Data Analysis and Monte Carlo Simulations

Ivan Sepulveda

Summer 2017- Spring 2018

A two-dimmensional Monte-Carlo simulation is used to obtain Electron Energy Distrubution Function demonstrated by electrons inside low temperature plasma generated in pure argon.

1 Introduction

Over the past handfull of decades the semiconductor industry has consistently miniturized computer chips but in recent times seems to have hit their limit. The process of using high-speed plasma discharges to create these integraded circuits is known as plasma etching. By studying the properties of plasma, one can hope to be able to improve plasma etching process.

The electron energy distrubition fuction is one of the key properties in understanding low-temperature plasma. One could attempt to experimentally measure the properties of a plasma by inserting a certain tool known as the Laugmuir probe. Unfortunately, as soon as the probe is inserted, the properties of the plasma would change.

Theoretically, plasma properties could instead be measured using numerical simulations. Many would initally attempt to solve the Boltzman equation. Doing so would require variables including number of electron density within the plasma, the element which composes the plasma as well as density of atoms within the plasma, and what types of collisions are possible (i.e. purely elastic, ionization, excitation, etc.).

Another effective approach would be to create a Monte-Carlo simulation. A Monte-Carlo simulation is a method of approximately solving mathematical and physics problems by the simulation of random qualities. In our study we simulate the propagation of electrons through argon plasma. The electrons follow the electrific field lines but their collisions with Argon atoms are somewhat random: collision angles as well as the types of collisions it experiences are based off probabilities from previously measured cross sections.

First, this report will cover our theoretical calculations and approach. Afterwards, we will go over the logical and analytical assumptions and calculations and how they are calculated and implemented in the scrict. Subsequently leading us to explain results obtain from running the script under different coditions by altering variables including propagations allowed and magnitude of the electric field applied. Finally, we will analyze our results and draw upon conclusions and whether or not they agree with both our theory and results measured from outside sources.

2 Theory

Because an electric field causes electron acceleration, if an electron is allowed to accelerate in an electric field contraining Argon plasma, it will collide with various argon atoms. During which, the electron could simply deflect off the atom elastically, ionize the atom, or transfer a quantized amount of kinetic energy to excite the argon atom. The probability for a certain collision is given with a cross-section for said collsion. Using previously measured cross-sections we determine the electron energy distribution fuction.

In our study we simulated a low-temperature argon plasma produced by heating argon gas with radio-frequency (RF) electromagnetic (EM) waves. Rather than creating 2-D constraints, we limit the number of propagations the electron undergoes. Our plasma has a variable DC bias creating an electric field in the +x direction. By Newton's 2nd law, the electron accelerates in the same direction as the electric field with magnitude as seen in Eq 2.1. Due to the unvarying force, the equations of motion for constant acceleration can be applied.

$$F_x = q_e E_x = m_e a_x \rightarrow a_x = (q_e E)/m_e \tag{2.1}$$

$$v_x = v_{0x} + a_x t \tag{2.2}$$

$$v_y = v_{0y} = const. \tag{2.3}$$

$$x = x_0 + v_{0x} t + \frac{1}{2} a_x t^2 \tag{2.4}$$

2.1 2D Collision Theory

The collision theory is derived from the conservation of momentum and total energy.

$$\vec{p_e} + \vec{p_a} = \vec{p_e}' + \vec{p_a}' + \vec{p_e} \tag{2.5}$$

$$m_e \vec{v_i} = m_a \vec{v_a} + m_e \vec{v_f} \tag{2.6}$$

$$m_e \vec{v_{ix}} = m_a \vec{v_{ax}} + m_e \vec{v_{fx}} \tag{2.7}$$

$$m_e \vec{v_{iy}} = m_a \vec{v_{ay}} + m_e \vec{v_{fy}} \tag{2.8}$$

After which we procede to use convervation of energy.

