Dr. Michelle Ernst, Dr. Štěpán Marek, Dr. Jan Wilhelm

Exercises using the CP2K package:

Geometry Optimization, Molecular Dynamics and Band Structure

Exercise 1: Electronic energy of the L-alanine crystal

Most parts of this exercise are adapted from the cubic Si example on the CP2K "How to" website: How to Calculate Energy and Forces

As a first step, we perform a static self-consistent Kohn–Sham density functional theory (DFT) calculation to obtain the electronic energy. Our model system is the L-alanine crystal, consisting of 13 atoms per molecule and 4 molecules in the unit cell.

The example input and output files are in `https://github.com/JWilhelm/CP2K_Computational_Methods_in_Crystallography`. The calculations were performed with CP2K version 2025.1.

Input files

We first look at the input files required for this calculation, which are:

- `Lalanine.inp`: the main input file for the calculation, which defines the system and the job parameters
- `Lalanine.cif`: the coordinate file when defined externally

Parameter files:

- `BASIS_SET`: file containing parameters for the basis sets that can be used by CP2K for this calculation
- `GTH_POTENTIALS`: file containing parameters for the pseudopotentials that can be used by CP2K for this calculation

A list of basis set and pseudopotential files may be found in `cp2k/data/`, which comes with a CP2K source release. These should cover most of the commonly used elements.

Input structure

Let us look at the main input: `Lalanine.inp` in detail. The filename and extension can be chosen freely. The input file is structured into ordered blocks and keywords, the order of which are unimportant. Each input block is referred to as a "section" in this tutorial, and some sections are "subsections" of other sections. Full definitions of the input file format and keywords are available via the CP2K input reference manual.

The main sections in the input file are:

- GLOBAL: contains general options for the CP2K run, such as the name of the job, the type of run etc.
- FORCE_EVAL: contains all parameters associated with the evaluation of forces on atoms, including the initial atomic coordinates.

We now look at each section in detail, starting with the GLOBAL section from Lalanine.inp:

```
&GLOBAL
  PROJECT LALANINE
  RUN_TYPE ENERGY
  PRINT_LEVEL MEDIUM
&END GLOBAL
```

Here we perform a static energy calculation; in this case, we must set RUN_TYPE to ENERGY. The keyword PROJECT sets the root name of the calculation, in this case L-alanine. Any output files automatically generated by CP2K will have the name prefixed by L-alanine. PRINT_LEVEL controls the default verbosity of the main output of CP2K, in this example it is set to "medium".

Next, we examine the FORCE_EVAL section line-by-line.

```
METHOD Quickstep
```

The keyword METHOD specifies QUICKSTEP as the approach for evaluating the forces on the atoms, i.e. density functional theory using the Gaussian and Plane Waves (GPW) method.

The subsection SUBSYS defines the simulation unit cell and the initial coordinates of the atoms in the calculation.

```
&SUBSYS
    &CELL
      ABC 5.94 12.274 5.806
      ALPHA_BETA_GAMMA 90.0 90.0 90.0
    &END CELL
    &COORD
      H       2.524500    3.041497    1.962428
      ...
    &END COORD
    &KIND H
      ELEMENT   H
      BASIS_SET DZVP-GTH-PBE
      POTENTIAL GTH-PBE-q1
    &END KIND
    &KIND C
      ELEMENT   C
```

```
      BASIS_SET DZVP-GTH-PBE
      POTENTIAL GTH-PBE-q4
    &END KIND
    &KIND N
      ELEMENT   N
      BASIS_SET DZVP-GTH-PBE
      POTENTIAL GTH-PBE-q5
    &END KIND
    &KIND O
      ELEMENT   O
      BASIS_SET DZVP-GTH-PBE
      POTENTIAL GTH-PBE-q6
    &END KIND
  &END SUBSYS
```

The subsection CELL defines the simulation unit cell used in a calculation. In this example, we have a tetragonal cell with $a = 5.94$ Å, $b = 12.274$ Å, $c = 5.806$ Å.
The initial atomic coordinates are specified in the COORD subsection. The default input format for atomic coordinates in CP2K is:

```
<ATOM_KIND> X Y Z
```

where X, Y and Z are Cartesian coordinates in Å. This can be changed by configuring keyword SCALED to .TRUE. in the COORD subsection, which makes the coordinate input X Y Z fractional with respect to the lattice vectors. One can also change the unit for the Cartesian coordinates by setting the keyword UNIT within the subsection. <ATOM_KIND> should be a label that corresponds to the definition of the elements in the KIND subsections.

