Lab 1 - The Zeeman effect

October 22, 2020

Name: Benjamin Stott, Student ID: 18336161,Lab partner=Matthew Geever

1 declaration

Declaration concerning plagiarism

I have read and I understand the plagiarism provisions in the General Regulations of the *University Calendar* for the current year, found at http://www.tcd.ie/calendar

I have completed the Online Tutorial in avoiding plagiarism 'Ready, Steady, Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write

STUDENT NUMBER: 18336161

SIGNED: Benjamin Stott

2 Abstract

This lab was performed to observe the zeeman effect(i.e the splitting of spectral lines due to a magnetic field) for red Cadmium light. This study was split into 3 experiments which each evaluated a different aspect of the zeeman effect.

The first part of this experiment measured the wavelength of red cadmium light to be $(652 \pm 8)nm$ which agreed significantly with the literature value obtained which was $(642nm)$.

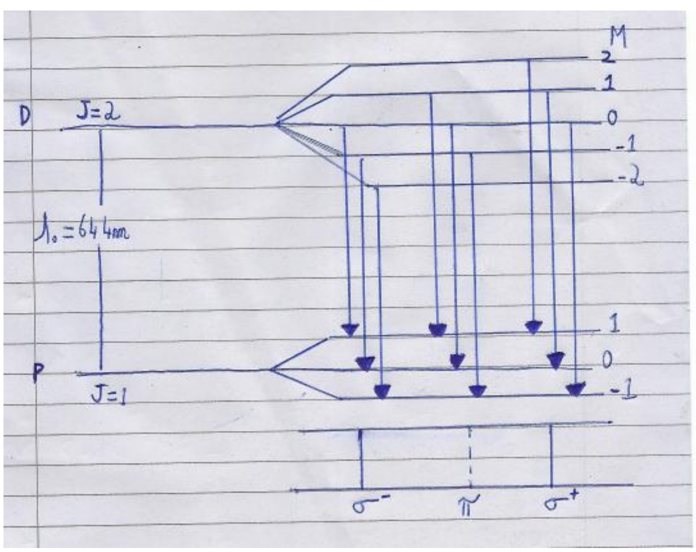
The second part of this experiment was to observe the properties of the transverse zeeman effect. It was found that $\triangle\lambda \propto$ B(the magnetic field). The bohr magneton was calculated to be $(11.554415122242694 \times 10^{-24}) \pm (2.5 \times 10^{-24})JT^{-1}$ which is within experimental error of the literature value$(9.2740100783 \times 10^{-24}JT^{-1})$. It was also noted that the outer triplet lines were linearly polarised at 90° angles with respect to the inner triplet line.

The final part of this experiment was to observe the properties of the longitudinal zeeman effect. It was observed that for the same magnetic field, the same shift in triplet lines $\triangle\lambda$ was observed. For magnetic fields caused by 5A and 7A these shifts were approximately $(9.72\times10^{-12})m\pm(3.9841347\times 10^{-12})m$ and $(1.00147\times10^{-11})m\pm(3.2\times10^{-12})m$ respectively. It was also observed that the two doublet lines were circularly polarised at 90° angles to one another.

3 Theory

Note: many theoretical derivations included as an appendix but a brief intro and some derivations are included here in order to clarify the experiment.
The zeeman effect was discovered by the dutch physicist Pieter Zeeman in 1902. The zeeman effect says that degeneracy in atomic energy levels can be removed via the application of a magnetic field which results in the splitting of spectral lines. This has utility in areas like astrophysics where the zeeman effect is observed around sunspots, indicating that they have a significant magnetic field.

If we are to speak about magnetism in any formal manner then the magnetic moment must be elaborated upon. The magnetic moment of an atom is influenced by its total angular momentum J. J is the sum of the spin angular momenta (S) and the orbital angular momenta(L). The spectroscopic state of an atom(i.e, how we observe its spectral lines) is determined by these three values of S, L and J. This is more commonly rewritten as $^{2S+1}L_J$. These normally have integer values of L from i.e 0,1,2,3 represented by the letter S,P,D and F. The transitions for the cadmium sample are denoted in the image below.



This energy splitting diagram results in a π component, and 2 $\pm\sigma$ components which each manifest as a spectral line, thereby creating triplets.
If the longitudinal setup, the lines are observed parallel to the magnetic field. The π component is not capable of being observed parallel too the magnetic field and thus the ring pattern manifests as a series of doublets, each a $\pm\sigma$ component.

The external magnetic field is applied along the z direction and as such we must focus on that in our analysis. The z direction has a quantum number M_J associated with it. Therefore

$$\triangle E = U_B M_J G$$

where G is the lande factor. A factor relevant to the energy levels when a weak magnetic field is applied. This factor usually manifests in the form

$$G = 1 + \frac{J(J+1) + S(S+1) - L(L+1)}{2J(J+1)}$$

The $\triangle\lambda$ between sets of triplets or doublets can be obtained by utilising the equation.

$$\triangle\lambda = \frac{\lambda_0^2 U_B B}{hc}$$

This is derived using the following equations:

$$E = -uB$$

(magnetic moment times magnetic field)

$$\triangle E = -u_B B$$

(bohr magneton by magnetic field)

$$E = \frac{hc}{\lambda}, \lambda = \frac{hc}{E}$$

$$\triangle\lambda = \frac{hc}{E} \times \frac{\triangle E}{E}$$

using $E = hc/\lambda$ in this equation

$$\triangle\lambda = \frac{\lambda_0^2 U_B B}{hc}$$

4 experimental setup and procedure

Experiment 1:(determine wavelength of red Cadmium light)

The aim of this experiment was to measure the wavelength of red cadmium light from a lamp source. A sketch of the setup and a photo is provided in figures 2 and 3 in the appendix. A cadmium lamp is turned on, the light then travels through lens 1, then through etalon, lens 2 further down the setup, a red cadmium light filter and finally into an eyepiece/videocam. The eyepiece is used for seeing the rings by eye, the video cam is used to obtain intensity distributions using the relevant videocom software on the pc connected to this. The optimal position of the setup was found by recording the intensity distribution at 256 pixels and adjusting until a very visible distribution of peaks was obtained. Once setup, a reading of the intensity was taken at 2048 pixels. Taking each

pixel to be a distance of 14 μm the radius as a function of ring order was measured and a plot of R_m^2 vs m was made. Since n and f were known as constants, the slope of this graph was used to obtain the value of λ_0.
This was done using the following equation

$$m\lambda_0 = 2nd(1 - \frac{r_m^2}{2n^2 f^2})$$

rearranging to gain a y=mx +c equation with radius squared as the y value and m as the x value we obtain:

$$(\frac{nf^2\lambda_0 m}{d}) - (\frac{2n^2 f^2}{d}) = r_m^2$$

$$mx + c = y$$

$$m = \frac{nf^2\lambda_0}{d}, \lambda_0 = m \times \frac{d}{nf^2}$$

Experiment:2 (Transverse zeeman effect)

The aim of this experiment was to measure the shift $\pm\triangle\lambda$ of the outer triplet lines and to show that $\triangle\lambda$ $\propto$B(the magnetic field). $\triangle\lambda$ can be measured by acknowledging that it is proportional to $r\triangle R$ (see theory). The procedure for experiment 1 was repeated but in this case a magnetic field was introduced by connecting the magnetic to a constant current source of a given desired value(see appendix figure 4). Intensity distributions were taken for current values of 4A,5A,6A and 7A, using the magnetic field calibration curve to calculate the value of B(mT). For each of these intensities, $r\triangle R$ was measured many times and an average value was taken. These values of $r\triangle R$ were plotted against the magnetic field values and using the slope of this graph, the value of the bohr magneton was obtained. This was done using the following equations.

$$\triangle\lambda = \frac{\lambda_0^2 U_B B}{hc}, \frac{\triangle\lambda}{\lambda_0} = \frac{-\lambda_0 r\triangle R}{n^2 f^2}$$

$$\triangle\lambda = \frac{-\lambda_0 r\triangle R}{n^2 f^2}$$

$$\frac{-\lambda_0 r\triangle R}{n f^2} = \frac{\lambda_0^2 U_B B}{hc}$$

$$r\triangle R = \frac{\lambda_0 n^2 f^2 U_B B}{hc}, y = mx + c$$

$$U_B = m(\frac{hc}{\lambda_0 n^2 f^2}$$

The second part of this experiment involved determining the polarisation state of the triplet lines. This was evaluated by placing a linear polariser and or a quarter wave plate in the setup and observing what effect the introduction of these had and what effect the change of angles had on the triplet lines. Screenshots of the triplet lines were taken using the partner camera and can be seen in the report. Intensity distributions were also taken for multiple angles at a constant current (6A).

