

# Projekt 5

Fit experimenteller Daten  
Simulation der Bewegung von Elektronen in Materie

**Christian Gommeringer**

betreut durch Dr. Hehl

Tübingen, den 10. April 2023

# 1 Anpassung experimenteller Daten

Zu Beginn dieser Betrachtung werden zunächst die Konzepte von gewichtetem Mittelwert sowie innerer und externer Varianz vorgestellt. Für eine Reihe von Messwerten mit bekanntem Fehler  $(x_i, \sigma_i)$  werden diese Größen wie folgt definiert. Der gewichtete Mittelwert ergibt sich durch

$$\bar{x} = \frac{\sum_i \frac{x_i}{\sigma_i^2}}{\sum_i \frac{1}{\sigma_i^2}}$$

Innere Varianz berechnet sich durch

$$\sigma_{m,int}^2 = \left( \sum_i \frac{1}{\sigma_i^2} \right)^{-1}$$

Die externe Varianz für eine betrachtete Größe bei n Messwerten berechnet sich durch

$$\sigma_{m,ext}^2 = \frac{1}{n-1} \frac{\sum_i \frac{(x_i - \bar{x})^2}{\sigma_i^2}}{\sum_i \frac{1}{\sigma_i^2}}$$

was sich durch Einführen von

$$\chi^2 = \sum_i \frac{(x_i - \bar{x})^2}{\sigma_i^2}$$

auch schreiben lässt als

$$\sigma_{m,ext}^2 = \frac{\chi^2}{n-1} \sigma_{m,int}^2$$

$$\begin{aligned} \bar{c} &= 299792.777 \\ \sigma_{m,int} &= 0.911 \\ \sigma_{m,ext} &= .616 \end{aligned}$$

Messwerte können auch dazu verwendet werden, um einen funktionalen Zusammenhang zwischen zwei Größen zu bestimmen. Wir möchten nun durch gegebene Messwerte Parameter einer Funktion abschätzen. Dazu wird die  $\chi^2$ -Funktion minimiert.

$$\chi^2(a, b) = \sum_i \frac{(y_i - a - bx)^2}{\sigma_i^2}$$

Bei einer linearen Funktion führt das auf ein lineares Gleichungssystem, das nach Definition der Größen

$$S_{\alpha,\beta} = \sum_i \frac{\alpha_i \beta_i}{\sigma_i^2}$$

gelöst wird durch die optimalen Parameter

$$a = \frac{S_{x,x}S_{y,1} - S_{x,1}S_{x,y}}{D}$$

$$b = \frac{S_{1,1}S_{x,y} - S_{x,1}S_{y,1}}{D}$$

mit  $D = S_{1,1}S_{x,x} - S_{x,1}^2$ . Durch Gauß'sche Fehlerfortpflanzung kann noch der Fehler der beiden Parameter bestimmt werden.

$$\sigma_a^2 = \frac{S_{x,x}}{D}$$

$$\sigma_b^2 = \frac{S_{1,1}}{D}$$

In unserem Fall haben wir Spannungswerte und fehlerbehaftete Werte für die Stromstärke.

<b><i>U/V</i></b>	<b><i>I/A</i></b>	<b><math>\Delta I/A</math></b>
0.5	0.065	$1 \cdot 10^{-2}$
1.0	0.206	$1 \cdot 10^{-2}$
1.5	0.405	$2 \cdot 10^{-2}$
2.0	0.492	$2 \cdot 10^{-2}$
2.5	0.606	$2 \cdot 10^{-2}$
3.0	0.782	$2 \cdot 10^{-2}$
3.5	0.865	$2 \cdot 10^{-2}$
4.0	1.018	$3 \cdot 10^{-2}$
4.5	1.199	$3 \cdot 10^{-2}$
5.0	1.327	$3 \cdot 10^{-2}$
5.5	1.408	$3 \cdot 10^{-2}$
6.0	1.627	$4 \cdot 10^{-2}$

Tabelle 1: Testdatenset Aufgabe 1.2

Ich möchte zunächst einen Zusammenhang der Form  $I = U/R + I_0$  fitten. Dazu können wir die obige Formel verwenden und a mit  $I_0$  und b mit  $1/R$

identifizieren. Den Fehler für R müssen wir aber neu berechnen, und erhalten wenn wir das tun

$$\sigma_R^2 = \frac{S D^2}{(S1, 1S_{x,y} - Sx, 1S_{y,1})^4}$$

Es ergeben sich danach folgende Werte für die Parameter

R	$I_0$	$\sigma_R$	$\sigma_{I_0}$	$\chi^2$	$\chi^2/(N-2)$
3.630	-0.066	0.045	0.008	17.883	1.788

Tabelle 2: Fitparamter für die Funktion  $I = U/R + I_0$  für die  $N = 12$  Messwerte

Für den Fall, dass wir  $a = 0$  wählen, berechnet sich der Widerstand als

$$R = \frac{S_{x,x}}{S_{x,y}}$$

und damit

$$\sigma_R^2 = \frac{S_{x,x}^3}{S_{x,y}^4}$$

R	$\sigma_R$	$\chi^2$	$\chi^2/(N-2)$
3.934	0.033	81.960	8.196

Tabelle 3: Fitparamter für die Funktion  $I = U/R$  für die  $N = 12$  Messwerte

Aus dem Ergebnis für das normierte  $\chi^2$  kann man schließen, dass beide Anpassungen nicht gut genug sind, da vermutlich zu wenige Messwerte vorliegen. Die affin-lineare Anpassungsfunktion schneidet hier jedoch besser ab.

Als nächstes erweitern wir das Regressionsmodell ein wenig, indem wir Linearkombinationen von Funktionen fitten und die Linearkoeffizienten bestimmen. Dazu betrachten wir eine Winkelverteilung einer Größe.

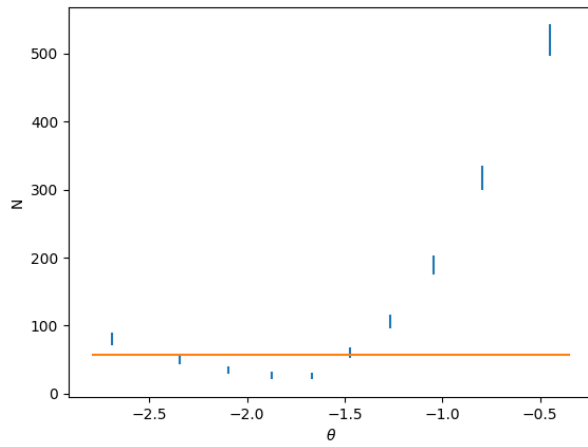
$\cos \theta$	$N$	$\sigma_N$
-0.9	81	9.00
-0.7	50	7.07
-0.5	35	5.92
-0.3	27	5.20
-0.1	26	5.10
0.1	60	7.75
0.3	106	10.30
0.5	189	13.75
0.7	318	17.83
0.9	520	22.80

