Opium Tutorial 1: Carbon

Goal: The purpose of this tutorial is to introduce the OPIUM input file and the basic features of OPIUM.

The first step to creating a pseudopotential in OPIUM is to construct the *param* file. It is the one and only input file for OPIUM. A basic param file for carbon is shown here:

```
[Atom]
C
3
100 2.00  -
200 2.00  -
210 2.00  -
#This is a comment

[Pseudo]
2 1.4 1.4  # Comments can be at the end of lines too
kerker

[XC]
lda
```

The param file consists of a series of *keyblocks* which are identified by a keyword in brackets. All of the data as well as the keywords themselves are case-insensitive. All of the entries in the keyblocks are free-form, i.e. spaces and carriage returns are ignored. Comments are indicated by the # symbol. A complete list of keyblocks can be found via the command line by entering `opium -k`.

In `c.param`, the first two keyblocks, `[Atom]` and `[Pseudo]` are mandatory. The `[Atom]` keyblock specifies first the atomic symbol. Then the number of orbitals in the *reference configuration* is specified. The reference configuration is the atomic configuration from which the pseudopotential will be constructed. Each orbital is then defined by a line specifying first, its n and l quantum numbers in the form of "$nl0$" (m is left 0 since spherical symmetry is enforced). This line continues with the occupation for this orbital. Finally, the last entry on the line is the initial guess for the eigenvalue for this state. Putting simply "-" instructs OPIUM to make its own guess.

The `[Pseudo]` keyblock specifies the number of valence states, the cutoff radius (r_c) for each valence state and the pseudopotential construction method.

The third keyblock `[XC]` specifies the choice of exchange/correlation function used for all calculations (*lda* refers to the LDA functional of Perdew and Zunger[1]. This is the default.).

In the first step of the calculation, we solve the all-electron Hamlitonian by running the `ae` calculation:

```
%> ./opium c c.log ae
```

We can then inspect the *c.log* file to see if the calculation converged without any errors:

```
%> cat c.log
```

```
=========================================================================
 Begin AE calculation
 =========================================================================
 Performing non-relativistic AE calculation...

 iter       Etot             Ebs             Ehxc       de_max   dv_max
   1      -78.0101223     -46.1066343     -31.9034880 0.73E+00 0.20E+01
   2      -76.9630569     -44.6770656     -32.2859913 0.10E+00 0.10E+01
   3      -75.3373196     -42.8552064     -32.4821132 0.18E+00 0.14E+00
.
.
  26      -74.8485240     -42.5925070     -32.2560171 0.11E-07 0.32E-07
  27      -74.8485240     -42.5925069     -32.2560171 0.52E-08 0.18E-07

 After    27 iterations...
 Energy:     -74.84852399  Ebs:     -42.59250694  Ehxc:     -32.25601705
```

```
   Orbital    Filling       Eigenvalues    Norm(rc->oo)      Peak
    |100>      2.000         -19.895705
    |200>      2.000          -1.001950      0.542895      1.224346
    |210>      2.000          -0.398599      0.589443      1.192923

 ========================================================================
 End AE calculation
 ========================================================================
------------ closing opium session ------------
```

The first section of the all-electron part of the log file shows that this calculation was performed using the non-relativistic Hamiltonian. Next, we see the convergence of the energy and potential. We can see that the largest change in any eigenvalue (`de_max`) and potential (`dv_max`) [2] is smaller than 1e-8 Ry. Finally, the last section shows the total energy and eigenvalues for the electronic configuration. Also, for the valence states, the norm of the wavefunction beyond the cutoff radius and the position of the outermost peak is shown.

We can also plot the all-electron valence wavefunction:

```
%> ./opium c c.log plot wa
```

As we can see from the plot and the log file, the outermost peaks of the valence wavefunctions are about 1 a.u. from the nucleus. It is normal to place the pseudopotential r_c at or somewhat beyond the outermost wavefunction peak. Also, the r_c's should be less than about 45% of the smallest bondlength in the target calculation.

