Summer School 2007 - QMC : Pseudopotential generation with Opium (A)

This page last changed on Jul 11, 2007 by jnkim.

Generating pseudopotentials with OPIUM

Acknowlegements:

We would like to express our deep gratitude to Dr. Eric Walter of The Center for Piezoelectrics by Design first for his development work on OPIUM. He also graciously made available a development version of OPIUM which includes the WC functional and Hartree Fock and provided tutorials from previous workshops which were invaluable in preparing this laboratory exercise.

Goals

- Introduce the process of generating pseudopotentials
- Learn the basic operation of the OPIUM pseudopotential code
- Gain some intuition about how to construct a transferable PP
- Avoid some of the common pitfalls in PP generation

Introduction to pseudopotential generation

The basic steps for generating a pseudopotential are:

1. Choose the XC functional to use in the construction
2. Solve the all-electron atom with your chosen functional
3. Choose which states to *pseudize* away as part of the core and which states to keep as valence
4. Choose the *reference state* for construction the PP
5. Choose core radii for each angular momentum channel, as well as any other parameters needed for the construction
6. Construct pseudo-orbitals for each valence state
7. Invert the radial Schroedinger equation to compute the pseudopotential
8. Unscreen the pseudopotential
9. Test pseudopotential for transferability
10. If PP is good enough, go home, otherwise, adjust parameters and goto 5.

Much of the work boils down to the solution of the radial Schroedinger equation, which, in Hartree units, is given by

$$-\frac{1}{2}\frac{du_{nl}^2}{dr^2} + \left[\frac{l(l+1)}{2r^2} + V_{\text{eff}}(r)\right] u_{nl}(r) = E_{nl}\, u_{nl}(r).$$

Here, we have rewritten the wavefunction as

$$\psi(\mathbf{r}) = \frac{u(r)}{r} Y_{lm}(\theta, \phi)$$

E_{nl} , integrating the equation with a standard method such as Runge-Kutta, and iteratively refining the eigenenergy until an eigenstate is found.

Once the all-electron atom has been iterated to self-consistency, pseudo-orbitals are constructed. In order to produce an accurate pseudopotential, the orbitals should have the following properties.

- The pseudo-valence orbital should exactly match the corresponding all-electron orbital for r .
- The pseudo eigenvalue should match the all-electron one
- The *partial norm* of the pseudopotential should match that of the all-electron orbital. This is defined as

$$\text{partial norm} = \int_0^{r_c} u_{nl}(r)^2 \, dr$$

and simply reflect what fraction of the orbital charge is contained inside the core.

- The pseudo-orbitals should be nodeless.
- In general, the orbitals should be smooth and not contain "unphysical" oscillations.
  There exist quite a few procedures for constructing orbitals that satisfy these properties. OPIUM implements three of these, which will be enumerated below. Since there is quite a bit of flexibility in orbitals that satisfy these criteria, adjusting the construction parameters to maximize transferability and smoothness is a bit of an art. This lab aims to get your feet wet (and hands dirty) in this art.

Once appropriate orbitals are constructed, the Schroedinger equation can be inverted to compute the effective potential for each channel. The transferability and convergence properties of the pseudopotential are tested, and if they are satisfactory, the channel potentials and pseudo-orbitals are written to a file for use in calculations.

Introduction to OPIUM

OPIUM aims to be a full-featured one-stop-shopping software package for creating, testing, and exporting psuedopotentials in various formats. It features:

1. Scalar-relativistic and non-relativistic pseudopotential generation
2. Ability to construct Optimized (RRKJ) or Kerker pseudopotentials
3. Partial core correction of Louie, Froyen and Cohen
4. Can generate and test pseudopotentials that support semicore states
5. Ghost state checking following the method suggested by Gonze, Stumpf, and Scheffler
6. Automatic plotting of wavefunctions, potentials, and density using xmgrace
7. Implementation of the designed non-local potential approach of Ramer and Rappe
8. Beta version includes generation of Hartree-Fock pseudopotentials

OPIUM is run as a command-line utility. The basic command format is

```
opium paramfile logfile command_1 [command_2] [command_3] ...
```

The possible commands are given as

Command	Description
	Atomic solve and pseudopotential

	construction
ae	solve all-electron atom
ps	generate pseudopotential
nl	solve pseudopotential of the atom for reference state
tc	solves the all-elec and pseudo atom for test configurations
all	abbreviation for "ae ps nl tc"
	Pseudopotential output style
pwf	generate *.pwf output (for BH)
fhi	generate *.fhi output (for ABINIT)
ncpp	generate *.ncpp output (for PWSCF)
recpot	generate *.recpot output (for CASTEP)
	Miscellaneous options
rpt	generate report file
plot [ptype]	Make a plot of type ptype
v	toggle verbose output

The plotting types are

ptype	Plot description
wa	All-electron wave functions
wp	Pseudo and all-electron wave functions
vs	Screened pseudopotentials
vi	Unscreened (ionic) pseudopotentials
ke	Kinetic energy convergence
den	Core, valence, and partial core correction densities
pcc	Same as den
qp	Bessel transform of NL wavefunctions and potentials
logd	Logarithmic derivatives as a function of energy

The parameter file

The main parameter file is usually name <SymbolName>.param, e.g. Si.param. It has several sections or *key blocks*. They are

- [Atom]
- [Pseudo]

- [Optinfo]
- [XC]
- [Pcc]
- [Relativity]
- [Grid]
- [Relgrid]
- [Tol]
- [Configs]
- [KBdesign]
- [Loginfo]

Not all key blocks are necessary in every parameter file.
Help on these key blocks can be obtained by running `opium -k` and on
this page.

