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 The tubelight model

Here we create a mathematical model for the tubelight. It will be a 1 dimensional model. We assume that the electrons are being injected through the cathode with zero energy, and due to the uniform electric field, are accelerated towards the anode. We assume this acceleration as 1 unit. We will use simple equations of 1 dimensional kinematics to make calculations regarding their speed and position in the tube:

$$v = u + at$$
$$x = ut + \frac{1}{2}at^2$$

All electrons reaching the anode are "lost" and replenished by new electrons injected by the cathode.

Those electrons which have energy greater than the threshold energy have an appreciable probability of colliding with atoms, which rise in energy to an excited state, and emit photons when they fall back to their normal stable energy levels (this is assumed to happen instantaneously). Since electrons cross the threshold due to their kinetic energy, the measure of threshold energy here will be the threshold speed. We also assume that whenever an electron suffers a collision, it comes to rest and again begins to accelerate from rest from that position.

A mathematical treatment and implementation of this model will be elaborated in the procedure.

3 Procedure

- We first initialize the simulation universe. Our simulation depends on spatial grid size (the number of divisions we are dividing the tubelight into for our analysis) n, number of electrons injected per turn `M`, number of turns to simulate `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]
```

   Note that we do this only once. We are not doing this here every instance of the loop. This is because the value of `ii` is updated also in step 10. In the code, this statement is placed outside the loop body to ensure it runs only once.

2. Calculate the change in position using some simple kinematics as follows (time interval is 1 unit):

$$dx_i = u_i \Delta t + \frac{1}{2} a (\Delta t)^2 = u_i + 0.5$$

```
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 (These electrons will have position $> n$)

```
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 uniform 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()
   ```

   Note that this method is not accurate. There is a better method to calculate the position of collision which will be discussed in a later section.

8. We now know which electrons have collided. These electrons will excite atoms that emit the photons. So we add those photons to the appropriate positions in 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 available, 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 = list(allIndices - set(ii))

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

   Note that we have made another optimisation here to calculate the empty slots. The emptySlots are simply those spots which do not have electrons. Therefore we simply subtract the array `ii` (containing indices of all the positions of the electrons) from the total set of indices.

10. We append the current position and velocity of the electrons in lists X and V. Note the point we mentioned in step 1. Immediately after this the loop repeats, and so we have eliminated the redundant `where()` command at the beginning by placing it outside the loop to ensure it runs once at the beginning.