It should be noted that experiment 2 was performed twice as it was recognised that errors had been made after data analysis of the first lab. The difference between the first and second trials of this experiment was that the angle of the lamp was changed slightly so as to attain a less skewed distribution than previously obtained. The results of both are included here and improvements gained are noted.

Experiment:3(The longitudinal zeeman effect)

The aims of this experiment were to show that for the same field there is the same $\triangle\lambda$ in the outer triplet lines as for the transverse zeeman effect. And to determine the polarisation of the two doublet lines. This was performed via the same procedure as experiment 2(i.e constant current source figure 4), the only difference being that the magnet was turned at a 90° angle to its original position as seen in figure 5. This allows a longitudinal rather than a transverse effect to be observed. The polarisation state of the doublet lines was observed by seeing what effects the linear polariser and or quarter wave plate had on the intensity distribution at various angles for the same current value.

Magnetic field values(mT) for varying currents(from manual

I [increasing] (A)	B (mT)	I [decreasing] (A)	B (mT)
4	394	7	594
5	485	6	555
6	552	5	493
7	594	4	407

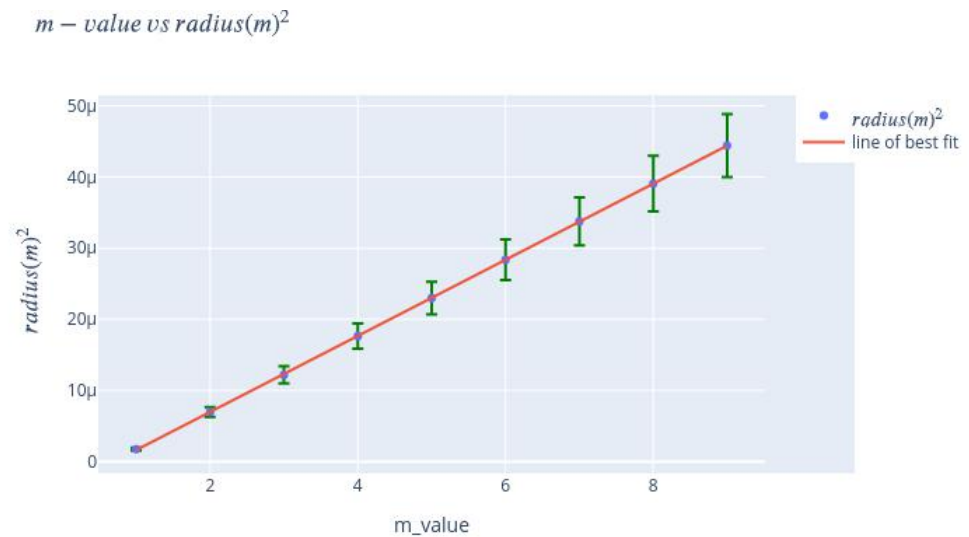
5 results and discussion

Experiment 1

Experiment 1 was carried out as detailed in the experimental procedure. From the distribution, the following values were obtained and using to construct a line fit. The error bars were given in percent value of the total as they were not very visible otherwise. This allows the relationship between error and radius to be seen more clearly. The intensity distribution is included in the appendix.

	m value	$radius(m)^2$	radius(m)	error r(m)	bestfit	error r squared$(m)^2$	% error r squared
0	1.0	1.75e-06	0.001323	1.4e-05	1.63e-06	4e-08	2.11640212
1	2.0	6.96e-06	0.002639	1.4e-05	6.98e-06	7e-08	1.06100796
2	3.0	1.22e-05	0.003493	1.4e-05	1.232e-05	1e-07	0.80160321
3	4.0	1.764e-05	0.0042	1.4e-05	1.767e-05	1.2e-07	0.66666667
4	5.0	2.299e-05	0.004795	1.4e-05	2.302e-05	1.3e-07	0.58394161
5	6.0	2.838e-05	0.005327	1.4e-05	2.837e-05	1.5e-07	0.52562418
6	7.0	3.376e-05	0.00581	1.4e-05	3.371e-05	1.6e-07	0.48192771
7	8.0	3.908e-05	0.006251	1.4e-05	3.906e-05	1.8e-07	0.44792833
8	9.0	4.441e-05	0.006664	1.4e-05	4.441e-05	1.9e-07	0.42016807

Experiment 1:best fit R_m^2 vs m



LinregressResult(slope=5.346888983333333e-06, intercept=-3.7161504722222233e-06, rvalue=0.9999901647122931, pvalue=9.823983708723082e-18, stderr=8.963216419708508e-09)
These values were obtained using the scipy linear regression function. Using the slope here, λ_0 was calculated using the formula

$$\lambda_0 = m(d/nf^2)$$

where m is as above, d=4mm,n=1.457 and f=15cm Using this I obtained $\lambda_0 = (652 \pm 8)nm$ The expected literature value is $643.8469nm$.(/www.rp-photonics.com/standard_spectral_lines.html). The obtained value is within 2% error of the expected value.

Experiment 2: Transverse zeeman effect

In experiment 2,4 datasets were analysed for current values of 4A,5A,6A and 7A. Each intensity distribution obtained was analysed and a graph of each was included in the appendix. Below we shall analyse the data tables and the average value of $r\triangle R$ obtained.

Relevant data from 4A current reading

	r(m)	del r(m)	r(del_r)m^2	error radius(m)	error r(del_r)m^2
0	0.001617	0.001183	1.91e-06	1.4e-05	6e-08
1	0.0028	0.000826	2.31e-06	1.4e-05	9e-08
2	0.003626	0.000665	2.41e-06	1.4e-05	1.1e-07
3	0.004291	0.000581	2.49e-06	1.4e-05	1.3e-07
4	0.004872	0.000518	2.52e-06	1.4e-05	1.4e-07
5	0.00539	0.000476	2.57e-06	1.4e-05	1.6e-07
6	0.005866	0.000434	2.55e-06	1.4e-05	1.7e-07
7	0.0063	0.000413	2.6e-06	1.4e-05	1.8e-07
8	0.006713	0.000385	2.58e-06	1.4e-05	1.9e-07
9	0.007098	0.000364	2.58e-06	1.4e-05	2e-07
10	0.007462	0.00035	2.61e-06	1.4e-05	2.1e-07
11	0.007812	0.000336	2.62e-06	1.4e-05	2.2e-07
12	0.008148	0.000308	2.51e-06	1.4e-05	2.3e-07
13	0.008456	0.000315	2.66e-06	1.4e-05	2.4e-07
14	0.008771	0.000301	2.64e-06	1.4e-05	2.5e-07
15	0.009072	0.00028	2.54e-06	1.4e-05	2.6e-07
16	0.009352	0.00028	2.62e-06	1.4e-05	2.7e-07
17	0.009632	0.000273	2.63e-06	1.4e-05	2.7e-07
18	0.009905	0.000259	2.57e-06	1.4e-05	2.8e-07
19	0.010164	0.000259	2.63e-06	1.4e-05	2.9e-07
20	0.010423	0.000245	2.55e-06	1.4e-05	3e-07
21	0.010668	0.000238	2.54e-06	1.4e-05	3e-07
22	0.010906	0.000231	2.52e-06	1.4e-05	3.1e-07
23	0.011137	0.000231	2.57e-06	1.4e-05	3.2e-07
24	0.011368	0.000231	2.63e-06	1.4e-05	3.2e-07

$(2.3011597777777742 \times 10^{-6}) \pm (1.7564826594839655 \times 10^{-7})$ is the average value of $r\triangle R$.

