}), \quad (2)$$

where θ is the Heaviside step function and ε_F the Fermi energy. In practice, the Kohn-Sham equations are solved for a number of k-points in reciprocal space and the integral is then approximated by a sum over all grid points,

$$\frac{1}{(2\pi)^3} \int_{\text{1.BZ}} d\mathbf{k} \rightarrow \frac{1}{N_{\mathbf{k}}} \frac{1}{(2\pi)^3} \sum_{\mathbf{k}}. \quad (3)$$

The grid is determined by the KPOINTS file and in our case reads:

```
k-points grid for integration in reciprocal space
0
Gamma
9 9 9
0 0 0
```

The first line is again a comment and the second line specifies that the k-point grid will be generated automatically by an internal routine. Line 3 specifies that at least one of the grid points should sit at the center of the Brillouin zone (Gamma point). Line 4 contains the sample size of the k-point grid: each reciprocal lattice vector is sampled with 9 grid points in this case. Because Si in diamond structure is cubic, this grid will be cubic. The larger the unit cell, the less k-points are necessary to sample the Brillouin zone. Line 5 describes a manual shift of the k-point grid, which is not relevant for us however. The denser the k-point grid, the more accurate the sampling - we will therefore have to run some convergence tests with respect to the grid density.

1.3 INCAR file

The INCAR file contains most of the input that we have to specify for our calculation. In our example, it reads:

```
SYSTEM = Si diamond structure

### Parallelization

NCORE = 2
```

```

KPAR    = 1

### SCF

ENCUT   = 500
EDIFF   = 1E-8
NELM    = 60
ISYM    = 2
ISTART  = 0
ICHARG  = 2

### Partial occupancies

ISMEAR  = 0
SIGMA   = 0.1

### Spin polarization

ISPIN   = 1

```

The first line, despite looking like a variable, is a comment that reminds you which system this INCAR is for. The first block concerns parallelization. NCORE specifies the number of cores we will use and KPAR specifies the number of k-points that are treated in parallel. For this small of a system, two cores and one k-point at a time are more than sufficient. (The workstation has a relatively new Intel Xeon Gold processor.)

The next block concerns the self-consistent field (SCF) cycle. ENCUT defines the energy cutoff that is applied when the Kohn-Sham orbitals are sampled with a basis set of plane waves. We will discuss this in more detail in a future theory lecture. Generally, one can write

$$\phi_{\mathbf{k},i}(\mathbf{r}) = \frac{1}{\sqrt{V}} \sum_{\mathbf{K}} C_{\mathbf{k},i} e^{i(\mathbf{k}+\mathbf{K})\cdot\mathbf{r}}, \quad (4)$$

where \mathbf{K} is a reciprocal lattice vector and $\chi_{\mathbf{k}+\mathbf{K}} = e^{i(\mathbf{k}+\mathbf{K})\cdot\mathbf{r}}/\sqrt{V}$ is the plane-wave basis. The cutoff energy determines the amount of plane waves that are included through $|\mathbf{k} + \mathbf{K}| \leq \sqrt{2mE_{\text{cut}}}/\hbar$. The more plane waves are included, the more accurate the sampling becomes. We will therefore have to run some convergence tests with respect to the plane-wave cutoff energy.

EDIFF specifies the convergence criterion for the SCF cycle in electron volts (eV), here set to 10 neV for high accuracy. NELM specifies the maximum amount of SCF cycles that are to be performed to prevent an infinite loop in case no convergence is achieved, 60 will be more than enough for us. ISYM specifies whether symmetries of the crystal structure are to be applied to the charge density, 2 means yes here. ISTART specifies how the wave functions for the SCF cycle should be initialized, 0 means initialize

the basis set from scratch. ICHARG specifies how the charge density is initialized, 2 means initialize from a superposition of atomic charge densities.

The next block concerns how partial occupancies of the orbitals are to be treated and ISMEAR specifies the method of smearing for them. 0 means that a Gaussian smearing will be applied and SIGMA determines the width of the smearing in eV. The method of smearing is sensitive to whether the material is metallic or semiconducting/insulating and we therefore have to specify it explicitly.

In the last block, ISPIN determines whether spin-polarized calculations should be performed, which is mostly relevant for magnetic materials. 1 here means that no spin polarization is used.

1.3 POTCAR file

Finally, the POTCAR file contains the pseudopotentials of the types of atoms used in the calculation. We will discuss pseudopotentials in a future theory lecture, but in short, they contain information about the shell structure of the atoms and in particular an approximation of the potential generated by the core electrons. This approximation of the core-electron potential makes sure that the basis set that is used to construct the Kohn-Sham orbitals only has to take into account the valence electrons, thereby substantially reducing computational cost.

