AEROSOL SCATTERING

Using AI Based Approach

An Exploration of AI-based Approaches for Aerosol Scattering Analysis

Table Of Contents

Background

"This college project focuses on predicting the intensity of scattered light, specifically after the scattering process. Leveraging a combination of experimental data and artificial intelligence techniques, our goal is to develop a predictive model that accurately estimates the intensity resulting from light scattering. This project contributes to a deeper understanding of light interactions and provides valuable insights for applications in atmospheric science and environmental monitoring. By integrating hands-on experimentation with AI methodologies, we aim to enhance the precision of Intensity prediction in the context of light scattering, contributing to the broader knowledge base in this specialized field."

History

Rooted in the historical exploration of light scattering phenomena, our college project bridges the insights of luminaries like Lord Rayleigh and Mie, who, in the late 19th and early 20th centuries, laid the groundwork for understanding the scattering of light by small and larger particles, respectively. Leveraging this historical foundation, we are integrating artificial intelligence into our contemporary approach to predict light wavelength post-scattering. By fusing the principles of Rayleigh and Mie scattering with advanced computational techniques, our project aims to contribute to the ongoing narrative of light interaction research, offering a modern perspective on these classical theories and pushing the boundaries of our ability to predict and comprehend light behavior in scattering scenarios.

Our Project

Code Enhancement

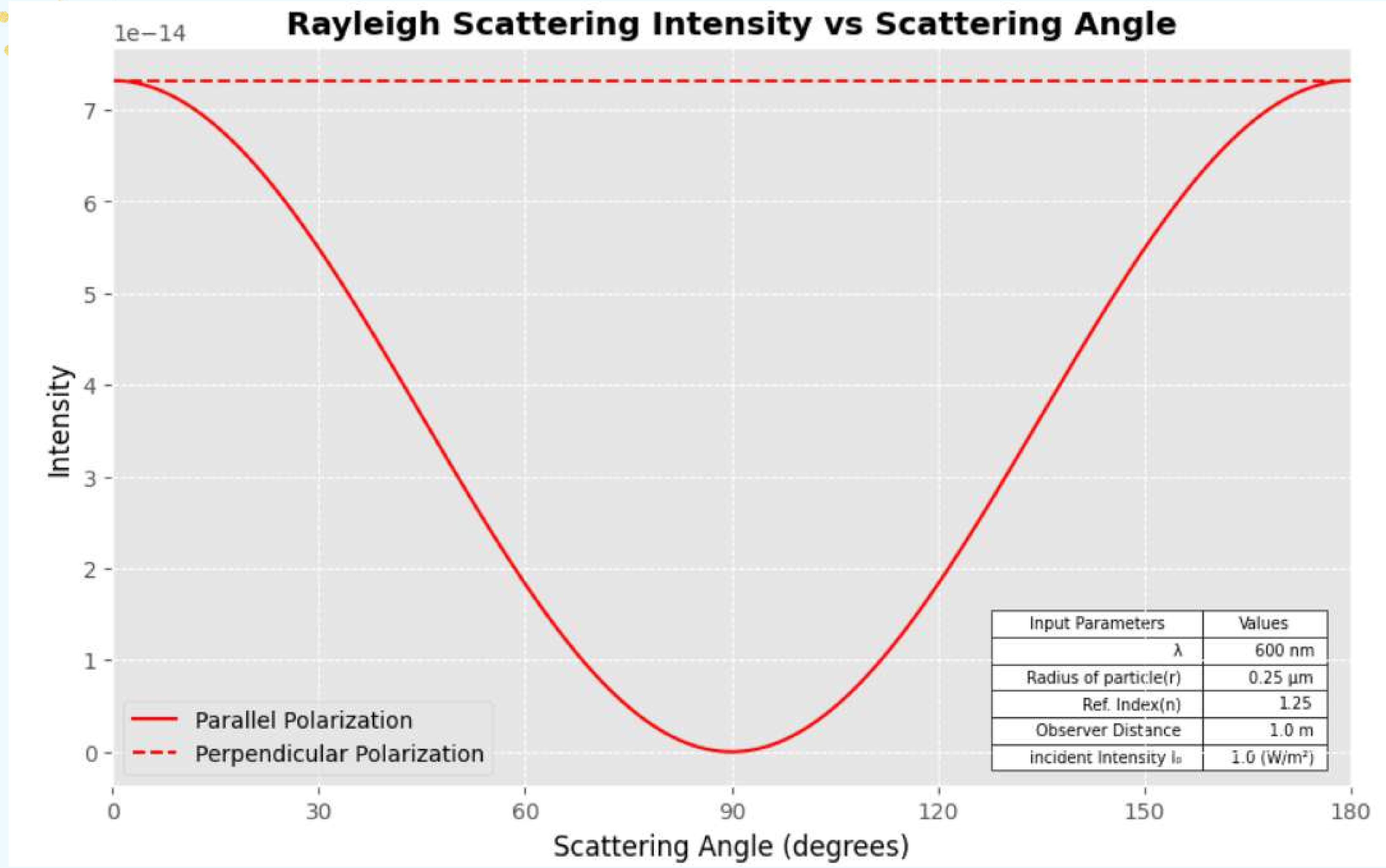
Rayleigh Scattering

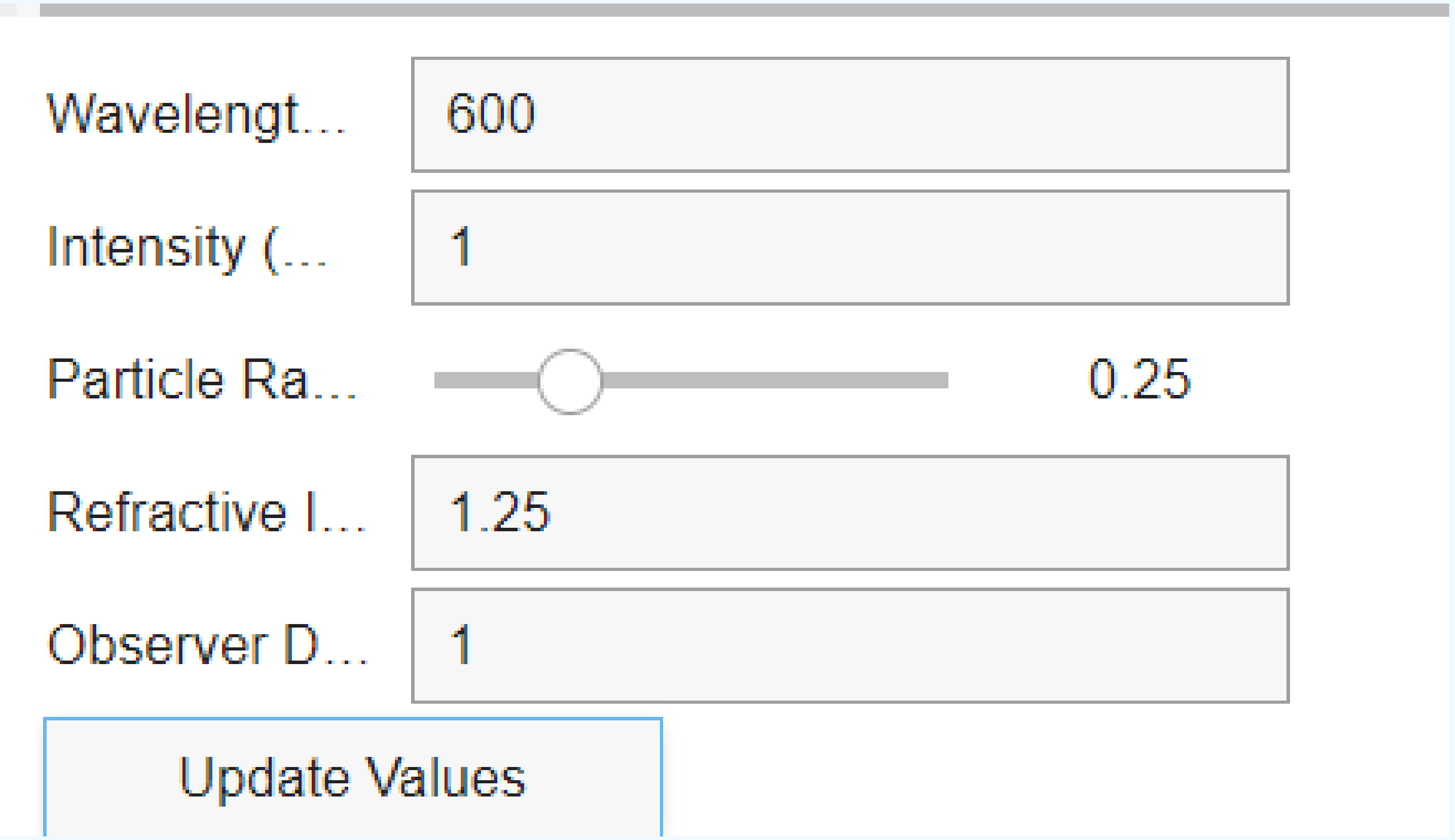
Mie Scattering

Conclusion

Tweaking the code

We modified the the provided code to an interactive tool for visualizing Rayleigh scattering intensity. Users can input parameters such as wavelength, intensity, particle radius, refractive index, and observer distance through sliders and text boxes. The tool calculates and plots Rayleigh scattering intensity for both parallel and perpendicular polarizations as a function of scattering angle, ranging from 0 to 180 degrees.