$$E_i = E_f + E_a + \Delta E_{loss} \tag{2.9}$$

$$\frac{1}{2} m_e {v_i}^2 = \frac{1}{2} m_e {v_f}^2 + \frac{1}{2} m_a {v_a}^2 + \Delta E_{loss} \tag{2.10}$$

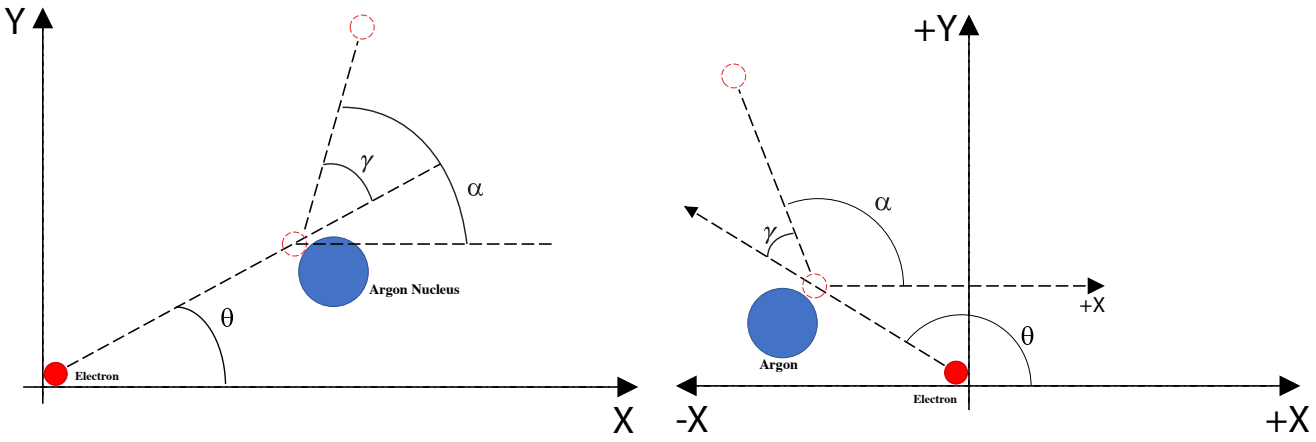
$$v_{ix} = \cos\theta \tag{2.11}$$

$$v_{ix} = \sin\theta \tag{2.12}$$

The atom cannot backscatter so to preserve the symetry in two our two dimensional setup, we defined angle θ as the incident angle of an electron relative to the x-axis and γ as the angle at which the electron scattered off the argon atom relative to its inital trajectory. By doing so, we can define angle α such that.

$$\tan\alpha = \tan(\theta + \gamma) \tag{2.13}$$

Therefore α is the electron scattering angle relative to the x-axis as seen in Fig. 2.1. We generated the scattered angle γ as a random number between $-\frac{\pi}{2} \leftrightarrow +\frac{\pi}{2}$, found α then expressed all quantities in terms of $\tan\alpha$. After applying these changes as well as manipulating Eqs. 2.5 -



(a) Quadrant 1

(b) Quarant 2

Figure 2.1: Collision Illustration

2.12, we obtained v_a and it's horizontal component v_{ax} which allowed us to determine the new x and y components of the electrons velocity $\vec{v_f}$ as well as the electrons final energy E_f

$$v_a = \frac{(v_{ix} + v_{iy}\tan\alpha)\frac{1}{\sqrt{1+\tan^2\alpha}} \pm \sqrt{(v_{ix} + v_{iy}\tan\alpha)^2\frac{1}{1+\tan^2\alpha} - \frac{2\Delta E}{m_a}(1+\frac{m_a}{m_e})}}{1+\frac{m_a}{m_e}} \tag{2.14}$$

$$v_{a,} = \frac{m_e}{m_a + m_e}\frac{v_{ix} + v_{iy}\tan\alpha}{\sqrt{1+\tan^2\alpha}} \tag{2.15}$$

$$v_{ax} = \frac{v_a}{\sqrt{1+\tan^2\alpha)}} \tag{2.16}$$

2.2 3D Collision Theory

Here in this section I will elaborate on the math and reasoning behind collisions similar to how I did with the 2D case. Temporarily using Kushner's figures but I am drawing my own in Adobe Illustrator! I'll also add those matrix equations and transformations in an easy to read format.