OPIUM example

For our first example, we will create a simple hydrogen pseudopotential. The parameter file, H.param, is given below:

```
[Atom]
H
1
100 1.00  -

[Pseudo]
1 1.80
opt

[Optinfo]
3.00  4

[Configs]
3
100 0.75 -

100 0.50 -

100 0.35 -
```

The `[Atom]` section describes the reference configuration for the all-electron atom. The first line is the atomic symbol. The next line gives the number of orbitals to compute, *k*, which is just one in this case. Following are *k* lines describing the atomic orbitals and their occupations.
The first number, refers to nlm, which, in this case, is (n=1,l=0,m=0), or the 1s orbital. The next number gives its occupation. The last number gives (optionally) a starting guess of the eigenenergy. In this case, we have told OPIUM to make its own estimate by including just a "-".
Unoccupied states can be included in the calculation giving a negative occupation (which is not a real occupation, just a signal to OPIUM), followed by a scattering energy. The scattering energy must be specified and cannot be left as just a "-". The valence orbitals to be reproduced by the pseudopotential should be given last.

The `[Pseudo]` section gives the basic description of how the construct the pseudopotential.
On the first line, we give the number of pseudopotential channels and their cutoff radii. In this example, this is a single channel with cutoff radius 1.8. The second line described the construction technique for the pseudopotential, which is this case is the *optimized* method.

The [Optinfo] key block gives the parameters for this construction. The first parameter, 3.00, is the cutoff wavevector, q c . The second parameter is the number of Bessel functions used to construct the optimized pseudopotential. There is only one line present here because we are creating a single-channel pseudopotential. One such line is required for each of the channel described in the [Pseudo] section.

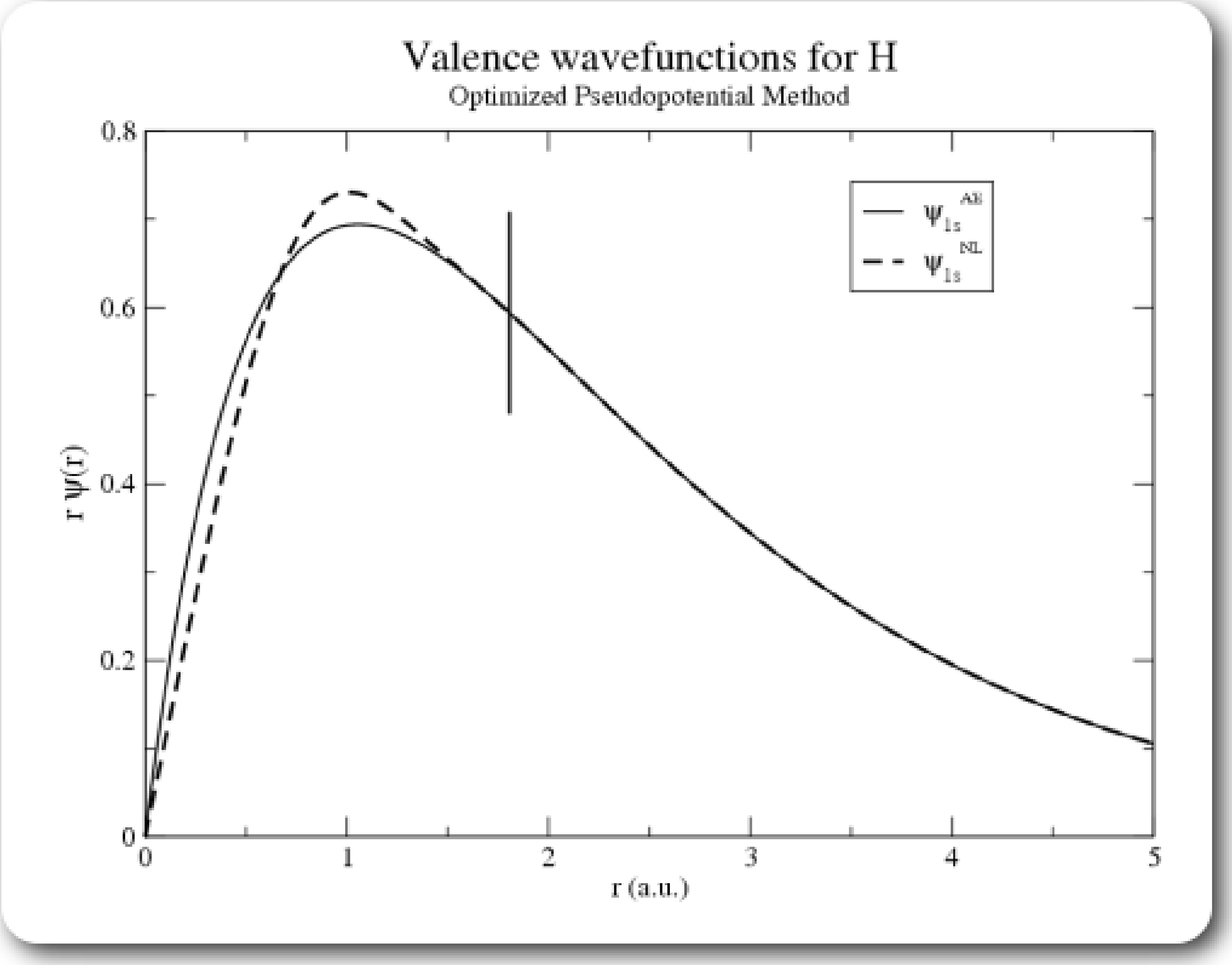
The [Configs] key block lists a set of valence configurations used to test the transferability of the pseudopotential. The first line gives the number of test configurations. The remaining lines give the test configurations. Here, we will test the pseudopotential for increasing level of partial ionization. The first configuration has lost a quarter of its electronic charge. The eigenenergies from the pseudopotential are compared to those of the all-electron atom. Using several configurations helps us to test how well the pseudopotential will perform in differing chemical environments.

