EE2703 : Applied Programming Lab
Assignment 6
Tubelight Model

Adityan C G
EE19B003

April 7 2021

1 Introduction

This Assignment is about creating an ideal model of a tubelight and plotting the main features like the electron flow and Collision intensity etc. based on the info collected from the user.

2 Parameters of the underlying problem

We define Parameters like,

```
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 #Std dev of Injected electrons at a time
```

After updating user inputs to these default values if any, using sys.argv. We define empty arrays of dimensions nM,

- xx - Position of electron inside tube light
- u - Velocity of electron
- dx - Displacement of electron

And we also define empty lists to keep account of the Positions after every time unit and Intensity of the Collisions and Velocities to make histograms at the end and analyse the progress of electrons.

```
X = []
V = []
I = []
```

3 Iterating for loop

In this part of the code we Accelerate the electrons that enter the tube light(Using a normal distribution) and once it reaches the threshold velocity, calculating the (random)probable electrons that goes through collision and also calculating the approximate position at which the collision happens.

```
    ii = where(xx>0)

#Updating Position, Dispalcement and Velocity
dx[ii] = u[ii] + 0.5
xx[ii] += dx[ii]
u[ii] += 1

#Finding Positions of electrons with postion above n and collision velocity
iix = where(xx > n)
kk = where(u >= u0)

#assigning zero parameters for passed out electrons
xx[iix] = 0
u[iix] = 0
dx[iix] = 0

#Getting the no. of electrons that will actually collide and lose energy by probability provided
ll = np.where(rand(len(kk[0])) <= p)
#Getting the indices
kl = kk[0][ll]
#Zero velocity after collision
u[kl] = 0

#Postion of collision that happened, using a random no.
rho = rand(len(kl))
xx[kl] = xx[kl] - dx[kl]*rho

#Intensity
I.extend(xx[kl].tolist())
```

Now we need to fill in newly inserted electrons based on the spaces available. If the no.of spaces available are more than the probable electrons to enter then there is no mismatch. However is there is less space then all the spaces will be filled.

```
    i0 = where(xx == 0) #Empty spaces
nv=(min(n*M-len(i0),m)) #if no empty spaces are left
xx[i0[:nv]]=1  #inject the new electrons
u[i0[:nv]]=0  #with velocity zero
dx[i0[:nv]]=0 #and displacement zero
```

```
#Add to velocities and Positions
X.extend(xx.tolist())
V.extend(u.tolist())
```

Also the positions and the velocities are updated in the array.

4 Plots required

The first histogram plot required is the "Electron Density". This is done through a population plot of the list X. The second plot is the "Light Intensity", population plot of I.

