EE2703: Assignment 6

KORRA SRIKANTH
EE19B033

April 29, 2021

Introduction

In this assignment, we model a tubelight as a one dimensional space of gas in which electrons are continually injected at the cathode and accelerated towards the anode by a constant electric field. The electrons can ionize material atoms if they achieve a velocity greater than some threshold, leading to an emission of a photon. This ionization is modeled as a random process. The tubelight is simulated for a certain number of timesteps from an initial state of having no electrons. The results obtained are plotted and studied.

Defining and Initializing the parameters:

The tubelight is simulated with the default parameters of $n = 100$, $M = 5$, $nk = 500$ and $Msig = 0.2$. A threshold speed is $uo = 7$, and an ionization probability is $p = 0.5$ are chosen.

Code:

A function to simulate the tubelight given certain parameters is written below:

```
 if (len(sys.argv)== 7) :
        n = int(sys.argv[1])          # Length of the tube-light.
        M = int(sys.argv[2])          # Number of Electrons injecter per turn.
        nk= int(sys.argv[3])          # Number of turns of simulation.
        uo= int(sys.argv[4])          # Threshold Velocity of Electrons.
        p = float(sys.argv[5])        # Probability of Ionization.
        Msig = float(sys.argv[6])     # Std-deviation of 'M'.
        print("Using user provided Arguments!!")
else :      # Default Values :
        n = 100
        M = 5
        nk= 500
        uo= 7
        p = 0.5
        Msig = 0.2
        print("Using Default Arguments!!")

# Initializing the vectors to zeros:
xx = np.zeros(n*M)
u  = np.zeros(n*M)
dx = np.zeros(n*M)
# Creating Three empty lists:
X=[]
V=[]
```

```
I=[]
```

Tubelight model and Running the simulation

If we consider the fact that an electron will collide after a dt amount of time, and then accelerate after its collision for the remaining time period, we need to perform a more accurate update step. This is done by taking time as the uniformly distributed random variable. Say dt is a uniformly distributed random variable between 0 and 1. Then, the electron would have traveled an actual distance of dx' given by

$$dx_i' = u_i + \frac{1}{2}dt^2$$

as opposed to $dx_i = u_i + 0.5$

We update the positions of collisions using this displacement instead. We also consider that the electrons accelerate after the collision for the remaining $1 - dt$ period of time. We get the following equations for position and velocity updates:

$$dx_i'' = \frac{1}{2}(1 - dt)^2$$

$$u_{i+1} = 1 - dt$$

With the following update rule:

$$xx_{i+1} = xx_i + dx_i' + dx_i''$$

Code:

```
for k in range(1,nk) :
                ii = np.where(xx>0)[0]     # indices where electrons are present.
                dx[ii] = u[ii] + 0.5       # updating the displacement.
                xx[ii] = xx[ii] + dx[ii]   # updating the position.
                u[ii]  = u[ii] + 1         # updating the velocity.

                # indices of electrons which reached the anode:
                aa = np.where(xx > n)[0]
                xx[aa] = 0        # setting the position to be 0.
                u[aa] = 0         # setting the velocity to be 0.
                dx[aa] = 0        # setting the displacement to be 0.

                # indices of electrons whose velocity is greater than threshold:
                kk = np.where(u >= uo)[0]
                ll=np.where(np.random.rand(len(kk))<=p)[0]
                # kl contains the indices of electrons which undergo collisions:

                kl=kk[ll]
                # Setting the velocity of collided electrons to 0:
                u[kl] = 0

                # setting position of collided electrons:
                xx[kl] = xx[kl] - dx[kl]*np.random.rand()
                # Updating the Intensity list:
                I.extend(xx[kl].tolist())
```

```
# Injected electrons:
m = int(np.random.randn()*Msig + M)
# Empty indices in the light:
ee = np.where(xx == 0)[0]

if len(ee) >= m:
      start = np.random.randint(len(ee))
      xx[ee[start: m+start]] = 1
      u[ee[start-m:start]] = 0
else :
      xx[ee] = 1
      u[ee]  = 0
filled = np.where(xx > 0)
# Updating the Position list:
X.extend(xx[filled].tolist())
# Updating the Velocity list:
V.extend(u[filled].tolist()
```