Relevant data from 5A current

	r(m)	del r(m)	r(del_r)m^2	error radius(m)	error r(del_r)m^2
0	0.001358	0.000308	4.2e-07	1.4e-05	4e-08
1	0.002674	0.000182	4.9e-07	1.4e-05	8e-08
2	0.003556	0.000126	4.5e-07	1.4e-05	1e-07
3	0.004242	0.000112	4.8e-07	1.4e-05	1.2e-07

$(4.5702300000000056 \times^{-7}) \pm (1.6904359533411313 \times 10^{-7})$ is the average value of $r\triangle R$

Relevant data from 6A

	r(m)	del r(m)	r(del_r)m^2	error radius(m)	error r(del_r)m^2
0	0.002695	0.00021	5.7e-07	1.4e-05	8e-08
1	0.003549	0.000161	5.7e-07	1.4e-05	1e-07
2	0.004228	0.000133	5.6e-07	1.4e-05	1.2e-07
3	0.004816	0.000112	5.4e-07	1.4e-05	1.4e-07
4	0.005341	0.000105	5.6e-07	1.4e-05	1.5e-07
5	0.005824	8.4e-05	4.9e-07	1.4e-05	1.6e-07
6	0.006258	9.1e-05	5.7e-07	1.4e-05	1.8e-07

current

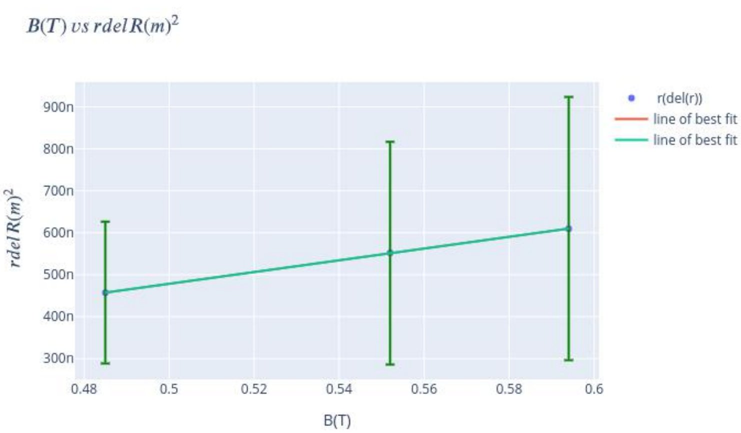
$(5.51222e \times 10^{-7}) \pm (2.65272 \times 10^{-7})$ is the average value of $r\triangle R$)

Relevant data from 7A current

	r(m)	del r(m)	r(del_r)m^2	error radius(m)	error r(del_r)m^2
0	0.002576	0.000231	6e-07	1.4e-05	8e-08
1	0.003458	0.000168	5.8e-07	1.4e-05	1e-07
2	0.004158	0.00014	5.8e-07	1.4e-05	1.2e-07
3	0.004746	0.000133	6.3e-07	1.4e-05	1.3e-07
4	0.005278	0.000119	6.3e-07	1.4e-05	1.5e-07
5	0.005768	0.000105	6.1e-07	1.4e-05	1.6e-07
6	0.006216	9.1e-05	5.7e-07	1.4e-05	1.8e-07
7	0.006622	9.1e-05	6e-07	1.4e-05	1.9e-07
8	0.007014	9.1e-05	6.4e-07	1.4e-05	2e-07
9	0.007385	8.4e-05	6.2e-07	1.4e-05	2.1e-07
10	0.007735	8.4e-05	6.5e-07	1.4e-05	2.2e-07

$(6.090610909090922 \times 10^{-7}) \pm (3.1372472727272727 \times 10^{-7})$ is the average value of $r\triangle R$

Since we are trying to demonstrate the proportionality between $\triangle\lambda$ and B, the average value for 4A is obviously an outlier. As can be seen on the figure in the appendix, it was difficult to discern triplets on the graph at such a current value which would hinder me from measuring$\triangle\lambda$ in triplets. Given this, only the last 3 values were used to make a plot of $Bvsr\triangle R$ as these clearly demonstrate a proportionality and have a reliable amount of triplets in their distributions to be valuable data.



The following data was obtained from the linefit using scipy's linear regression analysis.
1.3999889709937132e-06 is the slope of the graph
-2.2166067056358204e-07 intercept
0.9999516102025428 rvalue
0.006262875439864906 pvalue
1.377311825587088e-08 standard error

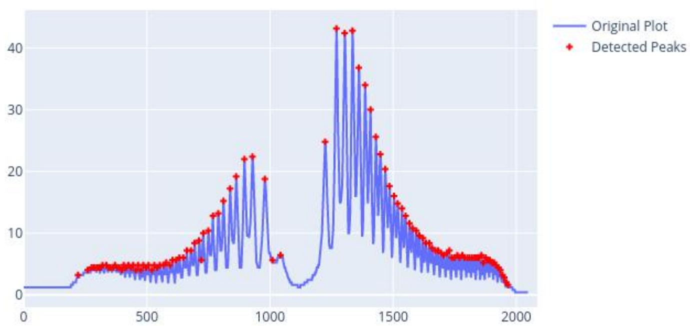
The value of the bohr magneton was obtained by using the slope of the graph and the following formula:

$$U_b = (m)(hc/(\lambda_0 n^2 f^2))$$

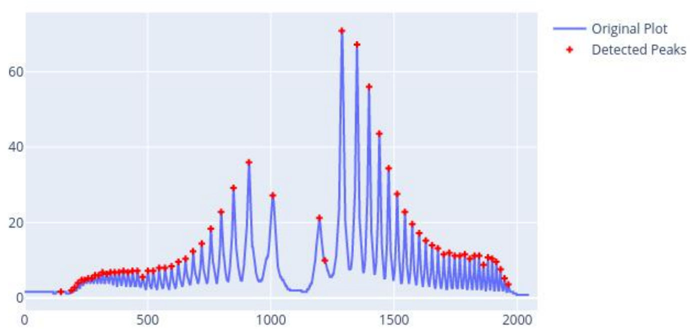
where $hc = 1.98644586 \times 10^{-25}$, n=1.457, λ_0=652nm, f=15cm and m = the slope of the line. The value of the bohr magneton obtained from this was $(11.554415122242694 \times 10^{-24}) \pm (2.5 \times 10^{-24}) JT^{-1}$ The value of the bohr magneton obtained from literature(https://physics.nist.gov/cgi-bin/cuu/Value?mub) was $9.2740100783 \times 10^{-24} JT^{-1}$. The obtained value is within 25% error of the expected value but within experimental error range of the expected value. Any error here is likely due to the use of 3 points instead of 4 as required.

The second part of this experiment was to determine the polarisation state of the triplet lines with respect to each other. Distributions were taken at 6A for values of 0° and 90° as this is where the variation can be most easily seen.

Distribution at 0° with linear polariser below(quarter polariser was observed to have no effect).



Distribution at 90° with linear polariser below(quarter polariser was observed to have no effect).



As can be observed above, at 0° using the linear polariser, the middle triplet line disappears. However, at 90°, the outer lines have disappeared and only the middle one is present. This means the outer lines are linearly polarised at 90° angles with respect to the inner line and vice a versa. Photos of this phenomenon from the eyepiece are also included in the appendix along with all intensity distributions for part 2.