Tabelle 4: Testdatenset Aufgabe 1.3

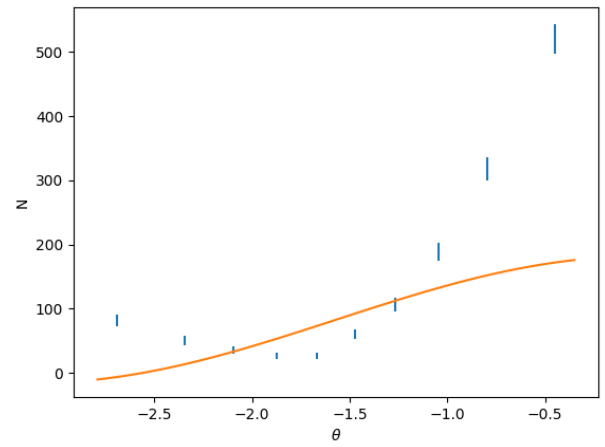
Zunächst fitte ich ein Polynom  $n$ . Grades in  $\cos \theta$ , wobei ich  $n \in \{0, \dots, 9\}$ , da 10 Parameter bei 10 Messpunkten zu fitten definitiv ein overfit darstellt. Ich nehme die Anpassung wie zuvor vor, indem ich das zugehörige lineare Gleichungssystem löse. Es ergibt sich

$n$	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$
0	57.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	82.7	99.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	47.3	186.0	273.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	37.9	126.5	312.0	137.6	0.0	0.0	0.0	0.0	0.0	0.0
4	39.6	119.1	276.5	151.9	52.6	0.0	0.0	0.0	0.0	0.0
5	39.9	121.4	273.2	136.6	56.9	16.7	0.0	0.0	0.0	0.0
6	38.4	130.3	332.7	85.8	-186.1	68.3	216.9	0.0	0.0	0.0
7	39.2	139.5	313.0	-39.7	-115.6	434.2	157.7	-279.0	0.0	0.0
8	38.3	145.9	380.5	-117.2	-678.5	646.8	1,491.4	-435.9	-914.1	0.0
9	40.0	179.0	302.8	-958.6	-110.4	5,723.3	223.8	-10,797.1	-72.7	6,561.0

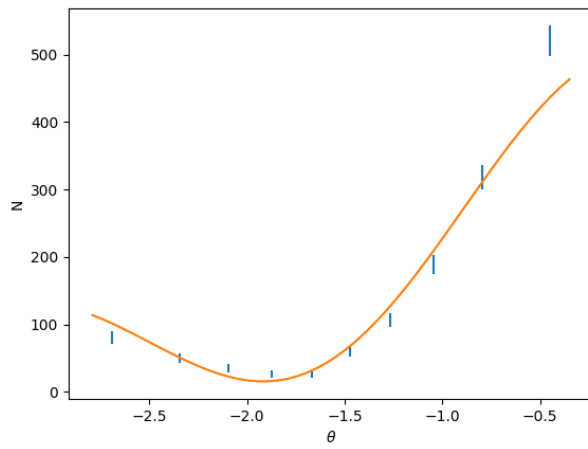
Tabelle 5: Fittparamter für die Polynome in  $\cos \theta$



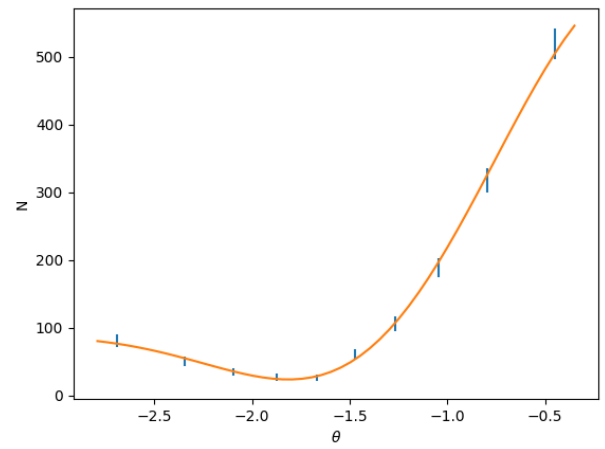
(a) Grad 0



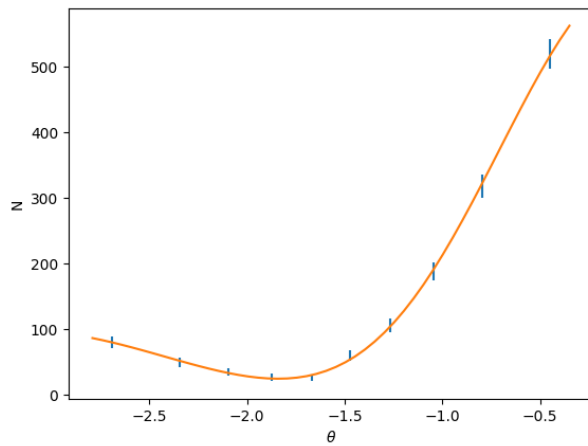
(b) Grad 1



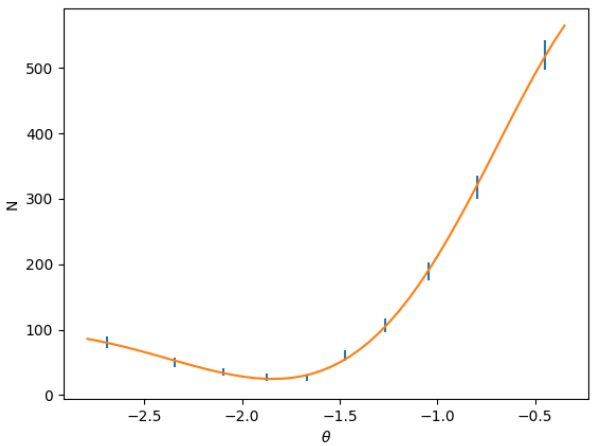
(c) Grad 2



(d) Grad 3

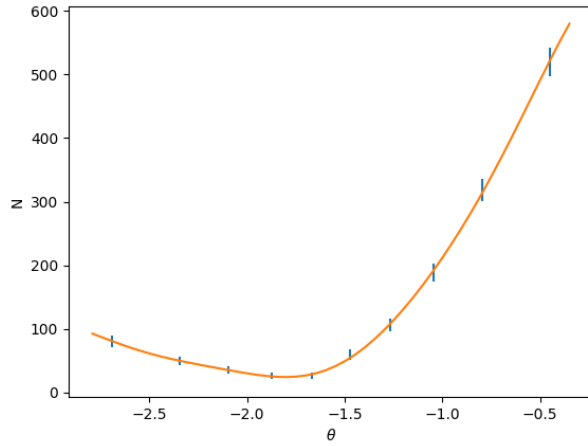


(e) Grad 4

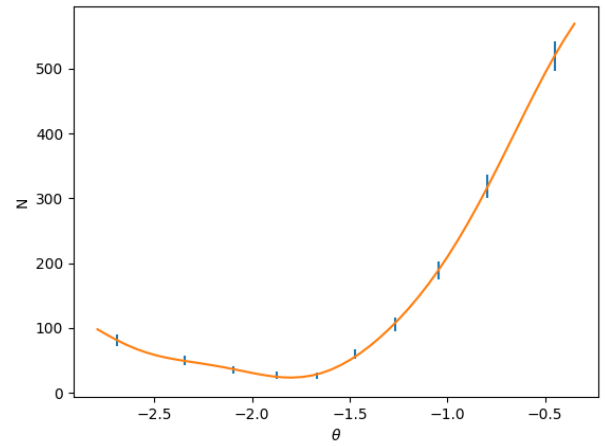


(f) Grad 5

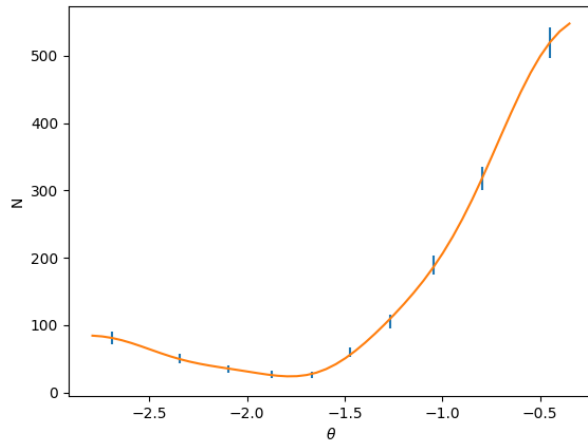
Abbildung 1: Darstellung der Regression durch n. Grad eines Polynoms  
 $\sum_{i=0}^n a_i \cos^i \theta$



