

Fit Lorentzprofiel

Ruben Van der Borcht, r0829907

1 Inleiding

```
[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
```

In dit document wordt de dataset 38.txt (zie Bijlage) met metingen van posities x [mm] en intensiteiten I met een arbitraire eenheid [arb.eenh.] geanalyseerd. Met een fit worden γ de schaalparameter, A de vermenigvuldigheidsfactor, y_0 de offset en x_0 de verschuivingsparameter berekend. Het Lorentzprofiel is gegeven door

$$I(x_j|\gamma, A, y_0, x_0) = \frac{A}{\pi} \frac{\gamma}{(x - x_0)^2 + \gamma^2} + y_0. \quad [1]$$

De paramters worden ook genoteerd als $\theta = (\gamma, A, y_0, x_0)$. Er is gegeven dat I gemeten is door fotonen te meten en dat elke meting uit een Poissonverdeling $P(I(x|\theta))$ komt. [1] Omdat het minimum van de I -waarden, (78 ± 9) arb. eenh. (1σ -fout), veel groter is dan 10, wordt de verdeling benaderd met een normale verdeling $N(I(x|\theta), I(x|\theta))$. De Least Square-waarde¹ wordt dan ook op dezelfde manier berekend als bij normaal verdeelde waarden.

Het model en de gegevens worden reeds ingeladen.

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

theta = ["gamma","A","y_0","x_0"] #Definieer enkele lijsten die later bij
    ↳iteraties gebruikt worden.
theta_latex = ["\gamma","A","y_0","x_0"]
theta_units=["mm","arb.eenh.\cdot mm","arb.eenh.", "mm"]

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

print(min(I)) #Bereken de kleinste I-waarde.
```

78.0

¹Verder worden de benamingen LS-waarde of χ^2 -waarde gehanteerd.

2 Plot van de dataset inclusief fit

De LS-waarde wordt hieronder berekend. Dat gebeurt op dezelfde manier als bij normaal verdeelde waarden.

```
[3]: def LS_intensity(theta): #Definieer een functie die de LS-waarde berekend bij
    ↳gegeven \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
```

De dataset valt te bekijken op Figuur 1. Er wordt aangenomen dat de fout op de x -waarden verwaarloosbaar is tegenover de fout op I .

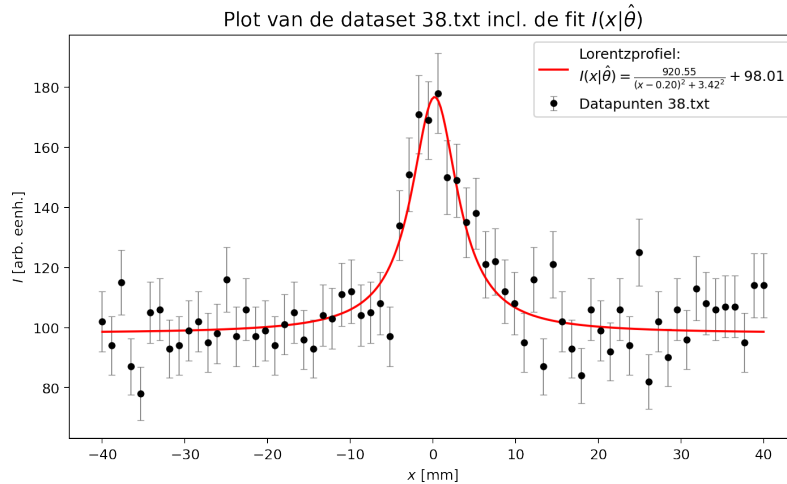
```
[4]: fig, ax = plt.subplots(nrows=1, ncols=1, dpi=180, figsize=(8, 5)) #Laad de
    ↳figuur en plot de datapunten met onzekerheid.
    ax.errorbar(x, I, yerr=np.sqrt(I), label="Datapunten 38.txt", marker="o",
    ↳markersize=4, fmt=" ",
        color="black", ecolor="gray", capsize=2.3, capthick=0.5, linewidth=0.
    ↳7)

    opt = minimize(LS_intensity,(100,1000,0,5)) #Bereken de beste schatter
    ↳\hat{\theta}.
    gamma,A,y_0,x_0=theta_hat=opt.x #De gok is gebaseerd op de
    ↳scatterplot van de datapunten.

