

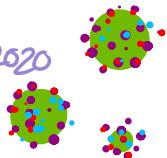
Risk assessment

20/06/2020

RISK ID		RISK DESCRIPTION	TREND	CURRENT	RESIDUAL
11003	SCI_SPA_Level 3_Gamma rays from Caesium_4.4 Beta-ray Spectrometer		Medium	Not Assessed	
RISK OWNER	RISK IDENTIFIED ON	LAST REVIEWED ON	NEXT SCHEDULED REVIEW		
Matthew Leong	25/02/2019	25/02/2019	25/02/2022		
RISK FACTOR(S)	EXISTING CONTROL(S)	PROPOSED CONTROL(S)	TREATMENT OWNER	DUE DATE	
Radiation - Sealed source Cs-137. Gamma ray radiation Exposure from the sealed Cs-137 (0.45uSv/hr @10cm 25/2/2019) (Beta rays will be fully attenuated.)	Control: Cs137 source is in a brass vacuum chamber surrounded by a lead sheet. Control: Operators are typically to stay at least 30cm away.	<i>Do not open or attempt to access the area where the radioactive source is. Do not handle the radioactive source</i>			
Radiation - Sealed source Cs-137. Gamma ray radiation Exposure from the sealed Cs-137 (0.45uSv/hr @10cm 25/2/2019) (Beta rays will be fully attenuated.)	Control: Cs137 source is in a brass vacuum chamber surrounded by a lead sheet. Control: Operators are typically to stay at least 30cm away.	<i>same as above</i>			
Physical Hazard-Stored Energy. Vacuum system can be hazardous if operation is disrupted incorrectly.	Control: Vacuum system only operated by teaching staff.	<i>Ask staff member if assistance is needed</i>			
Physical Hazard-Stored Energy. Vacuum system can be hazardous if operation is disrupted incorrectly.	Control: Vacuum system only operated by teaching staff.	<i>same as above</i>			
Physical Hazard-Electrical. Field Magnet power supply (20V DC, up to 10A)		<i>Apparatus is to be operated remotely. Ask staff member if assistance is needed</i>			
Physical Hazard-Electrical. Field Magnet power supply (20V DC, up to 10A)	Control: Coil terminal has a plastic cover, avoiding finger contact with terminal.	<i>same as above</i>			

- Measurement of β -ray Spectra -

05/08/2020



Ana Fabela Hinojosa - 27876594

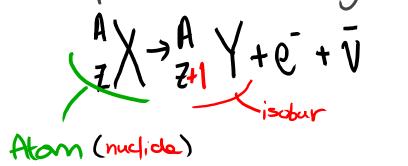
Booking of apparatus: 06/08 from 10:00 → 16:00 (original)
16/08 from 11:00 → 17:00 (additional)
18/08 from 10:00 → 16:00 (additional) ✓

Aim: we measure the energy spectrum of emitted β -rays and the total energy released by one of the nuclear transitions

Theory: There are 2-types of β -decay, we will focus on:

β^- -decay - electrons ejected from a nucleus such that a nuclide transforms into an isobar [1]

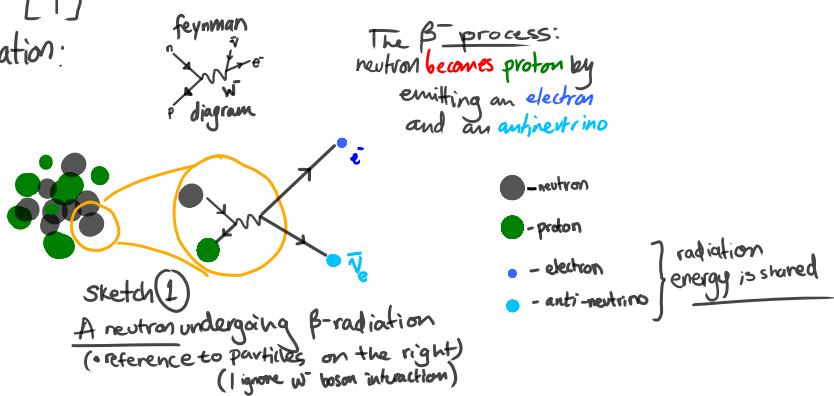
The process is described by the equation:



Atom (nuclide)

A - atomic number

Z - number of protons



The radiated electron may have kinetic energy ranging from zero to nearly the full disintegration energy. [6]

The total energy is divided between nucleide, electron, anti-neutrino [1]

Things to consider • The antineutrino has a rest mass $< 2 \times 10^{-7} m_e$ [6]

In the lab manual, we are shown in Fig 2. β -ray groups in the decay scheme of ^{137}Cs

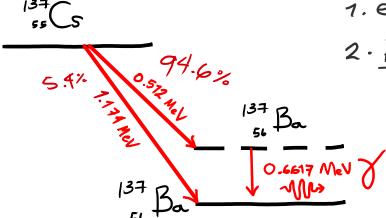
HIGHER ENERGY GROUP

An excited state of $^{137}_{55}\text{Ba}$
which de-excites by either

1. emitting a γ -ray [$E = 661.65 \text{ keV}$]

2. K conversion electron [$E = 624.21 \text{ keV}$]

common Internal conversion (IC)
electrons come from K shell (1s state) [5]



Sketch ② a copy of:
Fig 2 in Lab notes

WHY IS HIGHER ENERGY GROUP DIFFICULT
TO DETECT?

is it dependent on the degree of "forbiddenness"
of the transition?

nucleide : atomic species [2]

isobar : atoms of different elements that have the same number of nucleons [3]

Conversion electrons

The nucleus of an atom can be in an excited state after undergoing radioactive decay.

An atom can de-excite if this excess energy is transferred to an orbital electron - (likely a K-shell electron)

When the electrons are ejected from their orbits by gaining energy they are called conversion electrons and are emitted with a characteristic ENERGY.

[o] momentum spectrum of β particles

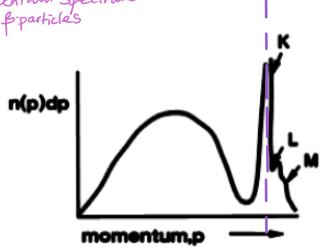


Figure 1: A representative momentum spectrum of beta particles from the decay of ^{137}Cs . The K, L and M conversion electron peaks are also depicted on the right. [o]

\times conversion line can be used for calibration of the momentum range of the spectrometer.

The energy of conversion electrons \approx $\left\{ \begin{array}{l} \text{The excited state energy of the nucleus} \\ \text{minus the binding energy of this inner shell electron to the atom.} \end{array} \right.$

We are told in the lab script that:

The ratio of the peak's amplitudes follows $\frac{K}{L+m} = 4$
Therefore we can assume that we will only observe the K conversion line.

-The Spectrometer-

Our apparatus is a thin magnetic lens spectrometer
(Detector: G-M tube with thin-wic end window)

05/08/2020

≈ 11/08/2020

BETA-RAY SPECTROMETER

35

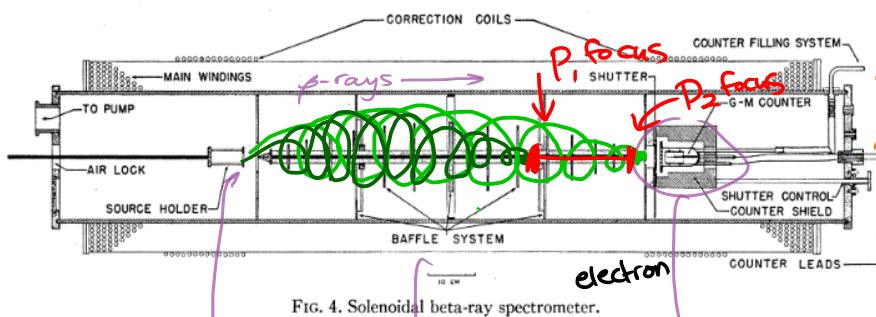


FIG. 4. Solenoidal beta-ray spectrometer.

^{137}Cs
only e^- of particular p (ie. T) range
are focused onto the detector
for a given $|B|$

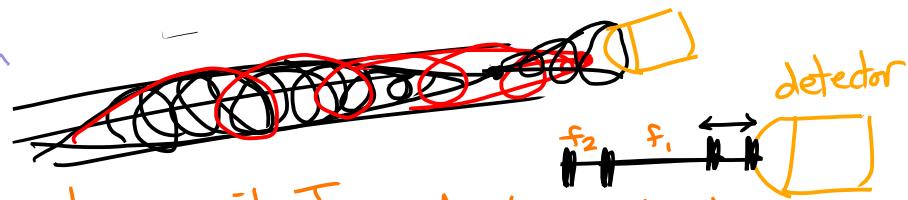
depends on I
of electromagnet

The effect of the magnetic field
on a cone of electron trajectories
diverging around the source
around the instrument's axis
and converging (depending
on their momentum)

[6] on the detector

Focal length
 \bar{e} of given momentum

3-axis Magnetometer measures \vec{B} in each direction
(position is adjustable)

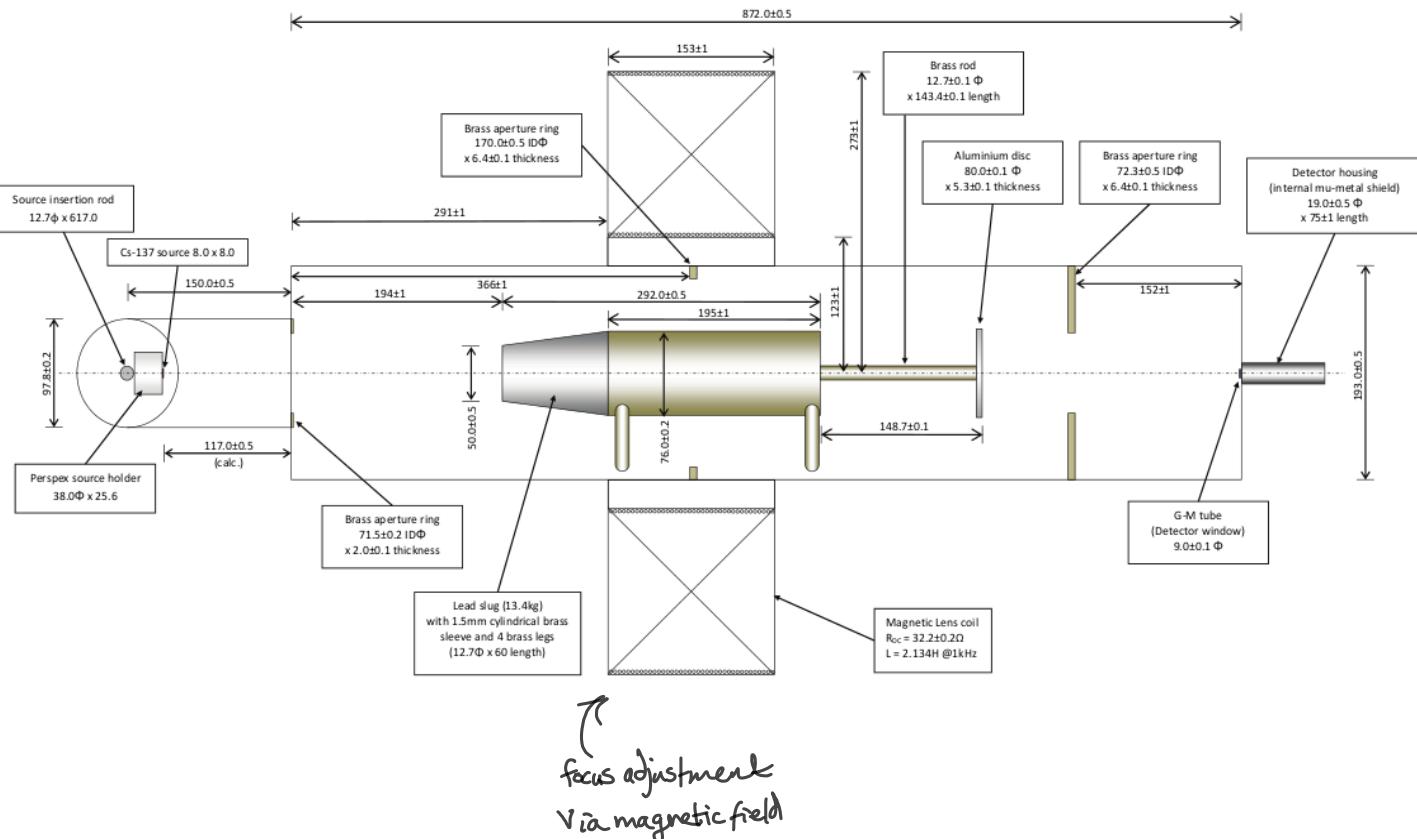


Lens coil I : adjusts focal length

05/08/2020

Schematic diagram of our Spectrometer -

Beta-Ray Spectrometer Schematic Diagram
(Side view)



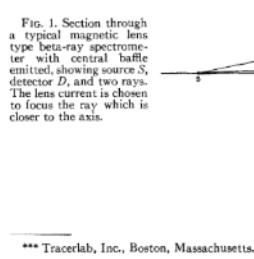
0 40 80 120 160 200 mm
Scale 1:4
All measurements in millimetres unless specified otherwise

Monash University - 3rd Year Physics Laboratory
Beta-ray Spectrometer Schematic Diagram (Side View)

Drawn by: John Golja

Date: 31/01/2012

14/08/2020



[70]

Downloaded 02 Nov 2010 to 130.194.10.86. Redistribution subject to AIP license or copyright; see http://rsi.aip.org/about/rights_and_permissions

[71] The effect of magnetic fields on the motion of charged particles

[70] $P = kI = R = Bp$ is the rigidity

determined by spectrometer's geometry.

