Risk Assesment form

3047	RISK DESCRIPTION	TREND	CURRENT	RESIDUAL
	SCI_SPA_Level 3_Alpha particles from Am-241_(1.2 Range and energy loss of alpha particles)		Negligible	Not Assessed

RISK OWNER	RISK IDENTIFIED ON	LAST REVIEWED ON	NEXT SCHEDULED REVIEW
Manuel Emilio Pumarol Crestar	05/07/2017	05/07/2017	05/07/2020

RISK FACTOR(S)	EXISTING CONTROL(S)	PROPOSED CONTROL(S)	TREATMENT OWNER	DUE DATE
Radiation -Sealed source Am-241. Radiation Exposure from Am-241 (<0.1uSv/hr @detector end, 1.4uSv/hr at chamber surface 24/5/2013)	Control: Am-241 source is in a sealed brass vacuum chamber, with perspex guard. Maximum exposure will be less than 0.1uSv/hr.	Student musn't interact with Radioactive source.		
Vacuum system can be hazardous if operated incorrectly.	Control: Vacuum system is positioned behind perspex so that students cannot easily adjust it.	Student musn't interact with Vacuum Chamber		

Range and energy loss of α-particles in air.

05/09/2020

Ana Fabela Hinojosa

Aim

We investigate the effects of air on α-particles; determining the energy spectrum of α-particles emitted from an ^{241}Am source to determine the mean energy of the particles emitted from a shielded source.

Introduction

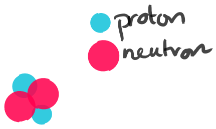


Fig 1. schematic drawing α-particle AKA Helium nucleus.

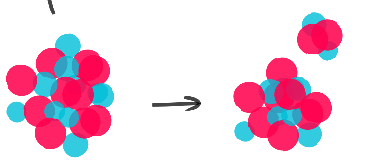
Ionising radiation of a relatively large particle such as the Helium nucleus is the least penetrating type of radiation

α-particles can be stopped completely by several cm of air [0]

Theory

Elements with large numbers of neutrons tend to undergo α-decay (emmission of α-particles)

After the expulsion of the particle from the nucleus the parent nucleus transforms to another nucleus with atomic number that is reduced by 2 and a mass number that is reduced by 4.



$${}^{A}_{Z}X \rightarrow {}^{A-4}_{Z-2}Y + {}^{4}_{2}He$$

Fig2. α-decay and equation

• The strong nuclear force is overpowered by the Coulomb repulsion between the α-particle and the rest of the nucleus which causes the ejection.

* Due to energy considerations nucleons are prefrentially ejected * bound as a Helium nucleus (?) [0]

Energy released from this event (mass defect)

$$E = mc^2$$

Conversion from volts to energy

$V = \frac{E}{Q} \Rightarrow VQ = E$

 : $Q = 2e^- = 3.204 \times 10^{-19} C$

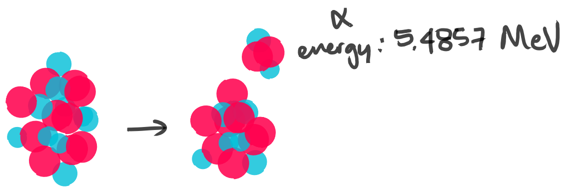
Energy of α-particle detected

$E = 5V \times 3.204 \times 10^{-19} C$

The decay path ^{241}Am

05/09/2020

The most probable decay-event occurs with 85% probability



$$^{241}_{95}Am \rightarrow \, ^{237}_{93}Ne + \, ^{4}_{2}He + \gamma : 59.6 \, keV$$
[2]

$$\tau_{Am} = 458\,y \rightsquigarrow \tau_{Np} = 2.14 \times 10^{6} \text{ years.}$$

Fig 3. Schematic drawing of ^{241}Am decay

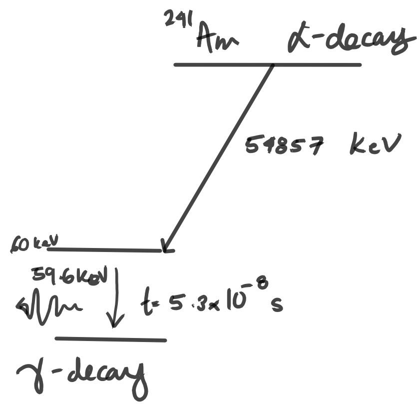


Fig 4. relevant transition copied from laboratory script [0]

The range of α-particles.

Because α-particles are charged, they interact with electrons in the surrounding medium ⇒ Kinetic energy loss.

Geiger's (1910) definition of the range of α-particles in air:

(1) $R \propto E^{3/2}$ R - distance from source
E - kinetic energy

Bethe-Bloch

The change in energy of heavy charged particles interacting with a gas

$$(2) \quad -\frac{dE}{dx} = B\rho E_0^{-0.73}$$

B - 2.58 MeV $(cm\ kg\ m^{-3})^{-1}$
E_0 - initial kinetic energy (MeV)
ρ - density of air at 1 atmosphere
($\rho = 1.204\, kg m^{-3}$ at 20°C and 101.325 kPa)
x - distance (cm)

Integrating (2) wrt distance

$$R = \int_0^R dx = \int_{E_0}^{0} \frac{dE}{(dE/dx)} = \frac{E_0^{1.73}}{1.73\, B_\rho} \qquad (3)$$

Thus the theoretical range is

$$R_{atm} = 0.186\, E_0^{1.73} \text{ (cm)} \qquad (4)$$

Non-atmospheric Pressure

05/09/2020

Energy loss is proportional to the number of gas molecules encountered. ∴ we use number density and pressure.

$$R = R_{atm} \frac{P_{atm}}{P} \qquad (4a)$$

P - pressure

Patm - atmospheric pressure

Assuming that the air behaves as an ideal gas

⇒ the density will vary linearly with pressure.

by $\rho = \frac{P}{R_d T}$

T - temperature (K)

P - pressure (Pa) *

R_d - gas constant (287.05 J/kg K)

$$\therefore R = 64.31 \frac{T}{P} E_0^{1.73} \qquad (5)$$

when the α-detector is fixed at x_0 (cm) away

There is a pressure P_0 (Pa)

preventing α-particles from reaching the detector

$$P_0 = 64.31 \frac{T}{x_0} E_0^{1.73} \qquad (6)$$

- E_0 (MeV)

Energy after passing through air

by integrating (2)

$$E = \left(E_0^{1.73} - 0.0156 \frac{P \Delta x}{T}\right)^{0.578} \quad (\text{MeV}) \qquad (7)$$

is the energy of α-particles after passing through a distance (Δx) (cm) with initial energy E_0 - (T (K), P (Pa))

Energy spectrum

All particles from a particular transition have the same energy if the medium is homogeneous, the range of α-particles should be identical. The molecular nature of air introduces a random variation into the effective path lengths.

Range straggling

05/09/2020

for a large number of particles (n_0), with initial velocity (V_0) the number dn having ranges between x and dx is normally distributed

$$\frac{dn}{dx} = n_0 \frac{1}{\alpha\sqrt{\pi}} e^{-\left(\frac{x-R}{\alpha}\right)^2} \quad (8)$$

α - straggling parameter
it has been shown to vary linearly with E_0

$$\therefore \ kR \cong \alpha \quad (9)$$

where k is a constant
and $k = 0.015$ for α-particles travelling through atmospheric pressure.

α - can be determined by $\alpha = 0.6066 \times FWHM$

we must consider that:

An additional dispersion in the energy spectrum occurs due to a gold layer that covers the surface where the radioactive source rests.

$\therefore$ we modify $\alpha^2 = \alpha_{air}^2 + \alpha_{source}^2$, (10)

↑ straggling parameter due to source construction.

