# FÍSICA COMPUTACIONAL I

## AJUSTE DE DADOS - DATA PARA ENTREGA: 28/05/2019

---

**Ajuste de dados usando o scipy.optimize**

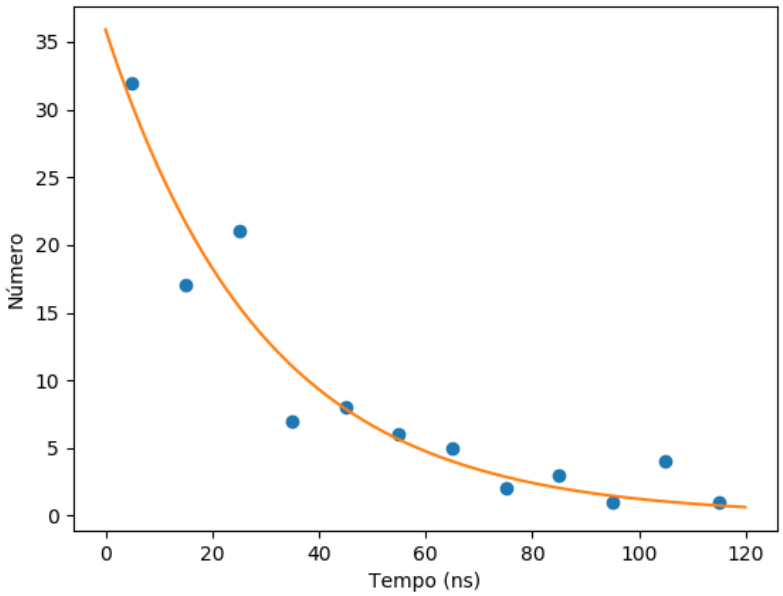
a) Importe os dados do arquivo 'decay.data' e a função curve_fit:
from scipy.optimize import curve_fit

b) Defina uma função, que será usada pelo curve_fit:
def func (x,a,b):
    return a*exp(-b*x)

c) Faça o ajuste:
param, pcov = curve_fit(func,xd,yd)
Os valores de $a$ e $b$ que minimizam o "erro" estão contidos na variável 'param'. O primeiro elemento desse vetor contém $a$ e o segundo $b$ .

d) Faça um gráfico similar ao mostrado abaixo. O modelo proposto parece se adequar aos dados experimentais? (Dica: com os valores de $a$ e $b$ obtidos via ajuste você pode definir uma função que descreve o decaimento. Forneça vários pontos para esta função para obter o gráfico).

e) Qual o valor de vida média que seu ajuste fornece? Compare com o valor tabelado para a vida média dos píons, que é 26 ns.




**Regressão linear e o método dos mínimos quadráticos**

a) Calcule o log dos valores de $y_d$ importados anteriormente (Dica: se você importar o log do numpy, a função aceitará como argumento um vetor.)

b) Faça um gráfico do z=log($y_d$) *versus* tempo. A dependência parece linear?

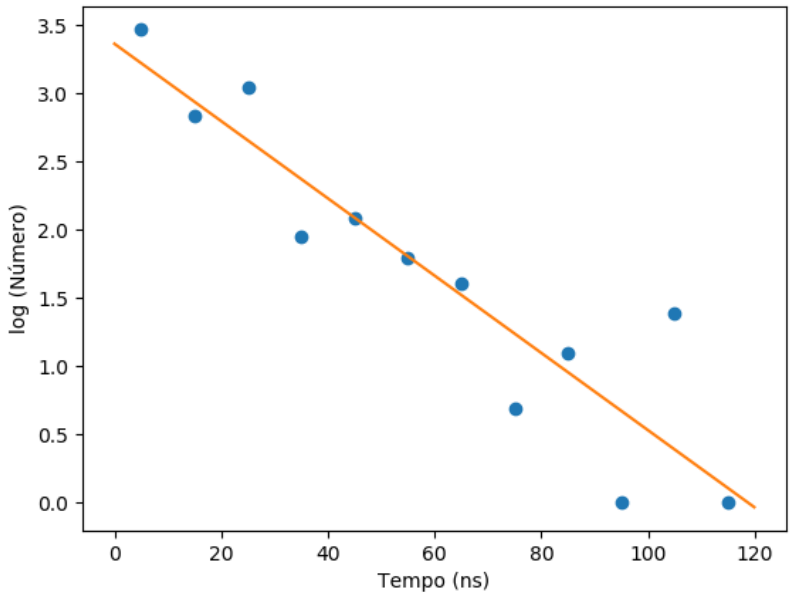
c) Encontre os parâmetros da reta ($f(x) = a + mx$) que minimiza os resíduos quadráticos, usando as fórmulas discutidas:

$$m = \frac{\sum_i x_i(z_i - \bar{z})}{\sum_i x_i(x_i - \bar{x})}$$

$$a = \bar{z} - m\bar{x}$$

(Dica: na fórmula acima $z = \log(y_d)$)

d) Faça um gráfico como o mostrado abaixo. Compare o valor obtido para a vida média com o valor tabelado.




**Ajuste de dados e erro experimental**

a) Importe os dados do arquivo 'scattering.data'.

b) De acordo com a teoria, a seção transversal para o espalhamento de nêutrons por um núcleo tem a seguinte dependência com a energia dos nêutrons:

$$f(E) = \frac{f_r}{(E - E_r)^2 + \Gamma^2/4}$$

f$f_r = \Gamma^{2}$

Despreze inicialmente o erro experimental, e repita o procedimento feito na atividade "Ajuste de dados usando o scipy.optimize" para obter valores de $f_r$, $E_r$ e $\Gamma$. O seu gráfico deve ser parecido com o mostrado abaixo. Compare os valores obtidos com $E_r = 78$ MeV e $\Gamma = 55$ MeV, que são os valores previstos pela teoria.

c) Faça um gráfico agora incluindo barras de erro. Para isto, primeiro importe a terceira coluna do arquivo de dados fornecido (que contém justamente os erros). Em seguida,

from pylab import errorbar ← importa a função
errorbar(xd,yd, yerr = erro,fmt = 'o',capsize=5) ← inclui as barras de erro.

xd, yd e erro são, respectivamente, a primeira, a segunda e a terceira coluna de dados do arquivo experimental. Você deve plotar o gráfico antes de incluir as barras de erro.

d) Os dados do arquivo foram obtidos da tabela mostrada abaixo, do livro do Rubin Landau. Olhando os valores fornecidos, o tamanho da barra de erro parece razoável? Você acha a informação fornecida pela barra relevante?

e) Altere o comando do curve_fit para incluir os erros como pesos durante o cálculo do ajuste:

param, pcov = curve_fit(func,xd,yd, sigma = erro)

Faça um gráfico com o novo ajuste, e compare os novos valores de $E_r$ e $\Gamma$ com os valores teóricos.

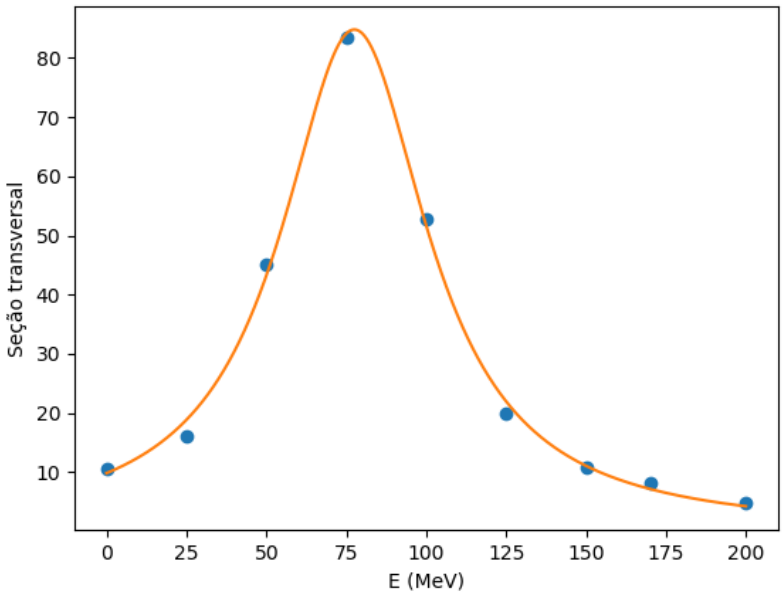



**Table 7.1** Experimental values for a scattering cross section ($f(E)$ in the theory), each with absolute error $\pm\sigma_i$, as a function of energy ($x_i$ in the theory).

| $i$ = | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $E_i$ (MeV) | 0 | 25 | 50 | 75 | 100 | 125 | 150 | 175 | 200 |
| $g(E_i)$ (mb) | 10.6 | 16.0 | 45.0 | 83.5 | 52.8 | 19.9 | 10.8 | 8.25 | 4.7 |
| Error (mb) | 9.34 | 17.9 | 41.5 | 85.5 | 51.5 | 21.5 | 10.8 | 6.29 | 4.14 |

**Relembrando alguns comandos**

**loadtxt**

Importa dados de um arquivo de texto externo. Função do numpy.
**Ex:** A = loadtxt(”dados.txt”,float)

**arange**

Cria um vetor contendo uma lista de valores. Funcão do numpy.
**Ex:** a = arange(0.0,1.0,0.1)

**Slicing**

Útil para pegar subconjuntos de um vetor ou de uma matriz.
**Ex:** xd = dados[:,0] - pega a primeira coluna de uma matriz de dados.