Assignment 5

Lakshya J ee19b035

April 8th, 2021

Introduction

In this assignment I will be attempting to simulate a tube light. I start off by accepting arguments by the user. If nothing is entered, default values are assumed. For the sake of readability, the values are accepted only in a particular format.
I have assumed the standard deviation to be 2 in any given case.
Here is the list of default values.

```
n=100
M=5
nk=500
u0=5
p=0.25
Msig=2
```

We are asked to initialize the vectors with values 0. Here is how I did that:

```
# Electron position
xx = np.zeros(n*M)
# Electron velocity
u = np.zeros(n*M)
# Displacement in current turn
dx = np.zeros(n*M)
```

Simulation

Now I begin with the for loop for the simulation. I run the loop nk times, since that is the default number. Given below is how I update the values of the vectors initialized as zero in the loop.

```
# Changing position due to acceleration
dx[ii] = u[ii]+0.5
# Changing position along x axis of accelerated electrons
xx[ii] = dx[ii] + xx[ii]
# Changing velocity of the electrons
u[ii] = u[ii] + np.ones(len(ii))
```

Now I update the values of I, V and X according to the values obtained in the particular iteration by updating ii in the end. I do that by initializing ii

outside the loop. I generate a uniform random value ρ to be multiplied with dx and change the positions of the electrons at the indices in kl. After adding the updated array to the list I I declare a random number m to fill the first m empty slots. However I make sure that m is lesser than the number of empty slots available by using the $min()$ function.

```
# kl contains indices of the electrons which now suffer inelastic collisions
kl = treshvel[ll]
# Setting all the values to 0
u[kl] = 0
# Changing position of the collided electron to a random position rho = random.uniform(0, 1)
# Multiplying any random number rho generated uniformly between 0 and 1
xx[kl] = xx[kl] - rho*dx[kl]
# Excited atoms result in photon emission, so we have to add a photon at that point
I.extend((xx[kl].T).tolist())
m = int(np.random.randn()*Msig+M)
# Checking where the slots are empty
empty = np.where(xx==0)[0]
# Filling the slots only such that the number of randomly selected slots is lesser than the number of empty slots
m = min(m, len(empty))
# Inititializing the values of the newly introduced electrons as 1
xx[empty[:m]] = 1
# Initializing other values at these slots as 0
u[empty[:m]] = 0
dx[empty[:m]] = 0
```

This is how I update $treshvel$ and ll. As sir as mentioned, I find the indices only where the probability of the electron excitation is lesser than p.

```
# Checking which velocities are above treshold
treshvel = np.where(u>u0)[0]
# Out of the excited ones only <=p of them are ionized
ll = np.where(np.random.rand(len(treshvel))<=p)[0]
```

At the end of the loop I update the value of ii so that it can be used in the beginning of each loop. I also appropriately append the arrays xx and u to X and V respectively.

```
# Resetting the values of ii at the end of the loop to be used in next loop
ii = np.where(xx>0)
# Adding new values to final X and V vectors
X.extend(xx[ii].tolist())
V.extend(u[ii].tolist())
```

Plots

Here are the plots I obtained for the simulation for the default values provided.