Task: Obtain the CIF file from the WebCSD (Refcode LALNIN23) and load it into CP2K in place of manual coordinates.

The subsection KIND gives definitions of elements in the calculation. There must be one KIND subsection per element. In this example, we have defined the basis set to be DZVP-GTH-PBE (double-ζ with polarisation basis optimised for Goedecker–Teter–Hutter PBE pseudopotentials); and the pseudopotentials such as GTH-PBE-q4 (Goedecker–Teter–Hutter PBE pseudopotential with 4 valence electrons).
The basis set and pseudopotential names must correspond to an existing entry in the corresponding basis set and pseudopotential files defined by the BASIS_SET_FILE_NAME and POTENTIAL_FILE_NAME keywords in the DFT subsection of the FORCE_EVAL section.

Task: Identify where these files are located.

After the SUBSYS section in Lalanine.inp comes the DFT subsection, which controls all aspects of the self-consistent Kohn–Sham density functional theory calculation. This subsection is relevant only if the METHOD keyword in FORCE_EVAL is set to QUICKSTEP.

```
BASIS_SET_FILE_NAME  BASIS_SET
POTENTIAL_FILE_NAME  GTH_POTENTIALS
```

As already mentioned above, the keywords BASIS_SET_FILE_NAME and POTENTIAL_FILE_NAME set the files that contain basis set and pseudopotential parameters.

```
&QS
  EPS_DEFAULT 1.0E-10
&END QS
```

General control parameters for QUICKSTEP are specified in the QS subsection. The parameter EPS_DEFAULT defines the global default tolerance applied throughout the calculation. More specific tolerance parameters (EPS_*), if present, override this global value.

```
&MGRID
  NGRIDS 4
  CUTOFF 300
  REL_CUTOFF 60
&END MGRID
```

The MGRID subsection defines how the integration grid used in QUICKSTEP calculations should be set up. QUICKSTEP employs a multigrid scheme to represent Gaussian functions numerically: Narrow and sharp Gaussians are mapped onto finer grids than wider and smoother Gaussians. In this case, we are telling the code to set up 4 levels of multigrids, with the planewave cutoff of the finest grid set to 300 Ry, and with the grid spacing underneath any Gaussian functions to be finer than the equivalent planewave cutoff of 60 Ry. The users should read the tutorial "Converging the CUTOFF and REL_CUTOFF" for details on how these parameters affect the grid constructed, and how to define a sufficient grid for their calculation.

Task: Increase the grid parameters and compare how this affects both the energy and the computational time.

```
&XC
  &XC_FUNCTIONAL PBE
  &END XC_FUNCTIONAL
&END XC
```

This subsection defines which exchange–correlation density functional we want to use. In this case we choose the PBE functional, which is consistent with the basis set and pseudopotential we have chosen.

```
&SCF
  SCF_GUESS ATOMIC
  EPS_SCF 1.0E-7
  MAX_SCF 300
  &DIAGONALIZATION
```

```
    ALGORITHM STANDARD
  &END DIAGONALIZATION
  &MIXING
    METHOD BROYDEN_MIXING
    ALPHA 0.4
    NBROYDEN 8
  &END MIXING
&END SCF
```

The SCF subsection defines all the settings related to methods used to find a self-consistent solution of the Kohn–Sham DFT formalism.
SCF_GUESS specifies how the initial trial electron density function $\rho(\vec{r})$ is generated. In this example (ATOMIC), the initial density is obtained from overlapping atomic charge densities. A good starting point for the electron density in the self-consistency loop is important for achieving convergence efficiently. EPS_SCF sets the tolerance of the charge density residual. This overrides the EPS_DEFAULT value set in the QS subsection. MAX_SCF sets the maximum number of self-consistency loops QUICKSTEP is allowed to perform for each ground-state energy calculation.

```
&DIAGONALIZATION ON
  ALGORITHM STANDARD
&END DIAGONALIZATION
```