Interactive

The interactive interface allows users to dynamically explore how changes in input parameters influence the scattering pattern.

How we achieved this goal

Widget based interaction

The use of IPython widgets allows for an interactive user interface, enhancing user experience by providing sliders and text boxes for input parameters.

Clear Visualisation

The code generates a clear and informative plot of Rayleigh scattering intensity against scattering angle,

Dynamic

The code enabled users to explore and visualize the impact of changing input parameters on Rayleigh scattering intensity in real-time.

COLAB-LINK

We wrote this code in python, and it works well with Google Colab, as demonstrated

Clear Visualisation

https://colab.research.
google.com/drive/1Tyr
d13-
vUmgj4sojzyquQVjCbo
owsS21?usp=sharing

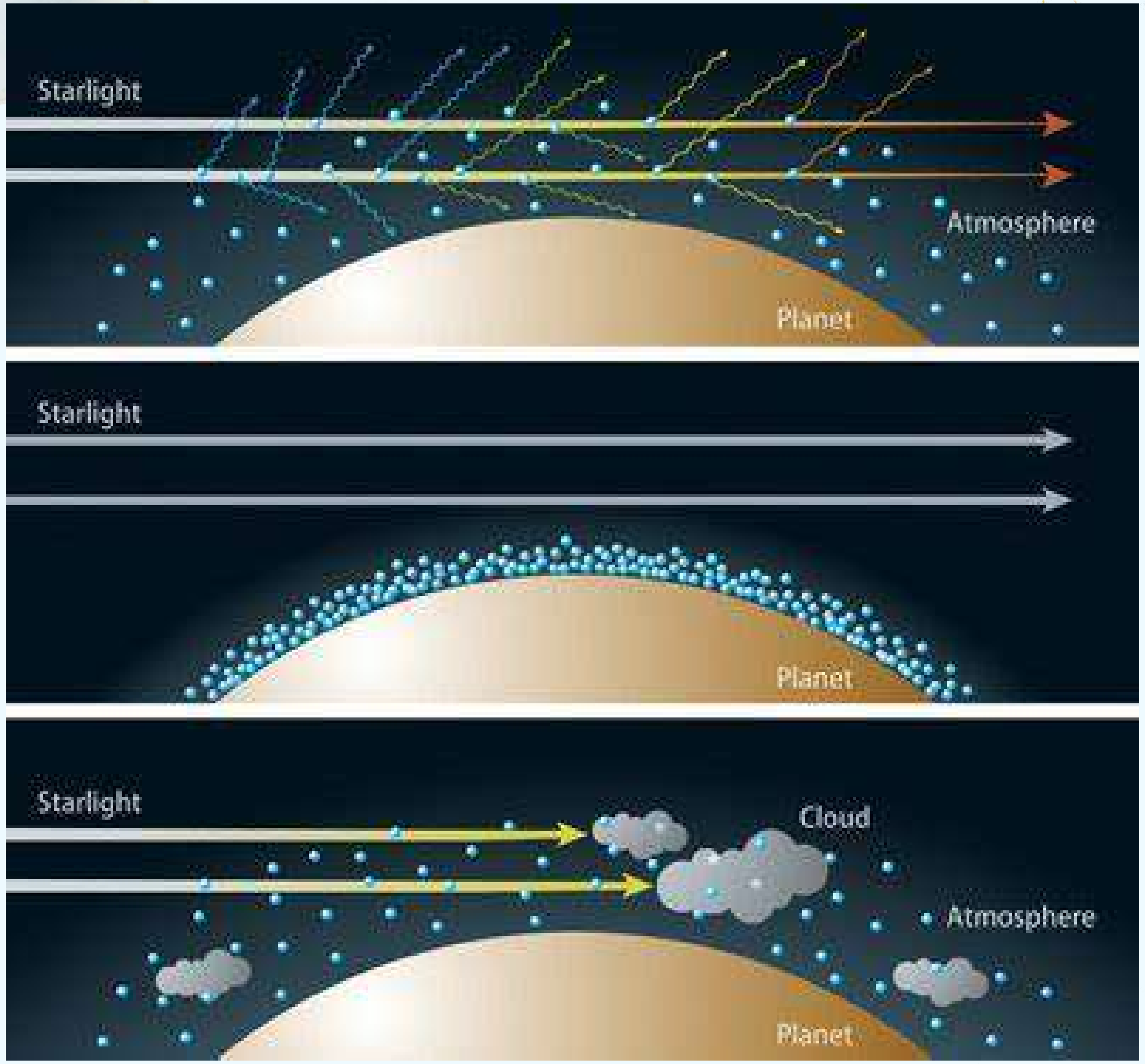
Rayleigh Scattering

https://colab.research.
google.com/drive/1xX
osKf4M4cwOXPmWOq
VYxCwL3CZu9CCP#scr
ollTo=Te9V1b7BfKcK

Google Colaboratory
google.com

Rayleigh Scattering

It is is the predominantly elastic scattering of light, or other electromagnetic radiation, by particles with a size much smaller than the wavelength of the radiation.



Before model preparation

Data Preparation

we have plotted the graph for Rayleigh scattering on default setting.
We got a .txt file from Mieplot software, then we converted txt to .csv file.

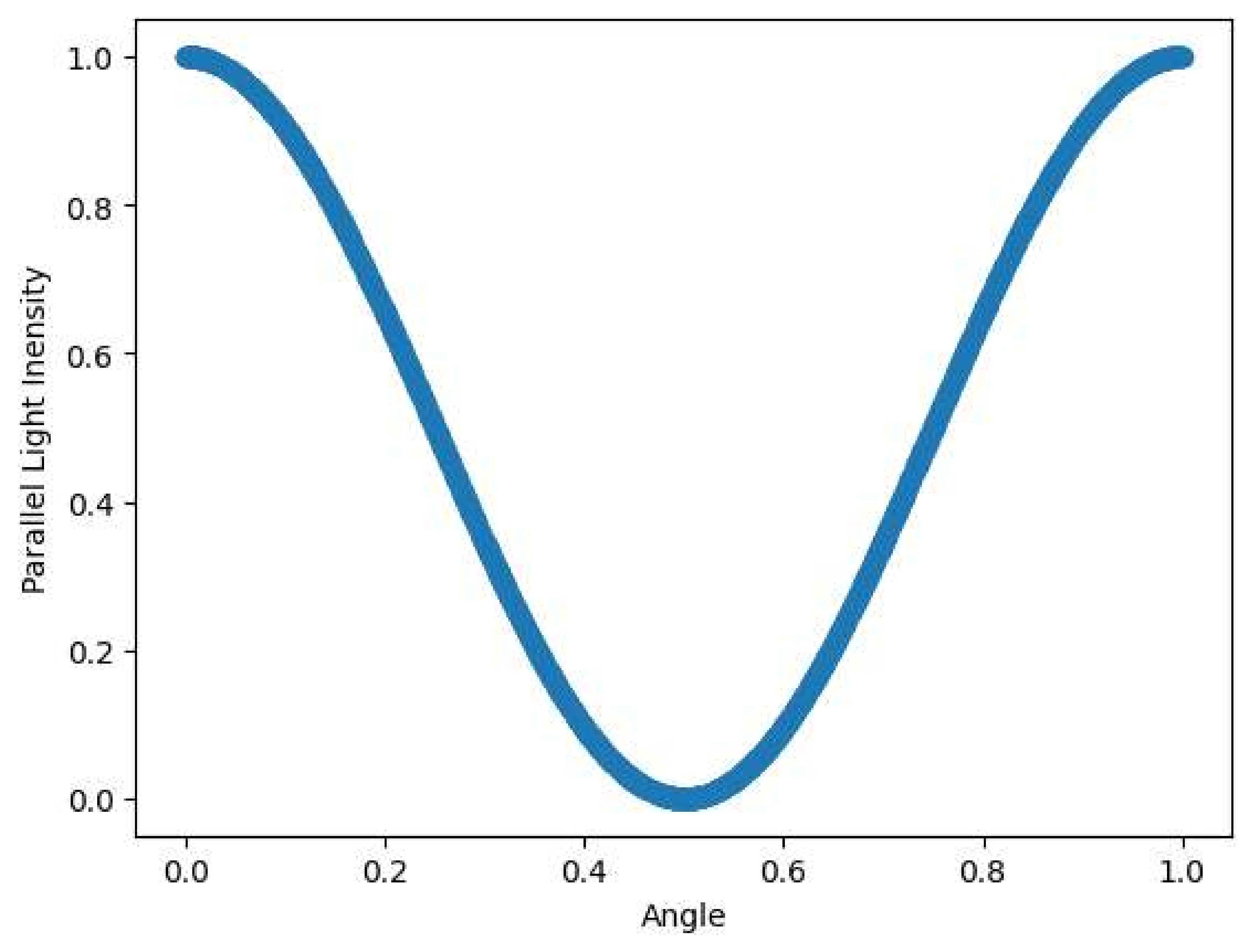
Feature selection

We have four features in dataset. Form those four features we have selected angle and parallel light feature for our model.

Model Selection

The relation between our both features is not linear.
So we cannot apply simple regression model here. like Linear regression.
so we have selected Gaussian regression model.

Graph Showing variation b/w angel and parallel light Intensity.