Experiment 3: The longitudinal zeeman effect The first aim of this experiment was to show that there is the same $\triangle\lambda_0$ in λ for the same field in the transverse and longitudinal zeeman effect.
Below is the tabulated data and average value of $r\triangle R$ for 7A in the longitudinal effect setup.

	r(m)	del r(m)	r(del_r)m^2	error radius(m)	error r(del_r)m^2
0	0.000721	0.000847	6.1e-07	1.4e-05	3e-08
1	0.001568	0.000462	7.2e-07	1.4e-05	5e-08
2	0.002107	0.000322	6.8e-07	1.4e-05	6e-08
3	0.002667	0.000273	7.3e-07	1.4e-05	8e-08
4	0.003129	0.000231	7.2e-07	1.4e-05	9e-08
5	0.003535	0.000217	7.7e-07	1.4e-05	1e-07
6	0.003892	0.000196	7.6e-07	1.4e-05	1.1e-07
7	0.004214	0.000175	7.4e-07	1.4e-05	1.2e-07
8	0.004515	0.000168	7.6e-07	1.4e-05	1.3e-07
9	0.004802	0.000154	7.4e-07	1.4e-05	1.4e-07
10	0.005075	0.000147	7.5e-07	1.4e-05	1.4e-07
11	0.005327	0.00014	7.5e-07	1.4e-05	1.5e-07
12	0.005572	0.000133	7.4e-07	1.4e-05	1.6e-07
13	0.005803	0.000133	7.7e-07	1.4e-05	1.6e-07
14	0.006027	0.000119	7.2e-07	1.4e-05	1.7e-07
15	0.006244	0.000126	7.9e-07	1.4e-05	1.8e-07

$(7.336555624999991 \times 10^{-7}) \pm (2.3491824999999997 \times 10^{-7})$ is the average value of $r\triangle R$.
The corresponding value taken in the same field but in the transverse orientation was $(6.090610909090922 \times^{-7}) \pm (3.1372472727272727 \times 10^{-7})$. These values are somewhat different , however, they are both within experimental error range of each other. The % difference in the values is 17% with a difference of
$(1.24 \times 10^{-7}) \pm (5.470358864 \times 10^{-7})$. Since these values are within experimental range we can take the principle as proven here. To convert $r\triangle R$ into a quantifiable value in metres we use the following equation

$$\triangle\lambda = (\lambda_0)(r\triangle R)(1/(nf)^2$$

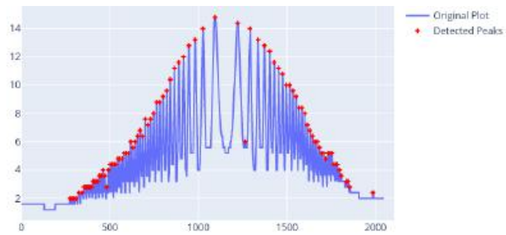
Using this and taking the value from the longitudinal effect part, $\triangle\lambda$ for a field of 7A is returned as approximately
$(1.00147 \times 10^{-11})m \pm (3.2 \times 10^{-12})m$

Now let's evaluate a 5A current value.
Below is the tabulated data and average value of $r\triangle R$ for 5A in the longitudinal effect setup.

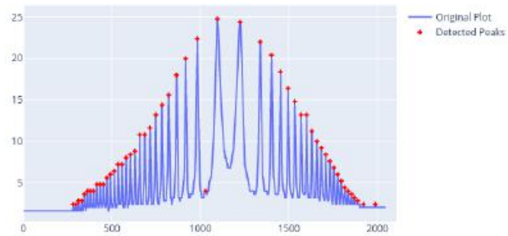
	r(m)	del r(m)	r(del_r)m^2	error radius(m)	error r(del_r)m^2
0	0.000889	0.000826	7.3e-07	1.4e-05	4e-08
1	0.001715	0.000406	7e-07	1.4e-05	5e-08
2	0.002464	0.000301	7.4e-07	1.4e-05	7e-08
3	0.00287	0.000245	7e-07	1.4e-05	8e-08
4	0.003381	0.000217	7.3e-07	1.4e-05	1e-07
5	0.003682	0.00021	7.7e-07	1.4e-05	1.1e-07
6	0.004095	0.000182	7.5e-07	1.4e-05	1.2e-07
7	0.00434	0.000175	7.6e-07	1.4e-05	1.2e-07
8	0.004697	0.000154	7.2e-07	1.4e-05	1.3e-07
9	0.004914	0.000147	7.2e-07	1.4e-05	1.4e-07
10	0.005222	0.00014	7.3e-07	1.4e-05	1.5e-07
11	0.005432	0.000126	6.8e-07	1.4e-05	1.5e-07
12	0.005719	0.000126	7.2e-07	1.4e-05	1.6e-07
13	0.005901	0.000126	7.4e-07	1.4e-05	1.7e-07
14	0.00616	0.000112	6.9e-07	1.4e-05	1.7e-07
15	0.006335	0.000119	7.5e-07	1.4e-05	1.8e-07
16	0.00658	0.000112	7.4e-07	1.4e-05	1.9e-07
17	0.006734	9.8e-05	6.6e-07	1.4e-05	1.9e-07
18	0.006972	9.8e-05	6.8e-07	1.4e-05	2e-07
19	0.007119	9.1e-05	6.5e-07	1.4e-05	2e-07
20	0.007343	9.1e-05	6.7e-07	1.4e-05	2.1e-07
21	0.007483	9.1e-05	6.8e-07	1.4e-05	2.1e-07
22	0.007693	8.4e-05	6.5e-07	1.4e-05	2.2e-07

$(7.121553478260867 \times 10^{-7}) \pm (2.9187808695652173 \times 10^{-7})$ is the average value of $r\triangle R$. The corresponding value taken in the same field but in the transverse orientation was $(4.5702300000000056 \times 10^{-7}) \pm (1.6904359533411313 \times 10^{-7})$. These values are within experimental error range of each other if we consider the max value of error the longitudinal value has $(2.9187808695652173 \times 10^{-7})$, this accounts for the difference in them. % difference in these values (in respect to the longitudinal value) is 35% which is appreciably close to acknowledge that for the same field there is approximately the same shift $\triangle\lambda$ for the transverse and longitudinal effect but appreciably different to acknowledge the size of the error bars in this experiment. As for the 5A current evaluation the final value of $\triangle\lambda$ is obtained by using the above formula, taking the longitudinal effect value and gaining an approximate value of $(9.72 \times 10^{-12})m \pm (3.9841347 \times 10^{-12})m$ The intensity distributions used for this are included in the appendix as we are not evaluating their appearance here.

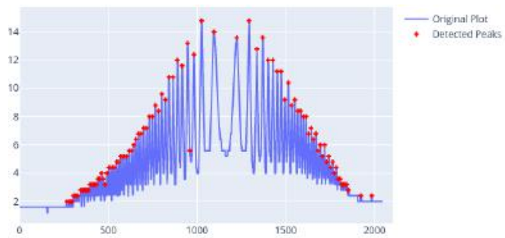
The second part of this experiment involved testing the polarisation of the doublet lines obtained, similarly to in experiment 2. The linear polariser alone had no effect. It only had an effect when placed in tandem with the quarter wave plate. Distributions for various angles(0,45 and 90 degrees) were taken and are shown below.
Intensity distribution 7A 0° using linear polariser and quarter wave plate



Intensity distribution 7A 45° using linear polariser and quarter wave plate



Intensity distribution 7A 90° using linear polariser and quarter wave plate



Since the use of a quarter wave plate was required it can be seen that these doublets are circularly polarised. At 45 degrees it appears that only one doublet is visible, both are visible at 0 degrees and 90 degrees which could mean they are polarised at 90 degree angles with respect to each other like for part 2 with the distincton of being circularly polarised.

6 error analysis

The error analysis for each experiment shall be elaborated upon in detail here.
Experiment 1:
The distance between each pixel is $14\mu m$, therefore the error in the distance r was $\pm 14\mu m$. There was no error in m since that was simply a count of ring order. The error in R_m^2 was measured by multiplying the fractional uncertainty of r by 2 and using this value upon each value of R_m^2. i.e

$$error R_m^2 = (2 \times ((14 \times 10^{-6})/R_m i) \times (R_m i^2)$$

where i represents values of R_m ranging from m=0 to m=9. The radius squared grows faster than 1/r decreases, therefore, the error in this experiment is proportional to the size of the radius squared and will only increase as radius does. When calculating the value of λ_0, the formula

$$\lambda_0 = m(d/nf^2)$$

was utilised. The error in this is equal to the respective error in the given values inserted into the equation error f= $10^{-3}m$, error slope= $48.963216419708508 \times 10^{-9}$, error d=$10^{-4}m$. Inserting these into the equation one leaving values with no error as 1, we gain $\pm 8nm$. Rounding errors may be present here due to the use of python to process the data and carry out all calculations.
Experiment 2:
The error in the radius remains the same here($14\mu m$). The error in $r\triangle R$ is somewhat different from part 1. Since the fractional uncertainties are different, we must add their respective uncertainties i.e

$$err(r\triangle R) = ((14 \times 10^{-6})/(R_m i) + (14 \times 10^{-6})/(\triangle R_m i)) \times (r_i \triangle R_i)$$

