Hall Effect in Semiconductors

Şükrü Çağlar, Ege Adıgüzel, Semih Dülger

PHYS443 Fall 2025 Experiment Report, 05.12.2025

Abstract

In this experiment diffraction behavior of electron from a graphite crystal is investigated using a circular vacuum tube with florescent material. The interference patterns on the florescent screen takes the shape of disks due to the symmetry of the system. First two peaks were observable in the range of this system and contained the constructive interference due to lattice constants d_{10} and d_{11} of the crystal. From the radius of this two disks, d_{10} and d_{11} are experimentally found to be $d_{10} = (0.172 \pm 0.049)$nm and $d_{11} = (0.127 \pm 0.026)$ nm, respectively. Real values for lattice constants are $d_{10} = 0.213$ nm and $d_{11} = 0.123$ nm [9] which puts both experimental finding within the range of 1σ.

I Introduction and Theory

History and Motivation: The wave nature of matter, hypothesized by de Broglie in 1924 ($\lambda = h/p$) [2], fundamentally expanded quantum mechanics into classical realm. This theory was confirmed in 1927 when Davisson and Germer, and independently G.P. Thomson, observed electron diffraction, demonstrating that massive particles exhibit interference characteristic of waves [4, 8]. This experiment is designed to observe this fenomena by accelerating electrons through polycrystalline graphite to generate Debye-Scherrer rings. [5] By analyzing these patterns with Bragg's Law, we verify the de Broglie relationship and experimentally determine the lattice constants of graphite.

Theory: According to de Broglie law, massive particles has a wavelength proportional to its momentum as:

$$\lambda = \frac{h}{p} \tag{1}$$

One of the ways to observe the affects of this wavelength in classical sense is investigating the diffraction event of a wave from a crystal lattice. The angle between the constructive peaks from a such diffraction event is given by Bragg's Law:[1]

$$n\lambda = 2d\sin(\theta) \tag{2}$$

Where d is the lattice constant of the crystal and n is the order of constructive peak. Although many classical objects have such massive momentum that corresponding wavelength is very small to observe distinguishable diffraction pattern, particles with small mass such as electron exhibits diffraction behavior.

In this experiment, electrons are accelerated by a potential difference between electron gun and the crystal. Their momentum is related to the accelerating potential as:

$$q_e V = \frac{p^2}{2m_e} \tag{3}$$

Plugging this in to the (1), we obtain:

$$\lambda = \frac{h}{\sqrt{2q_e m_e V}} \tag{4}$$

For the wavelength of the electron. We can then relate this to (2) with $n = 1$,we obtain the relation between diffraction angle and the acceleration potential:

$$\sin(\theta) = \frac{h}{d\sqrt{8q_e m_e}} \cdot \frac{1}{\sqrt{V}} \tag{5}$$

We can finally, treat this relation as a linear relation($y = slope \cdot x$) between $\sin(\theta)$ and $\frac{1}{\sqrt{V}}$ and interpret d as:

$$slope = \frac{h}{d\sqrt{8q_e m_e}}$$
$$d = \frac{h}{slope\sqrt{8q_e m_e}}$$

As a final note graphite lattice structure has **6 different lattice constants** due to its geometry, two of which creates peaks within the range of our experimental setup. As a result we expect to observe **two first order peaks** for $d_{11} = 0.123nm$ and $d_{10} = 0.213nm$.

II Setup and Method

Setup: The experimental setup consists of

- Electron Diffraction Tube
- 5-kV DC Power Supply
- Measuring tape
- 0-50 V Power Supply to fine tune electron flux
- Microammeter $0 - 50\,\mu A$

