

PHY407-Lab07

Miriam Diamond

Due Nov 1 2022

Computational Background

The shooting method for boundary value problems To use the shooting method, you typically have to solve a nonlinear equation for a root, using something such as the secant method. In Example 8.9, the shooting method is used with RK4 to find the ground state energy in a square well potential for the time independent Schrodinger equation. Let's look over the code now and see if we can understand what it's doing: it is integrating a pair of ODEs (see the `solve(E)` function, which calls the function `f(r, x, E)`, which in turn calls the function `V(x)`). The solution to these equations depends on the parameter `E`. In the secant method loop, `E` is adjusted until a root is found for the variable `psi`, which the physics in the problem requires to be zero. Stated another way, the program does the following:

- Initializes `E` with reasonable guesses.
- Finds how `psi` depends on `E` using `solve(E)`.
- Adjusts `E` using the secant method.
- Repeats until `E` has converged to within a target accuracy.

(In the Physics Background section, we'll discuss why `psi` should be zero.)

Scipy constants Again, you can use `scipy.constants` if you want to use actual values of the relevant physical constants. If you dig into that package, the amount of information is quite extensive. The constants useful for this lab would be:

```
import scipy.constants as pc
a = pc.physical_constants['Bohr radius'][0]
E0 = pc.physical_constants['Rydberg constant times hc in eV'][0]
pc.m_e # electron mass
pc.hbar
pc.e # elementary charge
pc.epsilon_0
```

Note the `[0]` indexing is for the value. The second index is the unit, the third index the precision, which we won't care about in this lab.

Physics Background

The Hydrogen atom The time-independent Schrodinger equation is

$$-\frac{\hbar^2}{2m}\nabla^2\psi(\mathbf{x}) + V(\vec{x})\psi(\vec{x}) = E\psi(\vec{x}) \quad (1)$$

Where \mathbf{x} is the spatial coordinate. For a central potential $V = V(r)$, we can assume the wavefunction ψ can be separated according to $\psi(\mathbf{x}) = R(r)Y_\ell^m(\theta, \phi)$, where we are using standard notation with Y_ℓ^m being the spherical harmonic of degree m and order ℓ . In this case $R(r)$ satisfies a second order ODE

$$\frac{d}{dr} \left(r^2 \frac{dR}{dr} \right) - \frac{2mr^2}{\hbar^2} [V(r) - E] R = \ell(\ell + 1)R \quad (2)$$

and is subject to certain boundary conditions. For the particular case of a hydrogen atom with an electron interacting electrostatically with a proton, we can write

$$V(r) = -e^2/(4\pi\epsilon_0 r), \quad (3)$$

and in that case the solutions are well known. An online reference for the solution can be found below:

<http://hyperphysics.phy-astr.gsu.edu/hbase/quantum/hydwf.html>

<http://hyperphysics.phy-astr.gsu.edu/hbase/hyde.html>

The energy levels are

$$E_n = -E_0/n^2,$$

where $E_0 \approx 13.6\text{eV}$ is the ionization energy of hydrogen and $n \in \mathbb{N}^*$. The boundary condition is that $R(r) \rightarrow 0$ as $r \rightarrow \infty$. The ground state energy with $n = 1$ is about -13.6eV , the first excited state energy for $n = 2$ is at -3.4eV , etc. The wave functions corresponding to $n = 1, 2, 3$ and various values of ℓ can be found at the links above. This problem is well known and solved, and so provides a good test case for our home-built shooting method. We will focus only on the radial part $R(r)$ but if you want you can also create plots of the three dimensional wave function of the different energy levels of the hydrogen atom.

Questions

1. Space Garbage [40%]

See Newman's Exercise 8.8 (p. 353). You won't actually do it in this lab, but build upon the results. Use the results for part (a) as they are given in the textbook. The solution to part (b), it is provided in the script `Newman_8-8.py` included with the lab.

- (a) Code it up again (or add to the starter code), this time using an adaptive step size approach with the initial $h = 0.01$. We wish to achieve a target error-per-second $\delta = 10^{-6} \text{ m s}^{-1}$ for the position (x, y) . To do so, create a plot of the individual points of the adaptive step size algorithm at each time step, and overlay it on top of the plot from `Newman_8-8.py`. For the non-adaptive scheme, you now need to use $h = 0.001$ ($N = 10,000$ time steps) in order to roughly achieve the same error-per-second over the course of the simulation.

When calculating ρ , use the formula

$$\rho = \frac{h\delta}{\sqrt{\epsilon_x^2 + \epsilon_y^2}}, \quad (4)$$

which uses the Euclidean error for the position in the (x, y) plane, $\sqrt{\epsilon_x^2 + \epsilon_y^2}$, as described in the text on pp. 359-360 (see the discussion around eqn. 8.54). Do you see the effect of the adaptive time step?