The head of the POTCAR for Si reads:

```
PAW_PBE Si 05Jan2001

(...)

POMASS = 28.085; ZVAL = 4.000 mass and valenz

(...)

Atomic configuration
6 entries
  n  l  j      E      occ.
  1  0  0.50  -1785.8828  2.0000
  2  0  0.50   -139.4969  2.0000
  2  1  1.50   -95.5546  6.0000
  3  0  0.50   -10.8127  2.0000
  3  1  0.50    -4.0811  2.0000
  3  2  1.50    -4.0817  0.0000
```

The first line marks the generation date of the pseudopotential file, as you can see it is more than twenty years old! PAW_PBE denotes that the projector augmented wave (PAW) method is used and that it is suitable for the generalized gradient approximation according to Perdew, Burke, and Ernzerhof (PBE). We will discuss what all these things mean in the following theory lectures.

Furthermore, we can see that $ZVAL = 4$ denotes the number of valence electrons. A brief comparison with the atomic configuration below tells us that the valence electrons consist of the 3s and 3p electrons of Si. There are no changes to be made in the POTCAR file, but we need to carry it along for every VASP calculation that we perform.

1.4 Output files

You can run VASP with the following command that uses the path to the executable and runs it in parallel using mpirun on two cores:

```
$ mpirun -n 2 /home/dominik/Dominik/VASP/vasp.6.3.2/bin/vasp_std
```

Best, you save the command in a jobscript to avoid retyping it all the time, for example in a file job_vasp.sh that contains:

```
#!/bin/bash
```

```
mpirun -n 2 /home/dominik/Dominik/VASP/vasp.6.3.2/bin/vasp_std
```

Now, just add executable rights to the file and it's ready to go:

```
$ chmod +x job_vasp.sh
$ ./job_vasp.sh
```

You will see on the terminal output that VASP initializes and then enters the main SCF loop, where the iterative scheme is run using a blocked Davidson algorithm (DAV):

```
running on      2 total cores

(...)

entering main loop
      N      E      dE      (...)
DAV:   1      0.682784784289E+01      0.68278E+01      (...)

(...)

DAV:  13      -0.108449986672E+02      -0.81291E-08      (...)
      1 F= -.10844999E+02 E0= -.10844999E+02  d E =-.149271E-07
```

N denotes the SCF cycle number, E the total energy, and dE the difference in total energy between cycles. You can see that it ends when the convergence criterion of $1E-8$ is achieved. The final output is the total free energy F that is given in eV. You will see that a lot of output files have been produced:

```
$ ls
CHG  CHGCAR  CONTCAR  DOSCAR  EIGENVAL  IBZKPT  INCAR  job_vasp.sh
KPOINTS  OSZICAR  OUTCAR  PCDAT  POSCAR  POTCAR  REPORT  vasprun.xml
WAVECAR  XDATCAR
```

Most of them are not relevant for us, and we will discuss the ones that are on the way. At the end of the OUTCAR file, you will find information about the time it took for the job to run, as well as the memory used. The SCF part of the terminal output is contained in OSZICAR, which serves as your reference file for the total energy. Not that if you start another VASP run, all output files are overwritten, so make sure to create different folders for different kinds of calculation!

2. Convergence testing for Si

Before computing actual physical quantities, we need to make sure that we are sufficiently sampling the Brillouin zone for k-point integration and the basis set with plane waves. We need enough k-points and a high-enough energy cutoff, so that the outcome of the SCF cycle is converged with respect to these parameters.

Task: Perform total-energy calculations for different values of ENCUT between 300 and 700 eV (in steps of 50 eV), as well as different densities of k-point grids in the KPOINTS file (n, n, n with n going from 4 to 15 in steps of 1). Extract the total energy from each of the runs and create two plots: one vs ENCUT and one vs n . For future calculations, choose ENCUT and KPOINTS parameters for which the total energy is converged. You may choose any program to plot the data.

Advice: Since VASP overwrites the output files with every run, it is advisable to perform different calculations in different folders so that all the data is saved. The easiest way to set this up is to write a bash script that allows you to generate input files, run the jobs, and extract the data. To have maximum control over the process, you should implement if-statements that separate the steps of input creation, job running, and data extraction, e.g. in the following general form:

```
#!/bin/bash

if [[ $1 = "-createinput" ]]; then

    (Loop for creating input files in different folders)

elif [[ $1 = "-runjobs" ]]; then

    (Loop for running VASP in the respective folders)

elif [[ $1 = "-extractdata" ]]; then

    (Loop for extracting the relevant data)

else

    echo "Tell me what to do!"

fi
```

Make sure to add comments to your script that explain what the different routines are doing so that you will be able to understand them also in the future. You can then run the script with the respective variables as input, for example:

```
$ ./convergence_script.sh -createinput
```

Bash scripts can take multiple variables, which gives you a lot of flexibility in tailoring your routines according to your needs.