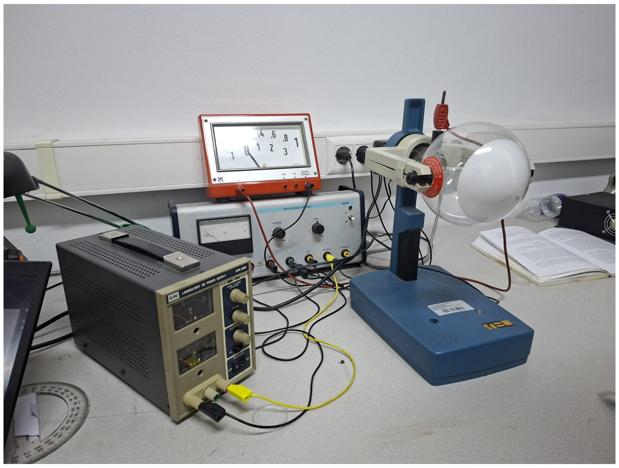

Figure 1: Electron diffraction tube with florescent screen

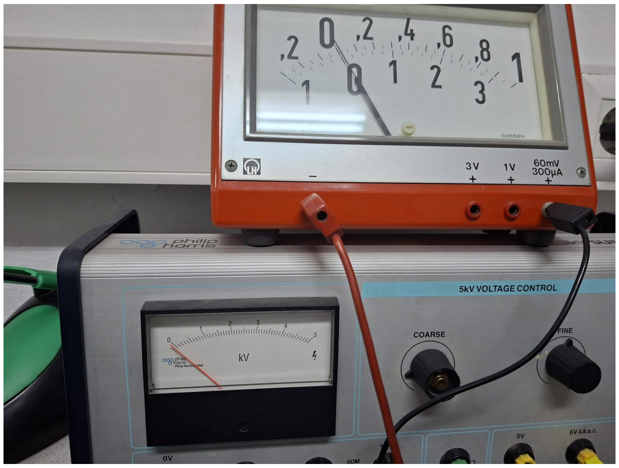

Figure 2: Power supply for the accelerating voltage

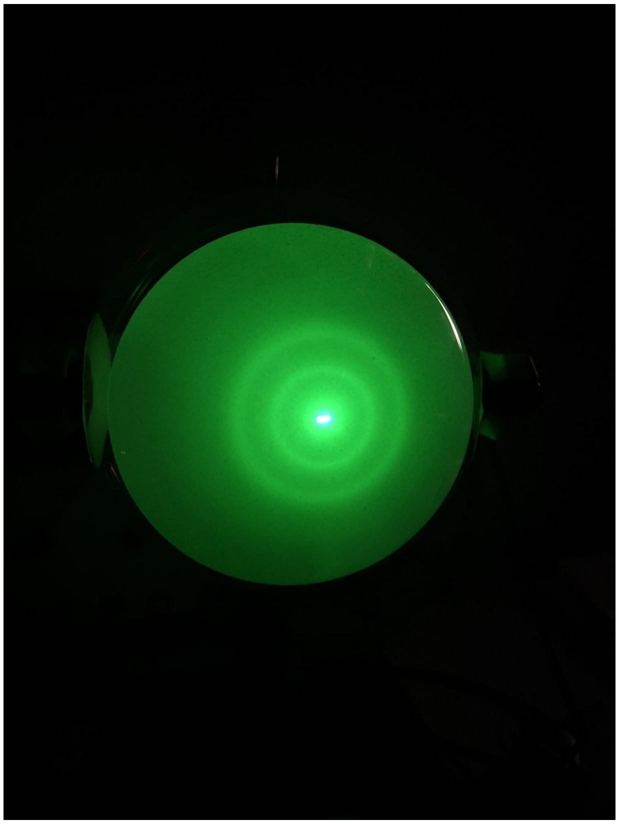

Figure 3: First two ring visible on the screen

Method: The procedure for this experiment using electron diffraction tube with graphite crystal is as follows:

1. **Calibrating the screen:** After a calibration voltage of around 3500 Volts is set, electron flux is fine tuned to make the pattern on the screen more visible. This is done by applying a small stopping potential to the crystal to filter the energetic electrons and obtain an even distribution. The current for this potential is recorded to adjust to power supply for each batch of accelerating voltage measurements.

2. **Data Taking:** For a starting value of accelerating potential 2900 Volts, inner and outer diameters of the diffraction ring is measured with measuring tape, both horizontally and vertically. Mean value and standard deviation of those measurements is then taken for the calculations. This process is repeated for both rings and measurement is repeated for 10 accelerating potentials between 2900-5000 Volts.

III Data

Dataset consists of 20 datat points, 10 datapoints for each ring. Each datat points contains 2 inner and 2 outer radius measurement with corresponding accelerating voltage. Below is the tables for $d_1 0$ ring and $d_1 1$ ring.

Voltage (V)	Vertical Radius		Horizontal Radius	
	In (cm)	Out (cm)	In (cm)	Out (cm)
2900	1.6	1.8	1.6	1.8
3200	1.4	1.6	1.3	1.5
3500	1.3	1.5	1.3	1.5
3800	1.3	1.5	1.2	1.4
3900	1.2	1.4	1.2	1.4
4000	1.3	1.5	1.2	1.4
4100	1.2	1.4	1.2	1.4
4300	1.2	1.4	1.1	1.3
4600	1.2	1.4	1.1	1.3
5000	1.1	1.3	1.1	1.3

Table 1: Data points for the first ring

Voltage (V)	Vertical Radius		Horizontal Radius	
	In (cm)	Out (cm)	In (cm)	Out (cm)
2900	2.7	2.9	2.7	2.9
3200	2.6	2.8	2.3	2.5
3500	2.4	2.6	2.3	2.5
3800	2.3	2.5	2.2	2.4
3900	2.3	2.5	2.2	2.4
4000	2.3	2.5	2.3	2.5
4100	2.2	2.4	2.1	2.3
4300	2.2	2.4	2.1	2.3
4600	2.2	2.4	2.1	2.3
5000	2.0	2.2	2.0	2.2

Table 2: Data points for the second ring

Values have $0.1cm$ and $100V$ of uncertainty on radius and voltage measurements, respectively. Those are discussed in more detailed in the analysis part. Also, current for the filtering (stopping) voltage is set to be 42 µA for all measurements.

