Home Assignment 2
Structure of Sodium Nanoclusters (20 points)

Magnus Rahm
rmagnus@chalmers.se

January 2020

Introduction

In this exercise, you will study the atomic and electronic structure of nanoclusters of sodium (Na) atoms using density functional theory (DFT) as implemented in GPAW. Nanoclusters are collections of atoms that are too large to be called molecules but too small to be referred to as nanoparticles. Na is very useful as a "model element" in DFT, since it for most practical applications can be treated as an element with only one valence electron while the remaining electrons can be "hidden" as parts of a screened core. This makes the calculations cheap, while most of the interesting physics is still captured.

When studying nanoclusters on the atomic scale, one is immediately faced with the question how the atoms should be arranged in space, i.e., what is the shape of the cluster in equilibrium? This is an optimization problem which has been studied a long time and is often used as a model problem when testing new optimization methods. In this exercise, you will use a genetic algorithm (GA) to optimize the structure of Na clusters with 6, 7, and 8 atoms. As the many details of GA is outside the scope of this course, it will here be used mostly as a black box that hopefully provides you with a reasonable result, which you will then analyze and discuss. Such a workflow is very common in computational materials physics. If everyone would implement everything we would not get much done.

How to succeed

Key to this exercise is to use the documentation of ASE and GPAW and whatever you find useful on the internet. DFT codes, and other codes used by physicists, are usually written by people who are not trained or paid to write superb documentation (`ASE` and `GPAW` are probably above average in this regard). It is considered part of the exercise to find out some things yourself[1]. If you find yourself cursing and feeling annoyed by inaccurate documentation and scripts that crash, you are probably on the right track.

Preferably you should work in pairs.

[1] Of course you are still allowed to ask questions!

Your report

Your report does *not* have to be a full-fledged report with introduction, theory, etc. It suffices to answer the below questions one by one.

Hand in your report as a PDF in Canvas. It is sufficient that one of you hands it in.

Task 1: Run the genetic algorithm (no points, but has to be done for what follows)

In the course git repository (in the `Na-clusters-GA-search` folder) you will find two scripts that together perform a search for optimal atomic structures of Na clusters. Take a look at these scripts and run them so that you optimize clusters with 6, 7, and 8 atoms, respectively. You should be able to run them without any modifications[2]. First run

```
python initialization.py -h
```

to find out how you specify the number of atoms. Make sure that the three different runs do not overwrite each other (running them in separate folders is probably a good idea). The first script (`initialization.py`) should take no more than than a minute or so to run, and can hence be run locally[3] if you have installed `ASE`, or even on the login node of `hebbe`. The second script (`ga.py`) typically takes a few hours and should *not* be run on the login node. If you have installed `gpaw` you can run it locally, but you are encouraged to instead submit a job to `hebbe`. On Canvas you will find a page on how to work with `hebbe`. Please be aware that the script is not adapted to be run in parallel[4], so make sure that you only use one core for these jobs! Otherwise it will probably crash.

You don't need to write anything about this task in your report.

Task 2: What does the genetic algorithm do? (3 points)

Have a look at the scripts you just ran. What do they do? Give a succinct explanation (not more than half a page) of what happens in the scripts; describe briefly what a genetic algorithm is and what it entails in our specific case. Focus on aspects related to the genetic algorithm. Do all parts of the algorithm make sense in our optimization problem? You should use the documentation of `ASE`'s `GA` module to understand the code.

Task 3: Expected outcome (no points, but prepares for the remaining tasks)

As you already know, simulations can easily yield bogus results if we are not careful. While you wait for your jobs to finish, it can be a good idea to think of what outcome you expect – how should the atoms be arranged such that the energy is minimized? Use your chemical intuition and make a

[2] In some rare cases, it may happen that a run crashes more or less inexplicably. Just run it again if that happens.

[3] If you run it locally, you may want to use a slightly older version of `ASE` than the most recent one, since the `ASE` version installed on `hebbe` is not compatible with the databases produced by the most recent version of `ASE`.

[4] It would probably be possible to adapt the script such that it runs in parallel. You may earn a gold star and an honorary mention if you do this.

guess! Build your guess in `ASE`, either by specifying coordinates in a script or by using the `ase gui`. (If you feel that your chemical intuition is imperfect, you are excused if you place the atoms more or less by random.) Write a script that uses `GPAW` to relax your custom-made structure, calculate the energy, and save the wavefunction in a `.gpw` file. To achieve this, you may use the GPAW manual, and you can also take inspiration from what happens inside the `ga.py` script you used previously. You are encouraged to play around with the different settings. Ensure that a single run takes no more than 1 core-hour (there is a quite strict limit on the computational resources available for the course, and more expensive tasks are coming...). It can be a good idea to structure this script well (it will probably not be very long), because you will have use for it shortly.

You don't need to write anything about this task in your report.

Task 4: Relaxation (1 point)

In the previous task you ran a script that took an initial set of coordinates and relaxed them until the energy reached a minimum. Why is this an insufficient method to search for the ground state (i.e., why do we need to bother about the GA)? Answer in one sentence.

Task 5: Collect the results (2 points)

When your GA simulations have finished, collect the most stable (lowest energy) structures found in each case. How can they be extracted? All candidates are stored in an SQLite database created by `ASE`. You should be able to view the contents from the command line, for example like so:

```
ase db gadb.db -c++ -L 0 -s=energy
```

The script also saves a trajectory with all investigated structures sorted by energy. You can watch it from the command line with

```
ase gui all_candidates.traj
```

Include an image of the lowest energy structures for Na_6, Na_7, and Na_8 in your report.

It will also be convenient for the following tasks if you save the structures as `xyz` files. To accomplish this, you will have to read the database from a python script. The below script gives some hints, but you may want to have a look in the documentation of `ASE` databases such that you ensure that you extract the actual ground state from the database.

```
from ase.db import connect
db = connect('gadb.db')
atoms = db.get('id=1').toatoms()
```

How do you save the thus obtained `ASE Atoms` object to file? Check out `ase.io.write`!

Task 6: Flat versus bulky (1 points)

If everything went according to plan, you should have found that the most stable Na_6 cluster is flat or almost flat, whereas Na_7 and Na_8 are more three-dimensional. It is a general feature of atomic clusters that below some size, they are essentially two-dimensional. In the lecture, on the other hand, you saw clusters optimized with a Lennard-Jones potential that were decidedly three-dimensional at very small sizes. Why is there such a "transition" to two-dimensional clusters and

why does it show up with DFT but not with a Lennard-Jones potential? Discuss! (Your discussion is allowed to be hand-wavy and speculative.)

Task 7: First and second most stable structure (2 points)

What is the energy difference between the lowest energy structure and the second lowest energy structure in each case (6, 7, and 8 atoms)? Note that the code probably has output the same "candidate" multiple times – here we are asking about the energy difference between the most stable structure and the second most stable structure that *differs significantly* in its atomic structure. Should we expect DFT to be able to reliably resolve this energy difference? Motivate!

Task 8: A closer look at Na_6 (4 points)

We will now look closer at the the first and second most stable Na_6 clusters you found in the first task. As you should have found in the previous task, these have very similar energies. Relax these structures again, using the script from task 3, but with different settings. (To be sure that you use the right structures, you are allowed to work with the structures in the course `git` repository, under `Na-clusters-GA-search/Na6-structures/`.) You can try any settings you like, but in particular, you should try different basis sets: plane waves, real space grids (`fd`) and possibly other LCAO bases, as well as at least another exchange-correlation functional. Also try to crank up one or more of the parameters. Does it make a difference? Is it always the same structure that is the most stable or does it depend on your parameters? Compile a table with the energies of the two structures for the different parameters you tried. For maximum points, this table should have at least six rows, one row for each set of parameters that you tried (i.e., one row per calculation). Briefly describe the difference between the calculations in words (for example, what are the basis functions and what is expanded in them?). Include the script you used (if you use different scripts for different sets of parameters, it suffices to include just one of them) for these calculations.

Task 9: Which calculation is the most accurate? (4 points)

In your calculations in the previous task, you should have got at least quantitatively different results with the different sets of parameters. Which calculation do you think is the most reliable and why? What could be changed to make an even more accurate DFT calculation? Are there methods other than DFT that would give more reliable results? Motivate!

Task 10: Looking at the wave functions (3 points)

Na_8 is sometimes referred to as a magic cluster, since it is very stable and thus often particularly abundant in an experimental ensemble of many clusters with different sizes. The explanation can be sought in the electronic structure. To understand this, do a new `GPAW` calculation with the most stable clusters (6, 7, and 8 atoms) and save the wavefunctions in cube files such that they can be visualized. Please include this script in your report. You are allowed to use the parameters you prefer for these calculations but note that the files may become very large for some sets of parameters; quite basic parameters should be sufficient for this task. To find out how to save cube

files, use your personal favorite search engine and type something like `plotting wave functions gpaw`. Use a software that can visualize cube files in 3D (we recommend VESTA although it is a little buggy on Linux) and plot *only the occupied* bands. (Which ones are occupied? Count the electrons or look in the log file from `gpaw`!) In your report, include all such wave function plots for Na_7 and Na_8. Why is Na_8 a magic cluster and Na_7 not? *Hint:* Do the wave functions remind you of something?