8.3 Pseudopotential generation

The generation of pseudopotentials is not difficult, but generating good pseudopotentials for some atoms may require a considerable amount of time and patience. In the following we will deal with a rather simple case.

Step 1: All-electron calculation See the example above for Si. The ground-state configuration $3s^2 3p^2$ is chosen.

Step 2: Logarithmic derivatives The pseudopotential should reproduce the logarithmic derivatives (i.e. the scattering properties) of the true potential, over the range of energies of interest and at any given distance r_d from the nucleus, except the core region. A typical value for r_d may be at midpoint of a typical bonding. A typical energy range could be a few Ry above and below the one-electron valence energies (Kohn-Sham eigenvalues).

We need to specify variables `rlderiv` (r_d), the number `nld` of logarithmic derivatives to be calculated, and the energy range, delimited by `eminld` and `emaxld`, on a grid with `deld` spacing (energies in Ry). Run `ld1.x` to calculate s, p, d logarithmic derivatives (`nld=3`): these are written to file `ld1.dlog`. The file contains energies in the first column, s, p, d,...logarithmic derivatives in the following columns. Inspect them and don't be scared by the wild oscillations: it is just the wavefunction going through 0. For a better plot in file `gnuplot.ps`, you can use the following script `plotld.gnu`:

```
set encoding iso_8859_15
set terminal postscript enhanced solid color "Helvetica" 20
set output "gnuplot.ps"
set key off
set xrange [-3:2] ; set yrange [-3:3]
set xlabel "Energy"; set ylabel "Log. der."
set label "s" at -1.0,0.1
set label "p" at -1.0,0.75
set label "d" at -1.0,1.25
set arrow from -0.797,-1.0 to -0.797,-0.3 lw 2
set arrow from -0.307,-0.6 to -0.307,0.0 lw 2 lt 2
plot "ld1.dlog" u 1:2 w l lw 2
replot "ld1.dlog" u 1:3 w l lw 2
replot "ld1.dlog" u 1:4 w l lw 2
```

The arrows indicate the energies of the bound states. Note that the $3d$ state is not bound in Si with LDA!

Step 3: Pseudopotential generation Let us now generate a pseudopotential for Si, starting from the same reference configuration as above. We need to specify `iswitch=3` and to add a new namelist, `&inputp`, where one has to provide the channel chosen as local potential (the $3d$ state: `lloc=2`), the type of pseudopotential (one projector per channel, `pseudotype=1`), the number of valence electrons (`zval=4`), the output pseudopotential file. The last lines contain a list of states to be pseudized, with corresponding pseudization

energies and matching radii (in this case, $r_c = 2.4$ for s, p, d). The local channel must be the last! The Rabe-Rappe-Kaxiras-Joannopoulos pseudization technique is used. Sample data file:

```
 &input
        atom='Si',
        iswitch=3,
        rlderiv=2.1,
        eminld=-11.0,
        emaxld=2.0,
        deld=0.01d0,
        nld=3,
        config='[Ne] 3s2 3p2 3d-1'
        dft='PZ'
 /
 &inputp
        lloc=2,
        pseudotype=1,
        file_pseudopw='Si.rrkj.UPF',
        zval=4.0,
 /
3
3S  1  0  2.00  0.00  2.40  2.40
3P  2  1  2.00  0.00  2.40  2.40
3D  3  2  0.00 -0.10  2.40  2.40
```

Note the special treatment of the $3d$ state: since it is not bound on the ground state, it is flagged by a (bogus) negative value of the charge in the all-electron configuration, and it is pseudized at energy -0.10Ry, chosen more or less arbitrarily in the typical energy range for valence electrons; $3s$ and $3p$ are instead pseudized at the Kohn-Sham energy of the bound state (this is the meaning of "0.00" as pseudization energy).