IV Analysis

de Broglie Wavelength:

It is useful to remind the variance of a multi variable function is given as:

$$\sigma_f^2 = \left(\frac{\partial f}{\partial x}\sigma_x\right)^2 + \left(\frac{\partial f}{\partial y}\sigma_y\right)^2 + ...$$
$$\text{for}: \ f = f(x, y, ...)$$

From above relation and (4), the uncertainty in de Broglie wavelength λ is given as:

$$\sigma_\lambda = \frac{h\sigma_V}{\sqrt{8q_e m_e V^3}} = \lambda\frac{\sigma_V}{2V}$$

where $\sigma_V = 100\ Volts$. We can than calculate the corresponding de Broglie wavelength for each accelerating potential with its uncertainty as:

Voltage (V)	λ (nm $\times 10^{-2}$)	σ_λ (nm $\times 10^{-4}$)
2900	2.2774	3.9266
3200	2.1680	3.3876
3500	2.0730	2.9615
3800	1.9895	2.6178
3900	1.9639	2.5178
4000	1.9391	2.4239
4100	1.9154	2.3358
4300	1.8703	2.1747
4600	1.8083	1.9655
5000	1.7344	1.7344

Table 3: de Broglie wavelength λ

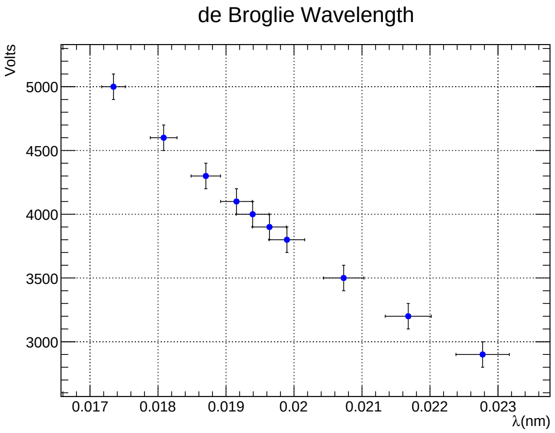

Figure 4: Wavelength vs. Accelerating potential

Calculating theta from radius:

Theta values are not directly measured in this experiment. Instead, we will derive them from the radius of the rings using the geometry of the apparatus.

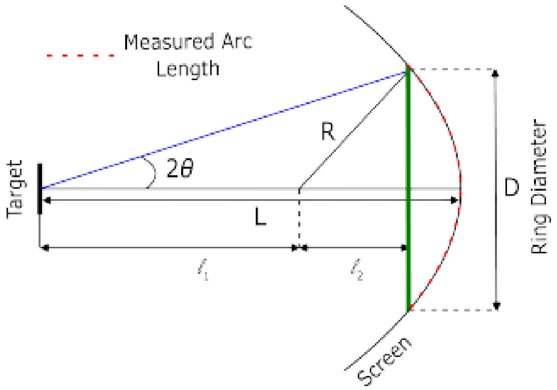

Figure 5: Geometry of the system

Relation between θ and the arc-radius $s = D/2$ is given as:

$$r = R\sin\left(\frac{s}{R}\right)$$
$$l_2 = R\cos\left(\frac{s}{R}\right)$$
$$\theta = \frac{1}{2}\arctan\left(\frac{r}{L-(r-l_2)}\right)$$

Remembering the variance relation, the uncertainty in theta is:

$$\sigma_\theta = \sqrt{\left(\frac{\partial\theta}{\partial s}\sigma_s\right)^2 + \left(\frac{\partial\theta}{\partial L}\sigma_L\right)^2 + \left(\frac{\partial\theta}{\partial R}\sigma_R\right)^2}$$

Final formula for σ_θ is not given here to keep the integrity of the report. Refer to the Appendix on θ error propagation for more detailed information. Also one can refer to data list for all measured and calculated values with their uncertainties.