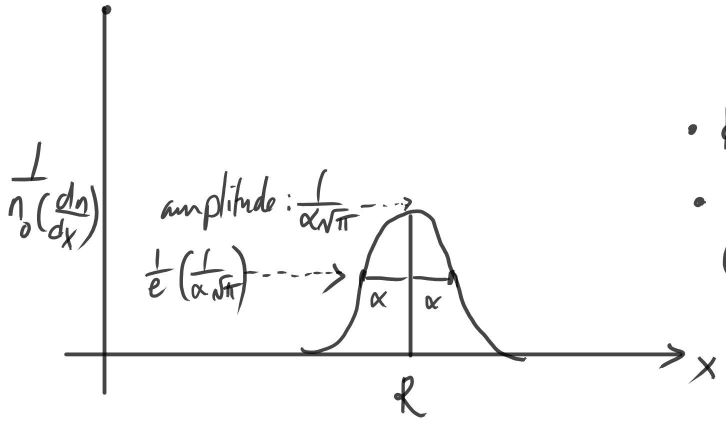
Integrating equation (8)
we obtain the number of α-particles remaining at a distance x.

$$n(x) = n_0 - \int_0^x dn \quad (11)$$

$$\therefore \quad n(x) = n_0 \left\{ 1 - \int_0^x \frac{1}{\alpha\sqrt{\pi}} \exp\left[-\left(\frac{x-R}{\alpha}\right)^2 \right] dx \right. \quad (12)$$

we examine the number of α-particles remaining as the pressure in the chambers is varied, rather than n(x)

08/09/2020



- R is mean range
- α is range straggling parameter (characterising distributionwidth.)

Fig 5. Gaussian distribution of α-particles

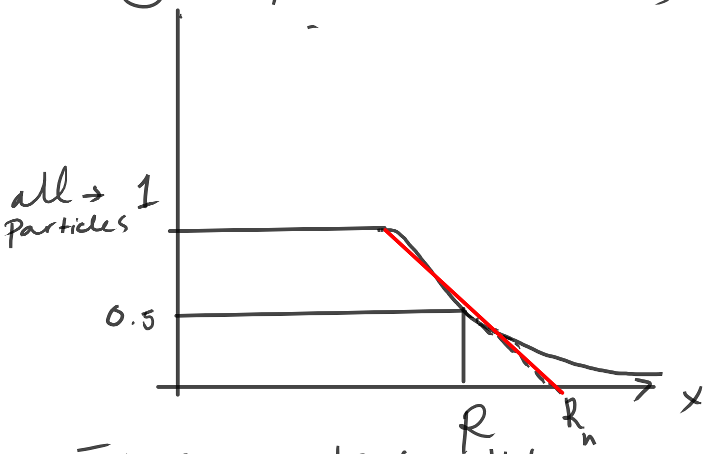


Fig 6. number of particles reaching detector as a function of distance.

from equation (5) : $R = 64.31 \frac{T}{p} E_0^{1.73}$

equation (6) : $p_0 = 64.31 \frac{T}{x_0} E_0^{1.73}$

we see that p can be substituted by x_0

(p is air pressure, x_0 is propagation distance

$$\therefore n(p) = n_0 \left\{ 1 - \int_0^p \frac{1}{\alpha\sqrt{\pi}} \exp\left[-\left(\frac{p-p_0}{\alpha}\right)^2 \right] dp \right\} \quad (12a)$$

where α is proportional to p_0

We can write (12a) using :

$$\operatorname{erf}(y) = \frac{2}{\sqrt{\pi}} \int_0^y e^{-t^2} dt$$

$$\therefore n(p) = \frac{n_0}{2}\left(1 - \operatorname{erf}\left(\frac{p-p_0}{\alpha}\right)\right) \quad (12b)$$

08/09/2020

Bragg-Kleeman rule

Range of α-particles in other media

The effective stopping power of materials is approximately inversely proportional to $\sqrt{A}$ (Where A is atomic mass, and proportional to the density of the material.

$$\frac{R_1}{R_0} = \frac{\rho_0}{\rho_1}\frac{\sqrt{A_1}}{\sqrt{A_0}} \qquad (13)$$

Comparing materials 1 and 0 (reference material)

R is range, ρ is material density

for air: $\sqrt{A_{air}} = 3.82$

We can use (13) to predict the energy loss for α-particles passing through materials.

Then we replace ρ for $\rho_1 \frac{\sqrt{A_{air}}}{\sqrt{A_1}}$

$$\therefore E = \left(E_0^{1.73} - 4.464\ \rho_1 \frac{\sqrt{A_{air}}}{\sqrt{A_1}} \Delta x\right)^{0.578} \quad \text{(MeV)}$$

Is the Energy of α-particles after passing through material 1

09/09/2020

How the detection works

When an incident charged particle enters the detector, electron-hole pairs are created as the particle is decelerated.
The charges are drawn to the appropriate surface contact, creating a charge pulse that is proportional to the energy of the charged particle.

The pulse amplifier is a charge-to-voltage converter provides an output voltage proportional to the charge produced by the input photodiode.

The multichannel analyser detects voltage pulses & records their heights in a series of internal bins producing a histogram of pulse heights

Pulse range: 0-5 V
The voltage range is divided into 1024 bins (equal dimension)

The set up

our settings allow the acquisition of data pulses that have a pulse height greater than 10 bins for 300 s in 1024 bins.
MCA will only record a new pulse after the input voltage drops below that threshold level (10)

Safety Hazards

09/09/2020

- Source chamber – do not interfere with it

Note: Electronics must come to thermal eqilibrium for at least 20 minutes before taking a run of measurements.

Method

*Adjusting the pressure in the chamber procedure

- Close "air admittance" valve
- Turn "on" Vacuum gauge.
- Activate pump
- Open valve marked "Vacuum pump line"
- When pressure is below 0.05 kpa wait approximately 1 minute and proceed to
- Close "Vacuum pump line"
- Deactivate pump.

The chamber pressure is controlled by admitting air into the chamber by using the "air admit" valve

If required repeat the steps outlined above to partially evacuate the chamber. *

α-particle detection.

- Turn on power switch of "NIM bin"
- Turn on MCA unit and DSO amplifier units
- Evacuate vacaum chamber as per procedure outlined above.
- ensure that amplifier output is connected to DSO (particle pulses: > 2V)
- ensure amplifier output is connected to MCA unit

MCA: (multichanel amplifier)

MCA software setup

09/09/2020

1. Launch "ADMCA"
   - accept default device (MCA 8000A) (Click Connect)
2. in menu:
   - MCA – "Acquisition setup"
     - ensure preset mode is "seconds"
     - ensure timer is set to "Live time"
     - preset time t=300s
     - "threshold" is set to 10
     - set "ADC channels" to 1024
     - click "Ok" to preserve settings

MCA will only record a new pulse after the input voltage drops below that threshold level (10)

Note: that Threshold value might have to be readjusted

3. Clear any data stored in the MCA memory and RESET

   Menu: MCA – "Delete data and Reset time"

4. Acquiring new data:

   Menu: MCA – "Start acquisition"