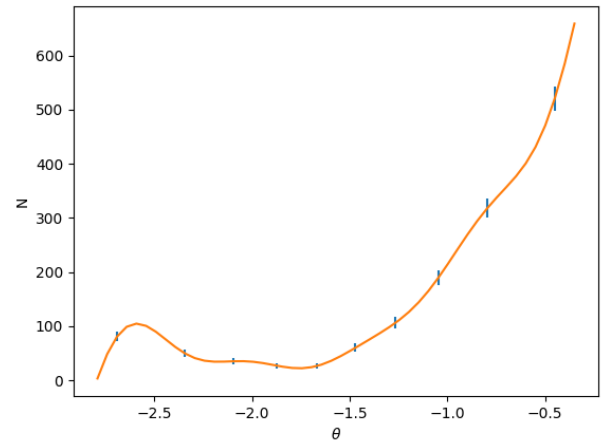
(a) Grad 6



(b) Grad 7



(c) Grad 8



(d) Grad 9

Abbildung 2: Darstellung der Regression durch n. Grad eines Polynoms  
 $\sum_{i=0}^n a_i \cos^i \theta$

$n$	$\chi^2$	$\chi^2/(10 - (n + 1))$
0	$8.34 \cdot 10^2$	92.6
1	$5.85 \cdot 10^2$	73.2
2	$3.64 \cdot 10^1$	5.2
3	$2.85 \cdot 10^0$	0.475
4	$1.69 \cdot 10^0$	0.337
5	$1.66 \cdot 10^0$	0.416
6	$9.35 \cdot 10^{-1}$	0.312
7	$7.68 \cdot 10^{-1}$	0.384
8	$5.66 \cdot 10^{-1}$	0.566
9	$1.26 \cdot 10^{-20}$	inf

Tabelle 6:  $\chi^2$  sowie reduziertes  $\chi^2$  für die Polynome in  $\cos \theta$

Eine äquivalente Untersuchung lässt sich anstellen, indem ich eine Linearkombinationen von Legendre Polynomen in  $\cos \theta$  an die Daten fitte. Da die Legendre Polynome n. Grads sind Polynome n. Grads, und da die zu lösenden Gleichungssysteme eindeutige Lösungen besitzen, was im Rahmen des Algorithmus zu deren Lösung ermittelt wurde, handelt sich bei den angepassten Legendre Polynomen um dieselben Funktionen wie zuvor. Die Linear Koeffizienten berechnete ich als

$n$	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$
0	57.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	82.7	99.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	138.5	186.0	182.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	142.0	209.1	208.0	55.0	0.0	0.0	0.0	0.0	0.0	0.0
4	142.3	210.2	214.4	60.8	12.0	0.0	0.0	0.0	0.0	0.0
5	142.3	210.5	214.6	62.1	13.0	2.1	0.0	0.0	0.0	0.0
6	143.1	211.1	218.7	64.7	25.1	8.7	15.0	0.0	0.0	0.0
7	142.9	208.8	217.7	58.8	22.7	-2.1	10.9	-10.4	0.0	0.0
8	140.9	207.5	206.8	55.7	2.9	-7.3	-14.9	-16.3	-18.2	0.0
9	142.7	247.0	216.0	149.0	20.1	127.0	6.1	115.5	-1.4	69.1

Tabelle 7: Fittparamter für angepasste Funktion  $\sum_{i=0}^n a_i P_i(\cos \theta)$



$n$	$\chi^2$	$\chi^2/(10 - (n + 1))$
0	8.34e+02	92.6
1	5.85e+02	73.2
2	36.4	5.2
3	2.85	0.475
4	1.69	0.337
5	1.66	0.416
6	0.935	0.312
7	0.768	0.384
8	0.566	0.566
9	7.8e-27	inf

Tabelle 8:  $\chi^2$  sowie reduziertes  $\chi^2$  für die Polynome in  $\cos \theta$

Anhand des reduzierten  $\chi^2$  lässt sich auch noch einmal die Äquivalenz der beiden Basen erkennen.

Als letzte Aufgabe dieses Abschnitts sollte ein Zerfallsprozess von Silber untersucht werden. Es soll angenommen werden, dass sich der zeitliche Verlauf an zerfallenden Teilchen der Sorte 1 und 2, wie folgt beschreiben lässt.

$$N(t) = N_1 \exp(-\lambda_1 t) + N_2 \exp(-\lambda_2 t)$$

Daraus ergibt sich die Anzahl an Zerfällen in einem Zeitintervall  $(t, t + \Delta t)$  als

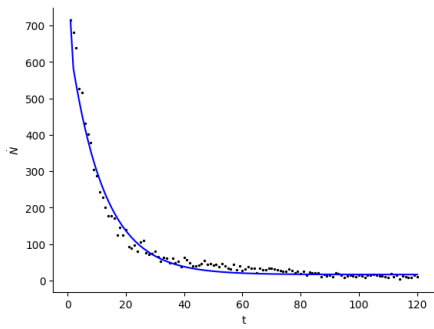
$$\begin{aligned} A(t) &= N(t) - N(t + \Delta t) + A_{\text{nat}} \\ &= N_1 \exp(-\lambda_1 t) (1 - \exp(-\lambda_1 \Delta t)) + N_2 \exp(-\lambda_2 t) (1 - \exp(-\lambda_2 \Delta t)) + A_{\text{nat}} \end{aligned}$$

Diese Funktion passte ich mit verschiedenen Verfahren an die gegebene Datenreihe an. Ich versuchte ein Downhill simplex Algorithmus nach Nelder und Mean. Diesen Algorithmus wandte ich zum einen nach einer Implementierung in den Numerical Recipes und zum anderen durch die minimize Funktion im Scipy Modul von Python an. Als zweiten Algorithmus versuchte ich den Conjugate Gradient Algorithmus wieder aus den Numerical Recipes und aus Scipy. Beide Implementierungen fanden für das gegebene Problem keine Lösung. Allerdings fand der Newton-CG Algorithmus aus Scipy eine gute Lösung, wie nachfolgender Tabelle entnommen werden kann.