    x_dots = np.linspace(np.min(x),np.max(x),300) #Plot het Lorentzprofiel voor de
    ↳beste schatter \hat{\theta}.
    ax.plot(x_dots, intensity(x_dots,opt.x[0],opt.x[1],opt.x[2],opt.x[3]), "r",
        label="Lorentzprofiel:\n"+r"$I(x\vert\hat{\theta})=\frac{0.2f}{(x-0.
    ↳2f)^2+0.2f^2}+0.2f$" % (A*gamma/np.pi,x_0,gamma,y_0))

    mini = LS_intensity(theta_hat) #Bereken de minimale LS-waarde ofwel \chi^2_0 van
    ↳\theta{\hat}.

    ax.set_ylabel(r"$I$ [arb. eenh.]") #Pas de lay-out van de plot aan.
    ax.set_xlabel(r"$x$ [mm]")
    ax.legend()
    ax.set_title(r"Plot van de dataset 38.txt incl. de fit
    ↳$I(x\vert\hat{\theta})$", fontsize=14)
    plt.figtext(0.5,-0.05, "Figuur 1: Een plot van de datapunten ($x,I$) uit de
    ↳dataset 38.txt. Er is geen fout op de $x$-waarden\n in de set gegeven, dus er
    ↳wordt verondersteld dat deze verwaarloosbaar is. De rode curve toont de beste
    ↳fit aan de datapunten.", wrap=True, horizontalalignment='center', fontsize=12)
    plt.tight_layout() ; plt.show()
```



Figuur 1: Een plot van de datapunten (x, I) uit de dataset 38.txt. Er is geen fout op de x-waarden in de set gegeven, dus er wordt verondersteld dat deze verwaarloosbaar is. De rode curve toont de beste fit aan de datapunten.

```
[5]: for i in range(len(theta)):                                #Print de componenten van
    print(theta_hat[i])                                         #Deze componenten zijn
    #nog niet afgerond op het juiste aantal beduidende cijfers.
```

```
gamma    3.42
A        845.92
y_0      98.01
x_0      0.20
```

Daaruit volgt dat $\hat{\theta} = (3.42 \text{ mm}, 845.92 \text{ arb.eenh.} \cdot \text{mm}, 98.01 \text{ arb.eenh.}, 0.20 \text{ mm})$, zodat het Lorentz-profiel

$$I(x|\hat{\theta}) = \frac{845.92}{\pi} \frac{3.42}{(x - 0.20)^2 + 3.42^2} + 98.01$$

wordt. Bij de waarden hierboven werd nog geen rekening gehouden met de onzekerheden die in de volgende paragraaf zullen worden gevonden. Wanneer deze berekend zijn, wordt het effectieve model met de juiste waarde $\hat{\theta}_{eff}$ gegeven.

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

Met behulp van de methode die in het opgaveblad werd besproken, wordt de onzekerheid op de verschillende parameters achtereenvolgens berekend. [1] Daarvoor werd een plot gemaakt van χ^2 in functie van elke component θ_i van θ , zoals op Figuur 2 te zien valt.

```
[6]: fig, bx = plt.subplots(nrows=2, ncols=2, dpi=180, figsize=(10, 8))

nu=len(x)-len(theta)                                         #Bereken de waarde van de 1-sigma-hypercontour.
sigma = mini+chi2.ppf(0.68,df=nu)
```

```

theta_uncertainty=[]
bounds = [3,1.5,0.15,24]

for i in range(len(theta_hat)):
    par=theta_hat[i]
    points = np.linspace(par-par*bounds[i],par+par*bounds[i],300) #Bereken de
    →grenzen van de plot.
    b=list(theta_hat)
    b[i]=points

    j = 0 if i%2==0 else 1 #Bepaal de subplot.
    k = 1 if i>1 else 0

    bx[j][k].plot(points,LS_intensity(b), label=r'$\chi^2(\theta_i)$') #Plot
    →\chi^2(par)

    bx[j][k].plot(theta_hat[i],mini,"o", color='red',
    →markersize=3,label=r'minimum') #Plot het minimum van \chi^2(par).