Where i ranges from the 1st element evaluated to the last element evaluated. Once again, error increases as radius increases. Rounding errors may be present here due to the use of python to process the data and carry out all calculations. The uncertainty of the $r\triangle R$ vs B graph was quite large due to the use of only 3 points. The 4A graph evaluated seemed to be unusable, this may be due to an non-optimal setup or possibly an error in the python code, though I was not able to locate this myself if there was any. The value of the bohr magneton was calculated using the formula

$$U_b = (m)(hc/(\lambda_0 n^2 f^2))$$

The error in the slope was obtained from the scipy regression standard error=$4.12267061911854 \times 10^{-7}$
The error in lambda = $\pm 8nm$ from the previous experiment. The error in f =$10^{-2}m$
. The error value obtained upon plugging in values with error into this equation and taking ones with no error as 1 was an error of $\pm 2.5 \times 10^{-24} JT^{-1}$.
The error in each average value of $r\triangle R$ was given by the sum of the errors divided by the square root of the sample size. This square root rule applies to any sample size error. i.e

$$error = (\sum_{n=1}^{N}(error_N)) \times (1/\sqrt{N})$$

Experiment 3:
The error evaluation is much the same as in part 2 for this experiment, however, the error in the value of $\triangle\lambda$ must also be obtained. $\triangle\lambda$ is related to $r\triangle R$ by the equation

$$\triangle\lambda = \lambda \times (r\triangle R/(nf)^2$$

since these factors by $r\triangle R$ are just a constant factor, whatever the error was in $r\triangle R$ multiplied by $lambda \times (1/(nf)^2$ is the error in $\triangle\lambda$. This factor has a value of $1.3650418745640001 \times 10^{-5}$.

7 conclusion

In conclusion, this experiment was largely successful within the margin of experimental error, though in some parts less accuracy than required was obtained i.e using 3 points instead of 4 to determine the bohr magneton. The wavelength of red cadmium light was found to be 652nm which agreed nicely with the literature value of 642nm, the polarisation of the transverse and longitudinal effects as well as the correlation in their wavelength shifts was proven and described, and from the transverse zeeman effect, a bohr magneton within experimental range of the expected value of $(11.554415122242694 \times 10^{-24}) \pm (2.5 \times 10^{-24}) JT^{-1}$ was calculated. The experiment was generally successful in finding qualitiative values of the zeeman effect and only fell somewhat short in numerical accuracy for some aspects.

8 Bibliography

1.Trinity college Dublin/junior sophister/lab manual ("The zeeman effect").
2./www.rp-photonics.com/standard_spectral_lines.html
3.https://physics.nist.gov/cgi-bin/cuu/Value?mub
4. Optics third edition/Eugene Hecht/ chapter 9.6 (interference/multiple beam interference/ The fabry-perot interferometer)/pages 413-418)
5. University physics with modern physics/ chapter 41.4(The zeeman effect)/pages 1399-1402.

9 appendix

10 appendix: experimental graphs + photos

Figure 1: Sketch of apparatus components

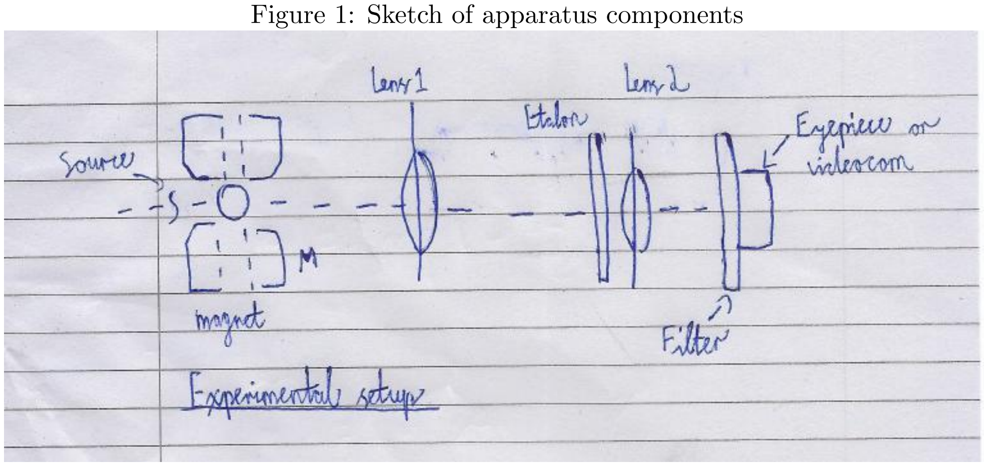


Figure 2: Setup for experiment 1: Measuring wavelength of red Cadmium light

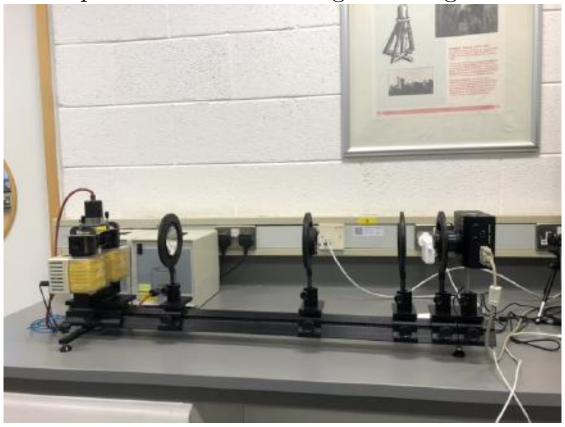


Figure 3: Setup for experiment 2 and 3: Turn constant current source on

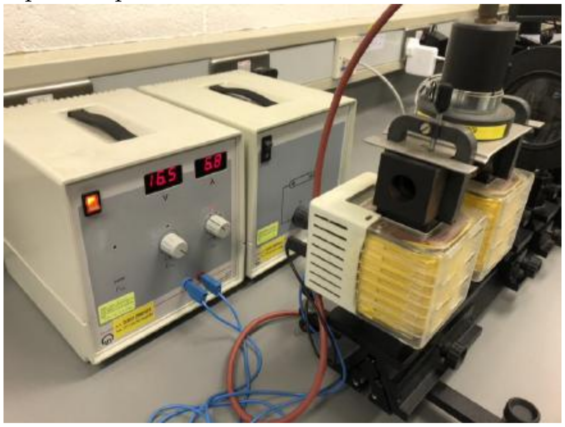
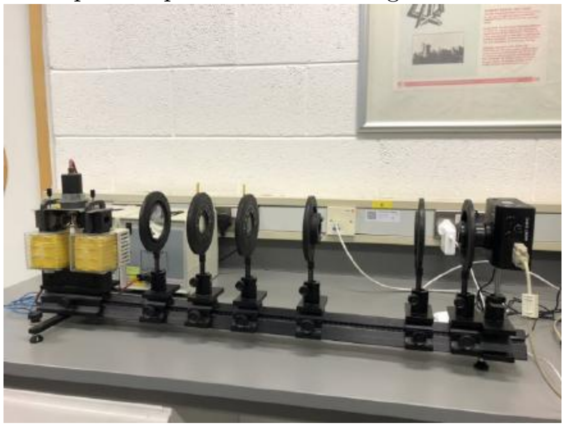


Figure 4: Setup for experiment 3:The longitudinal Zeeman effect



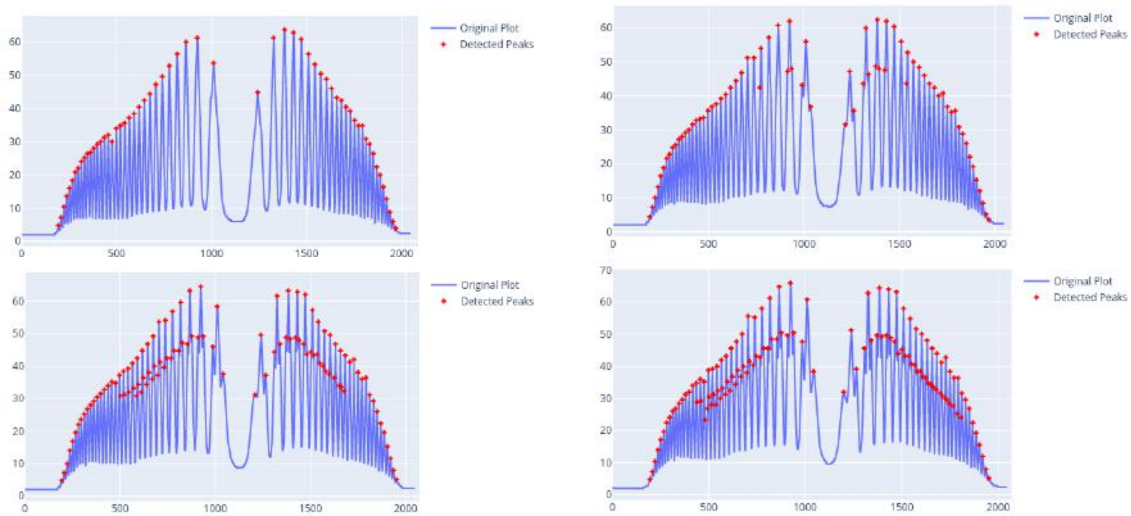


Figure 5: intensity distributions for current values of 4A,5A,6A and 7A.(4A top left, 5A top right, 6A bottom left,7A bottom right)