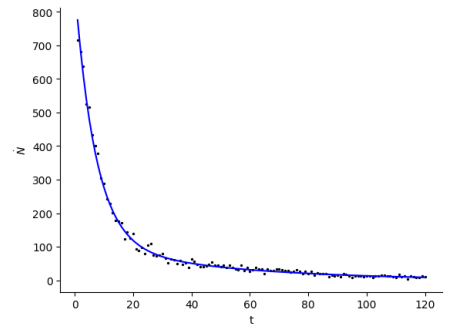
Type	$N_1$	$\lambda_1$	$N_2$	$\lambda_2$	$A_{\text{nat}}$	$\chi^2$
simplex recipes	7491.150757	0.017192	81.765198	4.840988	16.202220	483.066
simplex scipy	5046.085	2.828e-02	5225.919	4.258e-03	1.956e-12	134.157
Newton CG scipy	5010.452	4.492e-03	4995.188	2.836e-02	1.324	135.979

Tabelle 9: Ergebnisse der Anpassung für verschiedene Modelle

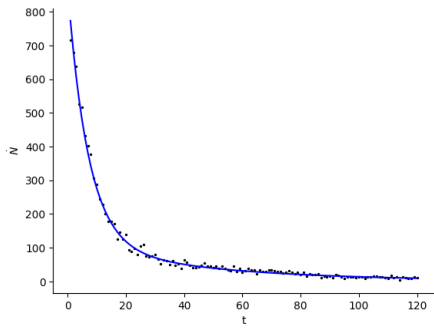
Es ist anzumerken dass sich die Lösungen der verschiedenen Verfahren relativ deutlich unterscheiden, was auf eine geringe der Lösungen schließen lässt oder zumindest auf eine Uneindeutigkeit. Das minimale  $\chi^2$  sind jedenfalls für die beiden Scipy Algorithmen ähnlich. Und auch die graphische Darstellung der beiden Lösungen sind nicht zu unterscheiden.



(a) simplex Algorithmus der Numerical Recipes



(b) Nelder-Mead Algorithmus von Scipy



(c) Newton Conjugate Gradient Algorithmus von Scipy

Abbildung 3: Darstellung der Regression durch die verschiedenen Optimierungsverfahren

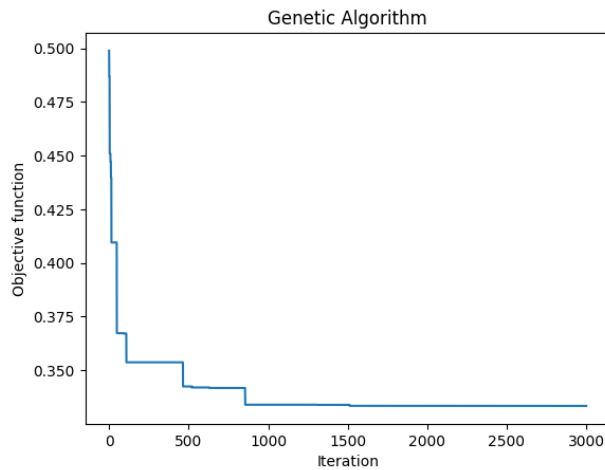
## 2 Evolutionäre Algorithmen

In diesem Abschnitt wird drei kurzes Anwendungsbeispiele für genetische Algorithmen vorgestellt. Zunächst soll das Maximum der Funktion

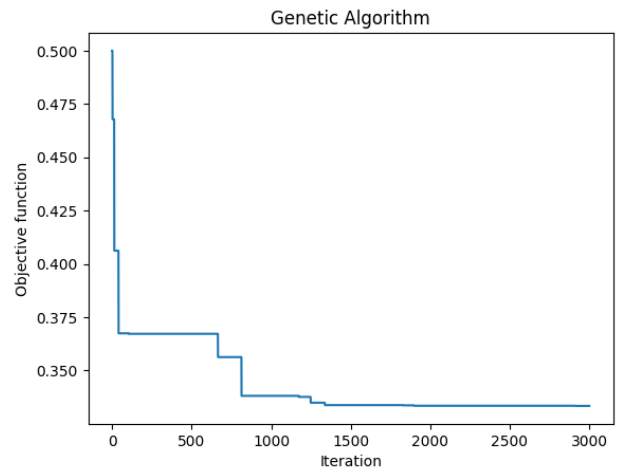
$$f(x, y) = \cos(9\pi r) \exp(-r^2/0.15) \quad \text{wobei } r^2 = (x - 0.5)^2 + (y - 0.5)^2$$

Der genetische Algorithmus des Python Moduls `geneticalgorithm` fand hier ein Maximum bei ungefähr  $(x, y) = (0.5, 0.5)$ . Für verschiedene Mutationsraten war der Algorithmus im Grunde gleich erfolgreich.

Mit einer Mutationsrate von 0.6 bestimmte er das Maximum auf 0.50004382, 0.49982225 und bei einer Mutationsrate von 0.1 auf 0.50003418, 0.49917879.



(a) MR = 0.6



(b) MR = 0.1

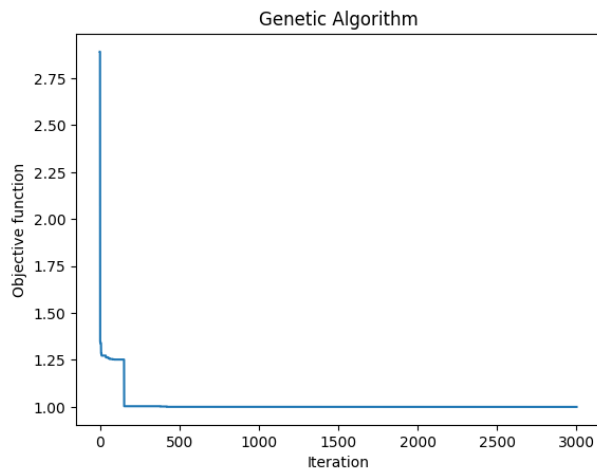
Abbildung 4: Vortschrittsverlauf für die beiden Mutationsraten

Als zweite Funktion sollte

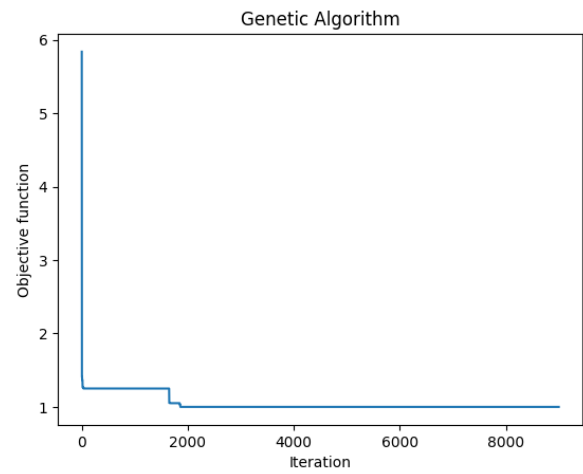
$$f(x, y) = 0.8 \exp(-r_1^2/0.3^2) + 0.879008 \exp(-r_2^2/0.03^2)$$

mit  $r_1^2 = (x-0.5)^2 + (y-0.5)^2$  und  $r_2^2 = (x-0.6)^2 + (y-0.1)^2$  gefittet werden.

Mit einer Mutationsrate von 0.5 wurde das Maximum auf 0.599861, 0.10051741 bestimmt, für MR=0.1 fand der Algorithmus das Maximum bei 0.59979587 0.10057644.



(a) MR = 0.5



(b) MR = 0.1

Abbildung 5: Vortschrittsverlauf für die beiden Mutationsraten

Zulezt fitte ich noch an ein Datenset, das eine Lichtkurve representieren soll, eine Reihe an sinus-Funktionen, addiert mit einer Geraden-Funktion.

$$f_{\text{model}} = a x + b + \sum_{m=0}^n A_m \sin\left(\frac{2\pi}{T_m} x + \phi_m\right)$$

Die Standardabweichung der Datenpunkte soll 5 betragen. Wenn man sich den Datenverlauf anschaut, ekennt man ein periodisches Verhalten, dessen prominente Frequenz schon durch den Fit mit einer sinus Funktion eingefangen wird.

m	a	b	$A_m$	$T_m$	$\phi_m$
1	1.003	19.137	11.374	20.118	3.305

Tabelle 10: Parameter der Lösung der für n=1. Der beste Funktionswert wurde als 769.970 gefunden

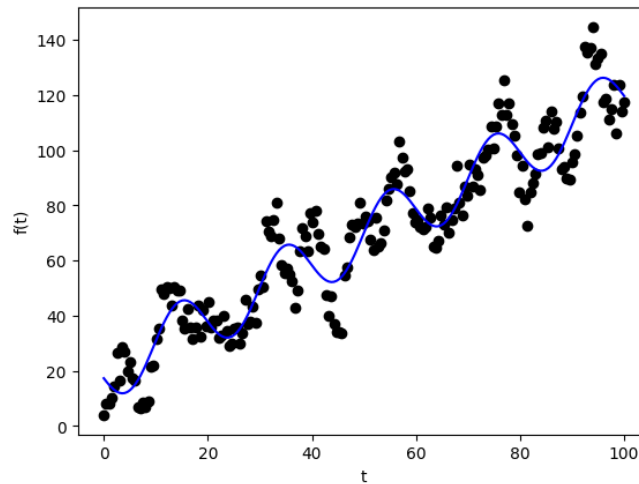


Abbildung 6: Datenpunkte mit der Regression durch 1 sinus Funktion

Wenn 2 sinus Funktionen gefittet werden, erkennt man schon eine deutlich bessere Anpassung an die Daten.

m	a	b	$A_m$	$T_m$	$\phi_m$
1	1.065	15.917	11.139	20.097	3.242
2			10.039	8.935	4.796

Tabelle 11: Parameter der Lösung der für n=2. Der beste Funktionswert wurde als 395.124 gefunden

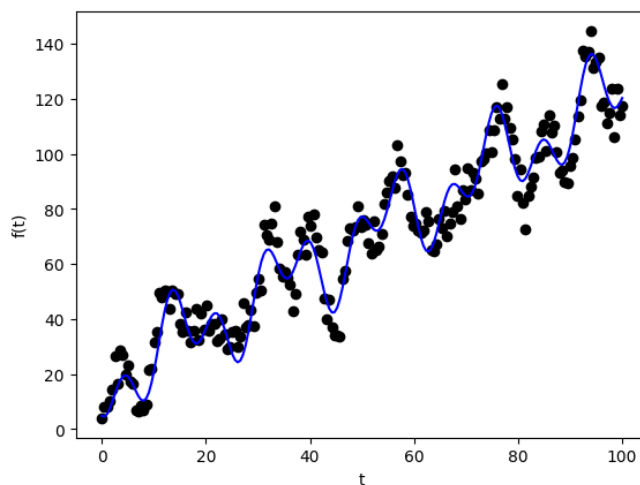


Abbildung 7: Datenpunkte mit der Regression durch 2 sinus Funktionen

Und bei drei sinus Funktionen wird die Anpassung noch einmal besser.

m	a	b	$A_m$	$T_m$	$\phi_m$
1	1.001	19.304	9.832	8.995	5.016
2			11.592	20.135	3.280
3			6.575	7.531	5.330

Tabelle 12: Parameter der Lösung der für n=3. Der beste Funktionswert wurde als 181.261 gefunden

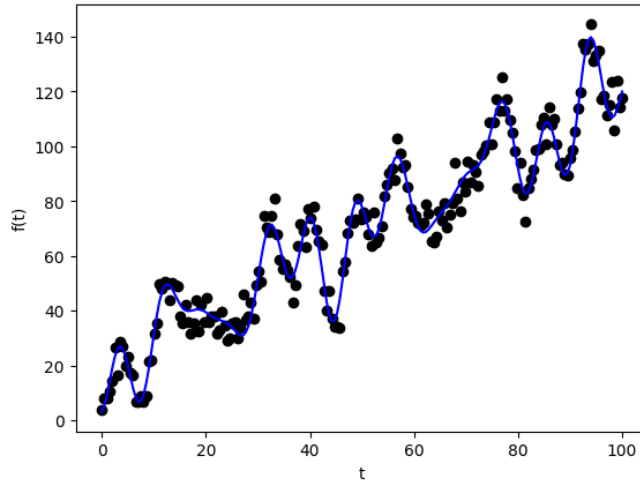


Abbildung 8: Datenpunkte mit der Regression durch 3 sinus Funktionen

Für die Anpassung mit 4 sinus Funktionen sind die Ergebnisse nicht mehr so schön. Die Fitfunktion liegt zwar eng an den Daten an und auch das  $\chi^2$  ist klein, allerdings ist die Periode der ersten sinus Funktion ungefähr so groß wie, der t-Abstand der Datenpunkte, was keinen großen Sinn ergibt. Außerdem wurden zwei sinus Funktionen mit sehr ähnlichen Frequenzen angenähert, wobei die Amplitude der ersten viel kleiner ist als die andere. Sie trägt daher nur wenig zur Anpassung bei, was darauf hindeuten kann, dass sie nicht wirklich eine Eigenschaft der Daten wiedergibt, sondern sich in einem Bereich bewegt der durch die Fehler wenig Aussagekraft ermöglicht.

m	a	b	$A_m$	$T_m$	$\phi_m$
1	1.017	18.672	6.694	0.538	4.154
2			11.454	20.388	3.462
3			1.960	8.484	6.177
4			10.826	8.911	4.679

Tabelle 13: Parameter der Lösung der für n=4. Der beste Funktionswert wurde als 182.922 gefunden

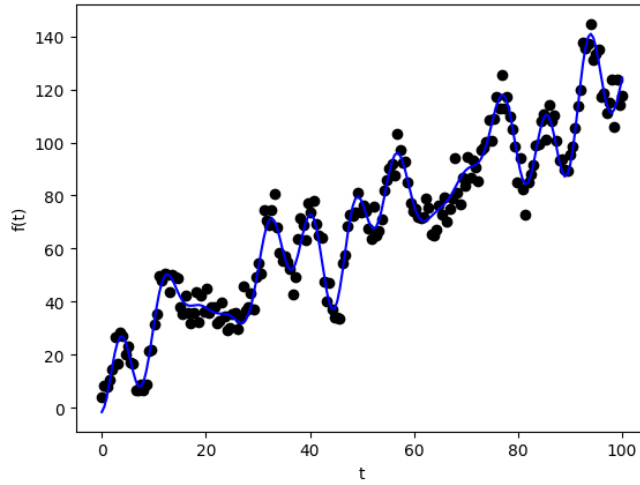


Abbildung 9: Datenpunkte mit der Regression durch 4 sinus Funktionen

Beim Ergebnis für 5 angepasste sinus Funktionen erkennen wir ungefähr dieselben Frequenzen wie bei der Anpassung mit 3 sinus Funktionen wieder. Es sind noch zwei Funktionen hinzugekommen mit ungefähr gleicher Amplitude und Frequenz aber Phasendifferenz von ungefähr  $\pi$ . Diese heben sich also gegenseitig weg, und tragen nicht zur Lösung bei. Die letzten beiden Ergebnisse zeigten eindeutige schwächen, die schließen lassen, dass der Fit mit 3 sinus Funktionen die Daten am besten representiert. In der Tat wurden die Daten auch mit 3 sinus Funktionen erzeugt.

m	a	b	$A_m$	$T_m$	$\phi_m$
1	1.028	17.902	13.651	25.323	0.635
2			9.636	9.039	5.149
3			6.644	7.436	4.796
4			11.098	19.899	3.078
5			12.635	25.593	3.997

Tabelle 14: Parameter der Lösung der für  $n=5$ . Der beste Funktionswert wurde als 176.626 gefunden



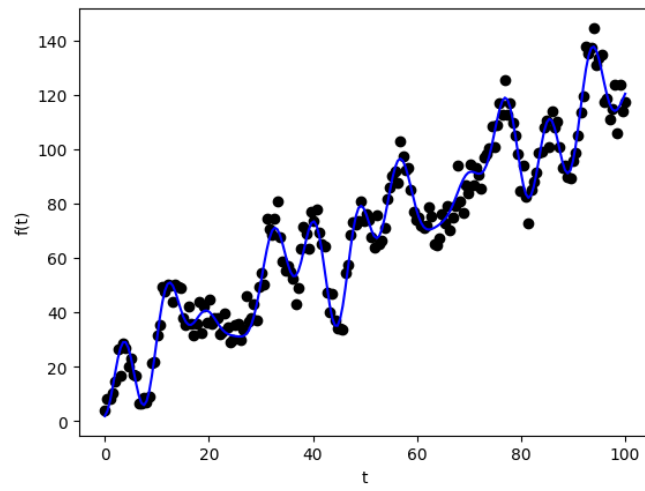


Abbildung 10: Datenpunkte mit der Regression durch 5 sinus Funktionen

### 3 Bewegung von Elektronen durch Materie

Wir unterscheiden als Wechselwirkung der Elektronen mit der penetrierten Materie zwischen zwei verschiedenen Streuungstypen, Mott Streuung und Møller Streuung. Der differentielle Wirkungsquerschnitt für Mottstreuung hat folgende Form.

$$\left(\frac{\partial\sigma}{\partial\vartheta}\right)_{\text{Mott}} \propto \frac{1}{T} \left( \frac{1}{\sin^4 \frac{\vartheta}{2}} - \beta \frac{1}{\sin^2 \frac{\vartheta}{2}} \right)$$

Hier streut das Elektron am schweren Kern des Materiestoffes. Da der Kern wesentlich schwerer als das Elektron. Ist das Schwerpunktsystem in guter Näherung das Laborsystem in dem der Kern ruht. Für Møller Streuung von Elektronen mit Elektronen ist dies nicht der Fall.

$$\left(\frac{\partial\sigma}{\partial\vartheta}\right)_{\text{Møller}} = \frac{\alpha^2}{4E_{\text{CM}}^2 \sin^4 \vartheta} (3 + \cos^2 \vartheta)^2$$

Hier müssen wir ins Schwerpunktsystem und wieder zurück transformieren. Dabei interessiert uns vor allem die Änderung der kinetischen Energie T sowie der Streuwinkel im Laborsystem. Um ins Schwerpunktsystem zu transformieren, müssen wir die Boost Geschwindigkeit so wählen, dass Schwerpunktschwindigkeiten der beiden Elektronen a, b gilt

$$v_{a,\text{CM}} = -v_{b,\text{CM}} \frac{\gamma_B (\gamma_a v_a - V_B \gamma_a)}{\gamma_B (\gamma_a - \frac{V_B \gamma_a v_a}{c^2})} = -\frac{\gamma_B (-V_B \gamma_b)}{\gamma_B \gamma_b}$$

wobei  $\gamma$  der Lorentzfaktor für die jeweilige Geschwindigkeit ist ( $v_b = 0$ ). Daraus erhält man

$$V_B = \frac{c^2}{v_a} \left( 1 - \sqrt{1 - \frac{v_a^2}{c^2}} \right)$$

Das Verfahren sieht nun so aus, dass ich zuerst das Labor system so drehe, dass der Impuls in Richtung z-Achse zeigt, danach den Impuls nach obiger Transformation ins Schwerpunktsystem transformiere, den Winkel gemäß des Streuvorgangs verändere, und dann wieder zurück gedrehte Laborsystem transformiere, von dem ich zum Schluss wieder ins ursprüngliche System wechsele. Um das Koordinatensystem so zu drehen, dass die z-Achse danach in Richtung eines Vektors

$$\hat{e}_r = \begin{pmatrix} \sin \vartheta \cos \varphi \\ \sin \vartheta \sin \varphi \\ \cos \vartheta \end{pmatrix}$$

zeigt, muss folgende Koordinatentransformation durchgeführt werden.

$$\begin{aligned}\mathbf{x}' &= R_y(\vartheta) \cdot R_z(-\varphi) \cdot \mathbf{x} \\ &= \begin{pmatrix} \cos \vartheta & 0 & -\sin \vartheta \\ 0 & 1 & 0 \\ \sin \vartheta & 0 & \cos \vartheta \end{pmatrix} \begin{pmatrix} \cos \vartheta & \sin \vartheta & 0 \\ -\sin \vartheta & \cos \vartheta & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{x}\end{aligned}$$

Für die Rücktransformation muss folglich mit  $R_z(\varphi) R_y(\vartheta)$  multipliziert werden. Es müssen natürlich alle oben beschriebenen Schritte explizit ausgeführt werden. Im Programm besitze ich die aktuelle Orientierung des Impulsvektors des Elektrons, ich bestimme den neuen Winkel  $\vartheta$  im gedrehten Laborsystem durch die oben beschriebene Lorentz-Schwerpunktstransformation und Rücktransformation und einen beliebigen Winkel  $\varphi$ . Diese Richtung lässt sich leicht im gedrehten Koordinatensystem darstellen durch den Vektor  $\hat{e}_r$  und danach kann ich mit Winkeln der ursprünglichen Orientierung des Impulses wieder ins feste Laborsystem zurücktransformieren.

Der Algorithmus hat folgende generelle Form:

- zunächst wird eine gewisse freie Weglänge bestimmt
- danach wird ausgewählt, ob Mott oder Møller Streuung stattfinden soll. In diesem Programm soll vereinfachend angenommen werden, dass beide Streuarten gleichhäufig auftreten.
- gemäß der gewählten Streuart wird ein Streuwinkel bestimmt.
- Es werden Impuls und kinetische Energie aktualisiert.
- und es wird dann ein neuer Durchgang gestartet, in dem immer nach Bestimmung der freien Weglänge die Position des Elektrons aktualisiert wird.