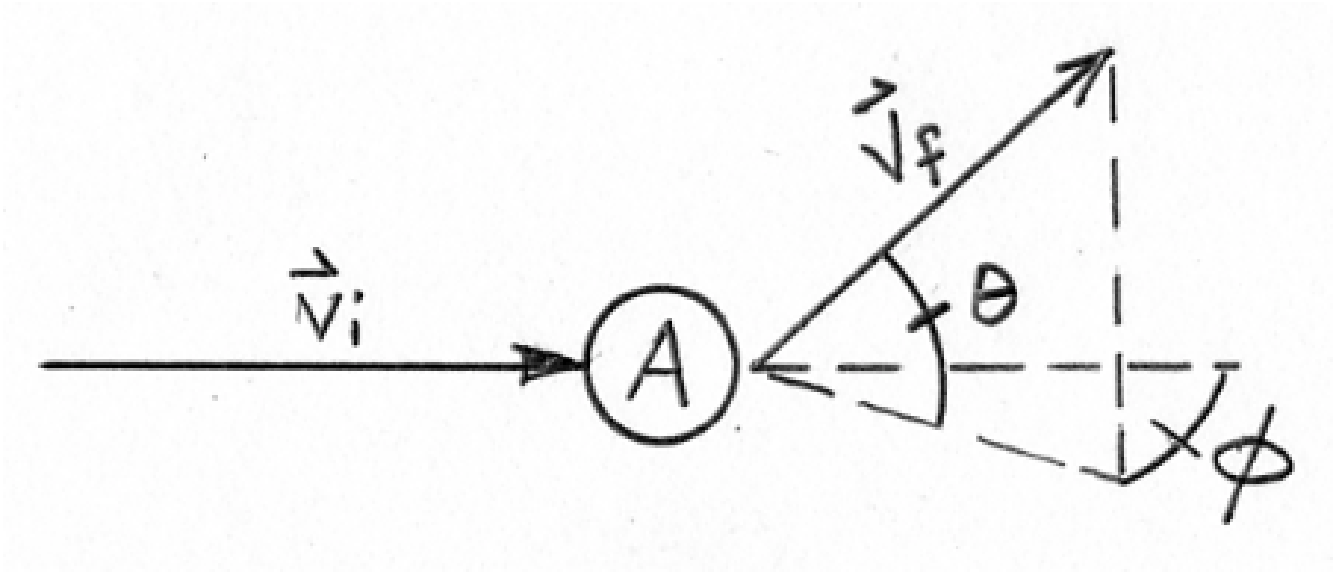


Figure 2.2: Cross Section vs Energy

Our 3-Dimmensional collision theory redefines some angles as seen in the figures below