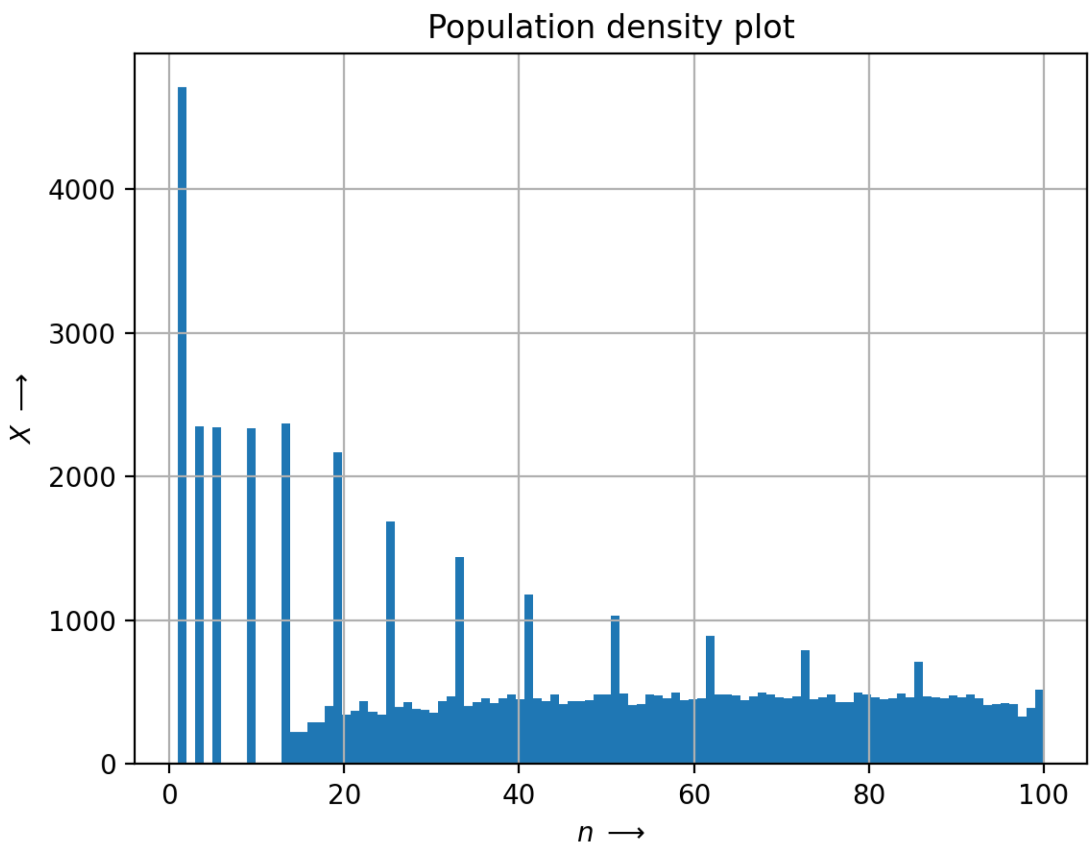

Figure 1: Population density plot

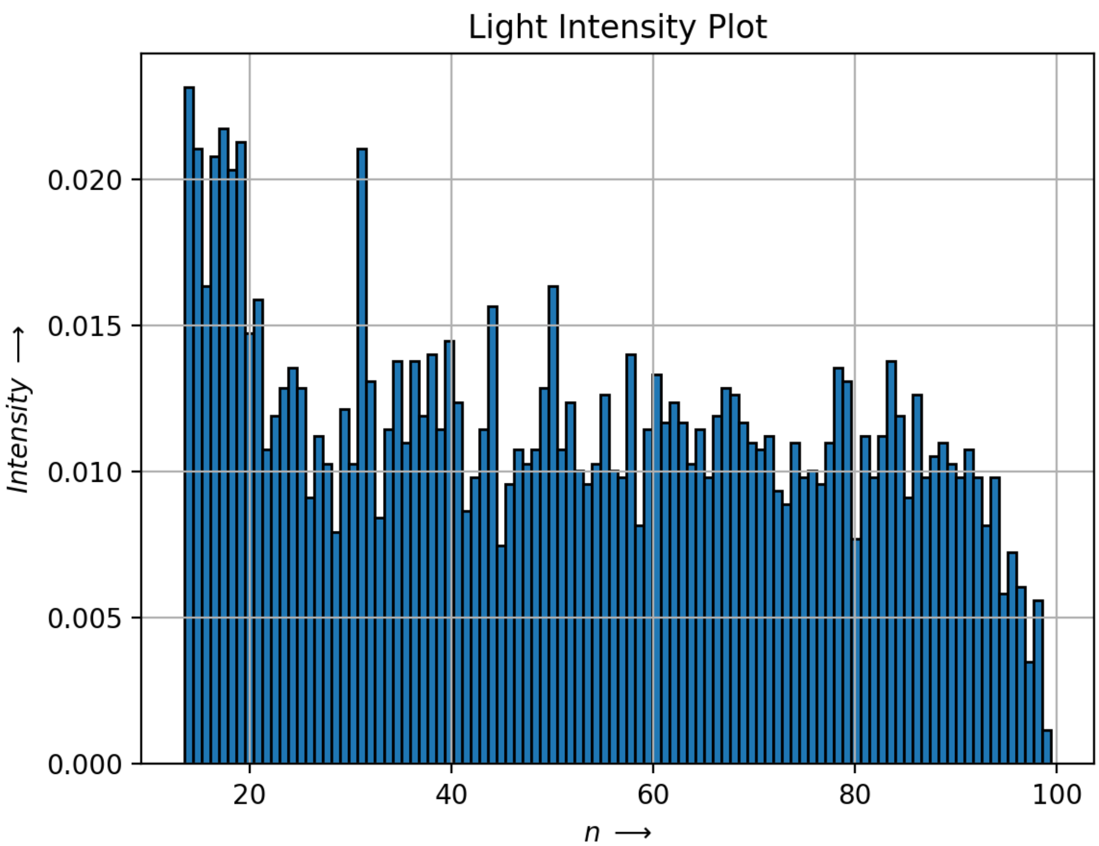