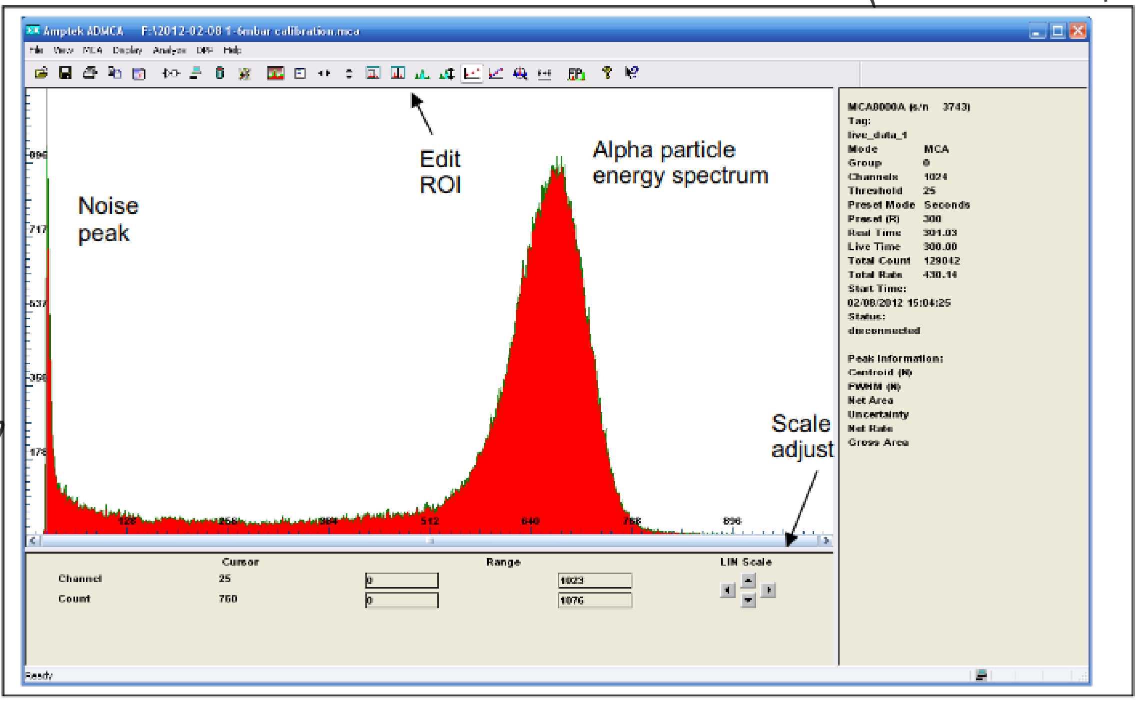
Things to know:

* The energy spectrum is updated every few seconds
* "Display – Scale" can be modified so to get the data on a linear or a logarithmic axis (Analysis should be done using a linear vertical scale)

09/09/2020

from ref: [0]

A typical screen shot appears below:



~~Fig 3:~~ Fig 7. Typical energy spectrum of the alpha particle source with the chamber partially evacuated

5. Verify that the highest energies in the spectrum are within the MCA window (use scale arrows to change the visible scale)

The height of the noise peak depends on the threshold

If threshold setting is low
Noise peak > energy peak.

Noise peak should ideally be similar amplitude to that of the general Background

* A large number of counts under the noise peak will cause errors in determining the α-spectrum when pressure in chamber is high and a large number of particles are stopped by air.

09/09/2020

Analysis (in Situ)

1. Defining ROI

- click "edit ROI" on toolbar
- drag cursor over region containing the peak
- exit by clicking "edit ROI"
- save spectra using → "File" — "Save as"

The file format: ASCII
can be read in python

We are assuming ROI is symmetric and will fit a gaussian

As pressure increases peak moves towards lower energy & becomes asymmetric

∴ we need to manually estimate the position of the peak

Gross area measure is the total counts within the ROI.

Zero energy of Detector (Background)

- output for zero signal ≠ 0 V
- The first bin of MCA spectrum may not correspond to zero input signal
- filling vacuum chamber with air will aid in the accurate determination of the detector's baseline signal

MCA detection range is 0-5V. Assuming oscilloscope is accurate allows us to convert measured voltage to a MCA bin.

Note: correct data by considering this possible offset.

Data collection

09/09/2020

1. Record energy spectra of α-particles
   Pressure range at least of: 15 values
   with 100s per setting

* Ensure threshold set to cut out majority of noise peak

2. For each pressure setting
   - measure area, width and estimate position of peak
   - save the recorded spectrum

Note: particles with energies below the threshold ARE NOT RECORDED. At high pressure when the spectrum is truncated by the threshold your measurement of the number of surviving particles is in error

You can correct this by extrapolation

Preliminary questions

1. WHY is the pressure varied and not the distance between the source and detector?

Both approaches are equivalent since α-particles lose energy when interacting with particles in the medium. Changing the distance between source and detector would require a much larger control chamber for the alpha particles to travel through. This is likely difficult to implement.

2. For a constant aquisition time, what effect does changing the number of channels on the MCA have on the S/N ratio and other parameters?

The spectrum becomes more or less refined when the number of channels is changed

Decreasing channels will make the spectrum smoother which makes the peak parameters like FWHM and uncertainty increase

The S/N ratio will be larger also.

09/09/2020

3. How does the width of energy peaks compare to the energy resolution of the detector?

Resolution ~ 12 keV noise amp system < 25 keV
Can you account for the difference?

If the width of the peaks was infinitesimaly narrow we would still observe "gaussian" peaks due to the resolution of the detector and the amp sys.
The sum in quadrature of uncertainty in resolution & acc.

$$\sqrt{(12\,keV)^2 + (25\,keV)^2} = 27\,KeV$$

is less than the actual observed FWHM of the peaks,

Some peaks have width approximately ~200 bins
in task 2 we found that the conversion factor is ~5 keV

$$200 \times 5\,keV = 1\,MeV$$

The FWHM is larger than the uncertainty

Quick notes from video

09/09/2020

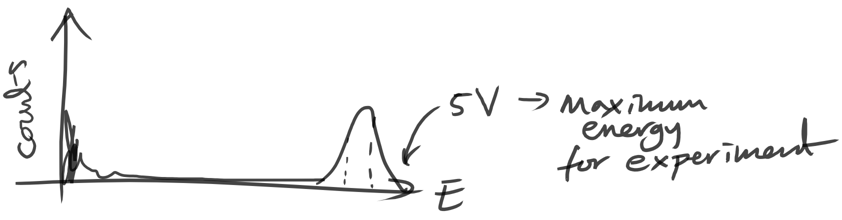


Fig 8. Quick sketch of energy spectrum

We are given
spectrum as a function of pressure

α { Range, Energy, FWHM } change with pressure.

Height in signal proportional energy of particles.

Threshold set up

Oscilloscope signal must be checked
low-voltage region
⇒ Low S/N ratio

* altering the scale of the window provides us with a view of the noise floor

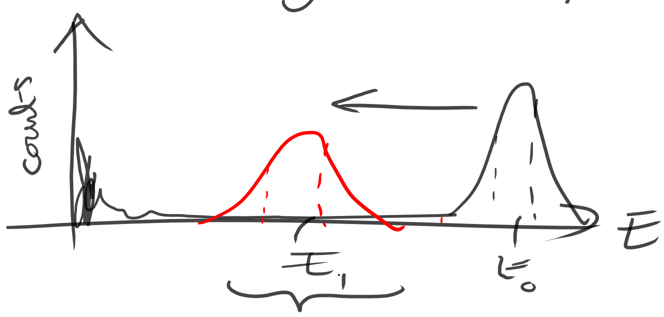
Calibrating energy scale requires that we use this noise energy value as the zero-energy threshold

Noise is likely gaussian distributed
the center must be close to the true mean background

— Cutting noise out —

from having observed the noise count in the correct scale we can re-set threshold in video: threshold ~40

. Proceed to start experimental runs for 300s
increasing chamber pressure



range is decreased
straggling parameter
increases FWHM

Fig 9. Quick sketch of 2 energy peaks from different pressures
ie. red peak is higher pressure
⇒ less energy

09/09/2020

Information from data acquired.