Plots:

Using these updates, we get the following plots:

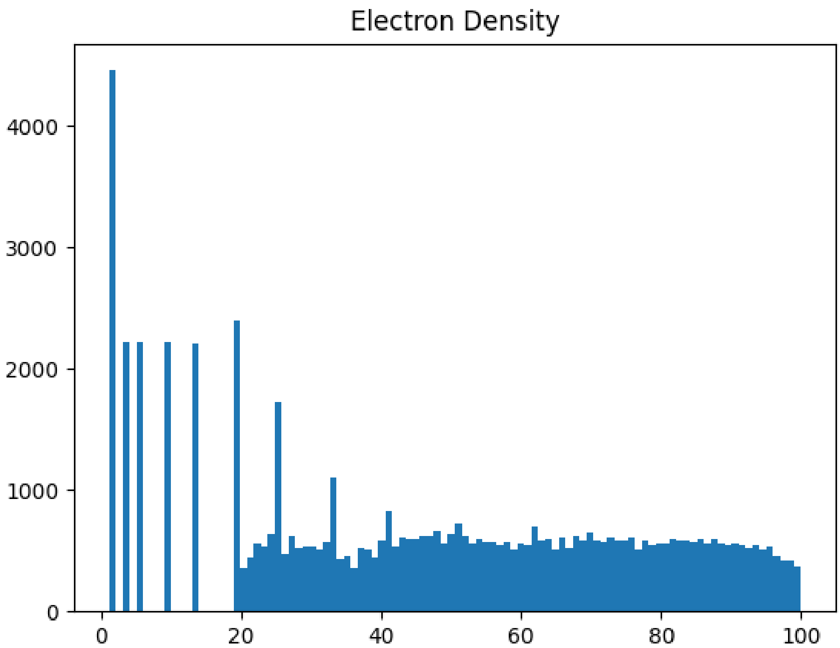

Figure 1: Electron Density Plot

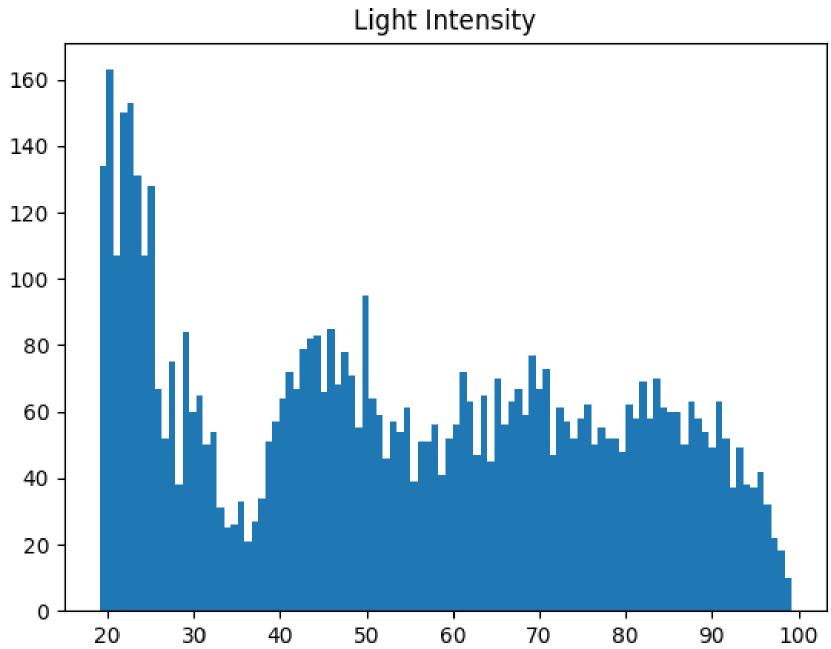

Figure 2: Emission Intensity Plot

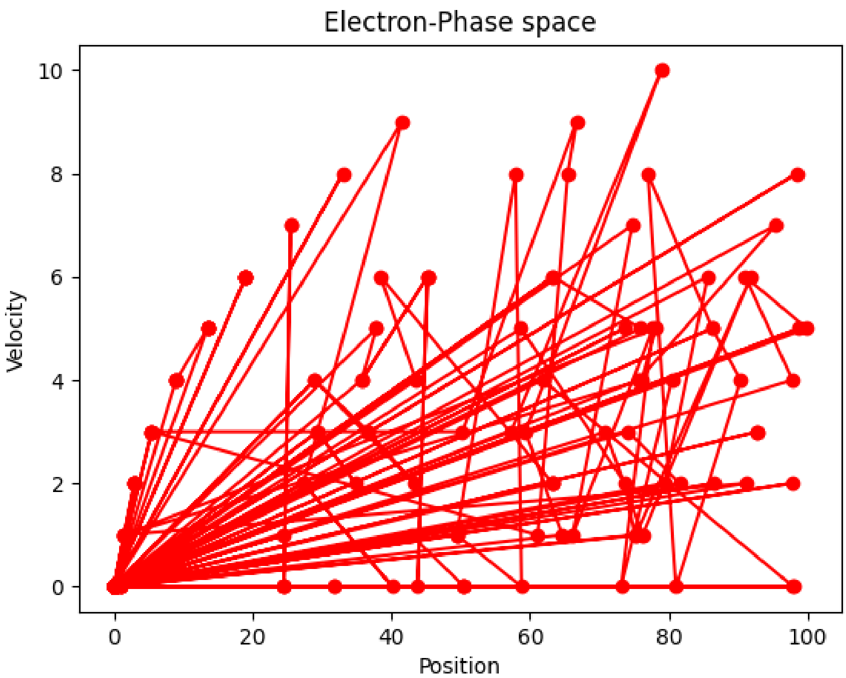

Figure 3: Electron Phase space Plot

Discussions

- The electron density is peaked at the initial parts of the tubelight as the electrons are gaining speed here and are not above the threshold. This means that the peaks are the positions of the electrons at the first few timesteps they experience.

- The peaks slowly smoothen out as x increases beyond 19. This is because the electrons achieve a threshold speed of 7 only after traversing a distance of 19 units. This means that they start ionizing the gas atoms and lose their speed due to an inelastic collision.

- The emission intensity also shows peaks which get diffused as x increases. This is due the same reason as above. Most of the electrons reach the threshold at roughly the same positions, leading to peaks in the number of photons emitted there.
- This phenomenon can also be seen in the phase space plot. Firstly, the velocities are restricted to discrete values, as the acceleration is set to 1, and we are not yet performing accurate velocity updates after collisions.
- One trajectory is separated from the rest of plot. This corresponds to those electrons which travel until the anode without suffering any inelastic collisions with gas atoms. This can be seen by noticing that the trajectory is parabolic. This means that $v = k\sqrt{x}$, which is precisely the case for a particle moving with constant acceleration.
- The rest of the plot corresponds to the trajectories of those electrons which have suffered at least one collision with an atom. Since the collisions can occur over a continuous range of positions, the trajectories encompass all possible positions after $x = 19$.

Conclusion:

- Since the threshold speed is much lower in the second set of parameters, photon emission starts occuring from a much lower value of x. This means that the electron density is more evenly spread out. It also means that the emission intensity is very smooth, and the emission peaks are very diffused.
- Since the probability of ionization is very high, total emission intensity is also relatively higher compared to the first case.
- We can conclude from the above observations that a gas which has a lower threshold velocity and a higher ionization probability is better suited for use in a tubelight, as it provides more uniform and a higher amount of photon emission intensity.
- Coming to the case where the ionization probability is 1, we observe that the emission instensity consists of distinct peaks. The reason that these peaks are diffused is that we perform the actual collision at some time instant within the interval between two time steps. This also explains the slightly diffused phase plot as well.