The DIAGONALIZATION subsection instructs CP2K to use the traditional diagonalisation method for obtaining the ground-state Kohn–Sham energy and electron density. The subsection heading also takes an argument; in this case it is set to ON, which is equivalent to .TRUE. or T, and enables diagonalisation. If the argument is omitted, the default is also .TRUE.. The alternative to diagonalisation is the Orbital Transform (OT) method. In that case, the user should either delete the DIAGONALIZATION block or set the heading to OFF (or .FALSE.), and instead add the OT subsection. The keyword ALGORITHM specifies the diagonalisation algorithm; here STANDARD means the standard LAPACK/SCALAPACK routines are used.

```
&MIXING T
  METHOD BROYDEN_MIXING
  ALPHA 0.4
  NBROYDEN 8
&END MIXING
```

The MIXING subsection defines the parameters for charge mixing in a self-consistency calculation. It can take a value of either .TRUE. (T) or .FALSE. (F), which switches charge mixing on or off. The default is .TRUE.. Note that this subsection only applies when using the diagonalisation method; the OT method uses a different approach to charge mixing, which is explained in other tutorials. The keyword ALPHA sets the mixing parameter: in this example, 40% of the output density is mixed with 60% of the input density to form the new input density in the next SCF iteration. The keyword METHOD specifies the mixing method; here Broyden mixing is used. The keyword NBROYDEN (an alias for NBUFFER) sets the number of previous iterations to be stored in the Broyden mixing algorithm.

Running the calculation and analysing the output files

To run the calculation, the user needs to have a working CP2K package compiled in the system `PATH`. The calculation can be started with the command:

```
cp2k.psmp -o Lalanine.out Lalanine.inp &
```

This launches CP2K in the background. The option `-o` redirects the CP2K output to the file `Lalanine.out`. Note that the `-o` option appends output from successive runs to the same file. If you want to start from scratch, you should delete `Lalanine.out` before launching a new calculation.

After the job has finished, the following files should be present:

- `Lalanine.out`
- `L-alanine-RESTART.wfn`

The file `Lalanine.out` contains the main output of the job. The file `Lalanine-RESTART.wfn` contains the final Kohn–Sham wavefunctions from the calculation.
If you want to start a new calculation from previously calculated wavefunctions, change the keyword `SCF_GUESS` in the `SCF` subsection of the `FORCE_EVAL` section to:

```
SCF_GUESS RESTART
```

This requires that the new calculation uses the same `PROJECT_NAME` as the one that generated the restart files. Otherwise, the restart wavefunction files need to be renamed accordingly.

Task: Try this and check how it affects the computational time.

In the output file, CP2K reports quantities such as the number of electrons, the number of occupied orbitals, the SCF convergence behaviour, the total energy, and the atomic forces. During the class we will examine these sections of the output in detail.

Exercise 2: Geometry optimization of the L-alanine crystal

To perform a geometry optimisation, we need to add a third main section to the input file:

- `MOTION`: contains all information related to structural changes during a calculation, such as geometry optimisation, cell optimisation, or molecular dynamics.

The relevant input for geometry optimisation is:

```
&MOTION
  &GEO_OPT
    OPTIMIZER BFGS
    MAX_ITER 100
  &END GEO_OPT
&END MOTION
```

The subsection GEO_OPT controls how the geometry optimisation is carried out. Here, the keyword OPTIMIZER specifies the optimisation algorithm; in this example we use BFGS, a quasi-Newton method that is efficient for molecular and solid-state systems. The keyword MAX_ITER sets the maximum number of optimisation steps (here 100). The optimisation will stop earlier if the convergence criteria are reached.

Task: Run the geometry optimisation of the L-alanine crystal. What do you have to do except adding the MOTION section? After the run, inspect the optimised structure using Mercury, VMD, VESTA, or another molecular visualisation program.

Exercise 3: Electronic band structure of monolayer MoS_2

The electronic band structure gives the relation between crystal momentum $\mathbf{k}$, band index n and the energy $\varepsilon_{n\mathbf{k}}$ of an electron in a crystal, according to Bloch's theorem. The electronic band structure can be computed using DFT in an approximate way via the Kohn-Sham equations

$$\left(-\frac{\nabla^2}{2m} + v_{\text{ext}}(\mathbf{r}) + v_{\text{Hartree}}(\mathbf{r}) + v_{\text{xc}}(\mathbf{r})\right)\psi_{n\mathbf{k}}(\mathbf{r}) = \varepsilon_{n\mathbf{k}}\psi_{n\mathbf{k}}(\mathbf{r})\,. \tag{1}$$

When using the standard exchange-correlation (xc) functionals like PBE, the band gap between the occupied valence bands and the empty conduction bands is usually underestimated with respect to experiment. Nevertheless, PBE often gives the correct band ordering, dispersions (i.e., curvature as function of $\mathbf{k}$), and orbital character of the bands.