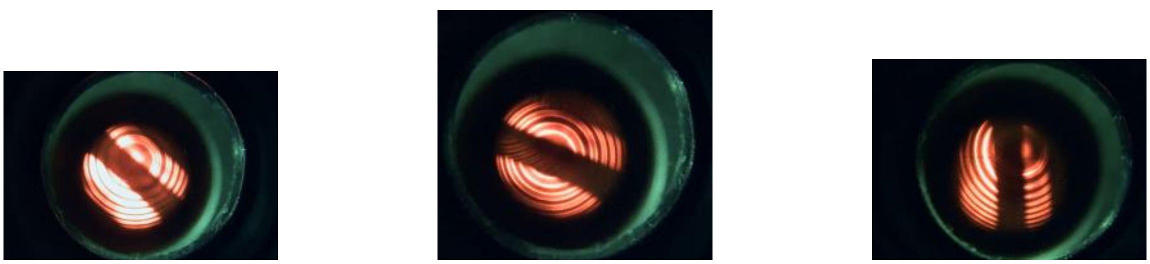


Figure 6: Pictures taken with the partner camera of the triplet lines at no polarisation, a 0 degree polarisation and a 90 degree polarisation in that order

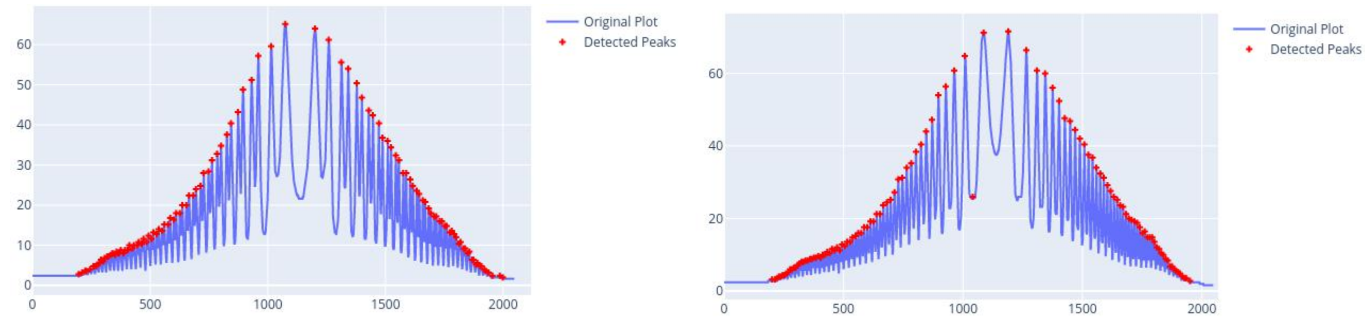


Figure 7: intensity distributions for current values of 5A and 7A in the longitudinal orientation

11 appendix: experimental code(python)

```python
import numpy as np
import os
import matplotlib.pyplot as plt
position,intensity=np.loadtxt('///home/Benjamin/Desktop/lab1 relevant files/exp1 peaks revised.csv', delimiter=',', skiprows=1,  unpack=True )
import plotly.graph_objects as go
import pandas as pd
from scipy.signal import find_peaks
fig = go.Figure(data=go.Scatter(y = intensity,mode = 'lines'))
intensities_peaks = find_peaks(intensity)[0]
fig = go.Figure()

fig.add_trace(go.Scatter(
    y=intensity,
    mode='lines',
    name='Original Plot'
))

fig.add_trace(go.Scatter(x=intensities_peaks,y=[intensity[j] for j in intensities_peaks]
                         ,mode='markers'
                         ,marker=dict(size=4,color='red',symbol='cross' ),
                         name='Detected Peaks'
))

fig.show()



#print(intensities_peaks)

sorting_out_noise= np.where(intensity>17.2)
#only accepting position values of a certain intensity
intensities_radius_right = sorted(i for i in intensities_peaks if i >= 1102)
intensities_radius_left=sorted((i for i in intensities_peaks if i<1100),reverse=True)
#defining left and right sides of circle relative to specific graph
intensities_radii_right=intensities_radius_right[:10]
```

```
position_no_noise=position[sorting_out_noise]
#condition where noise' is not present
intensities_radii_left = []
for element in intensities_radius_left:
    if element in position_no_noise:
        intensities_radii_left.append(element)
#sorts values based on this condition


left_right=zip(intensities_radii_left,intensities_radii_right)
def actual_radius(a):
    return (((a)/2)*(14)*((10)**-6))**2
#squared value of radius
def radius_value(a):
    return (((a)/2)*(14)*((10)**-6))
#radius value
radius_m_squared=[]
diameter=[]
radius_m_=[]
m=np.arange(1, 10, 1)
#for something in left_right:print(something)
for something in left_right:
    diameter.append(something[1]-something[0])
#the diameter is simply the difference in right and left peaks
for i in range(len(diameter)):
    radius_m_squared.append(actual_radius(diameter[i]))
for i in range(len(diameter)):
    radius_m_.append(radius_value(diameter[i]))
#obtain radius and radius squared in metres using the above functions

print(m)
print(radius_m_squared)
print(diameter)


import plotly.express as px
from sklearn.linear_model import LinearRegression
import plotly.graph_objects as go
import pandas as pd
df = pd.DataFrame({'m_value': m, '${radius(m)^2}$':radius_m_squared})
reg = LinearRegression().fit(np.vstack(df['m_value']), radius_m_squared)
df['radius(m)']= radius_m_
df['error r(m)']= 14*10**-6
df['bestfit'] = reg.predict(np.vstack(df['m_value']))
#df['error']= ((df['${radius(m)^2}$']**2)+(2*(df['${radius(m)^2}$']*14*10**-6))
    +((14*10**-6)**2))
df['error r squared$(m)^2$']=(2*((14*10**-6)/(df['radius(m)'])))*(df['${radius(m)^2}
    $'])
df['% error r squared']= (df['error r squared$(m)^2$']/df['${radius(m)^2}$'])*100
#df['error']= ((2*(df['${radius(m)^2}$']*14*10**-6)))
#df['percentage error']=(df['error']/df['${radius(m)^2}$'])*100
from pandas.plotting import table

pd.set_option('display.float_format', '{:.2g}'.format)
fig, ax = plt.subplots(figsize=(12, 2)) # set size frame
ax.xaxis.set_visible(False)
ax.yaxis.set_visible(False)  # obscures x and y axes
ax.set_frame_on(False)  # removes frame
```

```
tabla = table(ax, df.round(decimals=8), loc='upper right', colWidths=[0.17]*len(df.columns))  # where df is your data frame
tabla.auto_set_font_size(False) # allows fontsize to be set according to my needs
tabla.set_fontsize(12)

tabla.scale(1.2, 1.2) # sets size of table
plt.savefig('table.png', transparent=True)
#saves the figure

# plotly figure setup
fig=go.Figure()
fig.add_trace(go.Scatter(name=' ${radius(m)^2}$', x=df['m_value'], y=df['${radius(m)^2}$'].values, mode='markers',
                       error_y=dict(
        type='percent',
        array=df['error r squared$(m)^2$'],
        color='green',visible=True)) )
fig.add_trace(go.Scatter(name='line of best fit', x=m, y=df['bestfit'], mode='lines'))
#fig.add_trace(go.scatter( x=df['m_value'], y=df['${radius(m)^2}$'].values,mode='markers',
                 # error_y=df['error']))
#fig.add_trace(go.Scatter(
 #   x=df['m_value'], y=df['${radius(m)^2}$'],

  #  error_y=dict(
   #     type='data',
    #    array=df['error'],
     #   color='purple',visible=True)))
# plotly figure layout
fig.update_layout(xaxis_title = 'm_value', yaxis_title = '${radius(m)^2}$')
fig.update_layout(title="${m-value \, vs \, radius(m)^2}$")
#adds title and axes labels to the graph

fig.show()
#shows figure
from scipy.stats import linregress
linregress(m, radius_m_squared)
#shows detailed breakdown of linear regression values



from scipy.stats import linregress
linregress(m, radius_m_squared)
slope, intercept, r_value, p_value, std_err = linregress(m,radius_m_squared)
slope_line= slope
print(slope_line)



n=1.457
d=(4*10**-3)
f=(15*10**-2)
errorf=10**-3
errord=1*10**-4
wavelength0= (slope_line)*((d)/((n)*((f)**2)))
print('The wavelength of red Cadmium light is',wavelength0/10**-9,'nm')
wavelength1= (slope_line)*(d/((n)*(f**2)))
```

```
print(wavelength1/10**-9)
```

