EE2703: Applied Programming Lab
Assignment 6A
Tubelight Simulation

Devaganthan S S
EE19B018

April 29, 2021

0.1 Abstract

This assignment aims to Understand how Simulations can be Done in Python

0.2 Introduction

We try to simulate a 1-Dimensional Model of a Tubelight. A uniform Electric Field exists across the length of the Tubelight. Electrons are Injected at the cathode end of the tube light, which initially has zero energy. Due to the Electric Field, the electrons gain Kinetic Energy. When the Electrons gain a Threshold Energy, say E0, upon collision with Atoms in the Tube lights, the energy of the Electron is completely transferred to the atom which excites the atom, resulting in Light Emission. The Uncollided Electrons hit the other end of the tube light and loose all the energy. We try to simulate the Tube Light in Python. We plot the Light Intensity vs Position, Electron Density Vs Position, Phase space Diagram.

0.3 Results and Implementation

0.3.1 Initialization

We initialize the parameters that define the Tube Light Model. The parameters can also be given via the Command Line Arguments. The below code initializes the parameters

```
# Intiatializing the Values
n = 100  # spatial grid size.
M = 5  # number of electrons injected per turn.
nk = 500  # number of turns to simulate.
u0 = 5  # threshold velocity.
p = 0.25  # probability that ionization will occur
Msig = 2  # Standard Deviation
xx = zeros(n*M)  # Electron Postion Vector
u = zeros(n*M)  # ELectron Velocity Vector
dx = zeros(n*M)  # DIsplacement in current turn
I = []  # Intensity of Emitted Light
X = []  # Electron Position
V = []  # DIsplacement in current turn dx
```

0.3.2 Iteration

We execute a for loop nk number of times. In each iteration, we record positions of un-collided electrons, corresponding Velocities, the Displacement it had traveled as a array. From these vectors, we find the Intensity of Light, Electron Density, Velocity of Electron at each position over the nk Iterations. The below code executes the "*For*" Loop.

```
for k in range(1, nk+1):
    ii = where(xx > 0)
    dx[ii] = u[ii]+0.5  # Displacement the electron travels
    xx[ii] = xx[ii] + dx[ii]  # Recording the New Positions
    u[ii] = u[ii]+1  # New Speed of the electron
    jj = where(xx >= n)
    xx[jj] = 0  # Ejecting the Electrons
    u[jj] = 0
    dx[jj] = 0
    kk = where(u >= u0)  # Finding Electrons that has sufficient
   Energy
    # Probability of That Collision Emitting Light
    ll = where(rand(len(kk[0])) <= p)
    kl = kk[0][ll]
    u[kl] = 0
    rho = rand(len(kl))
    xx[kl] = xx[kl]-dx[kl]*rho
    I.extend(xx[kl].tolist())  # Recording the Intensity
    m = int(rand()*Msig+M)
    empty = where(xx == 0)
    t = (min(len(empty), m))
    xx[empty[:t]] = 1
    u[empty[0][:t]] = 0
    dx[empty[0][:t]] = 0
    X.extend(xx.tolist())  # Recording the Electron Density
    V.extend(u.tolist())  # Velocity of Electrons
```

0.3.3 Histogram for Light Intensity

We plot a Histogram Graph for the Light intensity vs Position, with the list "I" obtained from the For Loop. We can see that the Light Intensity is almost constant in the middle, from the Plot. The Below code plots the Histogram. The Program also outputs the Intensity in a tabular form

```
# histogram for light intensity
figure(0)
pops, bins, temp = hist(I, bins=np.arange(
    0, n+1, 1), edgecolor='white', rwidth=1, color='black')  #
   draw histogram
xpos = 0.5*(bins[0:-1] + bins[1:])
title("Light Intensity")
xlabel(r'Position$\rightarrow$')
ylabel(r'Intensity$\rightarrow$')
show()
```