Figure 2: Light Intensity plot

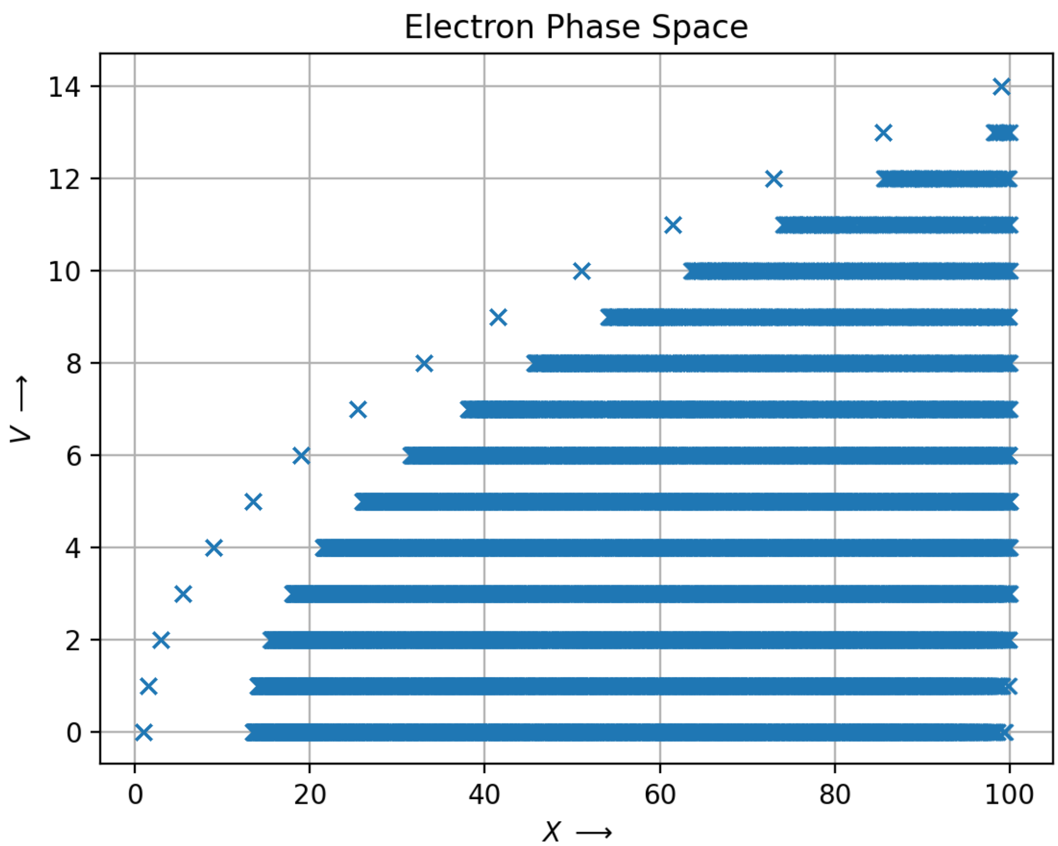

Figure 3: Phase space plot