Now, we can construct the pseudopotential:

```
%> ./opium c c.log ps
```

Again, we inspect the *c.log* file to see if the calculation converged without any errors:

```
%> cat c.log
```

```
========================================================================
 Begin PS construction
 ========================================================================
 Kerker pseudopotential           2
 --------------------Pseudizing state: |200>   --------------------
 point nearest rc :  781
 rc               :  1.400000
 actual rc        :  1.394319

 Psi(rc)          :     0.5714324127
 Slope at rc      :    -0.5466733176
 Curvature at rc  :     0.1938596519
 Del^2 Psi at rc  :    -0.5902839015
 Log Deriv at rc  :    -0.9566718749
 --------------------Pseudizing state: |210>   --------------------
 point nearest rc :  781
 rc               :  1.400000
 actual rc        :  1.394319

 Psi(rc)          :     0.4982140812
 Slope at rc      :    -0.4365000961
 Curvature at rc  :     0.3233964634
 Del^2 Psi at rc  :    -0.8152480487
 Log Deriv at rc  :    -0.8761295848
 ------------------------------
 Descreening potential
 valence charge          :   4.000000
 core    charge          :   0.000000

 ----Solving the Schrodinger equation for all states----
```

```
 State: 2s  AE eigenvalue =  -1.001950  PS eigenvalue =  -1.001886
 State: 2p  AE eigenvalue =  -0.398599  PS eigenvalue =  -0.398626

 ---Semilocal ghost testing---
 Local state: 2s

 Test  state: 2p
 KB energy : -10.956371  KB strength:   3.571735  KB cosine:  -0.325996
 WARNING! : schsl found a positive eigenvalue. n,l=:   3   1
    !WARNING! No solution for 1st excited state of local potential.  Setting e=0.0
 el0       :  -0.089117  el1        :   0.000000  eig      :  -0.398599
 No ghosts!  Ekb<0  and eig < el0

 No ghosts for local potential: 2s
 -----------------------------

 Local state: 2p

 Test  state: 2s
 KB energy :  11.556912  KB strength:   3.382582  KB cosine:   0.292689
 el0       :  -6.284721  el1        :  -0.253512  eig      :  -1.001950
 No ghosts!  Ekb>0  and el0 < eig < el1

 No ghosts for local potential: 2p
 -----------------------------

 ======================================================================
 End PS construction
 ======================================================================
------------ closing opium session ------------
```

The first line of the log file shows that we are using the Kerker pseudopotential construction method[3]. Next, we see the input information for the each of the valence states. After the pseudopotentials are made, the valence and core charges are computed. Next, the eigenvalues are computed using the pseudopotential. We see that the all-electron and pseudo eigenvalues agree for the valence orbitals. Finally, the *semi-local* ghost testing is performed. We will discuss ghosts in a later tutorial. All that is important for now is that choosing *one* of the valence states to be the local potential can yield a ghost-free pseudopotential. In the current potential, either the $2s$ or the $2p$ state can be the local potential.

We next plot the all-electron and pseudo ionic potentials:

```
%> ./opium c c.log plot vi
```

It is important to notice that both the s and the p pseudopotential become equal to the all-electron $(2Z_{\text{eff}}/r)$ beyond the cutoff radius. Before the cutoff radius, the pseudopotentials are much shallower than the all-electron potential.

At the end of the previous step, the eigenstates of the *semi-local* pseudopotentials were found. Typically, the pseudopotential is applied in a fully *non-local* way using the approach of Kleinman and Bylander [4]. OPIUM can solve for the eigenstates using the non-local form of the pseudopotential. The eigenstates of the reference configuration should be identical in the semi-local and non-local approaches.

```
%> ./opium c c.log nl
```

Again, we inspect the *c.log* file to see if the calculation converged without any errors:

```
%> cat c.log
```

```
 ======================================================================
 Begin NL calculation
 ======================================================================
 Using the s potential as the local potential

 iter       Etot             Ebs             Ehxc       de_max   dv_max
   1      -10.6957922      -2.8010240      -7.8947682 0.68E-04 0.21E-04
```

```
   2      -10.6957810      -2.8010168      -7.8947642 0.44E-05 0.10E-04
   3      -10.6957776      -2.8010143      -7.8947633 0.15E-05 0.52E-05
.
.
.
.
   9      -10.6957763      -2.8010128      -7.8947635 0.22E-07 0.14E-06
  10      -10.6957762      -2.8010128      -7.8947635 0.24E-07 0.77E-07
  11      -10.6957762      -2.8010128      -7.8947635 0.71E-08 0.44E-07

 After    11 iterations...
 Energy:     -10.69577621  Ebs:      -2.80101276  Ehxc:      -7.89476346

   Orbital    Filling       Eigenvalues    Norm(rc->oo)      Peak
    |100>      2.000          -1.001883      0.542905      1.240366
    |210>      2.000          -0.398623      0.589434      1.147294

 ---Non-local ghost testing---
 Local state: 1s

 Test  state: 2p
 WARNING! : schsl found a positive eigenvalue. n,l=:   3   1
 !WARNING! No solution for 1st excited state of local potential.  Setting e=0.0
 KB energy : -10.956371  KB strength:   3.571735  KB cosine:  -0.325996
 el0       :  -0.089115  el1        :   0.000000  eig      :  -0.398623
 No ghosts!  Ekb<0  and eig < el0
 ------------------------------

 No ghosts present for local potential
 ======================================================================
 End NL calculation
 ======================================================================
------------- closing opium session ------------
```

We see that the non-local solution yields the all-electron valence eigenvalues and wavefunction norm beyond the cutoff radius. Furthermore, we also see that for the choice of the local potential (the default is the s-potential) the pseudopotential does not contain a ghost.

