1)

4.11: Plot u vs x obtained from all three solutions and compare to see if there are differences.

Apply the Numerov Algorithm to solve:

$$u"(x) = -4\pi^2 u(x)$$

With $u(0) = 1$ and $u'(0) = 0$. Discuss the accuracy of the result by comparing it with the solution obtained via the fourth degree Runge-Kutta algorithm and with the exact result

Questions)

The Numerov algorithm requires Dirichlet boundary conditions in order to solve for the SHO, however, the problem gives us Cauchy boundary conditions. Do you have any recommendations on approaches for this?

> The starting point to run the Numerov recursion relation is the first two points: u[0] and u[1]. You can use the initial conditions specified in the question (u(0) and u'(0)) to deduce u[0] and u[1] to start the recursion.

How do we determine the x values for the corresponding u values in the Numerov program? Are they equivalent to the i values of each u value?

> Yes, i indexes the steps in x as usual. The spacing between x points is usually h, and you can define it as you wish. The Schrodinger code (Page 107) sets h in the beginning.

I'm unsure of how to obtain the exact values for comparison

> The solution is obtained by assuming a A*cos(B*x). Plugging into the equation and applying the boundary conditions gives A and B.

2)

4.12: Plot u vs x obtained from the exact solution and from the shooting method and compare to see if there are differences.

Apply the Shooting Method to solve the eigenvalue problem

$$u"(x) = \lambda u(x)$$

Questions)

"...I am confused how to start the problem"

> This code you might have to start from scratch. To solve this use a differential equation solver, i.e., the RK method, guess the eigenvalue, and use a root finder to vary the eigenvalue until the boundary value is correct.

I'm unsure of how to obtain the exact values for comparison

> The solution is obtained by assuming a $A * \cos(B * x)$. Plugging into the equation and applying the boundary conditions gives A and B.

3)

In this problem we will solve the one-dimensional time independent Schrödinger equation (Eq. 4.91 in your text) using two different methods. Consider a single electron of mass m_e in one dimension trapped in a potential well of the following form:

$$V(x) = \begin{Bmatrix} V_0 & \text{if } 0 < x < x_0 \\ V_0 & \text{elsewhere} \end{Bmatrix}$$

It's simplest to use atomic units (a.u.) with $m_e = e = h = c = 1$. Set the potential parameters to be $x_0 = 5$ a.u., and $V_0 = 50$ a.u.

a)

Use the code from Chapter 4.9 of the text (which uses the Numerov algorithm with secant method) to find the two lowest eigenvalues and eigenfunctions of the Schrödinger equation with the above potential.

b)

Then develop a program that does the same thing as part a), but with the Runge-Kutta and secant methods.

c)

To compare results from parts a) and b), plot the square of the eigenfunctions versus position for both methods to see if there are differences.

Questions)

How should we find/choose the initial values for the Runge-Kutta algorithm?

> Recall the method for doing the Schordinger equation. We need to start far out in the exponential region where we can pick small numbers to start. Your choice shouldn't change your answer. See Schrodinger code in the method to calculation the wave function: double u0 = 0, u1 = 0.01;

Which function should replace the g function in the Runge-Kutta algorithm?

> To determine the g function for Runge-Kutta, take a look at Eq. 4.92 in the text. Mapping Eq. 4.92 to two first order equations let's you determine g. Compare 4.92 with the equation you solved in Problem 4.12. Here you can see that the second order differential equation can be written as two first orders.
>
> You can also see how to do the mapping by comparing Eq. 4.92 with Eqs. 4.58-4.60.

"...is there any way you could tell me what the eigenvalues are supposed to be for the two lowest states since you aren't asking for the eigenvalues but just for psi?"

> The lowest two eigenvalues for this problem are 7.59 and 14.01.