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

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

4 Results for inaccurate distance update

1. Histogram for Light Intensity

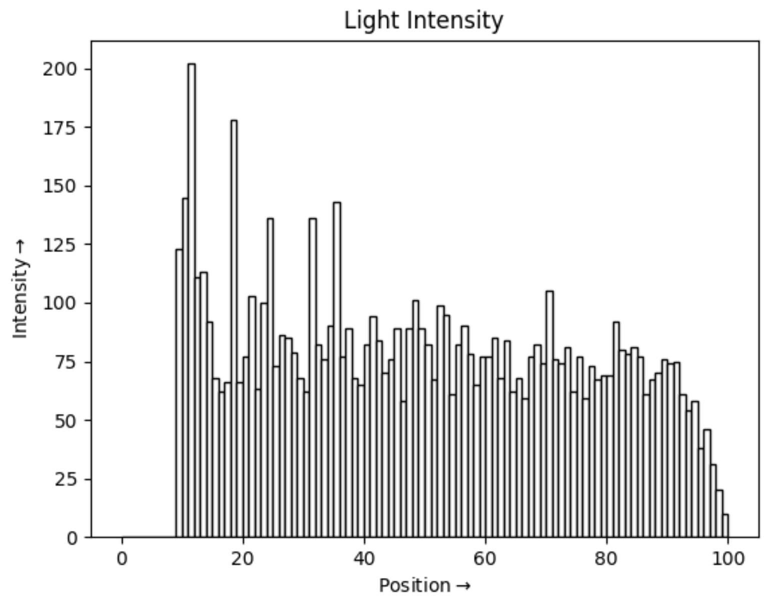

Figure 1: Light intensity vs. position

2. Histogram for Electron Density

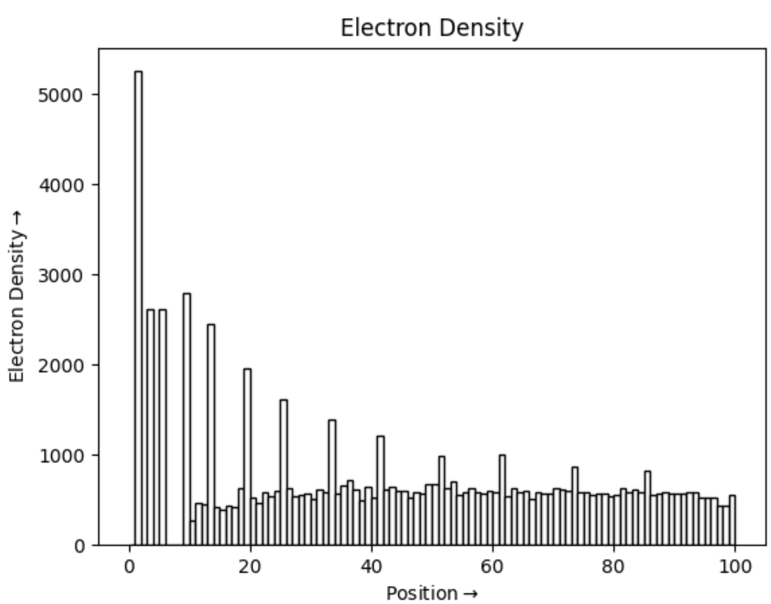

Figure 2: Electron density vs. position

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

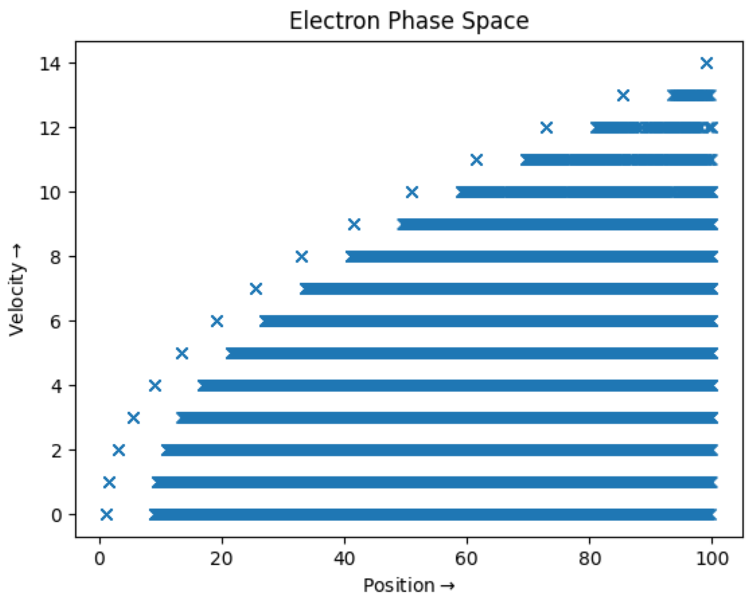

Figure 3: Electron Phase space (X vs V)

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

```
Intensity data:
xpos count
0.5    0
1.5    0
2.5    0
3.5    0
4.5    0
5.5    0
6.5    0
7.5    0
8.5    0
9.5    145
10.5   127
11.5   139
12.5   95
13.5   118
...
97.5    28
98.5    10
99.5    6
```

