

## EXERCISE 3

# PSEUDOPOTENTIALS

### 1. Introduction

Please refer to the user manual for the ATOM program for details on how to run the program and how to make sense of the output. For these exercises we have created the aliases `ae`, `pg`, and `pt` to perform all-electron, pseudopotential generation, and pseudopotential tests, respectively, and the alias `gp` to stand for `gnuplot -persist`. The alias `energies`, when used in the work directory, will show the inter-configuration energy changes (it is equivalent to `grep '&d' OUT`). The aliases `eigenvalues_X`, with `X` being `s`, `p`, `d`, will display the appropriate eigenvalues.

There are two directories:

- `All_electron` contains some exercises to illustrate some general issues involved in computing the electronic structure of the atom. Please see the file `Example.guide` and run the examples by typing, for example `ae si.ae.inp`. Those exercises are not essential to follow the examples of pseudopotential generation, but they might help to broaden your understanding.
- `PS_Generation` holds the exercises for pseudopotential generation and test. The material for each exercise is contained in a directory named after the element. Typically there are several `XX.YYY.inp` files, where `XX` is the element's symbol, and `YYY` is some identifier. Input files for tests have the word `test` somewhere in `YYY`.

### 2. Pseudopotential exercises

#### • Basic Example: Si

An easy example to start. Generate a pseudopotential by running `pg Si.tm2.inp` and analyze and plot the results. Then test the resulting pseudopotential by running `pt Si.test.inp Si.tm2.vps`.

#### • A hard element: C

C is a first-row element, and the  $2p$  state does not have nodes, as there are no other  $p$  states below it. Thus the pseudization cannot soften the wavefunction by a whole lot, and the pseudopotential can be quite hard. Here we explore two schemes for pseudopotential generation: Hammann-Schlüter-Chiang (code `hsc`), and Troullier-Martins (code `tm2`). Note how the `rc`'s can be significantly larger for `tm2` while maintaining the transferability. Check the "softness" or "hardness" of the resulting pseudopotentials by looking at their Fourier transform.

- **Core Corrections: Na**

There are two pseudopotential input files: One for a "normal" case without core corrections, and another with corrections. Note how the transferability of the pseudopotential improves with the use of non-local core corrections.

- **Core Corrections for Spin: Fe**

For calculations involving magnetic effects, it is essential to be able to reproduce correctly the energy changes associated to changes in spin configurations. Generate the pseudopotentials `Fe.pol.inp` (polarized ground-state configuration, core corrections), and `Fe.pol.nocc.inp` (polarized ground-state configuration, no core corrections), and check their relative transferability with `Fe.test.inp`. Note how the pseudopotentials created without core-corrections exhibit unsightly wiggles near the origin. This is an artifact of the un-screening procedure. You can try to change the pseudo-core radius (set to 0.7 in the examples) and monitor the shape of the pseudocore charge using the `charge.gplot` script.

You can also generate a pseudopotential based on the unpolarized ground state (`Fe.inp`). Test it for transferability (in a real solid-state calculation one only uses the average of the spin-up and spin-down pseudopotentials, anyway).

The above examples used the GGA. It has been shown that the LDA predicts the wrong ground-state for bulk Fe!

- **Core or valence?: Cu**

The  $d$  electrons in Cu (and in Ga, and others) can be treated either as "core" or "valence" (and actually as "core but corrected"). First generate and test the "3d in valence" pseudopotential found here (`Cu.3dtm2.inp`). (You will have to prepare an input file for the test.) Then prepare an input file for a "3d in core" pseudopotential. Generate the pseudo and test it. Finally, put core corrections to the pseudopotential of the "3d in core" case.

- **Semicore states: Ba**

A somewhat technical example involving semicore states. Both the  $5s$  and  $5p$  states, which are normally thought of as "core states", are put in the valence. As the program can only deal with one pseudized state per angular momentum channel, this implies the elimination of the "genuinely valence"  $6s$  state from the calculation (and also the  $6p$ , not occupied in the atom but involved in scattering of solid-state electrons). The resulting non-local pseudopotential does not "know" anything about the  $6s$  and  $6p$  states, so the test for the "ground state configuration", which involves those, fails. As explained in the lecture, SIESTA can make up for these shortcomings by constructing extra Kleinman-Bylander projectors to deal with the missing states.