It is instructive to plot the all-electron and pseudo wavefunctions on the same graph:

```
%> ./opium c c.log plot wp
```

It is important to see that, just as in the potential plot, the pseudo-wavefunctions are identical to the all-electron wavefunctions beyond the cutoff radius. Also, note that the s pseudowavefunction is positive everywhere, whereas the all-electron one has a node. The fact that the pseudo-wavefunctions are nodeless leads to a smaller basis set requirement in the target calculation.

When using a pseudopotential in a solid-state calculation, it is necessary to determine the size of the plane-wave basis set required to give a faithful representation of the potential. This value is usually referred to as the *cutoff energy* and is the kinetic energy of the highest frequency planewave contained in the basis. The energy cutoff can simply be determined in the target calculation by examining the total energy as a function of cutoff energy. OPIUM can also compute the energy cutoff for each angular momentum channel in the pseudopotential. The maximum of these will be the effective energy cutoff. This is done in OPIUM by running the *ke* step:

```
%> ./opium c c.log ae ps nl ke
```

```
%> cat c.log
```

```
.
.
.
 ======================================================================
 Begin KE convergence testing
 ======================================================================
 ----------------------------------------------------------------------
```

```
                          KE convergence
 ============================|100>============================
   Ecut(Ry)  KE/electron(Ry)         KE error/electron (mRy,meV)
      4       0.743783540                72.763000     989.991544
     23       0.809649517                 6.897023      93.838821
     32       0.815876483                 0.670057       9.116591
     55       0.816474176                 0.072363       0.984555
   ============================|210>============================
   Ecut(Ry)  KE/electron(Ry)         KE error/electron (mRy,meV)
     27       2.293264207                72.442332     985.628641
     50       2.358641014                 7.065525      96.131420
     82       2.364981131                 0.725408       9.869682
    126       2.365634458                 0.072081       0.980719

 ======================================================================
 End KE convergence testing
 ======================================================================
```

We see in the log that the *s* potential is converged to within 100 meV by about 23 Ry and 10 meV by about 32 Ry, whereas the *p* potential does not converge to 100 meV until 50 Ry and 10 meV by 82 Ry. A reasonable convergence error is on the order of 10 meV.. In practice, some observable parameter(s) (such as a equilibrium bond length or lattice constant) should be used to determine an efficient and accurate cutoff energy.

The pseudopotential cutoff energy can be reduced by increasing the cutoff radius or by using a more sophisticated pseudopotential construction method. These topics are left for a later tutorial.

The `KE` convergence can also be plotted:

```
%> ./opium c c.log plot ke
```

The pseudopotential can be exported to any one of the output formats that OPIUM supports at this point. For example, the *fhi* format is one that is supported by the code ABINIT:

```
%> ./opium c c.log fhi
```

The file *c.fhi* will contain the pseudopotential.

Although all of the steps were done separately in this tutorial, it is a good idea to run all steps of the calculation together so all of the details are in one log file. Furthermore, a summary of the output can be generated with the *rpt* command to yield the *report* file.

```
%> ./opium c c.log ae ps nl ke rpt
```

To Explore: Selecting an appropriate cutoff radius is an important step to making an accurate and efficient pseudopotential. Use OPIUM to determine how the kinetic energy cutoff varies as a function of the pseudopotential cutoff radius. How different is this for the *s* compared to the *p* states? Is the behavior monotonic?

---

[1] J. P. Perdew and A. Zunger, Phys. Rev. B **23**, 5048 (1981).
[2] `dv_max` is the largest change at any grid point in the self-consistent potential
[3] G. P. Kerker, J.Phys.C**13**, L189 (1980).
[4] L. Kleinman and D. M. Bylander, Phys. Rev. Lett. **48**, 1425 (1982).