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

    idx = np.argwhere(np.diff(np.sign(LS_intensity(b) - sigma*np.ones(300))))
    →flatten() #Bereken de snijpunten en plot ze.
    bx[j][k].
    →plot(points[idx[0]],sigma,"o",color='black',markersize=3,label=r'Snijpunten
    →$\chi^2(\theta_i)$ en $1\sigma$-hypercontour')
    bx[j][k].plot(points[idx[1]],sigma,"o",color='black',markersize=3)

    theta_uncertainty.append((np.
    →format_float_scientific(theta_hat[i]-points[idx[0]],precision=1,unique=True),np.
    →format_float_scientific(points[idx[1]]-theta_hat[i],precision=1,unique=True)))
    →#Maak een lijst van de onzekerheden.

    bx[j][k].set_ylabel(r"$\chi^2$ [geen dimensie]") #Pas de lay-out van de
    →subplot aan.
    bx[j][k].set_xlabel(r"%s$ [%s$]" % (theta_latex[i],theta_units[i]))
    bx[j][k].set_xlim(par-par*bounds[i],par+par*bounds[i])

lines, labels = fig.axes[-1].get_legend_handles_labels() #Pas de lay-out van de
→totale plot aan.
fig.legend(lines, labels,ncol=4, loc = 'upper center',bbox_to_anchor=(0.5, 0.
→945),fontsize=8.5)