In this exercise, we compute the band structure of monolayer MoS_2, a two-dimensional crystal which has been discovered in 2010 [doi:10.1103/PhysRevLett.105.136805].

(a) A CP2K input for a DFT band structure calculation can be found here:

`https://github.com/JWilhelm/CP2K_Computational_Methods_in_Crystallography/tree/main/3_MoS2_band_structure_exercise`

The atomic positions and the cell size need to be filled for MoS_2. To do so, navigate to the Computational 2D Materials Database (C2DB),

`https://c2db.fysik.dtu.dk/`

and search for MoS_2:

`https://c2db.fysik.dtu.dk/material/1MoS2-1`

Download the "XYZ" file (top right) which contains the atomic positions. Also fill in the lengths of the cell and the angles, as given on the C2DB website. Then run CP2K:

```
cp2k.ssmp DFT_bandstructure.inp | tee cp2k.out
```

(b) Check your output by comparing to the solution

https://github.com/JWilhelm/CP2K_Computational_Methods_in_Crystallography/tree/main/3_MoS2_band_structure_solution

The PBE band structure is contained in the file `bandstructure.bs`. You can plot the PBE band structure using a plotting script available via github; obtain the script via

```
git clone https://github.com/stefabat/cp2k-scripts
```

Then run the plotting script via

```
python3 cp2k-scripts/bin/cp2k_plot_bands.py bandstructure.bs --energy_range -2 3
```

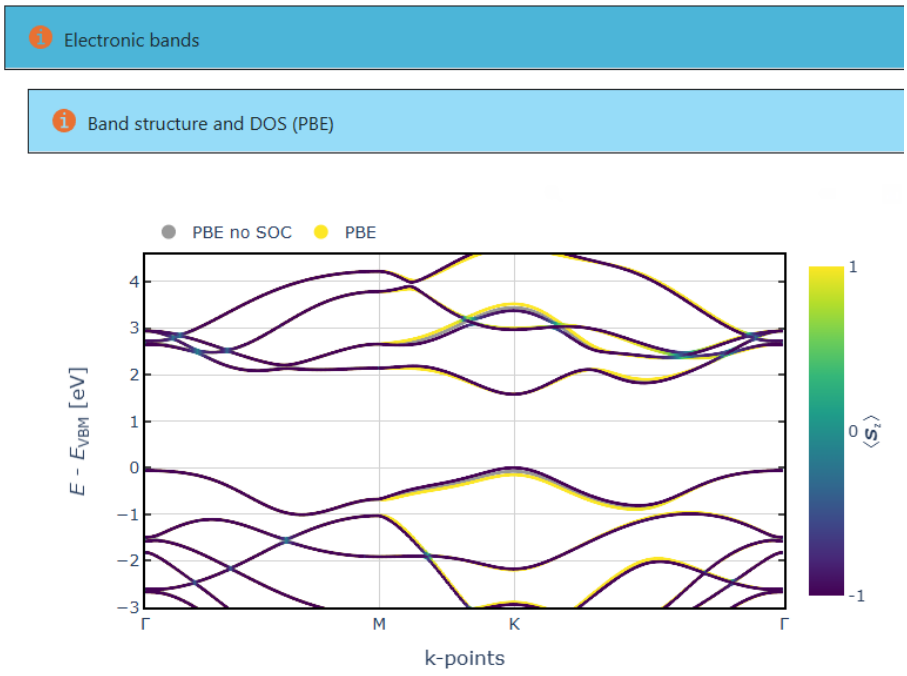
and compare your plot to the png file provided in the solution directory

https://github.com/JWilhelm/CP2K_Computational_Methods_in_Crystallography/tree/main/3_MoS2_band_structure_solution

