

Computational Chemistry Homework 2

Gray Laughlin, Jeonghyun Ko, Yujia Wang

Due: 09-29-2017 Fri

1 Lectures 1-2: Review of quantum mechanics

An electron is trapped in a one-dimensional box described by the potential (recall 1 bohr = 0.529177 Å, is the atomic unit of length):

$$V(x) = \begin{cases} 0, & -1 < x < 1 \text{ bohr} \\ \infty, & x \leq -1 \text{ or } x \geq 1 \text{ bohr} \end{cases}$$

- (a) Using the energy expression given in class, calculate the ground state (n=1) energy of the electron, in Hartree (the atomic unit of energy), in eV, and in kJ mol⁻¹.

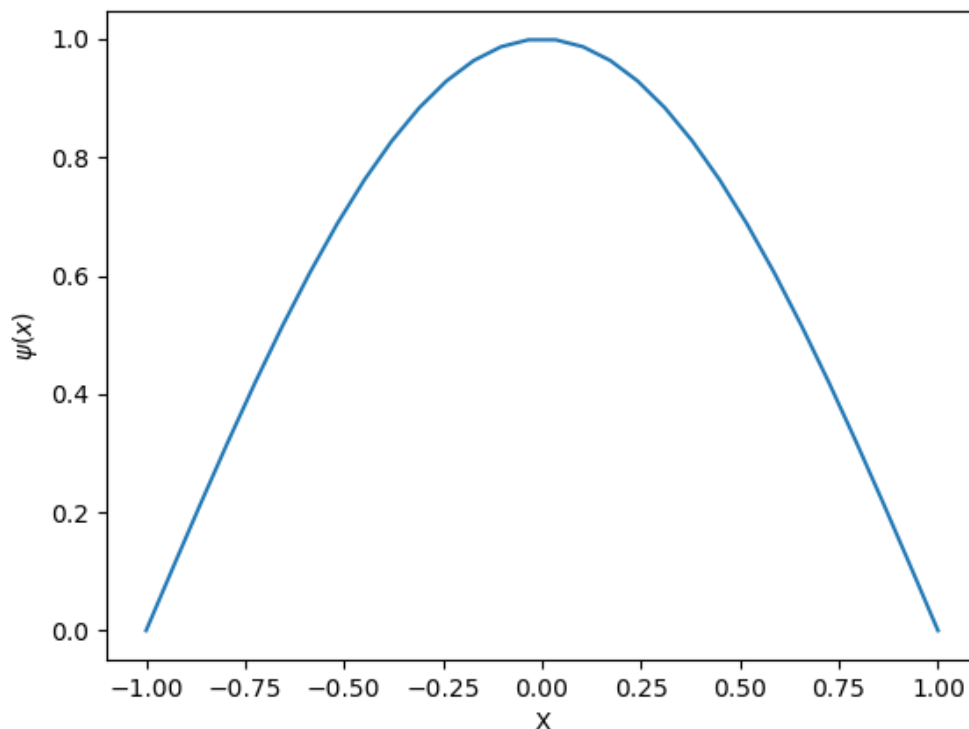
$$E = \frac{n^2 \pi^2 \hbar^2}{2m_e L^2}$$

For an electron in its ground state, n=1. In atomic units $\hbar = m_e = 1$.

$$E_{\text{ground}} = \frac{\pi^2}{(2)(2)^2} \text{ 1 hartree} = 27.212 \text{ eV} = 2625.50 \text{ kJ mol}^{-1} \quad E_{\text{ground}} = 1.2337 \text{ Hartree} = 33.5707 \text{ eV} = 3239.1 \text{ kJ mol}^{-1}$$

- (b) We found in class that the ground-state wavefunction for this electron is $\psi_1(x) = \cos(\pi x/2)$. Sketch this wavefunction and show that it obeys the proper boundary conditions, has zero nodes, and is normalized.