Fitting the data and calculating d:

As stated in (5), we can find a relation between $\sin\theta$ and $1/\sqrt{V}$ as:

$$\sin(\theta) = \frac{h}{d\sqrt{8q_e m_e}} \cdot \frac{1}{\sqrt{V}}$$

where slope is given as:

$$slope = \frac{h}{d\sqrt{8q_e m_e}}$$

linear fit results for both rings are as follows:

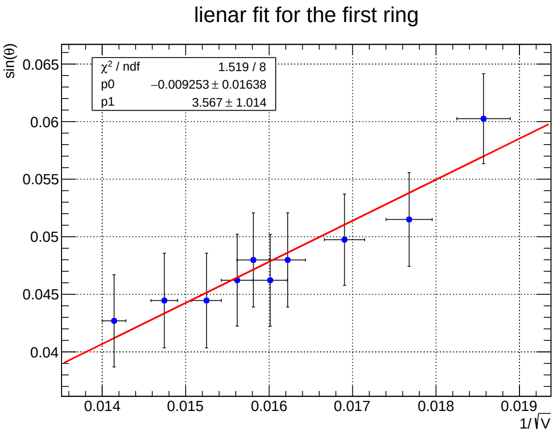

Figure 6: Fit for second ring. Corresponds to d_{10}. Slope is given by p_1.

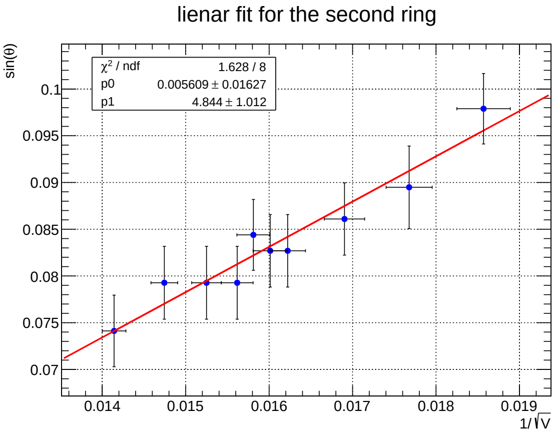

Figure 7: Fit for second ring. Corresponds to d_{11}. Slope is given by p_1.

Note that uncertainties in $\sin\theta$ and $1/\sqrt{V}$ are:

$$\sigma_{1/\sqrt{V}} = \frac{1}{2V^{3/2}} \cdot \sigma_V$$
$$\sigma_{\sin(\theta)} = \cos(\theta) \cdot \sigma_\theta$$

From the slope of both fits d can be calculated as:

$$d = \frac{h}{slope\sqrt{8q_e m_e}}$$
$$\sigma_d = \frac{d}{slope}\sigma_{slope}$$

Which is found to be (0.172 ± 0.049)nm for d_{10} and (0.127 ± 0.026) nm for d_{11}. Considering the real values for d are, $d_{10} = 0.213$ nm and $d_{11} = 0.123$ nm, deviation in our findings are:

$$\Delta d_{10} = \left| \frac{0.172 \text{ nm} - 0.213 \text{ nm}}{0.213 \text{ nm}} \right| \times 100\% \approx 19.25\%$$
$$\Delta d_{11} = \left| \frac{0.127 \text{ nm} - 0.123 \text{ nm}}{0.123 \text{ nm}} \right| \times 100\% \approx 3.25\%$$

which corresponds to a Z-Score of:

$$Z_{10} = \frac{|0.172 \text{ nm} - 0.213 \text{ nm}|}{0.049 \text{ nm}} \approx 0.84$$
$$Z_{11} = \frac{|0.127 \text{ nm} - 0.123 \text{ nm}|}{0.026 \text{ nm}} \approx 0.15$$

V Results and Conclusion

Experimental results for the graphite lattice spacings, d_{10} and d_{11}, aligns strongly with the theoretical values, with both results demonstrating less than a 1 σ deviation from accepted literature values. d_{11} spacing, which deviated by 0.15σ proves a strong validation of the results. Although still acceptable, the larger deviation for d_{10} (0.84σ) suggests an error factor that is present in both rings but more visible in the first ring, slightly affecting the measurement of the first ring radius. The most obvious suspect of errors are:

1. **Radius measurement:** The hand measurement of the **radius being not as precise and only resolves up to** $1mm$**.** Which makes possible to loose precision and miss small radius changes to the already small first ring when we change the accelerating potential.

2. **Variance in electron energy:** Not all the electrons are accelerated to the same speed despite the filtering potential. As a result the de Broglie wavelength of the electron beam is is not uniform but spreads to some extend. This is the fundamental reason why we see 2-D loops with thickness instead of a line drawing a circle. As a result our findings for the radius is an estimation of where would this line be and contains **uncertainty** which becomes more significant for **small measurements**.

Overall, the experiment confirms the wave-particle duality of electrons and provides an accurate measurement of the crystal structure of graphite. Some improvements for the future addressing the above problems and some more may be:

1. digital image analysis in place of hand measurements
2. more number of data points to better estimate the linear relation
3. More accurate measurements of the apparatus parameters such as L and R values to improve conversion from radius to θ .

.

