Assignment No 6

Pranav Phatak, EE19B105

April 8, 2021

Objective

The aim of this assignment is to simulate the functioning of a tubelight by splitting into 100 sections position wise and letting it run for a time period of 500 iterations.

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.

Simulation

The simulation for certain given variables is done using the following code:

```
xx = np.zeros(n*M)      #Electron position
u = np.zeros(n*M)       #Electron velocity
dx = np.zeros(n*M)      #Electron displacement per time step

I = []          #Stores value of intensity of emitted light at every time-step
V = []          #Stores value of electron velocity at every time-step
X = []          #Stores value of electron position at every time-step


for i in range(1,nk):
    ii = np.where(xx>0)[0]

    dx[ii] = u[ii] + 0.5
```

```
xx[ii] = xx[ii] + dx[ii]
u[ii] = u[ii] + 1.0

npos = np.where(xx>n)[0]
xx[npos] = 0.0
dx[npos] = 0.0
u[npos] = 0.0

kk=np.where(u>=u0)[0]
# array which gives indices of electrons which can cause collision
ll=np.where(rand(len(kk))<=p)[0]
kl=kk[ll]
# array to store indices of electrons which collided to give a photon

P = np.random.rand(len(kl))
# value to reset position by when collision occurs
xx[kl] = xx[kl]-np.multiply(dx[kl],P)
# position reset by small factor since collision can take place at any time
u[kl] = 0
# inelastic collision implies velocity reset to 0

I.extend(xx[kl].tolist())

m=np.random.randn()*Msig+M
empty_xpos = np.where(xx==0)[0]
electrons_generated = min(len(empty_xpos),(int)(m))
xx[empty_xpos[0:electrons_generated]] = 1.0
u[empty_xpos[0:electrons_generated]] = 0.0

X.extend(xx[ii].tolist())
V.extend(u[ii].tolist())
```

Plots

Case i) M = 5 , Msig = 1 , u0 = 5, p = 0.25

The 3 plots for the following values are as follows:

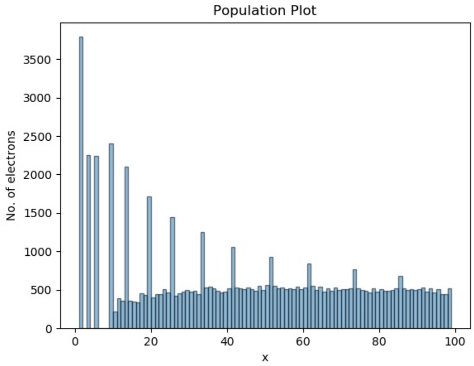

Figure 1: Population plot of electron density

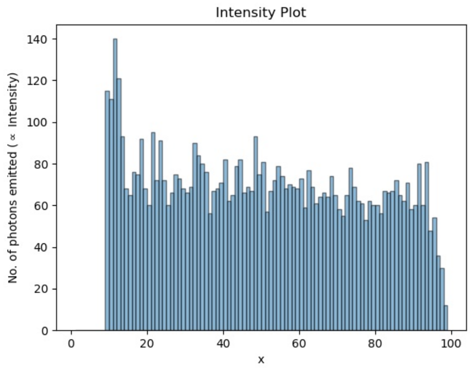

Figure 2: Population plot of Intensity

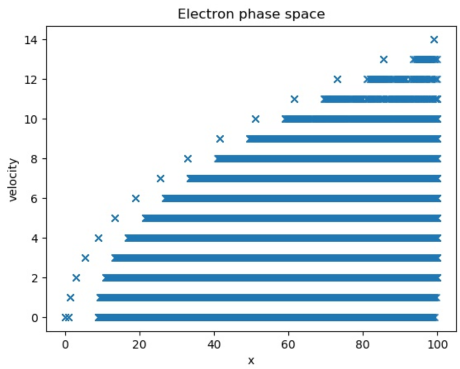

Figure 3: Electron Phase Space

Case ii) M = 10 , Msig = 2 , u0 = 7 , p = 0.5
The 3 plots for the following values are as follows:

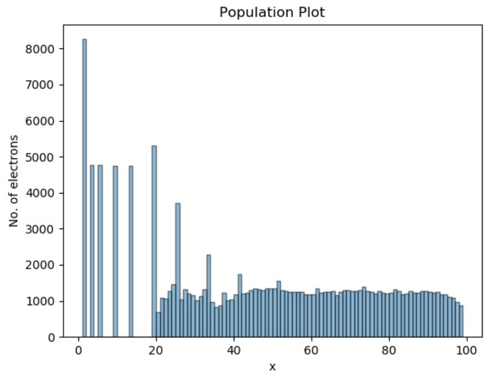

Figure 4: Population plot of electron density

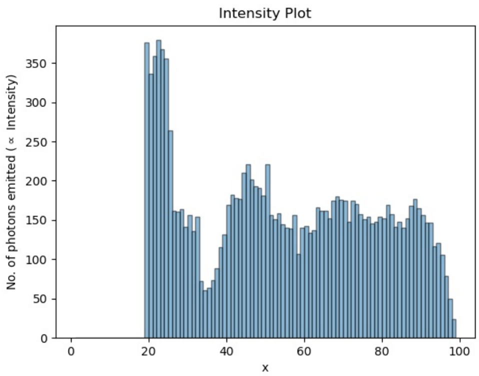

Figure 5: Population plot of Intensity

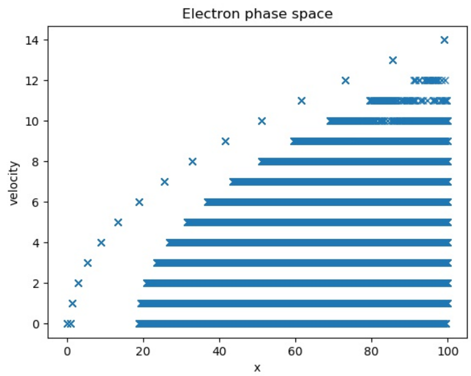

Figure 6: Electron Phase Space

Tabular Data

We use the following code lines to get the tabular data.

```
# Tabulating data for intensity vs position
bins = plt.hist(I,bins=np.arange(1,n,n/100))[1]
# Bin positions are obtained
count = plt.hist(I,bins=np.arange(1,n,n/100))[0]
# Population counts obtained
xpos = 0.5*(bins[0:-1] + bins[1:])
df = pd.DataFrame()
# A pandas dataframe is initialized to do the tabular plotting of values.
df['Xpos'] = xpos
df['count'] = count

base_filename = 'values.txt'
with open(base_filename,'w') as outfile:
    df.to_string(outfile)
```

The tabular data for xpos vs count is as follows.
Case i) M=5 , Msig=1 , u0=5 , p=0.25
The significant data for the same is:

	Xpos	count
7	8.5	0.0
8	9.5	130.0
9	10.5	128.0
10	11.5	128.0
11	12.5	111.0
12	13.5	90.0
..	....	....
91	92.5	57.0
92	93.5	56.0
93	94.5	56.0
94	95.5	46.0
95	96.5	44.0
96	97.5	31.0
97	98.5	19.0

Case ii) M=10 , Msig=2 , u0=7 , p=0.5
The significant data for the same is as follows:

	Xpos	count
17	18.5	0.0
18	19.5	376.0

19	20.5	336.0
20	21.5	359.0
..	....	.....
36	37.5	89.0
37	38.5	115.0
38	39.5	132.0
39	40.5	169.0
40	41.5	182.0
41	42.5	178.0
42	43.5	177.0
43	44.5	210.0
44	45.5	221.0
..	....	.....
93	94.5	121.0
94	95.5	106.0
95	96.5	79.0
96	97.5	50.0
97	98.5	24.0

Experimenting

It is my claim that the peak of Intensity is proportional to u0. It was seen so from the previous 2 experiments, in the first one we took u0 = 5, we got the first peak around 9.5, in the second one we took u0 = 7 and the peak went to around 19.5. It is also clear that as p increases, the peak should increase its magnitude since more collisions imply more intensity

To see these effects, we will now decrease u0 to 2, and increase p to 0.9. If the claims are correct, the peak should come closer to the origin with a larger magnitude

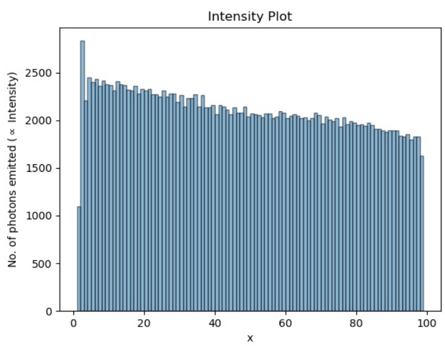

Figure 7: Population plot of intensity

We shall repeat this experiment with u0 = 10 and p = 0.2

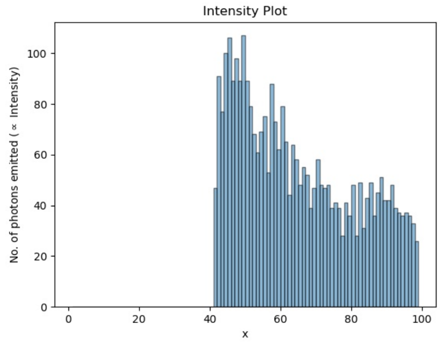

Figure 8: Population plot of intensity

It is clear from the above graphs that my claims were correct on how these 2 variables affect the Intensity of light emitted by a tubelight.

Conclusions

i)Since the threshold speed is much lower in the first 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.

ii)Since the probability of ionization is very high, total emission intensity is also relatively higher in the second case compared to the first case.

iii)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.

Remarks

In the attached code that I have written, I am getting 4 outputs, 3 graphs and a table printed to a text file. While making the report I have compiled the code several times with different variable values. In the attached zip file I have attached only the images and latex code which I used to form this document and not the general outputs of the code. The values.txt file has the data for a simulation of Case (i) as done in the plots and tabular data sections.

Note that: Since this is a random simulation, upon compiling the code, each time the user will get slightly different values but similar trends will be followed more or less for all the outputs.