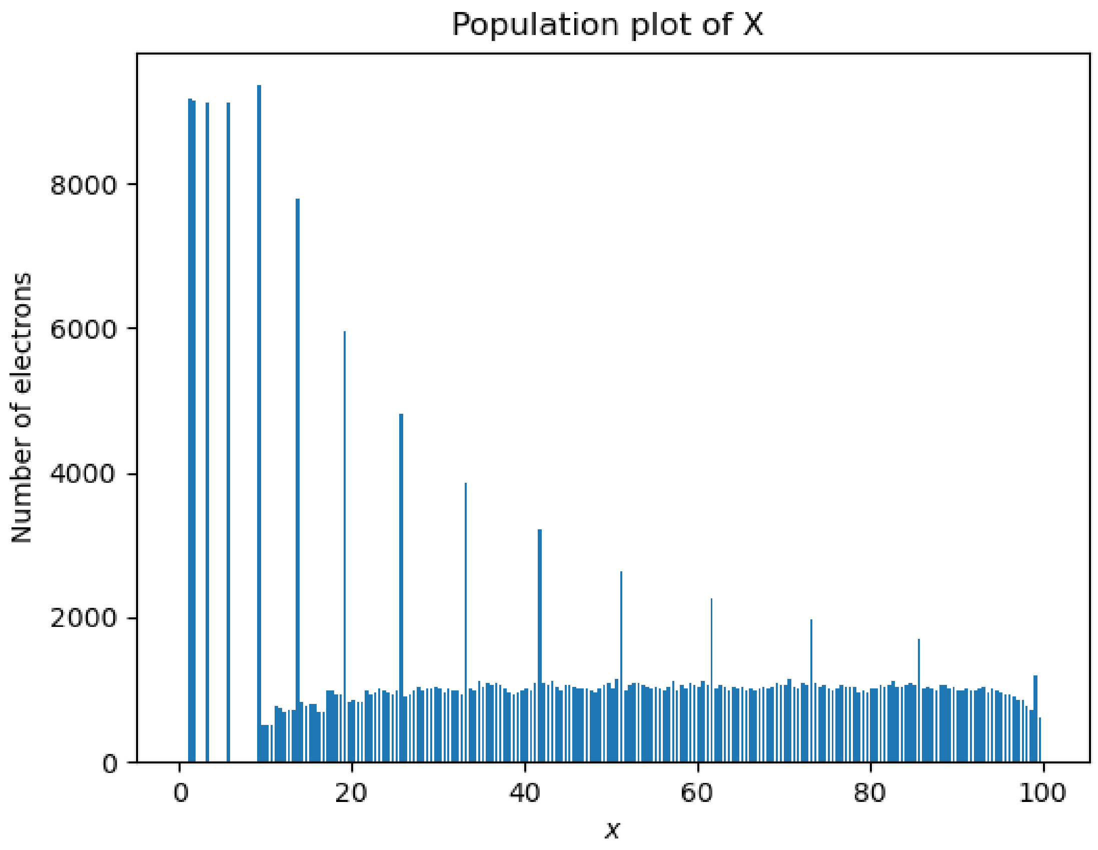


Figure 1

