

NAME Joshua Bowley	ID jb1753	DATE 10/10/25
EXP LABEL at02 - Atomic Spectroscopy & Duane-Hunt Law	PAGE 1	

Table of Contents:

Brugg's Law:

- Background Reading	2	2
- Brugg condition equation proof	3 - 5	
- Schematic Diagrams	5 - 6	
- Conditions / Parameters of Experiment	7	
- Method		
« Apparatus Configuration	7 - 8	
« Data Collection	8	
- Results		
- Sources of uncertainty	8	
- Full data tables	9 - 10	
- Error analysis	11 - 14	
- Jupyter notebook	15 - 25	

Atomic Spectroscopy:

- Background Reading Summary	27 - 29
- Method	
« Equipment	30
« Steps	31
« Conditions / Parameters of Experiment	31
« Safety	31
« Lighting Conditions	32
« Sources of uncertainty	32 - 33
« Core Equations	33
- Data Collection	33 - 34
- Data Analysis	35 - 38
- Jupyter Notebook	39 - 46

References:

- [1] Brugg, W. L. (1914). "Diffraction of short electromagnetic waves" - Proceedings of the Cambridge Philosophical Society: Mathematical and Physical Sciences.
- [2] (2025) "Visible Light - NASA Science" - science.nasa.gov/ems/09_visiblelight/ (Accessed 07/11/2025)

NAME	ID	DATE 10/10/25
EXP LABEL		PAGE 2

Duane Hunt Law:

Background Reading:

Free electrons inside an x-ray tube are accelerated by a high voltage V to emit x-rays in a continuous spectrum. The maximum energy of the x-rays is equal to the energy of the incoming electrons. Hence:

$$\left[eV = h\nu_{\max} = \frac{hc}{\lambda_{\min}} \right] \leftarrow \text{Duane Hunt Law}$$

where the symbols have their regular meanings.

A measurement of λ_{\min} vs V therefore provides a measurement of Planck Constant h (provided e & c are known).

This experiment makes use of Bragg reflections from a crystal of known lattice parameter in order to measure λ . The Bragg condition for strong reflections of x-rays from a crystal is:

$$\left[n\lambda = 2d \sin \theta \right] \leftarrow \text{Bragg's Law}$$

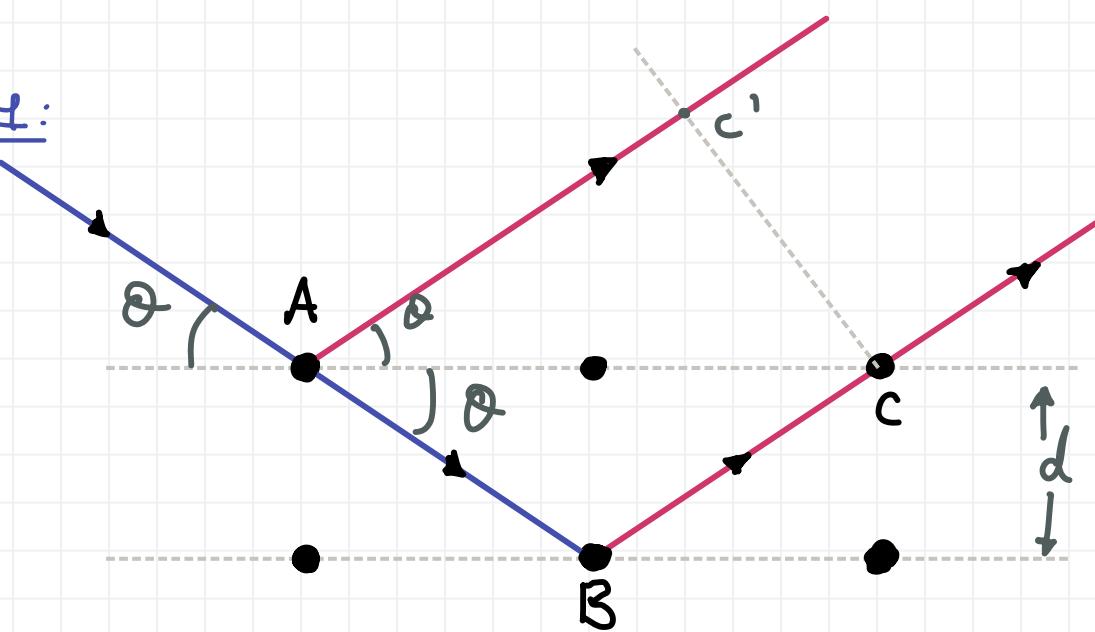
I will provide a proof below.

Brugg Condition Equation Proof:

Brugg's Approach is a Huygen's Construction for a reflected wave. [1]

Plane wave incident on planes of lattice points, with separation d , at an angle θ as shown in figure 1. Points A and C are on one plane, and B is on plane below. Points ABCC' form a quadrilateral.

Figure 1:



There is a path difference between ray reflected along AC' and other reflected along BC . This path difference is

$$(AB + BC) - (AC')$$

The two separate waves will arrive at point (infinitely far away) with the same phase and undergo constructive interference.

NAME	ID	DATE 10/10/25
EXP LABEL		PAGE 4

Only can happen if path length is an integer wavelength:

$$n\lambda = (AB + BC) - (AC')$$

where n and λ are an integer and the wavelength of incident wave respectively.

From geometry:

$$AB = BC = \frac{d}{\sin \theta}, \quad AC = \frac{2d}{\tan \theta}$$

from which follows that

$$AC' = AC \cdot \cos \theta = \frac{2d}{\tan \theta} \cos \theta$$

$$\Rightarrow AC' = \frac{2d \cos \theta}{\sin \theta} \cos \theta = \frac{2d}{\sin \theta} \cdot \cos^2 \theta$$

Putting everything together:

$$n\lambda = \left(\frac{d}{\sin \theta} + \frac{d}{\sin \theta} \right) - \left(\frac{2d}{\sin \theta} \cdot \cos^2 \theta \right)$$

$$\Rightarrow n\lambda = \frac{2d}{\sin \theta} (1 - \cos^2 \theta)$$

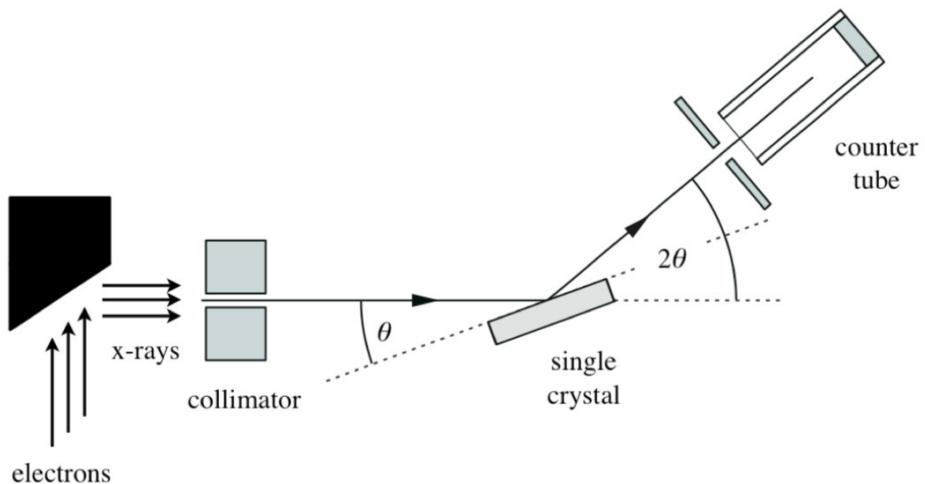
$$\Rightarrow n\lambda = \frac{2d}{\sin\theta} \cdot \sin^2\theta$$

$\therefore [n\lambda = 2ds\sin\theta] \leftarrow \text{Bragg's Law}$

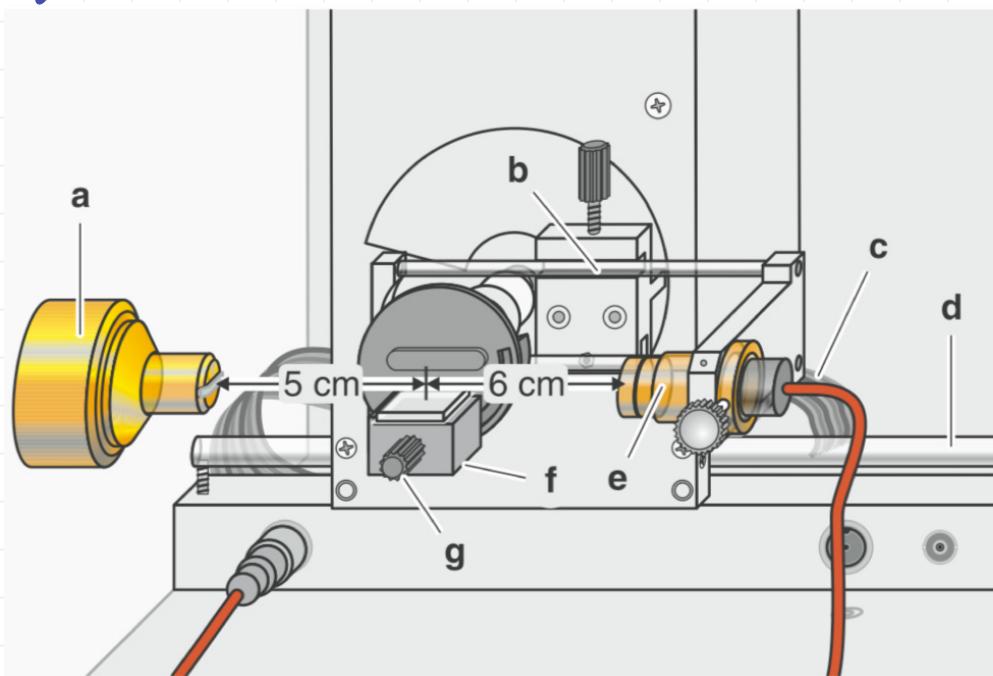
Schematic Diagrams:

11/10/25

Figure 2:

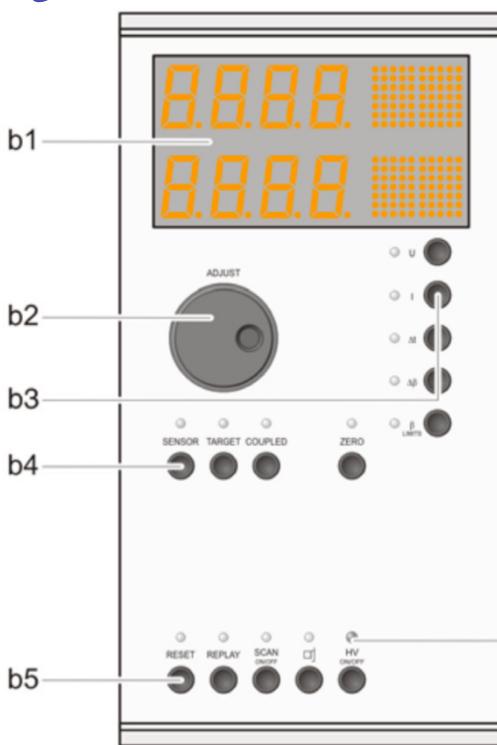


Schematic diagram of the diffraction experiment. Monochromatic x-rays are produced by Bragg-scattering from a sodium chloride crystal and their intensity is measured using a counter tube.

Figure 3:**Key:**

- a: collimator mount
- b: sensor holder
- c: goniometer ribbon cable
- d: goniometer guide rod
- e: sensor seat
- f: target holder
- g: target holder screw

¹ experiment set up for Bragg scattering x-ray monochromators.