Running the example

To run the example, save the input file as H.param and we type

```
opium H.param H.log ae ps nl plot wp
```

This should produce the plot below



no query string . This should always be true by construction at the reference configuration.

If we change the occupations of the orbitals in both the all-electron and pseudo-atom, the matching may no longer be perfect. If the pseudopotentials is good, however, there will still be a near-perfect correspondence.

Mini Project #1: Silicon with the WC functional

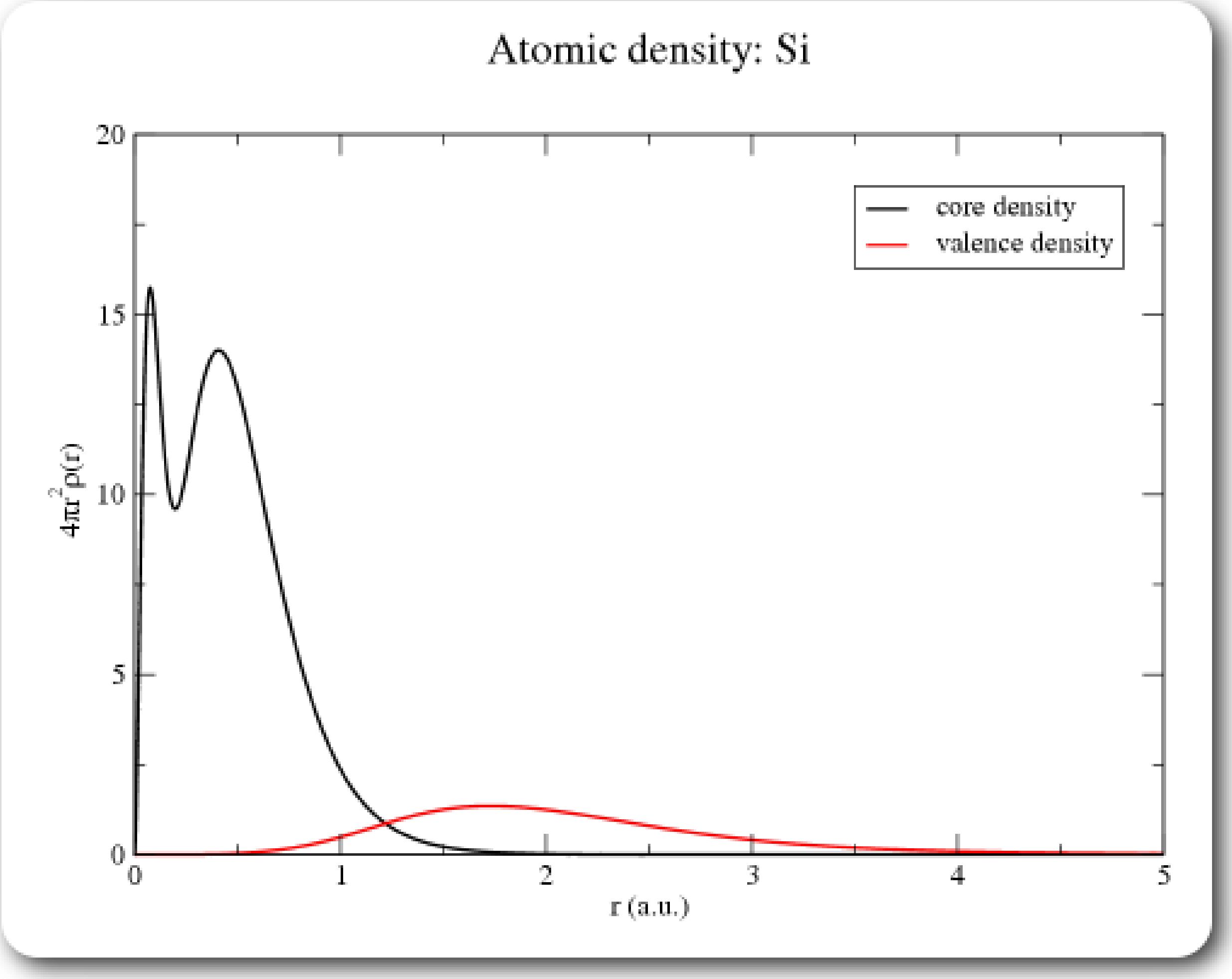
In this exercise, you will attempt to construct a high-quality pseudopotential for silicon using the Wu-Cohen exchange-correlation functional. We will attempt to construct PPs similar to the ones you used on Monday in the DFT laboratory. There is a good set of notes on the steps involved in generating pseudopotentials by Paolo Giannozzi here. We will roughly follow the steps he presents in these notes.