— Script Notes —

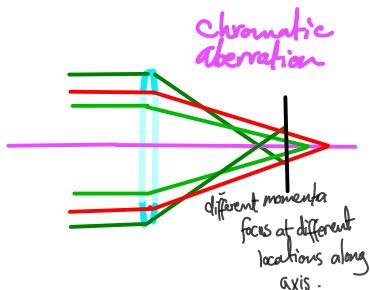
05/08/2020

The effect of the magnetic (\vec{B}) field:

- the focal length of the lens depends on the electron momentum

for a given \vec{B} :

- electrons with specific momentum are focused on the detector
- electrons carrying other momenta will focus on different points (chromatic aberration) and will be removed
- γ -rays are prevented from reaching the detector by a lead block on the axis



By varying the \vec{B} field we focus the detector to various momenta

Resolution \sim charge size and shape of buffers

* We need to divide the number of counts at each magnetic field \vec{B} (recording \vec{B})

~~typical resolution is $\sim 2\text{-}3\%$~~

Mathematical model

strength
magnetic field →
coil
Current ↓

in the absence of ferromagnetic materials: $B \propto I$

- Momentum (p)

$$p \propto B$$

Energy
kinetic ? yep

β -decay depends on Energy of transition (T)

part of T goes into "creating" the particle

(Contrast $\alpha \xrightarrow{\gamma} \gamma$ decay)

we must work with Total-mass energy (ω) of the β particle.

$$(1) \omega^2 = p^2 c^2 + \underbrace{m^2 c^4}_{e^- \text{ rest mass}}$$

→ relativistic units

$$(2) \omega = T + M c^2$$

(Since m is rest mass)

Questions

05/08/2020

$$(1) \omega^2 = p^2 c^2 + m_0^2 c^4$$

$$(2) \omega = \Gamma + m_0 c^2$$

1. By converting into relativistic units, where momentum $P = p/m_0 c$ and energy $W = w/m_0 c^2$, derive $W^2 = 1 + P^2$ from Equation (1).

$$P = p/m_0 c \quad (\text{relativistic momentum})$$

$$W = w/m_0 c^2 \quad (\text{relativistic energy})$$

$$\text{derive } W^2 = 1 + P^2$$

$$\text{we know (1)} \omega^2 = p^2 c^2 + m_0^2 c^4$$

$$\text{where } P = p/m_0 c$$

$$\therefore P m_0 c = p \Rightarrow (P m_0 c)^2 = p^2$$

Then (1) becomes:

$$\omega^2 = (P m_0 c)^2 c^2 + m_0^2 c^4$$

$$\therefore \omega^2 = P^2 m_0^2 c^4 + m_0^2 c^4$$

$$\therefore \omega^2 = (P^2 + 1) m_0^2 c^4$$

$$\text{since } W^2 = \frac{\omega^2}{m_0^2 c^4} \Leftrightarrow W^2 = P^2 + 1 \quad \square$$

later on we refer to ω as

$$\bar{W} = \sqrt{P^2 + 1}$$

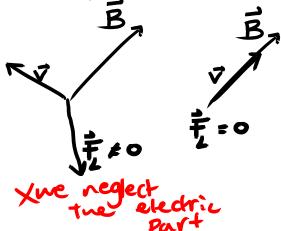
2.

What is the effect of the earth's magnetic field on the path of electrons in the spectrometer and how may this effect be nullified? Note the orientation of the spectrometer's principal axis relative to the horizontal component of the earth's field. Note: if doing the experiment online, the axis of the spectrometer is oriented (approximately) north-south. Consider how you might use the auxiliary Helmholtz coils and a 3-axis magnetic field probe (which is supplied with the experiment) in your experiment to mitigate these effects.

The magnetic field of earth will alter the trajectory of the electrons by means of the Lorentz force

Lorentz force

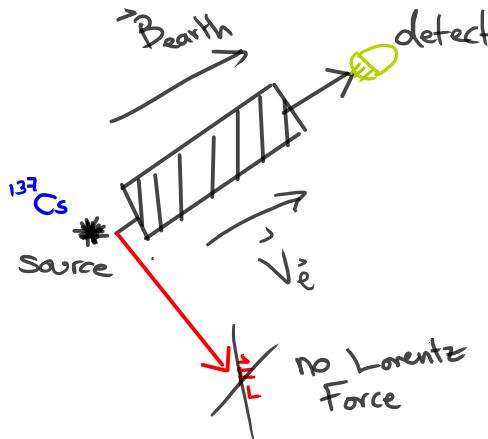
~~$$\vec{F}_{\text{total}} = e\vec{E} + e\vec{v} \times \vec{B}_{\text{total}}$$~~



effect of Magnetic field
on electrons moving at \vec{v}

$$e\vec{v} \times \vec{B} = e\vec{v} \times (\vec{B}_{\text{spec}} + \vec{B}_{\text{earth}})$$

To minimize the undesired effects on the electron beam
we align the \vec{v} and \vec{B} vectors

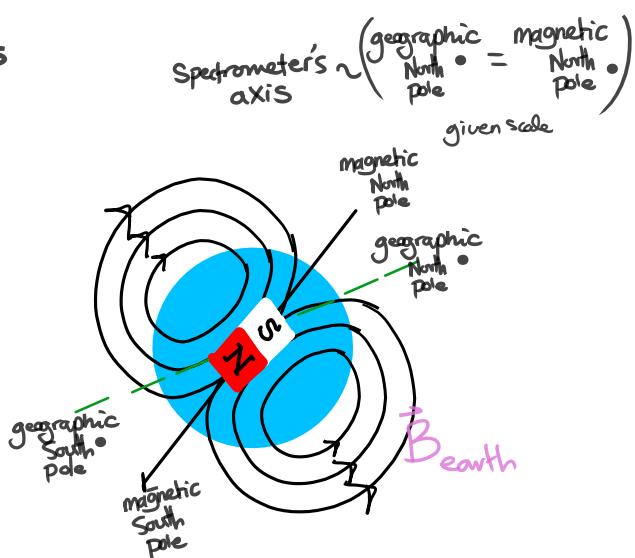


40

CLIFFORD M. WITCHER

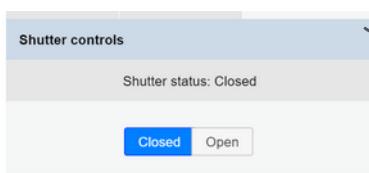
It should be noted that in such an instrument the effect of the earth's field (or other stray fields) cannot be completely neglected in the very low energy region. The axis of the instrument has been placed in the direction of the horizontal component of the earth's field to minimize any effect due to this component. There is no evidence that the operation of the instrument is appreciably affected by the vertical component of the earth's field at the lowest energies so far used (100 kev.) However, a compensating coil for the vertical component is being installed for work down to 10 to 25 kev.

Figure 8 shows the source after correction for absorption. The statistical error of 2% form of the resolution one can accurately of any beta-spectr approximations, p reasonably strong, the Ra E spectrum effect of the resolution $H_p 5100$ (1.11 MeV energy of 3.18 in on the basis of t



05/08/2020

3. How might you discover whether there is a constant background counting rate from the detector? Check whether there is such a background and correct for it in analysing your results.



* Set the shutter to **CLOSED**
This will shield the radioactive source
and so, any detected events will
be from background radiation

15/08/2020

This took me ages, I know

(Another way could be to de-focus the beam of electrons entirely)

4. You can acquire data in either constant time (time is your independent variable and the number of counts is the dependent variable) or constant count (the number of counts is the independent variable and the time is the dependent variable). You should recall from previous experiments that for a fixed time, the best estimation of the uncertainty in the number of counts, \bar{N} , is $\sqrt{\bar{N}}$. If you acquire to a fixed number of counts, what is the uncertainty in the measured time (due to the statistical nature of nuclear decay)?

Data collection

Options

1. counts N vs. Constant time intervals
2. Counts N vs. time

Note: for fixed time uncertainty in N is \sqrt{N}

What is the uncertainty in time due to statistical nature of nuclear decay?

fractional uncertainty in counts: $\frac{\sqrt{N}}{N} = \frac{1}{\sqrt{N}}$

fractional uncertainty in time: $4t = t \cdot \frac{\sqrt{N}}{N} \therefore$ fractional uncertainty is preserved.

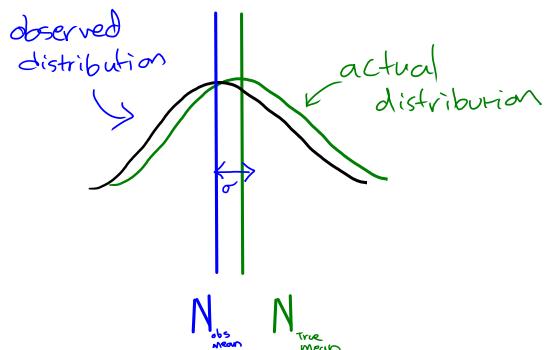
5. Consider how you should take your data. From your knowledge of the statistics of nuclear counting (as discussed in Question 4 above), is it more sensible to count for constant times at each current setting in the spectrometer coils or to collect a roughly constant number of counts? Provide a detailed answer on the choice(s) you have decided to make and explain why you made them.

Radioactivity is a poissonian process

I will run the experiment for a fixed amount of time
and record N

This N will be my observed approximation to the true N_{mean}
for this specific process

The observed distribution should
have N_{mean} within one standard
deviation of the true mean.



Hint: You may want to consider which data points of the momentum spectrum contribute the most to determining the end point energy (see the analysis section on determining the end point energy from the Kurie plot). How do you minimise the fractional uncertainty of these points? You may also want to consider how to minimise the fractional uncertainty for each data point.

05/08/2020

The experimental set-up

Apparatus

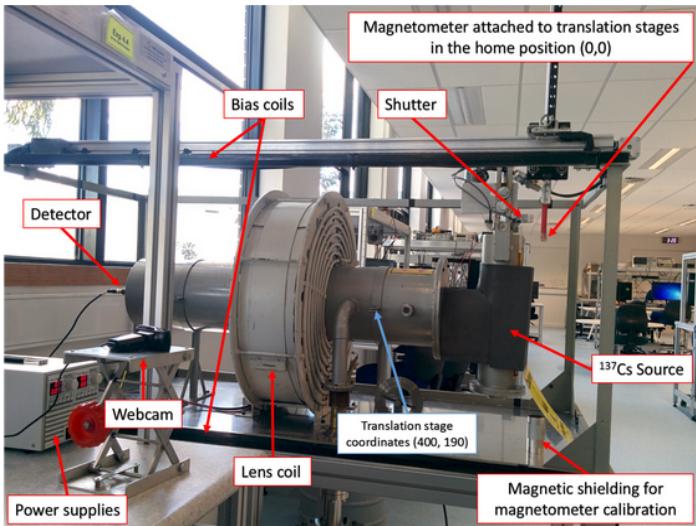
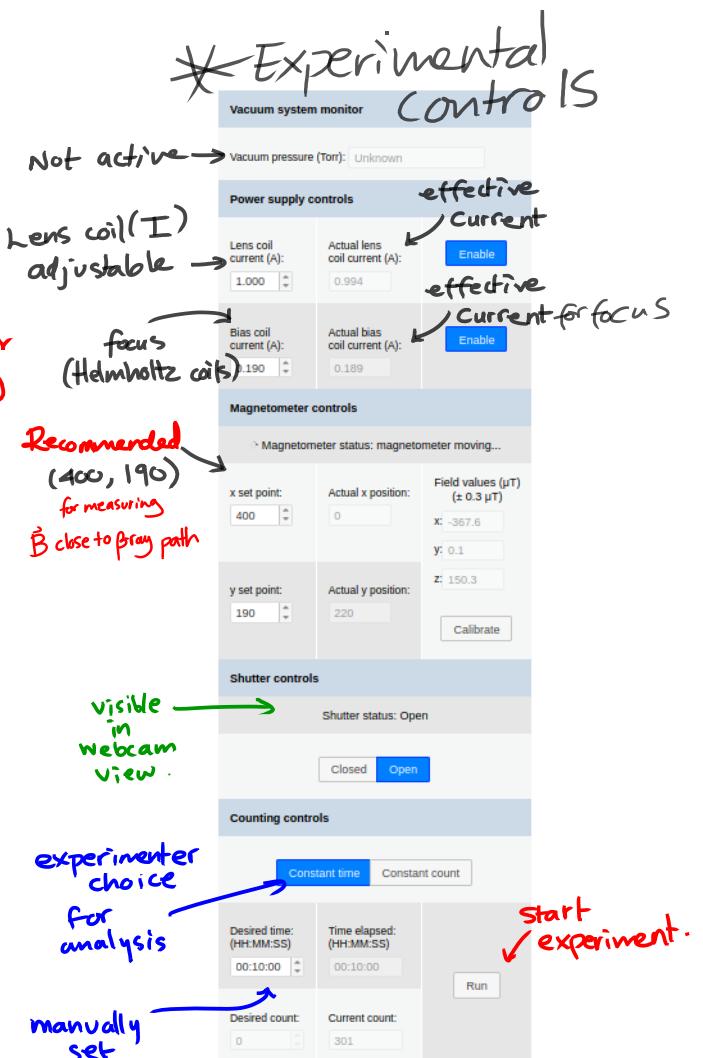
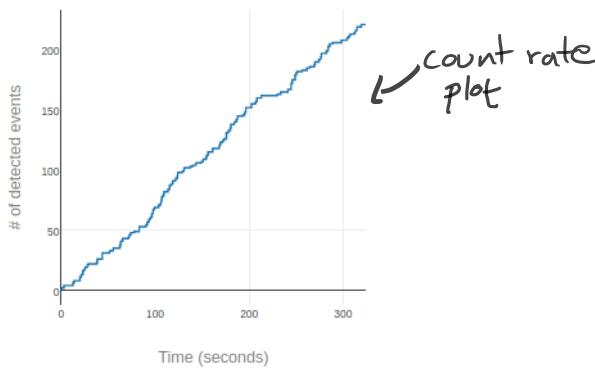
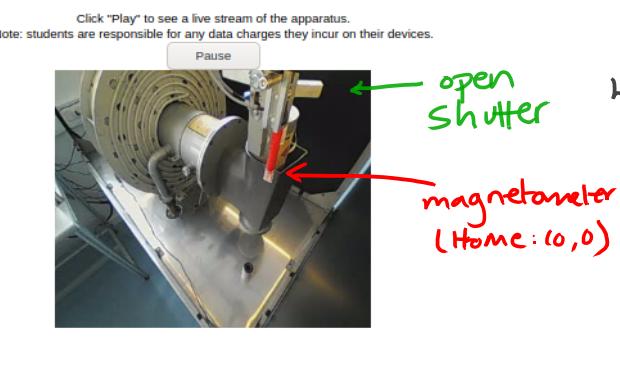


Figure 3: A labelled diagram of the beta-ray spectrometer apparatus located in the laboratory.

Control interface of the apparatus



Experimental Procedure

06/08/2020

① RUN

To familiarise myself with the control system

- ① I decided to make half integer increments to the lens coil current *

Starting with: lens coil current to 1.000A (0.9932 ± 0.0005 A)

I pressed **enable**

- ② I will not enable the bias coil current

- ③ I will not adjust the magnetometer position

- ④ I set the counting controls as **constant time**
and set the desired running time as 10 minutes ($\sim 600.000 \pm 0.015$ s)

* I repeated these steps 6 times *

Observing the data I see that when I have used

half integer steps (ie. Lens coil current = 1.500 A 2.500 A 3.500 A)

The Data table does not record my lens coil current * ← is this a control system Bug ??

- The bias coil current will be adjusted in my next set of experimental runs
- The next obvious problem is the lack of **magnetometer calibration**
this is to be corrected in my next set of experimental runs

TODO :

0. What is happening with my set up, why do I not get a current reading sometimes?

1. Should make sure magnetometer is in optimal position

2. NEED to figure out what the 3-axis magnetic field probe is doing

* 3. Eventually obtain full momentum spectra by recording # counts in a time period for various lens coil currents

figure out what size increments are adequate to resolve momentum spectrum

4. figure out if there is Background counting rate (?)

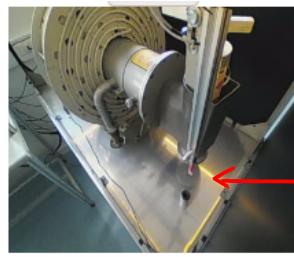
Done /
15/08/2020

Experimental Procedure

06/08/2020

Making sure magnetometer is in optimal position *

for the next run I adjusted
the coordinates of the magnetometer
 $(0,0) \rightarrow (400, 190)$



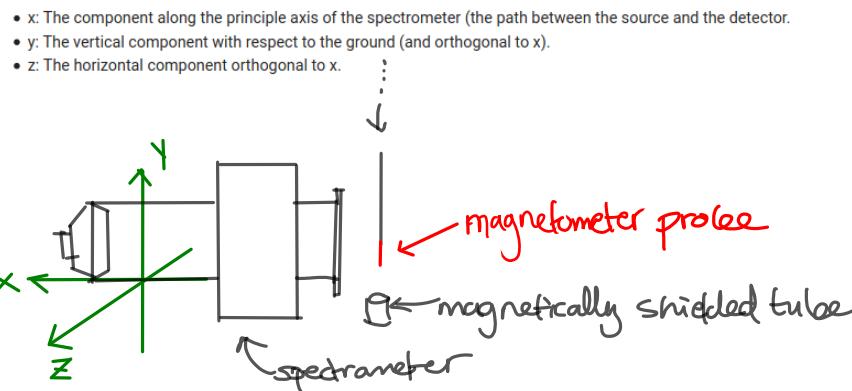
This is a screenshot of
the live camera after
the adjustment.

magnetometer probe
 $(400, 190)$

Calibrating the magnetic field probe

The magnetic field probe readings will initially be offset from the actual present field. You will likely see a significant "z" component, despite Earth's field not containing a significant component in this direction.

Note that the axes of the magnetometer probe correspond as follows:



when the probe is shielded
we consider the displayed values
as an accurate measurement
of the present magnetic field
within $0.3 \mu\text{T}$

- I proceeded to calibrate the magnetometer probe.

② RUN (Figuring out how to minimize the effect of the earth's \vec{B} field)

After adjusting the magnetometer probe

- I returned the lens coil current to the initial value

1.000 A ($0.9934 \pm 0.0005 \text{ A}$)

After increasing the bias coil current to 0.190 A ($0.1890 \pm 0.0005 \text{ A}$)
I noticed that this value makes the y-component value of the Field values
registered by the 3-axis magnetometer very close to zero.

I cannot adjust the bias coil current in a way that
decreases the other components.

Therefore I am going to try this as my first attempt
to cancel the magnetic field of the background

I press **run**

The experiment runs for 10 minutes (600.001 seconds)

Power supply controls	
Lens coil current (A): 1.000	Actual lens coil current (A): 0.993
Bias coil current (A): 0.190	Actual bias coil current (A): 0.189
Magnetometer controls	
~ Magnetometer status: magnetometer moving...	
x set point: 400	Actual x position: 0
Field values ($\pm 0.3 \mu\text{T}$) x: -367.3	y: 0.2
z: 149.8	
y set point: 190	Actual y position: 220
Calibrate	

Experimental Procedure

06/08/2020

(2) for the next coil current value:

I want to find out what is going on with my measurements when I do half-integer steps!
I won't increase the lens current (yet)

The process is similar to (1) RUN

1. Lens coil current = 1.000 A (still)

2. press **enable**

→ 3. The bias coil current is set to 0.190 A (0.1890 ± 0.0005 A)

4. experiment is not registering a Lens coil current

Power supply controls		
Lens coil current (A):	1.000	Actual lens coil current (A): 0.000
<input type="button" value="Enable"/>		
Bias coil current (A):		
0.190	Actual bias coil current (A): 0.189	<input type="button" value="Enable"/>
Magnetometer controls		
Magnetometer status: magnetometer moving...		
x set point:	400	Actual x position: 0
		Field values (μT) ($\pm 0.3 \mu\text{T}$) x: -18.7 y: 38.1 z: 2.7
y set point:	190	Actual y position: 220
		<input type="button" value="Calibrate"/>

← After observing the obtained data, I think I know what happened

— I didn't understand the control system interface —

In (1) RUN every time I changed the lens current I pressed **enable**.

In (2) RUN I didn't change the lens current but I pressed **enable**.

In the assumption that I was "activating" the experiment
I suspect that I disabled the lens current every second time I pressed **enable**

This possibly wasted ~ 90 min of data acquisition time.

*What did I learn from this?

1. How the buttons work! Brilliant!

* Since we can turn off the lens current completely
we can use our magnetometer probe to read earth's \vec{B} field
and adjust the bias coil current until we cancel earth's \vec{B} field out
completely.

(seems obvious in hindsight)

Experimental Procedure

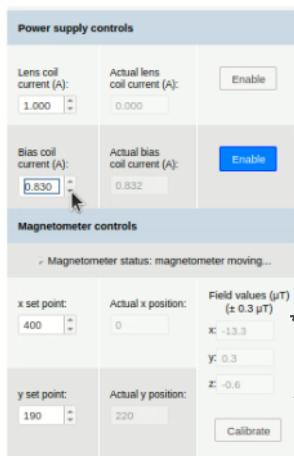
06/08/2020

③ RUN

Once again Starting with:

- ① lens coil current to 1.000 A (0.993 ± 0.0005 A)

I pressed **enable** ← To disable the lens
current briefly



- ② I increased the bias coil current until I see all displayed values as close to zero as I can get them.

∴ set value for bias coil current is: 0.830 A (This # varies slightly every run)

Given all the time "wasted", in order to get enough data
I decided to make quarter integer increments to the lens coil current }
(Hope this optimizes for both, number of data points & resolution of the spectrum)

this is earth's B field minimized

* ③ Then the process is as follows: *

- increase the lens coil current by 0.250 A
- Leave bias coil current as is, but verify that 3-axis are still close to zero every run. By briefly disabling the lens coil current.
- press run to detect N for 10 minutes
- Save plot "counts per time", record corresponding time
- Repeat until reaching lens coil current = 3.500 A

TODO:

- 0. what is happening with my set up, why do I not get a current reading? sometimes?
- 1. Should make sure magnetometer is in optimal position
- 2. NEED to figure out what the 3 axis magnetic field probe is doing

* ③ Eventually obtain full momentum spectra by recording # counts in a time period for various lens coil currents

figure out what size increments are adequate to resolve momentum spectrum ✓

4. figure out if there is Background counting rate

* A possible improvement to the Laboratory script:

The "experiment Procedure" tab should be moved to be before "Reading the CSV file" tab. This might just be my fault but I didn't understand how important the calibration was just from theory, it took me some time to learn this ⇒ wasted time!

Analysis

06/08/2020

(3) RUN (I will only consider this data in my analysis)

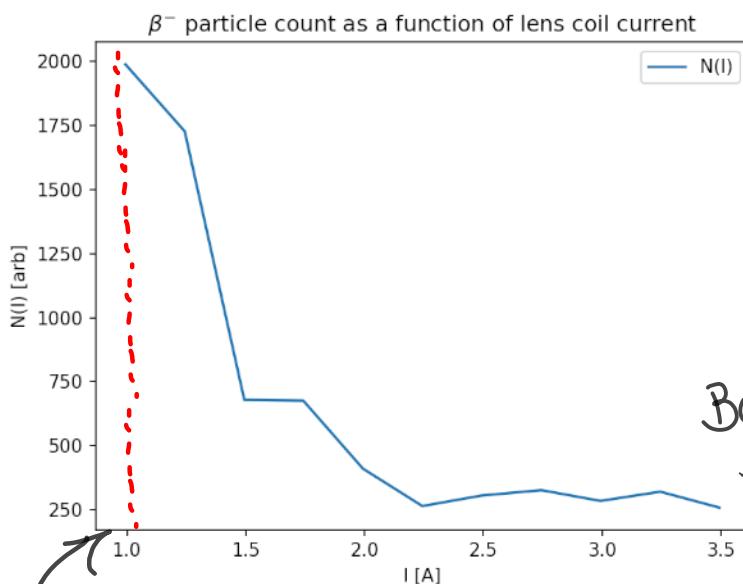
Valid Data

•	Duration (s)	Count	Lens coil current (A)	u(Lens coil current) (A)	Bias coil current (A)	u(Bias coil current) (A)
1	600.005	1988	0.99343	0.0005	0.83093	0.0005
2	600.005	1727	1.24269	0.0005	0.83106	0.0005
3	600.008	679	1.49371	0.0005	0.83091	0.0005
4	600.007	676	1.74207	0.0005	0.83111	0.0005
5	600.004	411	1.99288	0.0005	0.83104	0.0005
6	600.004	265	2.24291	0.0005	0.83106	0.0005
7	600.005	306	2.49286	0.0005	0.83096	0.0005
8	600	327	2.74431	0.0005	0.83099	0.0005
9	600.009	285	2.99308	0.0005	0.83097	0.0005
10	600.002	321	3.24456	0.0005	0.83102	0.0005
11	600.006	259	3.49335	0.0005	0.831	0.0005

•	Magnetometer x position	Magnetometer y position	Magnetometer x-axis field	u(Magnetometer)	Magnetometer y-axis field	u(Magnetometer)	Magnetometer z-axis field strength (μ T)	u(Magnetometer)
1	400	190	-362.61856	0.3	-35.86697	0.3	148.2517	0.3
2	400	190	-449.99378	0.3	-44.91145	0.3	185.44758	0.3
3	400	190	-537.9354	0.3	-53.9059	0.3	223.45595	0.3
4	400	190	-625.54941	0.3	-62.67791	0.3	261.6502	0.3
5	400	190	-714.12209	0.3	-71.35809	0.3	300.39517	0.3
6	400	190	-803.34455	0.3	-80.01129	0.3	338.86251	0.3
7	400	190	-891.95172	0.3	-88.22483	0.3	377.46732	0.3
8	400	190	-981.25639	0.3	-96.53003	0.3	416.37661	0.3
9	400	190	-1069.90218	0.3	-104.74846	0.3	454.82582	0.3
10	400	190	-1159.91273	0.3	-112.64221	0.3	494.20548	0.3
11	400	190	-1249.45357	0.3	-120.52033	0.3	532.30743	0.3

$N(I)$ vs I_{lens}

14/08/2020



Booked another lab session ~~16/08/2020~~
Bad Luck!

Stayed stuck for 20 min

Magnetometer controls

Magnetometer status: Calibrating...please wait

x set point:	Actual x position:	Field values (μ T) ($\pm 0.3 \mu$ T)
400	0	x: -18.4
		y: 48.3
		z: 4.9
y set point:	Actual y position:	Calibrate
190	220	

I don't think that I have enough current values,
+ Starting at 1A was probably a bad choice!

Booked another lab session ~~18/08/2020~~

~~BUGS BUGS BUGS~~

Updating my understanding

14/08/2020

Theory (Round II)

[9] Siegbahn

An electron (charge e , velocity \vec{v}) moving in a homogeneous magnetic field \vec{B} in a plane perpendicular to the lines of force, has radius of curvature r , the kinetic energy is:

$$B_e \vec{J} = \frac{mv^2}{r} = T$$

where $n = \frac{m_0}{\sqrt{1 - (V/c)^2}}$ (m_0 is rest mass)

∴ the momentum is expressed as a B_p -value

$$p = mv = eB_p$$

This relation is the reason why we use momentum coordinates as opposed to Energy

Since $E_{kinetic} = \frac{p^2}{2m}$ plotting the distribution as an energy distribution $n(E)E$. Then the measures relate as $dE = \frac{p}{m} dp$.

The corresponding intensities in the spectrum plot ($n(E)$ vs E) are multiplied by $\frac{m}{p}$.

The momentum distribution of the electrons in a magnetic spectrometer is $N(p)dp$

For this experiment, we use a spectrometer of fixed geometry ~~*~~ and variable \vec{B}

∴ The resolution: $R = \frac{\Delta(B_p)}{B_p} = \frac{\Delta p}{p} \sim 2-3\%$
(uncertainty)

where Δp is a measure of the accepted momentum band

- The rigidity -

[11] The effect of magnetic fields on the motion of charged particles

A measure of the momentum of the particle

Higher momentum \Rightarrow higher "resistance" to deflection by a magnetic field.

$R = B_p = \frac{p}{e}$ where
 B is magnetic field
 p is radius of the particle due to field
 p is momentum
 e is charge

Quade, Halliday (1947)
[10] The rigidity: $P = kI$

where k is a constant determined by the spectrometer's geometry ~~*~~

The current relates to the magnetic field as

$$I = \frac{eP}{K} B \quad \text{where:} \quad \begin{aligned} B &\text{ is magnetic field} \\ p &\text{ is radius of electron due to } B \\ e &\text{ is charge of electron} \end{aligned}$$

Theory

14/08/2020

[8] Relativistic Momentum and Energy for β^- decay

In Relativistic units

- electron's energy -

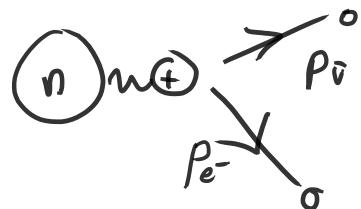
$$\bar{\omega} = \bar{T} + L = \sqrt{\bar{p}^2 + 1}$$

(in SI = $\omega = T + mc^2 = \sqrt{p_c^2 + m_0 c^2}$)

The rest energy of an electron is 0.511 MeV

$$\therefore \text{Decay energy} : \omega_0 \geq 0.511 \text{ MeV}$$

The antineutrino can't be detected: $E_\nu > c p_\nu$



$$\text{The electron has kinetic energy } \bar{T} = \bar{\omega} - 1 = \sqrt{\bar{p}^2 + 1} - 1$$

Since the isobar is very massive compared to the other particles

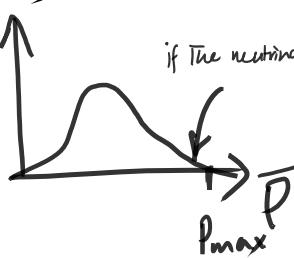
[Total decay energy:] \leftarrow momentum is conserved
 $\therefore \omega_0 \approx T + E_\nu$

— The number of electrons with a particular momentum —

$$n(\bar{p}) = K_1 F(z, \omega) \bar{p}^2 (\omega_0 - \omega)^2 S_n(\omega)$$

ω_0 is energy of transition

Notice that:
 $n(\bar{p}) = 0 \left\{ \begin{array}{l} \text{if } \bar{p} = 0 \\ \text{if } \bar{p} = p_{\max} \text{ if } \bar{p}_{\max} \neq 0 : (\omega_0 - \omega)^2 = 0 \text{ then } \omega_0 = \omega \end{array} \right\}$



[we want to measure \bar{p}_{\max} to calculate $T_{\max} = \sqrt{\bar{p}_{\max}^2 + 1} - 1$ which will let us determine ω_0]

Theory Analysis

-Calibration of energy and momentum range-

[o] The theories of beta-decay depend on the energy of the transition, but because part of the energy of the transition has to go into creating the beta particle (contrast alpha and gamma decay), we must work with the total mass-energy, w , rather than kinetic energy, T , of the beta. Recall that:

$$w^2 = p^2 c^2 + m_0^2 c^4, \quad (1)$$

where p is the momentum (of the beta particle), c is the speed of light, $m_0 c^2$ is the rest mass-energy of the electron and

$$w = T + m_0 c^2. \quad (2)$$

we are told that $T_k = 624.21 \text{ keV}$

(is the conversion energy spike)

we know $\bar{T} = \bar{w} - 1 = \sqrt{\bar{p}^2 + 1} - 1$ is the kinetic energy

$$\Rightarrow \bar{T}_k = \sqrt{\bar{p}_k^2 + 1} - 1$$

$$\therefore \bar{p}_k = \sqrt{(\bar{T}_k + 1)^2 - 1}$$

Then our first plot: $N(I)$ should look like →
momentum is expressed as:

$$* P = eB\phi \quad [9]$$

$$* P = K I \quad [10]$$

Then $P_k = K I_k$

To find the constant K we use $\sqrt{(\bar{T}_k + 1)^2 - 1} = K I_k$

$$\therefore \frac{\sqrt{(\bar{T}_k + 1)^2 - 1}}{I_k^2} = K \quad \text{WOWAH!}$$

after we have found our constant of proportionality (K) we can verify

The momentum spectrum

[o]

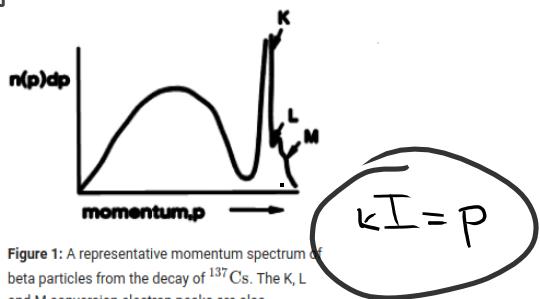


Figure 1: A representative momentum spectrum of beta particles from the decay of ^{137}Cs . The K, L and M conversion electron peaks are also depicted on the right.