Figure 4:**Key:**

- b1: display
- b2: adjustment dial
- b3: parameter selector keys
- b4: scan mode keys
- b5: operation keys
- b6: high-voltage indicator lamp

Figure 4: x-ray apparatus control panel

NAME	ID	DATE 11/10/25
EXP LABEL		PAGE 7

Conditions / Parameters of Experiment:

<u>Parameter:</u>	<u>Symbol:</u>	<u>Value:</u>
High Voltage [kV]	U	15, 17.5, 20, 22.5, 25, 27.5, 30, 32.5, 35
Emission Current	I	1.00 mA
Angular Step	$\Delta\beta$	0.1°
Time per step	Δt	10 s
Crystal	NaCl	100
d-spacing (for NaCl)	d	0.283 nm
Mode		Coupled (2θ)

Method:

First confirmed that sodium chloride crystal is mounted in the target holder (f) and the counter tube (e) is installed.

Then started software package X Ray.exe on connected computer.

Apparatus configuration:

On control panel, set the tube high voltage (U) to 35 kV and emission current (I) to 1.00 mA.

Set the measurement parameters:

- Measuring time per step $\Delta t = 10$ s
- Angular step width $\Delta\beta = 0.1^\circ$

Set the scan limits:

- Lower limit for target angle $\beta = 2.5^\circ$
- Upper limit for target angle $\beta = 10.0^\circ$

NAME	ID	DATE 11/10/2025
EXP LABEL		PAGE 8

Performed a 2 quick scans. First with the above settings, then one more with $\Delta t = 1$, so it was quicker. Confirmed the presence of peaks near the expected values for the Molybdenum Tube (approx 6.4° and 7.2°). This verified that the goniometer was correctly zeroed.

Data Collection:

Pressed COUPLED key to activate the 2:1 coupling model between the target crystal and sensor. (2θ coupling)

Set the lower limit of target angle to 2.5° and upper limit to 12.5°

For range of high voltages from 15 kV to 35 kV (increments of 2.5 kV inclusive, 9 sets of data), I repeated the following:

- Set the new high voltage (U) on the control panel.
- Kept emission current at $I = \pm 0.00 \text{ mA}$
- Pressed the SCAN key to start the measurement and data transfer to the PC.
- Once complete, saved the datafile as a pdf.

Once collected all datasets, reduced the high voltage and emission current to zero.

Closed the XRay software.

Results:

12/10/2025

Full table on next page.

Sources of uncertainty:

Angular measurement $\delta\theta$

Voltage measurement δV

Crystal lattice spacing δd ← In going to neglect this as it is negligible.

NAME	ID	DATE 12/10/2025
EXP LABEL		PAGE 9

X-Ray Apparatus - up to 35 kV

$\beta / {}^\circ$	R ₀ / 1/s	R ₁ / 1/s	R ₂ / 1/s	R ₃ / 1/s	R ₄ / 1/s	R ₅ / 1/s	R ₆ / 1/s	R ₇ / 1/s	R ₈ / 1/s	R ₉ / 1/s	R ₁₀ / 1/s
2.5	11.2	11	0.2	0.4	0.6	1.4	2.0	3.4	5.8	4.8	9.1
2.6	9.8	9	0.2	0.6	0.8	1.4	1.3	2.4	6.1	4.6	6.3
2.7	9.0	10	0.2	0.5	1.1	1.6	1.9	2.0	4.8	5.8	8.1
2.8	9.7	9	0.3	0.3	1.0	2.0	1.5	2.9	3.2	5.9	7.9
2.9	11.7	11	0.2	0.3	0.5	0.9	1.5	2.9	4.4	6.6	7.3
3.0	10.0	10	0.2	0.2	0.7	0.7	2.1	2.5	4.3	5.5	6.9
3.1	9.4	9	0.2	0.3	0.6	1.3	1.3	3.1	4.0	6.9	6.8
3.2	13.1	8	0.4	0.2	0.6	1.1	1.1	1.9	4.8	5.6	8.2
3.3	15.0	8	0.4	0.4	1.0	1.4	1.8	2.8	3.9	7.1	8.9
3.4	38.3	10	0.1	0.0	0.4	0.6	1.2	2.8	3.1	6.9	9.4
3.5	107.8	9	0.0	0.3	0.8	0.5	1.5	2.8	4.1	5.9	10.0
3.6	238.0	12	0.1	0.4	0.8	0.8	1.6	3.0	4.3	5.9	16.3
3.7	374.6	50	0.4	0.7	0.6	1.6	1.8	2.5	4.0	5.4	54.3
3.8	494.8	148	0.2	0.3	0.6	1.9	1.2	3.4	3.9	8.9	153.5
3.9	610.5	310	0.2	0.2	0.6	1.4	0.8	3.6	4.1	17.5	311.6
4.0	681.2	433	0.3	0.6	0.7	1.2	1.7	2.1	5.1	44.8	465.2
4.1	743.8	530	0.4	0.2	0.4	0.9	1.1	3.4	7.0	132.7	555.3
4.2	776.1	650	0.2	0.1	0.5	0.9	2.5	2.2	8.3	263.3	636.1
4.3	824.4	741	0.3	0.7	0.4	1.2	1.7	2.5	26.1	368.7	701.1
4.4	821.8	737	0.3	0.3	0.1	1.8	1.5	3.3	82.9	454.6	738.2
4.5	835.6	782	0.4	0.4	0.4	1.7	0.9	3.7	193.3	536.4	787.9
4.6	843.7	806	0.3	0.1	0.3	1.4	1.6	4.2	281.4	580.2	808.3
4.7	862.1	856	0.2	0.3	1.0	1.3	2.4	13.8	345.8	616.3	816.4
4.8	840.1	831	0.2	0.6	0.6	1.1	1.0	48.8	387.4	635.6	847.9
4.9	820.1	832	0.0	0.4	0.4	1.2	2.4	104.8	439.3	673.3	851.3
5.0	797.2	870	0.1	0.1	0.6	0.8	2.1	177.8	472.5	680.4	830.7
5.1	822.8	838	0.2	0.7	0.6	0.8	3.7	241.0	508.1	676.9	840.0
5.2	794.3	835	0.2	0.3	0.8	1.2	17.0	276.6	518.3	680.4	829.8
5.3	753.7	826	0.2	0.4	1.0	3.2	48.5	320.5	518.7	659.6	820.4
5.4	741.2	791	0.3	0.1	1.0	9.0	88.8	342.2	529.7	667.8	799.8
5.5	663.9	778	0.1	0.2	0.7	32.4	136.7	360.4	524.1	644.9	785.2
5.6	620.5	762	0.3	0.4	0.9	66.4	175.6	360.2	511.3	644.9	767.5
5.7	628.1	734	0.3	0.3	0.8	110.4	198.9	372.2	511.9	636.1	742.8
5.8	596.5	738	0.1	0.3	0.9	140.8	216.9	374.7	503.2	619.4	720.9
5.9	594.5	732	0.1	0.4	1.3	161.4	233.1	359.2	509.3	598.3	686.2
6.0	580.6	716	0.1	0.3	1.8	181.8	245.8	378.2	489.6	597.6	689.6
6.1	651.3	795	0.1	0.5	6.8	200.2	252.7	385.0	524.6	639.7	751.3
6.2	874.1	966	0.1	0.0	19.3	219.1	291.0	460.3	635.9	813.7	994.9
6.3	1025.4	1207	0.2	0.3	41.4	246.3	316.1	521.4	760.1	976.5	1221.1
6.4	970.5	1168	0.2	0.4	62.7	242.2	312.4	491.8	711.0	947.0	1167.7

NOTES

NAME	ID	DATE	12/10/2025
EXP LABEL		PAGE	10

$\beta / {}^\circ$	R ₀ / 1/s	R ₁ / 1/s	R ₂ / 1/s	R ₃ / 1/s	R ₄ / 1/s	R ₅ / 1/s	R ₆ / 1/s	R ₇ / 1/s	R ₈ / 1/s	R ₉ / 1/s	R ₁₀ / 1/s
6.5	765.7	857	0.4	0.6	87.9	228.5	270.0	421.5	579.4	710.0	851.5
6.6	584.5	669	0.0	0.2	94.9	217.5	263.6	364.6	475.4	559.1	649.3
6.7	531.2	603	0.0	0.3	112.8	214.5	254.2	334.1	437.5	505.3	584.5
6.8	490.8	554	0.2	0.5	118.4	213.3	260.3	327.1	432.1	507.9	569.7
6.9	623.7	682	0.2	0.2	116.6	218.5	259.5	367.1	494.8	602.1	701.8
7.0	1394.9	1416	0.4	0.5	124.2	260.4	339.6	561.8	864.4	1141.0	1430.0
7.1	1834.2	2100	0.1	0.7	121.1	303.7	424.0	780.5	1251.2	1661.8	2122.1
7.2	1663.6	2078	0.2	0.4	122.3	285.0	388.1	725.9	1185.3	1628.8	2058.7
7.3	1064.8	1280	0.2	0.5	123.1	240.6	284.5	501.3	730.7	1003.9	1261.7
7.4	616.1	648	0.3	2.8	117.3	190.4	230.5	319.1	420.4	543.8	650.5
7.5	426.4	450	0.1	8.3	114.9	182.6	199.8	261.7	332.1	394.4	445.0
7.6	342.1	383	0.4	12.7	115.8	174.2	194.6	238.2	276.8	335.0	387.1
7.7	319.9	338	0.2	19.7	110.7	159.8	183.2	231.2	273.8	317.1	351.6
7.8	309.6	335	0.2	26.4	105.4	149.2	170.9	213.3	267.1	293.7	363.6
7.9	295.7	357	0.4	29.5	101.5	149.7	163.0	202.8	234.1	273.3	348.6
8.0	293.4	337	0.4	31.7	95.6	142.3	154.1	191.1	227.0	260.6	339.8
8.1	276.5	353	0.3	32.2	95.1	132.8	148.7	180.6	210.0	245.4	328.4
8.2	285.2	323	0.2	37.6	87.7	123.5	134.0	166.9	198.2	233.7	327.9
8.3	271.9	302	0.1	38.7	88.7	115.4	131.3	155.4	186.4	221.0	327.5
8.4	253.7	316	0.2	39.0	84.2	94.4	123.2	151.3	173.3	229.9	302.3
8.5	253.8	274	0.3	36.1	73.1	95.4	107.8	143.3	169.9	221.0	294.0
8.6	245.5	289	0.9	39.1	73.8	82.7	111.4	137.0	154.8	218.5	288.4
8.7	248.6	284	1.9	37.1	74.9	78.2	101.8	129.4	150.5	204.3	283.4
8.8	235.6	259	3.8	33.4	68.1	78.6	95.2	116.0	136.3	214.1	268.5
8.9	228.7	259	6.0	35.3	60.1	69.4	87.0	107.1	135.5	201.5	263.4
9.0	225.3	219	7.9	32.7	56.3	65.9	86.9	96.9	123.6	206.4	256.4
9.1	214.0	254	8.1	29.1	55.8	59.7	78.8	90.9	127.5	188.0	253.3
9.2	206.6	243	9.3	27.8	49.7	56.7	70.5	83.4	129.5	184.5	232.8
9.3	193.9	209	9.3	20.9	43.6	49.6	60.6	79.5	118.6	182.6	220.1
9.4	188.9	221	8.9	19.2	40.2	42.1	56.2	70.1	110.0	168.6	212.4
9.5	179.9	215	8.6	20.9	34.2	36.8	46.0	59.9	119.0	162.4	212.2
9.6	175.1	173	8.8	21.3	29.7	35.9	43.6	56.5	113.9	156.3	202.9
9.7	167.0	196	8.9	18.7	31.3	33.2	42.4	52.8	116.9	155.7	197.1
9.8	160.9	178	7.7	17.4	27.1	31.7	38.8	52.2	109.7	152.3	184.5
9.9	158.3	185	7.4	16.7	20.2	28.0	34.7	56.9	106.5	152.5	179.0
10.0	150.7	193	6.4	14.8	23.4	27.7	36.2	56.7	100.5	146.8	176.3
10.1			6.9	14.4	20.5	23.6	29.2	55.6	100.3	137.1	179.8
10.2			6.6	10.7	18.9	21.2	27.7	54.2	99.6	141.2	164.9
10.3			7.9	10.8	16.7	18.1	22.2	61.3	101.5	133.9	165.1
10.4			5.8	9.2	15.4	18.6	23.7	55.2	99.6	132.2	156.2

$\beta / {}^\circ$	R ₀ / 1/s	R ₁ / 1/s	R ₂ / 1/s	R ₃ / 1/s	R ₄ / 1/s	R ₅ / 1/s	R ₆ / 1/s	R ₇ / 1/s	R ₈ / 1/s	R ₉ / 1/s	R ₁₀ / 1/s
10.5			5.3	10.8	12.5	15.3	19.9	59.4	98.6	123.8	150.3
10.6			3.6	9.3	12.1	14.4	19.0	56.7	89.9	121.0	157.8
10.7			6.0	8.4	14.2	13.9	19.7	60.3	95.3	119.2	147.4
10.8			3.8	7.7	14.3	12.7	23.0	57.3	93.2	116.9	146.2
10.9			4.3	8.5	12.4	11.5	24.6	62.9	88.8	114.8	141.9
11.0			3.4	6.5	9.0	10.7	28.5	63.6	89.8	110.9	142.5
11.1			4.6	6.2	10.5	10.8	29.6	62.0	89.8	115.7	146.5
11.2			3.6	6.3	10.1	9.7	33.6	60.1	92.2	117.1	141.2
11.3			3.4	5.2	7.0	9.2	34.6	58.7	85.6	112.3	134.1
11.4			2.2	3.1	5.8	6.8	30.8	60.1	85.6	103.4	129.2
11.5			3.0	3.4	5.9	6.8	33.6	54.7	80.4	103.5	122.2
11.6			1.4	2.9	6.1	4.9	31.8	56.4	80.1	100.6	120.0
11.7			2.2	2.4	4.6	4.3	32.4	57.4	82.5	101.4	118.5
11.8			1.9	1.8	3.9	5.2	33.5	52.8	77.7	99.1	116.8
11.9			1.4	1.9	4.2	7.1	35.7	58.2	72.5	96.2	117.6
12.0			1.6	1.8	3.3	8.2	33.8	55.0	73.5	87.5	112.0
12.1			0.7	1.4	3.1	9.6	34.9	51.8	71.8	89.4	117.1
12.2			0.8	1.7	2.5	13.1	35.2	49.2	71.0	93.0	110.3
12.3			0.9	1.8	3.3	14.0	35.9	53.2	67.9	86.1	121.1
12.4			1.4	1.6	2.0	15.3	34.5	52.0	68.9	79.8	112.7
12.5			0.9	2.1	4.2	18.1	34.5	53.3	77.9	98.1	131.0

NOTES

NAME	ID	DATE 13/10/25
EXP LABEL		PAGE 11

Error Analysis:

Fundamental Equations:

Duane-Hunt Law:

$$eV = \frac{hc}{\lambda_{\min}} \Rightarrow h = \frac{eV \lambda_{\min}}{c}$$

Bragg's Law ($n=2$) for x-rays:

$$\lambda = 2d \sin \theta$$

where $d = 0.283 \text{ nm}$ for NaCl crystal spacing.

Error propagation chain:

Measure $\theta \rightarrow$ calculate $\lambda \rightarrow$ calculate h

$$\therefore \delta\theta \rightarrow \delta\lambda \rightarrow \delta h$$

Uncertainty in λ from Bragg's Law:

$$\text{From } \lambda = 2d \sin \theta$$

Partial Derivative:

$$\frac{\partial \lambda}{\partial \theta} = 2d \cos \theta$$

Uncertainty:

$$\delta \lambda = \left| \frac{\partial \lambda}{\partial \theta} \right| \delta \theta = 2d \cos \theta \cdot \delta \theta$$

$$\text{Where } d = 0.283 \times 10^{-9} \text{ m}$$

$$\delta \theta = 0.1^\circ = 0.00175 \text{ rad (angular step uncertainty)}$$

NAME	ID	DATE 13/10/2025
EXP LABEL		PAGE 12

Linear Regression Method:

For each voltage V :

- 1) Take the linear region before the first peak in the intensity vs angle plot
- 2) Perform linear regression: $y = mx + c$
- 3) Find the x -intercept where $y = 0$: $\theta_{\min} = -\frac{c}{m}$
- 4) Convert to wavelength: $\lambda_{\min} = \frac{c}{d \sin \theta_{\min}}$

Then for all voltages:

- 5) Plot eV against $\frac{c}{\lambda_{\min}}$
- 6) Linear Regression gives slope = h

Uncertainty in θ_{\min} :

From linear regression $y = mx + c$, the x intercept is:

$$\theta_{\min} = -\frac{c}{m}$$

Partial derivatives:

$$\frac{\partial \theta_{\min}}{\partial c} = -\frac{1}{m}, \quad \frac{\partial \theta_{\min}}{\partial m} = \frac{c}{m^2}$$

Uncertainty in regression:

$$\delta \theta_{\min} = \sqrt{\left(\frac{\partial \theta_{\min}}{\partial c} \cdot \delta c \right)^2 + \left(\frac{\partial \theta_{\min}}{\partial m} \cdot \delta m \right)^2}$$

NAME	ID	DATE 13/10/2025
EXP LABEL		PAGE 13

$$\delta \theta_{\min} = \sqrt{\left(\frac{\delta c}{m}\right)^2 + \left(\frac{c}{m^2} \delta m\right)^2}$$

Final Planck's Constant Uncertainty:

$$\text{Plot } y = eV \text{ vs } x = \frac{c}{\lambda_{\min}}$$

Linear regression through origin:

$$y = h x$$

Slope uncertainty from weighted linear regression:

$$\delta h = \sqrt{\frac{1}{\sum w_i x_i^2}}$$

$$\text{where weights } w_i = \frac{1}{(\delta y_i)^2}$$

Uncertainty in x -values:

$$\delta x = \delta \left(\frac{c}{\lambda_{\min}} \right) = \frac{c}{\lambda_{\min}^2} \delta \lambda_{\min}$$

Uncertainty in y values:

$$\delta y = \delta(eV) = e \cdot \delta V$$

NAME	ID	DATE 13/10/2025
EXP LABEL		PAGE 17F

Experimental Parameters:

$$d = 0.283 \times 10^{-9} \text{ m}$$

$$\delta\theta = \pm 0.1^\circ = \pm 0.00175 \text{ rad} \text{ (angular step uncertainty)}$$

$$\delta V = \pm 0.1 \text{ kV} = \pm 100 \text{ V} \text{ (voltage measurement uncertainty).}$$

Duane–Hunt Law: X-ray Bremsstrahlung Analysis

Date: 12/11/2025

Author: Joshua Bowley

Aim

This notebook analyses X-ray bremsstrahlung spectra taken at different tube voltages.

The goals are:

- to extract the **cut-off wavelength** (λ_{\min}) for each spectrum using Bragg's law,
- to propagate angular and fitting uncertainties into (λ_{\min}),
- to perform a **weighted linear regression** of $(1/\lambda_{\min})$ versus tube voltage (V),
- to obtain an experimental value for **Planck's constant** (h) and compare it with the accepted value.

0. Experimental parameters

The experiment uses Bragg reflection from the (100) face of NaCl.

Key relations:

- **Duane–Hunt law**

$$\$ eV = h\nu_{\max} = \frac{hc}{\lambda_{\min}} \quad \Rightarrow \quad \frac{1}{\lambda_{\min}} = \frac{e}{hc} V \$$$

- **Bragg's law (first order, ($n = 1$))**

$$\$ \lambda = 2 d \sin\theta \$$$

Where:

- $d = 0.283 \text{ nm}$ is the NaCl (100) effective plane spacing used in this experiment.
- θ is the **crystal angle**; the apparatus is run in **COUPLED** mode so the displayed angle is θ .

Uncertainties:

- Angular step / alignment: we model an effective angular uncertainty $\Delta\theta$ (in degrees),
- Voltage readout uncertainty ΔV (in kV),
- Additional uncertainty from the **linear fit** to the rising edge is included through the covariance of the fit parameters.

NAME
E

```
In [8]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy import constants
```

```
In [10]: # Display options
pd.set_option("display.precision", 4)
pd.set_option("display.max_columns", 20)
```

```
In [12]: # Physical constants
d = 0.283e-9      # NaCl spacing used in Lab (m)
e = constants.e    # Elementary charge (C)
c = constants.c    # Speed of Light (m/s)
h_theoretical = constants.h # Accepted Planck constant (J·s)
```

```
In [14]: print(f"d = {d:.3e} m")
print(f"h_theoretical = {h_theoretical:.4e} J·s")

d = 2.830e-10 m
h_theoretical = 6.6261e-34 J·s
```

0.1 Experimental Uncertainties

Angular uncertainty (resolution + alignment):

$\Delta\theta = 0.10^\circ$

High-voltage uncertainty:

$\Delta V = 0.10 \text{ kV}$

These uncertainties propagate into λ_{\min} and ultimately h .

```
In [41]: df = pd.read_csv(r"C:\Users\joshb\OneDrive\PHYSICS BSc UNIVERSITY of EXETER\BSc Physics Stage 2\Practical Physics II\Lab 1\duane hunt data\duane hunt data.csv")
display(df.head())
print("Data shape:", df.shape)
```

	angle_deg	R2	R3	R4	R5	R6	R7	R8	R9	R10
0	2.5	0.2	0.4	0.6	1.4	2.0	3.4	5.8	4.8	9.1
1	2.6	0.2	0.6	0.8	1.4	1.3	2.4	6.1	4.6	6.3
2	2.7	0.2	0.5	1.1	1.6	1.9	2.0	4.8	5.8	8.1
3	2.8	0.3	0.3	1.0	2.0	1.5	2.9	3.2	5.9	7.9
4	2.9	0.2	0.3	0.5	0.9	1.5	2.9	4.4	6.6	7.3

NAME

17

Data shape: (101, 10)

E

```
In [43]: voltage_mapping = {  
    "R2": 15.0,  
    "R3": 17.5,  
    "R4": 20.0,  
    "R5": 22.5,  
    "R6": 25.0,  
    "R7": 27.5,  
    "R8": 30.0,  
    "R9": 32.5,  
    "R10": 35.0  
}  
voltage_mapping
```

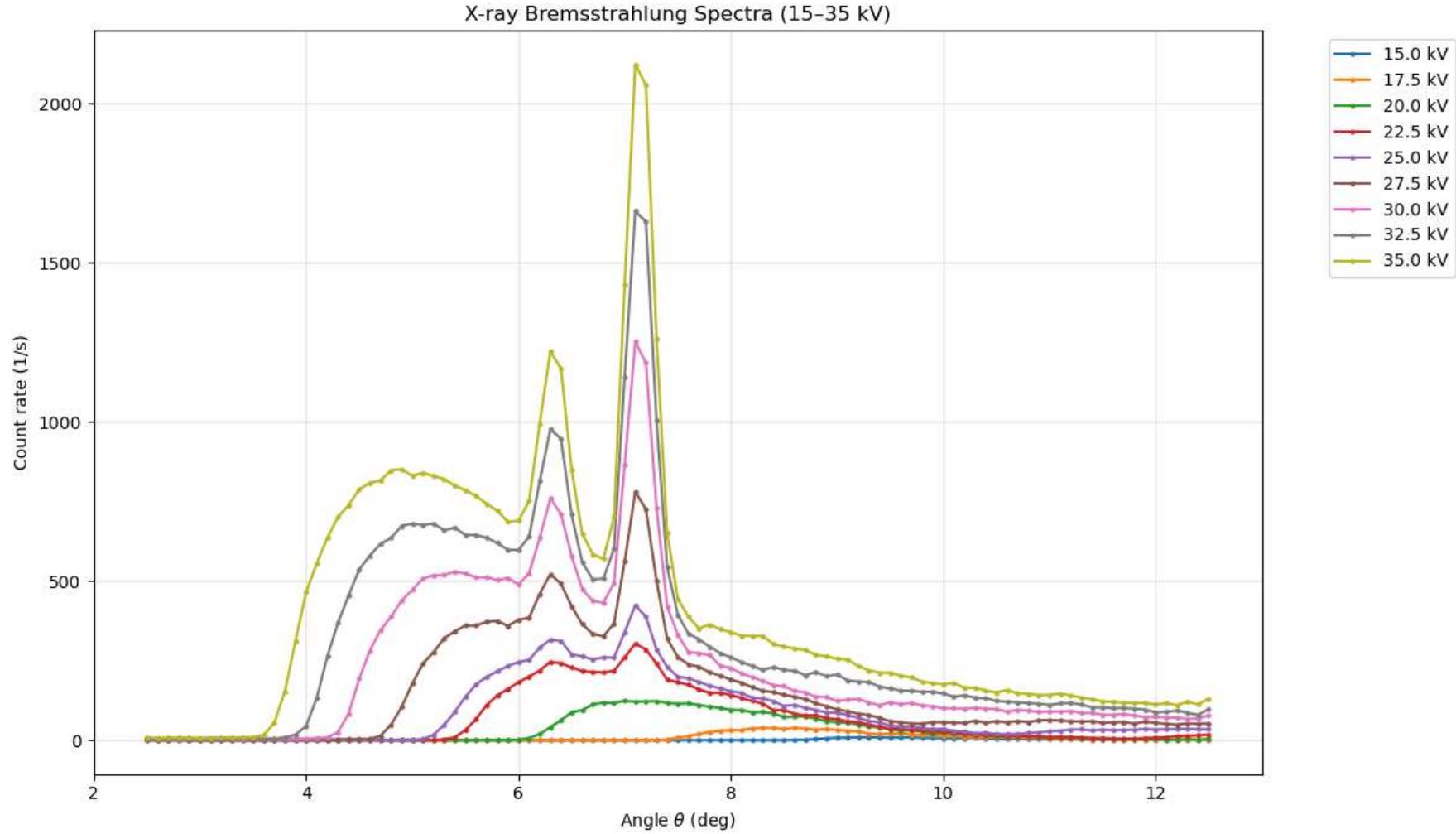
```
Out[43]: {'R2': 15.0,  
          'R3': 17.5,  
          'R4': 20.0,  
          'R5': 22.5,  
          'R6': 25.0,  
          'R7': 27.5,  
          'R8': 30.0,  
          'R9': 32.5,  
          'R10': 35.0}
```

1. Initial Visualisation

I plot all $I(\theta)$ spectra to inspect:

- the bremsstrahlung continuum,
- characteristic peaks,
- the low-angle rising edge that contains λ_{\min} .

```
In [46]: plt.figure(figsize=(12,7))  
for col, V in voltage_mapping.items():  
    plt.plot(df["angle_deg"], df[col], 'o-', markersize=2, label=f"{V} kV")  
  
plt.xlabel(r"Angle $\theta$ (deg)")  
plt.ylabel("Count rate (1/s)")  
plt.title("X-ray Bremsstrahlung Spectra (15-35 kV)")  
plt.grid(alpha=0.3)  
plt.legend(bbox_to_anchor=(1.05,1), loc="upper left")  
plt.tight_layout()  
plt.show()
```



2. Extracting the Cut-off Angle θ_{\min}

The Duane–Hunt cut-off corresponds to the *sharp rise* at the smallest angle.

Procedure:

1. Smooth counts.
2. Compute derivative $\frac{dI}{d\theta}$.

NAME

E

P

3. In the low-angle region (first 40%), locate the steepest rising edge (max derivative).
4. Fit a straight line: $I(\theta) = m\theta + c$
5. Solve for $I=0$: $\theta_{\min} = -\frac{c}{m}$

Uncertainty propagation:

If

$$\theta_{\min} = -\frac{c}{m}$$

then

$$\Delta(\theta_{\min})^2 = \left(\frac{\partial \theta_{\min}}{\partial m}\Delta m\right)^2 + \left(\frac{\partial \theta_{\min}}{\partial c}\Delta c\right)^2$$

with

$$\frac{\partial \theta_{\min}}{\partial m} = \frac{1}{m^2}, \quad \frac{\partial \theta_{\min}}{\partial c} = -\frac{1}{m^2}$$

I also add the base angular uncertainty $\Delta\theta$ in quadrature.

```
In [53]: def linear_model(theta, m, c):
    return m*theta + c
```

```
In [55]: def smooth(y, n=5):
    return np.convolve(y, np.ones(n)/n, mode="same")
```

```
In [57]: def find_cutoff_theta(theta_deg, counts, smooth_window=5, edge_window=4):
    theta = np.asarray(theta_deg)
    counts = np.asarray(counts)

    smooth_counts = smooth(counts, n=smooth_window)

    # Restrict to 40% of scan
    limit = int(0.4 * len(theta))
    theta_low = theta[:limit]
    counts_low = smooth_counts[:limit]

    dI = np.gradient(counts_low, theta_low)
    edge_idx = np.argmax(dI)

    i0 = max(0, edge_idx - edge_window)
    i1 = min(limit, edge_idx + edge_window + 1)

    theta_reg = theta_low[i0:i1]
    counts_reg = counts_low[i0:i1]

    sigma = np.sqrt(np.clip(counts_reg, 1, None))

    m_guess = (counts_reg[-1] - counts_reg[0]) / (theta_reg[-1] - theta_reg[0] + 1e-9)
    c_guess = counts_reg[0] - m_guess * theta_reg[0]
```

```

popt, pcov = curve_fit(
    linear_model,
    theta_reg,
    counts_reg,
    p0=[m_guess, c_guess],
    sigma=sigma,
    absolute_sigma=True
)
m, c = popt
m_err, c_err = np.sqrt(np.diag(pcov))

theta_min = -c/m

dtheta_dm = c / (m*m)
dtheta_dc = -1/m

theta_fit_err = np.sqrt((dtheta_dm*m_err)**2 + (dtheta_dc*c_err)**2)

return theta_min, theta_fit_err, (theta_reg, counts_reg, popt)

```

In [59]: `delta_theta_deg = 0.10`

```

In [90]: results = []
for col, V_kV in voltage_mapping.items():

    theta_deg = df["angle_deg"].values
    counts = df[col].values

    # Skip spectra that are too weak
    MIN_PEAK_COUNTS = 4.0

    if counts.max() < MIN_PEAK_COUNTS:
        # Store NaNs so the regression ignores this voltage
        results.append({
            "Voltage_kv": V_kV,
            "theta_min_deg": np.nan,
            "theta_min_err_deg": np.nan,
            "lambda_min_m": np.nan,
            "lambda_min_err_m": np.nan
        })
        print(f"Skipped {V_kV:.1f} KV (signal too weak).")
        continue

    # Normal cutoff extraction
    theta_min, theta_fit_err, fit_info = find_cutoff_theta(theta_deg, counts)
    theta_min_err_total = np.sqrt(theta_fit_err**2 + delta_theta_deg**2)

    theta_rad = np.deg2rad(theta_min)
    theta_err_rad = np.deg2rad(theta_min_err_total)

```

```

lambda_min = 2*d*np.sin(theta_rad)
lambda_min_err = 2*d*np.cos(theta_rad)*theta_err_rad

results.append({
    "Voltage_kV": V_kV,
    "theta_min_deg": theta_min,
    "theta_min_err_deg": theta_min_err_total,
    "lambda_min_m": lambda_min,
    "lambda_min_err_m": lambda_min_err
})

```

In [92]: `results_df = pd.DataFrame(results).sort_values("Voltage_kV")
display(results_df)`

	Voltage_kV	theta_min_deg	theta_min_err_deg	lambda_min_m	lambda_min_err_m
0	15.0	2.3278	36.0170	2.2989e-11	3.5550e-10
1	17.5	6.0556	154.6177	5.9709e-11	1.5189e-09
2	20.0	5.9630	1.0259	5.8800e-11	1.0080e-11
3	22.5	5.2784	0.3380	5.2069e-11	3.3248e-12
4	25.0	5.0009	0.3007	4.9339e-11	2.9590e-12
5	27.5	4.5826	0.2369	4.5221e-11	2.3328e-12
6	30.0	4.1667	0.1968	4.1124e-11	1.9390e-12
7	32.5	3.8551	0.1742	3.8054e-11	1.7173e-12
8	35.0	3.5575	0.1552	3.5121e-11	1.5307e-12

3. From θ_{\min} to λ_{\min}

Bragg's law:

$$\lambda_{\min} = 2 d \sin(\theta_{\min})$$

Uncertainty:

$$\Delta \lambda_{\min} = 2d \cos(\theta_{\min}) \Delta \theta_{\min}$$

In [95]: `results_df["V_V"] = results_df["Voltage_kV"]*1e3
results_df["inv_lambda"] = 1/results_df["lambda_min_m"]
results_df["inv_lambda_err"] = results_df["lambda_min_err_m"]/(results_df["lambda_min_m"]**2)

display(results_df)`

NAME
E

??

	Voltage_kV	theta_min_deg	theta_min_err_deg	lambda_min_m	lambda_min_err_m	V_V	inv_lambda	inv_lambda_err
0	15.0	2.3278	36.0170	2.2989e-11	3.5550e-10	15000.0	4.3499e+10	6.7268e+11
1	17.5	6.0556	154.6177	5.9709e-11	1.5189e-09	17500.0	1.6748e+10	4.2603e+11
2	20.0	5.9630	1.0259	5.8800e-11	1.0080e-11	20000.0	1.7007e+10	2.9155e+09
3	22.5	5.2784	0.3380	5.2069e-11	3.3248e-12	22500.0	1.9205e+10	1.2264e+09
4	25.0	5.0009	0.3007	4.9339e-11	2.9590e-12	25000.0	2.0268e+10	1.2155e+09
5	27.5	4.5826	0.2369	4.5221e-11	2.3328e-12	27500.0	2.2113e+10	1.1408e+09
6	30.0	4.1667	0.1968	4.1124e-11	1.9390e-12	30000.0	2.4316e+10	1.1465e+09
7	32.5	3.8551	0.1742	3.8054e-11	1.7173e-12	32500.0	2.6279e+10	1.1859e+09
8	35.0	3.5575	0.1552	3.5121e-11	1.5307e-12	35000.0	2.8473e+10	1.2409e+09

4. Weighted Linear Regression

Duane–Hunt predicts:

$$\$ y = \frac{1}{\lambda_{\min}} = m V, \text{ where } m = \frac{e}{hc}.$$

Hence:

$$\$ h = \frac{e}{mc}.$$

We perform a weighted regression using:

$$\$ w_i = \frac{1}{\sigma_{y_i}^2}.$$

```
In [98]: def model(V, m):
    return m*V

In [100... V = results_df["V_V"].values
y = results_df["inv_lambda"].values
sigma_y = results_df["inv_lambda_err"].values

In [102... popt, pcov = curve_fit(model, V, y, sigma=sigma_y, absolute_sigma=True)
m = popt[0]
m_err = np.sqrt(pcov[0,0])

In [104... h_exp = e/(m*c)
h_exp_err = h_exp*(m_err/m)

In [114... print("m =", m, "+/-", m_err)
print("h_exp =", h_exp, "+/-", h_exp_err)
```

N
E

23

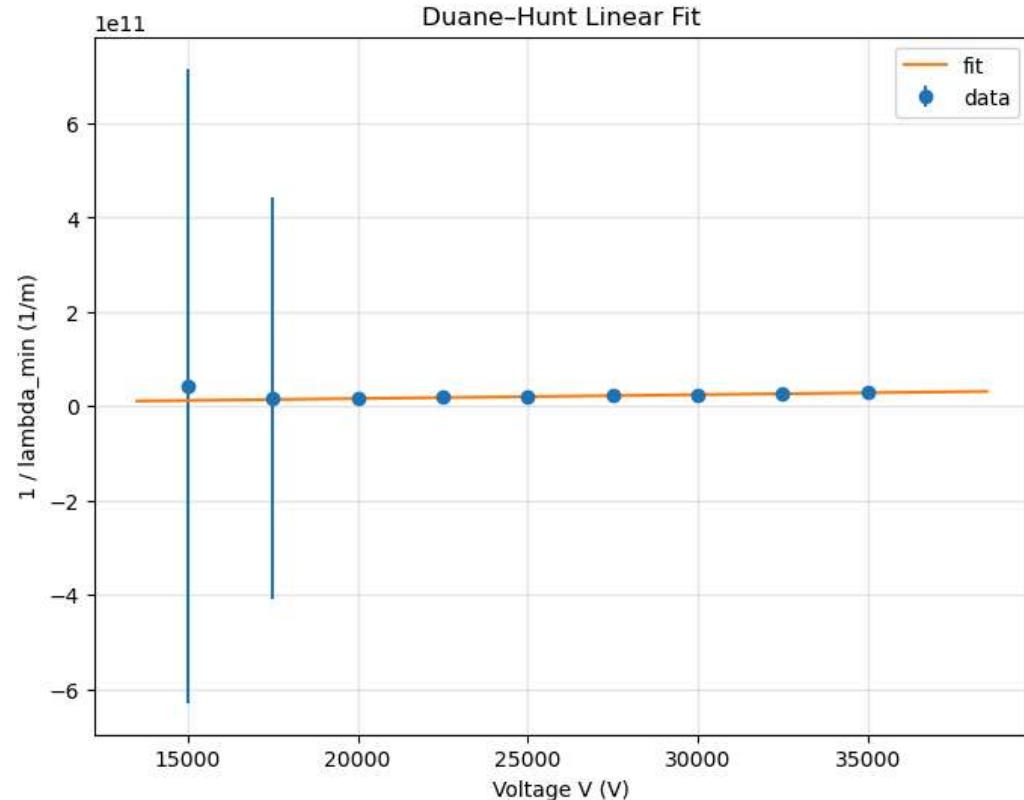
```
print("Accepted =", h_theoretical)
print("Percent diff =", 100*(h_exp - h_theoretical)/h_theoretical)

m = 814299.5633380024 +/- 16622.723561330153
h_exp = 6.563046614897891e-34 +/- 1.3397490863480596e-35
Accepted = 6.62607015e-34
Percent diff = -0.9511450026243515
```

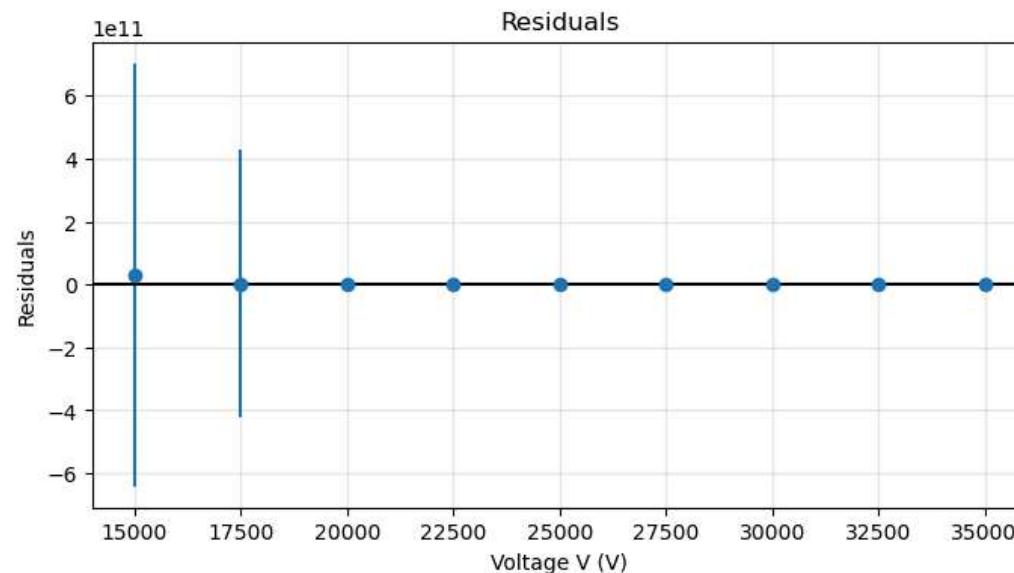
```
In [108]: Vfit = np.linspace(min(V)*0.9, max(V)*1.1, 200)
yfit = model(Vfit, m)
```

```
In [110]: plt.figure(figsize=(8,6))
plt.errorbar(V, y, yerr=sigma_y, fmt='o', label="data")
plt.plot(Vfit, yfit, label="fit")
plt.xlabel("Voltage V (V)")
plt.ylabel("1 / lambda_min (1/m)")
plt.title("Duane-Hunt Linear Fit")
plt.grid(alpha=0.3)
plt.legend()
plt.show()
```

24



```
In [112]:  
plt.figure(figsize=(8,4))  
residuals = y - model(V, m)  
plt.errorbar(V, residuals, yerr=sigma_y, fmt='o')  
plt.axhline(0, color='k')  
plt.xlabel("Voltage V (V)")  
plt.ylabel("Residuals")  
plt.grid(alpha=0.3)  
plt.title("Residuals")  
plt.show()
```



5. Final Result

From the weighted linear regression:

$$\$ h_{\text{exp}} = (6.56 \pm 0.13) \times 10^{-34} \text{ J-s} \$$$

Accepted value:

$$\$ h_{\text{theoretical}} = 6.6261 \times 10^{-34} \text{ J-s} \$$$

Percentage difference:

$$\$ \% \text{ diff} = -0.95 \% \$$$

6. Discussion

- The rising-edge linear method provides an objective determination of θ_{\min} .
- Systematic uncertainties come from:
 - alignment of the goniometer,
 - regions chosen for edge fitting,
 - voltage calibration,
 - crystal quality.
- High-voltage points contribute most strongly to determining h .

N
A
M
E

15

16

26



NAME	ID	DATE 19/10/2025
EXP LABEL		PAGE 27

Atomic Spectroscopy:

Background Reading Summary:

Quantum mechanics predicts that electrons bound to discrete energy levels, which can be (in theory) determined by solving the Schrödinger equation subject to appropriate boundary conditions.

When electrons are excited, they return to equilibrium by making transitions from excited state back down to lowest energy state available, and the energy released in doing so is released as photons. The energy of these photons is discrete and equal to the difference in energy levels of the transition, which is characteristic of the atom involved.

Thus different atoms have different atomic spectra.

In practice, the Schrödinger equation is only exactly solvable for the simplest atom: hydrogen. A simplified diagram showing the energy levels for the electron in a hydrogen atom (figure 1). Vertical lines show some of the transitions that can occur as an excited electron returns to its ground state.

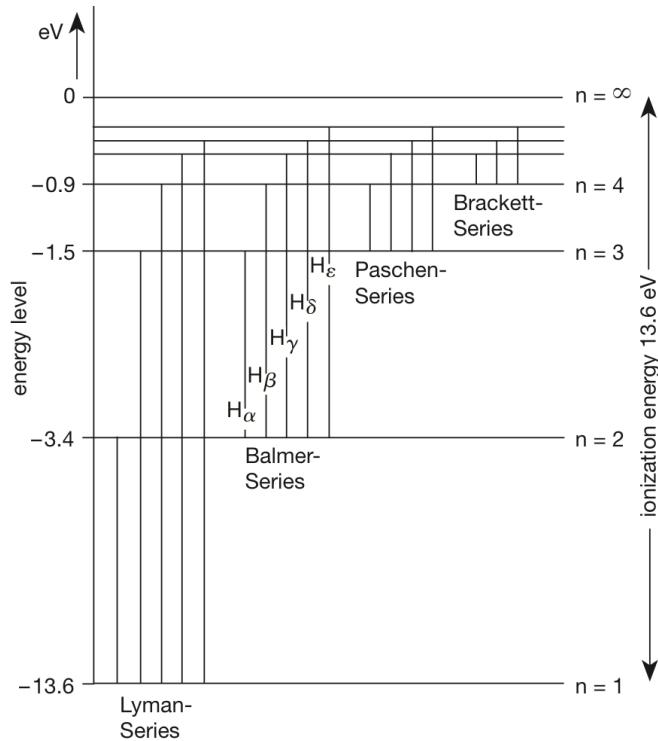


Figure 1: Simplified energy level diagram for the electron bound to a hydrogen nucleus. n is the principal quantum number, and n = ∞ corresponds to ionization.

NAME	ID	DATE 19/10/2025
EXP LABEL		PAGE 28

The Lyman Series includes all transitions in which the final state is the ground state (principle quantum number $n=1$). The photons involved in these transition are ultraviolet.

The Balmer Series includes all transitions in which the final state has $n=2$. The photons involved in the lowest 3/4 transitions of the Balmer Series are visible and these are the transitions I will be investigating.

Two further series are shown, both of which involve infrared photons.

The energy levels for hydrogen atom are given by:

$$E_n = -\frac{Ry}{n^2}, \quad Ry = \frac{m_0 e^4}{(2(4\pi\epsilon_0)^2 \hbar^2)} = 13.6 \text{ eV}$$

↑ Rydberg Constant.

Photon energy is equal to the difference in energy levels of the transition when an electron deexcites.

The Balmer Series always ends on $n=2$, therefore:

$$E_{\text{Balmer}} = E_n - E_2 \Rightarrow E_{\text{Balmer}} = Ry \left(\frac{1}{2^2} - \frac{1}{n^2} \right)$$

where n is integers 3, 4, 5, etc.

Using this equation, I will now calculate the energies and wavelengths of the first few photons in the Balmer Series and verify that they are in the visible region.

$\underline{n=3:} \quad E = \frac{hc}{\lambda} \Leftrightarrow \lambda = \frac{hc}{E_3}$ $E_3 = 13.6 \text{ eV} \left(\frac{1}{2^2} - \frac{1}{3^2} \right), \lambda_3 = \frac{hc}{E_3}$ $\Rightarrow E_3 = 1.89 \text{ eV}, \lambda_3 = 6.58 \times 10^{-7} \text{ m}$	$\underline{n=4:} \quad E_4 = 13.6 \text{ eV} \left(\frac{1}{2^2} - \frac{1}{4^2} \right)$ $\Rightarrow E_4 = 2.55 \text{ eV}, \lambda_4 = 4.875 \times 10^{-7} \text{ m}$	$\underline{n=5:} \quad E_5 = 13.6 \text{ eV} \left(\frac{1}{2^2} - \frac{1}{5^2} \right)$ $\Rightarrow E_5 = 2.86 \text{ eV}, \lambda_5 = 4.35 \times 10^{-7} \text{ m}$
--	--	---

NAME	ID	DATE 19/10/2025
EXP LABEL		PAGE 29

The visible range of light is between $3.8 \times 10^{-7} \text{ m}$ and $7 \times 10^{-7} \text{ m}$. [2]

$\lambda_3, \lambda_4, \lambda_5$ all lie within this range, and are all therefore visible.

As n increases, the energies of successive series get closer together, converging to a limit.

$$n \rightarrow \infty \Rightarrow E_{\text{Balmer}} \rightarrow R_y \left(\frac{1}{2^2} - 0 \right) \therefore \underline{\underline{E_{\text{Balmer}} \rightarrow 3.4 \text{ eV}}}$$

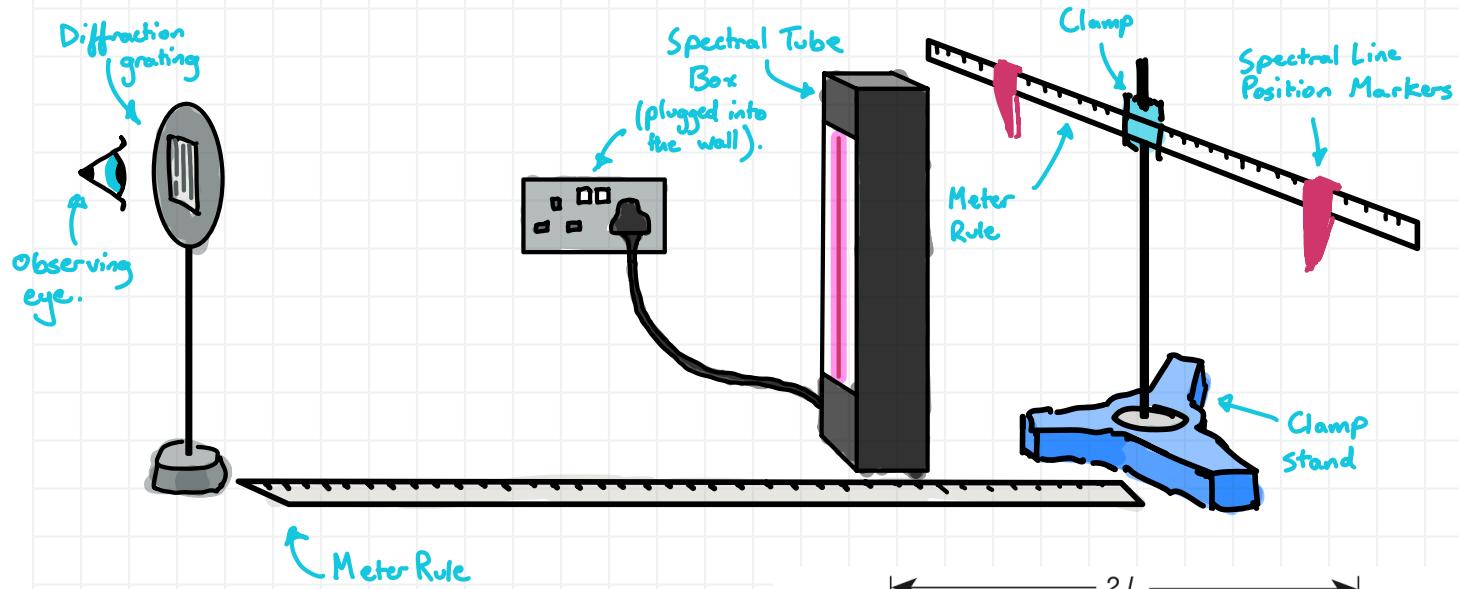
This means that wavelengths of photons as $n \rightarrow \infty$ tend towards:

$$\lambda = \frac{hc}{E} \Rightarrow \lambda \rightarrow 3.65625 \times 10^{-7} \text{ m} \underline{\underline{\approx 366 \text{ nm}}}$$

Method:

In this experiment, will be using a diffraction grating to calculate an experimental value for the Rydberg Constant. This will be done by calculating wavelengths for hydrogen Balmer Series emission spectra.

Equipment:



Top down view of experimental setup.

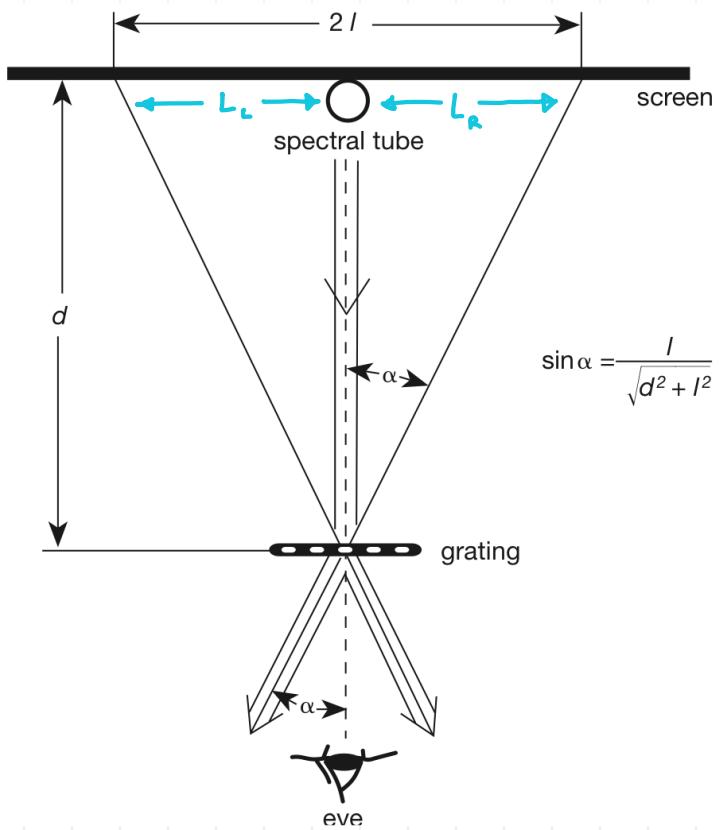
Includes everything to measure:

d - screen distance.

L - screen length.

α - angle of diffraction, which is calculated using measurements of screen distance 'd' and screen length 'L'.

$$\sin \alpha = \frac{1}{\sqrt{d^2 + L^2}}$$



NAME	ID	DATE 20/10/2025
EXP LABEL		PAGE 31

Steps: * Lab partner and I

- 1) First, we're going to use neon lamp (who's spectra wavelengths are known) to calibrate the diffraction grating.
- 2) To do this, we set up equipment as shown in diagrams, with neon gas tube in Spectral Tube Box.
- 3) Then putting eye at angled positions looking at spectra from gas tube through diffraction grating, then slide spectral line position markers to position of yellow & green emission lines.
- 4) Measure and record screen lengths for observed emission lines on both the left and right.
- 5) Using known conditions/parameters of the experiment and diffraction grating equation, calculating a value for diffraction grating constant.
- 6) Repeat steps 2, 3, 4 with hydrogen gas in Spectral Tube box, measuring screen lengths for red, green & blue emission lines.
- 7) Using calculated values for screen length, diffraction grating constant and rearranged diffraction grating equation.
- 8) Plot a graph of wavelengths and use that graph to obtain the Rydberg Constant.

Conditions/Parameters of Experiment:

Distance $d = 1\text{m}$ (every measurement)

Order of diffraction $p = 1$ (every measurement)

Central Position on ruler = 500 mm (every measurement)

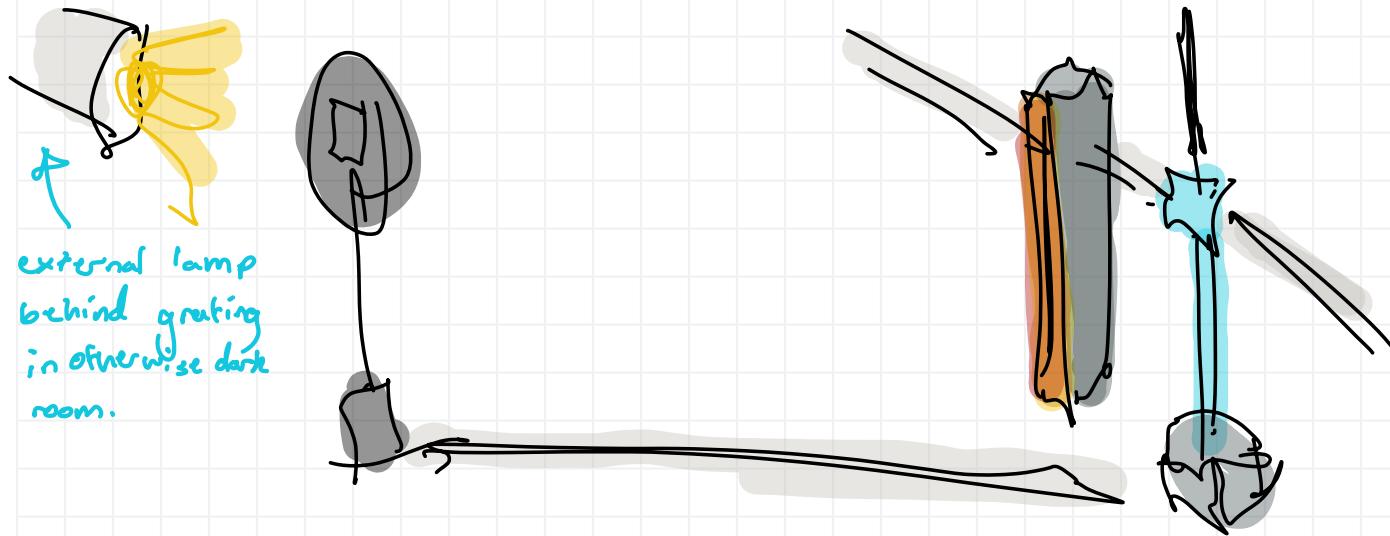
Safety:

Gas discharge tubes require between 1kV and 5kV to operate. Power supply delivers this voltage, but is current-limited to 2.5 mA, which is well below safe limit.

The discharge tubes are fragile and made of glass. Therefore, make sure to get demonstrator to handle them and to hold them on the wider sections (more robust, less likely to snap).

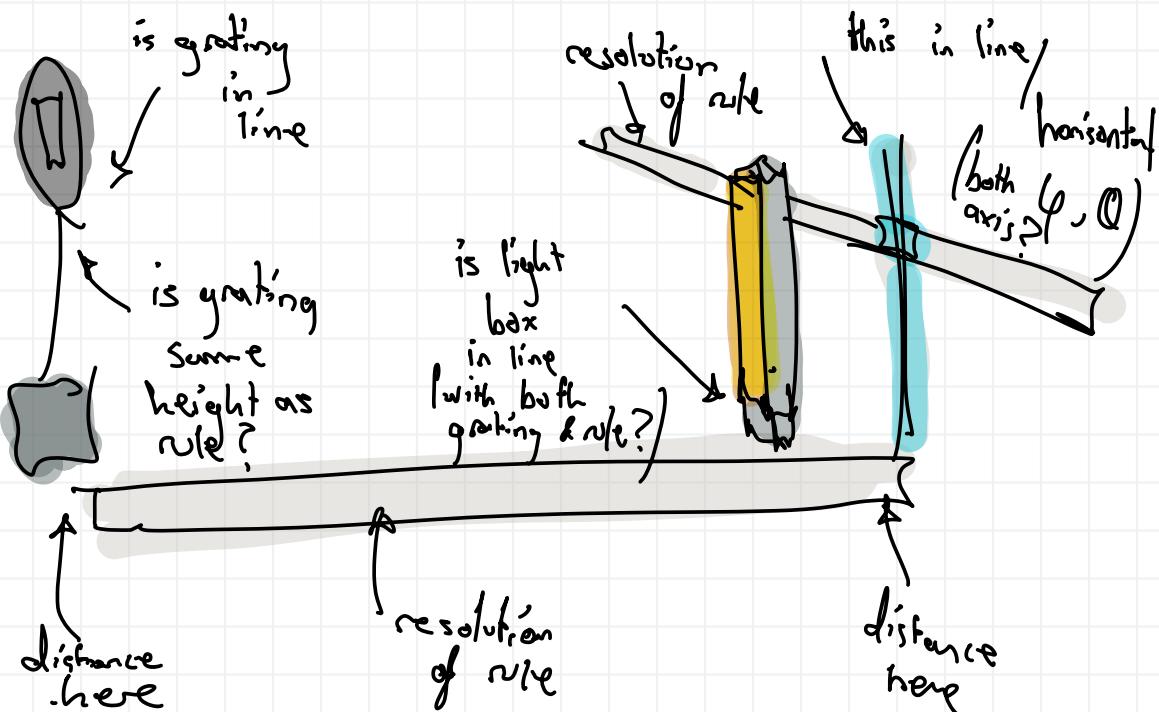
Lighting conditions:

Couldn't see ruler with light off, chose to put lamp behind:



Putting lamp on in background could affect the light spectrum, so chose position of lamp which didn't change the position of lines with it on vs off.

Sources of uncertainty:



NAME	ID	DATE 20/10/2025
EXP LABEL		PAGE 33

Other sources to consider:

Parallax in reading \rightarrow (fixed head position, read with one eye).

Is grating parallel to scale?

Parallax when reading screen lengths off ruler.

Core Equations:

diffraction grating equation

$$p\lambda = dg \sin \alpha$$

$$\frac{1}{\lambda} = R_H \left(\frac{1}{Z^2} - \frac{1}{n^2} \right)$$

I plan on doing detailed
data analysis in Jupyter Notebook
- please see Python file section later
for code & markdown cells.

Balmer equation.

Data Collection:

Neon:

yellow right - 858 mm

green right - 832 mm

yellow left - 100 mm

green left - 138 mm

two green lines, chose the line
closer to red.

Hydrogen:

blue right - 790 mm

red right - 906 mm

blue left - 165 mm

red left - 39 mm

later realised 'blue'
was actually green.

NAME	ID	DATE
EXP LABEL		PAGE 34

Fri 24th October :

Re-measuring emission line positions for H_z, as initially only measured 2 of 3 visible lines.

H_z emission line position calculator:

initially thought measured blue. Actually measured green.

Left:

Red: 50 mm

Green: 179 mm

Blue: 218 mm

Right:

Red: 948 mm

Green: 825 mm

Blue: 788 mm

This means that I have another set of results for hydrogen emission lines. It would have been good experimental practice for me to collect a second set of results for neon as well. This did not happen, and will be discussed in the discussion section of my report.

NAME	ID	DATE 25/10/2025
EXP LABEL		PAGE 35

Data Analysis:

Sources of error:

Position Measurements (L)

- Ruler resolution (each end) $\pm 1.000 \text{ mm}$
- $\hookrightarrow \delta L_{\text{res}} = \sqrt{1^2 + 1^2} = \pm 1.414 \text{ mm}$
- Parallax Error $\pm 1.000 \text{ mm}$
- Line center identification $\pm 1.000 \text{ mm}$

Total Line measurement uncertainty:

$$\hookrightarrow \delta L_{\text{meas}} = \sqrt{(1.414^2 + 1.000^2 + 1.000^2)} = \pm 1.723 \text{ mm}$$

Distance Measurements (d):

- Ruler resolution $\pm 1.00 \text{ mm}$
- Grating alignment uncertainty $\pm 5.000 \text{ mm}$
- Clump stand alignment uncertainty $\pm 5.000 \text{ mm}$

Total measurement uncertainty

$$\hookrightarrow \delta d_{\text{meas}} = \sqrt{1.000^2 + 5.000^2 + 5.000^2} = \pm 7.141 \text{ mm}$$

Angular Alignment Uncertainty:

The screen ruler may not be perfectly perpendicular to the line of sight.

- Angular uncertainty $\delta \theta = \pm 2.000^\circ = \pm 0.03491 \text{ rad}$

NAME	ID	DATE 25/10/2025
EXP LABEL		PAGE 36

Affect on position (L):

↳ Distance d is measured along zero order, so unaffected by screen angle.

↳ Position L is measured along potentially tilted screen

for each screen length (l_i):

$$\delta L_{\text{align},i} = l_i \times \tan(\delta\theta) \quad \leftarrow \text{for large } L, \text{ larger alignment error.}$$

Then for each measurement:

$$\delta L_{\text{total},i} = \sqrt{(1.723 \times 10^{-3})^2 + (\delta L_{\text{align},i})^2}$$

Error propagation through $\sin \alpha$:

Function: $\sin \alpha = \frac{l}{\sqrt{l^2 + d^2}}$

Partial derivatives:

$$\frac{\partial \sin \alpha}{\partial l} = \frac{d^2}{(l^2 + d^2)^{3/2}} \quad , \quad \frac{\partial \sin \alpha}{\partial d} = -\frac{ld}{(l^2 + d^2)^{3/2}}$$

Uncertainty:

$$\delta[\sin \alpha] = \sqrt{\left(\frac{\partial \sin \alpha}{\partial l} \cdot \delta L_{\text{total}} \right)^2 + \left(\frac{\partial \sin \alpha}{\partial d} \cdot \delta d_{\text{total}} \right)^2}$$

NAME	ID	DATE 25/10/2025
EXP LABEL		PAGE 37

Error propagation through grating spacing:

Function:

$$d_g = \frac{P \lambda_{ref}}{\sin \alpha} \quad (P=1)$$

Partial derivatives:

$$\frac{\partial d_g}{\partial \lambda} = \frac{1}{\sin \alpha}, \quad \frac{\partial d_g}{\partial [\sin \alpha]} = -\frac{\lambda_{ref}}{\sin^2 \alpha}$$

Uncertainty:

$$\delta d_g = \sqrt{\left(\frac{1}{\sin \alpha} \cdot \delta \lambda_{ref}\right)^2 + \left(-\frac{\lambda_{ref}}{\sin^2 \alpha} \cdot \delta [\sin \alpha]\right)^2}$$

where $\delta \lambda_{ref} = \pm 1.000 \times 10^{-11} \text{ m}$

Error Propagation through Hydrogen Wavelengths (λ_H):

Function:

$$\lambda_H = d_g \cdot \sin \alpha$$

Partial Derivatives:

$$\frac{\partial \lambda_H}{\partial d_g} = \sin \alpha, \quad \frac{\partial \lambda_H}{\partial [\sin \alpha]} = d_g$$

Uncertainty:

$$\delta \lambda_H = \sqrt{(\sin \alpha \cdot \delta d_g)^2 + (d_g \cdot \delta [\sin \alpha])^2}$$

NAME	ID	DATE 25/10/2025
EXP LABEL		PAGE 38

Error Propagation to Rydberg Constant:

Function:

$$\frac{1}{\lambda_H} = R_H \cdot \left(\frac{1}{z^2} - \frac{1}{n^2} \right)$$

(or $y = R_H x$, $\begin{cases} y = \frac{1}{x} \\ x = \left(\frac{1}{z^2} - \frac{1}{n^2} \right) \end{cases}$)

Uncertainty in $\frac{1}{\lambda_H}$:

$$\delta y = \frac{\delta \lambda_H}{\lambda_H^2}$$

Uncertainty in R_H (weighted linear regression):

$$\delta R_H = \frac{1}{\sqrt{\sum \left(\frac{x_i}{\delta y_i} \right)^2}}$$

Atomic Spectroscopy - Calculating Rydberg Constant

Date: 22/10/2025

Author: Joshua Bowley

Aim: This notebook analyses Neon calibration data to find the grating spacing and uses it to calculate Hydrogen wavelengths and the Rydberg constant.

In [279...]

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from scipy.optimize import curve_fit
```

0. Experimental Parameters

All fixed values (screen distance, center position, etc.).

Fundamental uncertainties (ruler resolutions, alignment errors, reference wavelength errors).

In [282...]

```
# Fixed values:

center = 0.500 # position of center of screen ruler, in m
screen_distance = 1.000 # m
p = 1 # only first order diffracted lines are considered
```

In [284...]

```
# Fundamental Uncertainties:

delta_l_meas = 1.723e-3 # total screen length uncertainty in meters
delta_d_meas = 7.141e-3 # total screen distance uncertainty in meters
delta_theta = np.radians(2) # angular uncertainty of screen in radians
delta_ref_wave = 0.01e-9 # uncertainty in reference wavelengths of neon green and yellow spectral lines
```

1. Calibration using neon

Using known reference wavelengths from NIST Atomic Spectra Database (Ne I):

585.25 nm (yellow) and 540.06 nm (green)

In [287...]

```
# Create a pandas DataFrame of measured data
neon_df = pd.DataFrame({
    "colour": ["yellow_L", "yellow_R", "green_L", "green_R"],
    "position_m": [100e-3, 858e-3, 138e-3, 832e-3],
```

N
E

40

```
        "wavelength_m": [585.25e-9, 585.25e-9, 540.06e-9, 540.06e-9]
    })
```

```
In [288...]: # Calculate screen length and its uncertainty
neon_df["l_m"] = np.abs(neon_df["position_m"] - center)
neon_df["delta_l_m"] = neon_df["l_m"] * np.tan(delta_theta)
```

```
In [289...]: # Calculate sin(alpha) and its uncertainty
# Intermediate calculations for clarity
l_sq_plus_D_sq = neon_df["l_m"]**2 + screen_distance**2
denominator_32 = l_sq_plus_D_sq ** (1.5)
```

```
In [290...]: # Partial derivatives for sin(alpha)
d_sin_dl = (screen_distance**2) / denominator_32
d_sin_dD = (-neon_df["l_m"] * screen_distance) / denominator_32
```

```
In [291...]: # Propagate uncertainties for sin(alpha)
variance_from_l = (d_sin_dl * neon_df["delta_l_m"])**2
variance_from_D = (d_sin_dD * delta_d_meas)**2
neon_df["sin(alpha)"] = neon_df["l_m"] / np.sqrt(l_sq_plus_D_sq)
neon_df["delta_sin(alpha)"] = np.sqrt(variance_from_l + variance_from_D)
```

```
In [367...]: # Calculate grating spacing d_g
neon_df["d_g_m"] = p * neon_df["wavelength_m"] / neon_df["sin(alpha)"]
```

```
In [293...]: # Intermediate calculations for d_g uncertainty
d_dg_dlambd = 1 / neon_df["sin(alpha)"]
d_dg_dsinalpha = -neon_df["wavelength_m"] / (neon_df["sin(alpha)"])**2
```

```
In [294...]: # Propagate uncertainties for d_g
variance_from_wavelength = (d_dg_dlambd * delta_ref_wave)**2
variance_from_sinalpha = (d_dg_dsinalpha * neon_df["delta_sin(alpha)"])**2
neon_df["delta_d_g_m"] = np.sqrt(variance_from_wavelength + variance_from_sinalpha)
```

```
In [295...]: # Display with column-specific formats
display(
    neon_df.style.format({
        "position_m": "{:.4f}",
        "wavelength_m": "{:.6e}",
        "l_m": "{:.4f}",
        "delta_l_m": "{:.6f}",
        "sin(alpha)": "{:.8f}",
        "delta_sin(alpha)": "{:.8e}",
        "d_g_m": "{:.8e}",
        "delta_d_g_m": "{:.8e}"
    })
)
```

NAME
E

41

	colour	position_m	wavelength_m	I_m	delta_I_m	sin(alpha)	delta_sin(alpha)	d_g_m	delta_d_g_m
0	yellow_L	0.1000	5.852500e-07	0.4000	0.013968	0.37139068	1.14117562e-02	1.57583385e-06	4.84208022e-08
1	yellow_R	0.8580	5.852500e-07	0.3580	0.012502	0.33705200	1.06488853e-02	1.73637896e-06	5.48594965e-08
2	green_L	0.1380	5.400600e-07	0.3620	0.012641	0.34038375	1.07267657e-02	1.58662098e-06	5.00003728e-08
3	green_R	0.8320	5.400600e-07	0.3320	0.011594	0.31508866	1.01158328e-02	1.71399375e-06	5.50272959e-08

Now, I am going to use the experimental values for d_g and propogated uncertainties δd_g to calculate a weighted mean value for d_g . This means that calculated values with smaller uncertainty will be favoured, and therefore the calibrated diffraction grating constant should be more accurate.

```
In [297... # Calculate weighted mean of d_g (weighted by 1/uncertainty^2)
weights = 1 / neon_df["delta_d_g_m"]**2
d_g_weighted_mean = np.sum(weights * neon_df["d_g_m"]) / np.sum(weights)
```

```
In [298... # Uncertainty in weighted mean (standard error of weighted mean)
delta_d_g_weighted_mean = 1 / np.sqrt(np.sum(weights))
print(f"Weighted mean grating spacing: {d_g_weighted_mean:.4g} ± {delta_d_g_weighted_mean:.4g} m")
```

Weighted mean grating spacing: 1.645e-06 ± 2.591e-08 m

I will also calculate a simple (unweighted mean) for comparison. In the case of a simple mean, the uncertainties can be given by:

$$\delta \bar{x} = \frac{s}{\sqrt{N}}$$

where s = sample standard deviation, and N = number of measurements.