1. Choose a density functional

The functional you choose should *usually* match the one that you will use in your final calculation. An exception to this rule may occur when using a mean-field method to generate a trial wave function for a quantum Monte Carlo calculation. Since we are generating the pseudopotentials for this past Monday's lab, we will use the Wu and Cohen functional.

2. Choose the partitioning between valence and core states

The first step in the actual construction of the pseudopotential is to decide which states to count among the "core" states to remove from the final calculation, and which states to retain in the "valence" states, which will be included. This depends strongly on the element you are studying and the degree of accuracy required. In general, the elements toward the right-hand side of the periodic table usually require only the *true* valence states to be included. For silicon, that is the 3s and the 3p electrons. With this partitioning, there is a pretty good separation between the valence and core electron densities, as shown in the plot below.



While we could, in principle, include the 2s and 2p states in as pseudo-valence states, doing so would make our calculations *much* more expensive, and would probably gain us relatively little in accuracy.

3. The reference configuration

The *reference configuration* defines the occupation of the valence orbitals we use to construct the pseudopotential. In general, it may be helpful to choose the occupations such that they approximate the chemical environment of the atoms in the intended system, which is, in this case, SiO2. What do you think would be an appropriate valence configuration? Note that the reference state does not have to be neutral, and orbital occupations can be fractional. While this is not physically possible for an isolated atom, it is very well defined mathematically.

Once you have decided, construct an initial Si.param file in the form

```
[Atom]
Si
6
100 2.00  -
200 2.00  -
210 6.00  -
300 ?.??  -
310 ?.??  -
320 ?.??  -

[XC]
```

```
wcgga

[Pseudo]
3 ?.?? ?.?? ?.??
tm
```

You will have to fill in all the ?'s. The first set reflect the occupations for the reference state. The second set under the `[Pseudo]` heading represent the three core radii. The `tm` tells OPIUM to use the Troullier-Martins method for constructing the pseudopotential. We will use an optimized method later. You need to fill in some values for the core radii in the `[Pseudo]` keyblock to be able to run OPIUM, but the values you choose are not important for this step. They are typically between 1.0 and 2.0 a.u. You will choose more reasonable values after examining the plots of the all-electron orbitals.

Note: In the neutral-atom ground-state of silicon, i.e. (1s2 2s2 2p6 2s2 3p2), the d channel is not bound. OPIUM is capable of handling this situation using the generalized state method of Hamann (1989). This is specified by a negative occupation followed by a reference energy, which must be specified and is usually chosen to be somewhat near the highest occupied orbital energy. For example, below we have chosen -0.1 Ry.

```
320 -1.0 -0.1
```

As an alternative, if you specify a reference configuration of a positive ion, the d channel will be bound, and you can specify a small (~0.1) occupation for that state.

4. Generate the all-electron wave functions for the atom

This is done by running OPIUM with the ae command.

```
opium Si.param Si.log ae plot wa
```

This will plot the all-electron wave functions for the valence states.

5. Choose the core radii, r_c for each channel

A core radius or *matching* radius must be selected for each l-channel to be included in the pseudopotential. Beyond this radius, the unscreened pseudopotential will exactly equal the bare Coulomb potential of the ion. If the pseudopotential has been properly constructed, the pseudo orbital will also match the corresponding all-electron orbital beyond this point.

Choosing the value for r_c involves some care. Firstly, it *must* be greater than the location of the outermost node of the all-electron wave function. For a calculation in a plane-wave basis, the smaller r_c , the harder the pseudopotential will be. Thus more plane-waves will be required to reach convergence in the final calculation. In general, however, a smaller core radius will also lead to better transferability. Thus, as always, there is a trade-off between accuracy and computational expense. Typically, r_c is chosen to be a little inside or outside the outer peak in the orbital charge density. Let us begin by choosing the r_c for each channel to be at the outermost peak of the all-electron orbital. The peak locations can be read off the graph or can be read from the Si.log file. As we will seen in step 8, it will be necessary to adjust these values to obtain a transferable pseudopotential.

If the d channel is unbound, it won't be plotted. Just choose a core radius the same or perhaps a bit larger than the that of the p channel.

6. Generate the pseudopotential

Opium is told to do this with the command

```
opium Si.param Si.log ae ps nl rpt plot wp
```

This also plots the valence pseudo and all-electron wave functions. In Si.param file, we had specified `tm` in the `[Pseudo]` section, instructing OPIUM to use the Troullier-Martins pseudopotential construction method. It can also generate pseudopotentials using the Kerker method (k) or the optimized method (opt).

The optimized method can potentially construct softer pseudopotentials than the Troullier-Martins method. In order to use it, we change the `tm` to `opt` in the input file, and then we must add the `[Optinfo]` keyblock to the Si.param file as shown below.

```
.
.
.
[Optinfo]
5.0 8
5.0 8
5.0 8
```

The `[Optinfo]` section contains one line for each angular momentum channel in the pseudopotential. The first value on the line gives the target cutoff wave vector. Opium attempts to make an orbital with a target energy cutoff of q_c^2 . We will discuss this cutoff in greater detail below. The second parameter is the number of bessel functions to use in the basis for optimization. This value should be at least 4 and may yield better results with 8-10. The authors of OPIUM do not recommend values larger than 10.