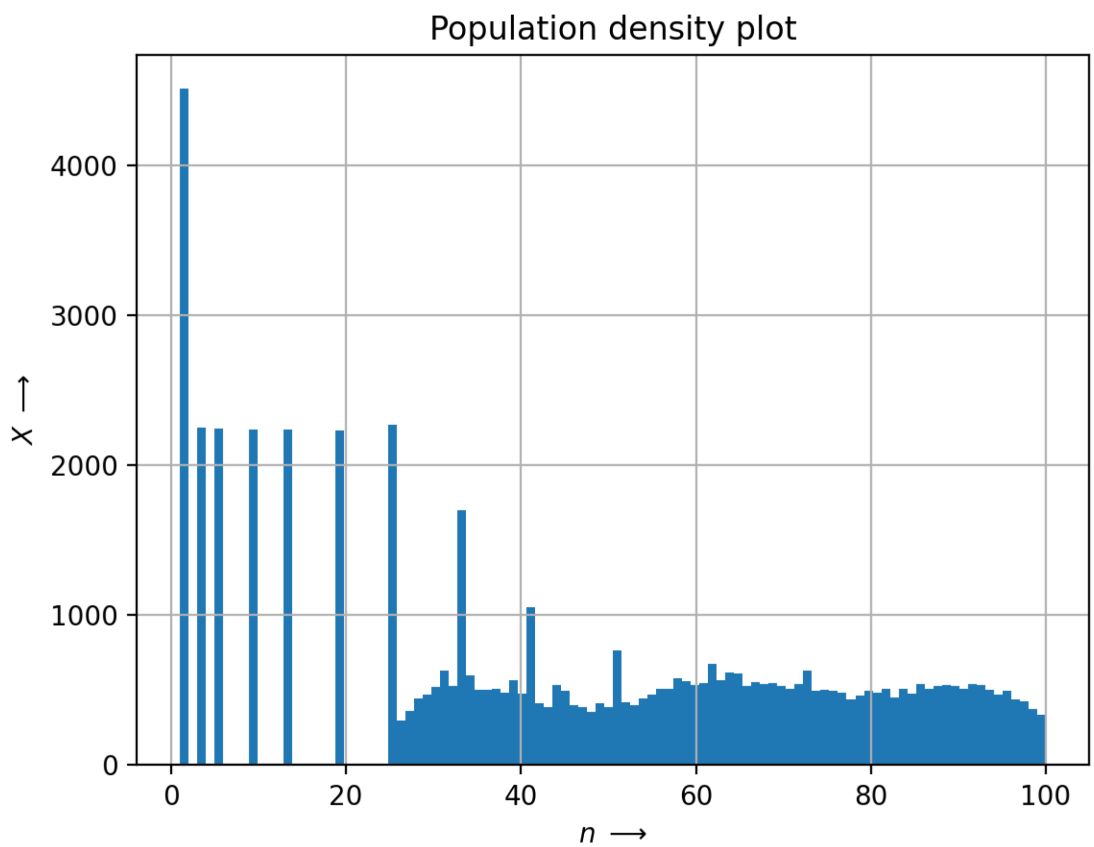

Figure 4: Population density plot

Here are the plots I obtained for $u_o = 7$ and $p = 0.5$.