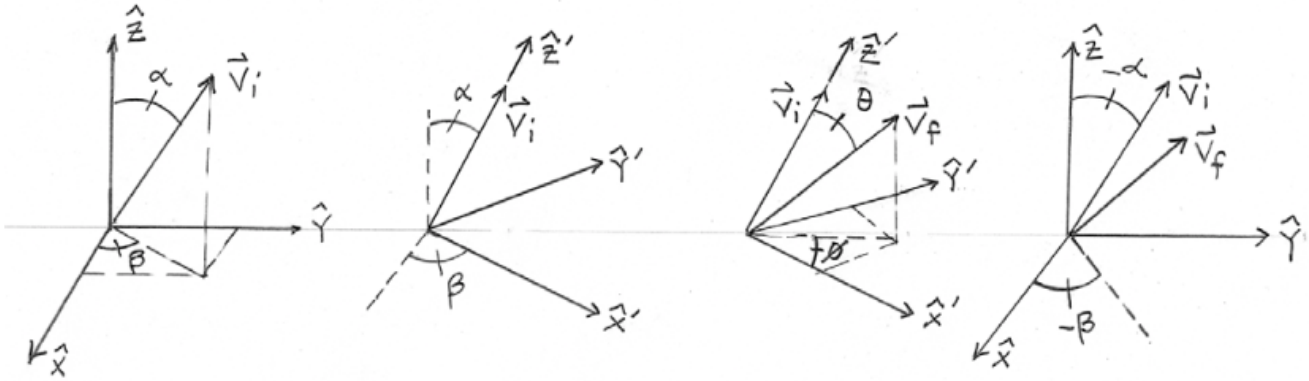


Figure 2.3: Collision Process

2.3 Cross Sections

The cross section of a certain type of collision is a function $\sigma(e)$ of the electrons energy ϵ

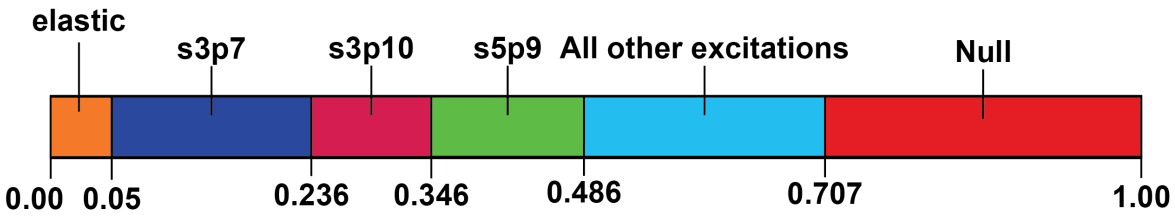


Figure 2.4: Weighted Probabilities for an electron with 15 eV.