Also compare the k-path we have chosen in the input to the k-path of the hexagonal crystal structure available in the appendix of

https://www.sciencedirect.com/science/article/pii/S0927025610002697

(c) Compare your DFT/PBE band structure to reference data, for example C2DB:

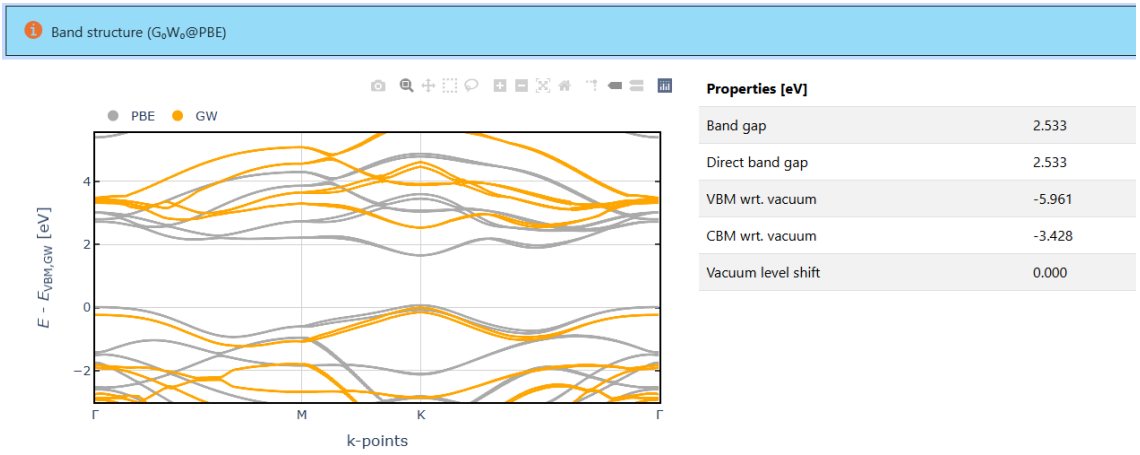


Compare also the gap (1.580 eV) on C2DB to the output of the plotting script.

(d) The following parameters are crucial to be tested for convergence (in case time permits):

- Basis set size: You can test larger basis sets like `TZV2P-MOLOPT-PBE-GTH-q6` (for S) and `TZV2P-MOLOPT-PBE-GTH-q14` (for Mo), contained in the basis file `BASIS_MOLOPT_UZH` (`https://github.com/cp2k/cp2k/tree/master/data`)
- *k*-point mesh: You can test larger *k*-grids, for example `SCHEME MONKHORST-PACK 16 16 1` instead of `SCHEME MONKHORST-PACK 8 8 1`.

(e) Compare your DFT/PBE band structure to the *GW* method, a higher level method deriving from Green's function theory [doi:10.3389/fchem.2019.00377], for example from C2DB:



What is the main difference of the *GW* band structure compared to DFT/PBE?

In case you like to execute *GW* band structure calculations of 2D materials with CP2K, you can check out doi:10.48550/arXiv.2507.18411 (Fig. 1, Fig. 4, Fig. 10, Table I) and the corresponding input and output files on github:

`https://github.com/RemiPasquier/Inputs_Outputs_Periodic_Small_Cell_GW_TMDC`

Can you identify in this paper the effect of spin-orbit coupling on the band structure?

Exercise 4: Molecular dynamics of water

In this exercise we perform a short ab initio molecular dynamics (AIMD) simulation of bulk liquid water and then switch to a semiempirical model (xTB) to reduce the computational cost. The input `water_dft.inp` and coordinate file `water.xyz` are provided in the GitHub repository.

AIMD (DFT) run

Use the AIMD input provided in the repository (see `water_dft.inp` and `water.xyz`). This setup performs an NVT simulation at 300 K with PBE+D3, GPW, GLE thermostat, and writes trajectory and energy files.

Task: Run a 20-step AIMD for water using the provided input. How long does it take on your machine? Inspect `water_dft-pos-1.xyz` and `water_dft-1.ener`.

Switching to a faster model (xTB)

A second input is provided that uses the xTB method (see `water_xtb.inp`). This keeps the same MD settings (ensemble, thermostat, timestep, system) but changes the electronic structure method for speed.

Task: Run the xTB input and compare wall time and SCF behaviour to the AIMD run.