Assignment 6: Simulations

Saurav Sachin Kale, EE19B141

April 11, 2021

1 Aim

- To simulate a tube light with a one dimensional model
- To plot the light intensity as a function of position at steady state and identify "dark spaces"
- To plot electron density vs. position
- To plot the phase space diagram of the electrons

2 Procedure

- We first initialize the simulation universe. Our simulation depends on spatial grid size n, number of electrons injected per turn `M`, number of turns to simulation `nk`, the threshold velocity `u0` and the probability of ionisation p The default values are mentioned below. If one wishes to test other values, one can enter them as space separated command line arguments in the same order as below. (See usage instructions in code)

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

- We initialize the electron position `xx`, electron velocity `u`, and displacement in current turn `dx` as one dimensional Numpy arrays of length nM. We also initialize intensity of emitted light `I`, electron position `X`, electron velocity `V`.

```
xx = np.zeros(n*M)
u = np.zeros(n*M)
dx = np.zeros(n*M)
I = []
X = []
V = []
```

- Now we loop nk times, and in each loop we perform the following operations:

1. Find those indices where electrons are present

```
ii = np.where(xx > 0)[0]
```

2. Calculate the change in position as follows: $dx_i = u_i \Delta t + \frac{1}{2} a (\Delta t)^2 = u_i + 0.5$

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

3. Add the change in position to the current position: $x_i \leftarrow x_i + dx_i$

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

4. Update the velocity $u_i \leftarrow u_i + 1$

```
u[ii] = u[ii] + 1
```

5. Set the position and velocity of electrons which reached the anode to zero

```
reachedAnode = np.where(xx > n)[0]
xx[reachedAnode] = 0
u[reachedAnode] = 0
dx[reachedAnode] = 0
```

6. Out of those electrons whose velocity is above threshold velocity, we select a uniformly distributed normal distribution whose indices are less than probability p. These are the electrons which collided. We now set their velocities to zero.

```
kk = np.where(u >= u0)[0]
ll = np.where(np.random.rand(len(kk)) <= p)[0]
kl = kk[ll]
u[kl] = 0
```

7. We obtain the actual point of collision and update the `xx` array like so: $x_i \leftarrow x_i - dx_i\rho$ where ρ is a random number between 0 and 1.

```
xx[kl] = xx[kl] - dx[kl]*np.random.rand()
```

8. We now know which electrons have collided. These electrons will excite atoms that emit the photons. So we add those positions to our list I.

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

9. The number of electrons injected is $m = randn() * \sigma_M + M$, where σ_M is standard deviation in M. We now find all the empty slots in xx, and then compare it with m. If the empty slots available are more than m, then we pick the first m empty slots. If $m >$ number of empty slots, then we take the available slots. We then inject electrons in these slots.

```
m = int(plab.randn()*Msig + M)
emptySlots = np.where(xx==0)[0]

if m > len(emptySlots):
        xx[emptySlots] = 1
        u[emptySlots] = 0
else:
        xx[emptySlots[:m]] = 1
        u[emptySlots[:m]] = 0
```

10. We append the current position and velocity of the electrons in lists X and V.

```
ii2 = np.where(xx > 0)[0]
X.extend(xx[ii2].tolist())
V.extend(u[ii2].tolist())
```

- We now generate histograms for I, X and phase space diagram X and V.