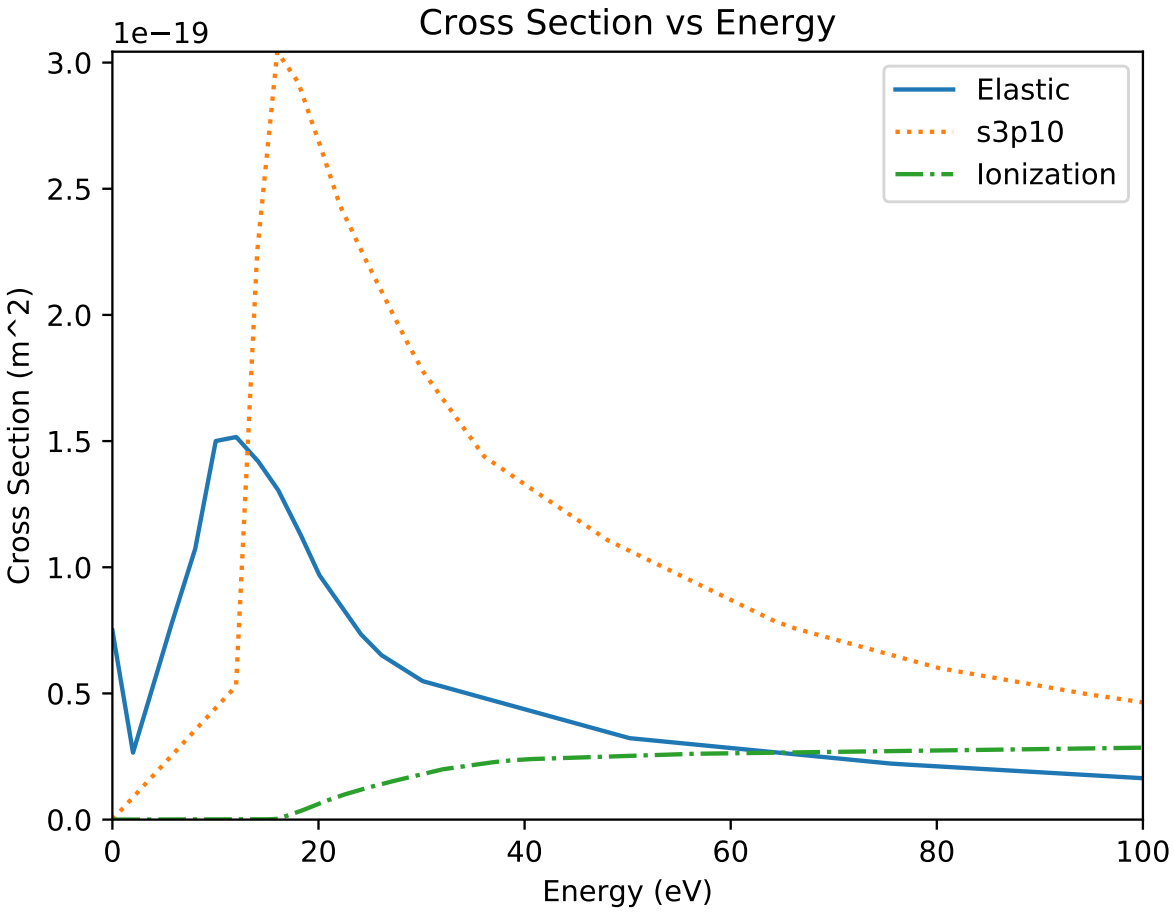


Figure 2.5: Cross Section vs Energy

2.4 Collision Frequencies

```
    def collision_frequency(energy_eV, process_cross_section_dictionary):
 if isnan(energy_eV) or energy_eV > 99.99:
    energy_eV = 99.99
 cf = (2*energy_eV*eV_J/me)**0.5
 cf = cf*process_cross_section_dictionary[iround(ener_eV)]
 cf = cf *argon_volume_density
 return cf

    def total_CF(energy, dictionary_of_cs_dictionaries):
sum = 0
for processD in dictionary_of_cs_dictionaries:
  sum += collision_frequency(energy, dictionary_of_cs_dictionaries[processD)
return sum
```

Note: The function "isnan()" returns a boolean depending on whether or not "energy_eV" is a real number or decimal. At timest calculations in pythan can become so large or small that it is not considered a number anymore. If this happens durring the collision-propagation process, the electrion comes back with the string "NaN" in place of a value for it's energy. This is a built-in aspect of python with the purpose as to not to overwhelm the computer. ("NaN being short for "Not a Number").

After moving the electron in this time according to the equations of motion, you should calculate the new position (x and y) and the new velocity v. We calculate the magnitude of the velocity as $v = \sqrt{v_x^2 + v_y^2}$. Since the only force acting upon the electron is the Coulomb force F = qE, all the work done by this force will be converted into electron's kinetic energy and is calculated using the standard $E = \frac{1}{2}m_e v^2$. where v is the magnitude of the electron's velocity. Up until the first collision the electron will only have a v_x component. To determine if the electron collided or not, first we determine its Maximum Total collision frequency and make the assumption that at that energy, the electron has to collide whether it be elastically or inelastic. For every other energy, there will be a null collision frequency (NCF) greater than zero.

Energy (eV)	$f_{elastic}$ $(\times 10^6 s^{-1})$	f_{excite} $(\times 10^6 s^{-1})$	f_{ion} $(\times 10^6 s^{-1})$	f_{tot} $(\times 10^6 s^{-1})$	f_{null} $(\times 10^8 s^{-1})$	$f_{tot+null}$ $(\times 10^8 s^{-1})$
0	0	0	0	0	55.54	55.54
5	8.96	15.29	0	2.425	53.11	55.54
19.42	26.82	527.16	139.26	5,553.810	0.00003	55.54
19.43	26.81	527.18	139.74	5,553.812	0.000009	55.54
19.44	26.79	527.19	140.21	5,553.813	0	55.54