Further questions:

Why we are observing rings instead of other shapes:

That is because the graphite crystal has a polycrystalline structure, meaning it does not consist of one big crystal but many crystals oriented in all directions. As a result, electrons spread to all possible direction with a given diffraction angle θ. This creates a circle.

Why we cannot observe higher-order diffraction maxima:

According to (2), $\sin(\theta) \propto n\lambda$ which for higher n, puts theta values out of the range of our screen. For the case of other 4 lattice constants of graphite crystal we did not measure, problem is with intensities of electrons in those peaks instead of the angle. 10 and d_{11} corresponds to the most densely packed atomic planes, making them the most intense and thus most easily visible diffraction rings.

Why there is a factor of 2 in front of $d\sin(\theta)$ in Bragg's law:

The factor of 2 corresponds to the total path difference $2(d\sin(\theta)$ of two interfering waves resulted from entering and exiting to the second layer of the graphite.

Why the angle shown in fig. 5 is 2θ instead of θ:

Theta measures the Bragg's angle which measures the angle between the incident beam and the crystal. Total diffraction angle between incident beam and the diffracted beam is 2 times this value.

What are the affects of our non-relativistic approach:

Eq. (1) is a non-relativistic approximation to the real de Broglie wavelength formula:

$$\lambda_{\text{rel}} = \frac{h}{\sqrt{2m_e eV}} \cdot \frac{1}{\sqrt{1 + \frac{eV}{2m_e c^2}}}$$

Which is smaller than the approximated value. For 5000 V of accelerating potential, this corresponds to %0.24 **difference** between the approximated and real values.

References

[1] W. L. Bragg. “The Diffraction of Short Electromagnetic Waves by a Crystal”. In: *Proceedings of the Cambridge Philosophical Society* 17 (1913), pp. 43–57.

[2] Louis de Broglie. “Recherches sur la théorie des quanta”. In: *Annales de Physique* 10 (1925), pp. 22–128.

[3] capta1Nemo. *PHYS443 Experiments GitHub Repository*. `https://github.com/capta1Nemo/443experiments.git`.

[4] Clinton Davisson and Lester H. Germer. “Diffraction of electrons by a crystal of nickel”. In: *Physical Review* 30.6 (1927), p. 705.

[5] P. Debye and P. Scherrer. “Interferenzen an regellos orientierten Teilchen im Röntgenlicht. I”. In: *Physikalische Zeitschrift* 17 (1916), pp. 277–283.

[6] E. Gülmez. *Advanced Physics Experiments*. Boğaziçi University, 1997.

[7] Charles Kittel. *Introduction to Solid State Physics*. 8th. Hoboken, NJ: John Wiley & Sons, 2005.

[8] George P. Thomson. “Diffraction of cathode rays by a thin film”. In: *Nature* 119.3007 (1927), p. 890.

[9] R. W. G. Wyckoff. *Crystal Structures*. Vol. 1. New York: Interscience Publishers, 1963.

Appendix

A.1 Error Propagation for theta

To determine the uncertainty σ_θ, we propagate the errors from the measured quantities s, L, and R. The variance is given by:

$$\sigma_\theta^2 = \left(\frac{\partial\theta}{\partial s}\sigma_s\right)^2 + \left(\frac{\partial\theta}{\partial L}\sigma_L\right)^2 + \left(\frac{\partial\theta}{\partial R}\sigma_R\right)^2 \tag{6}$$

To simplify the differentiation, we define the argument of the arctangent function as u:

$$u = \frac{r}{L - l_2} \qquad \Longrightarrow \qquad \theta = \frac{1}{2}\arctan(u) \tag{7}$$

The derivative of θ with respect to any variable x is:

$$\frac{\partial\theta}{\partial x} = \frac{\partial\theta}{\partial u}\frac{\partial u}{\partial x} = \frac{1}{2(1+u^2)}\frac{\partial u}{\partial x} \tag{8}$$

We now derive the partial derivatives of u with respect to $x = \{s, L, R\}$.

A.1.1 1. Dependence on Arc Radius (s)

Firstly we need the derivatives of the coordinates r and l_2 with respect to s:

$$\frac{\partial r}{\partial s} = \cos(s/R) \tag{9}$$

$$\frac{\partial l_2}{\partial s} = -\sin(s/R) \tag{10}$$

Applying the quotient rule to $u = r/(L - l_2)$:

$$\frac{\partial u}{\partial s} = \frac{\frac{\partial r}{\partial s}(L - l_2) - r(-\frac{\partial l_2}{\partial s})}{(L - l_2)^2} = \frac{(L - l_2)\cos(s/R) - r\sin(s/R)}{(L - l_2)^2}$$

A.1.2 2. Dependence on Distance (L)