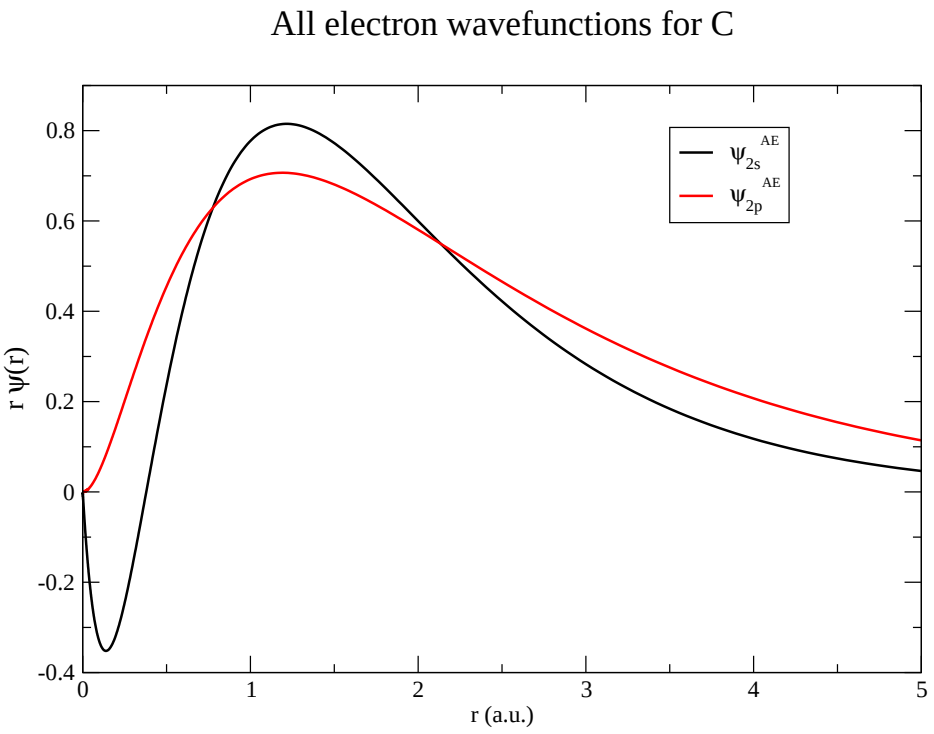

FIG. 1: Carbon all-electron wavefunctions

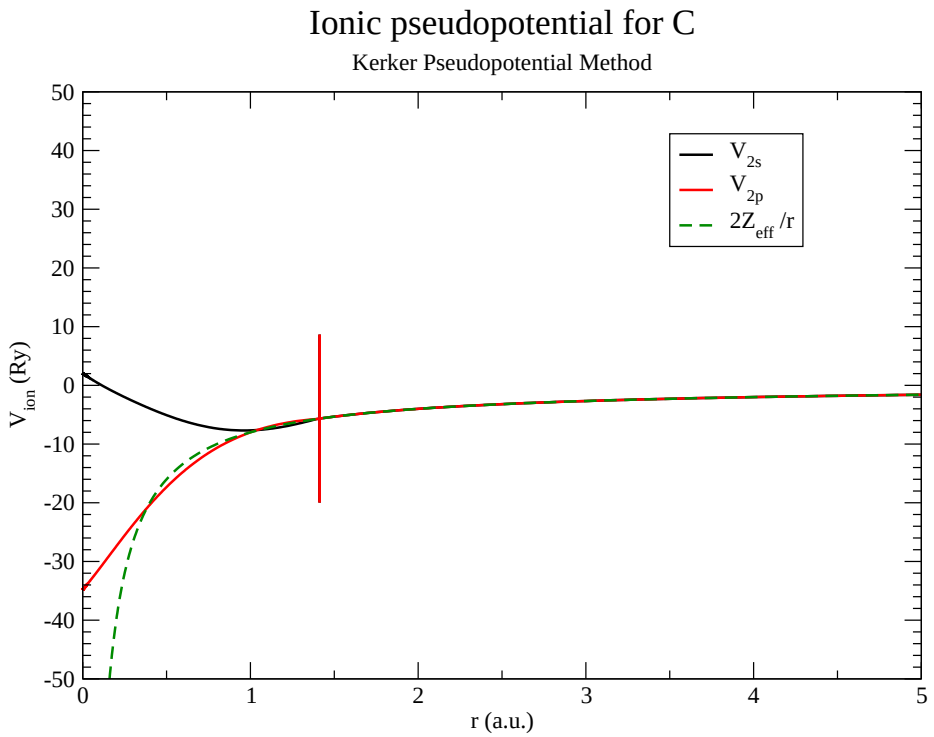

FIG. 2: Carbon ionic potentials