Some of the most useful commands and functions involve for-loops for cycling through folders, sed for modifying files, awk for processing column-structured input and simple floating-point calculations, grep for extracting data from output files, tail/head for getting the last/first lines of a file, cut for separating strings, echo for printing variables, and cat for printing contents of a file. Please see the tutorials from the exercise preparations sheet.

3. Geometry optimization for Si

In order to compute the ground-state properties of a material, we need to make sure that the structure file (POSCAR) that we provide contains in fact the ground-state configuration. In order to find the ground-state geometry, we need to perform an optimization of the crystal structure that generally includes the atomic coordinates, shape of the unit cell, as well as the unit-cell volume. In VASP, internal routines move the ions or change the lattice vectors by a small amount and compute the total energy or the forces between the ions before and after. An iterative scheme (e.g. using a conjugate gradient algorithm) so minimizes the total energy or forces between the atoms until the (global) minimum is reached. For a highly symmetric system such as Si, the only relevant variable in this optimization procedure is the volume of the unit cell and we can do it “by hand”.

Task: Perform different total-energy calculations while changing the volume of the unit cell. (Look at the POSCAR - what’s the easiest way to make this change?) Be aware that if you change the volume to unrealistic values, the SCF cycle may not converge. Plot the total energy vs unit-cell volume for varying values around the one already provided in the example POSCAR. Refine the step width once you’ve identified a range wherein the minimum lies. Plot the total energy vs unit-cell volume. For future calculations, use the structure file with the volume that minimizes the total energy.

4. Band structure of Si

With the previous two tasks completed, we are now set to perform calculations of relevant physical quantities. First, we will compute the band structure of Si. In order to do so, first run a regular total-energy calculation and save the charge density, which is contained in the CHGCAR file.

Next, we will perform a band-structure run. We need to perform calculations along a specific trajectory in reciprocal space, for which we have to define a new type of KPOINTS file:

```
k-points for band structure L-G-X-U and K-G
```

```
10
```

```
line
```

```
reciprocal
```

```
0.50000 0.50000 0.50000 1
```

```
0.00000 0.00000 0.00000 1
```

```
0.00000 0.00000 0.00000 1
```

```
0.00000 0.50000 0.50000 1
```

```
0.00000 0.50000 0.50000 1
```

```
0.25000 0.62500 0.62500 1
```

```
0.37500 0.7500 0.37500 1
```

```
0.00000 0.00000 0.00000 1
```

The first line is again a comment. The second line specifies that 10 k-points per trajectory are to be used and line 3 tells VASP to run the KPOINTS file in line mode. Line 4 simply says that the reciprocal lattice points are given in terms of reciprocal lattice vectors (not in absolute cartesian coordinates), and the following lines define the different band points that we outline in the first line. The 1 at the end of each line means that every band point will have the same weight. Because not the entire Brillouin zone is sampled, this run itself does not produce a meaningful charge density and we need to read it in from the CHGCAR produced in the previous run. Make the following changes to the INCAR file and then run VASP:

```
ICHARG = 11
```

```
### Bands and DOS
```

```
LORBIT = 11
```

ICHARG=11 tells VASP to read in the existing charge density and LORBIT=11 specifies that the atomic-orbital projected wave-function character of each Kohn-Sham orbital (band) is printed into the PROCAR file. The structure of the PROCAR file is as follows:

```
PROCAR lm decomposed
```

```
# of k-points: 40 # of bands: 8 # of ions: 2
```

```
k-point 1 : 0.50000000 0.50000000 0.50000000 weight = 0.02500000
```

```
band 1 # energy -4.01541108 # occ. 2.00000000
```