Listing 1: Code experiment1

```
#import relevant libraries and datafile
import numpy as np
import matplotlib.pyplot as plt
position,intensity=np.loadtxt('///home/Benjamin/Desktop/lab1 relevant files/exp2 7a
    revisedrevised.csv', delimiter=',', skiprows=1,  unpack=True )

import plotly.graph_objects as go
import pandas as pd
from scipy.signal import find_peaks
fig = go.Figure(data=go.Scatter(y = intensity,mode = 'lines'))
intensities_peaks = find_peaks(intensity)[0]
#locates intensities and organises them into an array

fig = go.Figure()

fig.add_trace(go.Scatter(
    y=intensity,
    mode='lines',
    name='Original Plot'
))

fig.add_trace(go.Scatter(x=intensities_peaks,y=[intensity[j] for j in
    intensities_peaks]
                         ,mode='markers'
                         ,marker=dict(size=5,color='red',symbol='cross' ),
                         name='Detected Peaks'
))

fig.show()

#makes plot of peaks and all intensity values versus position
#distinguishes peaks with a red cross symbol


sorting_out_noise= np.where(intensity>25)#subject to change depending on graph
intensities_radius_right = sorted((i for i in intensities_peaks if i >= 1157),
    reverse=False)
intensities_radius_left=sorted((i for i in intensities_peaks if i<1064),reverse=True
    )
# if we start in the center of the distribution this splits the intensities into
    right hand and left hand side peaks
#diameter is easily measured by the difference in these and therefore radius is
    easily measured
#indices_radii_right=indices_radius_right[0:-1]
intensities_radii_right=[]
position_no_noise=position[sorting_out_noise]
intensities_radii_left =[]
for element in intensities_radius_left:
    if element in position_no_noise:
        intensities_radii_left.append(element)
for element in intensities_radius_right:
    if element in position_no_noise:
        intensities_radii_right.append(element)
#filters out intensity values below a certain value as they are likely inaccurate
```

```
left_right=zip(intensities_radii_left,intensities_radii_right)
#makes a list of tuples
def actual_radius(a):
    return ((((a)/2)*(14)))*((10)**3)*10**-9
#defines function to calculate radius from raw pixel values of diameter
radius_m=[]
diameter=[]
#empty arrays

#for something in left_right:print(something)
for something in left_right:
    diameter.append(something[1]-something[0])
for i in range(len(diameter)):
    radius_m.append(actual_radius(diameter[i]))
#diameter ran through function to obtain individual radius values in metres

del_radius= [x - radius_m[i - 1] for i, x in enumerate(radius_m) if i > 0]
#del radius is just every second radius value minus the value that preceded it



r_deL_r=zip(radius_m[3:-1:3],del_radius[3:-1:3])
#creates another list of tuples
#filters these in such a manner that the difference between triplet peaks is
    concerned
#if not filtered this way, the difference between triplet groups would come into
    play and ruin the distribution
prod=[]
#this array will hold all r_del r values
r_deL_r=sorted(r_deL_r, key=lambda x: x[0])
#sorts these values in order depending on radius
for something in r_deL_r:
    prod.append(something[0]*something[1])
#finds product of r and del_r for each value of tuples
_radius=[]
_del_radius=[]
for something in r_deL_r:
    _radius.append(something[0])
for something in r_deL_r:
    _del_radius.append(something[1])
#creates arrays for data to be used in dataframe
prod_mean=np.mean(prod)
#finds mean r del r value

df=pd.DataFrame({'r(m)': _radius,'del r(m)':_del_radius})
df['r(del_r)$m^2$']=df['r(m)']*df['del r(m)']
df['error radius(m)']= 14*10**-6
df['error r(del_r)$m^2$']=(((2*14*10**-6)/df['del r(m)'])+((14*10**-6)/(df['r(m)']))
    )*(df['r(del_r)$m^2$'])
#define columns in dataframe
#all data now tabulated
error_of_mean_value= (((np.sum((df['error r(del_r)$m^2$'])))/(len(df['r(del_r)$m^2$'
    ]))))
from pandas.plotting import table

```

```
pd.set_option('display.float_format', '{:.2g}'.format)
fig, ax = plt.subplots(figsize=(12, 2)) # set size frame
ax.xaxis.set_visible(False)
ax.yaxis.set_visible(False)  # removes x and y axes
ax.set_frame_on(False)  # removes frame
tabla = table(ax, df.round(decimals=8), loc='upper right', colWidths=[0.17]*len(df.
    columns))
tabla.auto_set_font_size(False) # allows one to set fontsize manually
tabla.set_fontsize(12)

tabla.scale(1.2, 1.2) # change size of the returned table
plt.savefig('table.png', transparent=True)#saves image
print(df)
print(prod_mean,u"\u00B1",error_of_mean_value,'is the average value of r(del r)')
#prints result with error value beside it


from scipy.stats import linregress
x=[485*10**-3,552*10**-3,594*10**-3]
y=[4.57*10**-7,5.52*10**-7,6.094*10**-7]
#values r-delr vs B
linregress(x, y)
slope, intercept, r_value, p_value, std_err = linregress(x,y)
slope_line= slope
print(slope_line,'is the slope of the graph')
print(intercept,'intercept')
print(r_value,'r_value')
print(p_value,'p_value')
print(std_err,'standard error')
#prints each respective value of the scipy linear regression function
#linregress values
hc=1.98644586*10**-25
#constant
lambda_0= 652*10**-9
#from first experiment
#constants
U_b= (slope_line)*(hc/((lambda_0)))*(n^2f^2))
#note 20.937 is the value of (n**2f**2)**-1
print(U_b,'is the obtained value for the bohr magneton from our data')
#prints result
plt.plot(x,y)
#makes rough graph



from sklearn.linear_model import LinearRegression
#imports machine learning linear regression tool
import plotly.graph_objects as go
import pandas as pd
#imports dataset processing libraries
df = pd.DataFrame({'magnetic field(Tesla)': x, 'r(del(r))':y})
reg = LinearRegression().fit(np.vstack(df['magnetic field(Tesla)']), y)
#makes dataframe and uses these values
df['bestfit'] = reg.predict(np.vstack(df['magnetic field(Tesla)']))
#df['error']= ((df['${radius(m)^2}$']**2)+(2*(df['${radius(m)^2}$']*14*10**-6))
    +((14*10**-6)**2))
#df['error r del r']=(((2*14*10**-6)/df['del r'])+((14*10**-6)/(df['r'])))*(df['r(
    del_r)'])
```

