FYS-4096 Computational Physics: Exercise 4

Ilkka Kylänpää

Week 5

General:

- You should have access to `ex4_files` on teacher's repo.
- There you will find files `spline_class.py`, `read_xsf_example.py`, `dft_chargedensity1.xsf`, `dft_chargedensity2.xsf`, `signal_data.txt`, and `fft_example.py` that are needed or can be useful in these exercises.
- **Deadline always before noon on Monday the following week!** That is, for these problems before noon on Monday February 8.
- Better to send unfinished in time, than finished, but past deadline.

Problem 1: Visualizing crystal structure and electron density with XCrySDen and/or VESTA

- Install XCrySDen and/or VESTA on your computer.
- Download `dft_chargedensity1.xsf` and `dft_chargedensity2.xsf` from the repo.
- Play around with the possible options: rotation, cell repetition, atom information, color schemes, etc.
- Make representative figures (e.g., png figures) of the crystal structures with and without electron density isosurfaces. Include the figures to your repo.
- Choose three atoms and calculate their interparticle distances and respectice angles using XCrySDen and/or VESTA. Write the distances and the angle into a file called `distance_data.txt`.

Problem 2: Determining the number of electrons in the simulation

- Read the density and the lattice info, e.g., by using the provided `read_xsf_example.py` module.
- For both structures present in Problem 1, calculate the number of electrons in the simulation cell.
- For both structures determine the reciprocal lattice vectors.

Problem 3: Electron density along a line (1/2)

- For the electron density `dft_chargedensity1.xsf` determine the electron density along a line from $\mathbf{r}_0 = (0.1, 0.1, 2.8528)$ to $\mathbf{r}_1 = (4.45, 4.45, 2.8528)$ using 500 uniformly spaced points.
- Using XCrySDen or VESTA consider what this line corresponds to, and save a descriptive figure of the situation (from XCrySDen or VESTA).
- Make a figure of the electron density along the line. Include the figure to your repo.

Problem 4: Electron density along a line (2/2)

- For the electron density `dft_chargedensity2.xsf` determine the electron density along two lines using 500 uniformly spaced points:
  - first line is from $\mathbf{r}_0 = (-1.4466, 1.3073, 3.2115)$ to $\mathbf{r}_1 = (1.4361, 3.1883, 1.3542)$
  - second line goes from $\mathbf{r}_0 = (2.9996, 2.1733, 2.1462)$ to $\mathbf{r}_1 = (8.7516, 2.1733, 2.1462)$
- Using XCrySDen or VESTA consider what these lines correspond to, and save a descriptive figure of the situations (from XCrySDen or VESTA).
- Make figures of both electron densities along the line. Include the figures to your repo.

Example: Fourier transform with Python's numpy

- Determine the frequencies in the time varying signal in the file `signal_data.txt` using the `fft_example.py` file. You are also interested in removing undesired frequencies, which in this example are considered to be frequencies below 40 Hz and above 60 Hz.
- The data in the `signal_data.txt` file is given as (t, f), i.e., the first column is the time from $-T/2$ to $T/2$ and the second column has the signal $f(t)$.
- For reading in the data you can use for example

```
data = np.loadtxt('signal_data.txt')
t = data[:,0]
f = data[:,1]
```

- Notice: saving such data could be done, e.g., as

```
np.savetxt('signal_data.txt',np.transpose([t,f]))
```

Returning your exercise

1. Make a new folder “exercise4” to your existing Computational Physics repo.
2. Create a file solvedProblems.txt in folder “exercise4”. Inside it, write a comma separated list of problems you have solved, e.g., 1,2,3x.
3. Make sure all your source files (and figures created by other software) are under version control and push them to GitLab.
4. Push your commits (and possible tags) to GitLab before noon on Monday February 8:
   `git push --all && git push --tags`