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.linspace(-1,1,30)
5 f = np.cos(np.pi*x/2)
6
7 plt.figure()
8 plt.plot(x,f)
9 plt.ylabel('$\psi(x)$')
10 plt.xlabel('X')
11 plt.savefig('./fig1.png')
12 plt.show()
```



The wavefunction plot shows zero nodes because it only intersects the x-axis at the bounds. The boundary conditions were that the wavefunction must go to zero when x is greater than 1 or less than -1, because the potential energy goes to infinity in this range. The wavefunction satisfies this as shown in the plot. We can test for normalization by taking an inner product over the bounds.

```

1 import numpy as np
2 from scipy.integrate import quad
3
4 # Normalizing
5 def integrand(x):
6     return np.cos(np.pi * x / 2) ** 2
7
8 ans, err = quad(integrand, -1, 1)
9 print(ans)

```

1.0

$$\langle \psi_1 | \psi_1 \rangle = \int_{-\infty}^{\infty} \cos^2\left(\frac{\pi x}{2}\right) dx = 1$$

- (c) Suppose you approximate the true ground-state wavefunction by $\phi_1(x) = 1 - x^2$. Calculate the expectation value of the energy (in Hartree) for this approximation. How does your answer compare to the energy you calculated in (a)?

The expectation value of total energy of the ground state electron is given as: $\langle E \rangle = \frac{\langle \phi | \hat{H} | \phi \rangle}{\langle \phi | \phi \rangle}$

By approximating the true wavefunction as $\phi_1(x) = 1 - x^2$ and using the bounds $x = [-1, 1]$:

$$\langle \phi | \hat{H} | \phi \rangle = \frac{8}{3} \text{ Hartree}$$

Where \hat{H} is the 1D kinetic energy operator, $-\frac{1}{2} \frac{\partial^2}{\partial x^2}$.

$$\langle \phi | \phi \rangle = \frac{16}{15} \text{ Hartree}$$

$$\langle E \rangle = \frac{5}{4} \text{ Hartree}$$

Or we normalize the wavefunction first $\phi_1(x) = 1 - x^2$: $\tilde{\phi}_1(x) = \left(\frac{1}{\langle \phi_1 | \phi_1 \rangle}\right)^{1/2} \phi_1$. The expectation value of the energy can then be calculated using the Hamiltonian operator as $\langle E \rangle = \langle \tilde{\psi} | H | \tilde{\psi} \rangle$.

```

1 import numpy as np
2 from scipy.integrate import quad
3
4 # Normalizing
5 def integrand(x):
6     return (1 - x ** 2) ** 2
7
8 c, err = quad(integrand, -1, 1)
9
10 # The wavefunction can also be written as a polynomial
11 # (-1*x^2 + 0*x + 1) / sqrt(c)
12 p = 1 / np.sqrt(c) * np.array([-1, 0, 1])
13
14 def func(x):
15     return -0.5 * 1 / np.sqrt(c) * (1 - x ** 2) * np.polyder(np.polyder(p)) # np.polyder(p) calculates the first derivative of p
16
17 E, err = quad(func, -1, 1)
18
19 print ('The expectation value of the energy is {0:.2f} hartree.'.format(E))

```

The expectation value of the energy is 1.25 hartree.

The energy is 1.3 % greater than the energy in part (a).

(d) Can you guess an even better approximate wavefunction? Guess a candidate, evaluate the expectation value of its energy, and compare to question (c). Did you do any better than my guess?

Linear combination of two functions helps to get better approximate wavefunction. The wavefunction has to obey the boundary conditions.

The expectation value of total energy of the ground state electron is given as: $\langle E \rangle = \frac{\langle \phi | \hat{H} | \phi \rangle}{\langle \phi | \phi \rangle}$

By approximating the true wavefunction with the function, $\phi_1(x) = (1 - x^2) + (1 - x^4)$, we might be able to get a more accurate energy estimation.

$$\langle \phi | \hat{H} | \phi \rangle = \frac{716}{105} \text{ Hartree}$$

$$\langle \phi | \phi \rangle = \frac{1552}{315} \text{ Hartree}$$

$$\langle E \rangle = 1.384 \text{ Hartree}$$

The new basis function did not work as well as $1 - x^2$.

Another try: Let $\psi_1(x) = c_1(1 - x^2) + c_2(1 - x^4)$ and set $c_1 = 1.2$. After normalization, we can get $c_2 = -0.203293$.