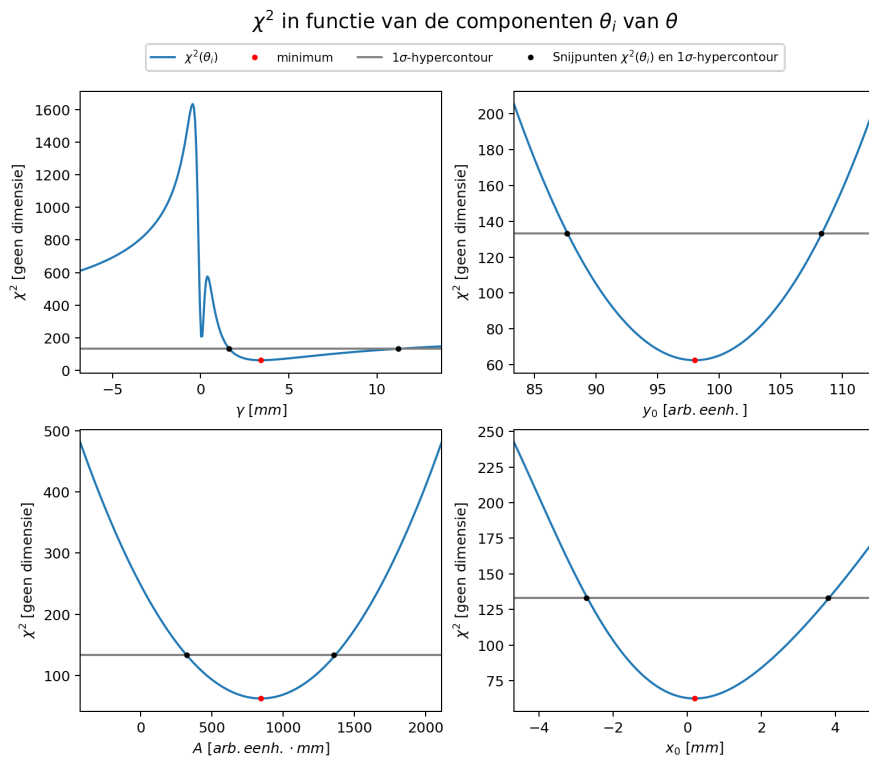
```

```

fig.suptitle(r'$\chi^2$ in functie van de componenten $\theta_i$ van $\theta$')
fig.set_figwidth(14)
width = {'width': 1}
plt.figtext(0.5,-0.05, r"Figuur 2: Vier plots van $\chi^2$ in functie van elk van de vier componenten van $\theta$. Op de grafieken werd het minimum van de functie aangeduid, dat bij $\theta = \hat{\theta}$ wordt aangenomen. Daarnaast vallen de $1\sigma$-hypercontour en de snijpunten van $\chi^2$ met deze contour te zien. Hieruit worden de onzekerheden op de $\theta_i$'s afgeleid.", wrap=True, horizontalalignment='center', fontsize=12)

```

[6]: Text(0.5, -0.05, "Figuur 2: Vier plots van χ^2 in functie van elk van de vier componenten van θ . Op de grafieken werd het minimum van de functie aangeduid, dat bij $\theta = \hat{\theta}$ wordt aangenomen. Daarnaast vallen de 1σ -hypercontour en de snijpunten van χ^2 met deze contour te zien. Hieruit worden de onzekerheden op de θ_i 's afgeleid.")



Figuur 2: Vier plots van χ^2 in functie van elk van de vier componenten van θ . Op de grafieken werd het minimum van de functie aangeduid, dat bij $\theta = \hat{\theta}$ wordt aangenomen. Daarnaast vallen de 1σ -hypercontour en de snijpunten van χ^2 met deze contour te zien. Hieruit worden de onzekerheden op de θ_i 's afgeleid.

```

[7]: for i in range(len(theta)): #Print de onzekerheden.
      print('%s:\t' % theta[i], ' ', theta_uncertainty[i][0], ' ', theta_uncertainty[i][1], ' ')

```

```
gamma: [ 1.8e+0 , 7.8e+0 ]
A:      [ 5.2e+2 , 5.1e+2 ]
y_0:    [ 1.0e+1 , 1.0e+1 ]
x_0:    [ 2.9e+0 , 3.6e+0 ]
```

De parameters met hun onzekerheden (1σ -fout) worden dus gegeven door

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

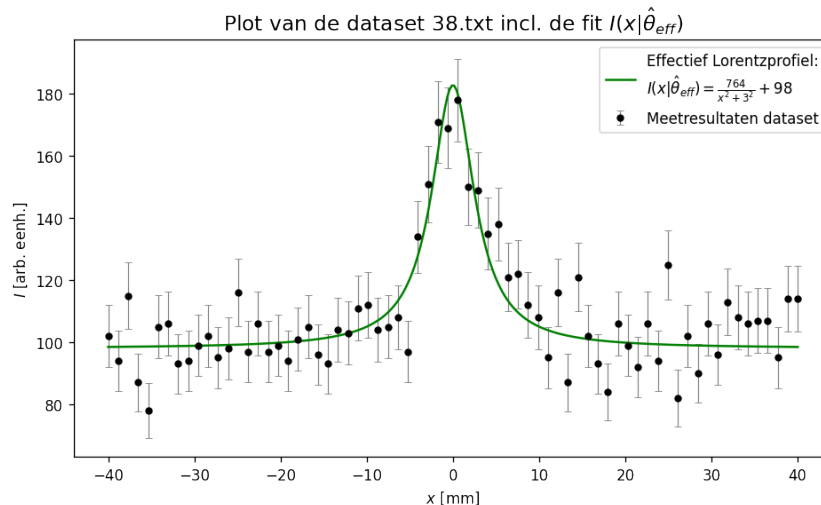
Bijgevolg is $\hat{\theta}_{eff} = (3 \text{ mm}, 800 \text{ arb.eenh.} \cdot \text{mm}, 98 \text{ arb.eenh.}, 0 \text{ mm})$. Hiermee kan de effectieve fit berekend en geplot worden (zie Figuur 3). Die is

$$I(x|\hat{\theta}_{eff}) = \frac{800}{\pi} \frac{3}{x^2 + 9} + 98$$

```
[8]: theta_eff=(3,800,98,0)
fig, ax = plt.subplots(nrows=1, ncols=1, dpi=120, figsize=(8, 5)) #Laad de
    ↳figuur en plot de datapunten met onzekerheid.
ax.errorbar(x, I, yerr=np.sqrt(I), label="Meetresultaten dataset",marker="o",
    ↳markersize=4, fmt=" ",
        color="black", ecolor="gray", capsize=2.3, capthick=0.5, linewidth=0.
    ↳7)

ax.plot(x_dots,
    ↳intensity(x_dots,theta_eff[0],theta_eff[1],theta_eff[2],theta_eff[3]),
    ↳"green", #Plot de effectieve fit.
        label="Effectief Lorentzprofiel:\n"+r"$I(x \text{ \texttt{\textbackslash hat\{}}\text{ \texttt{\textbackslash theta}}_{\text{eff}})}=\frac{0.0f}{x^2+0.0f^2}+0.0f$" % (theta_eff[1]*theta_eff[0]/
    ↳np.pi,theta_eff[0],theta_eff[2]))

ax.set_ylabel(r"$I$ [arb. eenh.]") #Pas de lay-out van de plot aan.
ax.set_xlabel(r"$x$ [mm]")
ax.legend()
ax.set_title(r"Plot van de dataset 38.txt incl. de fit
    ↳$I(x\text{ \texttt{\textbackslash hat\{}}\text{ \texttt{\textbackslash theta}}_{\text{eff}})}$",fontsize=14)
plt.figtext(0.5,-0.05, "Figuur 3: Een plot van de datapunten ($x,I$) uit de
    ↳dataset 38.txt. De groene curve toont de effectieve fit aan de datapunten.",
    ↳wrap=True, horizontalalignment='center', fontsize=12)
plt.tight_layout() ; plt.show()
```



Figuur 3: Een plot van de datapunten (x, I) uit de dataset 38.txt. De groene curve toont de effectieve fit aan de datapunten.

4 Kwaliteit van de fit

Nu wordt onderzocht of de gevonden fit aanvaardbaar is. Daarvoor wordt een rechtezijdige hypothesetest uitgevoerd. De nulhypothese H_0 is dat χ_0^2 een χ_ν^2 -verdeling volgt. [1] Er wordt gewerkt op significantieniveau $\alpha = 5\%$. Deze keuze van α zorgt voor een kleine kans dat de fit foutief verworpen of foutief aanvaard wordt. [2]

```
[9]: chi_2_0=LS_intensity(theta_eff) #Bereken \chi^2_0.
print('chi^2_0:\t\t',chi_2_0)

nu=len(x)-len(theta) #Bereken \chi^2_red.
chi_2_red=chi_2_0/nu
print('chi^2_red:\t',chi_2_red)
```

```
chi^2_0:          63.54571100967008
chi^2_red:        0.9628138031768194
```

Eerst word het model geëvalueerd met behulp van de teststatistiek χ_{red}^2 . Aangezien $\chi_{red}^2 \approx 1$, zal de fit goed aansluiten bij de datapunten. Het model is geen overfit ofwel te precies want dan zou $\chi_{red}^2 < 1$. Verder is het ook geen onderfit ofwel te ruw want dan zou $\chi_{red}^2 > 1$.

Vervolgens wordt bepaald of het Lorentzprofiel een goed model is voor de dataset met behulp van p -waarden. Daarvoor wordt de p -waarde $p(\chi_\nu^2 > \chi_0^2)$ berekend. Uit het model volgt dat $\nu = N - p = 66$.