```
In [300... # Calculate simple (unweighted) mean for comparison
d_g_mean_m = np.mean(neon_df["d_g_m"])
delta_d_g_mean_m = np.std(neon_df["d_g_m"], ddof=1) / np.sqrt(len(neon_df))
```

```
In [301... print(f"Weighted mean grating spacing: {d_g_weighted_mean:.4g} ± {delta_d_g_weighted_mean:.4g} m")
print(f"Mean grating spacing: {d_g_mean_m:.3e} ± {delta_d_g_mean_m:.4g} m")
```

Weighted mean grating spacing: 1.645e-06 ± 2.591e-08 m

Mean grating spacing: 1.653e-06 ± 4.187e-08 m

Both weighted and unweighted mean grating spacing lie within the other's uncertainty values, further validating these experimental values.

2. Hydrogen Spectrum

```
In [304... # Create a pandas DataFrame of measured data
hydrogen_df = pd.DataFrame({
    "colour": ["red_La", "red_Lb", "red_Ra", "red_Rb", "green_La", "green_Lb", "green_Ra", "green_Rb", "blue_L", "blue_R"],
```

```

        "position_m": [39e-3, 50e-3, 906e-3, 948e-3, 165e-3, 179e-3, 790e-3, 825e-3, 218e-3, 788e-3],
    })

In [305... # Calculate screen length and its uncertainty
hydrogen_df["l_m"] = np.abs(hydrogen_df["position_m"] - center)
hydrogen_df["delta_l_m"] = hydrogen_df["l_m"] * np.tan(delta_theta)

In [306... # Calculate sin(alpha) and its uncertainty
# Intermediate calculations for clarity
H_l_sq_plus_D_sq = hydrogen_df["l_m"]**2 + screen_distance**2
H_denominator_32 = H_l_sq_plus_D_sq ** (1.5)

In [307... # Partial derivatives for sin(alpha)
H_d_sin_dl = (screen_distance**2) / H_denominator_32
H_d_sin_dD = (-hydrogen_df["l_m"] * screen_distance) / H_denominator_32

In [308... # Propagate uncertainties for sin(alpha)
H_variance_from_l = (H_d_sin_dl * hydrogen_df["delta_l_m"])**2
H_variance_from_D = (H_d_sin_dD * delta_d_meas)**2
hydrogen_df["sin(alpha)"] = hydrogen_df["l_m"] / np.sqrt(H_l_sq_plus_D_sq)
hydrogen_df["delta_sin(alpha)"] = np.sqrt(H_variance_from_l + H_variance_from_D)

```

I am choosing to use the weighted mean grating spacing, as I value the statistical accuracy gained by weighing the mean by the uncertainties. The difference between the two choices is similar, with the unweighted mean having a larger uncertainty.

```

In [310... # Calculate wavelength and its uncertainty
hydrogen_df["wavelength_m"] = d_g_weighted_mean * hydrogen_df["sin(alpha)"]

In [311... # Propagate uncertainties for wavelength_a
H_variance_from_d_g = (hydrogen_df["sin(alpha)"] * delta_d_g_weighted_mean)**2
H_variance_from_sin_a = (d_g_weighted_mean * hydrogen_df["delta_sin(alpha)"])**2
hydrogen_df["delta_wavelength_m"] = np.sqrt(H_variance_from_d_g + H_variance_from_sin_a)

In [312... # Display with column-specific formats
display(
    hydrogen_df.style.format({
        "position_m": "{:.4f}",
        "l_m": "{:.4f}",
        "delta_l_m": "{:.6f}",
        "sin(alpha)": "{:.4g}",
        "delta_sin(alpha)": "{:.4g}",
        "wavelength_m": "{:.3e}",
        "delta_wavelength_m": "{:.3e}"
    })
)
```

	colour	position_m	l_m	delta_l_m	sin(alpha)	delta_sin(alpha)	wavelength_m	delta_wavelength_m
0	red_La	0.0390	0.4610	0.016098	0.4187	0.01231	6.888e-07	2.297e-08
1	red_Lb	0.0500	0.4500	0.015714	0.4104	0.01216	6.751e-07	2.266e-08
2	red_Ra	0.9060	0.4060	0.014178	0.3762	0.01151	6.189e-07	2.130e-08
3	red_Rb	0.9480	0.4480	0.015645	0.4088	0.01214	6.726e-07	2.260e-08
4	green_La	0.1650	0.3350	0.011698	0.3176	0.01018	5.226e-07	1.866e-08
5	green_Lb	0.1790	0.3210	0.011210	0.3056	0.009876	5.028e-07	1.808e-08
6	green_Ra	0.7900	0.2900	0.010127	0.2785	0.009157	4.582e-07	1.671e-08
7	green_Rb	0.8250	0.3250	0.011349	0.3091	0.009964	5.085e-07	1.825e-08
8	blue_L	0.2180	0.2820	0.009848	0.2714	0.008961	4.465e-07	1.634e-08
9	blue_R	0.7880	0.2880	0.010057	0.2768	0.009109	4.553e-07	1.661e-08

3. Determination of Rydberg Constant

I identified the hydrogen spectral lines by comparing my measured wavelengths to the known Balmer series transitions:

- Red lines (~655-660 nm) → n = 3 → 2 transition ($H - \alpha$)
- Green lines (~490 nm) → n = 4 → 2 transition ($H - \beta$)
- Blue lines (~435 nm) → n = 5 → 2 transition ($H - \gamma$)

This assignment follows from the Rydberg formula - larger n values give longer wavelengths (redder light), and my measured wavelengths match the expected order for the Balmer series (transitions to n=2).

Now I will calculate the Rydberg constant using the Balmer series formula and weighted linear regression.

In [315...]

```
# Add n values based on colour mapping
n_map = {
    "red_La": 3, "red_Lb": 3, "red_Ra": 3, "red_Rb": 3,
    "green_La": 4, "green_Lb": 4, "green_Ra": 4, "green_Rb": 4,
    "blue_L": 5, "blue_R": 5
}
```

In [316...]

```
hydrogen_df["n"] = hydrogen_df["colour"].map(n_map)
```

In [317...]

```
# Calculate weighted statistics for each n group
n_vals = hydrogen_df["n"].unique()
results = []
```

In [318...]

```

for n in n_vals:
    group = hydrogen_df[hydrogen_df["n"] == n]
    weights = 1 / group["delta_wavelength_m"]**2
    weighted_mean = np.sum(weights * group["wavelength_m"]) / np.sum(weights)
    weighted_uncertainty = 1 / np.sqrt(np.sum(weights))
    count = len(group)

    results.append({
        "n": n,
        "wavelength_mean": weighted_mean,
        "delta_wavelength_mean": weighted_uncertainty,
        "measurement_count": count
    })

balmer_df = pd.DataFrame(results)
display(balmer_df)

```

	n	wavelength_mean	delta_wavelength_mean	measurement_count
0	3	6.622607e-07	1.117774e-08	4
1	4	4.959622e-07	8.937450e-09	4
2	5	4.508466e-07	1.164794e-08	2

Now I will calculate the Rydberg constant using the Balmer formula and weighted linear regression.

In [371...]

```
# Calculate y = 1/Lambda and propagate uncertainty
balmer_df["y"] = 1 / balmer_df["wavelength_mean"]
balmer_df["delta_y"] = balmer_df["delta_wavelength_mean"] / balmer_df["wavelength_mean"]**2
```

In [390...]

```
# Calculate x = (1/2^2 - 1/n^2)
balmer_df["x"] = (1/4) - (1 / balmer_df["n"]**2)
```

In [392...]

```
# Display with column specific formatting:
display(balmer_df.style.format({
    'wavelength_mean': '{:.3e}',
    'delta_wavelength_mean': '{:.3e}',
    'y': '{:.0f}',
    'y_err': '{:.0f}',
    'x': '{:.4f}'
}))
```

n	wavelength_mean	delta_wavelength_mean	measurement_count	y	delta_y	x
0	3	6.623e-07	1.118e-08	4	1509979	25485.660568 0.1389
1	4	4.960e-07	8.937e-09	4	2016283	36334.277585 0.1875
2	5	4.508e-07	1.165e-08	2	2218049	57304.893999 0.2100

Now I will perform weighted linear regression to determine the Rydberg Constant.

```
In [395...]: # Linear regression model for fitting
def linear_model(x, R_H):
    return R_H * x

In [416...]: # Perform weighted Linear regression
popt, pcov = curve_fit(linear_model, balmer_df["x"], balmer_df["y"],
                       sigma = balmer_df["delta_y"], absolute_sigma=True, p0=1.1e7) # initial guess R_H accepted value

In [399...]: R_H_best = popt[0]
R_H_err = np.sqrt(pcov[0][0])

In [401...]: print(f"Experimental Rydberg Constant: R_H = {R_H_best:.0f} ± {R_H_err:.0f} m⁻¹")
print(f"Accepted R_H = 10973731.6 m⁻¹")
print(f"Percentage difference: {((R_H_best - 10973731.6)/10973731.6*100):.2f}%")

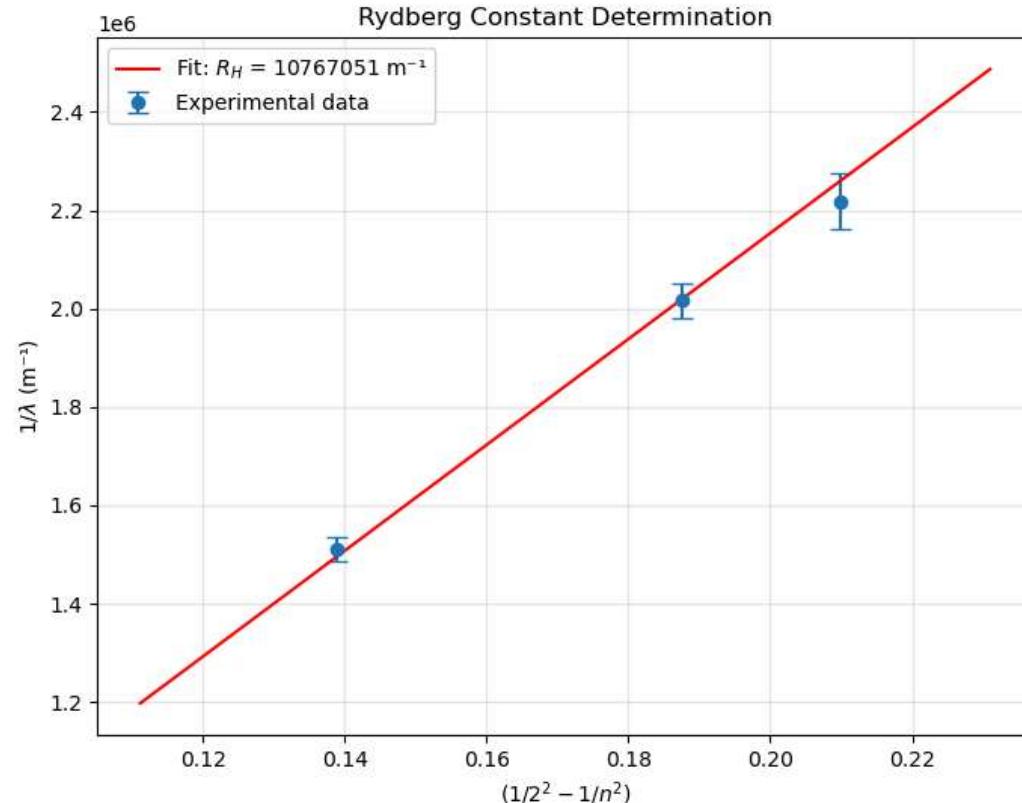
Experimental Rydberg Constant: R_H = 10767051 ± 119730 m⁻¹
Accepted R_H = 10973731.6 m⁻¹
Percentage difference: -1.88%
```

Finally, I will plot to visualise the linear relationship and fit quality.

```
In [404...]: # Generate fit line and create plot
x_fit = np.linspace(balmer_df['x'].min() * 0.8, balmer_df['x'].max() * 1.1, 100)
y_fit = linear_model(x_fit, R_H_best)

In [410...]: plt.figure(figsize=(8, 6))
plt.errorbar(balmer_df['x'], balmer_df['y'], yerr=balmer_df['delta_y'],
             fmt='o', capsize=5, label='Experimental data')
plt.plot(x_fit, y_fit, 'r-', label=f'Fit: $R_H$ = {R_H_best:.0f} m⁻¹')
plt.xlabel('$1/2^2 - 1/n^2$')
plt.ylabel('$1/\lambda$ (m⁻¹)')
plt.title('Rydberg Constant Determination')
plt.legend()
plt.grid(True, alpha=0.3)
plt.show()
```

46



In [412]:

```
# Save graph
plt.savefig('rydberg_constant_plot.png', dpi=300, bbox_inches='tight')
```

```
<Figure size 640x480 with 0 Axes>
```