7. Decide whether core to include partial core corrections

The exchange-correlation energy functionals are, in general, nonlinear in the electron density. If we include only the valence charge in our final DFT calculation, we will then be making some error if there is significant overlap of the valence and core charge density. If the valence-core overlap is small, the error will be small.

In order to correct this error, we can add back the core charge density to each atom before computing the LDA or GGA potential. This is known as a *nonlinear core correction*. For heavy atoms, however, the charge density will be sharply peaked and will need a fine mesh to represent it. Thus, for computational efficiency, the applied correction is usually only a *partial core correction*, in which only some of the core charge is included in the final calculation.

If we look back to plot in Step 2, we see that there is some overlap of the valence and core charge density. While not strictly necessary, if we wish to achieve a high-accuracy pseudopotential, a partial core correction may help in this case. We can do this by adding a new keyblock to Si.param:

```
.
.
.
[Pcc]
1.0
lfc
```

This instructs OPIUM to use the partial core correction method of Louie, Froyen, and Cohen. The 1.0 instructs OPIUM to ensure that the partial core density should match the real core density for r . For r , the partial core density is a smooth function that goes to zero at the origin. The matching radius should be chosen such that regions with significant valence density see the correct core density. If the radius is chosen to be too small, however, the density may not be easibly representable on the FFT grid used in the plane-wave calculation, yielding poor results. After adding this keyblock, you can plot the partial core density with

```
opium Si.param Si.log ae ps nl tc plot den
```

Finally, we should note that partial-core corrections are strictly associated with nonlinearities in density functionals. They cannot be used in QMC calculations. Therefore, if you intend to use the pseudopotential in QMC, no core corrections should be included.

8. Check transferability

This may be the most difficult step in our process. We will first test how transferable the pseudopotential we have generated is. We will then adjust the parameters we have set thus far to attempt to maximize this transferability, recheck with each change.

A. Check eigenvalues and partial norms for various configurations

The first, perhaps most direct test of transferability is to compute the energies (eigenenergies and total energy) for the all-electron and pseudo atoms for a number of configurations. The eigenenergies can be compared direclty, while the total energies will not agree, since the all-electron calculations include the core states. The differences in total energy between the different atomic configurations should, however, be in good agreement if the pseudopotential is transferable.

To perform this check, we must first specify test configurations in the `Si.param` file. This is done in the `[Configs]` keyblock. The first line gives the number of configurations. Each configuration is specified by n lines, where n is the number of valence orbitals. Each line has the same format as in the reference configuration in the `[Atom]` keyblock , i.e. "nlm occ E_guess".

```
.
.
.
[Configs]
4
#
300 2.00  -
310 0.00  -
320 0.01  -

#
300 1.50  -
310 1.50  -
```

```
320 0.10  -

#
300 1.50  -
310 1.00  -
320 0.1   -

#
300 1.00 -
310 1.00 -
320 0.10 -
```

Here we give three test configurations with different s and p occupations. We can then run opium with the "tc" (test configurations) and "rpt" (generate report) command.

```
opium Si.param Si.log ae ps nl tc rpt
```

We then look at the `Si.rpt` file. Scroll down to the `TC report` section of the file. The test configurations we specified above are delimited by the lines of "=" signs. The part we are interested in are the eigenvalue and partial norm differences on the lines headed by `AE-NL`. These give us some indication of the transferability. As a rule of thumb, the eigenvalue differences should not be more than a couple of mRy and partial norm difference should not be more than a couple of millielectrons.

B. Check total energies for various configurations

The second, perhaps more direct, test of transferability is to compute the total energy for the all-electron and pseudo atoms for a number of configurations. The total energies will not agree, since the all-electron calculations include the core states. The differences in energy between the different atomic configurations should be in good agreement if the pseudopotential is transferable. The total energy differences are given at the end of the `.rpt` file. If the chosen test configurations are reasonable, as the ones given above, one should be able to achieve total energy errors of less than a few meV.

C. Check logarithmic derivatives

A pseudopotential is transferable if it has the same scattering properties as the all-electron atom for a wide range of scattering energies. The scattering angle can be related to the radial derivative of the logarithm of the wave function, which is usually called just the *logarithmic derivative*. OPIUM can plot the logarithmic derivatives with the `plot logd` command. In order to make these plots, however, we need to specify the desired configuration and range of energy for the plots in the Si.param file. These are specified in the `[Loginfo]` keyblock:

```
.
.
.
[Loginfo]
0
1.8 -5.0 2.0
```

Once we have added this keyblock to the Si.param file, we can run

```
opium Si.param Si.log ae ps nl plot logd
```

D. Adjust parameters and repeat

Let us try to improve the pseudopotential transferability. We have a few options for proceding. The first approach is to decrease the matching radii (aka core radii) for each channel. Try decreasing these about 20%, rerun OPIUM, and check the report file for our transferability metrics? As noted above, decreasing the core radii will also make the pseudopotential *harder*, i.e. it will require a higher plane-wave energy cutoff. We will check this in the next step.

We can also change the target reciprocal-space cutoff in the `Optinfo` section. With a smaller r_c, we will need a larger cutoff.