3 Results for inaccurate distance update

1. Histogram for Light Intensity

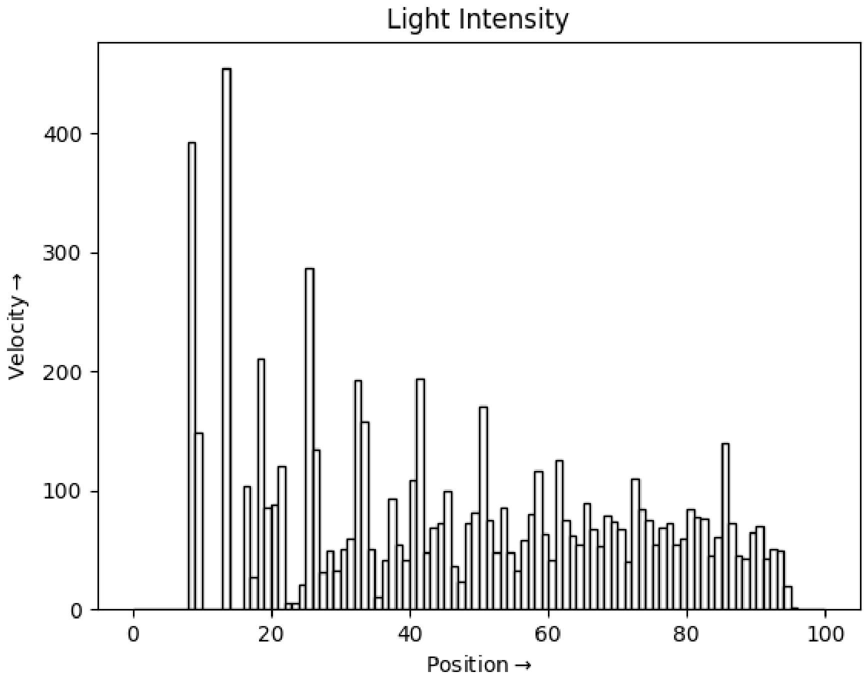

Figure 1: Light intensity vs. position

2. Histogram for Electron Density

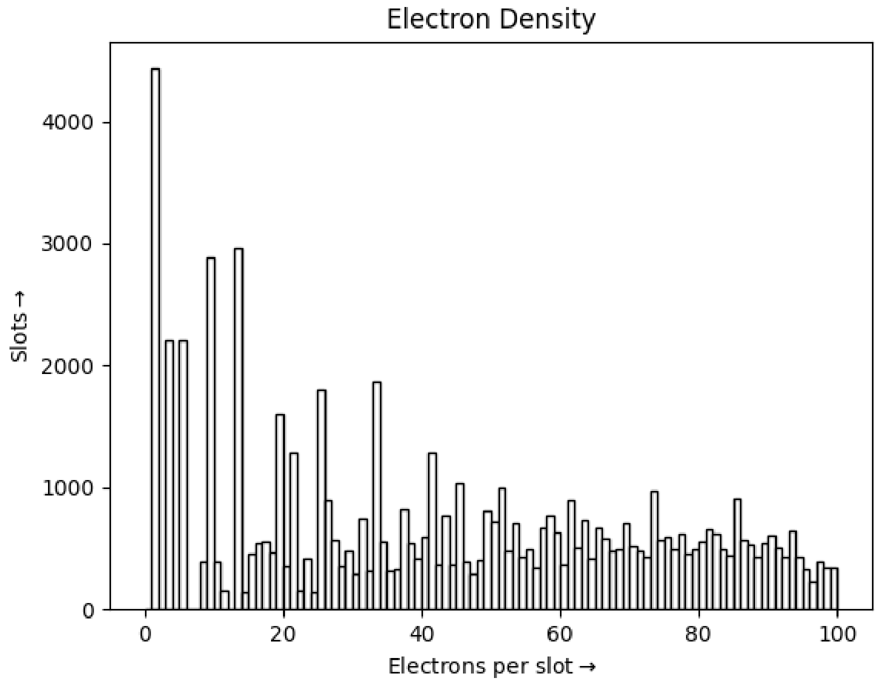

Figure 2: Electron density vs. position

3. Phase space plot for electrons (X vs V)

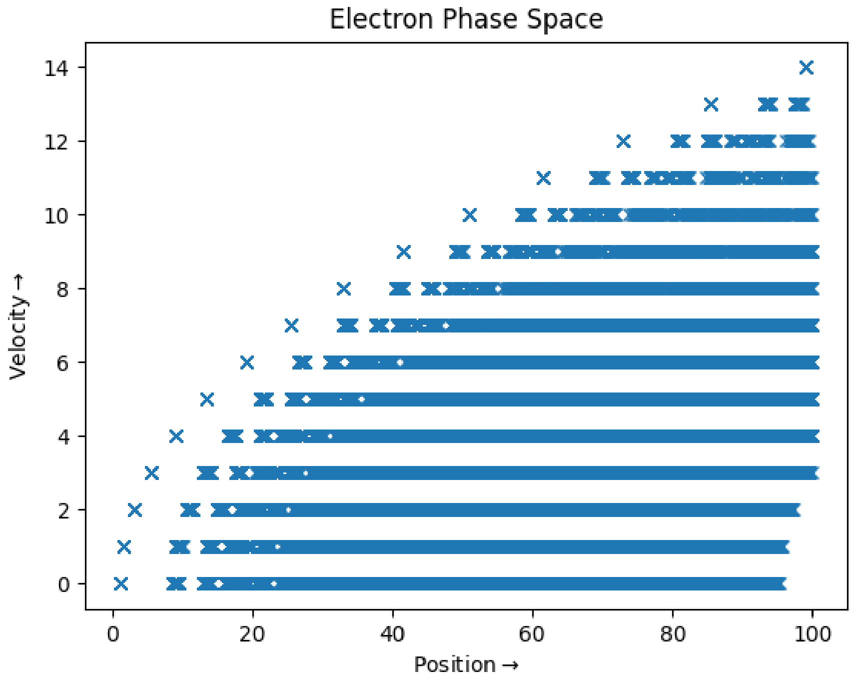

Figure 3: Phase space (X vs V)