5 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
```

6 Results for accurate distance update

1. Histogram for Light Intensity

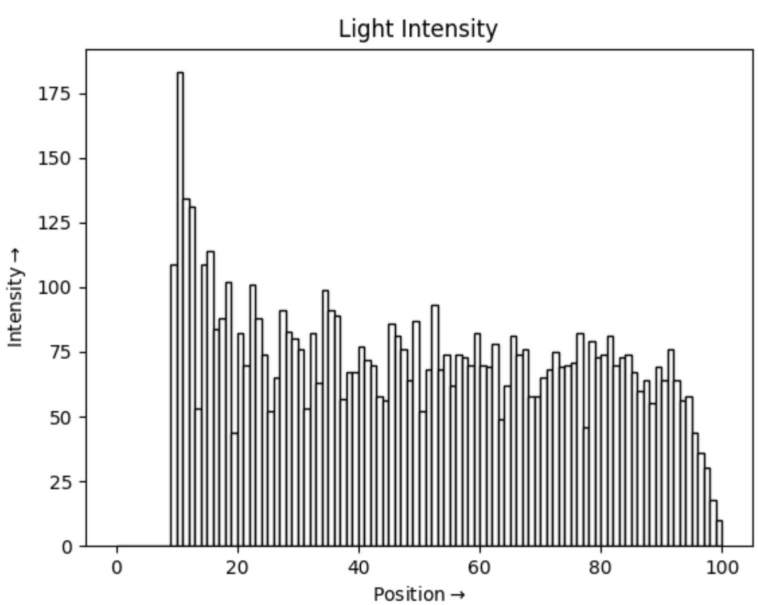

Figure 4: Light intensity vs. position

2. Histogram for Electron Density

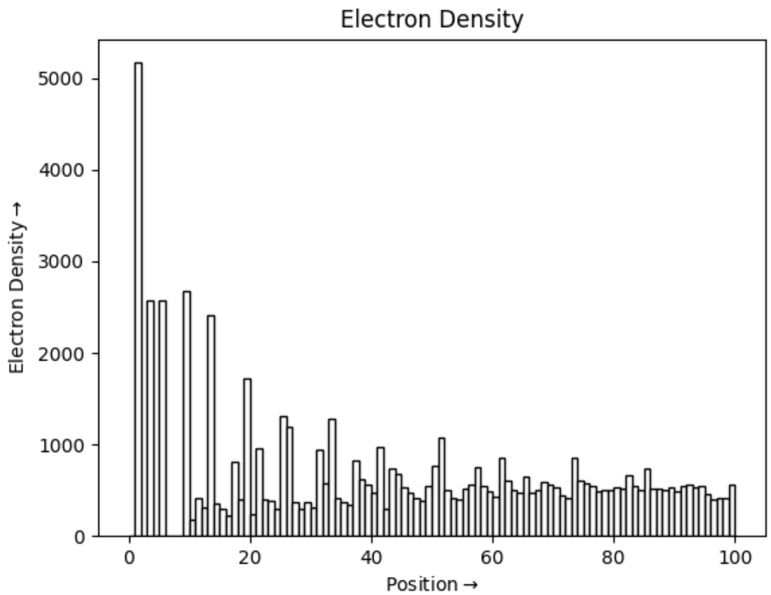

Figure 5: Electron density vs. position

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

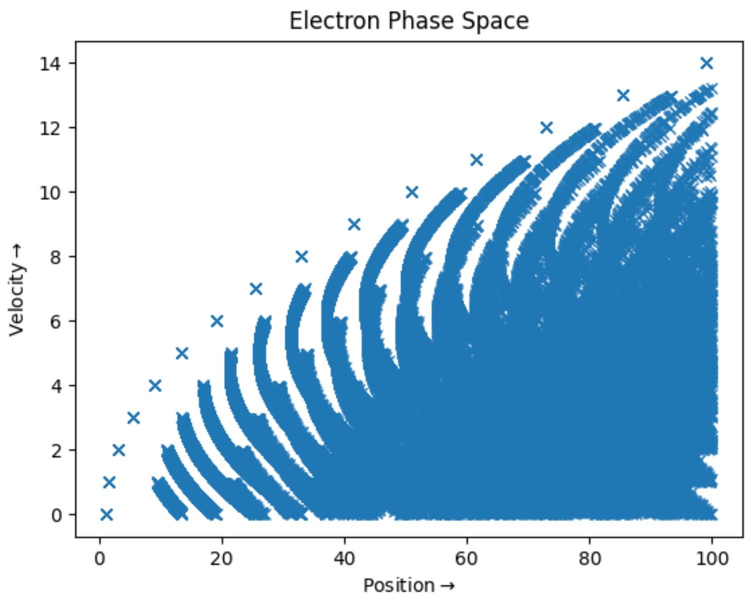

Figure 6: Electron Phase space (X vs V)

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

```
Intensity data:
xpos   count
0.5     0
1.5     0
2.5     0
3.5     0
4.5     0
5.5     0
6.5     0
7.5     0
8.5     0
9.5     129
10.5     188
11.5     176
12.5     155
13.5     75
14.5     157
...
97.5     29
98.5     20
99.5     5
```

7 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. Sure enough, for the default case, $x = \frac{1}{2}(5^2) = 12.5$, and we can see that the tubelight is dark till approximately 10 units. Changing the gas has the effect of changing u_0.
- 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. 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.