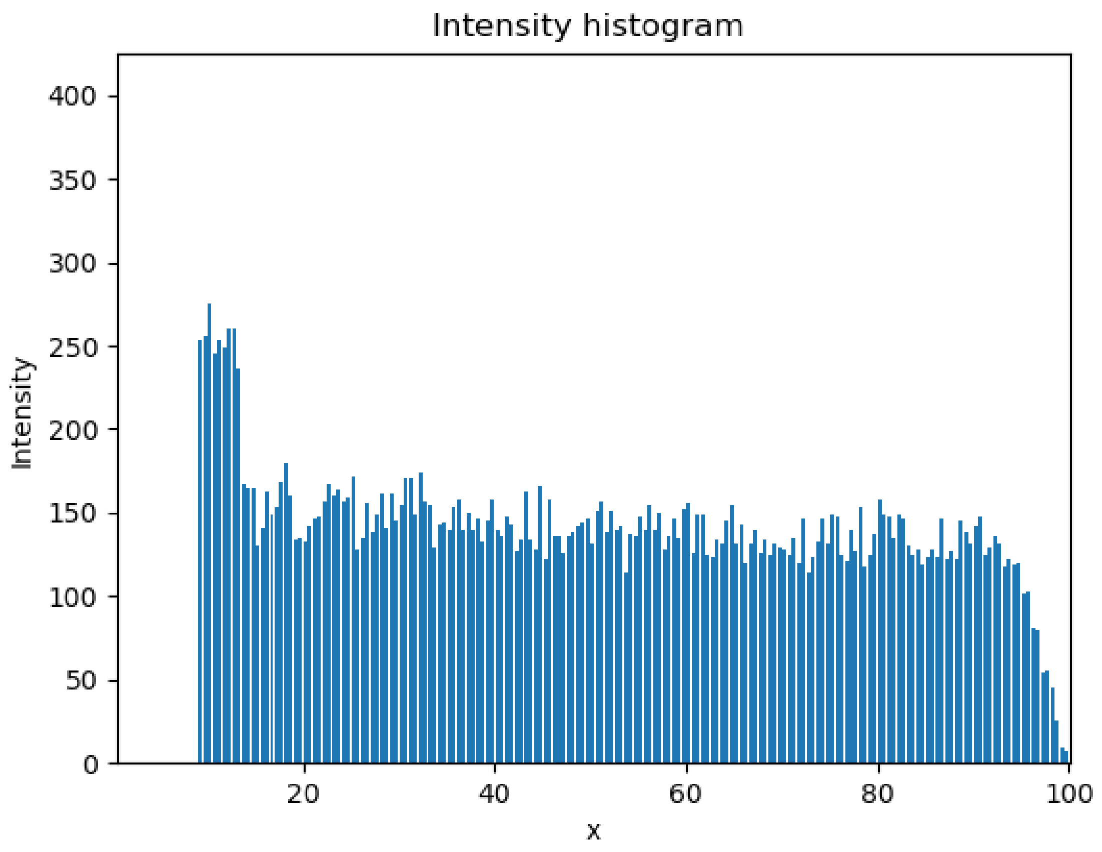
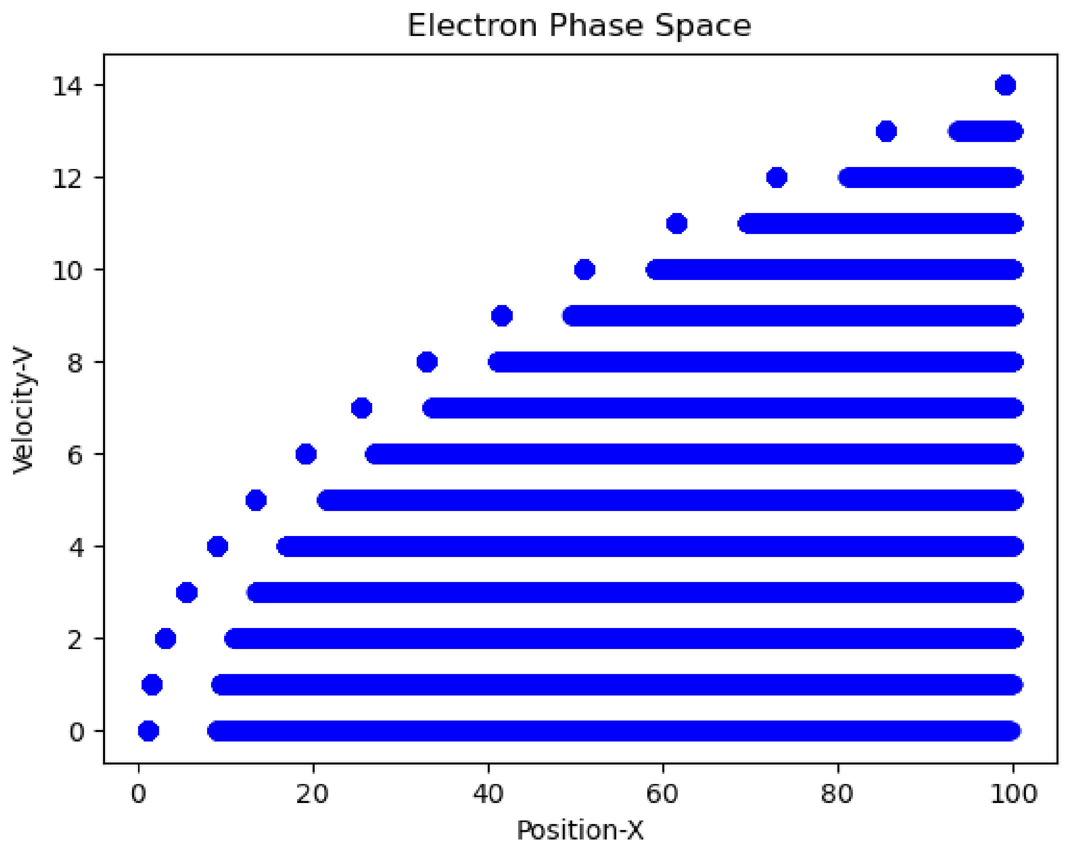