Running the code with the above input data will write a pseudopotential to file `Si.rrkj.UPF` (note that the suffix `.UPF` implies that the pseudopotential is written in the separable, nonlocal form even if it is generated in the semilocal form). Check that the one-electron levels are the same for all-electron and pseudopotential calculations. Compare the all-electron and pseudo orbitals: `ld1.wfc` and `ld1ps.wfc`, respectively. For a nice plot, you can generalize the script `plotwfc.gnu` above:

```
set encoding iso_8859_15
set terminal postscript enhanced solid color "Helvetica" 20
set output "gnuplot.ps"
set key off
set xrange [0:5]
set label "Si LDA" at 4.0,2.5
set xlabel "a.u."
set ylabel "{/Symbol y}_{nl} (r),    {/Symbol f}_l (r)"
set label "3s" at 2.7,0.6
set label "3p" at 2.7,-0.7
```

```
set arrow from 2.4,0.4 to 2.4,0.6 lw 2
set arrow from 2.4,-0.4 to 2.4,-0.6 lw 2
plot "ld1.wfc" u 1:3 w l lw 2
replot "ld1.wfc" u 1:4 w l lw 2
replot "ld1ps.wfc" u 1:(-$2) w l lw 2
replot "ld1ps.wfc" u 1:3 w l lw 2
```

The arrows point to the matching radii. Note that the relative sign of all-electron and pseudo-orbitals is arbitrary! this is why the pseudo $3p$ state is plotted with reversed sign. Do the same for the logarithmic derivatives, in `ld1.dlog` and `ld1ps.dlog`, respectively. Important: choose an energy range that is is relevant for valence electrons. For instance, edit the script `plotld.gnu` as follows:

```
set encoding iso_8859_15
set terminal postscript enhanced solid color "Helvetica" 20
set output "gnuplot.ps"
set key off
set xrange [-3:2]
set yrange [-3:3]
set xlabel "Energy"
set ylabel "Log. der."
set label "s" at -1.0,0.1
set label "p" at -1.0,0.75
set label "d" at -1.0,1.25
set arrow from -0.797,-1.0 to -0.797,-0.3 lw 2
set arrow from -0.307,-0.6 to -0.307,0.0 lw 2 lt 2
set arrow from -0.10,0.0 to -0.10,0.6 lw 2 lt 3
plot "ld1.dlog" u 1:2 w l lw 2
replot "ld1.dlog" u 1:3 w l lw 2
replot "ld1.dlog" u 1:4 w l lw 2
replot "ld1ps.dlog" u 1:2 w l lw 2 lt 7
replot "ld1ps.dlog" u 1:3 w l lw 2 lt 7
replot "ld1ps.dlog" u 1:4 w l lw 2 lt 7
```

The chosen energy range is -3 to 2 Ry. The arrows locate the pseudization energy (where all-electron and pseudo logarithmic derivatives are equal by construction). Nice, isn't it? but this is a simple case, and we are looking at the reference configuration only.

Step 4: Pseudopotential testing As a strict minimun, one has to test for *transferability* on a few atomic configurations that differ from the reference one (the one used to produce the pseudopotential). Data for testing require to specify `iswitch=2` and to select a number (`nconf`) of (pesudo-)atomic configurations, supplied in namelist `&test`:

```
 &input
         atom='Si',
         iswitch=2,
         config='[Ne] 3s2 3p2 3d-1'
         dft='PZ'
```

```
/
&test
       file_pseudo='Si.rrkj.UPF',
       nconf=5,
       configts(1)='3s2 3p2',
       configts(2)='3s2 3p1.5 3d0',
       configts(3)='3s1 3p3',
       configts(4)='3s1 3p2',
       configts(5)='3s2 3p1 3d0'
/
```

Note that fractionary occupation numbers are allowed: mathematically, it is a well defined problem; physically, you may think that you are mimicking an atom in a molecule or a solid.

Running the code with the above input data produces a summary of the results in file `ld1.test`. All-electron and pseudo-energy differences are referred to the first specified configuration, in this case the ground state. Note that *absolute* pseudo-energies have no physical meaning! they depend on the specific pseudopotential. Only energy *differences* are significant.

You may want to verify that the all-electron and pseudo-orbitals vary, not by much but by a visible amount, from a configuration to another. Can you do this with gnuplot?