# PH 3120 - Computational Physics Laboratory 1

## CPL100 - Introductory Session

**Section 1 & 2**

1. The quantum mechanical linear harmonic oscillator is a model for systems describing the behavior of a particle under the influence of a harmonic potential.

   The wavefunctions of the oscillator are given by,

$$\Psi_n(\alpha x) = \left(\frac{\alpha}{\sqrt{\pi}\, 2^n n!}\right)^{1/2} e^{-\frac{(\alpha x)^2}{2}} H_n(\alpha x)$$

   The probability densities for a wavefunction at mode $n$ is given by $|\Psi_n^2|$

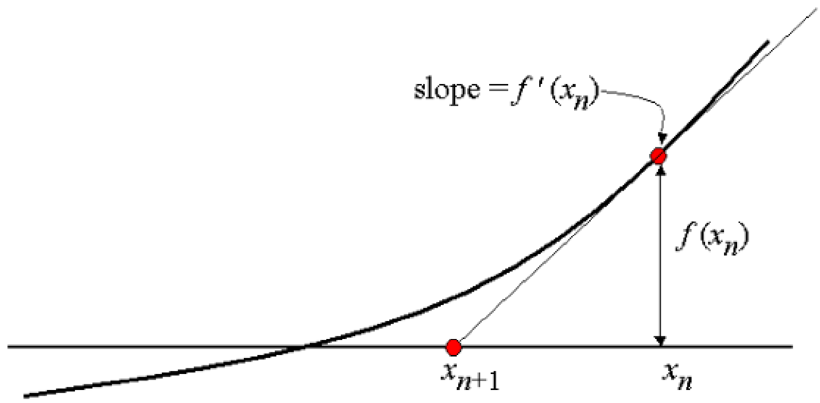
   Where $H_n(\alpha x)$ are the Hermite Polynomials.

   Pick a suitable range for x such that x ranges from a negative value to a positive value and plot the wavefunctions for $n$ = 0,1,2,3,4,5 in separate graphs. Choose three of them and plot in the same graph. The graphs should be labeled appropriately according to the $n$ value. Take $\alpha = 1$. Plot the probability densities for $n$ = 0,1,2,3,4,5 in separate graphs as well. Note that each graph should be shown in the same range of $x$.

   The Hermite polynomials are given below

$$\begin{aligned}
H_0(x) &= 1 \\
H_1(x) &= 2x \\
H_2(x) &= 4x^2 - 2 \\
H_3(x) &= 8x^3 - 12x \\
H_4(x) &= 16x^4 - 48x^2 + 12 \\
H_5(x) &= 32x^5 - 160x^3 + 120x \\
H_6(x) &= 64x^6 - 480x^4 + 720x^2 - 120
\end{aligned}$$

2. The Newton-Raphson method (also known as Newton's method) is a way to quickly find a good approximation for the root of a real-valued function. It uses the idea that a continuous and differentiable function can be approximated by a straight line tangent to it.




suppose you need to find the root of a continuous, differentiable function *f(x)*, and you know the root you are looking for is near the point $x = x_0$. Then Newton's method tells us that a better approximation for the root is,

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

### Pseudocode for Newton-Raphson method

```
1. Start

2. Define function as f(x)

3. Define derivative of function as g(x)

4. Input:

     a. Initial guess x0

     b. Tolerable Error e

     c. Maximum Iteration N

5. Initialize iteration counter step = 1

6. Do while abs f(x1) > e

     If g(x0) = 0

          Print "Mathematical Error"

          Stop
```

```
    End If

    x1 = x0 - f(x0) / g(x0)

    x0 = x1

    step = step + 1

    If step > N

        Print "Not Convergent"

        Stop

    End If

7. Print root as x1

8. Stop
```

Find,

i. The positive and negative roots of $x^2 - 5$

ii. The positive root of $x^3 + 4x^2 - 3$

iii. The positive roots of $2 - x^2 sin(x)$ less than 3 (Hint: only two roots are less than 3)

3. A ball is thrown from a cliff which is 15 m above the sea level, having an initial velocity of 24 $ms^{-1}$ at an angle of elevation equal to 54˚.

    i. How long does the ball take to land on the sea?

    ii. How far from the cliff does the ball land on the sea?

You may use a graph to verify your answer/s. The answers should be generated algorithmically, and the algorithmic method/s should be demonstrated in the code clearly without directly writing the final answers.

4. The potential created at a radial distance $r$ from the center of a sphere with a surface charge density $\sigma_0$ and a radius $R$ is given by the following equation.