Figure 2

Finally, "Electron Phase space" plot is required. Before that the intensity data is obtained from the ****hist**** plot of Intensity needs to be printed

```
y = histogram[0]
bins = histogram[1]
xpos = 0.5*(bins[1:-2][::2]+bins[3:][::2])
print("\t\t\t\tIntensity Data")
data = c_[xpos,y[1:][::2]]
df = pd.DataFrame(data , columns=['Xposition','Count'])
print(df)
```



5 Using different distributipn to determine position of collision

Now for a better distribution to calculate the exact position of collision, we can us Maxwell boltzman distribution, or Quadratic Probability distribution.Both the corresponding plots have been shown.

(Next Page)

```
    import scipy
    x = len(kl)
    #For Maxwell Boltzman distribution
rho = scipy.stats.maxwell.rvs(size=x)
#For quadratic distribution
rho = power(0.5,x)
```

5.1 Maxwell distribution plots

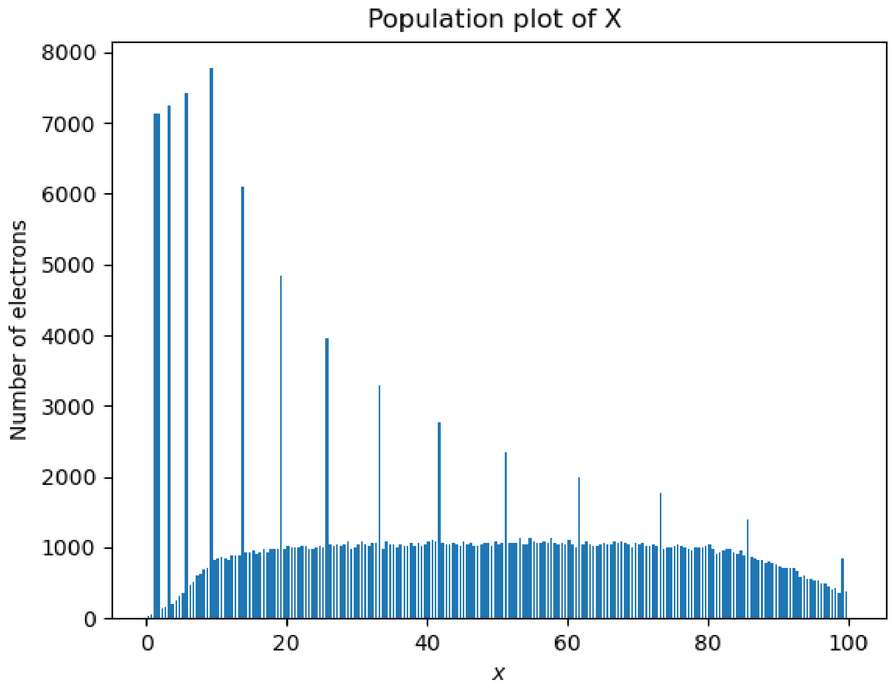


