Physics 305 – Computational Physics, Fall 2020
Term Project
Full project submission Due Date: Tuesday December 15, 5pm
Presentation Phase: November 30 - December 11

The program in your term project can be either submitted as a python program or ipython notebook, where the latter is preferred. The program, an explanation of what the program does, along with answers to **all** questions asked should be uploaded to d2l.

You are expected to write a term paper (in word or Latex) on your project that discusses the problem you are trying to solve, the basic equations that govern the problem, includes plots that show the solutions, and describes the solution and the numerical method involved. In addition, you must demonstrate that your solution is correct by showing that the code converges at the expected order. If your code does not converge at the expected order you should try to identify potential reasons for why this is the case. You are expected to work on your term project by yourself..

Your term project will receive full credit **only** if: (a) the program runs successfully without errors using python 3, (b) the programs have explanatory comments and variable names that identify with the problem equations you are trying to solve, (c) give the correct output, and (d) demonstrate the validity of the solution through convergence plots. No credit will be given to late term projects.

The term paper is as important as the code (50% of the term project credit will go to the code and the other 50% to the paper). Answers to the questions and analysis requested below should be elaborated in the report. Plots should be clearly labeled and be properly described in the report, and not just shown. You will need to explain what each and every plot demonstrates. A polished paper written in word or LaTex (preferred, e.g. please try overleaf) is expected to get full credit.

Note: Before you present results from numerical integrations that answer the questions in the project, it is critical to *first* perform the convergence tests for one case, and to estimate errors. This will tell you how small a step size is necessary for accurate solutions. Only after errors are estimated, does it make sense to run your code for producing results that help you learn more about the system you study.

I. THOMAS-FERMI MODEL OF NEUTRAL ATOMS

This project is aimed to find numerical solutions to the Thomas-Fermi equation in the case of neutral atoms. The Thomas–Fermi (TF) model is a quantum mechanical theory for the electronic structure of many-body systems developed semiclassically shortly after the formulation of the Schrödinger equation. You will be solving the for atom potential and given that you will be solving for the electron density.

The TF equation is a 2nd-order, non-linear ordinary differential equation and is written as follows

$$\frac{d^2\phi}{dx^2} = \frac{\phi^{3/2}}{\sqrt{x}}. \tag{1}$$

The equation is strictly speaking a two-point boundary value problem with boundary conditions $\phi(0) = 1$ and $\phi(\infty) = 0$. However, we will be solving this equation as an initial value problem by using the solution to the equation for very small x to compute the derivative. Before we discuss the initial conditions, let us introduce some more useful background.

For chemical potential $\mu = 0$, the TF equation is a model of a neutral atom, with an infinite charge cloud where electron $n(r)$ is everywhere nonzero and the overall charge is zero. Once a solution for ϕ has been obtained the electron density is given by the following equation

$$n(r) = \left(\frac{5}{3}C_{\rm kin}\right)^{-3/2}\left[\frac{Ze^2}{r}\phi\left(\frac{r}{b}\right)\right]^{3/2}, \tag{2}$$

where Z is the atomic number, $b = \frac{1}{4}\left(\frac{9\pi^2}{2Z}\right)a_0$, with a_0 the Bohr radius, and $C_{\rm kin} = \frac{3h^2}{40m_e}\left(\frac{3}{\pi}\right)^{2/3}$. Thus, after a solution to the TF equation has been obtained, the density of the electron cloud can be obtained from Eq. (2).

To integrate the TF equation you need "initial" or better stated boundary conditions at $x = 0$. The condition for ϕ as stated above is $\phi(0) = 1$. You will also need an initial condition for the derivative of ϕ at $x = 0$, i.e, $d\phi/dx|_{x=0}$. This will be obtained from the solution of the TF equation for small x, which is given by

$$\phi(x) = 1 - Bx + \frac{1}{3}x^2 - \frac{2B}{15}x^4 + x^{3/2}\left[\frac{4}{3} - \frac{2B}{5}x + \frac{3B}{70}x^2 + \left(\frac{2}{27} + \frac{B^2}{252}\right)x^3 + \dots\right]. \tag{3}$$

Where the constant B is given by $B = 1.588071022611375312718684509 42$. Use Eq. (3) to find what is the value of $\left.\frac{d\phi}{dx}\right|$ for small x and thus start the numerical integration of the TF equation. To solve numerically the TF equation you will first need to cast it to first order form. Show your work in the term project report. Once you have that, the steps for integrating are the following

Step 1. Set $\phi(0)$ and $\left.d\phi/dx\right|_{x=0}$.

Step 2. Integrate the TF equation from the center of the atom outward treating the dimensionless radius x as a "time" parameter marching forward.

Step 3. The potential ϕ is supposed to go to 0 at infinity. Thus, integrate to large enough x, so that ϕ becomes close to 0.

In solving the equation numerically you will have terms at the right-hand-side that are multiplied by $1/\sqrt{x}$, and hence are singular at $x = 0$. Thus, if you were to start evaluating the right-hand-side at $x = 0$, the code would crash because it would encounter terms that are $1/0$. To avoid this start the integration at $x = \epsilon$ (with $0 < \epsilon \ll 1$) and use the solution for small x to determine the value of $\phi(\epsilon)$ and $\left.d\phi/dx\right|_{x=\epsilon}$. Do not set ϵ close to the round-off error precision of the machine otherwise there will be inaccuracies. You will need to experiment with different values of ϵ and demonstrate with plots that your solution of $\phi(x)$ is independent of the value of ϵ, once ϵ has become smaller than a certain value.

1. Implement RK4, to integrate numerically the first-order of ODEs that corresponds to the TF equation (1). Show a plot of your numerical solution and overlay the solution for small x to see how they compare. When do you start (at what x) to see significant deviations between the numerical solution and the solution for small x?

2. Using your solution for ϕ introduce a properly normalized dimensionless electron density (call it $\tilde{n}$), and a dimensionless radius (call it $\tilde{r}$, and make a plot of $\tilde{n}$ vs $\tilde{r}$. Describe the behavior of the charge density. How far does the electron cloud extend? Do you think the TF model is a realistic model for atoms?

3. **Self-convergence**: Use a number of step sizes and make a plot to demonstrate that the code solution for $\phi(x = 10)$ self-converges. Does the order of self-convergence match your expectation? If not try to explain why this is case.

4. Using the order of convergence you determined, employ Richardson extrapolation to determine an error for the solution for $\phi(x = 10)$.