```

1 import numpy as np
2 from scipy.integrate import quad
3 from sympy import Symbol, Derivative

```

```

4  x= Symbol('x')
5  hbar = 1
6  m = 1
7
8  def integrand(x):
9      return (1.2 * (1 - x**2) - 0.203293 * (1 - x**4)) * (-hbar ** 2)/(2 * m) * (-2.4 + 2.43952 * x**2)
10
11 soln = quad(integrand, -1, 1)
12
13 print ('The expectation value of the energy = {0:1.4f} Hartree.'.format(soln[0]))

```

The expectation value of the energy = 1.2338 Hartree.

The energy is very close to the energy in part (a) (1.2337 Hartree).

2 Lectures 3: Many-electron atoms

Hartree's father performed the first calculations on multi-electron atoms by hand. Today those same calculations (much better ones, in fact) can be done in the blink of an eye on a computer. In this problem you will use a code first developed by Herman and Skillman in the 1960's to calculate the wavefunctions and energy of an atom using the Hartree-Fock-Slater (HFS) model, an early predecessor to DFT. The necessary software, rewritten in C++, is available at [/afs/crc.nd.edu/users/w/wschnei1/CBE547/fda.tar.gz](http://afs/crc.nd.edu/users/w/wschnei1/CBE547/fda.tar.gz).

Copy the software to your home directory, unpack (`tar -xzf fda.tar.gz`), change into the directory (`cd fda`) and compile the code (`make fda`) to create the fda executable. Look at the `OOREADME` file for information about the computer program and the format of the input. If you are brave, glance through the various source files (`*.cxx`) to get a sense of what the code is doing. Note that the code uses atomic units, Hartree for energy and bohr for distance.

(a) Run the `Ar.inp` example included in the directory (`fda Ar`). If all goes well, you should get an output file (`Ar.out`) and a dump file (`Ar.dmp`). Look at the `Ar.out` file to answer these questions:

- How many self-consistent field (SCF) iterations does the calculation take to converge?
- What is the final calculated HFS energy of the atom?
- What are the identities (1s, 2p, etc.) and energies of the occupied atomic orbitals?

output

```

1  itr = 29
2  l = 0
3  n = 1, e = -116.93655734 ( -233.87311468)
4  n = 2, e = -11.60370286 ( -23.20740572)
5  n = 3, e = -1.10223823 ( -2.20447646)
6  l = 1
7  n = 2, e = -9.27212718 ( -18.54425437)
8  n = 3, e = -0.57350833 ( -1.14701666)
9  cnv = 0.000009, mix = 0.499982
10 integrated charge = 17.999977 out of 18.000000
11
12          Orbital Summary
13  nl   occ      E      KE    <1/r>    <r>
14  1s   2.00  -116.9366  155.6552  17.6458  0.0856
15  2s   2.00   -11.6037   25.6407   3.5930  0.4087

```

```

16  2p   6.00   -9.2721   25.0012   3.5259   0.3675
17  3s   2.00   -1.1022    4.4193   1.0227   1.3584
18  3p   6.00   -0.5735    3.4406   0.8812   1.5596
19
20      Energy Summary
21  kinetic energy   =   542.0811
22  potential energy = -1068.9087
23  one-electron energy = -735.2963
24  two-electron energy =  208.4688
25
26  total energy = -526.8275
27  virial ratio =  -1.9719

```

It takes 29 iteration steps to converge.

The final calculated HFS energy of the atom is -526.8275 Hartree.

The identities and the energies of the occupied atomic orbitals orbital energies are here below.

nl	occ	E	KE	<1/r>	<r>
1s	2	-116.9366	155.6552	17.6458	0.0856
2s	2	-11.6037	25.6407	3.593	0.4087
2p	6	-9.2721	25.0012	3.5259	0.3675
3s	2	-1.1022	4.4193	1.0227	1.3584
3p	6	-0.5735	3.4406	0.8812	1.5596

- (b) The fda code solves the HFS equations on a radial grid. The `Ar.dmp` file contains the radial grid values and the total charge density in two columns of length 300, followed by an output of each orbital on the same grid. Plot out the charge density and each of the orbitals.

See plots