4. Tabulating intensity vs. position (For complete list refer code)

```
Intensity data:
xpos count
0      0.5
1      1.5
2      2.5
3      3.5
4      4.5
5      5.5
6      6.5
7      7.5
8      8.5
9      9.5
10     10.5
11     11.5
12     12.5
13     13.5
14     14.5
...
97     97.5
98     98.5
99     99.5
```

4 Accurate distance update

If we take into account that time is distributed uniformly, not the positions, we can make the distance update more accurate by the following:

$$dx = udt + \frac{1}{2}a(dt)^2 + \frac{1}{2}a(1-dt)^2$$

- $udt + \frac{1}{2}a(dt)^2$ term accounts for the change in position due to velocity between time $(k-1)\Delta t$ and $(k-1)\Delta t + dt$.
- Now after dt the velocity becomes zero, till time $k\Delta t$ the distance is simply $\frac{1}{2}at^2 = \frac{1}{2}(1-dt)^2$
- The velocity also becomes $u \leftarrow (1-dt)$ due to the acceleration.

The code looks like this:

```
dt = np.random.rand(len(kl))
xx[kl] = xx[kl] - dx[kl] + ((u[kl] - 1) * dt + 0.5* dt * dt) + 0.5*(1 - dt)**2
u[kl]=1 - dt
```

