Fit Lorentzprofiel

October 31, 2021

1 Fit Lorentzprofiel

Ruben Van der Borght Wiskunde-Fysica, r0829907

[1]:
```
import numpy as np #Importeer enkele nodige packages.
import math
from scipy.optimize import minimize, fsolve
import matplotlib.pyplot as plt
from scipy.stats import chi2
import nbconvert
#TO DO
#REFERENTIES
#BEDUIDENDE CIJFERS AANPASSEN
#DIE ENE BREUK MET EEN IN DE TELLER
```

In dit document wordt een dataset met metingen van posities x [mm] en intensiteiten I met een arbitraire eenheid [arb.eenh] geanalyseerd. Met een fit worden x_0 de verschuivingsparameter, γ de schaalparameter, A de vermenigvuldigheidsfactor en y_0 de offset berekend. Het Lorentzprofiel is gegeven door

$$I(x_j|x_0, \gamma, A, y_0) = \frac{A}{\pi}\frac{\gamma}{(x-x_0)^2+\gamma^2} + y_0$$

Er is gegeven dat I gemeten is door fotonen te meten en dat I een Poissonverdeling $P(\alpha(x|\theta))$ volgt. Omdat alle waarden van I veel groter zijn dan 10, benaderen we de verdeling met een normale verdeling $N(\alpha(x|\theta), \alpha(x|\theta))$.

[2]:
```
dataset = np.loadtxt("38.txt", delimiter=" ").T
x=dataset[0]
I=dataset[1]

theta = ["\gamma","A","y_0","x_0"]
theta_units=["mm","arb.eenh.\cdot mm","arb.eenh.","mm"]

print(min(I))
```

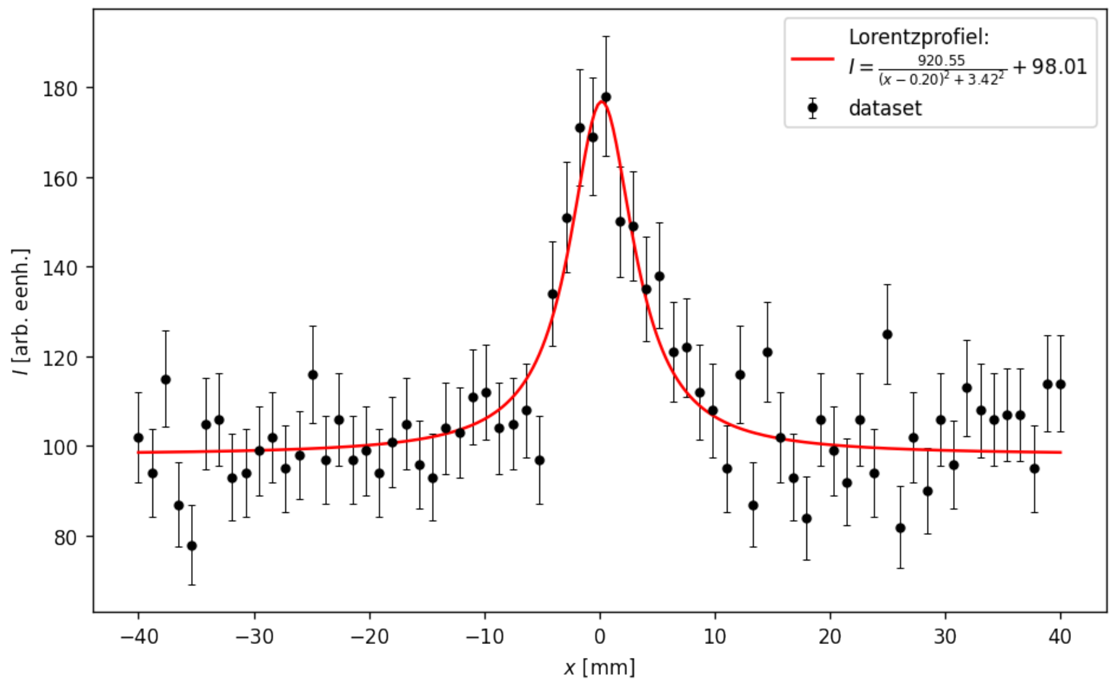
```
78.0
```

1.1 Plot van de dataset inclusief fit

De dataset valt te bekijken op onderstaande grafiek.