```

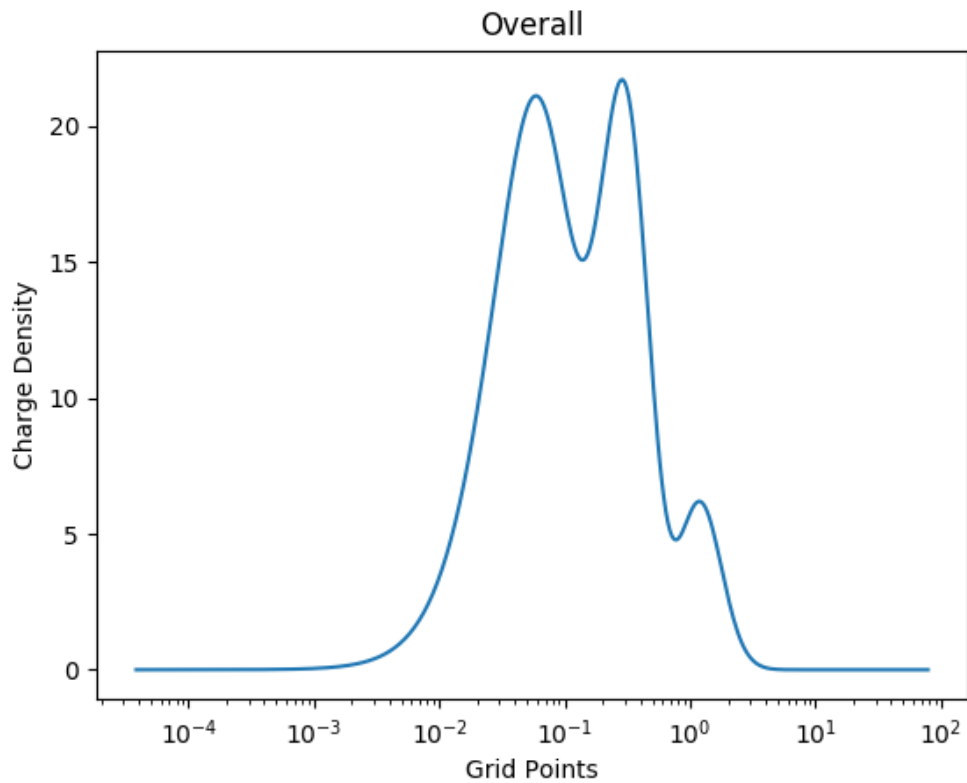
1  import matplotlib.pyplot as plt
2  import numpy as np
3
4  # Lets open the file in read mode
5  with open('FDA/Ar/Ar.dmp', 'r') as f:
6
7      # Read all the lines
8      lines = f.readlines()
9
10     # made list of grid points and total charge densities
11     grid_points = []
12     total_charge_densities = []
13
14     for line in lines[3:303]:
15
16         # split the lines into two columns
17         grid_point, tot_charge_density = line.split()
18
19         # We need to convert each line to a float add it to our lists
20         # store the each data in the lists
21         grid_points.append(float(grid_point))
22         total_charge_densities.append(float(tot_charge_density))
23
24     # Now the individual orbitals
25     one_s_charge_density = [float(x) for x in lines[304:604]]
26     two_s_charge_density = [float(x) for x in lines[605:905]]
27     two_p_charge_density = [float(x) for x in lines[906:1206]]
28     three_s_charge_density = [float(x) for x in lines[1207:1507]]
29     three_p_charge_density = [float(x) for x in lines[1508:1808]]
30
31     # plot the total charge densities

