DETERMINAREA COEFICIENTULUI

DE ATENUARE MASICĂ PENTRU RADIAȚIA GAMMA

Stamatie Mihnea-Stefan
Bolfă Robert Ion
Ghiță Andrei-Iulian
324CB

Scopul lucrării

Determinarea coeficientului de atenuare masică a radiațiilor γ pentru diferite materiale în vederea verificării legii de atenuare a radiației γ în substanță.

Teoria lucrării

Radiația gamma este atenuată la trecerea printr-un strat de substanță prin absorbție și împrăștiere.

Principalele procese de atenuare sunt: efectul fotoelectric, efectul Compton și formarea de perechi.

Coeficientul de atenuare al unei substanțe este o măsură a cantității de radiație gamma absorbită sau împrăștiată de o unitate de lungime de substanță.
Intensitatea radiației gamma care a traversat un strat de grosime x este atenuată exponențial. Aceasta este descrisă de ecuatia:

$$I = I_0 e^{-\mu x}$$

unde I_0 reprezintă intensitatea fasciculului de radiații la intrarea în substanță, iar I este intensitatea fasciculului după traversarea stratului de grosime x.

Descrierea instalației experimentale

Sursa de radiații este de forma unei capsule , care conține un preparat de 60Co. Instalația de măsură este formată dintr-un detector cu scintilații așezat în fața colimatorului și conectat la un calculator. Plăcuțele de *Pb*, *Al*, și *Fe* se așază pe rând între colimator și detector, cu scopul determinării coeficientului de atenuare al fiecărui material.

Semnalul transmis de detector este înregistrat în computer sub forma unei spectrograme. Aria cuprinsă sub graficul acestei spectrograme este proporțională cu intensitatea radiației detectate și poate fi utilizată în locul lui I în formula teoretica. Aceasta va fi notata cu “A” in datele analizate.

Modul de lucru

Se inregistrează spectrograma radiației de fond și se determină intensitatea radiației emise direct de sursă pentru a determina fotopeak-uri specifice spectrului pentru 60Co.

Se repetă măsurătorile pentru fiecare dintre materialele amintite (*Pb, Fe, Al*) la grosimi de *0.5 cm, 1 cm, 1.5 cm, 2 cm, 2.5 cm*. Placuțele vor fi puse pe rând deasupra colimatorului mărind astfel treptat grosimea de atenuator. Pentru fiecare grosime se determină intensitatea radiației transmise într-un interval de timp determinat, in intervalul fotopeak-urilor determinate anterior.

Colectarea datelor

Datele colectate in urma măsurătorilor anterioare se pot reprezenta conform urmatorului tabel:

	X_0 = 0cm	X_1 = 0.5cm	X_2 = 1cm	X_3 = 1.5 cm	X_4 = 2cm	X_5 = 2.5cm
A (atenuat de Aluminiu)	3728	3457	3137	2979	2727	2513
A (atenuat de Fier)	3969	3043	2344	1845	1516	1169
A (atenuat de Plumb)	3812	2295	1425	835	491	312

Interpretarea datelor

Cu ajutorul acestor date putem calcula coeficientul de atenuare specific fiecarui material.

Pornim de la formula teoretica, in care putem inlocui *I* cu A

$$I = I_0 * e^{(-\mu * x)} \Rightarrow A = A_0 * e^{(-\mu * x)}$$

Putem sa logaritmam si obtinem:

$$\ln A = \ln A_0 - \mu * x$$

unde A si x sunt cunoscute.

Ecuatia obtinuta reprezinta o dreapta deci putem sa folosim tehnica de regresie liniara, cu aproximare in sensul celor mai mici patrate obtinand astfel panta, echivalenta in acest caz cu μ.

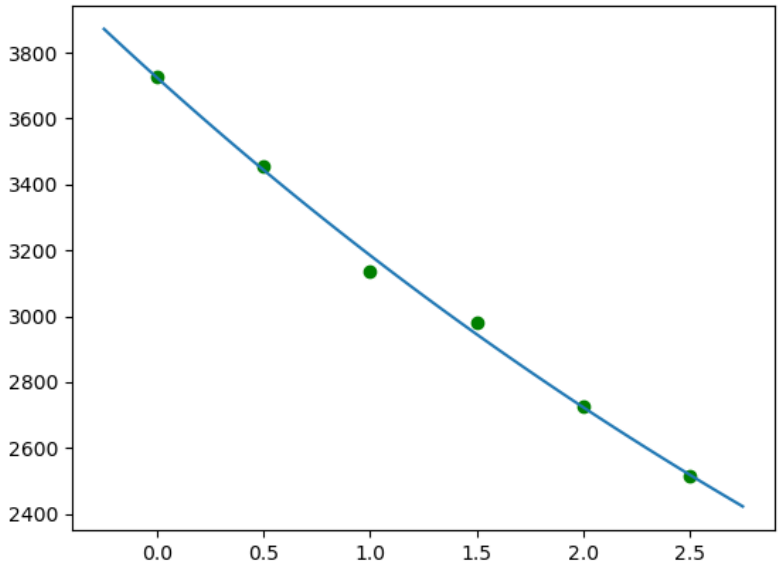
Luand in considerare datele pentru un singur material pe rand obtinem un sistem de ecuatii supradeterminat, care scris sub forma matricială va fi:

$$A*x = B \Rightarrow A^t * A * x = A^t * B \Rightarrow x = (A^t * A)^{(-1)} * A^t * B$$

Unde x este un vector coloană ce contine a si b (din ecuatia dreptei a + bx = y, pe care incercăm să o aproximăm), A va contine valorile grosimilor materialelor iar B va contine valorile masurate de detector.

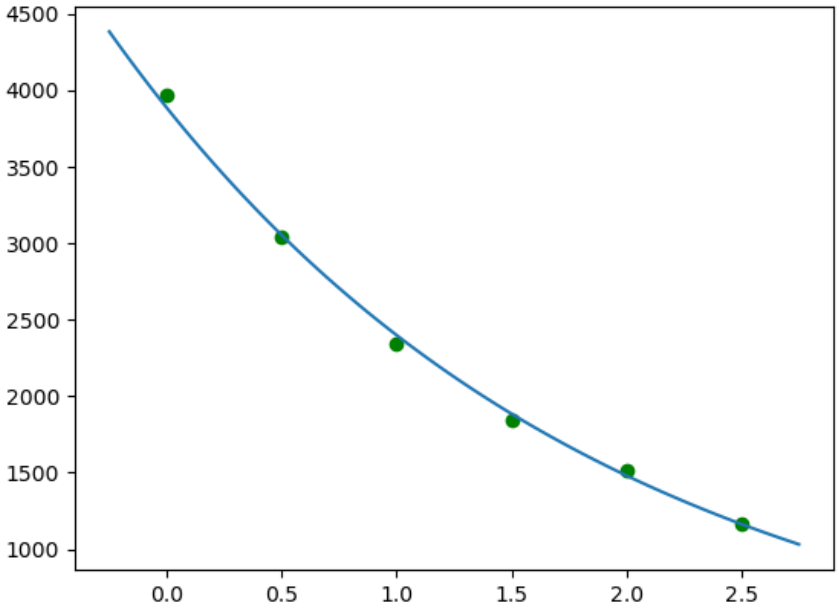
Astfel, putem efectua calculele cu ajutorul unui program realizat in Python si obtinem:

Aluminiu



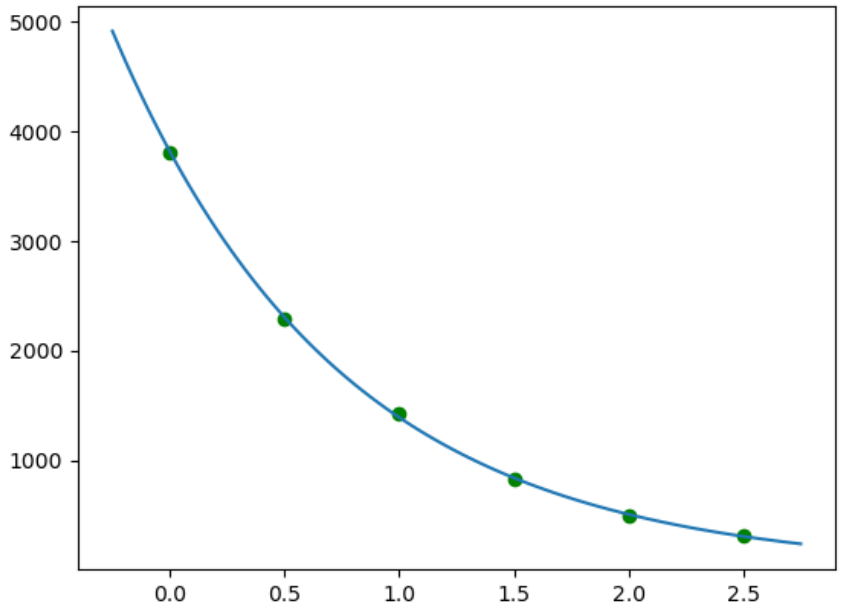
Obtinem panta = μ = 0.156

Fier



Obtinem panta = μ = 0.482

Plumb



Obtinem panta = μ = 1.010

COD PYTHON

```python
import numpy as np
import matplotlib.pyplot as plt
from simple_term_menu import TerminalMenu

exponentialMessage = "Finding coefficients (a and b) of exponential function \n y = a * exp (b * x)\n"
linearMessage = "Finding coefficients (a and b) of linear function \n y = a * x + b\n"

terminal_menu = TerminalMenu(["Linear ( y = a * x + b )", "Exponential ( y = a * exp (b * x) )"],
title="Select a function to begin function fitting:")
choice_index = terminal_menu.show()

show_plot = TerminalMenu(["No", "Yes"], title="Do you want to show the plot?").show()

match choice_index:
  case 0:
    welcomeMessage = linearMessage
  case 1:
    welcomeMessage = exponentialMessage
  case _:
    welcomeMessage = "Unexpected error"
    exit(-1)
print (welcomeMessage)

inputFile = input ("Enter file name: (default: input.txt) ")
if inputFile == "":
  inputFile = "input.txt"
while True:
  try:
    points = np.loadtxt(inputFile)
    break
  except OSError:
    inputFile = input ("\""+ inputFile +"\" not found.\n\n Enter file name: (default: input.txt) ")
  except KeyboardInterrupt:
    exit(-1)
  except Exception:
    print ("Unexpected error")
    exit(-1)

if show_plot:
  plt.plot(points[0,:], points[1,:], 'go')

match choice_index:
  case 0:
    transform = lambda x: x
  case 1:
    transform = lambda x: np.log(x)
  case _:
    print ("Unexpected error")
    exit(-1)
Y = np.atleast_2d(transform(points[1,:])).T

X = np.atleast_2d(points[0,:]).T
X = np.hstack((np.full(X.shape, 1), X))

R = np.linalg.inv(X.T @ X) @ X.T @ Y

a = R[0,0].item()
b = R[1,0].item()
print ("a = " + str(a))
print ("b = " + str(b))

start_range = points[0,:].min()
end_range = points[0,:].max()
padding = (end_range - start_range) / 10
start_range -= padding
end_range += padding
x_values = np.linspace(start_range, end_range, 100)

match choice_index:
  case 0:
    result = a + b * x_values
  case 1:
    result = np.exp(b * x_values + a)
  case _:
    print ("Unexpected error")
    exit(-1)
y_values = result

if show_plot:
  plt.plot(x_values, y_values)
  plt.show()
```