Theory / Analysis

14/08/2020

How long do we want to count events for?

To determine the time interval used to count events, we need to use our uncertainty

$$\frac{u(n)}{n} = \frac{\sqrt{n}}{n} = \frac{1}{\sqrt{n}}$$

if we want our uncertainty to be

$$u(n) \leq 5\%$$

Then

$$\frac{1}{\sqrt{n}} \leq \frac{1}{20} \Rightarrow n = 20^2 = 400$$

first estimate for time interval if our first attempt at the experiment had $t_1 \approx 600$ s

$$\text{then } \frac{n}{t} = \frac{2000}{t_1} = \frac{400}{t_2} \Rightarrow \frac{10}{3} = \frac{2}{t_2} \Rightarrow t_2 = 360 \text{ s}$$

In our first attempt at the experiment we observed that the count decreased as the lens current increased. So this should be in our mind as we run the experiment in order for our count to continue having an uncertainty of $\leq 5\%$.

Experimental Procedure (Round II)

18/08/2020

Calibrate magnetic field probe

* the coordinates of the magnetometer are set to: (400, 190)

These values →
are an accurate measurement of
the field within $0.3 \mu\text{T}$

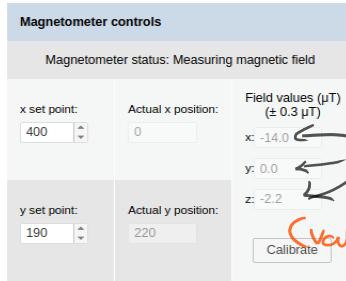


Nulling the Earth's field

lens coil current to 0A ($0 \pm 0.0005\text{A}$)

press **enable** ← To disable the lens
current briefly

AFTER earth's \vec{B} field
is "minimized"



ENABLE bias coil current
and increase the bias coil current until
all displayed Field values are
close to zero

bias coil current is: 0.720 A ($0.716 \pm 0.0005\text{A}$)

- Set counting controls to **constant time**
• set the count running-line to $360\text{s} (\pm 0.015\text{s})$

Background rate Count

- Set shutter to **closed**

- proceed to count the background radiation
for 4 runs (360s each run),
record each count run.

eyeball estimate
↙

Background counts	
✓	188
✓	158
✓	147
✓	192

(1) β^- particle count RUN

-Calibration of energy and momentum range-

The process is as follows:

1. Set the shutter status to **open**

2. **ENABLE** lens coil current.

we start with lens coil current set to 0A ($0.0000 \pm 0.0005\text{A}$)

* lens coil current increments will be 0.1A ($0.0940 \pm 0.0005\text{A}$) *

* Leave bias coil current as is, but verify that 3-axis are still close to zero every run. By briefly disabling the lens coil current prior to running count.*

- press **run** to detect current dependent count ($n(I)$)

- increase current by 0.1A

- Repeat until reaching **MAX** lens coil current = 3.6000A ($3.5245 \pm 0.0005\text{A}$)

Keep in mind:

1 what size increments are adequate to resolve momentum spectrum

Initial increments 0.1A

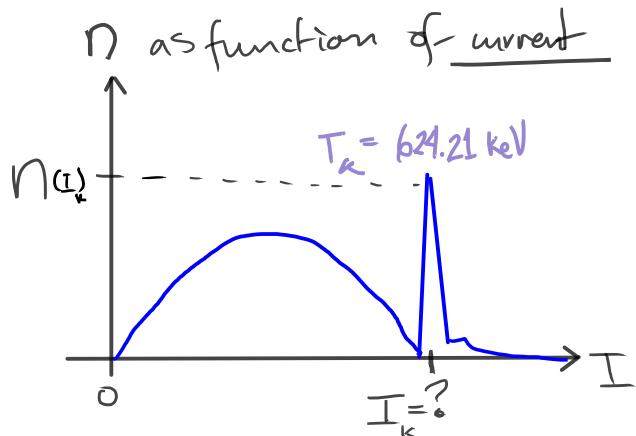
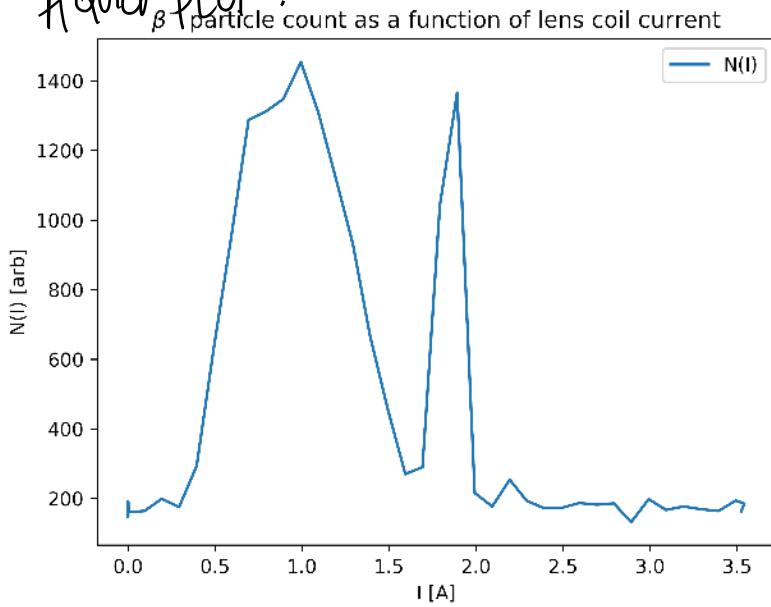
Run at 11:01am was set to incorrect current 0.0031A

2 do we need to modify the time interval?

Analysis (Round II)

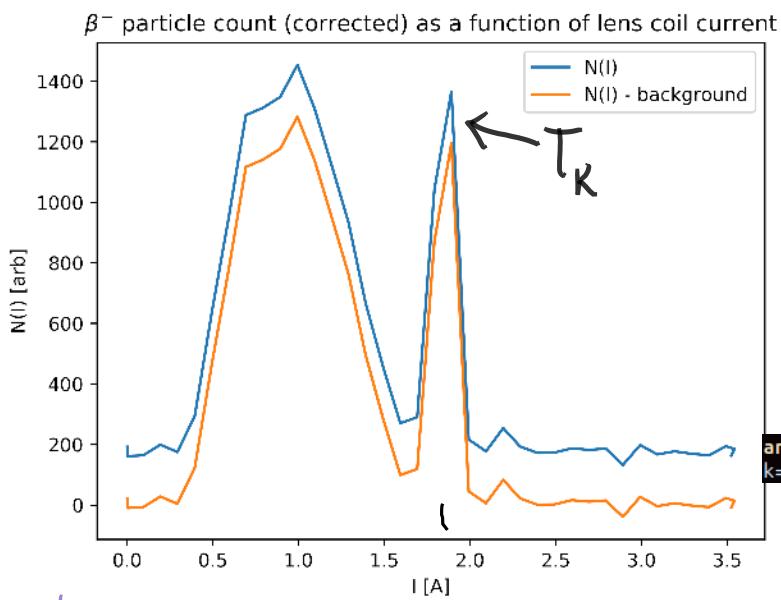
18/08/2020

A quick plot:



Since $T_K = 624.21 \text{ keV}$

$$I_K = 1.8916 \pm 0.0005$$

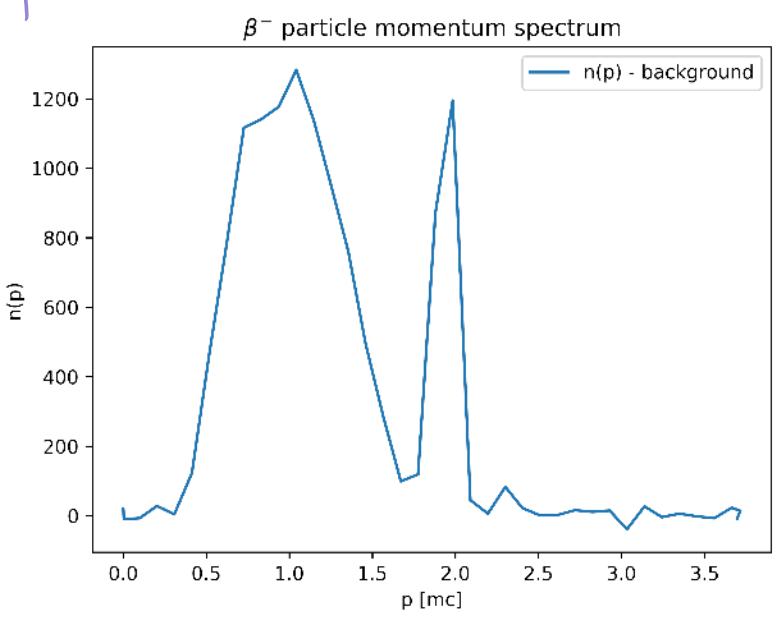


and we know $\bar{p}_K = k I_K = \sqrt{(T_K + L)^2 - 1}$
 $\therefore \frac{\sqrt{(T_K + L)^2 - 1}}{I_K^2} = k$

```
# Finding constant of proportionality in p = kI
# calibration peak (K) index of K peak is i=20
T_K = 624.21 * keV / rel_energy_unit
k = np.sqrt((T_K + 1)**2 - 1) / lens_current[20]
print(f" {k=}")
```

```
ana@ana-XPS-13-9343:~/Documents/uni/PHS3000$ python3 betarays.py
k=1.048711908196117
```

Plot: The momentum spectrum



```
# The momentum spectrum
lens_current = np.array(lens_current)
p_rel = k * lens_current
# print(p_rel)

plt.plot(
    p_rel, count - avg_background_count, marker="None",
    linestyle="--",
    label="n(p) - background"
)
plt.title(r"\beta^- particle momentum spectrum")
plt.xlabel("p [MeV/c]")
plt.ylabel("n(p)")
plt.legend()
spa.savefig('count_vs_momentum_no_background.png')
plt.show()
```

$$\bar{p} = k I$$

what about uncertainty?

Analysis (Round II)

18/08/2020

— Uncertainty in $\bar{P} = kI$

we calculated $\bar{P}_k = kI_k$

$$\text{where } k = \frac{\sqrt{(\bar{T}_k + 1)^2 - 1}}{I_k^2}$$

① The uncertainty in k :

$$u(k) = k \sqrt{\left(\frac{u(T_k)}{T_k}\right)^2 + \left(\frac{u(I_k)}{I_k}\right)^2}$$

where $u(T_k) = 0$ and $u(I_k) = 0.0005A$

$$\therefore u(k) = k \sqrt{\left(\frac{u(I_k)}{I_k}\right)^2} = k \frac{0.0005A}{1.89164} = 0.0002$$

② The uncertainty in \bar{P} :

$$u(\bar{P}) = p \sqrt{\left(\frac{u(k)}{k}\right)^2 + \left(\frac{u(I)}{I}\right)^2}$$

```
# The momentum spectrum
lens_current = np.array(lens_current)
p_rel = k * lens_current
u_p_rel = p_rel * np.sqrt((u_k / k)**2 + (0.0005 / lens_current)**2)
print(f"absolute uncertainty u(p_rel):\n {u_p_rel}")
print(f"fractional uncertainty u(p_rel) / p_rel:\n {(u_p_rel / p_rel)}")
```

```
absolute uncertainty u(p_rel):
[0.00052436 0.00052436 0.000525    0.00052709 0.00053061 0.00053563
 0.00054192 0.00054948 0.0005584   0.00056843 0.00057983 0.00059226
 0.0006057  0.00061988 0.000635    0.00065088 0.00066813 0.00068566
 0.0007036  0.00072235 0.00074155  0.00076167 0.00078212 0.00080299
 0.00082383 0.00084554 0.00086743  0.00088994 0.00091269 0.00093492
 0.00095797 0.00098148 0.00100485  0.00102899 0.00105294 0.00107704
 0.00110124 0.00111333 0.00110882]
fractional uncertainty u(p_rel) / p_rel:
[2.786364e+01 1.46833251e-01 5.32672848e-03 2.59748527e-03
 1.72679689e-03 1.29521414e-03 1.04668075e-03 8.84174670e-04
 7.68757889e-04 6.84655465e-04 6.19267215e-04 5.68515632e-04
 5.28049702e-04 4.95603205e-04 4.68633808e-04 4.46161339e-04
 4.26516001e-04 4.10232132e-04 3.96417943e-04 3.84305110e-04
 3.73812227e-04 3.64435428e-04 3.56244631e-04 3.49012222e-04
 3.42707174e-04 3.36939560e-04 3.31813574e-04 3.27140414e-04
 3.22941074e-04 3.19266567e-04 3.15839663e-04 3.12689868e-04
 3.09857827e-04 3.07204115e-04 3.04809214e-04 3.02609052e-04
 3.00586582e-04 2.99639251e-04 2.99988514e-04]
ana@ana-XPS-13-9343:~/Documents/uni/PHS3000$
```

Vertical uncertainty

we calculated:

$$n_{\text{corrected}} = n - n_{\text{background}}$$

$$\therefore u(n_{\text{corrected}}) = \sqrt{u(n)^2 + u(n_{\text{background}})^2}$$

where $u(n) = \sqrt{n}$ and ...