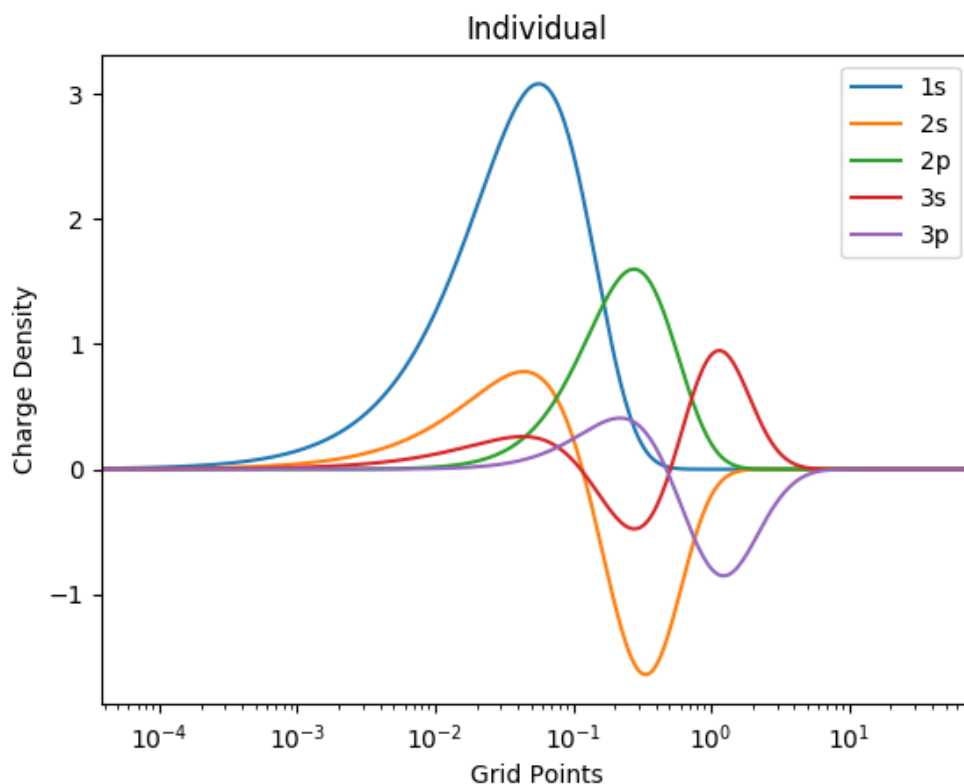
```

```

32 plt.figure()
33 plt.semilogx(grid_points, total_charge_densities)
34 plt.xlabel('Grid Points')
35 plt.ylabel('Charge Density')
36 plt.title('Overall')
37 plt.savefig('Ar-overall-charge-density.png')
38
39 # plot the individual orbitals
40 plt.figure()
41 plt.semilogx(grid_points, one_s_charge_density, label='1s')
42 plt.semilogx(grid_points, two_s_charge_density, label='2s')
43 plt.semilogx(grid_points, two_p_charge_density, label='2p')
44 plt.semilogx(grid_points, three_s_charge_density, label='3s')
45 plt.semilogx(grid_points, three_p_charge_density, label='3p')
46 plt.xlabel('Grid Points')
47 plt.ylabel('Charge Density')
48 plt.title('Individual')
49 plt.xlim(min(grid_points), max(grid_points))
50 plt.legend()
51 plt.savefig('Ar-orbital-charge-density.png')
52 plt.show()

```





(c) Choose one of the d block atoms. From the periodic table, figure out its electronic configuration and create an fda input file for it (follow the instructions in OOREADME for how to specify the atomic number and the orbital occupancies of your atom). Run the fda calculation on your atom.

- What is the final calculated HFS energy of the atom? How does it compare to Ar?
- What are the identities (1s, 2p, etc.) and energies of the occupied atomic orbitals?
- We chose Cu as an example. The electron configuration of Cu is $1s^2 2s^2 2p^6 3s^2 3p^6 3d^{10} 4s^1$. After 30 iteration steps, the calculation was converged.

input

```

1 300 0.0001 30.0
2 50 0.00001 0.10 0.50 0.682 0.0042
3 29.0 7
4 1 0 1.0 1.0
5 2 0 1.0 1.0
6 2 1 3.0 3.0
7 3 0 1.0 1.0
8 3 1 3.0 3.0
9 3 2 5.0 5.0
10 4 0 1.0 0.0

```

output