9. Check for plane-wave convergence

Now, we need to check to see what kind of plane-wave cutoff, E c, will be needed for our final calculations. OPIUM will check this if you include the keywork `ke` on the command-line.

```
opium Si.param Si.log ae ps nl tc ke rpt plot ke
```

The last part plot the kinetic energy for each valence orbital as a function of the plane-wave cutoff. The report file also gives a convenient summary of the required cutoff to converge the kinetic energy within three different tolerances. Since we are interested in high-accuracy, we will consider only the most stringent, i.e. the 1 meV convergence. With relatively small core radii, the required enery cutoff will be around 50 Ry = 25 Hartree. For our purposes, this is quite reasonable.

10. Check for "ghost states"

The original form for the pseudopotential operator used in DFT calculations is known as *semilocal*. In spherical coordinates, the application of the pseudopotential involves integration over the angular coodinates, but not the radial coordinate. Thus the operator is nonlocal in theta and phi, but local in r.

$$V_{\mathrm{SL}} = V_{\mathrm{local}}(r) + \sum_{l=0}^{l_{\mathrm{max}}} \sum_{m=-l}^{l} |Y_{lm}\rangle [V_l(r) - V_{\mathrm{local}}(r)] \langle Y_{lm}|$$

The application of the semilocal pseudopotential operator is expensive in plane-wave calculations. Kleinman and Bylander suggested a way to improve the efficiency of the calculation by converting the pseudopotential from semilocal form to a fully nonlocal one:

$$\hat{V}_{\mathrm{NL}} = V_{\mathrm{local}}(r) + \sum_{l=0}^{l\mathrm{max}} \sum_{m=-l}^{l} \frac{[V_l(r) - V_{\mathrm{local}}(r)]|\phi_{nl}^{\mathrm{PS}}\rangle\langle\phi_{nl}^{\mathrm{PS}}|[V_l(r) - V_{\mathrm{local}}(r)]}{\langle\phi_{nl}^{\mathrm{PS}}|[V_l(r) - V_{\mathrm{local}}(r)]|\phi_{nl}^{\mathrm{PS}}\rangle}$$

Here, the semilocal Y_{lm} projectors have been replace by fully nonlocal projectors, $/phi_{NL}$, which are simply the atomic pseudo-orbitals, and then potentials are applied on either side of the projectors.

This *separable* form has useful properties. First, if applied to the atom in its reference state, one obtains formally the same answer as will the semilocal form. Secondly, the projections now involved primarily simple dot products, allowing efficient calculation. Finally, the computational cost scales much better with system size than the semilocal form.

An unfortunate side-effect of the Kleinman-Bylander procedure is that, if care is not taken, one can have unphysical states appear in the calculation. For local potential and semilocal pseudopotentials, the Wronskian theorem guarantees that for a given l , the atomic eigenenergies increase with the number of radial nodes. For pseudopotentials in the Kleinman-Bylander form, however, there is no such guarantee, and it occasionally happens that an eigenfunction with one or more nodes has lower energy than the nodeless "ground state". These are known as "ghost states" in the literature. Gonze, Kackell, and Scheffler described the problem and showed how ghost states can be detected. Their method is implemented in OPIUM and potential ghost states are identified in the `.rpt` file.

These are lower-energy eigenfunctions of the fully-nonlocal operator that are not eigenfunctions of the semilocal operator. Fortunately, one can usually eliminate the ghost states by appropriately choosing the *local* channel (i.e. the channel potential seen by all angular momentum states with l > l max). By default, the s (l=0) channel is made local. This behavior can be overridden in the `[KBdesign]` keyblock. For example, to make the d (l=2) channel local, one would add the following to the Si.param file:

```
.
.
.
[KBdesign]
d
```

Opium includes automatic checks for ghost states. These can be seen the the Si.rpt file that is generated when `rpt` is included in the commmand line. It reports the existence of ghost states in two ways. First, it check to see if any ghost states will exist if each channel is chosen as the local channel. Secondly, it will note in which channel the ghost state will appear. For most reasonable choices for silicon, there are no ghost states.

It should be noted that if you selected a reference configuration with an unbound d channel, OPIUM cannot check this channel for ghost states. For this reason, you might choose to use a reference state with sufficient positive charge to ensure a bound d state.

11. Write to your format of choice

OPIUM currently supports 6 output formats:

keyword	writes format for program
pwf	Bh
recpot	CASTEP
fhi	ABINIT
ncpp	PWSCF
psf	SIESTA
casino	Casino

For example, to generate the pseudopotential for ABINIT,

```
opium Si.param Si.log ae ps nl tc rpt fhi
```

Mini Project #2: Oxygen with the WC functional

The construction of a psuedopotential for oxygen will largely follow the same steps we used for silicon. Here we will highlight some of the salient features which are different. Let us start with a simple parameter file for oxygen:

```
[Atom]
O
4
100 2.00  -
200 2.00  -
210 4.00  -
320 -1.0  -0.1

[XC]
wcgga

[Pseudo]
3 ?  ?  ?
opt

[Optinfo]
15.0  10
15.0  10
15.0  10
```

Note that the d orbital is unbound for the ground-state configuration, so we must use the Hamann generalized state method, indicated by the negative occupation.
Pick reasonable values for the core radii, and run OPIUM with

```
opium O.param O.log ae
```

Check the log file for the location of the peaks of the wave function in the O.log file.
Let us first set the s and p core radii to correspond to the locations of these peaks. Set the d-channel radius to be the same as the p for now.