Figure 3

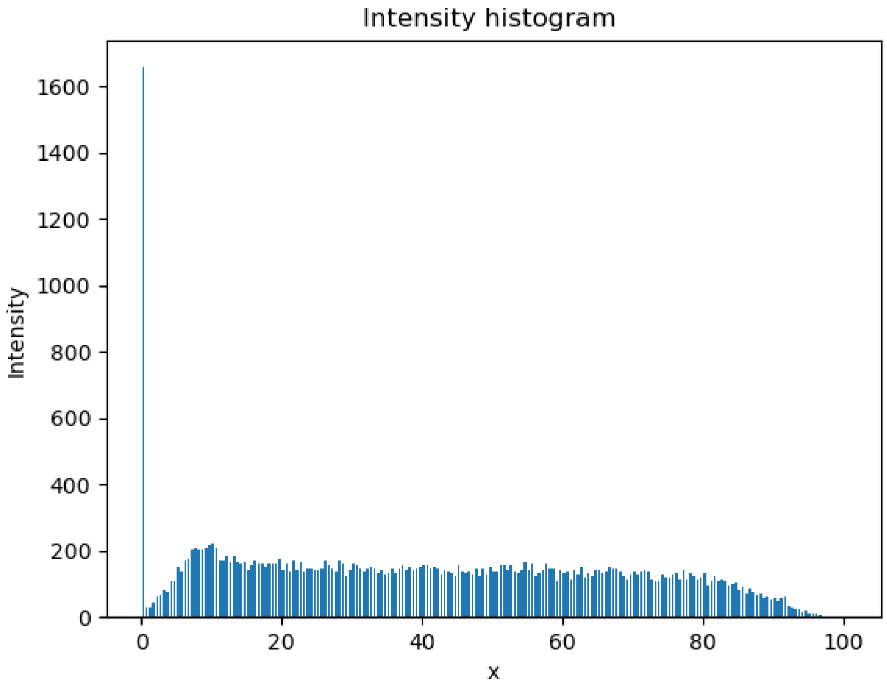


Figure 4

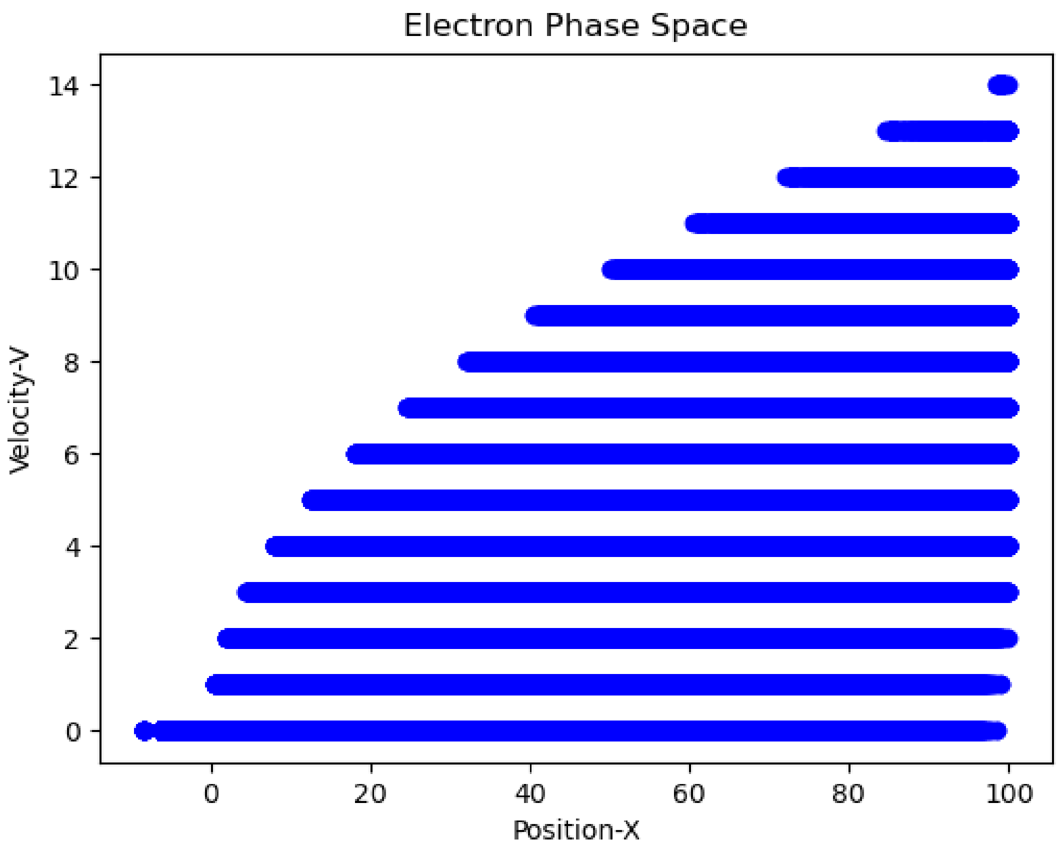

Electron Phase Space
Velocity-V
Position-X
0
2
4
6
8
10
12
14
0
20
40
60
80
100

Figure 5

5.2 Quadratic distribution plots

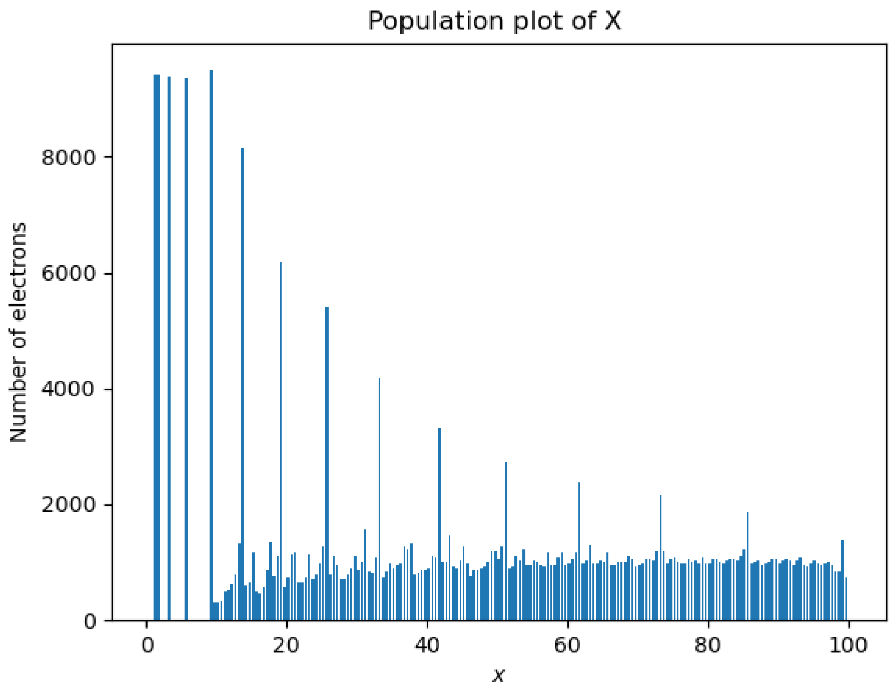


Figure 6

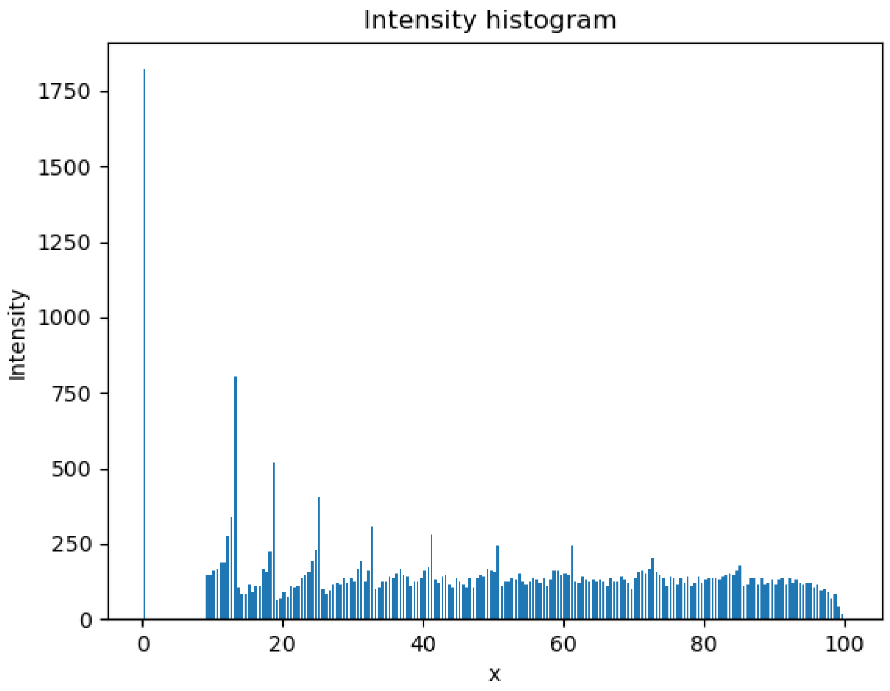


Figure 7

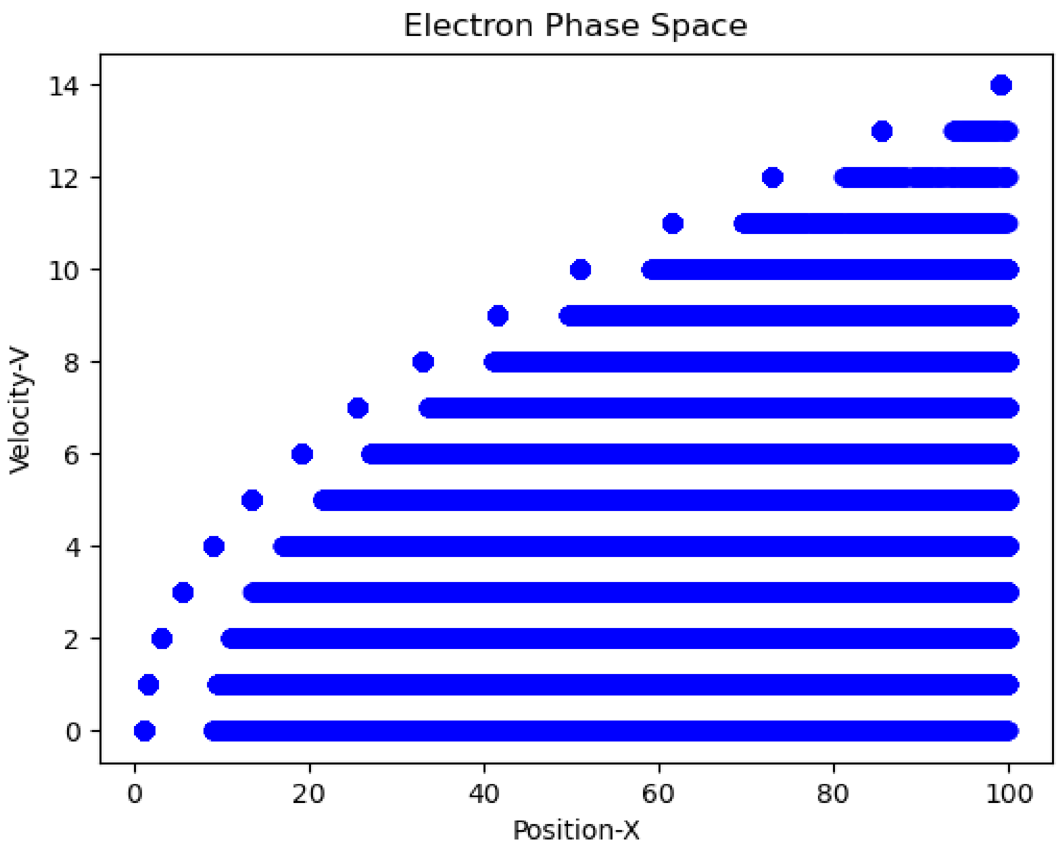


Figure 8

6 Conclusion

The Intensity Histogram shows us the collisions take place well after the initial positions of the tube light.The region upto 10 is where electrons are building up their energy. Beyond that is a region where the emission decays, representing the fewer energetic electrons that reached there before colliding. At 40 is the next peak. But this is a diffuse peak since the zero energy location of different electrons is different.A fairly complex simulation, yet it was done in just a few lines of Python.