Submit code, plot, and a short description of what you see.

- (b) Once you have convinced yourself that your new code is working, compare the clock time it takes to execute (with the $\delta = 10^{-6}$) to the time taken for the original non-adaptive code to execute (with $h = 0.001$, $N = 10,000$).

Submit printed output or a short answer.

- (c) For your new adaptive code, create a plot of the size of the time step as a function of time. (You may drop the first several points in your plot because it takes a little while for the adaptive routine to settle down into predictable behaviour.)

Try to relate the time step size to the solution. Under what circumstances do the time steps tend to be relatively short or relatively long?

Submit code, plot(s), and written answers.

- (d) **Optional** For the non-adaptive scheme, optimize the value of h in order to achieve as closely as possible the error-per-second $\delta = 10^{-6} \text{ m s}^{-1}$ over the course of the simulation.

2. Hydrogen Atom [60%]

In this exercise you will use shooting and RK4 to find the bound states of hydrogen for a couple of cases. The calculation will involve finding both the energy eigenvalues and the related eigenfunctions for the radial part of the Schrodinger equation. In your calculation, you will need to set initial conditions on $R(r = 0)$ and an intermediate function, say, $S(r = 0)$. Because the RHS of (2) diverges at $r = 0$, you should carry out your integration starting at $r = h$, where h is the step size, and set $R(h) = 0$ and $S(h)$ to a constant (1 is fine). This will cause the eigenfunction to go to zero at $r = 0$, which, it turns out, is not a good assumption for one of the cases below, but will not affect the energy eigenvalue calculation too much.

Now do the following:

- (a) Starting from (2), and with reference to Section 8.6.3, Example 8.9, and Newman's code `squarewell.py` online, write out the second order ODE in r as a pair of coupled first order ODEs for R and S , and implement this in Python. Some notes:
- In the secant method loop in Example 8.9, the energy E is adjusted until the value `psi` is as close to zero as possible because `psi` represents the wavefunction at the right-hand boundary of the infinite square well. In the hydrogen atom there is no right wall; instead, your domain extends to "infinity", but you can get reasonable answers for a large value of r , and you need to set the wavefunction R to zero there. This is similar to the issues raised in Exercise 8.14 in Newman.
 - There are a few adjustable parameters in this exercise: one is the stepsize h and the other is the maximum value of r , which is r_∞ . To start with, set $h = 0.002a$ and $r_\infty = 20a$, where $a \approx 5 \times 10^{-11} \text{ m}$ is the Bohr radius. You can make r_∞ larger and make h smaller to improve the solution. You will need to find out which one makes more of a difference in different situations.
 - You will need to set a left boundary condition on R near $r = 0$. For convenience, set $R(h) = 0$, even though this is not required by the mathematics and indeed contradicts the solution for $n = 1$, $\ell = 0$. This is a source of inaccuracy in the code.
 - You will also need to set a target energy convergence. This is `e/1000` in `squarewell.py` but you should look at the impact of reducing this.
 - Finally, you will need to initialize your eigen energies at some level for the secant method to work. It is easy to miss the energies if you aren't careful. It can be a good idea to bracket the energies, so, for different n , you can choose initial values of $E1 = -15 \cdot e/n^2$ and $E2 = -13 \cdot e/n^2$.

Submit the code.

- (b) Calculate numerically the ground state energy ($n = 1$) and the first excited state energy ($n = 2$) for $\ell = 0$, and the energy for $n = 2$ and $\ell = 1$ (which is supposed to be the same as the energy for $n = 2$ and $\ell = 0$). These can all be compared to the known solution for the hydrogen atom. Look at the effect of adjusting the various parameters discussed above.

Submit written answers.

- (c) For the three cases in Part (b), modify your program to plot the normalized eigenfunction R . Normalize by calculating $\int |R(r)|^2 dr$ over the range of r and plot it as a function of r over a finite range (this is similar to Exercise 8.14(c)). You can use either the trapezoidal or Simpson's rule to do the integration.

For some eigenvalues, you might obtain spurious large values of R at the right-hand end (r_∞) of the wave function. By definition, the wave function should go to 0 at the boundary. It does not because we don't have the energy *exactly* right and it's extremely sensitive to the energy value. If the energy is off by even the tiniest bit then the value tends to diverge. This is not a problem for determining the energy itself. But it can give problems when normalizing and plotting the wave function.

Submit code and plots.

- (d) **Optional** The first online reference in the Physics Background section includes analytical solutions for $R(r)$. Scale these solutions so they can be plotted as overlays on top of your numerically calculated solutions. How do the two solutions compare in terms of overall shape and zero crossings for the wave functions?