Since r and l_2 do not depend on L, $\frac{\partial u}{\partial L}$ is simply:

$$\frac{\partial u}{\partial L} = r \cdot \frac{\partial}{\partial L}(L - l_2)^{-1} = \frac{-r}{(L - l_2)^2} \tag{11}$$

A.1.3 3. Dependence on Curvature (R)

The derivatives of r and l_2 with respect to R are:

$$\frac{\partial r}{\partial R} = \sin(s/R) - \frac{s}{R}\cos(s/R) \tag{12}$$

$$\frac{\partial l_2}{\partial R} = \cos(s/R) + \frac{s}{R}\sin(s/R) \tag{13}$$

Applying the quotient rule to u for a final time:

$$\frac{\partial u}{\partial R} = \frac{(L - l_2)\frac{\partial r}{\partial R} - r(-\frac{\partial l_2}{\partial R})}{(L - l_2)^2}$$

Combining these terms, the final uncertainty σ_θ is the following unfortunate equation:

$$\sigma_\theta = \frac{\sqrt{\left[\left(L\cos\left(\frac{s}{R}\right) - R\right)\sigma_s\right]^2 + \left[\left(-R\sin\left(\frac{s}{R}\right)\right)\sigma_L\right]^2 + \left[\left(L\left(\sin\left(\frac{s}{R}\right) - \frac{s}{R}\cos\left(\frac{s}{R}\right)\right) + s\right)\sigma_R\right]^2}}{2\left(L^2 + R^2 - 2LR\cos\left(\frac{s}{R}\right)\right)} \tag{14}$$

You can check the repository [3] and specifically "calculate_theta_precise" function for code implementation of eq 6. which is a bit easier to follow.

Additional Material

Table 4: Processed Data for 1st ring (d_{10}):

V (V)	Ord.	Vertical (cm)		Horizontal (cm)		Filt. (μA)	λ (nm)		R (cm)		θ (rad)	
		In	Out	In	Out		Val ($\times 10^{-2}$)	σ ($\times 10^{-4}$)	Avg	σ	Val ($\times 10^{-2}$)	σ ($\times 10^{-4}$)
2900	1st	1.6	1.8	1.6	1.8	42	2.2774	3.9266	1.70	0.1155	6.0286	39.1360
3200	1st	1.4	1.6	1.3	1.5	42	2.1680	3.3876	1.45	0.1291	5.1521	43.8870
3500	1st	1.3	1.5	1.3	1.5	42	2.0730	2.9615	1.40	0.1155	4.9762	39.7130
3800	1st	1.3	1.5	1.2	1.4	42	1.9895	2.6178	1.35	0.1291	4.8001	44.1560
3900	1st	1.2	1.4	1.2	1.4	42	1.9639	2.5178	1.30	0.1155	4.6238	39.8990
4000	1st	1.3	1.5	1.2	1.4	42	1.9391	2.4239	1.35	0.1291	4.8001	44.1560
4100	1st	1.2	1.4	1.2	1.4	42	1.9154	2.3358	1.30	0.1155	4.6238	39.8990
4300	1st	1.2	1.4	1.1	1.3	42	1.8703	2.1747	1.25	0.1291	4.4474	44.4140
4600	1st	1.2	1.4	1.1	1.3	42	1.8083	1.9655	1.25	0.1291	4.4474	44.4140
5000	1st	1.1	1.3	1.1	1.3	42	1.7344	1.7344	1.20	0.1155	4.2708	40.0780

Table 5: Processed Data for 2nd ring (d_{11}):

V (V)	Ord.	Vertical (cm)		Horizontal (cm)		Filt. (μA)	λ (nm)		R (cm)		θ (rad)	
		In	Out	In	Out		Val ($\times 10^{-2}$)	σ ($\times 10^{-4}$)	Avg	σ	Val ($\times 10^{-2}$)	σ ($\times 10^{-4}$)
2900	2nd	2.7	2.9	2.7	2.9	42	2.2774	3.9266	2.80	0.1155	9.8047	37.8880
3200	2nd	2.6	2.8	2.3	2.5	42	2.1680	3.3876	2.55	0.2082	8.9605	59.6190
3500	2nd	2.4	2.6	2.3	2.5	42	2.0730	2.9615	2.45	0.1291	8.6202	41.1060
3800	2nd	2.3	2.5	2.2	2.4	42	1.9895	2.6178	2.35	0.1291	8.2786	41.3440
3900	2nd	2.3	2.5	2.2	2.4	42	1.9639	2.5178	2.35	0.1291	8.2786	41.3440
4000	2nd	2.3	2.5	2.3	2.5	42	1.9391	2.4239	2.40	0.1155	8.4496	38.0210
4100	2nd	2.2	2.4	2.1	2.3	42	1.9154	2.3358	2.25	0.1291	7.9356	41.6000
4300	2nd	2.2	2.4	2.1	2.3	42	1.8703	2.1747	2.25	0.1291	7.9356	41.6000
4600	2nd	2.2	2.4	2.1	2.3	42	1.8083	1.9655	2.25	0.1291	7.9356	41.6000
5000	2nd	2.0	2.2	2.0	2.2	42	1.7344	1.7344	2.10	0.1155	7.4188	38.4150

This dataset is also available at [3].