File name indicates pressure that each energy Spectrum is recorded at.
(ie. 33.05 kPa = 33.05mbar.mca)
resolution is ±0.01 kPa
accuracy is 0.25%
Laboratory temperature: 21.0±0.5 °C

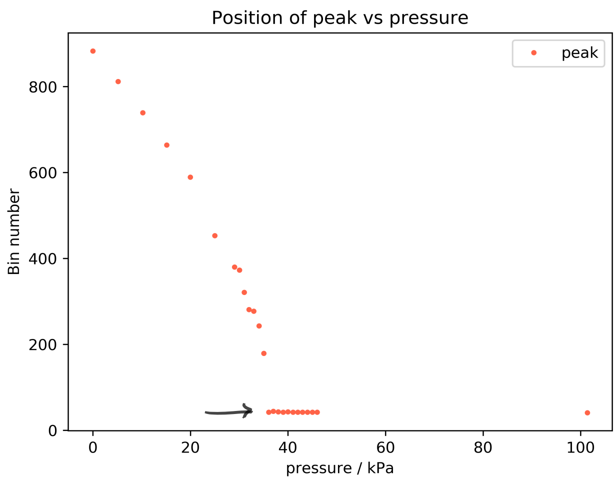
Data files from 0.00 – 101.39 kPa are provided

MCA threshold = 42
acquisition time = 100s (live time)

Background threshold = 4 → noise floor of detector/amp system

— Analysis tasks —

1. Determine the pressure corresponding to the absorption of α-particles with the most probable energy
Where is the peak at zero energy?



$\therefore P_0 = 36.1 \pm 0.5\,kPa$

Fig 11(a) peak position as a function of pressure
centre peak at approximately 36.06

$$u(P_0) = \frac{(36.06 - 35.05)}{2}\,kPa$$

$$\therefore u(P_0) = 0.5\,kPa$$

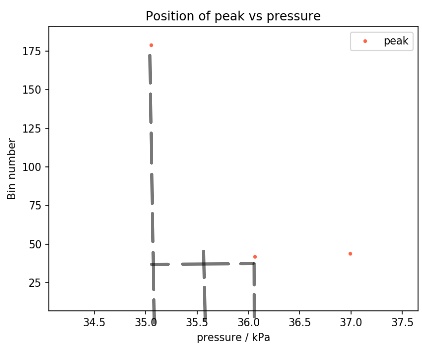


fig 11(b) Value of uncertainty must be $\frac{1}{2}$ the distance between 35.05 and 36.06 kPa

Note that for other (P) measurements

$$u(P) = \sqrt{u(resolution)^2 + u(accuracy)^2} = \sqrt{(0.0025P)^2 + (0.01)^2}$$

10/09/2020

2. We proceed to determine the energy of the α-particles emitted from the source.

R is range, p is air pressure, from equation (5)

let: $R_0 = x_0$

$$R_0 = 64.31 \frac{T}{p} E_0^{1.73}$$

$$\therefore E_0 = \sqrt[1.73]{\frac{R_0 p}{64.31 T}} = 4.45 \text{ MeV}$$

for simplicity:

let $S = \frac{Rp}{64.31T}$, $u(S) = \sqrt{\left(\frac{u(R)}{R}\right)^2 + \left(\frac{u(p)}{p}\right)^2 + \left(\frac{u(T)}{T}\right)^2} \cdot S$

Then: $\frac{u(E_0)}{E_0} = \frac{1}{1.73}\left(\frac{u(S)}{S}\right) = 0.05 \text{ MeV}$

$\therefore E_0 = 4.45 \pm 0.05$ MeV

We now compare the value we found to the known energy of the α-particles emitted from ^{241}Am

$E_\alpha = 5.4857$ MeV vs $\therefore E_0 = 4.45 \pm 0.05$ MeV

The difference between these values is

1.03 ± 0.05 MeV

likely due to energy loss due to the gold layers of the source construction.

10/09/2020

We must determine the calibration factor for the MCA in terms of the energy:

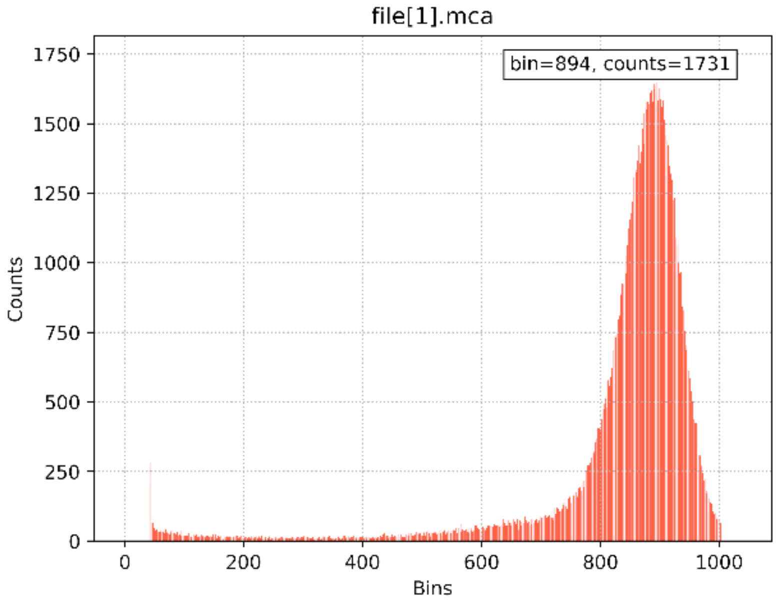


Fig. 10. Spectrum obtained from file "δ_00mbar.mca". $p = 0.00$ kPa

$E_\delta = 4.45 \pm 0.05$ MeV corresponds to Bin 894

Then:

894 bins = 4.45 ± 0.04 MeV

$\therefore$ 1 bin $= \frac{4.45 \text{ MeV}}{894} = 4.97$ keV

with uncertainty:

$$u(1\text{bin}) = \left(\frac{0.05}{4.45}\right) \times 4.97 = 0.05$$

$\therefore$ 1 bin $= 4.97 \pm 0.05$ keV

3. Calculate the anticipated range (R) for α-particles passing through a thick gold sample.

As per section 7. equation (13)

$$R_{Au} = R_{atm} \frac{\rho_{atm}}{\rho_{Au}} \frac{\sqrt{A_{Au}}}{\sqrt{A_{air}}},$$

where: $R_{atm} = 0.186 E_\delta^{1.73}$ Theoretical

$\therefore R_{atm} = 2.46 \pm 0.04$ cm

$\sqrt{A_{air}} = 3.82$

at 20°C, 1 atm: $\rho_{atm} = 1.284 \text{ kgm}^{-3}$

$\sqrt{A_{Au}} = 14.03$

$\rho_{atm} = 1.284 \text{ kgm}^{-3}$

Then

$$\frac{u(R_{Au})}{R_{Au}} = \sqrt{\left(\frac{u(R_{atm})}{R_{atm}}\right)^2 + \cancel{\left(\frac{u(\rho_{atm})}{\rho_{atm}}\right)^2} + \cancel{\left(\frac{u(\rho_{Au})}{\rho_{Au}}\right)^2} + \cancel{\left(\frac{1}{2}\frac{u(A_{atm})}{A_{atm}}\right)^2} + \cancel{\left(\frac{1}{2}\frac{u(A_{Au})}{A_{Au}}\right)^2}}$$

uncertainty contribution from crossed terms is negligible

$$\therefore \frac{u(R_{Au})}{R_{Au}} = \frac{u(R_{atm})}{R_{atm}}$$

$\therefore R_{Au} = (6.02 \pm 0.09) \times 10^{-4}$ cm

10/09/2020

Determine the thickness of the source's gold coating and compare your result to the manufacturer's value. (5×10^{-6} cm)

from equation (14)

$$E = \left(E_0^{1.73} - 4.464\,\rho_{Au} \frac{\sqrt{A_{air}}}{\sqrt{A_{Au}}} \Delta x \right)^{0.578}$$

if $E = 5.4857$ MeV

we rearrange it.

$$\Delta x = \frac{\sqrt{A_{Au}}}{-4.464\,\rho_{Au}\sqrt{A_{air}}} \left(\sqrt[0.578]{E} - E_0^{1.73} \right) \quad [cm]$$

$\Delta x = 2.5 \times 10^{-4}$ cm this is much larger than expected

what is the uncertainty ?

uncertainty contribution from crossed terms is negligible

$$u(\Delta x) = \sqrt{\cancel{\left(\frac{1}{2}\frac{u(A_{Au})}{A_{Au}}\right)^2} + \cancel{\left(\frac{1}{2}\frac{u(A_{Air})}{A_{Air}}\right)^2} + \cancel{\left(\frac{u(\rho_A)}{\rho_A}\right)^2} + \cancel{\left(\frac{1}{0.578}\frac{u(E)}{E}\right)^2} + \left(1.73\frac{u(E_0)}{E_0}\right)^2 +}$$

$$u(\Delta x) = 1.73\left(\frac{u(E_0)}{E_0}\right) = 1.5 \times 10^{-2}\ cm$$

Then

$$\Delta x = (0.025 \pm 1.5) \times 10^{-2}\ cm$$

The uncertainty is also large though

I have no clue!

IDea: maybe E varies ?

will try again ASAP

4. Produce a theoretical plot of: 11/09/2020
range vs Energy α-particles in air
(atmospheric pressure for energies E_0: 0 - 6 MeV)

the theoretical range is

$$R_{atm} = 0.186 E_0^{1.73} \text{ (cm)} \qquad (4)$$

Calculate the range in air at atmospheric pressure for this experiment and include this point in your graph.

$$R_{atm} = 2.47 \pm 0.04 \text{ cm}$$

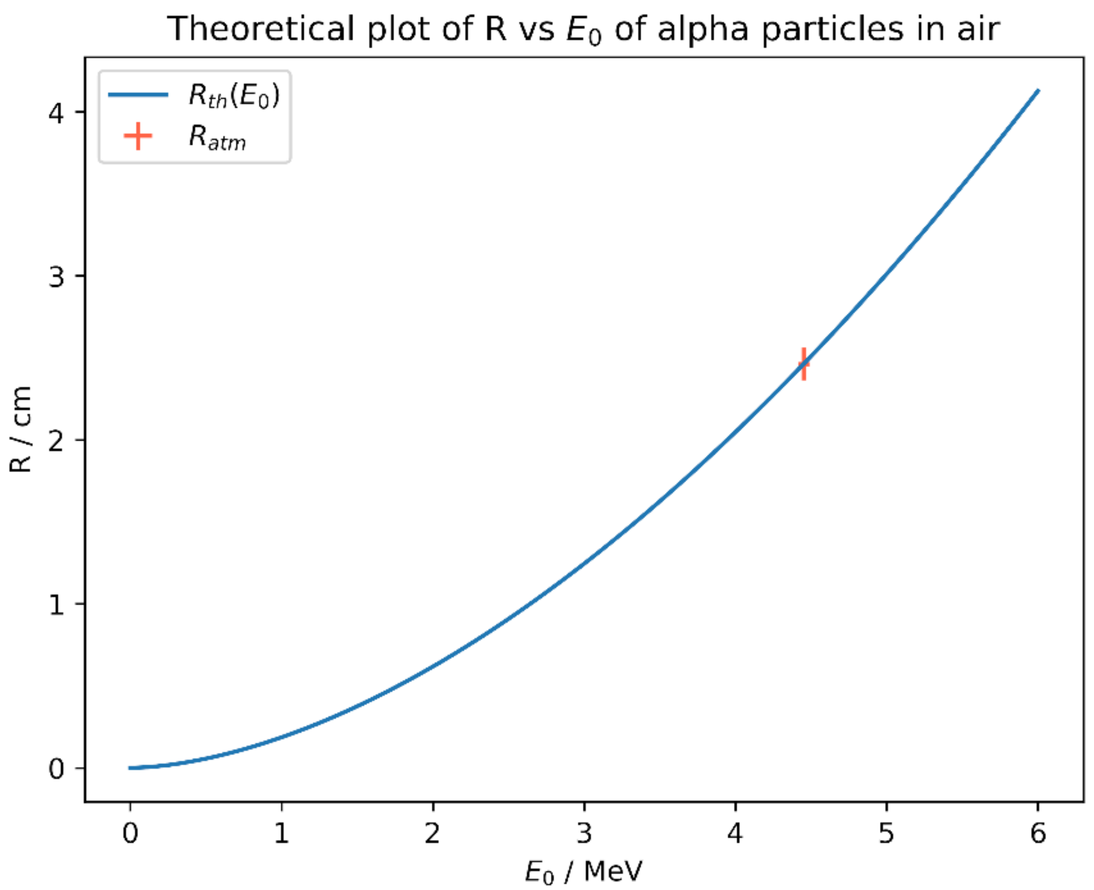


fig 11. Theoretical relation: Initial particle energy and Range. We included The calculated value Ratm.

12/09/2020

5. Use your data to produce a plot of the width of the energy spectra as a function of pressure. Comment on the observed behaviour.

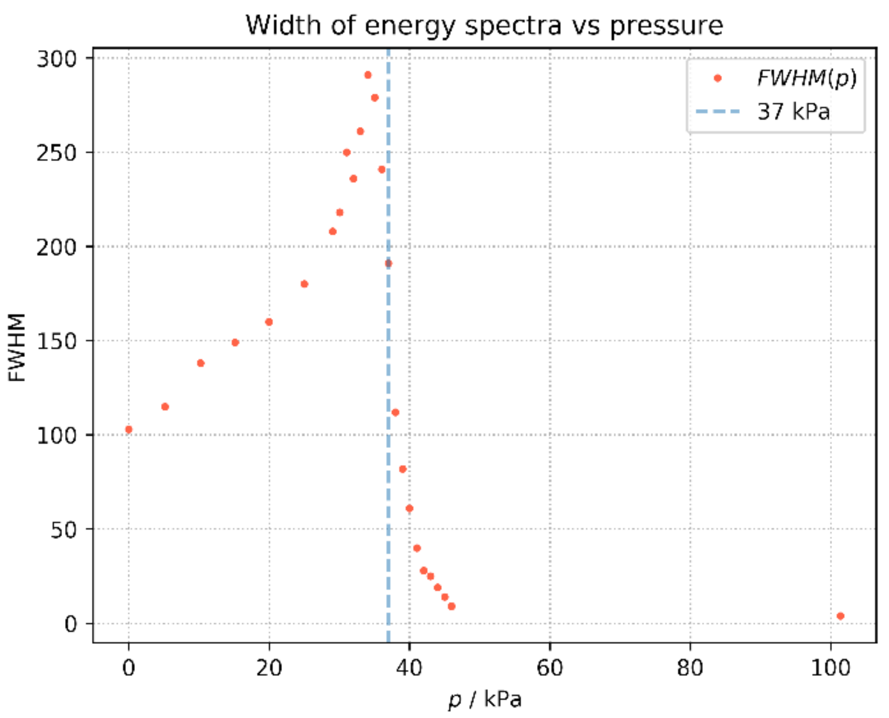


fig 12. FWHM increases with pressure up to the value ~ p = 37 kPa from then onwards we cannot assume our results for FWHM is accurate.

