EE2703: Applied programming Lab
Assignment 6: Simulation Tubelight

Vishnu Varma V
EE19B059

April 7, 2021

1 Introduction

In this assignment we will simulate a tube light. As a result of uniform electric field inside the tubelight the electrons are accelerated, they start at cathode with zero energy and accelerate in this field. When the energy reaches beyond the threshold energy they can drive atoms to excited state, when these atoms relax light is emitted, the electron loses all its energy and the process starts again. The simulation is carried out for certain number of timesteps from an initial state of having no electrons.

2 Declaring Variables

The following code declares the variables required for simulation, these are the default values and user input can be given using commandline arguments:

```
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 = 2.0       #Standard Deviation
```

The following code declares vectors and lists to store electron information and information to be extracted from simulation:

```
#vectors to hold electron information
xx = np.zeros(n*M)       #xx : Electron Position
u = np.zeros(n*M)        #u : Electron Velocity
dx = np.zeros(n*M)       #dx : Displacement in current turn

I = []  #I : Intensity of emitted light
```

```
X = []  #X : Electron Position
V = []  #V : Electron Velocity
```

3 Simulation

The electrons are in a chamber, its position must satisy $0 < x < L$ where $L = n$ for this simulation.
So those electrons whose position in greater than zero are active electrons so they are stored and the postion and velocity is updated accordingly as they are accelerated due to the electric field.
Code:

```
ii = np.where(xx>0)
#updating the position and velocity of these electrons
dx[ii] = u[ii] + 0.5
xx[ii] += dx[ii]
u[ii] += 1
```

Those electrons that reached anode are collected and their position and velocity is set to zero.
Code:

```
anode = np.where(xx>=n)
#the position and velocity of these electrons are set to zero
xx[anode] = 0
u[anode] = 0
```

Now finding those electrons that have energy greater than threshold energy it is analogous to find electrons with velocity greater than threshold velocity. After finding it we create a uniformly distributed random vector and find those indices that have vallues less than p. Now reverting back these indices to find the indices of those energetic electrons that suffer a collision.
Code:

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

Since they suffered an inelastic collision their velocity is set to zero and their position could have occured at any point between previous x_i and current x_i. So assigning it a random position.
Code:

```
u[kl] = 0
xx[kl] -= dx[kl]*np.random.rand()
```

Now again injecting m new electrons which is randomly distributed with mean M and standard deviation M_{sig}. These new electrons are added to

the free slots available (The slots off electrons whose position is zero), and setting the position and velocity as done initially.
Now again repeating the same process for the newly injected electrons, finding the active electrons whose position is positive and the same procedure is followed.

The Intensity, Position, Velocity lists are updated for every loop which are plotted below: