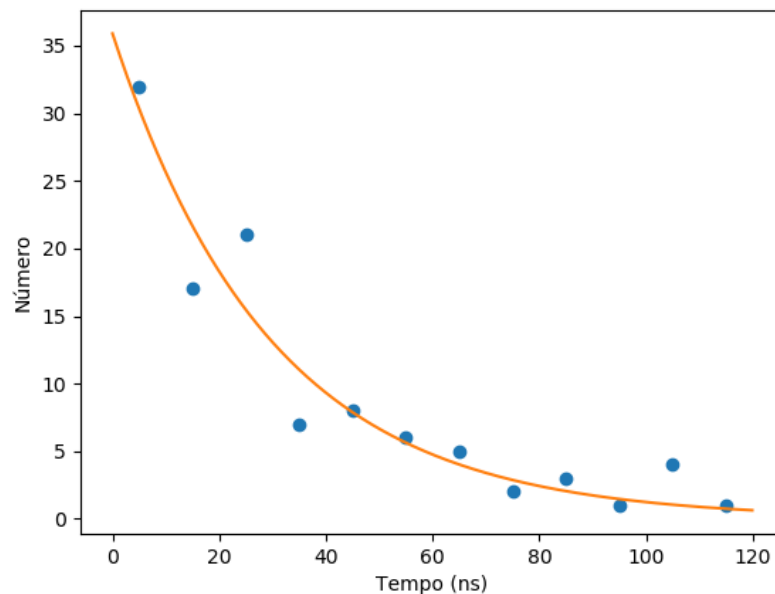


FÍSICA COMPUTACIONAL II

AJUSTE DE DADOS - Entrega: 16/04

Ajuste de dados usando o `scipy.optimize`

- Importe os dados do arquivo 'decay.data' e a função `curve_fit`:
`from scipy.optimize import curve_fit`
- Defina uma função, que será usada pelo `curve_fit`:
`def func(x,a,b):`
 `return a*exp(-b*x)`
- Faça o ajuste:
`param, pcov = curve_fit(func,xd,yd)`
- Faça um gráfico similar ao mostrado abaixo. O modelo proposto parece se adequar aos dados experimentais? Calcule o R^2 do seu ajuste.
- 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

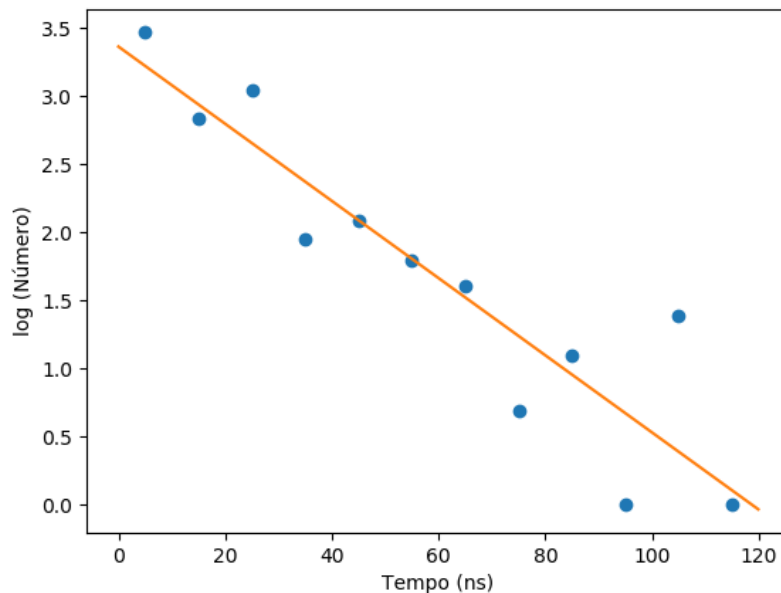
- Calcule o log dos dados importados anteriormente (Dica: se você importar o log do `numpy`, a função aceitará como argumento um vetor.)

- b) Faça um gráfico do $\log(yd)$ *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:

$$m = \frac{\sum_i x_i (y_i - \bar{y})}{\sum_i x_i (x_i - \bar{x})}$$

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

- 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:

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

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 Γ . 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 = yerr,fmt = 'o',capsize=5) ← inclui as barras de erro.
```

xd, yd e yerr são, respectivamente, a primeira, a segunda e a terceira coluna de dados do arquivo experimental.

- 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 = yerr)
```

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

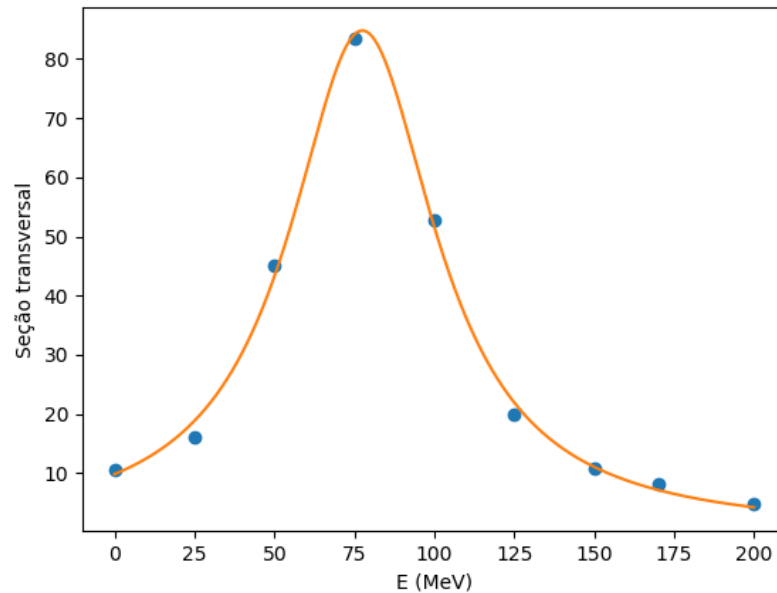


Table 7.1 Experimental values for a scattering cross section ($f(E)$ in the theory), each with absolute error $\pm\sigma_i$, as a function of energy (x_i in the theory).

$i =$	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. Função 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.