6. Produce a plot of the number of surviving particles as a function of pressure

Firstly, for each of our datasets, we must do a shape extrapolation of the peak.

we choose to take an average of the first * 50 values present in each data set. *

This is done to get a representative curve (that accounts for the shape and the noise of our data)

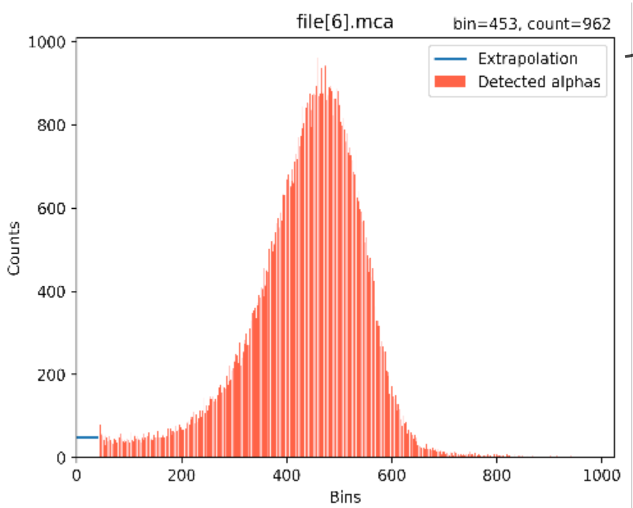


fig. 13 Spectrum for p = 25.01 kPa extrapolation is drawn in blue.

12/09/2020

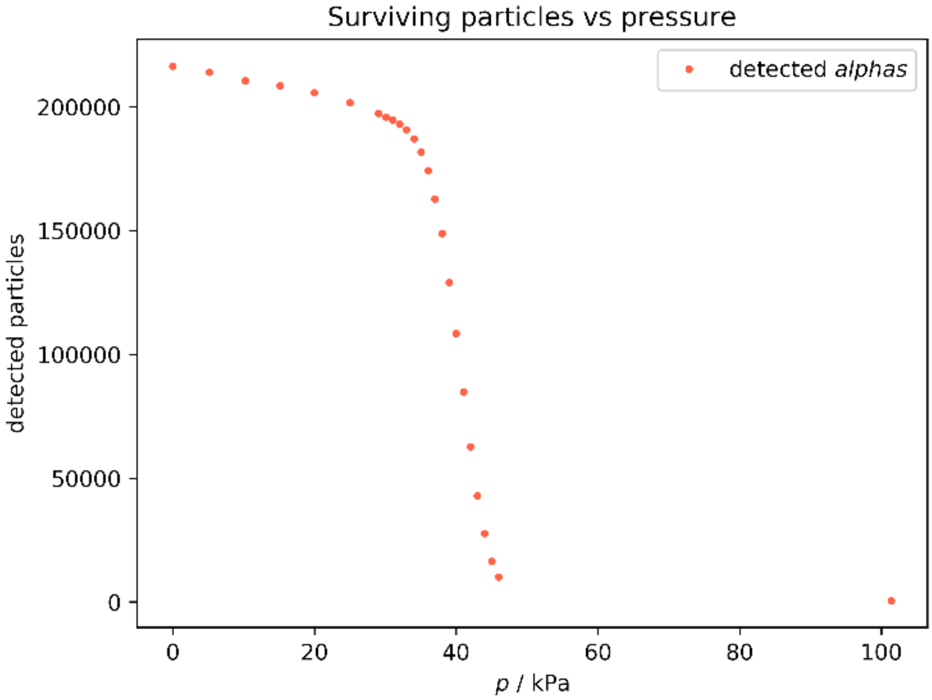


Fig 14. Range straggling effect.
Here we have offset our "surviving" particle count by the extrapolation we mentioned in the previous page

7. Determine α (straggling parameter) from (12 b)

$$\therefore \frac{n(p)}{n_0} = \frac{1}{2}\left(1 - \operatorname{erf}\left(\frac{p-R}{\alpha}\right)\right)$$

The value of α is expected to lie within $0.0015R$

After performing a fit to the points shown in fig 14 we must convert our straggling parameter to distance units.

```python
def f(p_values, p_R, α):
    # model for optimize.curve_fit()
    return (1 / 2) * (1 - special.erf((p_values - p_R) / α))

def Task_7_8(f, x, y):
    p_0 = p_zero / kilo # [kPa]
    u_p_0 = u_p_zero / kilo
    # print(x)
    # print(f"{p_0}")

    # determining straggling parameter α
    popt, pcov = scipy.optimize.curve_fit(f, x, y)
    # To compute one standard deviation errors on parameter α
    perr = np.sqrt(np.diag(pcov))

    p_R, pα = popt
    u_p_R, u_pα = perr

    p = np.linspace(x[0], x[-1], 25)
    optimal_fit = f(p, p_R, pα)

    plt.plot(
            x, y, marker='o', linestyle='None', markersize=2.5, color='tomato',
            label=r"detected $alphas$"
    )
    plt.plot(
            p, optimal_fit, marker="None",
            linestyle="-",
            label="fit"
    )

    plt.grid(linestyle=':')
    plt.xlabel(r'p / kPa')
    plt.ylabel('Detected particles')
    plt.title(r'Number alpha particles as a function of pressure')
    plt.legend()
    spa.savefig(f'alphas_vs_pressure.png')
    # plt.show()
    return p_R, pα, u_p_R, u_pα, p_0, u_p_0
```

13/09/2020

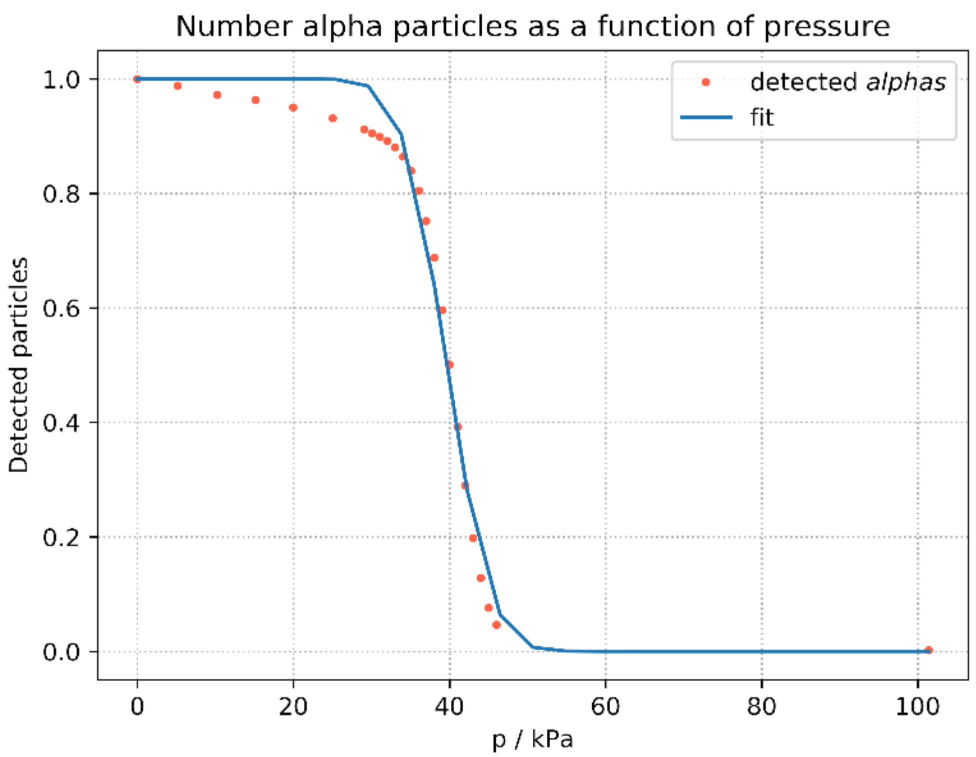


fig 15. Normalised number of particles reaching the detector as a function of chamber pressure. Blue graph is the non-linear fit. The fit is not very good.

to convert our α to distance units, we use $\frac{P_\alpha}{P_R} = \frac{x_\alpha}{x_R}$.

propagating uncertainty: $u(x_\alpha) = x_\alpha \sqrt{\left(\frac{u(P_\alpha)}{P_\alpha}\right)^2 + \left(\frac{u(P_R)}{P_R}\right)^2 + \left(\frac{u(x_R)}{x_R}\right)^2}$

we found

```
pa = 6.4 ± 0.4 kPa
```

we know $P_R = 1\,atm$, $X_R = R_{atm}$

So we will use these in our conversion

```
EXPECTED RESULT a ≅ k * R_atm = 0.0369 ± 0.0005 cm
Experimental a = 0.15 ± 0.01 cm
number of σ away from true result: 245.421
```

we are pretty far off. we either made a mistake or the gold layer appears to greately increase the range struggling parameter.