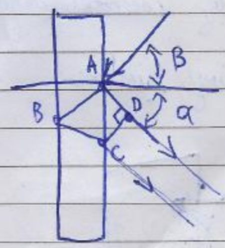
```python
#df['% error r del r']= (df['error r squared']/df['${radius(m)^2}$'])*100
#df['error']= ((2*(df['${radius(m)^2}$']*14*10**-6)))
#df['percentage error']=(df['error']/df['${radius(m)^2}$'])*100
print(df)
#different functions which return different columns of data e.g error
# plotly figure setup
fig=go.Figure()
fig.add_trace(go.Scatter(name=' r(del(r))', x=df['magnetic field(Tesla)'], y=df['r(
    del(r))'].values, mode='markers'))
fig.add_trace(go.Scatter(name='line of best fit', x=x, y=df['bestfit'], mode='lines'
    ,error_y=dict(
        type='data',
        array=[1.6904359533411313e-07 ,2.65272e-07, 3.1372472727272727e-07  ],
        color='green',visible=True)))
        #creates figure with error values given as array
fig.add_trace(go.Scatter(name='line of best fit', x=x, y=df['bestfit'], mode='lines'
    ))
fig.update_layout(xaxis_title = 'B(T)', yaxis_title = '${rdelR(m)^2}$')
fig.update_layout(title="${B(T) \, vs \, rdelR(m)^2}$")
#creates title and axes labels for figure
```

Listing 2: Code experiment2 through which all distributions from 4A to 7A were run. This code was also used for experiment 3

12 appendix: Theoretical derivations not in theory section

· Prove:

$$\Delta = 2nd\cos B$$



$$\Delta = n_f\left[(\overline{AB})+(\overline{BC})\right] - n_1(\overline{AD})$$

$$(\overline{AB}) = (\overline{BC}) = \frac{d}{\cos(B)}$$

$$\Delta = \frac{2n_f d}{\cos B} - n_1(\overline{AD})$$

$$(\overline{AD}) = (\overline{AC})\sin\alpha$$

$$(\overline{AD}) = (\overline{AC})\frac{n_f}{n_1}\sin B$$

$$(\overline{AC}) = 2d\tan B$$

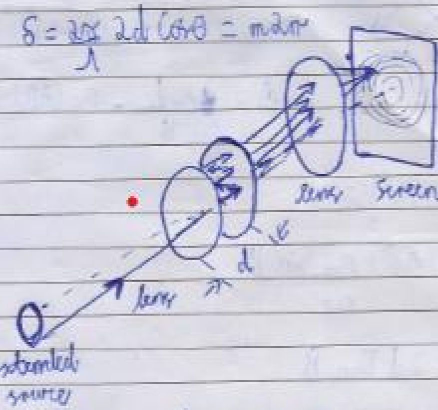
$$\Delta = \frac{2n_f d}{\cos B}(1 - \sin^2 B)$$

$$\therefore \Delta = 2n_f d\cos B$$

- Explain why, assuming there is a range of values of angle α, the interference pattern of light of wavelength λ should consist of concentric rings.

Fringes are described by airy function,

$$\delta = \frac{2\pi}{\lambda} 2d \cos\theta = m 2\pi$$



Parallel beams of light will converge on the same point when passed through a focal lens.
Since this happens for beams at the top and bottom of the lens for varying intensities, we obtain a circular fringe pattern.

- mathematical derivation:
remember Taylor series

$$f(x) = \frac{1}{1-x} = \sum_{n=0}^{\infty} x^n = 1 + x + x^2 + x^3 + \ldots$$

Incident wave $E_0 e^{-i\omega t}$ Transmitted as sum of waves

$$E_t = E_0 t_1 t_2 e^{-i\omega t} + E_0 t_1 t_2 r_1 r_2 e^{-i(\omega t - \delta)} + E_0 t_1 t_2 r_1^2 r_2^2 e^{-i(\omega t - 2\delta)}$$

Take the sum of this progression for $r_1 r_2 e^{i\delta}$

$$E_t = E_0 t_1 t_2 e^{i\omega t}\left[\frac{1}{1 - r_1 r_2 e^{i\delta}}\right]$$

and multiplying by the complex conjugate

$$I_t = E_t E_t^{*} = E_0^2 t_1^2 t_2^2 \left[\frac{1}{1 + r_1^2 r_2^2 - 2 r_1 r_2 \cos\delta}\right]$$

$$E_0^2 = I_0 \quad r_1 r_2 = R \quad t_1 t_2 = T \qquad \cos\delta = (1 - 2\sin^2\delta/2).$$

$$I_t = I_0 \frac{T^2}{(1-R)^2}\left[\frac{1}{1 + \frac{4R}{(1-R)^2}\sin^2\delta/2}\right]$$

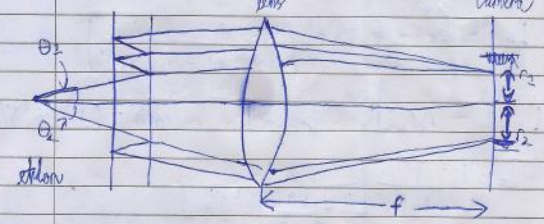
If there is no absorption in the reflecting surface $T = (1-R)$ then defining

$$\frac{4R}{(1-R)^2} = \phi$$

$$I_t = I_0\left[\frac{1}{1 + \phi\sin^2\delta/2}\right]$$

Produces the Airy function

- Show that when the rings are viewed in the focal plane of lens L2, focal length f, the radius r is related to the corresponding angle α by $\tan(\alpha) = \frac{r}{f}$



$$m\lambda = 2nd\cos\theta \rightarrow \tan\theta_m = \frac{r_m}{f}$$

$$r_m = f\tan\theta_m \approx f\theta_m$$

- Show that, to a good approximation

$$m\lambda = 2nd\left(1 - \frac{r_m^2}{2n^2f^2}\right)$$

- Proof:

$$m\lambda = 2nd\cos\theta$$

$$\cos^2\theta = 1 - \sin^2\theta$$

$$\cos\theta = 1 - \frac{2n^2\sin^2\theta_m}{2n^2}$$

for small angles $\sin\theta \approx \theta$

$$\cos\theta = 1 - \frac{\theta_m^2}{2n^2} \qquad \tan\theta = \frac{r_m}{f} \quad \theta \approx \frac{r_m}{f}$$

$$\cos\theta = 1 - \frac{r_m^2}{2n^2f^2}$$

$$m\lambda = 2nd\cos\theta$$

$$m\lambda = 2nd\left(1 - \frac{r_m^2}{2n^2f^2}\right)$$

What is meant by the order number of a ring? Show also that, to a good approximation

$$\frac{\Delta\lambda}{\lambda} = -\frac{r\Delta r}{n^2 f^2}$$

The order number of a ring refers to the number of wavelengths away from the center of the distributions at which a ring is located. This goes in integer numbers 1,2,3...etc.

$$m\lambda = 2nd\left(1 - \frac{r_m^2}{2n^2f^2}\right)$$

$$2m\lambda_2 = 2nd\left(1 - \frac{(r_m+\Delta r)^2}{2n^2f^2}\right) \qquad (r_m+\Delta r)^2 = r_m^2 + r_m\Delta r + \Delta r^2$$

$$m\lambda_0 = 2nd\left(1 - \frac{r_m^2}{2n^2f^2}\right)$$

$$\Delta\lambda = 2nd\left(1 - \frac{(r_m^2 + r_m\Delta r + \Delta r^2)}{2n^2f^2} - 1 + \frac{r_m^2}{2n^2f^2}\right)$$

$$\Delta\lambda = 2nd\left(\frac{r_m\Delta r + \Delta r^2}{2n^2f^2}\right)$$

Δr^2 is very small so $\frac{r_m\Delta r + \Delta r^2}{2n^2f^2} \approx \frac{r_m\Delta r}{2n^2f^2}$

$$\Delta\lambda = 2nd\left(\frac{r_m\Delta r}{2n^2f^2}\right) = $$

$$\lambda = 2nd\left(1 - \frac{r_m^2}{2n^2f^2}\right)$$

$$\frac{\Delta\lambda}{\lambda} = \frac{2r_m\Delta r}{2n^2f^2}\left(\frac{2n^2f^2}{2n^2f^2 - r_m^2}\right) = \frac{r_m\Delta r}{n^2f^2} - \cancel{\frac{r\Delta r}{r_m^2}}^{\,0}$$

$$\frac{\Delta\lambda}{\lambda} = -\frac{r_m\Delta r}{n^2f^2}$$