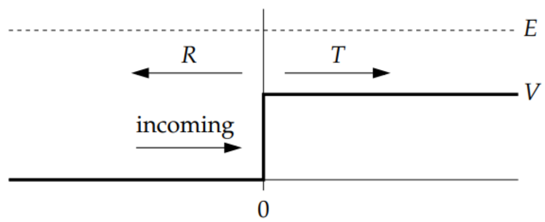
$$V(r,\theta) = \begin{Bmatrix} \dfrac{\sigma_0 r}{2\epsilon_0}\left[P_1(\cos\theta) - \dfrac{1}{4}\left(\dfrac{r}{R}\right)^2 P_3(\cos\theta) + \dfrac{1}{8}\left(\dfrac{r}{R}\right)^4 P_5(\cos\theta)\right], & r \leq R \\ \dfrac{\sigma_0 R^3}{2\epsilon_0 r^2}\left[P_1(\cos\theta) - \dfrac{1}{4}\left(\dfrac{R}{r}\right)^2 P_3(\cos\theta) + \dfrac{1}{8}\left(\dfrac{R}{r}\right)^4 P_5(\cos\theta)\right], & r > R \end{Bmatrix}$$

Here $r$ and $\theta$ are usual cylindrical coordinates. $P_n(\cos\theta)$ are the Legendre Polynomials. The Legendre Polynomials are,

$$
\begin{aligned}
P_0(x) &= 1 & P_1(x) &= x \\
P_2(x) &= \frac{1}{2}(3x^2-1) & P_3(x) &= \frac{1}{2}(5x^3-3x) \\
P_4(x) &= \frac{1}{8}(35x^4-30x^2+3) & P_5(x) &= \frac{1}{8}(63x^5-70x^3+15x)
\end{aligned}
$$

Your task is to obtain four graphs to demonstrate the behavior of the potential inside and outside the sphere. The behavior of each graph should be explained in words as well. Feel free to use $r$, $R$, and $\theta$ as you wish.

5. A well-known quantum mechanics problem involves a particle of mass m that encounters a one-dimensional potential step, like this:




The particle with initial kinetic energy $E$ and wavevector $k_1 = \sqrt{2mE}/\hbar$ enters from the left and encounters a sudden jump in potential energy of height $V$ at position $x = 0$. By solving the Schrödinger equation, one can show that when $E > V$ the particle may either (a) pass the step, in which case it has a lower kinetic energy of $E - V$ on the other side and a correspondingly smaller wavevector of $k_2 = \sqrt{2m(E - V)}/\hbar$, or (b) it may be reflected, keeping all of its kinetic energy and an unchanged wavevector but moving in the opposite direction. The probabilities $T$ and $R$ for transmission and reflection are given by,

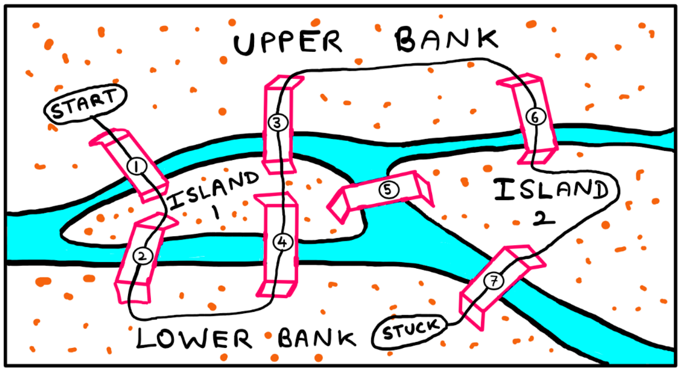
$$T = \frac{4k_1k_2}{(k_1+k_2)^2} \qquad R = \left(\frac{k_1-k_2}{k_1+k_2}\right)^2$$

Suppose we have a particle with mass equal to the electron mass $m = 9.11\times10^{-31}\,kg$ and energy (E) equals to $10\,eV$.

i. Write a Python program to compute and print out the transmission and reflection probabilities using the formulas above. Note that $1\,eV = 1.6 \times 10^{-19}\,J$.

ii. Find the transmission and reflection probabilities by changing the potential step height (V) from 0 to $9\,eV$ and show the variation of transmission and reflection probabilities with respect to potential step height in separate graphs.

6. The following diagram shows the bridges of the historic Prussian city Königsberg, which is now Kaliningrad in Russia. A visitor wants to find out whether they can cross all these bridges exactly once and come back to the starting point. What do you think? Write an algorithm to find out the answer.