13/09/2020

8. from our fit we obtain a value for the range pressure.

```
Previous p_0: 36.10 ± 0.50 kPa
fit p_0 39.65 ± 0.18 kPa
number of σ away from true result: 19.816
ana@ana-XPS-13-9343:~/Documents/uni/PHS3000/code$
```

I think the fit p_0 is more accurate given that the method I used to "find" p_0 was simply to interpolate between data points without any computation.

Discussion

14/09/2020

We were unable to determine the thickness of the gold coating of the source to a matching accuracy to the manufacturer's reported value. This was likely due to us misunderstanding the role of the energy as a variable parameter, rather than a constant. Unfortunately we ran out of time in order to pursue alternative paths of enquiry.

Our determined value for the straggling parameter is very different from the theoretical prediction

Hypothesis: The result we found takes into account the interaction of alpha particles with the gold layers of the source.

Whilst the theoretical value may not be representing the exact type of construction that we encountered in our apparatus.

Conclusion

14/09/2020

We measured the straggling parameter of alpha particles traveling in a pressure controled chamber and we found this value $\alpha = 0.15 \pm 0.1$ cm to be highly discrepant with the prediction $\alpha_{th} = 0.0369 \pm 0.0005$ cm.

Logbook 2: References

[0] Monash SPA (2020) 1.2: Range and Energy Loss of α-particles in Air. Retrieved from Moodle (student site)

[1] Wikipedia (2020). Alpha decay. Retrieved from https://en.wikipedia.org/wiki/Alpha_decay

[2]Wikipedia (2020). Americium-241. Retrieved from https://en.wikipedia.org/wiki/Americium-241

[3]Wikipedia (2020). Gold. Retrieved from https://en.wikipedia.org/wiki/Gold

09/14/20 04:14:04 /home/ana/Documents/uni/PHS3000/code/alpha_particles.py

```
# PHS3000
# alpha particles - Range and energy loss
# Ana Fabela, 09/09/2020
import os
from pathlib import Path
import monashspa.PHS3000 as spa
import numpy as np
import matplotlib.pyplot as plt
from scipy import special
import scipy.optimize

plt.rcParams['figure.dpi'] = 150

folder = Path('spectra')
os.makedirs(folder, exist_ok=True)

# Global prefixes and values SI units
x = np.linspace(0,1036,1024) # bins array

kilo = 1e3
cm = 1e-2 # [m]
g = 1e-3 # [kg]

eV = 1.602e-19 # [J]
MeV = eV * 1e6 # [J]
keV = eV * 1e3 # [J]

p_zero = 36.1 * kilo # [Pa]
u_p_zero = 0.5 * kilo

A_air = 14.5924
A_Au = 196.966570

rho_atm = 1.284 # [kg m**-3]
rho_gold = 19.30 * (g / cm**3) # [kg/ m3]

T = 294.15 # [K]
u_T = 0.5 # [K]

x_0  = 6.77 # [cm]
u_x_0 = 0.1 # [cm]

alpha_energy = 5.4857 # [MeV]


def read_files():
    read_folder = Path('log2')
    files = list(os.listdir(path=read_folder))
    data_files = []
    p_values = []

    files.sort(key=lambda name: int(name.split('_')[0]) if name[0].isdigit() else -1)
    for i, file in enumerate(files):
        # print(i, file)
        if i >= 1:
            p_value = float(file.rstrip('mbar.mca').replace('_', '.'))
            p_values.append(p_value)
        header, data = spa.read_mca_file(read_folder/file)
        data_files.append(data)
    p_values = np.asarray(p_values)
    u_p = np.sqrt((0.0025 * p_values)**2 + 0.01**2) # resolution and accuracy [kPa]
    return p_values, u_p, data_files

def calibrate_axis(x, E_0, u_E_0):
    # calibration factor for x-axis
    _1bin = 5 * keV # [J]
    u_1bin = (u_E_0 / E_0) * _1bin
    # therefore x -> E
    E = x * _1bin # [J]
    u_E = (u_1bin / _1bin) * E
    return E, u_E

def plot_data(x, y, i, average, ax=None):
    xmax = np.argmax(y)
```

```python
    ymax = y.max()
    half_y_max = ymax / 2
    L = np.argmin((y[:xmax] - half_y_max)**2)
    R = np.argmin((y[xmax:] - half_y_max)**2) + xmax
    peak_width = R - L

    # if i >= 1:
    #     text= "bin={:.0f}, count={:.0f}".format(xmax, ymax)
    #     if not ax:
    #         ax=plt.gca()
    #     ax.annotate(text, xy=(xmax, ymax), xytext=(0.7, 1.02), textcoords='axes fraction')

    # plt.bar(x, y, color='tomato',label="Detected alphas")
    # plt.plot(x[:41], average, label="Extrapolation") # extrapolation cuve
    # # plt.axhline(y=half_y_max, linestyle=':', alpha=0.3, label="Half max")
    # # plt.fill_betweenx([0, ymax + 20], [L, L], [R, R], alpha=0.3, zorder=10)
    # plt.xlim([0, 1024])
    # plt.xlabel('Bins')
    # plt.ylabel('Counts')
    # plt.title(f'file[{i}].mca')
    # plt.legend()

    # spa.savefig(folder/f'file{i}.png')
    # plt.show()
    # plt.clf()
    return xmax, ymax, peak_width

def energy_peaks(p_values, data_files):
    max_counts = []
    max_positions = []
    peak_widths = []
    total_events = []

    for i, signal in enumerate(data_files):
        # curve extrapolation for the threshold region
        average_list = [np.mean(signal[42:92])] * 41
        # sum of all events including extrapolation
        total = np.around(np.sum(signal) +  np.sum(average_list), decimals=0)
        total_events.append(total)

        # print(f"{np.sum(signal)} + {np.sum(average_list)} = {total}")

        # barcharts to visualise our files
        xmax, countmax, peak_width = plot_data(x, signal, i, average_list)

        max_positions.append(xmax)
        max_counts.append(countmax)
        peak_widths.append(peak_width)
    peak = max_positions[1]
    return peak, signal, total_events, max_positions, max_counts, peak_widths


def plot_pressure_vs_energy(p_values, max_positions):
    # pressure vs Energy peak
    plt.plot(p_values, max_positions[1:],'o', color='tomato', markersize=2.5, label="peak")
    plt.xlabel('pressure / kPa')
    plt.ylabel('Bin number')
    plt.title(f' Position of peak vs pressure ')
    plt.legend()
    spa.savefig(f'peak_position_vs_pressure.png')
    plt.show()

def Task_2(x_0, u_x_0):
    # Calculating E_0
    # from equation (5)
    R_0 = x_0
    u_R_0 = u_x_0
    # print(f"\n{T = :.2f} K, R = {R_0} cm, p = {p_zero / kilo} kPa")
    E_0 = (R_0 * p_zero / (64.31 * T))**(1 / 1.73)
    u_E_0 = (E_0 / 1.73) * np.sqrt((u_R_0 / R_0)**2 + (u_p_zero / p_zero)**2 + (u_T / T)**2)
    print(f"\nE_0 = {E_0:.2f} ± {u_E_0:.2f} MeV")
    # # difference in energy
    diff_range_E = alpha_energy - E_0
    u_diff_range_E = u_E_0
    print(f"\n{diff_range_E = :.2f} ± {u_diff_range_E:.2f}")
    return R_0, u_R_0, E_0, u_E_0