2.5 Cumulative Probabilities

We'll have to come back to this

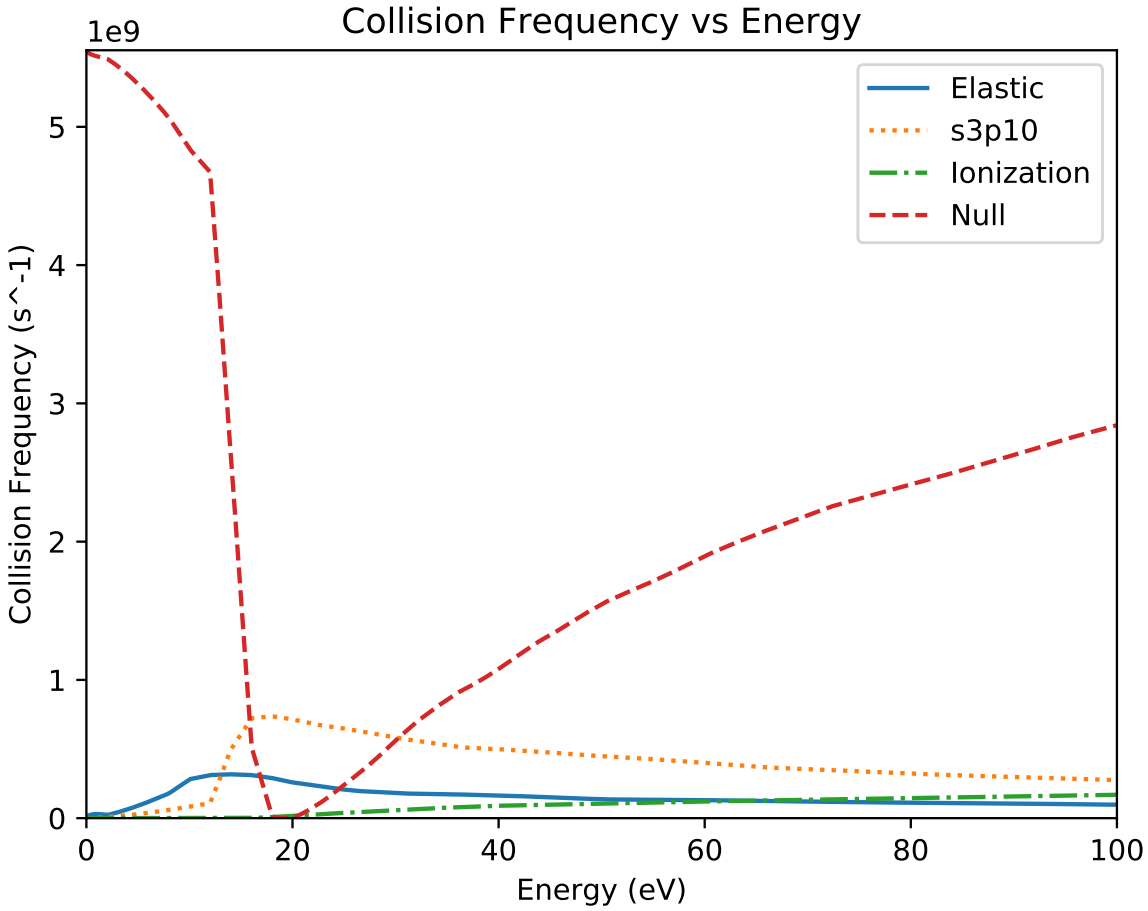


Figure 2.6: Collision Frequency vs Energy

3 Code

3.1 Code Structure

As can be seen in Figure 3.1 script was split into two main processes: running the simulation and plotting subsequent results. The former requires many more requisites: in order to create, propagate, and collide the electron, we first need to determine which collision the electron will experience any at all. Probablities for each collision type will depend on the electrons energy just before impact. Our simulation is accurate up to two decimal points for electron energy in units of electron volts. Unfortuantely we run into an obsticle since the cross sections provided are given in multiples of two. By runing a quick script to linearly interpolate these cross sections given in the DAT files we overcome this obstacle. Plotting the results is a bit more straightforward. Specific python libraries allow us to quickly create a scatterplot for Energy Distrubution as well as the locations of the collisions exprienced. One could even sacrifice computational runtime and plot the path experienced by every single electron.