```

1      Orbital Summary
2  nl   occ   E      KE      <1/r>      <r>

```

```

3  1s  2.00  -325.9926  409.2882  28.6152  0.0527
4  2s  2.00  -39.3532   78.0568   6.2572  0.2363
5  2p  6.00  -34.7802   77.8063   6.2308  0.2051
6  3s  2.00   -4.4010   17.1817   1.9870  0.7120
7  3p  6.00  -2.9273   15.4940   1.8626  0.7384
8  3d 10.00  -0.4205   10.4720   1.4713  0.9331
9  4s  1.00   -0.2645    1.1132   0.4398  2.9912
10
11      Energy Summary
12 kinetic energy   = 1674.6889
13 potential energy = -3315.2096
14 one-electron energy = -2310.8336
15 two-electron energy = 670.3128
16
17 total energy = -1640.5208
18 virial ratio = -1.9796

```

The final calculated HFS energy of the Cu atom is -1640.5208 Hartree.

It is about 3.11 times the total energy of Ar atom.

The identities and the energies of the occupied atomic orbitals orbital energies are here below.

nl	occ	E	KE	<1/r>	<r>
1s	2	-325.9926	409.2882	28.6152	0.0527
2s	2	-39.3532	78.0568	6.2572	0.2363
2p	6	-34.7802	77.8063	6.2308	0.2051
3s	2	-4.401	17.1817	1.987	0.712
3p	6	-2.9273	15.494	1.8626	0.7384
3d	10	-0.4205	10.472	1.4713	0.9331
4s	1	-0.2645	1.1132	0.4398	2.9912

- (d) The orbital energies are a rough approximation of the energy to remove an electron from that orbital. Use your result to estimate the first ionization energy of your atom. How does it compare with the experimental first ionization energy?

The experimental first ionization energy of Cu is 745.5 kJ/mol (ref. <https://www.webelements.com/copper/atoms.html>)

From our HFS result, the estimated first ionization energy of Cu is 0.2645 Hartree (= 694.4 kJ/mol).

There is a 6.85 % difference from the experimental result.

- (e) You can also do calculations on anions or cations. Modify the input file for your atom by removing one of the valence electrons, to make it a cation. Rerun fda on the cation.
- How does the HFS energy of the cation compare to the neutral metal atom?
 - Do the energies of the orbitals go up or down from the neutral to the cation?
 - Do the electrons get closer to or further from the nucleus in the cation compared to the neutral? Use the expectation values of the distances from the nucleus (<r>) to answer the question.

input

```

1 300 0.0001 30.0
2 50 0.00001 0.10 0.50 0.682 0.0042

```



```

3 29.0 6
4 1 0 1.0 1.0
5 2 0 1.0 1.0
6 2 1 3.0 3.0
7 3 0 1.0 1.0
8 3 1 3.0 3.0
9 3 2 5.0 5.0

```

output

```

1
2      Orbital Summary
3  nl   occ      E      KE      <1/r>      <r>
4  1s   2.00    -326.3601  409.2915  28.6154  0.0527
5  2s   2.00    -39.7146   78.0586   6.2573  0.2363
6  2p   6.00    -35.1421   77.8096   6.2309  0.2051
7  3s   2.00     -4.7657   17.1741   1.9866  0.7121
8  3p   6.00     -3.2918   15.4907   1.8625  0.7383
9  3d  10.00     -0.7824   10.5383   1.4778  0.9229
10
11      Energy Summary
12  kinetic energy   = 1674.2338
13  potential energy = -3314.5026
14  one-electron energy = -2300.4296
15  two-electron energy = 660.1608
16
17  total energy = -1640.2688
18  virial ratio = -1.9797

```

- The total energy of Cu cation is -1640.2688 Hartree which is 0.252 Hartree higher than that of neutral Cu (-1640.5208 Hartree).
- All orbital energies of Cu cation decrease
- The average distances of 3d orbitals from the nucleus is decreased. The change of the average distance of the others are marginal.

(f) The difference in total energy between your neutral and cation calculations is another estimate of the first ionization energy of your atom. How does this estimate compare with experiment?

The estimated first ionization energy of Cu is 0.2645 Hartree.

But the the energy difference between Cu and Cu+ is 0.252 Hartree.

very close!