5 Results for accurate distance update

1. Histogram for Light Intensity

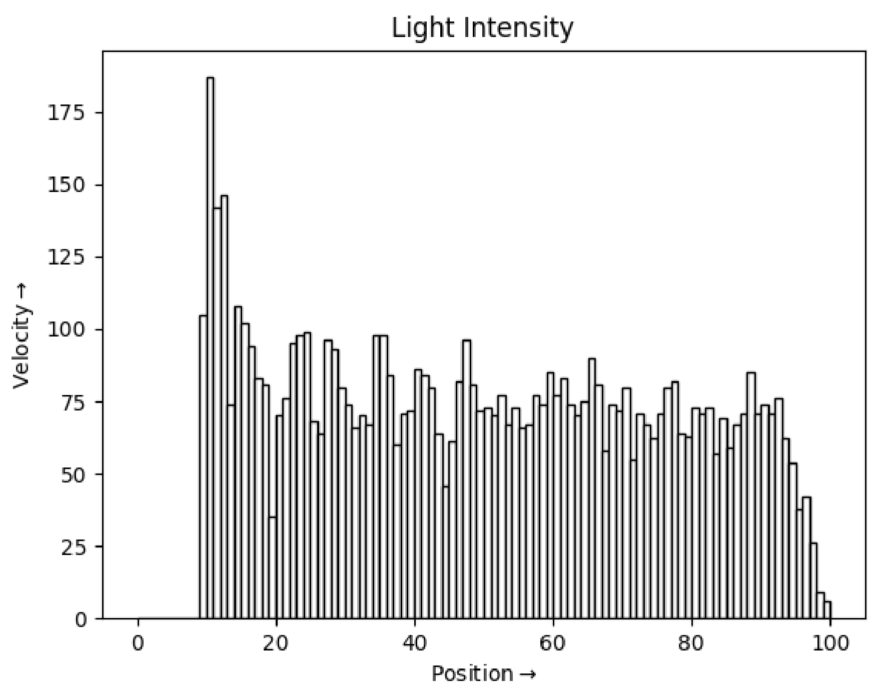

Figure 4: Light intensity vs. position

2. Histogram for Electron Density

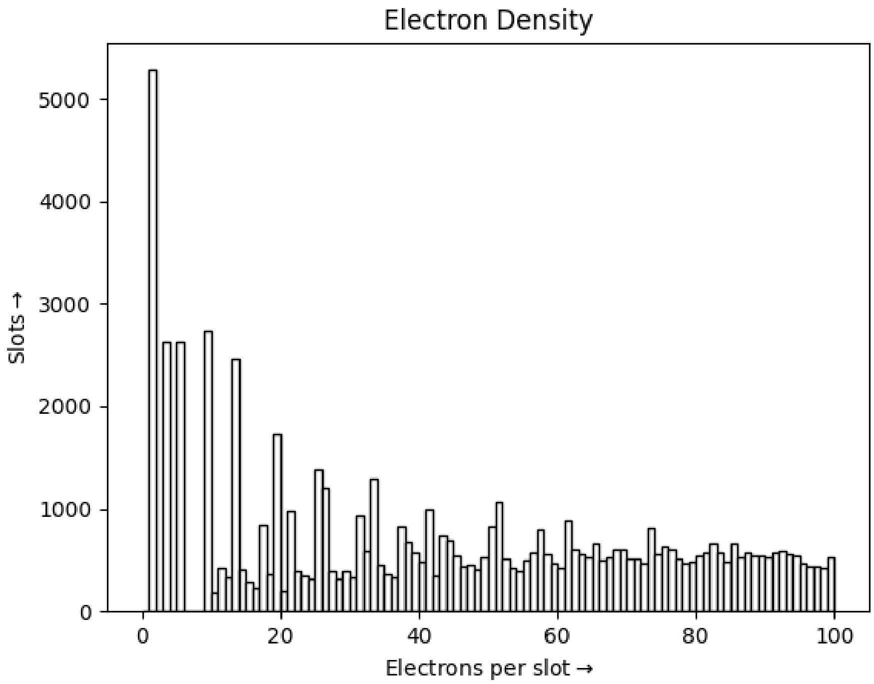

Figure 5: Electron density vs. position

3. Phase space plot for electrons (X vs V)

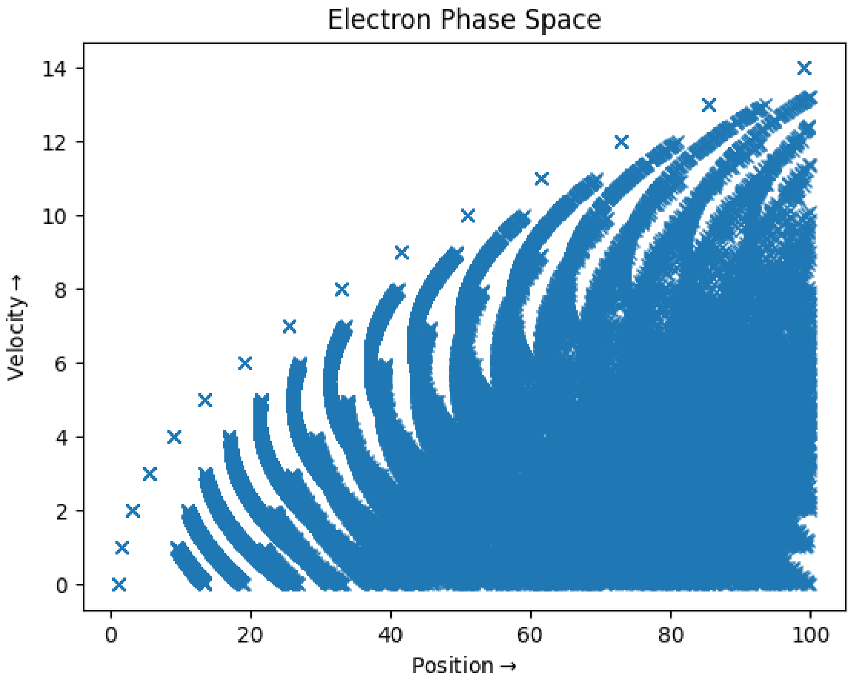

Figure 6: Phase space (X vs V)

4. Tabulating intensity vs. position (For complete list refer code)

```
Intensity data:
xpos   count
0      0.5
1      1.5
2      2.5
3      3.5
4      4.5
5      5.5
6      6.5
7      7.5
8      8.5
9      9.5
10     10.5
11     11.5
12     12.5
13     13.5
14     14.5
...
97     97.5
98     98.5
99     99.5
```

6 Conclusions

- A 1 dimensional model was made and simulated for a tube light.
- Light intensity, Electron density, and phase space diagrams were plotted with and without the accurate updation of distance.
- The tubelight has a dark spot in the region where the electrons have not yet reached threshold velocity. This spot is from cathode side. Here the electrons are not excited enough and therefore very less photons are emitted. In fact, we can calculate the width of this region.

$$u_0 = at$$

$$x = \frac{1}{2}at^2$$

$$x = \frac{1}{2}\frac{u_0^2}{a}$$

- So until $x = \frac{1}{2}u_0^2$, there is a dark spot in the tubelight
- Electron density histogram shows that electron density is highest near the cathode, since that is the point where they are injected and have zero velocity.
- Electron phase space diagram shows bands of electrons having the same velocity. These bands are straight horizontal lines for inaccurate distance update, but follow a curved path for accurate distance update.
- Phase space curve is bounded by a parabolic envelope.