$$\therefore u(n_{\text{corrected}}) = \sqrt{n + u(n_{\text{background}})^2}$$

$u(n_{\text{background}})$ is calculated by considering 4x6 minutes experimental runs as a 24 minutes block. The method is as follows:

1. we sum the 4 background radiation run counts
2. then we calculate $\sqrt{n_{\text{background}}}$
3. we divide $\sqrt{n_{\text{background}}}$ by the number of experimental runs where we counted the background radiation events (for us the number is 4)

```
# uncertainty in the corrected count
corrected_count = count - avg_background_count
u_corrected_count = np.sqrt(count + u_avg_background_count**2)
```

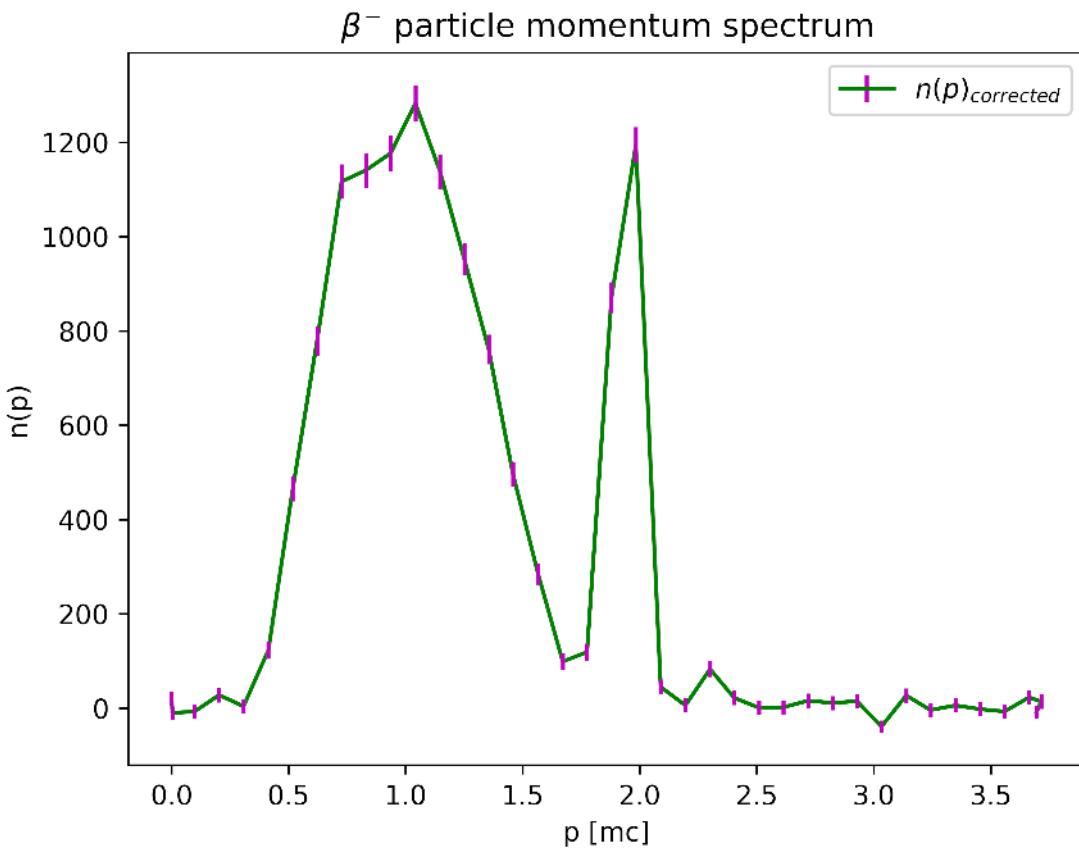
```
background_count = []
# correcting our data by removing avg background count
for row in background_count_data:
    background_count.append(row[5])
avg_background_count = np.mean(background_count)
# print(f"We want to subtract this background count from our data (avg_background_count={})")
# calculating fractional uncertainty in total background count (delta_t = 24 min)
total_background = np.sum(background_count)
u_avg_background_count = np.sqrt(total_background) / 4
```

The variable $u_avg_background_count$ is the expected uncertainty for a 6 minute background radiation events counting run.

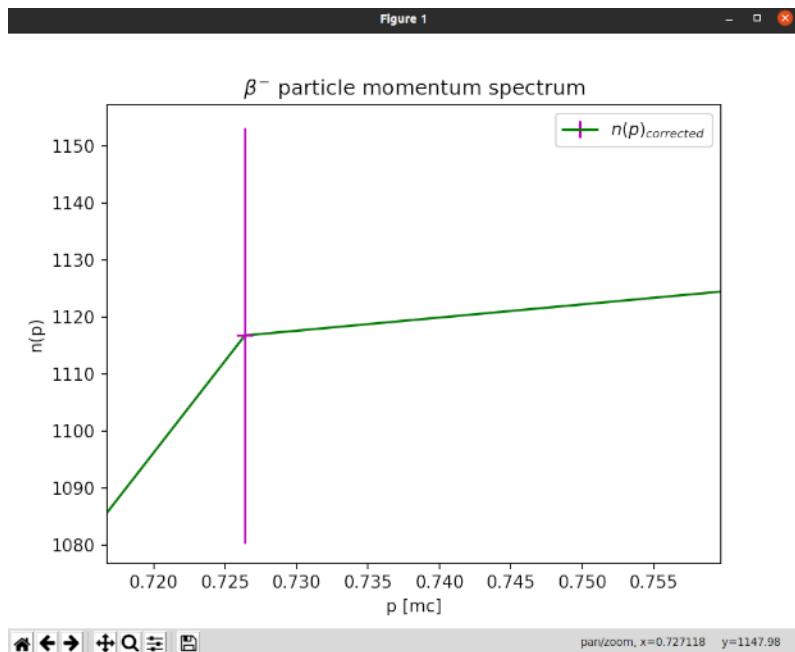
Analysis (Round II)

Plot with uncertainties -

18/08/2020



our uncertainty in P is not visible at this scale



This is the zoomed version
using matplotlib UI

CODE

```
plt.errorbar(  
    p_rel, corrected_count, xerr=u_p_rel, yerr=u_corrected_count,  
    marker="None", ecolor="m", label=r"$n(p)_{\text{corrected}}$", color="g", barsabove=True  
)  
  
plt.title(r"$\beta^-$ particle momentum spectrum")  
plt.xlabel("p [mc]")  
plt.ylabel("n(p)")  
plt.legend()  
plt.savefig('count_vs_momentum_no_background_error.png')  
plt.show()
```

Analysis (Round II)

15/08/2020

Kurie Plot

$$n(p)dp = K_1 F(z, \omega) p^2 (\omega_0 - \omega)^3 S_n(\omega) dp \quad (3)$$

p is momentum of electron

$F(z, \omega)$ is Fermi function \rightsquigarrow accounts for Coulomb attraction: β and A/Z

$Z \rightarrow$ charge of A/Z

$S_n(\omega)$ is shape factor $\text{constant if decay is allowed}$ $\therefore e^- \text{ carries zero angular momentum}$ else: $\underline{\text{forbidden!}}$

ω_0 is decay energy

$\therefore \omega_0 - \omega$ is energy of neutrino

for our analysis we will use a modified Fermi funct.

$$F(z, \omega) \rightsquigarrow G = \frac{p F(z=55, \omega)}{\omega}$$

The tabulated data is:

	A	B	C	D	E	F	G
1	Modified Fermi function, G, for Z=55						
2	Note: In this table, "p" is in relativistic units, i.e. unit momentum is $m_0 c^2$						
3							
4	p	G					
5	0.0	6.591					
6	0.1	6.582					
7	0.2	6.552					
8	0.3	6.506					
9	0.4	6.448					
10	0.5	6.387					
11	0.6	6.329					
12	0.7	6.275					
13	0.8	6.224					
14	0.9	6.177					
15	1.0	6.132					
16	1.2	6.046					
17	1.4	5.964					
18	1.6	5.886					
19	1.8	5.812					
20	2.0	5.742					
21	2.2	5.675					
22	2.4	5.612					
23	2.6	5.553					
24	2.8	5.496					
25	3.0	5.443					

this is a typo relativistic momentum has units $m c$

We interpolate this data in order to find G for our p's

```
from scipy.interpolate import interp1d
interpolated_fermi_data = interp1d(fermi_data[:, 0], fermi_data[:, 1],
kind='cubic')
```

We named our function

Interpolated_fermi(p)

UNITS

Our momentum data

comes from the relation $P = K I$

where $[I] = \text{Amps}$, which is an SI unit

In order to use Interpolated_fermi(p)

we must convert the units of p from SI into relativistic units

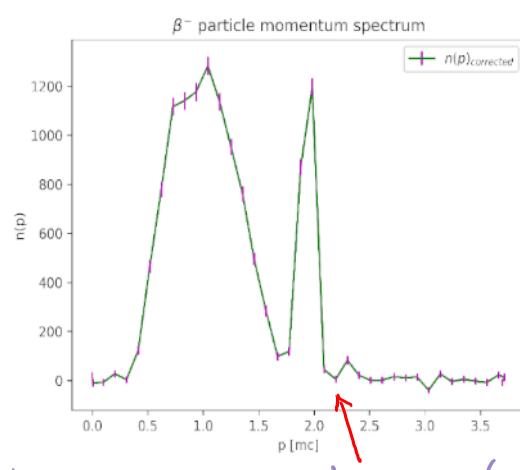
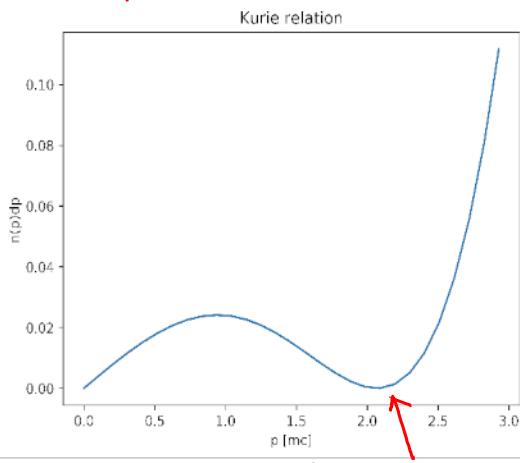
we do this by defining a variable

```
rel_energy_unit = mass_e * c**2 # to convert SI into relativistic or viceversa
```

Analysis (Round II)

19/08/2020

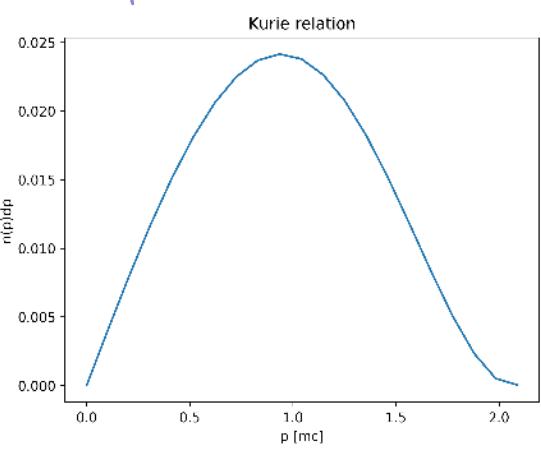
- Comparing our plots -



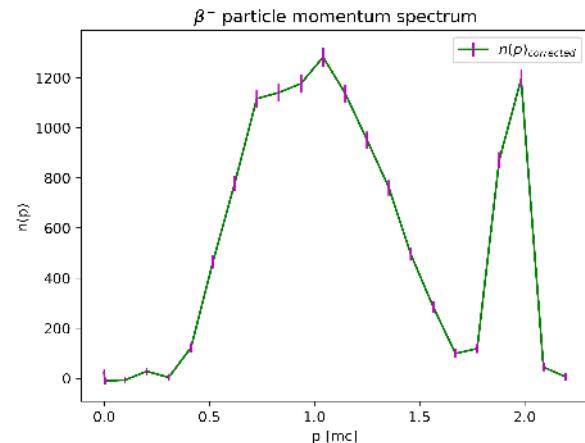
Theoretical $n(p)/dp$ minimum when $p=p_{max}$. measured minimum for p_{max}

We know that the Kurie relation doesn't take into account the k -peak
in fact the data past the minimum at p_{max} is non physical

∴ we will
exclude these
non-physical points



V.S.



$\bar{p}[:22]$ is our
theoretical sliced range.

```
##### KURIE PLOT #####
dp_rel = p_rel[1:-p_rel[0]]  
  
# getting interpolated
fermi_data = spa.betamay.modified_fermi.function_data
interpolated_fermi = interp1d(fermi_data[:,0], fermi_data[:,1], kind='cubic')  
  
##### THEORETICAL #####
# Desintegration energy is 137 disintegrated by beta minus emission to the ground state of Ba-137 (5.6 %)
w_0_rel = 174.5
w_B_rel = w_0_rel / rel_energy_unit
p_0_rel = np.sqrt(w_0_rel**2 - 1) / (mass_e * c)
# print(p_0_rel)  
  
# defining the theoretical count (Kuriefunction)
K_1 = 1 # ?
S_0 = 1
def n(p_rel):
    w_rel = np.sqrt(p_rel**2 + 1) # relativistic energy units
    n = K_1 * S_0 * (w_rel * interpolated_fermi(p_rel) / p_rel) * p_rel**2 * (w_0_rel - w_rel)**2
    return n, w_rel
n_p_rel, w_rel = n(p_rel[:22]) # call and unpack n(p)  
  
# equation (3) in script
N = n_p_rel * dp_rel  
  
plt.figure()
plt.plot(p_rel[:22], N, marker="None",
          linestyle="--")
plt.title("Kurie relation")
plt.xlabel("p [mc]")
plt.ylabel("n(p)/dp")
spa.savefig("Kurie_plot.png")
plt.show()  
##### THEORETICAL #####

```

```
#####
plt.figure()
plt.errorbar(p_rel[:23], corrected_count[:23], xerr=u_p_rel[:23], yerr=u_corrected_count[:23],
             marker="None", ecolor="m", label=r"$n(p)_{corrected}$", color="g", barsabove=True)
)  
  
plt.title(r"\beta^- particle momentum spectrum")
plt.xlabel("p [mc]")
plt.ylabel("n(p)")
plt.legend()
spa.savefig('count_vs_momentum_no_background_error.png')
plt.show()  
#####
EXPERIMENTAL #####

```

$\bar{p}[:23]$ is our corresponding experimental sliced range.