3.2 Calculating Time of Flight

Given the plasma's particle density and the electrons energy, it's time of flight (TOF) can be calculate. Here we define our TOF to be the time interval between collisions. We calculate

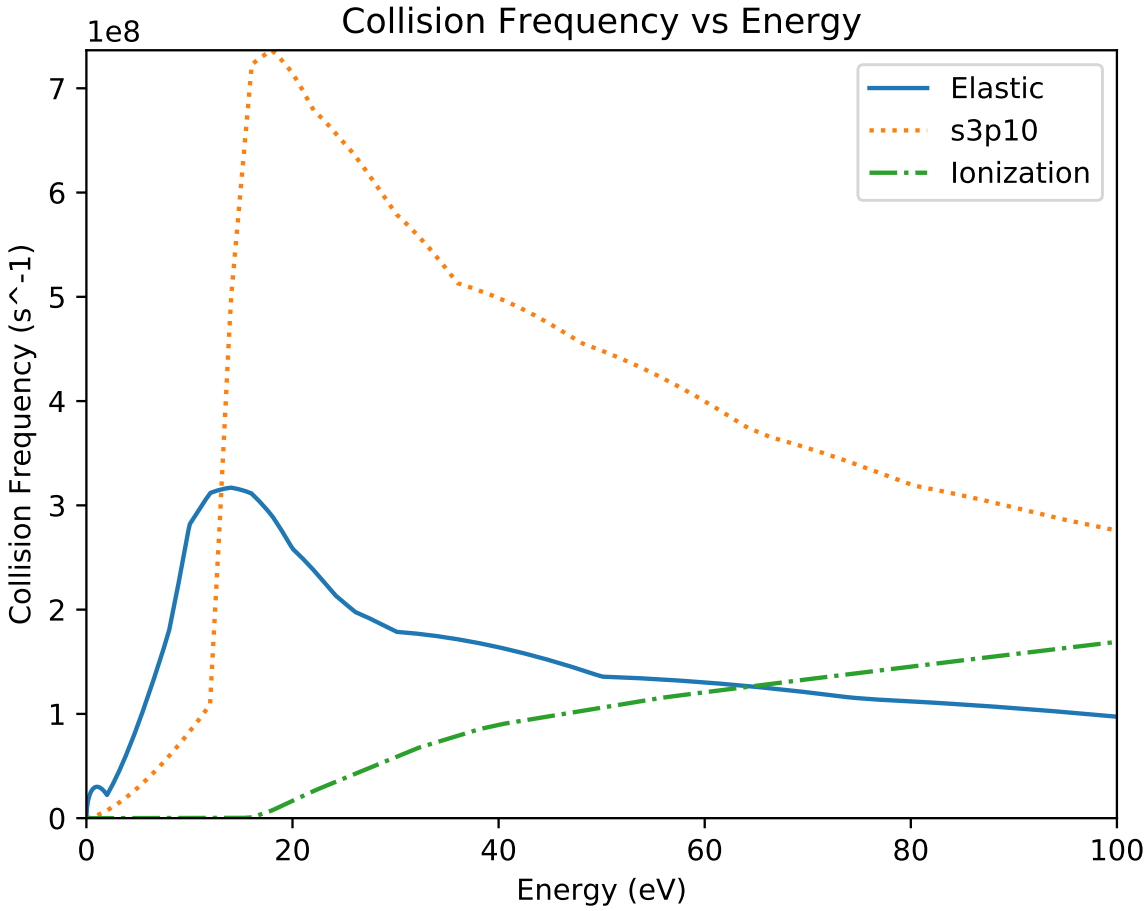


Figure 2.7: Collision Frequency vs Energy

our TOF as Δt in equations 3.1 - 3.2 as a function of Total Collision Frequency (TCF) and r, a randomly generated float between 0 and 1. Unfortunately we run into an issue at first propagation. Becuse initial energy is set at 0eV, $TCF = 0$. This is corrected by applying a condition such that if the electron has less than 0.01eV, it's Δt will be exactly 100 nanoseconds.

