Lab Assignment 1: Quantum Chemistry and the Hydrogen Emission Spectrum

The periodic table is central to chemistry. According to Britannica, "Detailed understanding of the periodic system has developed along with the quantum theory of spectra and the electronic structure of atoms, beginning with the work of Bohr in 1913." In this lab assignment, a University of Alberta student explores the Bohr model's accuracy in predicting the hydrogen emission spectrum, using observed wavelengths from a US National Institute of Standards and Technology (NIST) database.

Version 0: Get Started

Unzip `V0GetStarted.zip` into your Working Directory. In the Files pane, look for `hSpectrum.py`, a text file of Python code, `hSpectTest_v0.txt`, a text file of iPython Console input/output, also called Console side effects, and `hSpectPlot_v0.png`, an image file of additional Console side effects.

Open, in the Editor, and review `hSpectrum`. The Python program, a script, consists of two header sections and three code sections. The second section, having `import` statements, elaborates on a title with a one-paragraph description. Code sections start with comment headers having short titles.

The program has no syntax errors. Select `Run >> Run` from the menu to run, or execute, the code. No runtime errors occur if you enter `2`, as specified in `hSpectTest_v0`, when prompted for input. Open `hSpectTest_v0` in the Editor and compare it to the Console text. They should be identical (ignore any plot-related warning). Compare the Plots pane plot to `hSpectPlot_v0`. They should be identical.

As you will edit `hSpectrum`, make a copy for easy reference. Select `File >> Save as...` from the menu and save it as `hSpectrum_v0.py` in the Working Directory. Before you submit Version 1 or 2 of your `hSpectrum`, the lab instructor may ask you to rename the file, e.g., to append your CCID to the filename. If you need to save a plot as a `.png` file select `Save plot as...` after right clicking.

Version 1: Simulate Data

Unzip the two files, `hSpectTest_v1.txt` and `hSpectPlot_v1.png`, in the `V1SimulateData.zip` file into your Working Directory. When you complete Version 1 sufficiently and test it as specified, the side effects of your `hSpectrum` program should match `hSpectTest_v1` and `hSpectPlot_v1`.

Under `EXPERIMENT DATA`, a NumPy array, `nist`, is created. Revise a statement, so that the list, `data`, has three more entries at the end: `397.1198`, `389.0166`, and `383.6485`. You may split the right-hand-side (RHS) over two lines of code. Entries are observed wavelengths (NIST relative intensity of at least `1.2e-03`) in the visible band (`380` to `750` `nm`) of the hydrogen emission spectrum (vacuum).

Under `MODEL SETUP`, a value is assigned to a variable, `rydberg`, representing the "Rydberg constant". Revise the `print` statement to output its value, with no fractional digits or decimal point, and to display its units, `1/m`. Bohr predicted its value, R, from the electron mass, m_e, the fundamental charge, e, the permittivity of free space, ϵ_0, the Planck constant, h, and the (vacuum) speed of light, c, as follows:

$$R = \frac{m_\mathrm{e} e^4}{8\epsilon_0^2 h^3 c}.$$

Assign a RHS expression to the Rydberg constant, `rydberg`, according to the Bohr model. Under `MODEL SETUP`, first create a variable for each required physical constant. Enter RHS values to eight significant figures. Use metric, i.e., Système International (SI), units. Enter short unit names as in-line comments. Instead of copying values from another student, look up the constants in a reputable source.

The Bohr model simulates observed wavelengths, λ, in an emission spectral series in terms of energies due to transitions from initial states, n_i, to a final state, n_f, where states are integers that index stable electron orbits about the atom's nucleus. Variables for these states already exist in the program as a vector, `ni`, and a scalar, `nf`, respectively. Using the Rydberg constant, R, the model is as follows:

$$\lambda = \frac{1}{R\left(\frac{1}{n_\mathrm{f}^2} - \frac{1}{n_\mathrm{i}^2}\right)}.$$

Compare `hSpectPlot_v1` to the plot produced by running your program. Under `SIMULATION DATA`, use Matplotlib to annotate the existing plot with the required axis labels and title. Before you add a legend, compute the Bohr model data points and plot them with the indicated colour and marker style. Convert from model units, `m`, to plot units, `nm`. Do *not* use loops or Python expressions that fall outside the Programming Basics scope. Use NumPy, a productive way to meet these requirements.

Immediately under `EXPERIMENT DATA` and `SIMULATION DATA`, write comments to summarize what the section does, or is supposed to do, in your own words. Complete the program's comment header. Ask the lab instructor for advice, during your lab session, on what percentages to allocate sources of physical constants. Submit your Version 1 solution, `hSpectrum`, by the Version 1 deadline.

Version 2: Analyze Error

Unzip the *three* files, `hSpectTest_v2.txt`, `hSpectPlot_v2.png`, and `hSpectBar_v2b.png`, in the `V2AnalyzeError.zip` file into your Working Directory. When you complete Version 2 sufficiently and test it as specified, side effects of your `hSpectrum` should match `hSpectTest_v2`, `hSpectPlot_v2`, and `hSpectBar_v2b`. Unzip a *fourth* file, `hSpectBar_v2a.png`, to help you complete Version 2.

With a consistent comment header, create a new script section, `ERROR ANALYSIS`, after the existing section, `SIMULATION DATA`. Under `ERROR ANALYSIS`, use NumPy and Matplotlib to compute, plot, and annotate a bar graph representing the difference between the experiment and simulation data, namely the wavelengths from the NIST database minus those predicted by the Bohr model. Do not use loops, *etc.*, as before. Compare `hSpectBar_v2a` to this second plot. They should be identical.

Introduce a correction to the Rydberg constant by adding statements under `MODEL SETUP`. The Bohr model implicitly assumed an infinite nuclear mass. To account for a finite nuclear mass, an improved model multiplies the original Rydberg constant by a scale factor, which is as follows for hydrogen:

$$\frac{m_\mathrm{p}}{m_\mathrm{p} + m_\mathrm{e}}.$$

Look up the proton mass, m_p, to eight significant figures and code the above correction. Run your improved program. Compare `hSpectBar_v2b` to your second plot. They should be identical.

Examine `hSpectTest_v2`. Enter a string expression, `"m"+chr(8315)+chr(185)`, in the Console and observe what it returns. Correct `MODEL SETUP` accordingly. Under `ERROR ANALYSIS`, compute the worst-case error and `print` it as shown. The worst-case error is the maximum *absolute* difference between the observed and predicted wavelengths. Use NumPy. Do not use loops, *etc.*, as before. The maximum and absolute value functions are `np.max` and `np.abs`, given the `import` statements.

Under `MODEL SETUP` and `ERROR ANALYSIS`, and for any undocumented code section, summarize what the section does, or is supposed to do, in your own words. Review the program's initial header. Ask the lab instructor for advice, during any lab session, on potential revisions to percentages you gave sources of physical constants. Submit your Version 2 solution, `hSpectrum`, by the Version 2 deadline.

Revision History

This document and related files were created and reviewed, in 2022, by Dileepan Joseph and Wing Hoy. They were further reviewed and edited, in 2024, by Antonio Andara Lara and Dr. Joseph. The concept was motivated, in part, by an entry in the ENCMP 100 Programming Contest – Equilibria.