Analysis (Round II)

16/08/2020

Kine linearised

$$n(\bar{p}) = K_1 F(z=55, \bar{\omega}) \bar{p}^2 (\omega_0 - \bar{\omega})^2 S_n(\bar{\omega})$$

written in terms of the interpolated fermi function

$$\left(\frac{n(\bar{p})}{\bar{p}\bar{\omega}G}\right)^{\frac{1}{2}} = K_2 (S_n(\bar{\omega}))^{\frac{1}{2}} (\omega_0 - \bar{\omega})$$

in the case of the allowed transition ($S_0(\bar{\omega}) = 1$)

$$\left(\frac{n(\bar{p})}{\bar{p}\bar{\omega}G}\right)^{\frac{1}{2}} = K_2 (\omega_0 - \bar{\omega})$$

19/08/2020

Known Uncertainties

- $u(n) = \sqrt{n}$
- $u(p) = p \sqrt{\left(\frac{u(\omega)}{k}\right)^2 + \left(\frac{u(E)}{T}\right)^2}$

$$\left\{ \begin{array}{l} \text{Then} \\ \frac{u(\bar{p}^2)}{\bar{p}} = 2 \sqrt{\left(\frac{u(\omega)}{k}\right)^2 + \left(\frac{u(E)}{T}\right)^2} \\ \therefore u(\bar{p}^2) = 2 \sqrt{\left(\frac{u(\omega)}{k}\right)^2 + \left(\frac{u(E)}{T}\right)^2} \bar{p} \end{array} \right.$$

since: $\bar{\omega} = \sqrt{\bar{p}+1}$ $\rightsquigarrow u(\bar{\omega}) = \frac{1}{2} \sqrt{(u(\bar{p}^2))^2} = \sqrt{\sqrt{\left(\frac{u(\omega)}{k}\right)^2 + \left(\frac{u(E)}{T}\right)^2} \bar{p}}$

$$\therefore u(\bar{\omega}) = u(\bar{p})$$

Since $G = \frac{\bar{p}F}{\bar{\omega}}$

$$u(G) = \left(\sqrt{\left(\frac{u(p)}{\bar{p}}\right)^2 + \left(\frac{u(\omega)}{\bar{\omega}}\right)^2} \right) G$$

uncertainty in interpolated fermi
 $u_{\text{interpolated_fermi}} = np.sqrt((u_p_{\text{rel}}[:21] / p_{\text{rel}}[:21])^{**2} + (u_x / x)^{**2}) * \text{interpolated_fermi}(p_{\text{rel}}[:21])$

- LINEARFIT uncertainty -

* let $y = \left(\frac{n(p)}{p\bar{\omega}G}\right)^{\frac{1}{2}}$ *

Then $\frac{u(y)}{y} = \frac{1}{2} \sqrt{\left(\frac{u(n)}{n}\right)^2 + \left(\frac{u(p)}{p}\right)^2 + \left(\frac{u(\bar{\omega})}{\bar{\omega}}\right)^2 + \left(\frac{u(G)}{G}\right)^2}$ where $\left(\frac{u(\bar{p})}{\bar{p}}\right)^2 + \left(\frac{u(\bar{\omega})}{\bar{\omega}}\right)^2 = 2 \left(\frac{u(\bar{p})}{\bar{p}}\right)^2$

$$\therefore \frac{u(y)}{y} = \frac{1}{2} \sqrt{\left(\frac{u(n)}{n}\right)^2 + 2 \left(\frac{u(p)}{p}\right)^2 + \left(\frac{u(G)}{G}\right)^2}$$

We actually have a bit of a problem . . .

Analysis (Round II)

19/08/2020

Kurie Linearised

$$\left(\frac{n}{pwG}\right)^{\frac{1}{2}} = K_2 (\omega_0 - \bar{\omega}) \Leftrightarrow y = K_2 (\omega_0 - X)$$

- While setting our fit, we obtained this error -

ValueError: The model function generated NaN values and the fit aborted! Please check your model function and/or set boundaries on parameters where applicable. In cases like this, using "nan_policy='omit'" will probably not work.

Debugging Hypothesis: negative numbers causing nan values come from our correction to the count

```
ana@ana-XPS-13-9343:~/Documents/uni/PHS3000$ python3 betarays.py
corrected_count[:23]=array([ 20.75, -10.25, -7.25, 27.75, 3.75, 122.75, 464.75,
    778.75, 1116.75, 1140.75, 1146.75, 1182.75, 1137.75, 951.75, * interpolated_fermi(p_rel[:23])
    760.75, 495.75, 283.75, 98.75, 118.75, 870.75, 1194.75, int[:23].clip(min=0))**2
    44.75, 5.75])
```

```
ana@ana-XPS-13-9343:~/Documents/uni/PHS3000$ python3 betarays.py + interpolated_fermi(p_rel[:23])
betarays.py:160: RuntimeWarning: invalid value encountered in sqrt
y = np.sqrt(corrected_count[:23] / (p_rel[:23] * x * interpolated_fermi(p_rel[:23])))
array([409.01493712, * nan, nan, 4.52307962,
    1.3394485, 6.52641981, 11.18086471, 12.97994659,
    14.09505136, 13.0479514, 12.20325659, 11.80983331,
    10.36396901, 8.88089041, 7.46523865, 5.6870446,
    4.06488621, 2.275336798, 2.37526926, 6.13356024,
    6.86791015, 1.27202851, 0.43731927], count[:23].clip(min=0)**2 + (u_p_rel[:23] / p_rel[:23]).clip(min=0)**2)
array([9.87093819e+03, * nan, nan, 1.26734468e+00,
    2.63576258e+00, 4.87942754e-01, 3.13576759e-01, 2.62795654e-01,
    2.30431099e-01, 2.10665976e-01, 1.93506952e-01, 1.78207876e-01,
    1.67537398e-01, 1.59355677e-01, 1.53227081e-01, 1.52834293e-01,
    1.59822945e-01, 2.03765591e-01, 1.82454466e-01, 1.16024128e-01,
    1.07909799e-01, 2.28647936e-01, 5.63802484e-01])
```

it's looking like we found our bug.

∴ To get rid of these non-physical values of $n(p)$.

We use a clipped version of our array corrected_counts

```
#this clips negative counts which are non physical
corrected_count = corrected_count.clip(min=0)
y = np.sqrt(corrected_count[:23] / (p_rel[:23] * x * interpolated_fermi(p_rel[:23])))
y_regularised = np.sqrt(corrected_count[:23].clip(min=0) / (p_rel[:23] * x * interpolated_fermi(p_rel[:23])))
u_y = (y_regularised / 2) * np.sqrt((u_corrected_count[:23] / corrected_count[:23].clip(min=0))**2 + (2 * (u_p_rel[:23] / p_rel[:23]))**2 + (u_interpolated_fermi / interpolated_fermi(p_rel[:23]))**2)
```

we then use a regularised y in order to calculate our uncertainty

and now our linear fit works!

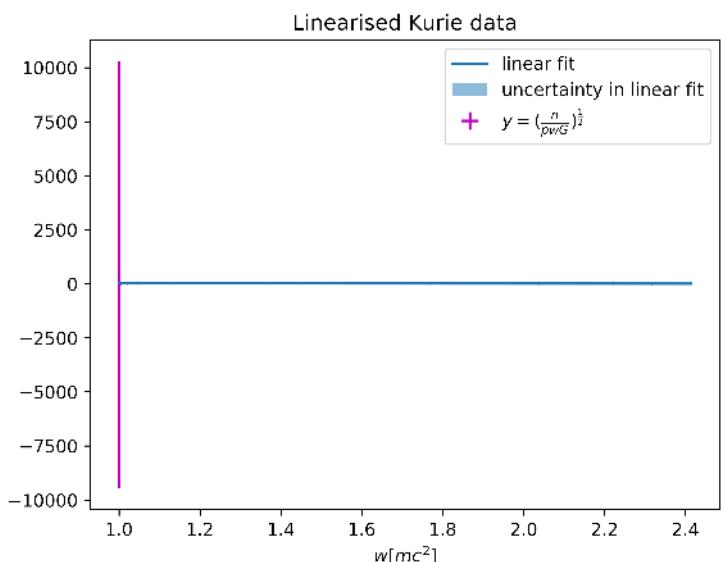
But it looks like
The uncertainty
is a bit large on
the points close to zero

(CODE model)

due to

Subtraction of the background radiation

• We know that: $n(p) \neq 0$

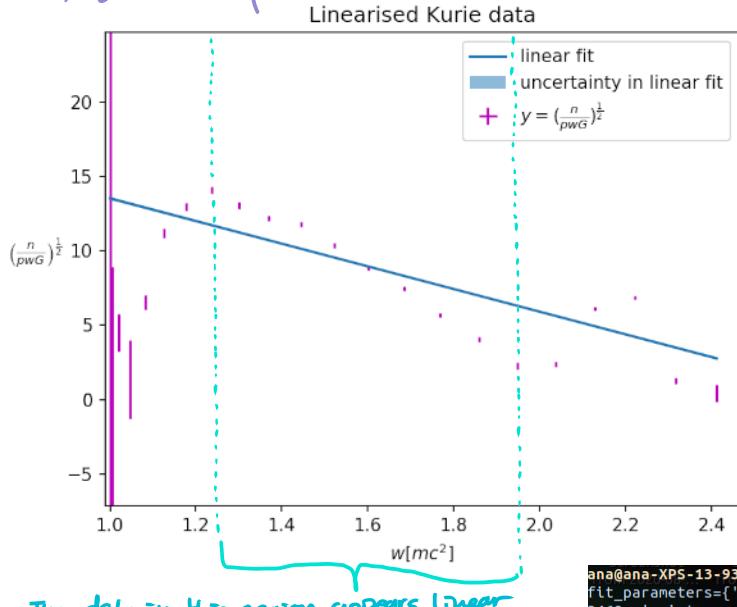


Analysis (Round II)

19/08/2020

It is clear that for an allowed transition, a plot of $\left(\frac{n}{pwG}\right)^{\frac{1}{2}}$ against w yields a straight line intersecting the w -axis at w_0 . Forbidden transitions may lead to a non linear Kurie plot, which still yields the correct end point. Providing the non-linearity is not too marked, the end point may still be determined accurately. Determine the end point of your spectrum in this way.

This is our plot zoomed in



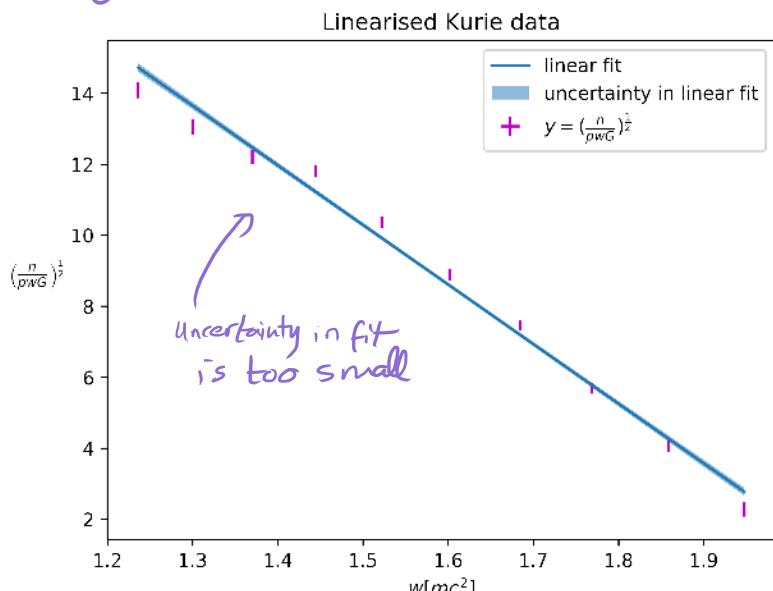
$$\left(\frac{n}{pwG}\right)^{\frac{1}{2}} = K_2(w_0 - \bar{w})$$

$$= K_2 w_0 - K_2 \bar{w}$$

1. from fit get gradient and y-intercept

```
ana@ana-XPS-13-9343:~/Documents/uni/PHS3000$ python3 betarays.py
fit_parameters={'slope': -16.787851289977773, 'u_slope': 0.2676432982059722, 'intercept': 35.47179379493469, 'u_intercept': 0.4333735111012989}
```

ENHANCE! BY SLICE



$$-K_2 = \text{slope} = -16.7879$$

$$u(\text{slope}) = 0.2676$$

$$\therefore K_2 = 16.7879 \pm 0.2676$$

$$K_2 w_0 = \text{Intercept} = 35.4718$$

$$u(\text{Intercept}) = 0.43339$$

$$\therefore w_0 = \frac{\text{Intercept}}{K_2}$$

$$\therefore u(w_0) = \sqrt{\left(\frac{u(K_2)}{K_2}\right)^2 + \left(\frac{u(\text{Intercept})}{\text{Intercept}}\right)^2} w_0$$

```
ana@ana-XPS-13-9343:~/Documents/uni/PHS3000$ python3 betarays.py
fit_parameters={'slope': -16.787851289977773, 'u_slope': 0.2676432982059722, 'intercept': 35.47179379493469, 'u_intercept': 0.4333735111012989}
K_2=16.787851289977773
sy = (frac{n}{pwG})^(frac(1){2})
color="g", barsab
w_0=2.1129442465464193
u_w_0=0.0424398995686856
w_0_rel=2.297460725203179
```

we calculated $w_0 = \frac{1.174 \text{ MeV}}{mc^2}$



Sceptical
duck

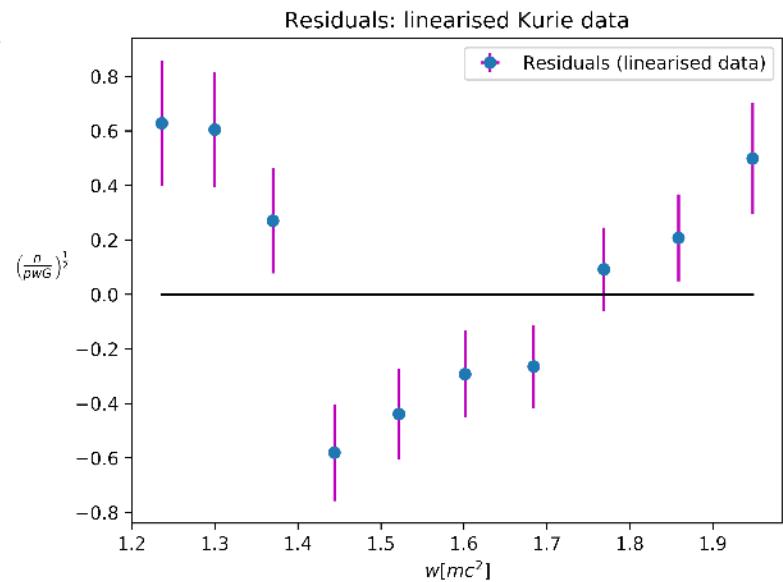
$$\therefore w_0 = 2.11 \pm 0.01$$

Analysis (Round II)

19/08/2020

Residuals

As expected
it doesn't look too
good!