7. Certain atomic nuclei are unstable, and will over time decay into other nuclei, through the emission of radiation. We call such atomic nuclei radioactive. The process of radioactive decay is completely random, but we can model how much remains of the original matter after a certain time for large collections of atoms.

   From an original mass $N_0$ of radioactive material, the remaining mass after a time t (in seconds) is given by the equation for radioactive decay:

$$N(t) = N_0 e^{-\frac{t}{\tau}}$$

   $\tau$ is the so-called 'mean lifetime' of the radioactive material, and represents the average lifespan of a single nucleus in the radioactive material. A larger value indicates more stable nuclei, and it can vary from $10^{30}$ s for very stable materials to $10^{-20}$ s for very unstable materials.

   a. Carbon-11 is an unstable carbon isotope and has a mean lifetime of $\tau = 1760\, s$. Make a program that calculates how much remains of an original mass $N_0$=4.5 kg of carbon-11 after 10 minutes.

   b. The mean lifetime makes for a pretty formula, but we are more often talking about the half-life of radioactive material. The half-life represents the time it takes for the material to reduce to exactly half of its original mass. The relation between the mean lifetime and the half-life is given as

$$\tau = \frac{t_{\frac{1}{2}}}{ln2}$$

The half-life of carbon-11 is $t_{\frac{1}{2}}$ = 1220 s. Rewrite your program such that it first calculates the mean lifetime from the half-life, and then calculates the remaining mass, just like in exercise a. Check that you get the same results as in exercise a.

8. According to Fourier transformations, sin waves can be used to create square, triangle and sawtooth waves.

$$sq(t) = \frac{4}{\pi}\sum_{k=1}^{\infty}\frac{sin(2\pi(2k-1)ft)}{2k-1} \Rightarrow Square\ wave$$

$$tr(t) = \frac{8}{\pi^2}\sum_{k=0}^{N-1}(-1)^k(2k+1)^{-2}sin(2\pi f(2k+1)t) \Rightarrow triangle\ wave$$

$$sw(t) = a\left(\frac{1}{2} - \frac{1}{\pi}\sum_{k=1}^{\infty}(-1)^k\frac{sin(2\pi kft)}{k}\right) \Rightarrow Sawtooth\ wave$$

Where, f = frequency, t = time, N = number of nodes, a = amplitude.

i. Using frequency as 1 Hz and time range from 0 to 2 at 200 iterations, plot these waves using the above equations. Take amplitude (a) as 1.
ii. To get a better square wave clip the resultant wave at 0.5 and -0.5 y values. Show your results. (Use clipping attribute available in numpy using the syntax ***np.clip(wave_function_ values, clip_min, clip_max)***)

9. Following equation depicts how a certain rabbit population in a jungle varies over the time.

$$X_{n+1} = r\,X_n(1 - X_n)$$

Where r is the rate of growth.

I. Starting from r = 0.5 increase in 1.0 increments till 3.5 and after that increase in 0.1 increments till 3.9. Observe the population (x) against the iteration (n) for each case. (Population is taken as a probability, so it should be between 0 and 1. For this question take x = 0.5, take n from 0 to 1000 in 100 steps). State your observations in dedicated plots. Comment on the apparent changes as the rate of growth increases.

II. What you experienced (Deviation from the stability) in part 1 is the base of the physical principle chaos theory. This happens with a phenomenon called bifurcation. It is believed that any function (y = f(x) : in this case y = $X_{n+1}$ and x = $X_n$) that shows a 'single maxima' given the appropriate number of iterations, eventually shows this behavior. Show that this is indeed the case for the above equation. (It is recommended to define a function for the equation first.)

III. To get a better idea about bifurcations, draw a plot of x vs r. Change r from 0.5 to 4 in 100000 steps and for each r, generate and use a random x in between 0 and 1. Use 100 iterations (i = 100). For better plotting, set 'linestyle' to 'none' and 'marker' to 'pixel'. Observing the plot, state what is happening as the growth rate increases.

10. In this exercise, you need to take a few measurements of your body parts

    1. Measure the length of your pointer left hand and right hand (Take 5 measurements for each finger) and record the details in the sheet given here.
    2. Visualize the data set in an appropriate way and highlight the important findings. Use error bars to present your data. You can use the syntax **`matplotlib.pyplot.errorbar(x_axis_data, y_axis_data, yerr = y_axis_errors, xerr = x_axis_errors, fmt = ‘o’, capsize = 5)`**

*******