[3]:
```
fig, ax = plt.subplots(nrows=1, ncols=1, dpi=120, figsize=(8, 5))
ax.errorbar(x, I, yerr=np.sqrt(I), label="dataset",marker="o", markersize=4,␣
 ↪fmt=" ", color="black", ecolor="black", capsize=2, capthick=0.6, linewidth=0.6)

def intensity(x,gamma,A,y_0,x_0):
    I = A*gamma/(np.pi*((x-x_0)**2+gamma**2))+y_0
    return I

def LS_intensity(theta):
    gamma,A,y_0,x_0=theta
    LS=0
    for i in range(len(x)):
        LS+=(I[i]-intensity(x[i],gamma,A,y_0,x_0))**2/I[i]
    return LS

opt = minimize(LS_intensity,(100,850,0,3.5))

gamma,A,y_0,x_0=theta_hat=opt.x
x_dots = np.linspace(np.min(x),np.max(x),300)
ax.plot(x_dots, intensity(x_dots,opt.x[0],opt.x[1],opt.x[2],opt.x[3]), 'r',
        label='Lorentzprofiel:\n'+r'$I=\frac{%0.2f}{(x-%0.2f)^2+%0.2f^2}+%0.2f$'␣
 ↪% (A*gamma/np.pi,x_0,gamma,y_0))

mini = LS_intensity(theta_hat)

for i in range(len(theta)):
    print(r"$%s$" % theta[i],theta_hat[i])


ax.set_ylabel(r"$I$ [arb. eenh.]")
ax.set_xlabel(r"$x$ [mm]")
ax.legend()
plt.tight_layout() ; plt.show()
```

```
$\gamma$ 3.418751769174252
$A$ 845.9176096705844
$y_0$ 98.01376852384685
$x_0$ 0.20303213676389797
```



Bijgevolg is $\hat{\theta} = (3.42, 846, 98.0, 0.203)$, zodat het Lorentzprofiel

$$I(x|\hat{\theta}) = \frac{846}{\pi} \frac{3.42^2}{(x-0.203)^2 + 3.42^2} + 98.0$$

wordt, ofwel

$$I(x|\hat{\theta}) = \frac{1}{(x-0.203)^2 + 3.42^2} + 98.0 \quad .$$

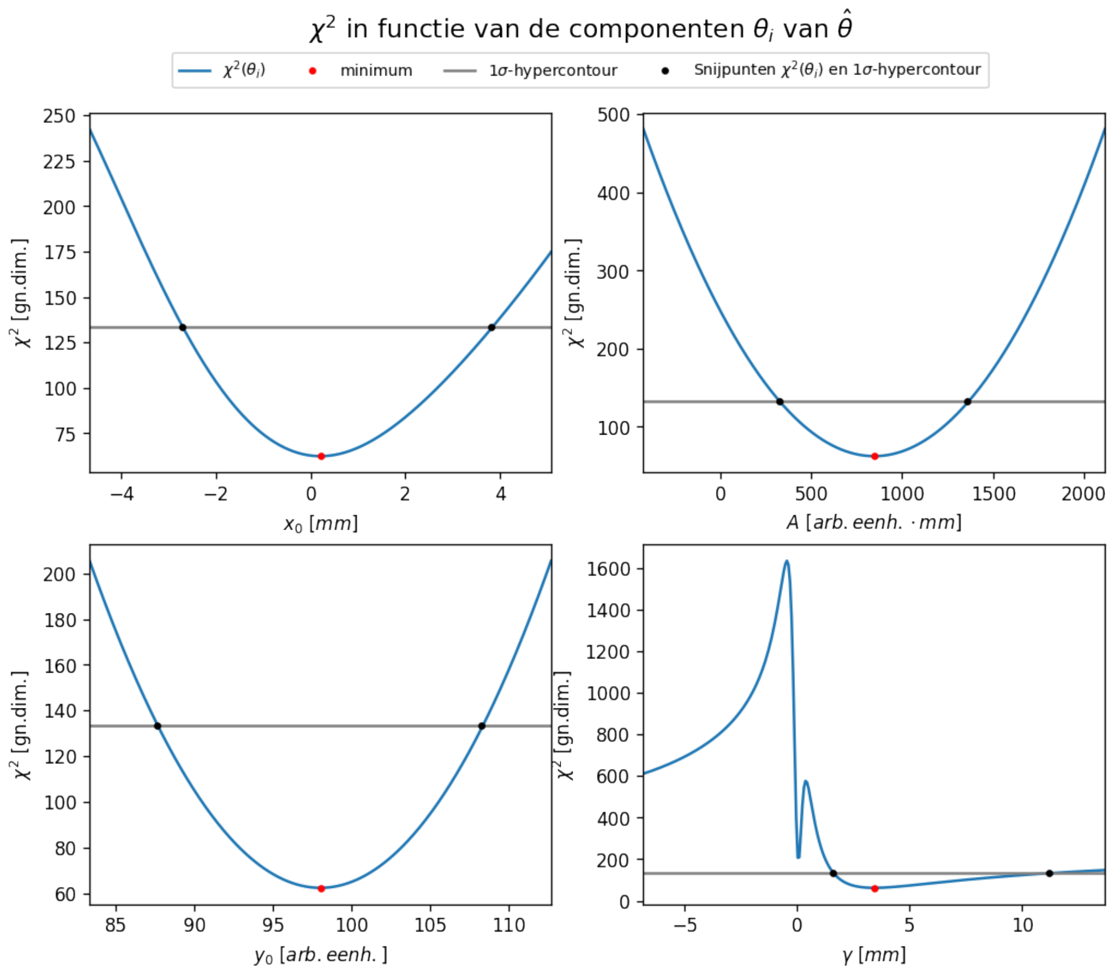
1.2 Onzekerheden op de gefitte $\hat{\theta}$

Met behulp van de methode die in het opgaveblad werd besproken (ref.) wordt de onzekerheid op de verschillende parameters achtereenvolgens berekend.

[4]:
```
fig, bx = plt.subplots(nrows=2, ncols=2, dpi=120, figsize=(10, 8))
nu=70-4

bounds = [3,1.5,0.15,24]
sigma = mini+chi2.ppf(0.68,df=nu)
theta_uncertainty=[]

for i in range(len(theta_hat)):
    param=theta_hat[i]
    points = np.linspace(param-param*bounds[i],param+param*bounds[i],300)
    b=list(theta_hat)
    b[i]=points
```

```
    j = 1 if i%2==0 else 0
    k = 0 if i>1 else 1

    bx[j][k].plot(points,LS_intensity(b),label=r'$\chi^2( \theta_i)$',zorder=-1)

    bx[j][k].
 ↪plot(theta_hat[i],mini,"o",color='red',markersize=3,label=r'minimum',zorder=1)

    bx[j][k].plot(points,sigma*np.ones(300),'gray',label=r'$1␣
 ↪\sigma$-hypercontour',zorder=-1)

    idx = np.argwhere(np.diff(np.sign(LS_intensity(b) - sigma*np.ones(300)))).
 ↪flatten() #https://stackoverflow.com/questions/28766692/intersection-
    #of-two-graphs-in-python-find-the-x-value
    bx[j][k].
 ↪plot(points[idx[0]],sigma,"o",color='black',markersize=3,label=r'Snijpunten␣
 ↪$\chi^2(\theta_i)$ en $1\sigma$-hypercontour',zorder=1)
    bx[j][k].plot(points[idx[1]],sigma,"o",color='black',markersize=3,zorder=1)
    theta_uncertainty.append((np.
 ↪format_float_scientific(theta_hat[i]-points[idx[0]],precision=0,unique=True,exp_digits=1),
                              np.
 ↪format_float_scientific(points[idx[1]]-theta_hat[i],precision=0,unique=True,exp_digits=1)))
    bx[j][k].set_ylabel(r"$\chi^2$ [gn.dim.]")
    bx[j][k].set_xlabel(r"$%s$ [$%s$]" % (theta[i],theta_units[i]))
    bx[j][k].set_xlim(param-param*bounds[i],param+param*bounds[i])
lines, labels = fig.axes[-1].get_legend_handles_labels()
fig.legend(lines, labels,ncol=4, loc ='upper center' ,bbox_to_anchor=(0.5, 0.
 ↪945),fontsize=8.5)
fig.suptitle(r'$\chi^2$ in functie van de componenten $\theta_i$ van␣
 ↪$\hat{\theta}$',fontsize=14)
```

[4]:
```
Text(0.5, 0.98, '$\\chi^2$ in functie van de componenten $\\theta_i$ van
$\\hat{\\theta}$')
```



[5]:
```
for i in range(len(theta)):
```

```
  File "C:\Users\ruben\AppData\Local\Temp/ipykernel_15392/2179139369.py", line 1
    for i in range(len(theta)):
                               ^
SyntaxError: unexpected EOF while parsing
```

De parameters met hun onzekerheden worden dus gegeven door

$$\gamma = 3^{+8}_{-2}\text{ mm} \qquad A = (800 \pm 500)\text{ arb.eenh.} \cdot \text{mm} \qquad y_0 = (98 \pm 10)\text{ mm} \qquad x_0 = 0^{+4}_{-3}\text{ mm}$$

1.3 Kwaliteit van de fit

[ ]:
```
theta=opt.x[0],opt.x[1],opt.x[2],opt.x[3]
chi_2=LS_intensity(theta)
print('chi²_0:\t\t',chi_2)
nu=len(x)-len(theta)
chi_2_red=chi_2/nu
print('chi²_red:\t',chi_2_red)
```

Aangezien $\chi^2_{red} \approx 1$, zal de fit goed aansluiten bij de steekproef. Het model is geen overfit want dan zou $\chi^2_{red} > 1$, en ook geen onderfit want dan zou $\chi^2_{red} < 1$.

Nu wordt aan de hand van bovenstaande berekeningen bepaald of het Lorentz profiel een goed model is voor de dataset. Daarvoor wordt de p-waarde $p(\chi^2_\nu > \chi^2_0)$ berekend. Uit het model volgt dat $\nu = N - p = 66$

[ ]:
```
p = 1-chi2.cdf(chi_2,df=nu)
print(p)
```

De gevonden p-waarde is groter dan het significantieniveau $\alpha = 5\%$, dus de gevonden fit wordt aanvaard. Het Lorentzmodel sluit dus goed aan bij de gegeven dataset.