$$E > 0.01eV : dt = \frac{-\log(1-r)}{TCF} \tag{3.1}$$

$$E <= 0.01eV : dt = 10^{-7}s \tag{3.2}$$

4 Results

- Simulated experiment
- 2d Ar Plasma
- atom density
- we included all the excitation collisions from
- ground - 24 excel

Table 1: My caption

Energy (eV)	$f_{elastic}$ $(\times 10^8 s^{-1})$	f_{s3p7} $(\times 10^8 s^{-1})$	$f_{otherexcite}$ $(\times 10^6 s^{-1})$	f_{ion} $(\times 10^7 s^{-1})$	f_{tot} $(\times 10^8 s^{-1})$	f_{null} $(\times 10^8 s^{-1})$
0	0.00	0.00	0.00	0.00	0.00	55.54
5	0.90	0.19	133.49	0.00	2.43	53.11
10	2.81	0.55	377.57	0.00	7.14	48.40
15	3.15	10.20	2615.37	0.02	39.51	16.03
19.41	2.68	16.57	3614.86	1.39	55.54	0.00
19.42	2.68	16.57	3614.81	1.39	55.54	0.00
19.43	2.68	16.57	3614.76	1.40	55.54	0.00
19.44	2.68	16.57	3614.71	1.40	55.54	0.00
25	2.07	16.38	3433.12	3.83	53.17	2.37
30	1.79	15.30	3221.72	5.85	49.89	5.65
35	1.73	14.36	3013.47	7.61	46.99	8.55
40	1.64	13.47	2874.28	8.97	44.75	10.79

- 1st excited

We start off by generating some intial conditions. First electron is created its initial position, velocity, and acceleration are set to zero. For simplicity, we round electron mass and charge down to their first decimal: $m_e = 9.1 \cdot 10^{-31} kg$, $q_e = 1.6 \cdot 10^{-19} C$. We assume that number of argon atoms per unit area is $N = 10^{21} m^2$.

4.1 Interpolating Cross Sections

After interpolating all given cross sections and plugging them into eq. 4.1 we find that $TCF_{Max} = 5,553.813 \cdot 10^6$ at 19.44 eV. At this exact energy, we make it so that the electron has to collide, or in other words at this energy $f_{null} = 0$ Hz. As displayed in the table below, we see that null frequency quickly starts to increase the within just a few electron volts of the our peak.

$$f = N_A \sigma(\epsilon) \sqrt{\frac{2\epsilon}{m_e}} \tag{4.1}$$

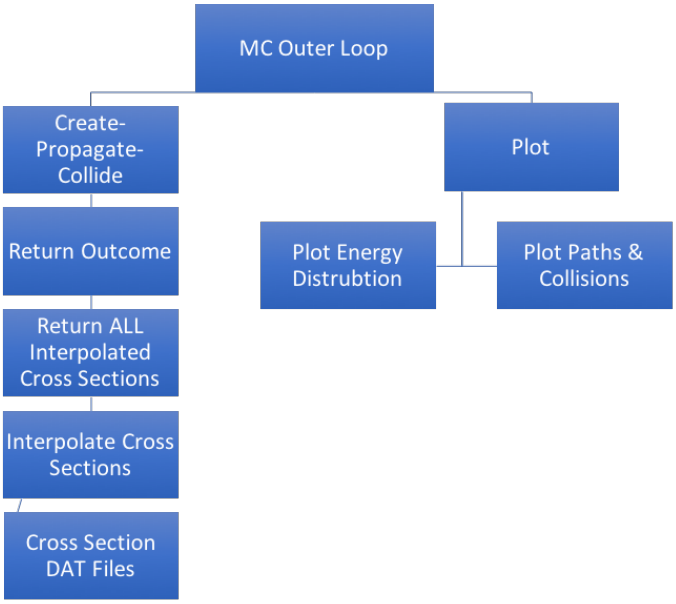


Figure 3.1: Code Structure