In diesem Programm müssen einige Werte nach einer gewissen Wahrscheinlichkeitsverteilung gezogen werden. Deshalb stellt sich die Frage wie man eine aus einer Gleichverteilung eine Verteilung generiert die einer gewissen Wahrscheinlichkeitfunktion  $p(x)$   $x \in D$  (Definitions Menge) entspricht. Die Wahrscheinlichkeit einen Wert im Intervall  $(a, b) \subseteq D$  zu ziehen, ist

$$\int_a^b p(x) dx = \frac{F_p(b) - F_p(a)}{\int_D p(x) dx}$$

mit der Stammfunktion von  $p$   $F_p$ . Die Wahrscheinlichkeit einen Wert im Intervall  $(a, b)$  gemäß der Wahrscheinlichkeit  $p(x)$  zu ziehen ist also gleich der Wahrscheinlichkeit einen Wert im Intervall  $(F_p(a), F_p(b))$  gleichverteilt zu ziehen. Dies führt sich infinitesimal dazu fort, dass die Wahrscheinlichkeit den Wert  $F_p(x)$  zu ziehen jener entspricht den Wert  $x$  nach der Wahrscheinlichkeitsverteilung  $p(x)$  zu ziehen. Im Fall, dass es für die Wahrscheinlichkeitsfunktion keine Stammfunktion hat, geht man so vor, dass man eine integrierbare einhüllende  $f(x)$  ( $F := \int_D f(x) dx$ ) wählt, mit deren Hilfe man nach obigem Verfahren Vorschlagswerte konstruiert und diese akzeptiert mit Verhältnis, wie nah  $p$  an dieser Stelle an  $f$  ist. Wenn man für die Akzeptanzwahrscheinlichkeit

$$p_{\text{akz}} = \frac{p(x)}{f(x)} \frac{\int_D f(x) dx}{\int_D p(x) dx} := \frac{p(x)}{f(x)} \frac{N_f}{N_p}$$

wählt erhält man für die Gesamtwahrscheinlichkeit einen Punkt  $x$  zu ziehen

$$p_{\text{ges}} = p_{\text{Vorschlag}} p_{\text{akz}} = \frac{f(x)}{N_f} dx \frac{p(x)}{f(x)} \frac{N_f}{N_p} = \frac{p(x)}{N_p} dx,$$

die gewünschte Wahrscheinlichkeit. Nach Vorstellung des Verfahrens möchte ich nun einen Blick auf die Wahrscheinlichkeitsfunktionen werfen. Zur Berechnung der Wahrscheinlichkeitsfunktion der freien Weglänge, verwenden wir, dass die Wahrscheinlichkeit in einem Volumen mit der Tiefe  $\Delta x$  zu Stoßen gleich dem Verhältnis von durch Streuquerschnitt abgedeckter Fläche zur Querschnittsfläche des Volumens ist.

$$\tilde{p}(\Delta x) = \varrho V \sigma / A = \varrho \sigma \Delta x := \lambda \Delta x$$

Wenn wir die Strecke durch die Materie nun quantisieren, erhalten wir für die Wahrscheinlichkeit,

dass das Teilchen im Wegstreckenintervall  $(n \Delta x, (n + 1)\Delta x)$  stößt

$$\begin{aligned} p(n\Delta x, (n + 1)\Delta x) &= : p(x = n\Delta x) dx \\ &= \frac{1}{N(\text{range})} \Pi_n (1 - p(\Delta x)) p(\Delta x) \\ &= \frac{1}{N(\text{range})} \left(1 - \frac{\lambda n \Delta x}{n}\right)^n p(\Delta x) \\ &= \frac{1}{N(\text{range})} \left(1 - \frac{\lambda x}{n}\right)^n \lambda \Delta x, \end{aligned}$$

wobei  $N(range)$  eine vom erlaubten Bereich von  $x$  abhängt. Wenn wir nun  $\Delta x \rightarrow dx$ ;  $n \rightarrow \infty$  gehen lassen erhalten wir

$$p(x) dx = \lambda e^{-\lambda x} dx$$

und mit Normierung für  $x \in (0, \infty)$

$$p(x) dx = \frac{1}{\lambda} e^{-\lambda x} dx$$

Bei Betrachtung der differentiellen Streuquerschnitte für Mott und Møller Streuung stellen wir fest, dass diese nicht integrierbar sind, da das Integral über den Raumwinkel divergiert. Im Grunde genommen beinhalten diese Streuquerschnitte deshalb keine Wahrscheinlichkeitsaussage. Wenn wir diese differentiellen Streuquerschnitte dennoch verwenden wollen, müssen wir den divergenten Teil abschneiden. Ich habe die Simulation für verschiedene cut-off-Winkel durchgeführt. Außerdem wurde in diesem Algorithmus ein Energieübertrag für einen Winkel  $\theta < 0.2$  vernachlässigt, und in diesem Fall Energie und Impuls gleich belassen. Es wurde hingegen immer kontinuierlich  $T$  reduziert nach einem Verhältnis

$$\frac{dT}{dx} \propto -\frac{x}{T}.$$

Den Wert von  $\lambda$  habe ich mit Daten aus dem Internet auf  $2.5 \cdot 10^9 m^{-1}$  abgeschätzt. Da der Proportionalitätsfaktor  $a$  des CSDA allerdings auch nicht gegeben ist und die cut-off-Winkels auch willkürlich ist, belasse ich es hier bei einer unspezifischen Simulation, bei der nur die Form der Elektronenbahn eingefangen werden soll. Als kinetische Startenergie wurde  $T = 20 MeV$  verwendet.

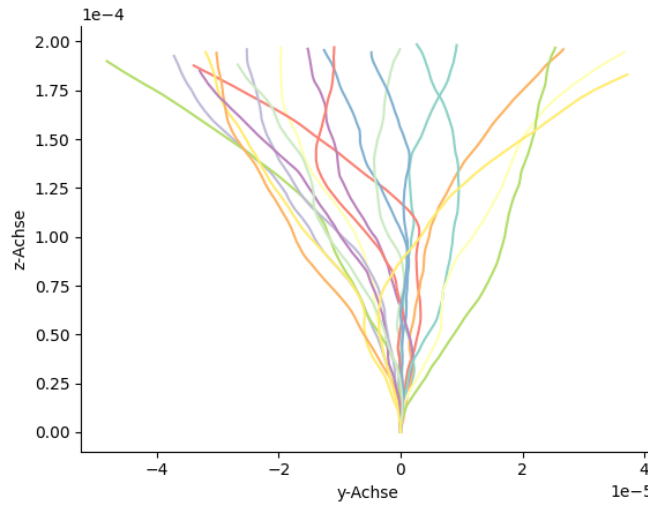


Abbildung 11: 2D Schnitt der Elektronen Bahn für cut-off-Winkel  $\theta_{\text{cut}} = 0.0004$ ,  $a = 10^6$