```
[10]: p = chi2.sf(chi_2_0,df=nu) #Bereken de p-waarde.
print(p)
```

```
0.5628016941707908
```

De gevonden p -waarde is groter dan het significantieniveau $\alpha = 5\%$, dus de gevonden fit wordt aanvaard. Omdat het model aan de twee evaluatiecriteria voldoet, wordt besloten dat het Lorentzmodel goed aansluit bij de gegeven dataset.

5 Conclusie

Het Lorentzmodel beschrijft de dataset 38.txt goed. Voor de beste fit is $\hat{\theta}_{eff} = (3 \text{ mm}, 800 \text{ arb.eenh.} \cdot \text{mm}, 98 \text{ arb.eenh.}, 0 \text{ mm})$. Het Lorentzmodel wordt dan

$$I(x|\hat{\theta}_{eff}) = \frac{800}{\pi} \frac{3}{x^2 + 9} + 98.$$

Deze fit wordt aanvaard op $\alpha = 5\%$.

Referenties

- [1] Fitopgave. Experimentele basistechnieken in de natuurkunde G0P32A. KU Leuven; 2021.
- [2] Verdonck T. Statistiek I G0U47A. KU Leuven; 2020.

Bijlage

Tabel 1: Dataset 38.txt (deel 1). De set bestaat uit posities x gekoppeld aan intensiteiten I die gemeten werden door fotonen te tellen.

$x[\text{mm}]$	$I[\text{arb.eenh}]$
-4.0000000000000000e+01	1.0200000000000000e+02
-3.884057971014492949e+01	9.4000000000000000e+01
-3.768115942028985188e+01	1.1500000000000000e+02
-3.652173913043478137e+01	8.7000000000000000e+01
-3.536231884057971087e+01	7.8000000000000000e+01
-3.420289855072464036e+01	1.0500000000000000e+02
-3.304347826086956275e+01	1.0600000000000000e+02
-3.188405797101449224e+01	9.3000000000000000e+01
-3.072463768115942173e+01	9.4000000000000000e+01
-2.956521739130434767e+01	9.9000000000000000e+01
-2.840579710144927361e+01	1.0200000000000000e+02
-2.724637681159420310e+01	9.5000000000000000e+01
-2.608695652173912904e+01	9.8000000000000000e+01
-2.492753623188405498e+01	1.1600000000000000e+02
-2.376811594202898448e+01	9.7000000000000000e+01
-2.260869565217391397e+01	1.0600000000000000e+02
-2.144927536231883991e+01	9.7000000000000000e+01
-2.028985507246376585e+01	9.9000000000000000e+01
-1.913043478260869534e+01	9.4000000000000000e+01
-1.797101449275362128e+01	1.0100000000000000e+02
-1.681159420289855078e+01	1.0500000000000000e+02
-1.565217391304347672e+01	9.6000000000000000e+01
-1.449275362318840621e+01	9.3000000000000000e+01
-1.33333333333333215e+01	1.0400000000000000e+02
-1.217391304347825809e+01	1.0300000000000000e+02
-1.101449275362318758e+01	1.1100000000000000e+02
-9.855072463768113522e+00	1.1200000000000000e+02
-8.695652173913043015e+00	1.0400000000000000e+02
-7.536231884057968955e+00	1.0500000000000000e+02
-6.376811594202898448e+00	1.0800000000000000e+02
-5.217391304347827941e+00	9.7000000000000000e+01
-4.057971014492750328e+00	1.3400000000000000e+02
-2.898550724637679821e+00	1.5100000000000000e+02
-1.739130434782609314e+00	1.7100000000000000e+02
-5.797101449275317009e-01	1.6900000000000000e+02

Tabel 2: Dataset 38.txt (deel 2). De set bestaat uit posities x gekoppeld aan intensiteiten I die gemeten werden door fotonen te tellen.

$x[\text{mm}]$	$I[\text{arb.eenh}]$
5.797101449275388063e-01	1.7800000000000000e+02
1.739130434782609314e+00	1.5000000000000000e+02
2.898550724637679821e+00	1.4900000000000000e+02
4.057971014492757433e+00	1.3500000000000000e+02
5.217391304347827941e+00	1.3800000000000000e+02
6.376811594202898448e+00	1.2100000000000000e+02
7.536231884057976060e+00	1.2200000000000000e+02
8.695652173913046568e+00	1.1200000000000000e+02
9.855072463768117075e+00	1.0800000000000000e+02
1.101449275362318758e+01	9.5000000000000000e+01
1.217391304347826519e+01	1.1600000000000000e+02
1.333333333333333570e+01	8.7000000000000000e+01
1.449275362318840621e+01	1.2100000000000000e+02
1.565217391304348382e+01	1.0200000000000000e+02
1.681159420289855433e+01	9.3000000000000000e+01
1.797101449275362484e+01	8.4000000000000000e+01
1.913043478260869534e+01	1.0600000000000000e+02
2.028985507246377296e+01	9.9000000000000000e+01
2.144927536231884346e+01	9.2000000000000000e+01
2.260869565217391397e+01	1.0600000000000000e+02
2.376811594202898448e+01	9.4000000000000000e+01
2.492753623188406209e+01	1.2500000000000000e+02
2.608695652173913970e+01	8.2000000000000000e+01
2.724637681159420310e+01	1.0200000000000000e+02
2.840579710144928072e+01	9.0000000000000000e+01
2.956521739130434412e+01	1.0600000000000000e+02
3.072463768115942173e+01	9.6000000000000000e+01
3.188405797101449934e+01	1.1300000000000000e+02
3.304347826086956275e+01	1.0800000000000000e+02
3.420289855072464036e+01	1.0600000000000000e+02
3.536231884057971797e+01	1.0700000000000000e+02
3.652173913043478137e+01	1.0700000000000000e+02
3.768115942028985899e+01	9.5000000000000000e+01
3.884057971014493660e+01	1.1400000000000000e+02
4.0000000000000000e+01	1.1400000000000000e+02