```

```python
def Task_3(E_0, u_E_0, rho_atm, rho_gold, A_Au, A_air, alpha_energy):
    # Calculating R_Au:
    # equation (4)
    R_atm = 0.186 * E_0**1.73
    u_R_atm = R_atm * 1.73* u_E_0 / E_0
    # print(f"\nTheoretical Range in 1 atm: {R_atm = :.2f} ± {u_R_atm:.2f} cm")

    # Calculating anticipated range for particles travelling through gold
    # equation (13)
    R_Au = R_atm * (rho_atm / rho_gold) * np.sqrt(A_Au / A_air)
    u_R_Au = R_Au * (u_R_atm / R_atm)
    # print(f"{R_Au = } ± {u_R_Au} cm")

    # Calculating the thickness (delta_x) of the gold coating
    # rearranged equation (14)
    delta_x = np.sqrt(A_Au) * (E_0**1.73 -alpha_energy**(1/0.578))  / (4.464 * rho_gold * np.sqrt(A_air))
    u_delta_x = 1.73 * u_E_0 / E_0
    # print(f"\n{delta_x = } ± {u_delta_x} cm")
    return R_atm, u_R_atm, R_Au, u_R_Au, delta_x, u_delta_x


def Task_4(E_0, u_E_0, R_atm, u_R_atm):
    # theoretical plot of range vs energy of alpha particles in air at atm pressure
    x, y = [E_0], [R_atm]
    E_th =  np.linspace(0,6,1024) # (0 - 6) MeV array
    R_th = 0.186 * (E_th**1.73)

    plt.plot(
        E_th, R_th, marker="None",
        linestyle="-", label=r"$R_{th}(E_0)$"
        )
    plt.errorbar(
            x, y, xerr=u_E_0, yerr=u_R_0,
            marker="None", linestyle="None", ecolor="tomato",
            label=r'$R_{atm}$', color="tomato", barsabove=True
        )

    plt.xlabel(r'$E_0$ / MeV')
    plt.ylabel('R / cm')
    plt.title(r'Theoretical plot of R vs $E_0$ of alpha particles in air')
    plt.legend()
    spa.savefig(f'theoretical_R_vs_E.png')
    plt.show()

def Task_5(peak_widths, p_values):
    # plot of FWHM vs pressure
    plt.plot(
        p_values, peak_widths[1:], 'o', color='tomato', markersize=2.5, label=r"$FWHM(p)$"
        )
    plt.axvline(x=p_zero/kilo, linestyle='--', alpha=0.5, label="37 kPa" )
    plt.grid(linestyle=':')
    plt.xlabel(r'$p$ / kPa')
    plt.ylabel('FWHM')
    plt.title(r'Width of energy spectra vs pressure')
    plt.legend()
    spa.savefig(f'FWHM_vs_pressure.png')
    plt.show()

def Task_6(total_events):
    # The effect  of  range  straggling.
    # How does the number of surviving particles vary with pressure.

    plt.plot(
        p_values, total_events[1:], 'o', markersize=2.5, color='tomato', label=r"detected $alphas$"
        )
    # plt.grid(linestyle=':')
    plt.xlabel(r'$p$ / kPa')
    plt.ylabel('detected particles')
    plt.title(r'Surviving particles vs pressure')
    plt.legend()
    spa.savefig(f'alphas_vs_pressure.png')
    plt.show()


def f(p_values, p_R, α):
    # model for optimize.curve_fit()
    return (1 / 2) * (1 - special.erf((p_values - p_R) / α))
```

```python

def Task_7_8(f, x, y):
    p_0 = p_zero / kilo # [kPa]
    u_p_0 = u_p_zero / kilo
    # print(x)
    # print(f"{p_0}")

    # determining straggling parameter α
    popt, pcov = scipy.optimize.curve_fit(f, x, y)
    # To compute one standard deviation errors on parameter α
    perr = np.sqrt(np.diag(pcov))

    p_R, pα = popt
    u_p_R, u_pα = perr

    p = np.linspace(x[0], x[-1], 25)
    optimal_fit = f(p, p_R, pα)

    plt.plot(
            x, y, marker='o', linestyle='None', markersize=2.5, color='tomato',
            label=r"detected $alphas$"
    )
    plt.plot(
            p, optimal_fit, marker="None",
            linestyle="-",
            label="fit"
    )

    plt.grid(linestyle=':')
    plt.xlabel(r'p / kPa')
    plt.ylabel('Detected particles')
    plt.title(r'Number alpha particles as a function of pressure')
    plt.legend()
    spa.savefig(f'alphas_vs_pressure.png')
    # plt.show()
    return p_R, pα, u_p_R, u_pα, p_0, u_p_0

def compare(p_R, pα, u_p_R, u_pα, p_0, u_p_0):
    # conversion to distance units
    p_atm = 101.325 # [kPa]
    u_p_atm = np.sqrt((0.0025 * p_atm)**2 + 0.01**2) # [kPa]

    xα = (pα / p_atm) * R_atm
    u_xα = xα * np.sqrt((u_pα / pα)**2 + (u_p_atm / p_atm)**2 + (u_R_atm /R_atm)**2)
    # comparison to theory
    k = 0.015
    diff_range = R_atm - xα
    how_many_sigmas = diff_range / u_xα
    print(f"\nEXPECTED RESULT α ≅ {k * R_atm = :.4f} ± {k * u_R_atm:.4f} cm")
    print(f"Experimental α = {xα:.2f} ± {u_xα:.2f} cm")
    # print(f"difference {diff_range:.3f}")
    print(f"number of σ away from true result: {abs(how_many_sigmas):.3f}")

    diff_p = p_0 - p_R
    how_many_sigmas = diff_p / u_p_R
    print(f"\nPrevious p_0: {p_0:.2f} ± {u_p_0:.2f} kPa")
    print(f"fit p_0 {p_R:.2f} ± {u_p_R:.2f} kPa")
    # print(f"difference {diff_p:.3f}")
    print(f"number of σ away from true result: {abs(how_many_sigmas):.3f}")


### * FUNCTION CALLS *###

p_values, u_p, data_files = read_files()

peak, signal, total_events, max_positions, max_counts, peak_widths = energy_peaks(p_values, data_files)

plot_pressure_vs_energy(p_values, max_positions)

R_0, u_R_0, E_0, u_E_0 = Task_2(x_0, u_x_0)

R_atm, u_R_atm, R_Au, u_R_Au, delta_x, u_delta_x = Task_3(E_0, u_E_0, rho_atm, rho_gold, A_Au, A_air,
alpha_energy)

E, u_E = calibrate_axis(x, E_0, u_E_0)

# Task_4(E_0, u_E_0, R_atm, u_R_atm)
```

```python

# Task_5(peak_widths, p_values)

# Task_6(total_events)

y = total_events[1:] / total_events[1]
p_R, pα, u_p_R, u_pα, p_0, u_p_0 = Task_7_8(f, p_values, y)
# The straggling parameter can be expressed either as a pressure or a distance
# (at 1 atm pressure) it's just proportional to the range value in the units
 # you've expressed it in.
print(f"\n{pα = :.1f} ± {u_pα:.1f} kPa")

compare(p_R, pα, u_p_R, u_pα, p_0, u_p_0)

# Which  result do you think is more accurate and why?


```