ion	s	py	pz	px	dxy	dyz	dz2	dxz	x2-y2	tot
1	0.222	0.004	0.004	0.004	0.000	0.000	0.000	0.000	0.000	0.235
2	0.222	0.004	0.004	0.004	0.000	0.000	0.000	0.000	0.000	0.235


```

tot      0.445  0.009  0.009  0.009  0.000  0.000  0.000  0.000  0.000  0.000  0.471

band      2 # energy    -1.34331873 # occ.    2.00000000

ion       s    py    pz    px    dxy    dyz    dz2    dxz    x2-y2    tot
   1  0.101  0.030  0.030  0.030  0.000  0.000  0.000  0.000  0.000  0.191
   2  0.101  0.030  0.030  0.030  0.000  0.000  0.000  0.000  0.000  0.191
tot      0.202  0.060  0.060  0.060  0.000  0.000  0.000  0.000  0.000  0.000  0.383

(...)

k-point   2 :   0.44444444 0.44444444 0.44444444  weight = 0.02500000

band      1 # energy    -4.16861360 # occ.    2.00000000

ion       s    py    pz    px    dxy    dyz    dz2    dxz    x2-y2    tot
   1  0.218  0.005  0.005  0.005  0.000  0.000  0.000  0.000  0.000  0.232
   2  0.218  0.005  0.005  0.005  0.000  0.000  0.000  0.000  0.000  0.232
tot      0.436  0.009  0.009  0.009  0.000  0.000  0.000  0.000  0.000  0.000  0.464

band      2 # energy    -1.11594609 # occ.    2.00000000

ion       s    py    pz    px    dxy    dyz    dz2    dxz    x2-y2    tot
   1  0.105  0.030  0.030  0.030  0.000  0.000  0.000  0.000  0.000  0.195
   2  0.105  0.030  0.030  0.030  0.000  0.000  0.000  0.000  0.000  0.195
tot      0.210  0.060  0.060  0.060  0.000  0.000  0.000  0.000  0.000  0.000  0.390

(...)

```

The header provides some information about the number of k-points, bands, and atoms. The remainder of the file contains the k-points along the trajectories that we specified in the KPOINTS file, the energies of the bands (in eV), as well as the contributions to the Kohn-Sham bands from the individual atoms and their atomic orbitals.

Task: Extract the energies of the different bands for each k-point and plot the band structure for the entire trajectory in reciprocal space. See whether you can identify the band gap - is it a direct or indirect gap?

5. Density of states of Si

Next, we will compute the electronic density of states (DOS) of silicon. The DOS is very sensitive to energies and we will switch to a more accurate method for the treatment of partial occupancies in the k-point integration: the tetrahedron method, which samples the Brillouin zone as tetrahedra, within which the band energies are linearized in k . This allows a better sampling of complex Fermi surfaces than the Gaussian smearing method, but only works for semiconductors and insulators. Make the following changes

to the INCAR file (with respect to the total-energy calculation, not the band-structure calculation):

```
### Bands and DOS
```

```
LORBIT = 11
```

```
NEDOS = 1001
```

```
### Partial occupancies
```

```
ISMEAR = -5
```

LORBIT=11 tells VASP to print out the DOSCAR file that contains the density of states. NEDOS sets the number of energy bins at which the DOS is evaluated, 1001 should be sufficient. ISMEAR=-5 specifies the tetrahedron method for k-point integration.

In addition to switching to the tetrahedron method, we need to increase the density of the k-point grid in the KPOINTS file. Double the number of sampling points for each reciprocal lattice vector. Now run VASP and look at the output files. You will find the DOSCAR file, which has the following structure:

```

  2   2   1   0
0.2079688E+02  0.3889087E-09  0.3889087E-09  0.3889087E-09  0.5000000E-15
1.000000000000000E-004
CAR
Si diamond structure
  16.19303400   -7.32146350 1001      6.08921359      1.00000000
  -7.321   0.0000E+00  0.0000E+00
(...)

  16.19303400   -7.32146350 1001      6.08921359      1.00000000
  -7.321   0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  (...)
(...)

  16.19303400   -7.32146350 1001      6.08921359      1.00000000
  -7.321   0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  (...)
(...)

```

You can look up the meaning of the first four lines in the VASP Wiki. For our purpose, the important part begins after the SYSTEM specification as given in the INCAR (Si diamond structure). The following lines contain:

```

16.19303400      -7.32146350 1001      6.08921359      1.000000000
-7.321  0.0000E+00  0.0000E+00

```

(...)

The first line contains information about the energy range and the Fermi energy: E_{\max} , E_{\min} , NEDOS, E_{Fermi} , and a scaling factor (here 1). The following lines contain each the energy E , DOS(E), and cumulative DOS for the different energy bins. After this large list, the atomic-orbital resolved partial DOS for the first atom in the unit cell begins:

```

16.19303400      -7.32146350 1001      6.08921359      1.000000000
-7.321  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  (...)

```

This data contains the energy E and then the partial DOS for the s , p_y , p_z , p_x , d_{xy} , d_{yz} , $d_{z^2-r^2}$, d_{xz} , and $d_{x^2-y^2}$ orbitals. After this list, the same data is printed for the second atom of the unit cell, and so on.

Task: Plot the total DOS, as well as s - and p -orbital resolved partial DOS for a range of ± 6 eV around the Fermi energy with data extracted from the DOSCAR file. How large is the band gap? Compare the value of the band gap to common literature values, is it accurate?