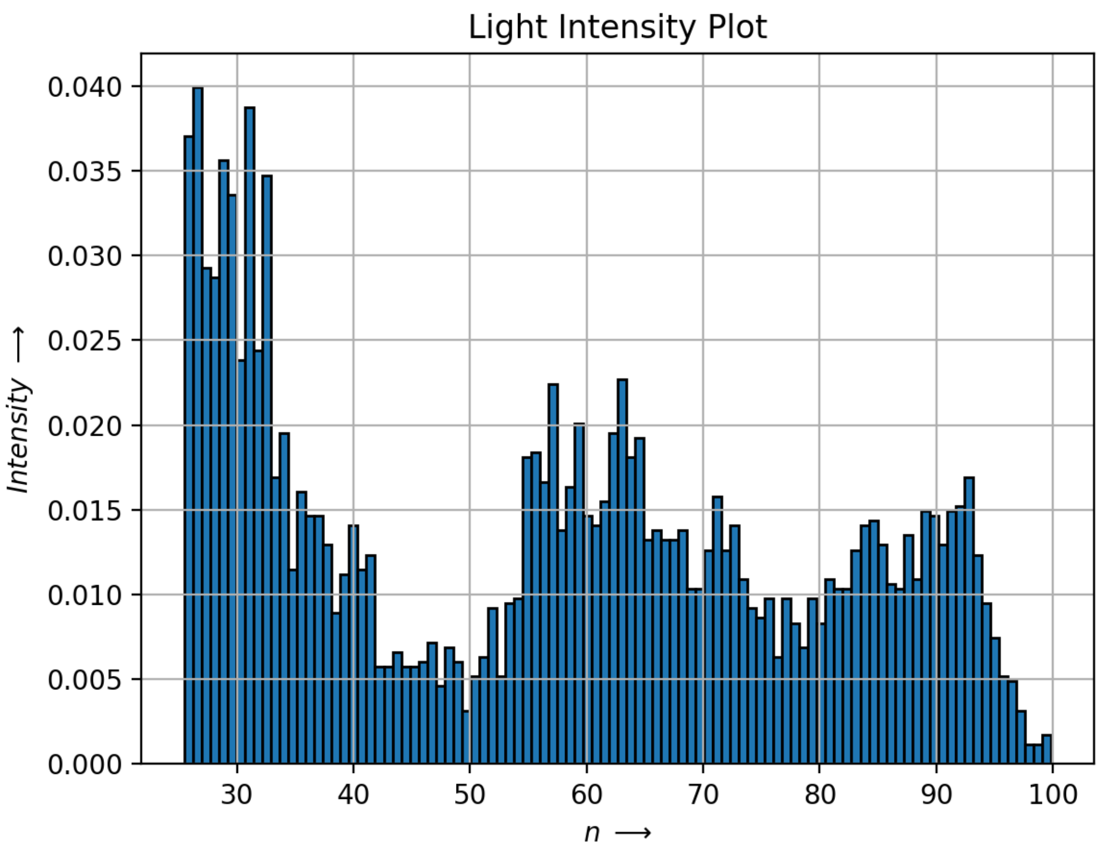

Figure 5: Light Intensity plot

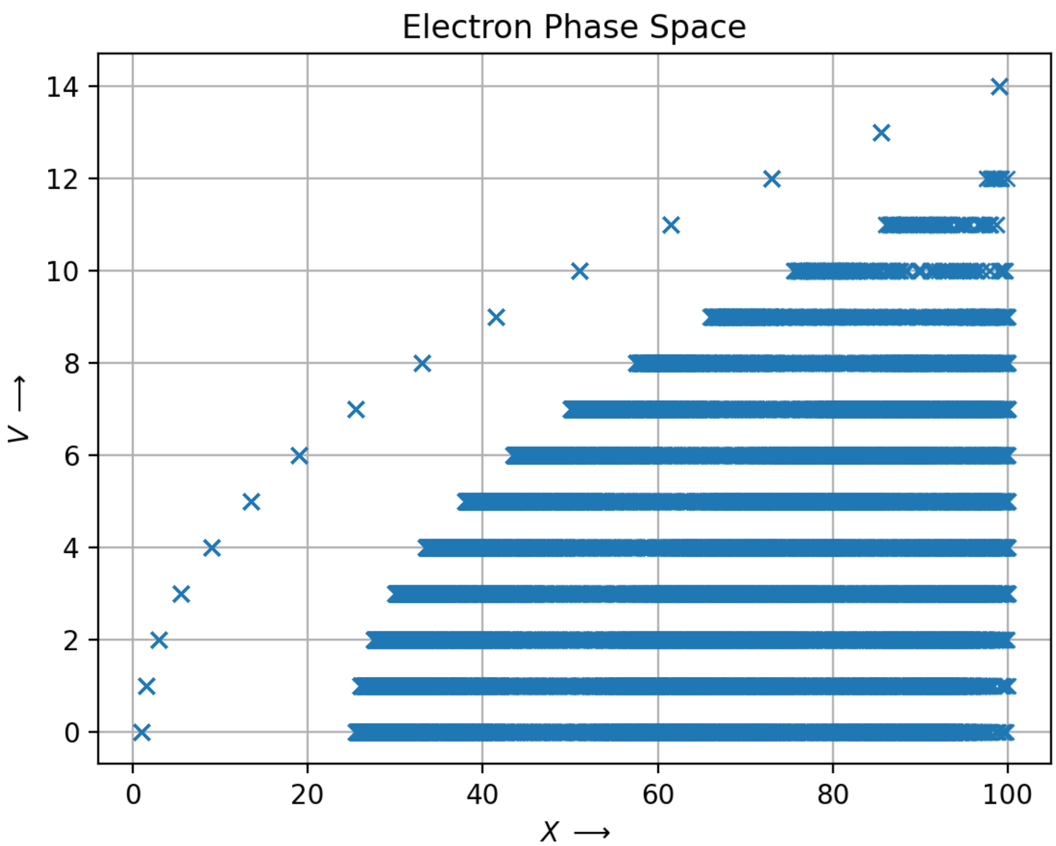

Figure 6: Phase space plot

```
zsh
lakshyajagadish@Lakshyas-MacBook-Pro Assignment6 % python3 EE19B035_Lakshya_Assi
gnment6.py
Default values have been assigned
Intensity Data
xpos    count
13.953377       0.023152
14.812212       0.021047
15.671046       0.016370
16.529881       0.020813
17.388716       0.021749
18.247551       0.020345
19.106385       0.021281
19.965220       0.014733
20.824055       0.015902
21.682890       0.010757
22.541724       0.011927
23.400559       0.012862
24.259394       0.013564
25.118229       0.012862
25.977063       0.009120
26.835898       0.011225
27.694733       0.010290
28.553568       0.007951
29.412403       0.012161
30.271237       0.010290
```

Figure 7: Terminal output for default values

Here is the terminal output I obtained for the default values. I have plotted the density histogram and hence in the count column all these values are reflected as fractions.

Conclusion

It can be seen that the phase space plot has a parabolic relationship in line with the equation of motion s under constant acceleration which we have used. The light intensity can be seen to be observed after a certain values of x, which coincides with the point from which electrons are found in the phase space plot as well. It then reaches a sort of a local minima and goes on increasing after that point.
From the population density plots it can be noticed that the electrons are closely packed after the x from which the light intensity begins and their population can be considered to be almost a constant. It reached a peak right at the beginning at around the 0th mark.