Comparison to theory

20/08/2020

```
ana@ana-XPS-13-9343:~/Documents/uni/PHS3000$ python3 -i betarays.py
fit_parameters={'slope': -16.78785128997773, 'u_slope': 0.2676432982059722, 'intercept': 35.47179379493469, 'u_intercept': 0.4333735111012989}
K_2=16.78785128997773
w_0=2.1129442465404193
u_w_0=0.0424398995686856
w_0_rel=2.297460725203179
>>> diff = w_0_rel - w_0
>>> print(diff)
0.18451647866275955
>>> diff / u_w_0
left ( \frac{n}{(p \cdot w \cdot G)} \right )^{1/2} - right ( \frac{n}{(p \cdot w \cdot G)} \right )^{1/2} \approx 4.347712424816989
>>> plt.savefig('Kurie linear data plot .png')
```

$$\frac{w_{\text{rel}} - w_0}{\sigma}$$

Our result is 4.35σ away from the expected result.

it appears that we have underestimated our uncertainty

SciPy.org To optimise our fit.

SciPy.org Docs SciPy v1.5.2 Reference Guide Optimization and root finding (scipy.optimize ...)

scipy.optimize.curve_fit

```
scipy.optimize.curve_fit(f, xdata, ydata, p0=None, sigma=None, absolute_sigma=False, check_finite=True, bounds=(-inf, inf), method='None', jac=None, **kwargs)
Use non-linear least squares to fit a function, f, to data.
Assumes ydata = f(xdata, *params) + eps.
```

R: `optpt : array`
e: Optimal values for the parameters so that the sum of the squared residuals of
t: `f(xdata, *optpt)` - `ydata` is minimized.
u: `pcof : 2-D array`
r: The estimated covariance of `optpt`. The diagonals provide the variance of the parameter
n: estimate. To compute one standard deviation errors on the parameters use
s: `perr = np.sqrt(np.diag(pcof))`.

Scipy.optimize.curve_fit

requires a model $f = mx + c$

The parameters $\begin{cases} m \rightarrow K_2 = 16.7879 \pm 0.2676 \\ x \rightarrow \bar{w} \pm u(\bar{w}) \\ c \rightarrow K_2 w_0 = \text{Intercept} = 35.4718 \pm 0.4334 \end{cases}$

we need to propagate uncertainty [to obtain $u(f)$]
as we know

$$u(f(x, m, c))^2 = \left(\frac{\partial f}{\partial x} u(x) \right)^2 + \left(\frac{\partial f}{\partial m} u(m) \right)^2 + \left(\frac{\partial f}{\partial c} u(c) \right)^2$$

$$u(f(x, m, c)) = \sqrt{(m u(x))^2 + (x u(m))^2 + (u(c))^2}$$

CODE

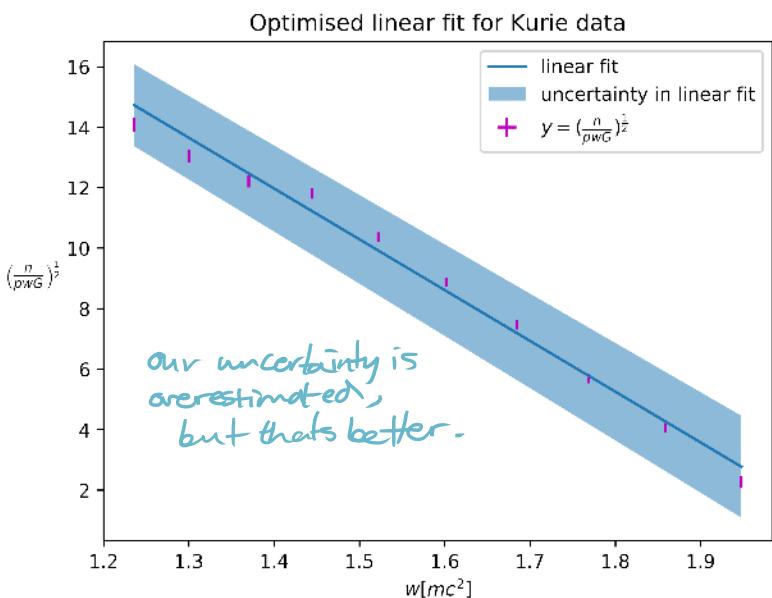
```
# linear model for optimize.curve_fit()
def f(x, m, c):
    return m * x + c
# uncertainty in linear model f
u_f = np.sqrt((K_2 * u_x)**2 + (x * u_K_2)**2 + (u_intercept)**2)
# optimising our fit, unpack into optpt, pcof
optpt, pcof = scipy.optimize.curve_fit(f, x, y, sigma=u_y, absolute_sigma=False)
# To compute one standard deviation errors on the parameters use
perr = np.sqrt(np.diag(pcof))
opt_K_2, opt_intercept = optpt
u_opt_K_2, u_opt_intercept = perr
```

```
ana@ana-XPS-13-9343:~/Documents/uni/PHS3000$ python3 betarays.py
linear fit gradient: K_2 = 16.78785128997773
linear fit intercept: intercept = 35.47179379493469
THIS IS OUR EXPECTED RESULT theory_w_0_rel = 2.297460725203179
THIS IS OUR RESULT w_0 = 2.1129442465404193 ± 0.0424398995686856
optimised gradient -16.787851225186017 ± 0.6656759006640471
optimised intercept 35.47179369119481 ± 1.0778760504058646
```

Analysis (Round II)

Optimised fit plot -

20/08/2020



```
# using our results to find opt_w_0
# recall that:
# opt_K_2, opt intercept = popt
# u_opt_K_2, u opt intercept = perr
opt_w_0 = opt_intercept / opt_K_2
u_opt_w_0 = np.sqrt((u_opt_K_2 / - opt_K_2)**2 + (u_opt_intercept / opt_intercept)**2) * opt_w_0

print("EXPECTED RESULT theory_w_0_rel = ")
print("post-optimisation result opt_w_0 = ") ± {u_opt_w_0}\n"
```

```
ana@ana-XPS-13-9343:~/Documents/unil/PHS3000$ python3 -i betarays.py
linear fit gradient: K_2 = 16.78785128997773
linear fit intercept: intercept = 35.47179379493469

EXPECTED RESULT theory_w_0_rel = 2.297460725203179
pre-optimisation result w_0 = 2.1129442465404193 ± 0.0424398995686856

optimised gradient -16.787851225186017 ± 0.6656759006640471
optimised intercept 35.47179369119481 ± 1.0778760504058646

EXPECTED RESULT theory_w_0_rel = 2.297460725203179
post-optimisation result opt_w_0 = 2.112944248515746 ± 0.10555548613871778

>>> diff = theory_w_0_rel - opt_w_0
>>> diff / u_opt_w_0
1.7480519813526991 ← 1.740 away from true value
```



$$\therefore w_0 = 2.11 \pm 0.11 \text{ mc}^2$$

we still don't have agreement but our result makes more sense

-non-relativistic result -

$$w_0 = 1.08 \pm 0.05 \text{ MeV}$$

Siegbarh Sn(̄)

$$S_n(\bar{w}) = \bar{w}^2 - 1 + (w_0 - \bar{w})^2 \quad [16]$$

is the best fit if $\Delta j = \pm 2$ (First forbidden)

$$\begin{aligned} \left(\frac{n}{pWG}\right)^{\frac{1}{2}} &= K_2 S_n(\bar{w}) / (w_0 - \bar{w}) \\ &= K_2 \sqrt{\bar{w}^2 - 1 + (w_0 - \bar{w})^2} / (w_0 - \bar{w}) \\ &= K_2 w_0 \sqrt{\bar{w}^2 - 1 (w_0 - \bar{w})^2} - K_2 \bar{w} \sqrt{\bar{w}^2 - 1 (w_0 - \bar{w})^2} \end{aligned}$$

This relation is non-linear anymore
we would need to write a new fit in order to use it to analyse our data

This might be more comprehensive than our linearised Kurie relation

The Shape factor

Investigate the results of using the shape factor given in Siegbahn, page 497.

From your determined end point energy, calculate the corresponding beta particle kinetic energy, and compare to the accepted value of the (maximum) beta particle energy for ^{137}Cs .

Discussion

20/08/2020

The experimental process was bumpy and complicated.

We had very little clue of what was to be expected when operating the apparatus, but also our early understanding of the theory was incomplete.

This is very visible when we read the method we followed.

during our first attempt at running the apparatus on (06/08/2020)

We expected that learning about the control interface was going to be a more efficient process.

Our aim during **(1) RUN** was to understand the experimental control interface, this aim was reasonable but our naïve approach made the process very slow. We arbitrarily chose our starting current to be 1.00A (0.9932 ± 0.0005 A). This choice had problematic repercussions, since the momentum spectrum was incomplete. Our initial choice in the increments to the current was not too ideal given the range of the current available in the apparatus.

We chose to run the experiment in constant time. This choice remained throughout every run we planned. Our choice to set every count to be (600.00 ± 0.05 s) proved to be non-ideal given that we ran out of time during our first booking with the experimental apparatus.

As we already implied, the controls interface was highly unintuitive during **(1) RUN** we noticed that our set current was recorded as zero values every time we set a half integer value for the lens coil current. We decided to continue with **(1) RUN** without modification to the method. We aimed to solve the observed problems during **(2) RUN**.

During **(2) RUN** our first goal was to calibrate the magnetometer probe. After this we started investigating how to operate the bias coil current and its effect in the recording of the magnetic field.

We set the value of the bias coil current to 0.190A (0.1890 ± 0.0005 A) and noticed that only the y-component of the field approached zero.

We were unable to understand a key part of our experimental controls at this stage. The ENABLE button activated the current, but if pressed again it would deactivate the current too.

Because of this "small" misunderstanding we observed the same problem with our record of the current. Everytime we made an increment we would press the enable button, tragically deactivating the current.

We understood this during **(2) RUN**.

This discovery lead us to understand how to attempt to cancel the magnetic field of earth.

(3) RUN our choice for the initial lens coil current to be 1.00A (0.9931 ± 0.0005 A) remained, but this time we were able to get a record of every increment made to our lens coil current. We decided to set our increments to be 0.250 A in order to obtain a more comprehensive spectrum.

After disabling the lens coil current we set the value of the bias coil current to 0.830A which allowed us to minimize earth's magnetic field.

—unsolved problems on (06/08/2020)—

- During every run, we chose an inadequate range for our current (1A to 3.5A)
- we were unable to come with an idea to figure out the count of the background radiation.
- We plotted our result and learned about the inadequacy of our range but unfortunately we ran out of time during this session.

This experience yielded NO analysis.

Discussion

20/08/2020

After all of these issues, we had another opportunity to expand in our theoretical understanding of the experiment.

The relations between current, momentum and rigidity we obtained from Siegbahn ($R = Bp$) and Quade, Halliday ($P = kI$) allowed us to determine the proportionality constant between momentum, current and the magnetic field originated in the apparatus.

Since the Rigidity is a measure of a particle's momentum
higher Rigidity \Leftrightarrow higher momentum.

We decided that it was important for us to reduce the running time for the count of each current.
we made it a priority that we should only count for as long as necessary to maintain a count uncertainty of less than 5%.

Experimental Procedure (Round II)

We were more organised this time around, in addition during our educational hiatus we came up with a way to determine the background count -

We decided that we would use current increment values of 0.1 A to maximise our chance at finding an adequately shaped momentum spectrum -

To minimise earth's magnetic field we set the bias current to 0.720 A (0.716 ± 0.0005 A). Similarly to before, we set our experiment to run in constant time, and before starting **① RUN** we proceeded to close the shutter for the Cs-137 source in order to record the background radiation for 6 minutes. (we did this 4 times).

① RUN was better planned on 18/08, this allowed us to have a more automatic process outlined. we did not encounter issues for any of the counts, we started counting from the range: [0, 3.6 A]

where $0A = 0.0000 \pm 0.0005$ A and $3.6000A = 3.5245 \pm 0.0005$ A.

We were able to calibrate our momentum spectrum with the expected K-peak and we were able to correct our count for an average of the 4 recorded counts of the background radiation.

The background correction led us to encounter an error with the numerical evaluations of negative counts inside a square root which had to be addressed by clipping our un-physical negative counts.

We proceeded to calculate a non zero uncertainty in the clipped values by defining a "regularised" value for the linearised count as per the model in equation 4.

Our linear fit was successful after these corrections but the momentum data had to be sliced to a smaller apparently linear range as was demonstrated by the plots in our analysis.

Since all uncertainties were propagated correctly to the best of our knowledge. We decided to try our hand at optimisation.

Discussion

20/08/2020

The linear fit produced a result with an unreasonably small uncertainty. This lead us to try an optimisation function based on a Least squares fit (from scipy library)

We used a linear model, our energy data, and linear count data as inputs to this optimisation function.

The function can be configured to use only the relative magnitudes of uncertainties and scaling them by a constant factor that is set by imposing that $\chi^2_{\text{red}} = 1$.

This by definition improves the fit.

This means that our uncertainties are scaled to match the sample variance residuals after our fit occurs.

This certainly improved the scale of our uncertainty in the result for the total released energy. Putting our value 1.746 away from the true value for Cs-137

our result after optimisation is $w_0 = 1.08 \pm 0.05$ MeV.

Reading
the
docs

near verbatim
from reference
[17]

Conclusion

20/08/2020

Using a magnetic spectrometer we measured the energy spectrum of electrons emitted by β^- radiation from a Cs-137 source. Our investigation focused in determining the relationships between current, moment am and energy.