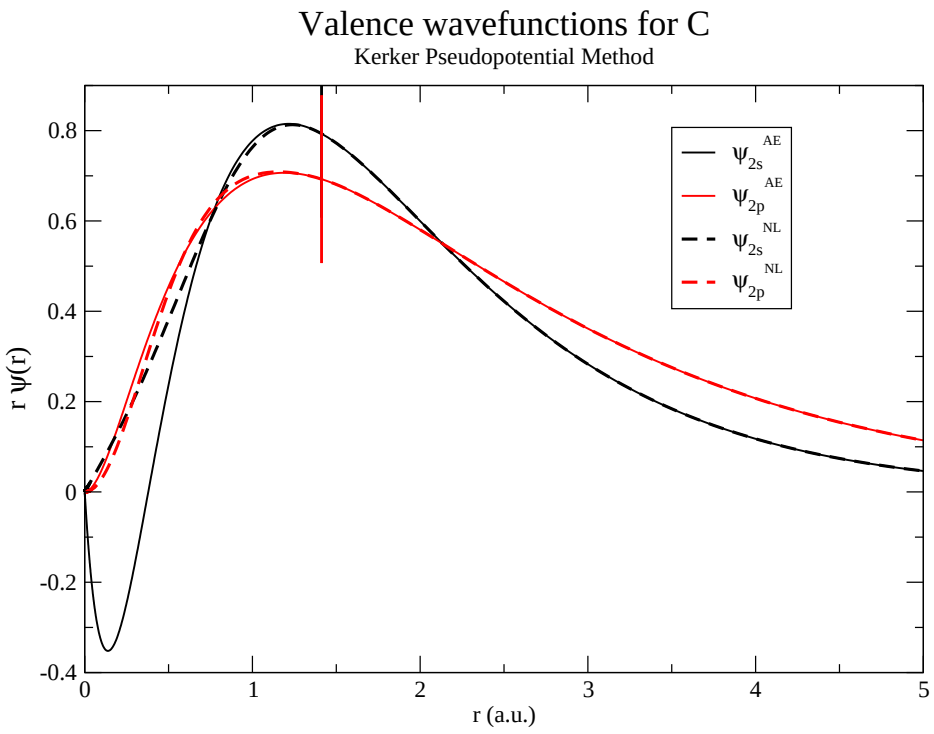

FIG. 3: Carbon all-electron and pseudo wavefunctions

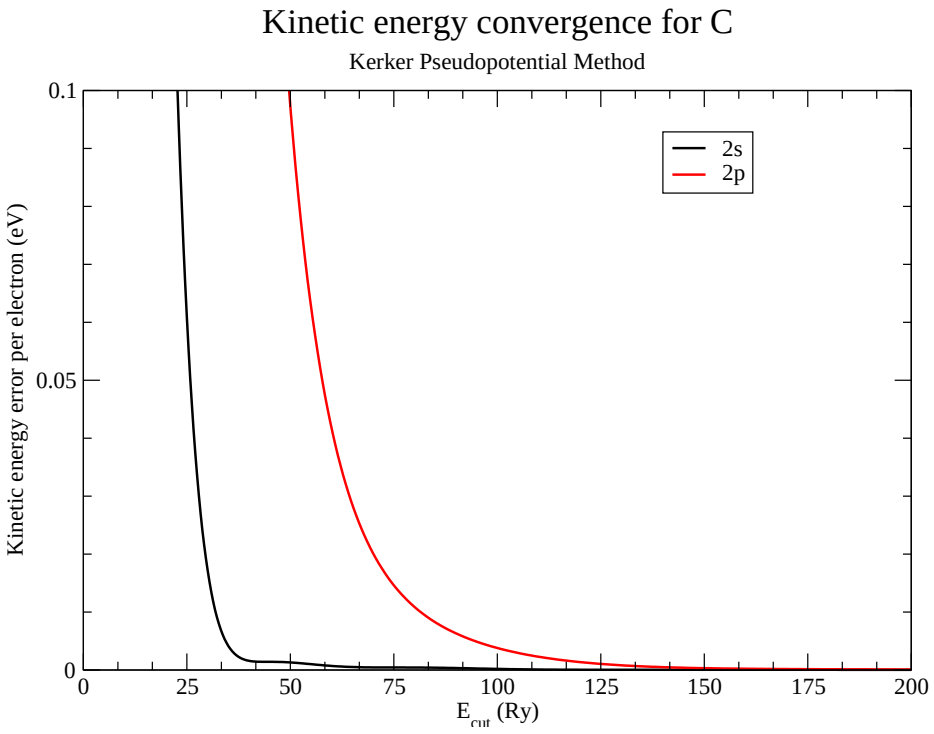

FIG. 4: Carbon pseudopotential kinetic energy convergence