Model Preparation

Checking for null value

```
angle            0
perpendicular    0
parallel         0
unpolarized      0
dtype: int64
```

Data Splitting for model training and model evaluation.

Model implementation

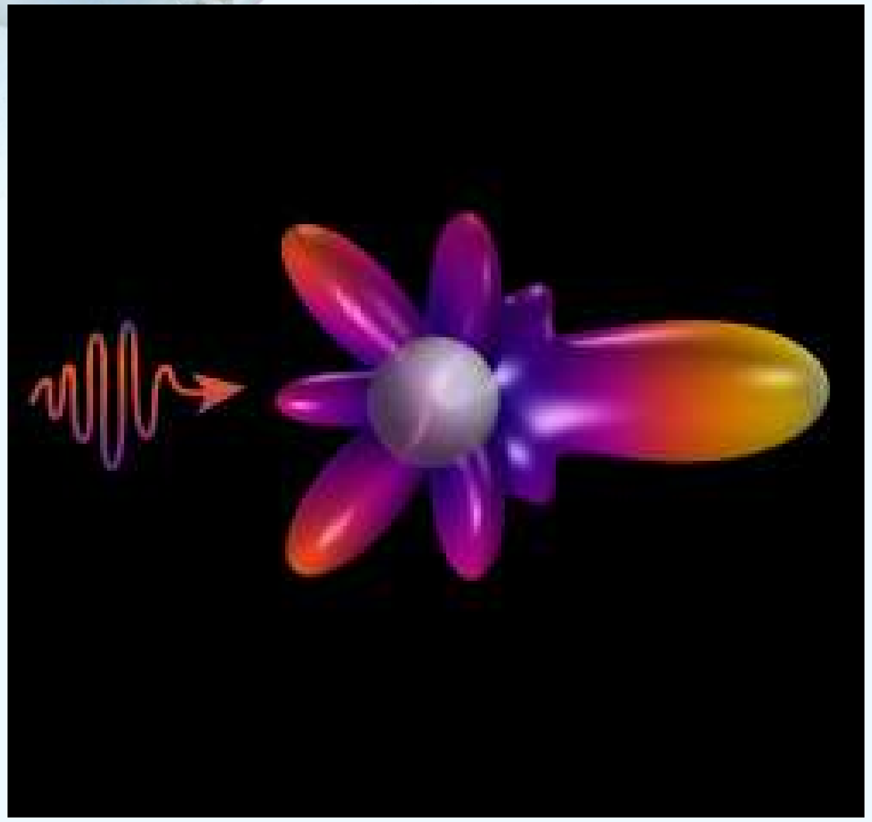
```
GaussianProcessRegressor
GaussianProcessRegressor(kernel=1**2 * RBF(length_scale=1),
                         n_restarts_optimizer=10)
```

Model Predication

```
y_pred, sigma = gp.predict(X_test, return_std=True)
```

Mie Scattering

- When particles are subjected to intense shock loading ,particulates emit from the surface and in order to understand the properties of these particulates we have to measure the size distribution of the particulates.
- Two methods to measure the size distribution of particulates in mie-scattering are:-
  Multi-angle Mei Scattering and Multi-wavelength extinction.



Mei Scattering occurs when the size of the particulates is same as wavelength of light

This is the scattering function

$$|S(\theta,\alpha)|^2 = \sqrt{||S_1(\theta,\alpha)||^2 + ||S_2(\theta,\alpha)||^2},$$

where

$$S_1(\theta,\alpha) = \sum_{k=1}^{\infty} \frac{2k+1}{k(k+1)} \left(a_k(\alpha)\pi_k(\theta) + b_k(\alpha)\tau_k(\theta)\right)$$

$$S_2(\theta,\alpha) = \sum_{k=1}^{\infty} \frac{2k+1}{k(k+1)} \left(a_k(\alpha)\tau_k(\theta) + b_k(\alpha)\pi_k(\theta)\right)$$

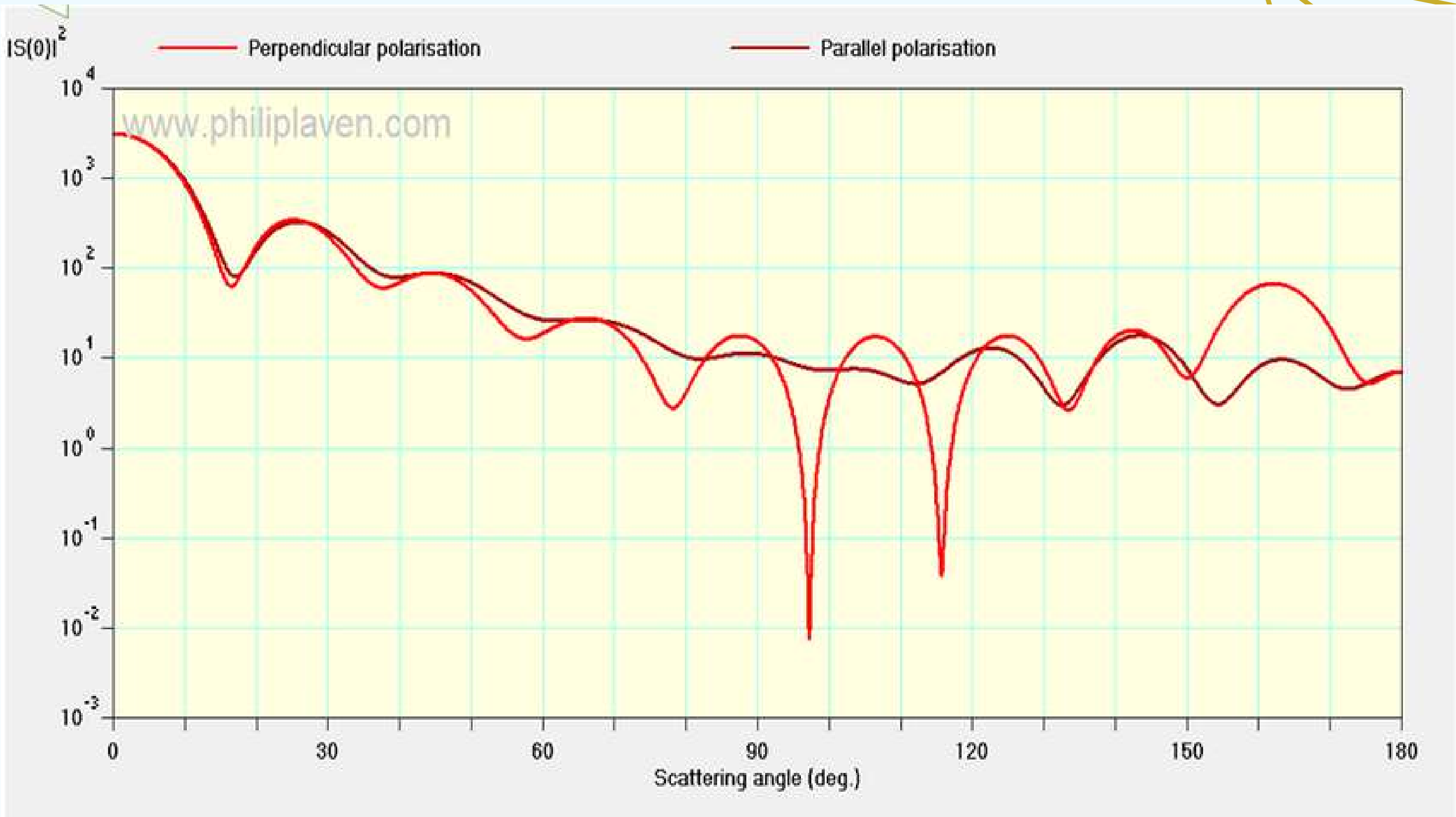
Mei Scattering coefficients

$$a_k(\alpha) = \frac{\nu\psi_k(\nu\alpha)\psi_k'(\alpha) - \psi_k(\alpha)\psi_k'(\nu\alpha)}{\nu\psi_k(\nu\alpha)\xi_k'(\alpha) - \xi_k(\alpha)\psi_k'(\nu\alpha)}$$