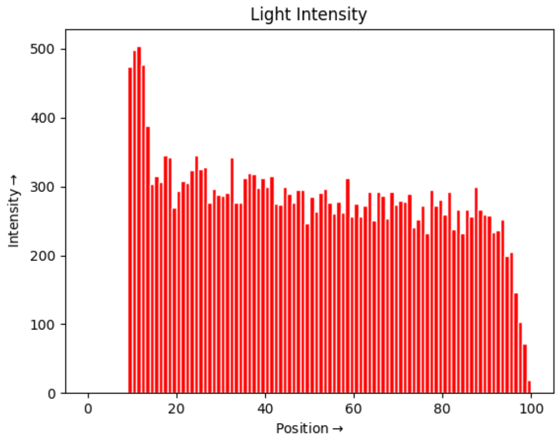

Figure 1

0.3.4 Histogram for Electron Density

We plot a Histogram Graph for the Electron Density vs Position, with the list "X" obtained from the For Loop. We can see that the Electron Density is almost constant in the middle, from the Plot. There is also a peak at the start since electrons are injected at n=0 The Below code plots the Histogram.

```
# histogram for electron density
figure(1)
hist(X, bins=np.arange(0, n + 1, 1), edgecolor='white', rwidth=1,
    color='red')
title("Electron Density")
xlabel(r'Position$\rightarrow$')
ylabel(r'Number of Electrons$\rightarrow$')
show()
```

0.3.5 Phase Space Diagram

For each Electron, we plot its corresponding position and Velocity to get the Phase space diagram. From the Plot we can see that the electron velocity increases as "n" increases, but the electron Density decreases. The below plots the phase space.

```
# phase space diagram
figure(2)
plot(X, V, 'o', color='red')
title("Electron Phase Space")
xlabel(r'Position$\rightarrow$')
ylabel(r'Velocity$\rightarrow$')
show()
```

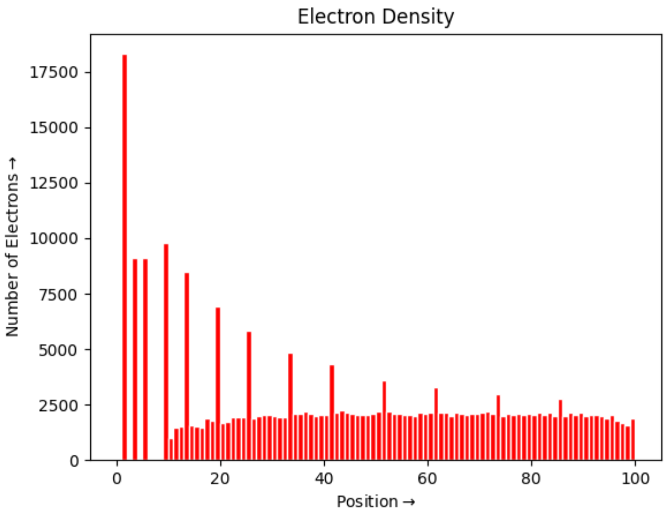

Figure 2

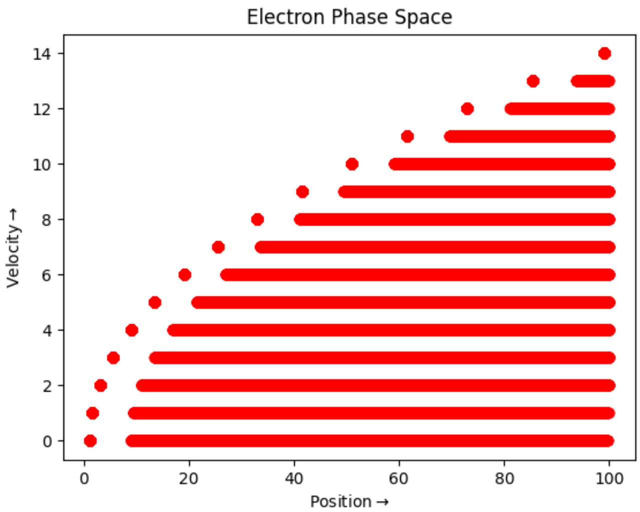

Figure 3

0.4 Conclusion

The Tube Light Model was simulated using Python. We learnt how to Plot Population Plots in Python.