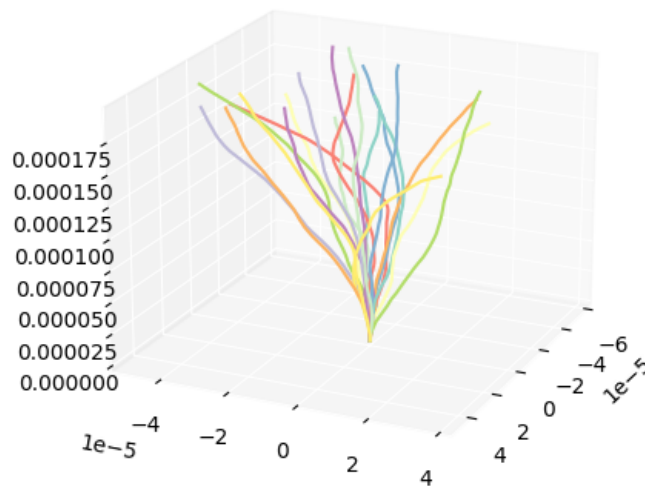


Abbildung 12: 3D Schnitt der Elektronen Bahn für dieselben Parameter

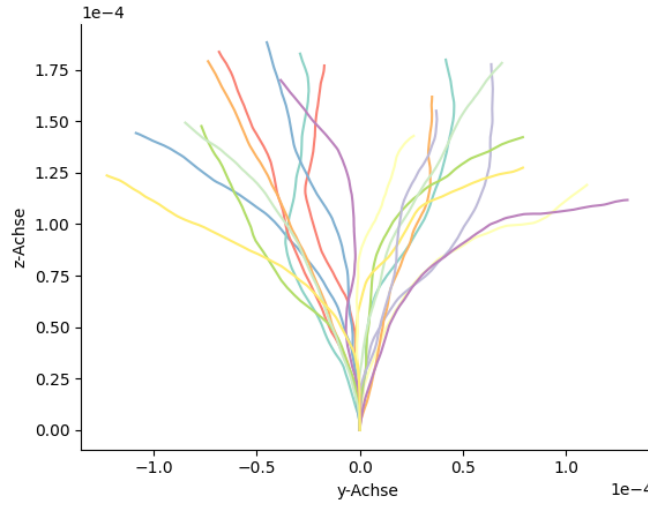


Abbildung 13: 2D Schnitt der Elektronen Bahn für cut-off-Winkel  $\theta_{\text{cut}} = 0.0008$ ,  $a = 10^6$

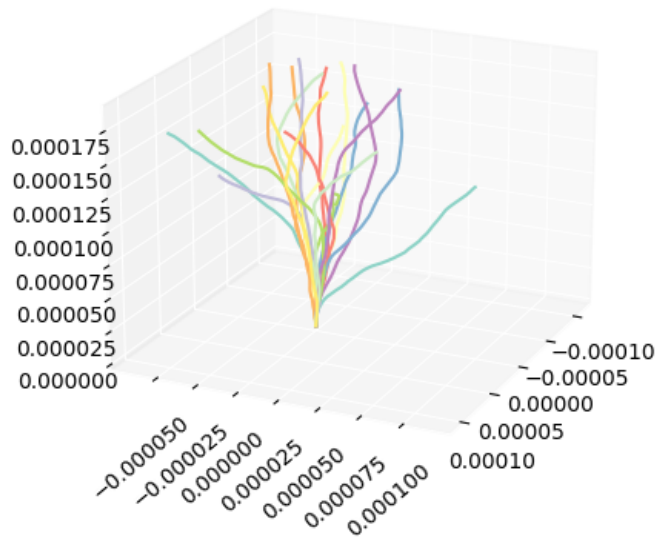


Abbildung 14: 3D Schnitt der Elektronen Bahn für dieselben Parameter

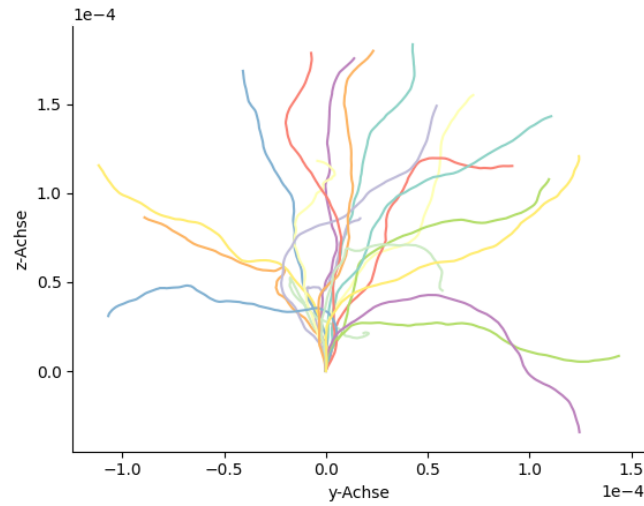


Abbildung 15: 2D Schnitt der Elektronen Bahn für cut-off-Winkel  $\theta_{\text{cut}} = 0.0016$ ,  $a = 10^6$

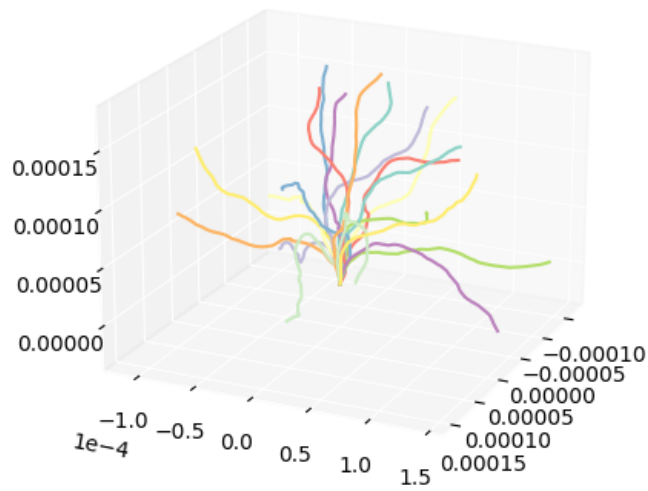


Abbildung 16: 3D Schnitt der Elektronen Bahn für dieselben Parameter

Wie in den Schaubildern zu erkennen, hängt das Maß an Ablenkung der Elektronen logischer Weise vom cut-off-Winkel ab. Es lässt sich noch anmer-



ken, dass als prägnanter Unterschied zur Anmutung echter Elektronenbahnen auffällt, dass es keine "Verzweigungen" gibt. Das lässt sich dadurch erklären, dass in unserer Simulation die Elektronen völlig unabhängig von einander sind, und jeweils ein vollkommen unberührtes Medium vorfinden. Es gibt hier keine Effekte, die mit einer Veränderung des Mediums durch ein vorheriges Elektron hervorgerufen werden.

## 4 Fazit

Wir machten uns in dieser Aufgaben Serie an mehreren Beispielen mit einigen Optimierungsverfahren sowie deren Schwächen vertraut. Es wurde für mich deutlich, dass das Scipy modul ein wirklich starkes und einfach handhabbares Werkzeug ist. Auch die Verwendung von evolutionären Algorithmen brachte akzeptable Ergebnisse hervor. Im zweiten Teil Simulierten wir die Bewegung von Elektronen durch Materie, was aufgrund der notwendigen räumlichen Vorstellung für die Koordinatendrehung und die Wiederholung relativistischer Beziehungen Spaß machte. Am Ende war auch das Ergebnis zufriedenstellend, und die erzeugte Elektronenbahn hatte die erwartete Form.