We found that the total energy released by the observed nuclear transition was $W = 1.08 \pm 0.05 \text{ MeV}$. Our result disagrees with the expected theoretical result.

```

1 # PHS3000 - LOGBOOK1
2 # Betarays - radioactive decay Cs - 137
3 # Ana Fabela, 15/08/2020
4 import monashera.PHS3000 as spa
5 from scipy.interpolate import interp1d
6 import numpy as np
7 import pandas as pd
8 import pytz
9 import matplotlib
10 import matplotlib.pyplot as plt
11 from pprint import pprint
12 import scipy.optimize
13
14 plt.rcParams['figure.dpi'] = 150
15
16 # Globals
17 c = 299792458 # [m/s]
18 mass_e = 9.109338356e-31 # [kg]
19 eV = 1.602176634e-19 # [J]
20 MeV = 1e6 * eV
21 keV = 1e3 * eV
22 rel_energy_unit = mass_e * c**2 # to convert SI into relativistic or viceversa
23
24 data = spa.betaray.read_data(r'beta-ray_data.csv')
25
26 # valid data slicing from csv file
27 j = 0
28 for row in data:
29     # print(f'{j=}')
30     if row[0] == pd.Timestamp('2020-08-18 10:26:00+10:00', tz=pytz.FixedOffset(360)):
31         valid_data = data[j:]
32         continue
33     j+=1
34
35 background_count_data = []
36 count = []
37 lens_current = []
38 u_lens_current = []
39 for row in valid_data:
40     if row[3] == 'Closed':
41         # print(row[3])
42         background_count_data.append(row)
43         continue
44     count.append(row[5])
45     lens_current.append(row[6])
46     u_lens_current.append(row[7])
47
48 background_count = []
49 # correcting our data by removing avg background count
50 for row in background_count_data:
51     background_count.append(row[5])
52 avg_background_count = np.mean(background_count)
53 # print(f"We want to subtract this background count from our data (avg_background_count={})")
54 # calculating fractional uncertainty in total background count (delta_t = 24 min)
55 total_background = np.sum(background_count)
56 u_avg_background_count = np.sqrt(total_background) / 4
57
58 # uncertainty in the corrected count
59 corrected_count = count - avg_background_count
60 u_corrected_count = np.sqrt(count + u_avg_background_count**2)
61
62
63 # Finding constant of proportionality in p = kI
64 # calibration peak (K) index of k peak is i=20
65 T_K = 624.21 * keV / rel_energy_unit
66 k = np.sqrt((T_K + 1)**2 - 1) / lens_current[20]
67 # print(f"(k={k})")
68 u_k = k * (0.0005 / lens_current[20])
69 # print(f"absolute uncertainty: (u_k = {u_k})")
70 # print(f"fractional uncertainty: (u_k / k) = {u_k / k} = {u_k / k}\n")
71
72 # The momentum spectrum
73 lens_current = np.array(lens_current)
74 p_rel = k * lens_current
75 u_p_rel = p_rel * np.sqrt((u_k / k)**2 + (0.0005 / lens_current)**2)
76 # print(f"absolute uncertainty u(p_rel):\n{u_p_rel}")
77 # print(f"fractional uncertainty u(p_rel) / p_rel:\n{(u_p_rel / p_rel)}")
78
79 # plot
80 plt.figure()
81 plt.errorbar(
82     p_rel, corrected_count, xerr=u_p_rel, yerr=u_corrected_count,
83     marker="None", ecolor="m", label=r"$\text{Sn}(\text{p})_{\text{corrected}}$",
84     color="g", barsabove=True
85 )
86 plt.title(r"\beta^{-} particle momentum spectrum")
87 plt.xlabel("p [mc]")
88 plt.ylabel("n(p)")
89 plt.legend()
90 spa.savefig('count_vs_momentum_no_background_error.png')
91 plt.show()
92
93 ##### KURIE/Fermi PLOT #####
94
95 dp_rel = p_rel[1]-p_rel[0]
96
97 # getting interpolated!
98 fermi_data = spa.betaray.modified_fermi_function_data
99 interpolated_fermi = interp1d(fermi_data[:,0], fermi_data[:,1], kind='cubic')
100
101 ##### THEORETICAL #####
102 # Desintegration energy
103 # Cs-137 disintegrates by beta minus emission to the ground state of Ba-137 (5,6 %)
104 theory_w_0 = 1.174 * MeV
105 theory_w_0_rel = theory_w_0 / rel_energy_unit
106 p_0_rel = np.sqrt(theory_w_0_rel**2 - 1) / (mass_e * c)
107 # print(p_0_rel)
108
109 # defining the theoretical count (Kuriefunction)
110 K_1 = 1 # ?
111 Sn = 1
112 def n(p_rel):
113     w_rel = np.sqrt(p_rel**2 + 1) # relativistic energy units
114     n = K_1 * Sn * (w_rel * interpolated_fermi(p_rel) / p_rel) * p_rel**2 * (theory_w_0_rel - w_rel)**2
115     return n, w_rel
116
117
118 n_p_rel, w_rel = n(p_rel[:22]) # call and unpack n(p)
119
120 # equation (3) in script
121 N = n_p_rel * dp_rel
122
123 # plot
124 plt.figure()
125 plt.plot(
126     p_rel[:22], N, marker="None",
127     linestyle="--"
128 )
129 plt.title("Kurie relation")
130 plt.xlabel("p [mc]")
131 plt.ylabel("n(p) dp")
132 spa.savefig('Kurie_plot.png')
133 plt.show()
134
135 ##### THEORETICAL #####
136 ##### EXPERIMENTAL #####
137
138 # plot
139 plt.figure()
140 plt.errorbar(
141     p_rel[:23], corrected_count[:23], xerr=u_p_rel[:23], yerr=u_corrected_count[:23],
142     marker="None", ecolor="m", label=r"$\text{Sn}(\text{p})_{\text{corrected}}$",
143     color="g", barsabove=True
144 )
145 plt.title(r"\beta^{-} particle momentum spectrum")
146 plt.xlabel("p [mc]")
147 plt.ylabel("n(p)")
148 plt.legend()
149 spa.savefig('count_vs_momentum_no_background_error.png')

```

```

150 plt.show()
151
152 ##### EXPERIMENTAL #####
153 ##### KURIE/Fermi PLOT #####
154 ##### Linear fit #####
155 # initial slice [:23]
156 # second slice [8:18]
157 n_p_rel, w_rel = n(p_rel[8:18])
158
159 # our sliced data linearised
160 x = w_rel
161 u_x = u_p_rel[8:18]
162
163 # uncertainty in interpolated fermi
164 u_interpolated_fermi = np.sqrt((u_p_rel[8:18] / p_rel[8:18])**2 + (u_x / x)**2) * interpolated_fermi(p_rel[8:18])
165
166 # this clips negative counts which are non physical
167 corrected_count = corrected_count.clip(min=0)
168
169 # LINEARISED KURIE
170 y = np.sqrt(corrected_count[8:18] / (p_rel[8:18] * x * interpolated_fermi(p_rel[8:18])))
171 # regularising y to avoid zero u_y
172 y_regularised = np.sqrt(corrected_count[8:18].clip(min=1) / (p_rel[8:18] * x * interpolated_fermi(p_rel[8:18])))
173 u_y = (y_regularised / 2) * np.sqrt((u_corrected_count[8:18] / corrected_count[8:18]).clip(min=1)**2 + (2 * (u_p_rel[8:18] / p_rel[8:18]))**2) + (u_interpolated_fermi / interpolated_fermi(p_rel[8:18]))
174
175 fit_results = spa.linear_fit(x, y, u_y=u_y)
176 # making our linear fit with one sigma uncertainty
177 y_fit = fit_results.best_fit
178 u_y_fit = fit_results.eval_uncertainty(sigma=1)
179
180 # calculating values from fit results
181 fit_parameters = spa.get_fit_parameters(fit_results)
182 # print(f"fit_parameters={fit_parameters}")
183
184 # using our results to find w_0
185 K_2 = - fit_parameters["slope"]
186 u_K_2 = fit_parameters["u_slope"]
187 intercept = fit_parameters["intercept"]
188 u_intercept = fit_parameters["u_intercept"]
189 w_0 = intercept / K_2
190 u_w_0 = np.sqrt((u_K_2 / K_2)**2 + (u_intercept / intercept)**2) * w_0
191
192 print(f"linear fit gradient: (K_2 = {K_2})")
193 print(f"linear fit intercept: (Intercept = {intercept})")
194
195 print(f"EXPECTED RESULT (theory_w_0_rel = {w_0})")
196 # pre-optimisation result
197 print(f"pre-optimisation result (w_0 = {w_0} ± {u_w_0})")
198
199 # plot
200 plt.figure()
201 plt.errorbar(
202     x, y, xerr=u_p_rel[8:18], yerr=u_y,
203     marker="None", linestyle="None", ecolor="m",
204     label=r"$y = (\frac{n}{p \cdot G})^{(\frac{1}{2})}$", color="g", barsabove=True
205 )
206 plt.plot(
207     x, y_fit, marker="None",
208     linestyle="",
209     label="linear fit"
210 )
211 plt.fill_between(
212     x, y_fit - u_y_fit,
213     y_fit + u_y_fit,
214     alpha=0.5,
215     label="uncertainty in linear fit"
216 )
217 plt.title("Linearised Kurie data")
218 plt.xlabel(r"$\sqrt{mc^2/G}$")
219 plt.ylabel(r"$\sqrt{\left(\frac{n}{p \cdot G}\right)^{(\frac{1}{2})}}$)", rotation=0, labelpad=18)
220 plt.legend()
221 spa.savefig("Kurie_linear_data_plot_.png")
222 plt.show()
223
224 ##### Linear fit #####
225 ##### Linear fit residuals#####
226
227 linear_residuals = y_fit - y # linear residuals (linear best fit - linearised data)
228
229 # plot
230 plt.figure()
231 plt.errorbar(
232     x, linear_residuals, xerr=u_p_rel[8:18], yerr=u_y,
233     marker="o", ecolor="", linestyle="",
234     label="Residuals (linearised data)"
235 )
236 plt.plot([x[0], x[-1]], [0, 0], color="k")
237 plt.title("Residuals: linearised Kurie data")
238 plt.xlabel(r"$\sqrt{mc^2/G}$")
239 plt.ylabel(r"$\sqrt{\left(\frac{n}{p \cdot G}\right)^{(\frac{1}{2})}}$)", rotation=0, labelpad=18)
240 plt.legend()
241 spa.savefig("linear_residuals_Kurie_linear_data.png")
242 plt.show()
243
244 ##### Linear fit residuals#####
245
246 # linear model for optimize.curve_fit()
247 def f(x, m, c):
248     return m * x + c
249
250 # optimising our fit, unpack into popt, pcov
251 popt, pcov = scipy.optimize.curve_fit(f, x, y, sigma=u_y, absolute_sigma=False)
252 # To compute one standard deviation errors on the parameters use
253 perr = np.sqrt(np.diag(pcov))
254
255 opt_K_2, opt_intercept = popt
256 u_opt_K_2, u_opt_intercept = perr
257
258 print(f"optimised gradient (opt_K_2) ± (u_opt_K_2)")
259 print(f"optimised intercept (opt_intercept) ± (u_opt_intercept)\n")
260
261 optimised_fit = f(x, opt_K_2, opt_intercept)
262 # uncertainty in linear model f given optimal fit
263 u_f = np.sqrt((opt_K_2 * u_x)**2 + (x * u_opt_K_2)**2 + (u_opt_intercept)**2)
264
265 # using our results to find opt_w_0
266 opt_w_0 = opt_intercept / - opt_K_2
267 u_opt_w_0 = np.sqrt((u_opt_K_2 / opt_K_2)**2 + (u_opt_intercept / opt_intercept)**2) * opt_w_0
268
269 print(f"EXPECTED RESULT (theory_w_0_rel = {opt_w_0})")
270 print(f"post-optimisation result (opt_w_0 = {opt_w_0} ± {u_opt_w_0})")
271 print(f"non-relativistic w_0 = (opt_w_0 * rel_energy_unit / MeV) ± (u_opt_w_0 * rel_energy_unit / MeV)\n")
272
273 # OPTIMISED FIT PLOT
274 plt.figure()
275 plt.errorbar(
276     x, y, xerr=u_p_rel[8:18], yerr=u_y,
277     marker="None", linestyle="None", ecolor="m",
278     label=r"$y = (\frac{n}{p \cdot G})^{(\frac{1}{2})}$", color="g", barsabove=True
279 )
280 plt.plot(
281     x, optimised_fit, marker="None",
282     linestyle="",
283     label="linear fit"
284 )
285 plt.fill_between(
286     x, optimised_fit - u_f,
287     optimised_fit + u_f,
288     alpha=0.5,
289     label="uncertainty in linear fit"
290 )
291 plt.title("Optimised linear fit for Kurie data")
292 plt.xlabel(r"$\sqrt{mc^2/G}$")
293 plt.ylabel(r"$\sqrt{\left(\frac{n}{p \cdot G}\right)^{(\frac{1}{2})}}$)", rotation=0, labelpad=18)
294 plt.legend()
295 spa.savefig("OPTIMISED_Kurie_linear_data_plot_.png")
296 plt.show()
297
298 ##### optimised fit residuals#####

```

```

299
300 optimised_residuals = optimised_fit - y
301 # plot
302 plt.figure()
303 plt.errorbar(
304     x, optimised_residuals, xerr=u_p_rel[8:18], yerr=u_f,
305     marker="o", ecolor="", linestyle="None",
306     label="Residuals (linearised data)"
307 )
308 plt.plot([x[0], x[-1]], [0,0], color="k")
309 plt.title("Residuals: optimised fit for linear Kurie data")
310 plt.xlabel("5w [mc^2]$")
311 plt.ylabel("$\\left( \\frac{n}{p \cdot G} \\right)^{\\frac{1}{2}}$",
            rotation=0, labelpad=18)
312 plt.legend()
313 spa.savefig("OPTIMISED_linear_residuals_Kurie_linear_data.png")
314 plt.show()
315
316 # #####optimised fit residuals#####

```

Logbook 1: References

- [0] Monash SPA (2020) 4.4 Measurement of β-ray spectra. Retrieved from Moodle (student site)
- [1] Wikipedia (2020). Beta decay. Retrieved from https://en.wikipedia.org/wiki/Beta_decay
- [2] Wikipedia (2020). Nuclide. Retrieved from <https://en.wikipedia.org/wiki/Nuclide>
- [3] Wikipedia (2020). Isobar. Retrieved from [https://en.wikipedia.org/wiki/Isobar_\(nuclide\)](https://en.wikipedia.org/wiki/Isobar_(nuclide))
- [4] Nave (2001). Hyperphysics: Energy and momentum spectra for Beta decay. Retrieved from <http://hyperphysics.phy-astr.gsu.edu/hbase/Nuclear/beta2.html>
- [5] Wikipedia (2019). Internal conversion. Retrieved from https://en.wikipedia.org/wiki/Internal_conversion
- [6] Witcher (1941) [An Electron Lens Type of Beta-Ray Spectrometer.pdf](#) Retrieved from (4.4 Measurement of β-ray spectra: References)
- [7] Bell, et al. (2015). The Effect of the Earth's and Stray Magnetic Fields on Mobile Mass Spectrometer Systems. <https://doi.org/10.1007/s13361-014-1027-4>
- [8] Krane, Section 2.7(also 2.8);12.8 (also 12.6 pp.386-387). <http://phet.colorado.edu/en/simulation/beta-decay>
- [9] Siegbahn, (1968) [Beta Ray Spectrometer Theory and Design pp79-201.pdf](#) Retrieved from (4.4 Measurement of β-ray spectra: References)
- [10] Quade, EA Halliday, D (1947) [Quade, EA Halliday, D 1947.pdf](#) Retrieved from (4.4 Measurement of β-ray spectra: References)
- [11] Wikipedia (2019). Rigidity (electromagnetism). Retrieved from [https://en.wikipedia.org/wiki/Rigidity_\(electromagnetism\)](https://en.wikipedia.org/wiki/Rigidity_(electromagnetism))
- [12] R.G.HelmerandV.P.Chechev (2006)Cs-137_tables. Retrieved from http://www.nucleide.org/DDEP_WG/Nuclides/Cs-137_tables.pdf
- [13] Wikipedia (2020). Chi-square distribution. Retrieved from https://en.wikipedia.org/wiki/Chi-square_distribution
- [14] Chris Billington (2020) victoria.py. Retrieved from <https://github.com/chrisbillington/chrisbillington.github.io/blob/master/victoria.py>
- [15] Wikipedia (2020). Propagation of uncertainty Retrieved from https://en.wikipedia.org/wiki/Propagation_of_uncertainty
- [16] Siegbahn, (1968) [Beta Ray Spectrometer vol1 pp496-498 K Siegbahn.pdf](#) Retrieved from (4.4 Measurement of β-ray spectra: References)
- [17] The SciPy community (2020) scipy.optimize_curve. Retrieved from https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html