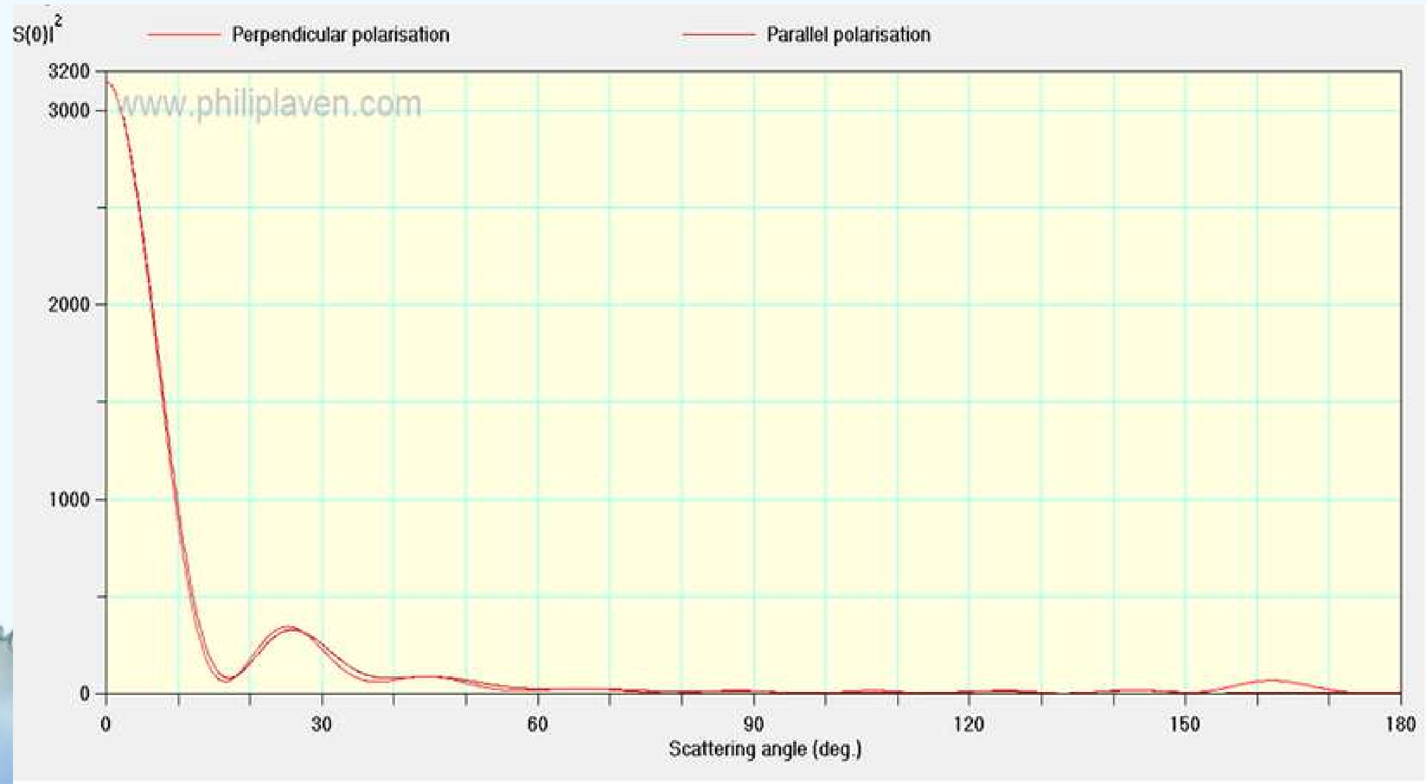
$$b_k(\alpha) = \frac{\psi_k(\nu\alpha)\psi_k'(\alpha) - \nu\psi_k(\alpha)\psi_k'(\nu\alpha)}{\psi_k(\nu\alpha)\xi_k'(\alpha) - \nu\xi_k(\alpha)\psi_k'(\nu\alpha)}$$

Plot for both Parallel and Perpendicular intensity

Logarithmic scale for intensity



Linear scale for intensity



How we achieved this goal

Widget based interaction

The use of Python program allows for an interactive user interface, enhancing user experience by providing sliders and text boxes for input parameters.

Clear Visualisation

The code generates a clear and informative plot of Mei scattering intensity against scattering angle,

Before model preparation

Data Preparation

we have plotted the graph for Mei scattering on default setting.
We got a .txt file from Mieplot software, then we converted txt to .csv file.

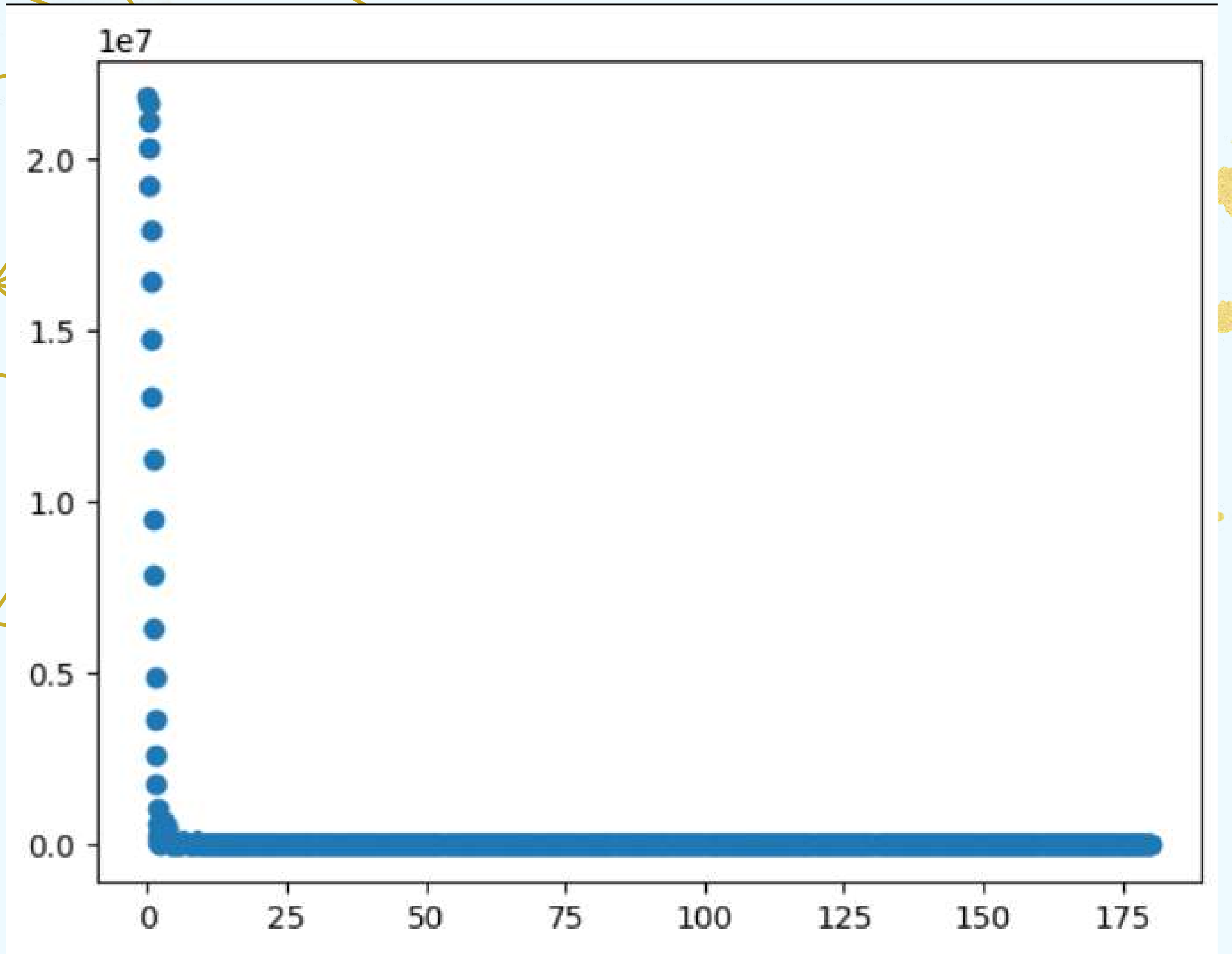
Feature selection

We have four features in dataset.
Form those four features we have selected angle and parallel and perpendicular light feature for our model.

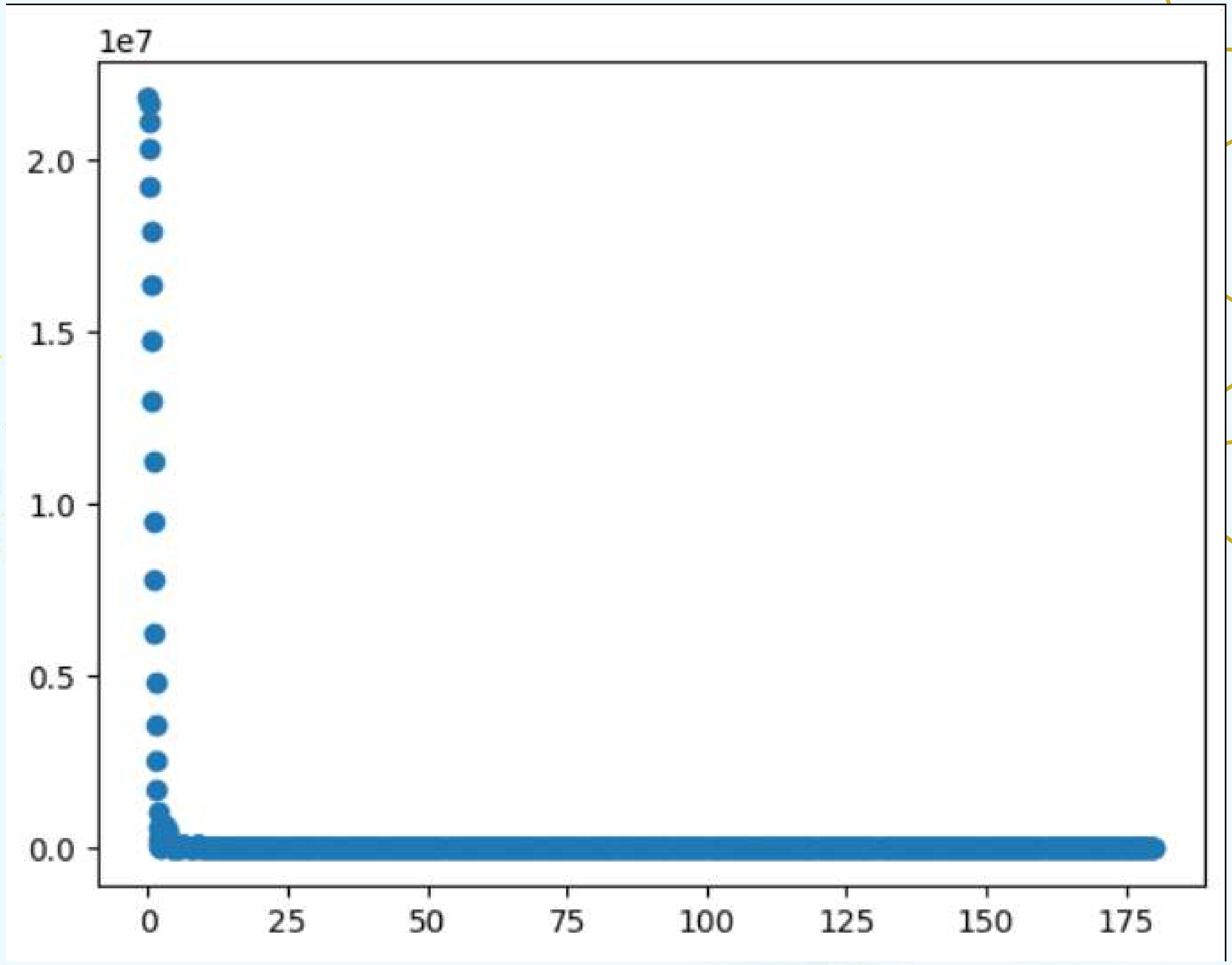
Model Selection

The relation between our both features is not linear.
So we cannot apply simple regression model here. like Linear regression.
so we have selected Gaussian regression model.

Results we got from the code for linear intensity



Perpendicular feature



Parallel feature

Results we got from the code for logarithmic intensity

Perpendicular feature

Parallel feature

COLAB-LINK

We wrote this code in python, and it works well with Google Colab, as demonstrated

Mei Scattering

https://colab.research.google.com/drive/1M0JmNIIYSkI_FJRm9TM64xqvxpa8QyRm?usp=sharing

Google Colaboratory
google.com

We have updated the code for Mei Scattering

https://colab.research.google.com/drive/1W9eZCcGt6zX7PiXxVVY109vWSfmRFggR?usp=sharing

We have updated the code for Rayleigh Scattering

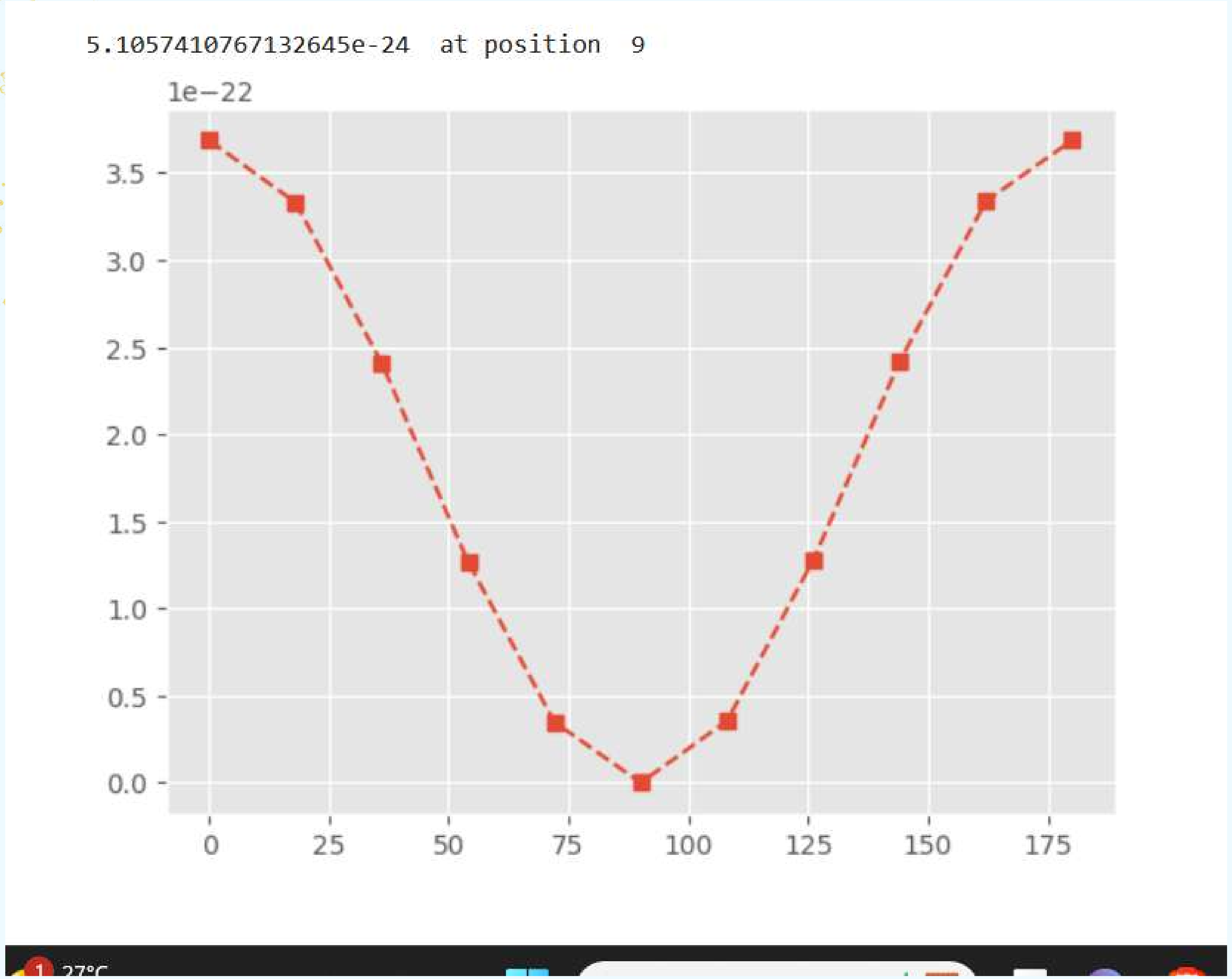
https://colab.research.google.com/drive/1hnu0J4jjjVA7pcTyMq3j9v56OrdJXslIi?usp=sharing#scrollTo=GXo_WhOhsdmR

Prog ress!

We have allowed for the instantaneous creation of a dataset depending on the parameters given by the user. The code will automatically train itself on the dataset which we have created (that depends on various parameters). Then, we have used Gaussian Regression to get the predict the values and calculate the various evaluation metrics like accuracy.

Comparison of Slopes

First, we have shown for 10 divisions. Here we clearly see that the slopes are linear and the graph is made up of joining various slopes. In the first mode, we have compared the slope inputted with that present in thegraph and if found, it has displayed else it has been displayed that the slope does not belong to that particle size. We have allowed for some bandwidth too.

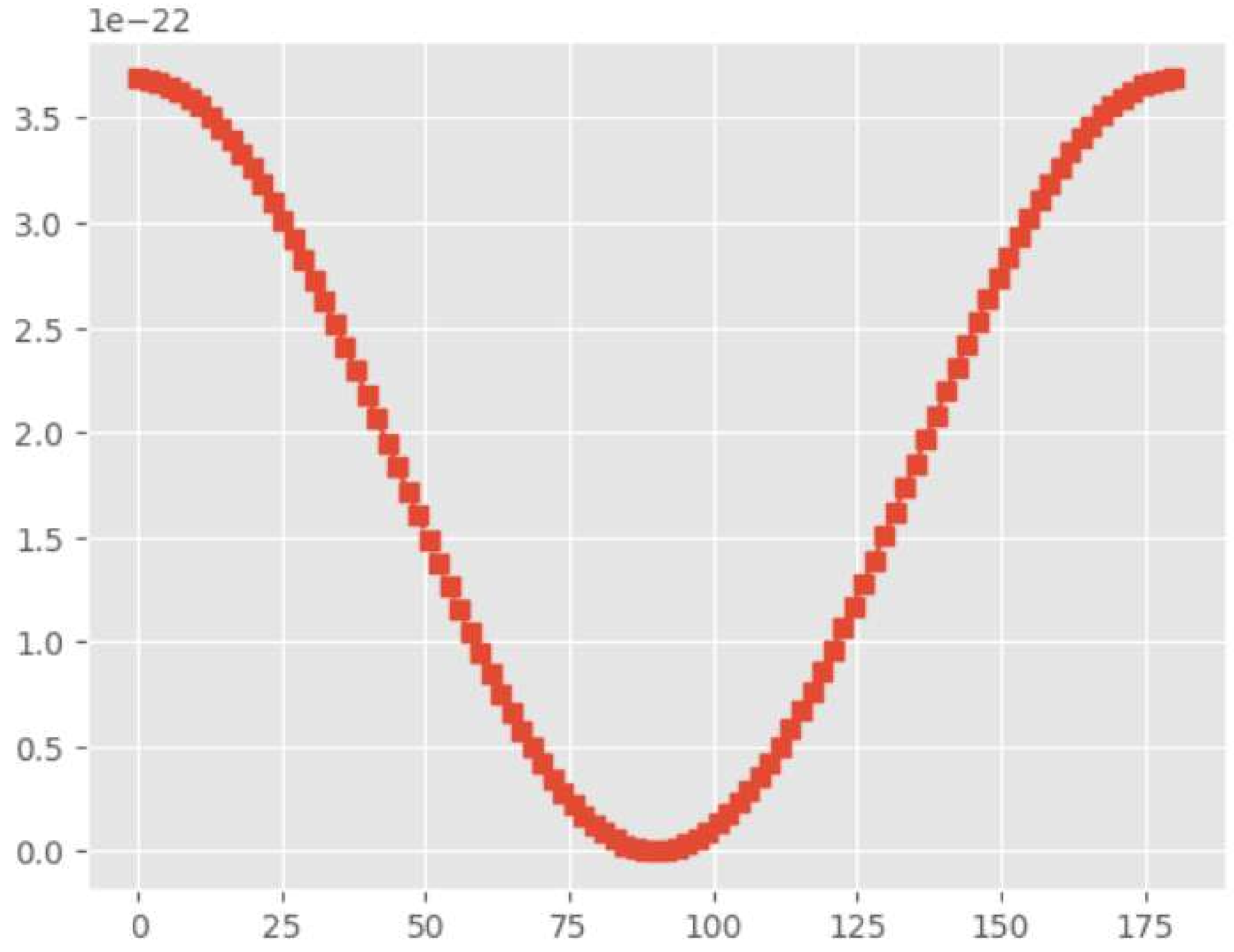


Comparison of Slopes

On increasing the number of comparisons, the graph slowly starts to resemble the original curve. Hence,our method is tending to high accuracy.
It basically works by finding the nearest slope within possible ranges.

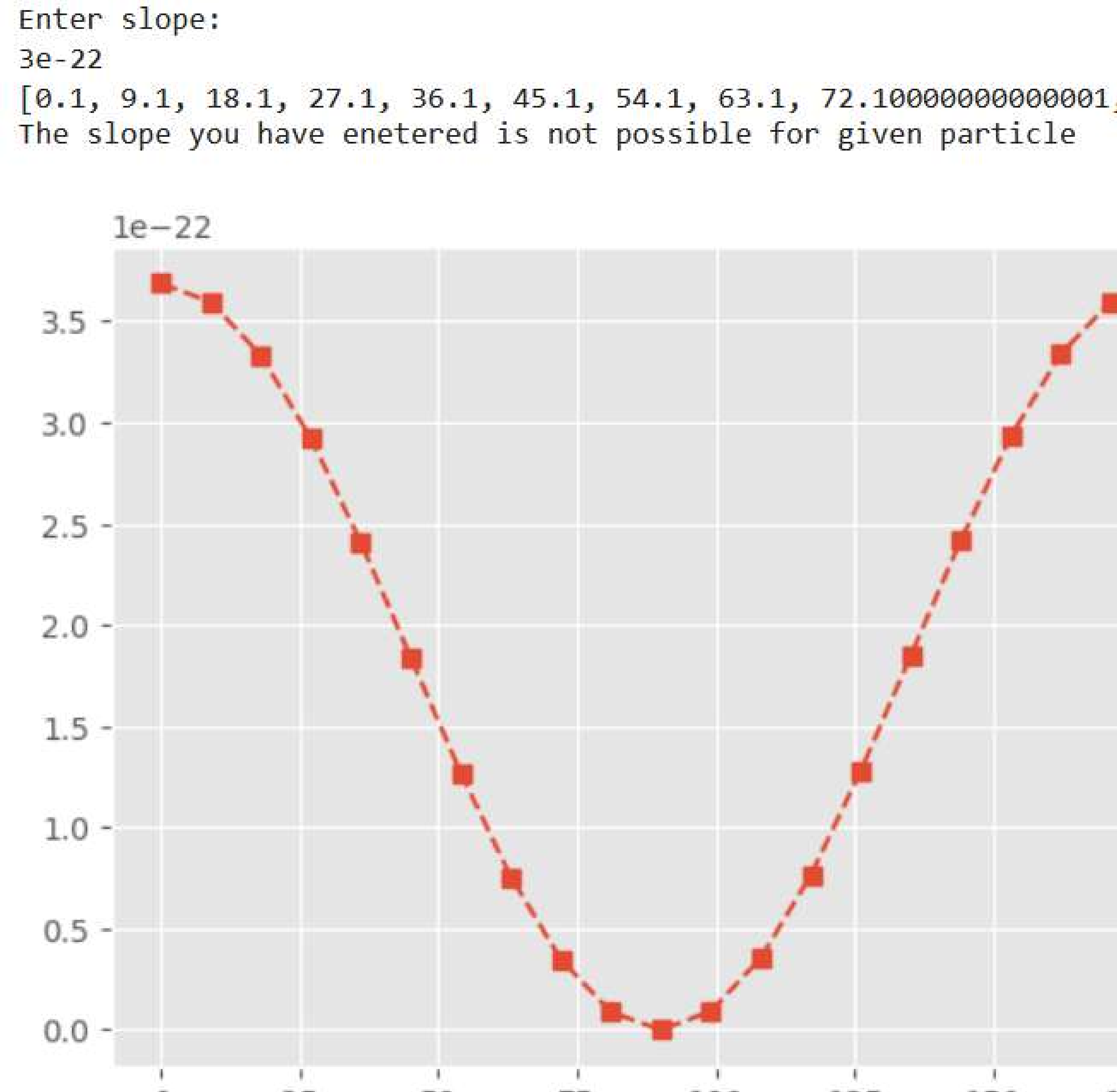
```
9.839066678401271e-25
5.828905612842495e-25
1.9079762399753423e-25
-----------The slope that matches the input in the graph are -------------

5.067979190987675e-24  at position  86
```



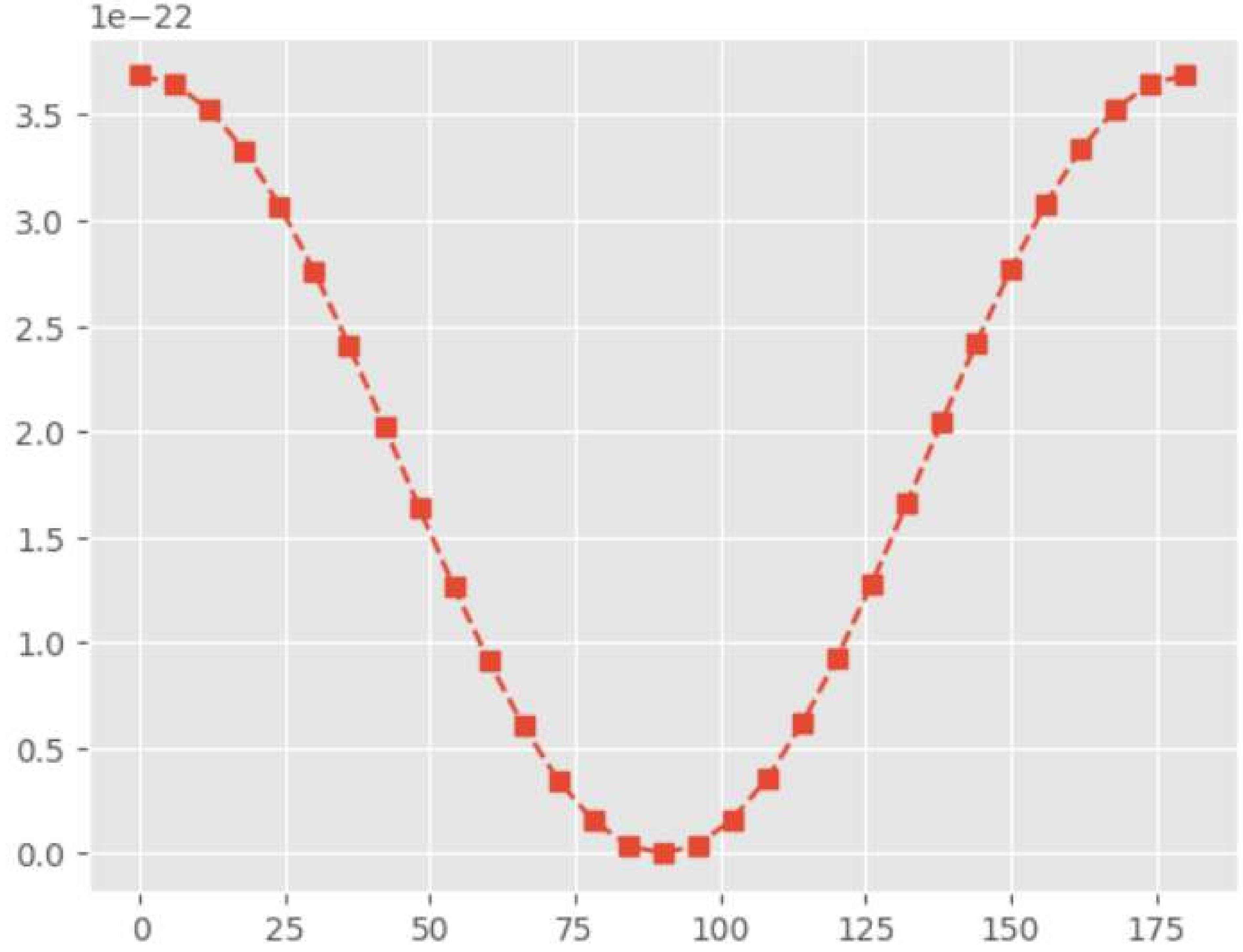
Single slope

In this mode, we are given a single slope value which is used to compare with the curve and outputs the segment which matches with it the most. If a wrong value of slope is inputted then it prompts that it is not possible for the given particle with the wavelength.



```
1] -----------The slope that matches the input in the graph are -------------

   3.1909037317844617e-24  at position  28
```

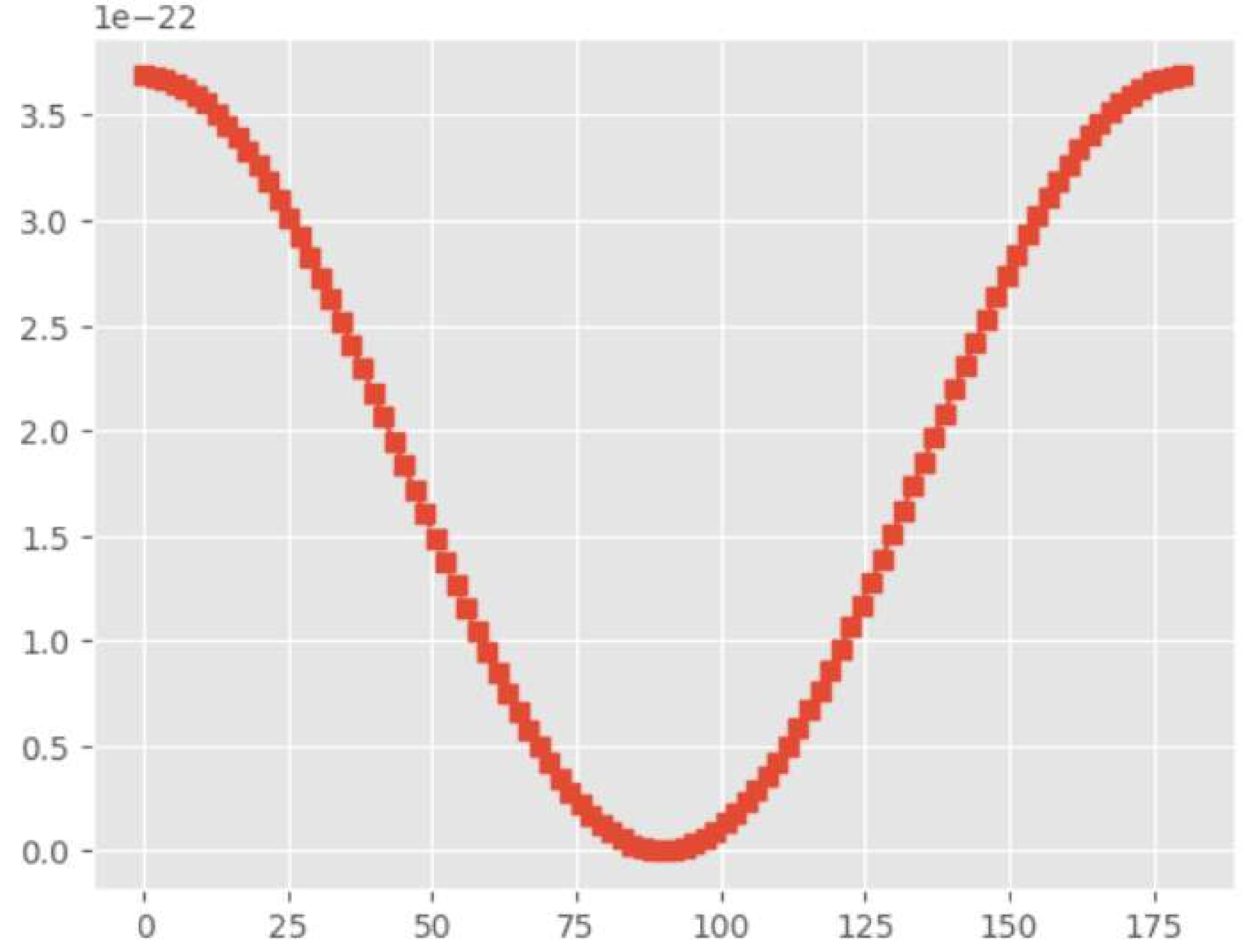


Here we clearly see that our given input of 3e-24 matches with the slope at position 28

Instantaneous slope of a single angle

```
9.839066678401271e-25
5.828905612842495e-25
1.9079762399753423e-25
-----------The slope that matches the input in the graph are ------------

5.067979190987675e-24  at position  86
```



Given a single angle as input, it finds out the instantaneous slope for that angle and compares the curve and predicts where it matches with the most.

Dual Angle

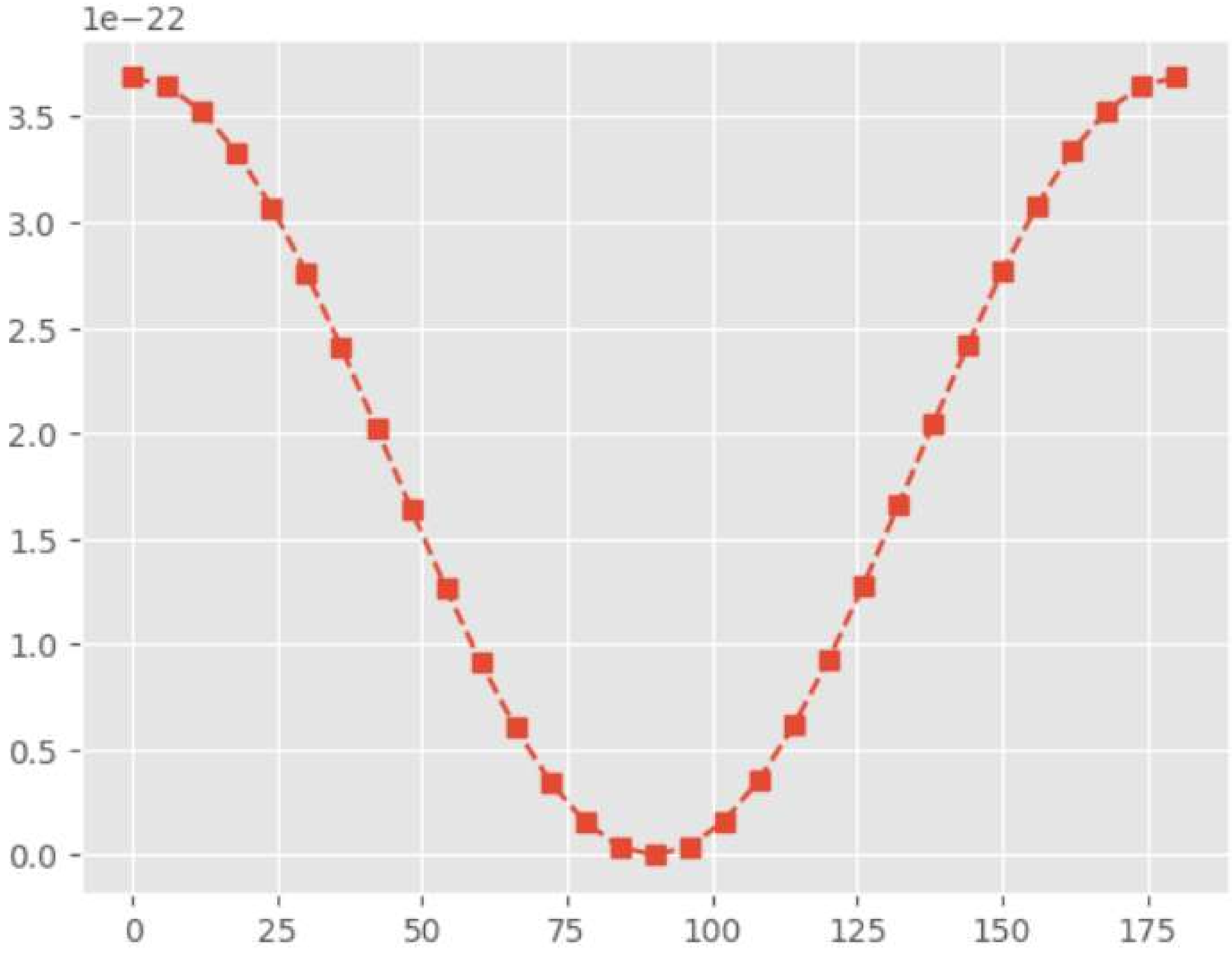
It takes two values of angle as input and uses the model to find out the intensity values for the respective angles and uses the data to calculate to average of the two slopes which is then compared with the curve to output the segment which matches the best with it in the curve.

```
Enter no. of divisions of graph
30
Select a mode:
1.Slope
2.Angle(single)
3.Angle(dual)
4.Slope(dual)
Enter mode (1,2,3,4):
3
Enter larger angle:
30
Enter smaller angle:
20
Slope we are looking for is :  [-4.90359406e-24]
[0.1, 6.1000000000000005, 12.100000000000001, 18.1, 24.1, 30.1, 36.1, 4
-----------The slope for the various divisions in the graph are -------

-6.934277732381143e-25
-2.0053981399480677e-24
-3.2297229849743235e-24
-4.312893437626867e-24
-5.207569751496303e-24
-5.8746502785331656e-24
-6.284980398698544e-24
-6.420626716758789e-24
-6.2756608377113885e-24
-5.856418466094243e-24
-5.181222505295065e-24
-4.2795822587504904e-24
-3.190903731784453e-24
```

```
1.962767400084509e-24
6.600364330201139e-25
-----------The slope that matches the input in the graph are ------------

-5.181222505295065e-24  at position  11
```