Codes

All the codes here and more can be found in the GitHub repository [3].

```python

def fit_lin(x, sx, y, sy, file_name="fit",title_name="fit", visual_scale_factor_x=1, visual_scale_factor_y=1):
    ROOT.gStyle.SetOptFit(1)
    graph = ROOT.TGraphErrors(len(x))
    for i in range(len(x)):
        graph.SetPoint(i, x[i], y[i])
        graph.SetPointError(i, sx[i], sy[i])

    fit_func = ROOT.TF1("fit_func", "[0]␣+␣[1]*x", min(x), max(x))
    fit_func.SetParameters(0, 1)  # initial guesses

    graph.Fit(fit_func)
    cal_parameters=[(fit_func.GetParameter(i),fit_func.GetParError(i)) for i in range(2)]

    starting_point=0
    if visual_scale_factor_x != 1 or visual_scale_factor_y !=1:
        print(f"Applying␣a␣visual␣scaling␣factor␣of␣{visual_scale_factor_x}␣to␣x␣and␣{visual_scale_factor_y}␣to␣x␣and␣from␣data␣points{starting_point+1}.")
        for i in range(starting_point,len(x)):
            graph.SetPointError(i, sx[i] * visual_scale_factor_x, sy[i] * visual_scale_factor_y)
    graph.SetMarkerStyle(20)
    graph.SetMarkerColor(ROOT.kBlue)
    graph.SetTitle(title_name)
    graph.GetYaxis().RotateTitle(0)
    # Move it slightly left so it doesn't hit the numbers
    graph.GetYaxis().SetTitleOffset(1.5)

    c = ROOT.TCanvas("c", "Line␣Fit", 800, 600)
    graph.Draw("AP")
    fit_func.Draw("same")
    c.Update()
    stats = graph.GetListOfFunctions().FindObject("stats")
    if stats:
        stats.SetX1NDC(0.15)  # lower-left x (0 to 1)
        stats.SetY1NDC(0.75)  # lower-left y
        stats.SetX2NDC(0.45)  # upper-right x
        stats.SetY2NDC(0.87)   # upper-right y
        stats.SetTextSize(0.03)
        c.Modified()

    ROOT.gPad.SetTicks(1, 1)
    ROOT.gPad.SetGrid(1, 1)
    c.Update()
    c.SaveAs(f"figures/{file_name}.pdf")
    del c, graph, fit_func
    return cal_parameters

def fit_plot(x,sx,y,sy,file_name="fit",title_name="fit", visual_scale_factor_x=1, visual_scale_factor_y=1):
    ROOT.gStyle.SetOptFit(1)
    graph = ROOT.TGraphErrors(len(x))
    for i in range(len(x)):
        graph.SetPoint(i, x[i], y[i])
        graph.SetPointError(i, sx[i], sy[i])

    starting_point=0
    if visual_scale_factor_x != 1 or visual_scale_factor_y !=1:
        print(f"Applying␣a␣visual␣scaling␣factor␣of␣{visual_scale_factor_x}␣to␣x␣and␣{visual_scale_factor_y}␣to␣x␣and␣from␣data␣points{starting_point+1}.")
        for i in range(starting_point,len(x)):
            graph.SetPointError(i, sx[i] * visual_scale_factor_x, sy[i] * visual_scale_factor_y)
    graph.SetMarkerStyle(20)
    graph.SetMarkerColor(ROOT.kBlue)
    graph.SetTitle(title_name)
    graph.GetYaxis().RotateTitle(0)
    # Move it slightly left so it doesn't hit the numbers
    graph.GetYaxis().SetTitleOffset(1.5)

    c = ROOT.TCanvas("c", "Line␣Fit", 800, 600)
    graph.Draw("AP")
    c.Update()

    ROOT.gPad.SetTicks(1, 1)
    ROOT.gPad.SetGrid(1, 1)
    c.Update()
    c.SaveAs(f"figures/{file_name}.pdf")
    del c, graph

x_1st, sx_1st, y_1st, sy_1st = prepare_fit_data(df_1st, sigma_V)
x_2nd, sx_2nd, y_2nd, sy_2nd = prepare_fit_data(df_2nd, sigma_V)

print("---␣Fitting␣1th␣Order␣---")
params_1st = fit_lin(x_1st, sx_1st, y_1st, sy_1st,
                     title_name="lienar␣fit␣for␣the␣first␣ring;␣1/#sqrt{V};sin(#theta)",
                     file_name="1st␣Order(10)")
print("\nFit␣Results␣(Intercept,␣Slope):")
print(f"1st␣Order:␣{params_1st}")
print("\n---␣Fitting␣2nd␣Order␣---")
params_2nd = fit_lin(x_2nd, sx_2nd, y_2nd, sy_2nd,
                     title_name="lienar␣fit␣for␣the␣second␣ring;␣1/#sqrt{V};sin(#theta)",
                     file_name="2nd␣Order(11)")

print("\nFit␣Results␣(Intercept,␣Slope):")
print(f"2nd␣Order:␣{params_2nd}")
```