Now, define 3 reasonable configurations for testing in the `Configs` keyblock as in silicon. Remember that oxygen will tend to accept electrons from silicon.
ℹ Configurations with too many electrons will have also have an unbound p channel. This effectively limits the oxidation state of the test configurations to about $O^{0.75-}$ or so.

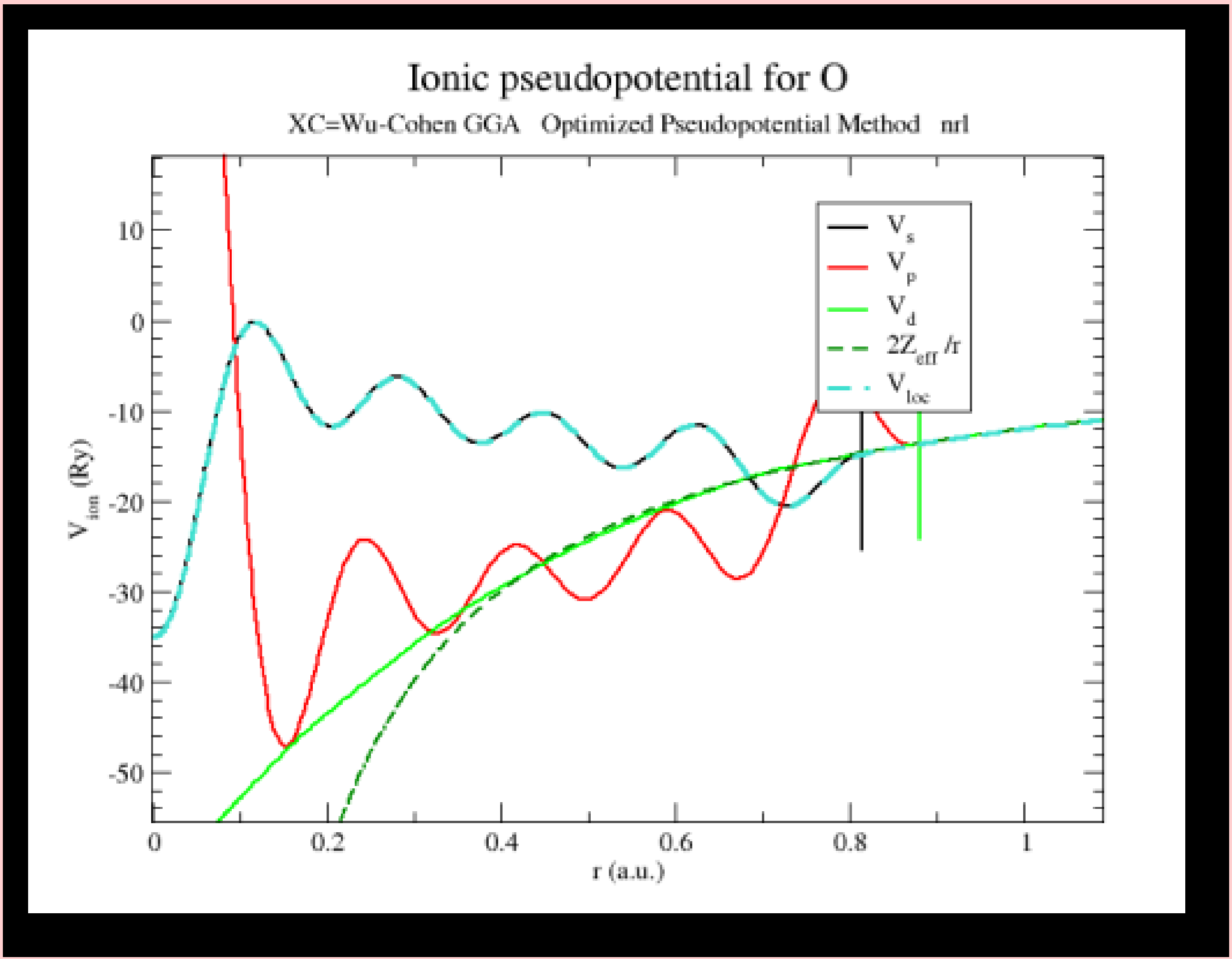
Let us run OPIUM, running the major tests:

```
opium O.param O.log ae ps nl tc ke rpt plot vi
```

This will also plot the unscreened pseudopotential.

Potential pitfalls

It is a good idea to occasionally plot the unscreened pseudopotential while fiddling with the parameters. If you you get a potential with many oscillations, it is probably because you chose the target wave-vector, q_c too small in the `[Optinfo]` keyblock. Increasing this value usually removes this unphysical behavior. An example of a pseudopotential with this problem is shown below.



```
...
[Pcc]
0.5
lfc
```

Scroll down to the `KE report` section of O.rpt and check the Ecut necessary for 1 meV convergence. It's nearly 200 Ry! This is pretty hard for a GGA pseudopotential. Fortunately, we reduce this significantly by pushing out the core radii. On the other hand, our transferability is quite good. Let us see if we can decrease the plane-wave energy cutoff required for convergence a little without compromising transferability too much. During this process, we should keep in mind that for high-pressure applications, we may need to use a harder pseudopotential than we would for typical zero-pressure studies. In particular, under high compression we must make sure that our pseudocores to not overlap. Also, if we intend to use the potentials in quantum Monte Carlo calculations, we are usually interested in very high accuracy, and hence we don't want to skimp on our energy cutoff and invalidate our entire simulation.