Dual slopes

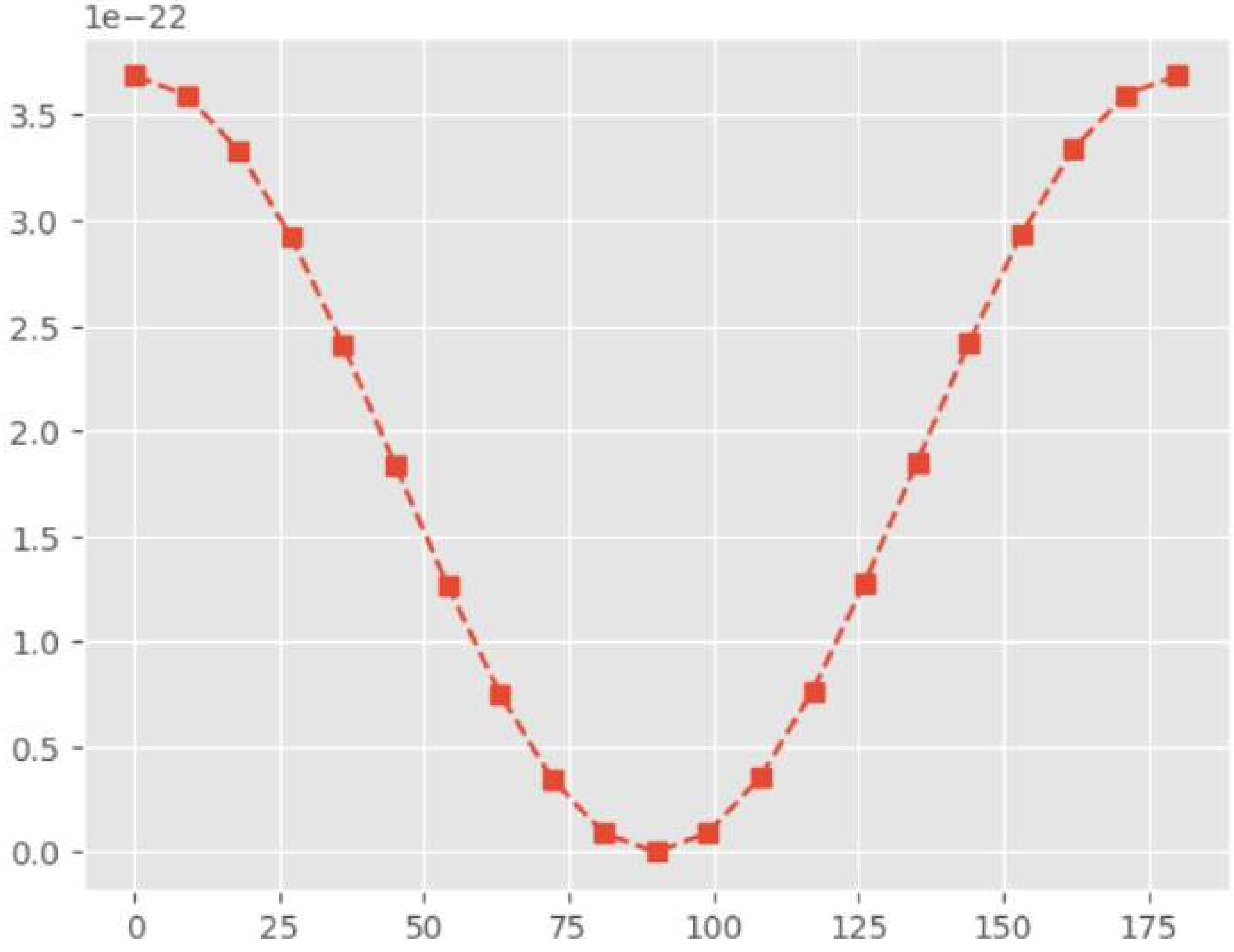
We also allowed for entering two different slopes, and then the average of two slopes was taken and then comparison was made to check if that slope is possible for our particle dataset with a given size.

```
Enter no. of divisions of graph
20
Select a mode:
1.Slope
2.Angle(single)
3.Angle(dual)
4.Slope(dual)
Enter mode (1,2,3,4):
4
enter slope 1:
1e-24
enter slope 2:
3e-24
[0.1, 9.1, 18.1, 27.1, 36.1, 45.1, 54.1, 63.1, 72.10000000000001, 81.10000
-----------The slope for the various divisions in the graph are ----------

-1.0241982983062914e-24
-2.928167633800712e-24
-4.545507519555155e-24
-5.7179014588824024e-24
-6.330587364452187e-24
-6.323591270993627e-24
-5.697598004879103e-24
-4.513884148547429e-24
-2.888319861675779e-24
-9.800267026354993e-25
1.0241982983062902e-24
```

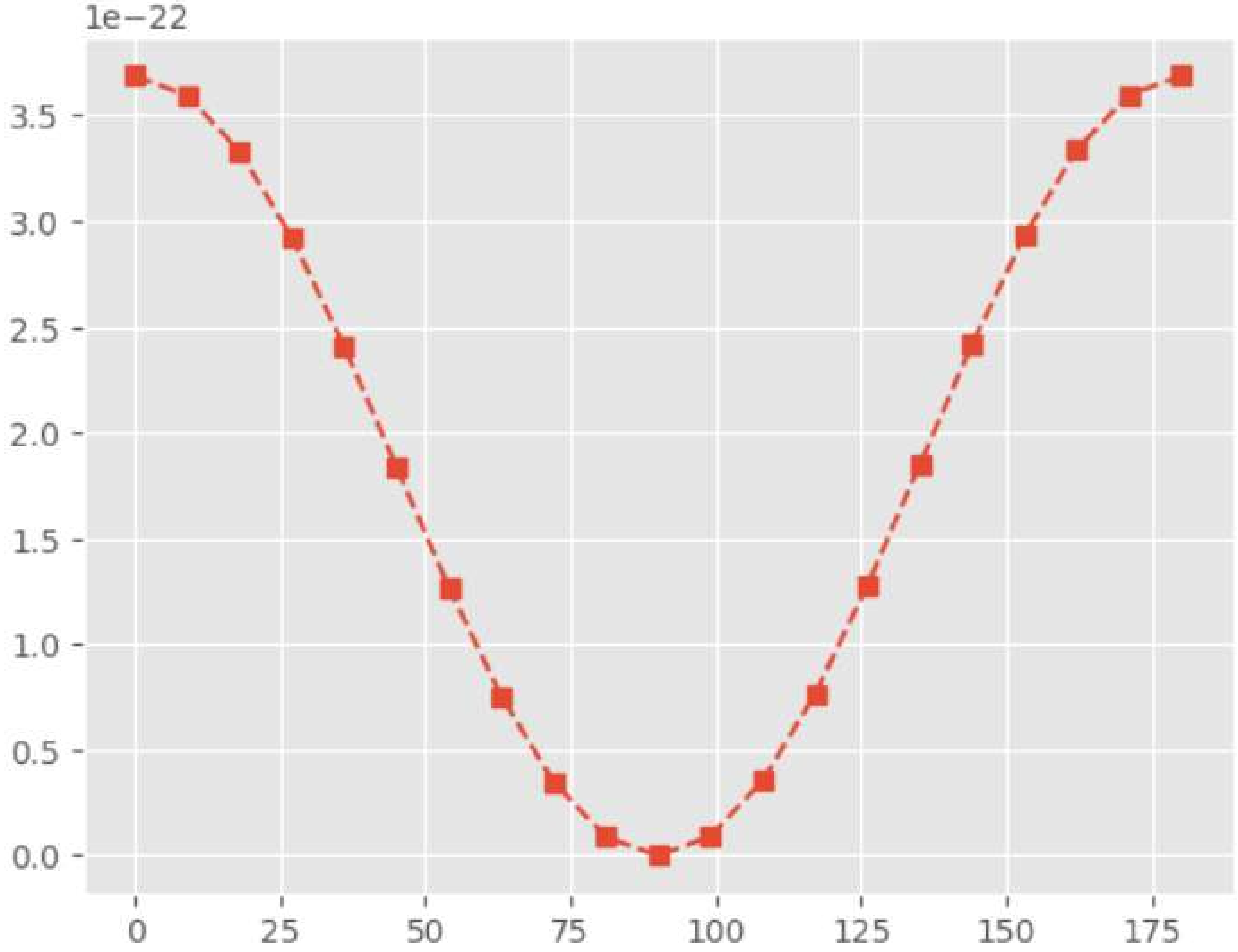
```
9.911643810056775e-25
----------The slope that matches the input in the graph are ------------

2.888319861675775e-24  at position  19
```



```
9.911643810056775e-25
----------The slope that matches the input in the graph are ------------

2.888319861675775e-24  at position  19
```



THANK YOU

Aerosol Scattering

TEAM MEMBERS

Kruttichhwas Pradhan (B22EE091)
Aditya Kumar(B22ME004)
Aditya Giri(B22MT004)
Sonal Raj(B22ME062)
Anchitya Kumar(B22BB009)
Suhani Agrawal(B22ME063)