Code 1: Pyroot functiones used in plotting and fitting (with their applications)

```
def get_lambda(V,V_s=None):

    coeff_meters = h / np.sqrt(2 * m_e * q)

    coeff_nm = coeff_meters * 1.0e9
    lambda_nm=[]
    for i in V:
        lambda_nm.append(float(coeff_nm / (np.sqrt(i))))

    if V_s is not None:
        d_uncertainty_nm=[]
        for i in range(len(V_s)):
            d_uncertainty_nm.append(float(lambda_nm[i] * (V_s[i] /(2*V[i]))))
        return lambda_nm,d_uncertainty_nm

    return lambda_nm

def calculate_theta_precise(s_arc, s_err, L=14.0, L_err=0.3, R=4.3, R_err=0.1):

    # 1. Vectorize inputs
    s = np.array(s_arc)
    sigma_s = np.array(s_err)

    # 2. Intermediate Variables
    phi = s / R
    sin_phi = np.sin(phi)
    cos_phi = np.cos(phi)
    r = R * sin_phi
    r2 = R * cos_phi

    # Denominator for the arctan argument
    denom = L - (R-r2)
    # The argument for arctan
    u = r / denom

    # theta = 0.5 * arctan(u)
    theta_rad = 0.5 * np.arctan(u)

    # --- Error Propagation Starts Here ---

    # Partial derivatives of phi
    dphi_ds = 1.0 / R
    dphi_dR = -s / (R**2)

    # Partial derivatives of r
    dr_ds = R * cos_phi * dphi_ds
    dr_dR = sin_phi + R * cos_phi * dphi_dR

    # Partial derivatives of r2 (l2)
    dr2_ds = -R * sin_phi * dphi_ds
    dr2_dR = cos_phi - R * sin_phi * dphi_dR

    # Partial derivatives of u = r / denom
    # du = (dr * denom - r * d(denom)) / denom^2
    # d(denom) = dL - dr2

    # du/ds
    d_denom_ds = -dr2_ds
    du_ds = (dr_ds * denom - r * d_denom_ds) / (denom**2)

    # du/dL
    d_denom_dL = 1.0
    du_dL = (0 * denom - r * d_denom_dL) / (denom**2) # = -r / denom^2

    # du/dR
    d_denom_dR = -dr2_dR
    du_dR = (dr_dR * denom - r * d_denom_dR) / (denom**2)

    # Partial derivatives of theta = 0.5 * arctan(u)
    dtheta_du = 0.5 / (1 + u**2)

    # Final Derivatives for Sigma
    dtheta_ds = dtheta_du * du_ds
    dtheta_dL = dtheta_du * du_dL
    dtheta_dR = dtheta_du * du_dR

    theta_err = np.sqrt(
        (dtheta_ds * sigma_s)**2 +
        (dtheta_dL * L_err)**2 +
        (dtheta_dR * R_err)**2
    )

    return theta_rad, theta_err

# --- Helper Function to Prepare Data ---
def prepare_fit_data(df, sigma_V):

    V = df['Voltage␣(V)'].to_numpy()
    theta = df['theta_val_rad'].to_numpy()
    theta_err = df['theta_unc_rad'].to_numpy()

    x = 1.0 / np.sqrt(V)
    # Error Propagation for x = V^(-0.5)
    # sigma_x = x * 0.5 * (sigma_V / V)
    sx = x * 0.5 * (sigma_V / V)

    y = np.sin(theta)
    # Error Propagation for y = sin(theta)
    # sigma_y = cos(theta) * sigma_theta
    sy = np.cos(theta) * theta_err
```

```python

    return x, sx, y, sy

def get_lattice_spacing(slope, slope_err=None):

    coeff_meters = h / np.sqrt(2 * m_e * q)
    coeff_nm = coeff_meters * 1.0e9
    d_nm = coeff_nm / (2 * slope)

    if slope_err is not None:
        d_err_nm = d_nm * (slope_err / slope)
        return d_nm, d_err_nm

    return d_nm
```

Code 2: Functions for λ and θ along with some helper functions