Adjust the core radii of the three channels. How far can you push them out before the eigenvalue errors exceed about 0.5 mRy? You may be able to achieve better potentials by also adjusting the q_c for each channel.

When you have found a satisfactory pseudopotential, use OPIUM to write it out to files for use in ABINIT, PWSCF, and Casino.

Mini Project #3: Carbon with Hartree-Fock

Current LDA and GGA density functional use approximate forms of both exchange and correlation interactions. In contrast, Hartree-Fock provides the exact exchange interaction, but no correlation. While the jury is still out, there is reason to believe that having the correct exchange behavior in pseudopotential used in QMC is important. For this reason, many people believe that it is a good idea to use HF pseudopotentials in QMC simulations.

Eric Walter has graciously provided us with an unreleased, bleeding-edge copy of OPIUM which can generate HF pseudopotentials. In this last exercise, we will generate one for carbon. A HF pseudopotential can be created by using the keyword `hf` in the `[XC]` keyblock. With standard DFT functionals, the exchange-correlation potential is itself local. This allows us to naturally produce a local ionic potential outside the core. The Hartree-Fock potential, however, is inherently nonlocal and requires a special localization and smoothing procedure. OPIUM can do this with a number of different methods, specified in the `[hfsmooth]` keyblock. The remaining sections are the same as before. Below is a skeleton parameter file using the neutral atom as the reference state.

```
[Atom]
C
3
100 2.00  -
200 2.00  -
210 2.00  -

[XC]
hf         # turns on hf

[hfsmooth]
1           # this chooses how to do the HF psp smoothing
            # as in the Trail-Needs procedure
            # use 2 or 3 (either is fine) for an optimtized potential,
            # use 1 for TM.
            # To turn off the smoothing use: 0, but then the psp can
            # only be used within OPIUM, not in target calcs.

[Pseudo]
2 ? ?
tm

[KBdesign]
?

[Configs]
3
#
200 2.00  -
210 1.75  -
#
200 2.00  -
210 2.25  -
#
200 2.00  -
210 2.50  -
```

Note that we are using the `tm` (Troullier-Martins) construction, rather than the optimized construction, for

simplicity. Hence, the `[Optinfo]` keyblock is not needed.
Fill in the ?'s in the parameter file and run

```
opium C.param C.log ae ps sl ke tc rpt plot vi
```

Note that the `nl` (nonlocal) command has been replaced in this case with `sl`, since OPIUM does not currently generate the Kleinman-Bylander form for Hartree-Fock pseudopotentials.

Adjust the core radii as you have in the previous exercises, and see how the transferability changes. Artifacts of the localization and smoothing process may make it difficult to eliminate eigenenergy errors. Try also using the "optimized" method instead of Troullier-Martins. This will require the `[Optinfo]` section as before, and switching the `[hfsmooth]` type, as described in the example input file.

Again, when you are satisfied with your work (or run out of time), write the pseudopotential to files for ABINIT, PWSCF, and Casino.

Suggestions for Further Study

1. Designed Nonlocal Pseudopotentials

Ramer and Rappe introduced a method to improve the transferability of nonlocal pseudopotentials by altering the form of the local potential. If you are interested in this method, consult reference 4 below and read the documentation on the OPIUM http://opium.sf.net.

2. Relativisitic corrections

For atoms significantly heavier than the ones included here, relativitic effects become very important for the core orbitals. While the valence states do not require a relativistic treatment, the proper treatment of the core states is necessary since the shape of the core orbitals influences the valence orbitals. OPIUM can include the dominant relativistic effects by solving the Schroedinger equation in *scalar relativistic* form. A fully relativistic solution to the atomic problem requires solution of the Dirac equation, with its spinor formalism. The scalar relativistic equations include all the relativistic effects which can be retained without the use of spinors. This form may be specified in the `[Relativity]` keyblock of the parameter file.

Suggested input files

⚠ Warning! Plot spoilers contained below.

We have included examples of reasonable input files for each of the mini-projects. We do not claim that they will produce the absolute optimal pseudopotentials (as if that existed), but they should produce a reasonable starting point. If you cannot get your own input files to work, consulting these files may help you to locate the problem.

Mini Project #1 input

Si.param

Mini Project #2 input

O.param

Mini Project #3 input

C.param

References

1. D. R. Hamann, M. Schlüter, and C. Chiang, *Norm-Conserving Pseudopotentials*, Phys. Rev. Lett. 43, 1494 (1979)
2. Gonze, Kackel, and Scheffler, *Ghost states for separable, norm-conserving ab-initio pseudopotentials*, Phys. Rev. B **41**, 12264.
3. Paolo Giannozzi, *Notes on pseudopotential generation*
4. Nicholas J. Ramer and Andrew M. Rappe, *Designed nonlocal pseudopotentials for enhanced transferability*, Phys. Rev. B **59**, 12471
5. Leonard Kleinman and D.M. Bylander, *Efficacious Form for Model Pseudopotentials*, Phys. Rev. Lett. 48, 1425
6. I. Grinberg, N. J. Ramer, and A. M. Rappe, *Quantitative criteria for transferable pseudopotentials in density functional theory*, Phys. Rev. B 63, 201102 (2001)
7. D. R. Hamann, Phys. Rev. B 40, 2980 (1989)

Pseudopotential libraries

- Casino pseudopotential library