

Vorversuch

November 6, 2023

#Fakultät für Physik

0.1 Physikalisches Praktikum P1 für Studierende der Physik Praktikumsvorversuch

1 Datenverarbeitung am Beispiel des Pendels

Name: *surname1* Vorname: *name1* E-Mail: *mail1*

Name: *surname2* Vorname: *name2* E-Mail: *mail2*

Gruppennummer: *groupnumber*

Betreuer: Valentin

Versuch durchgeführt am: *date*

Beanstandungen:

Testiert am: _____ Testat: _____

```
[4]: # Hilfsfunktionen. Nicht weiter beachten
from IPython.core.display import Markdown
def M_helper_func(txt):
    #print('fr'+txt.replace("{","{{").replace("}","}}").replace("!", "{").
    ↪replace("!", "}")+'')
    excreturnvalue = ""
    loc = {}
    exec('excreturnvalue = fr"<div style=\'margin-left: 8px;\>'+txt.
    ↪replace("{","{{").replace("}","}}").replace("!", "{").replace("!", "}")+'</
    ↪div>', globals(), loc)
    display(Markdown(loc["excreturnvalue"]))
    return loc["excreturnvalue"]

from IPython.core.magic import (register_line_magic,
                                register_cell_magic)
@register_line_magic
def m(line):
    M_helper_func(line)

import os
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
import scipy as sci
import kafe2
mpl.rcParams['figure.dpi'] = 100
```

2 Durchführung

2.1 Aufgabe 1: Umgang mit großen Datensätzen

Hinweise zu allen hier durchzuführenden Messungen finden Sie in der Datei [Hinweise-Aufgabe-1.md](#).

- Öffnen Sie die Dateien `data_raw.csv` und `data_down_sampled.csv` per Doppelklick und untersuchen die Spalten und Zeilen.
- Geben Sie die Größe der Dateien in MB an.
- Stellen Sie die einzelnen Spalten jeweils in einem Diagramm, als Funktion der Zeit t dar.

Verwenden Sie für alle weiteren Untersuchung die Datei `data_down_sampled.csv`.

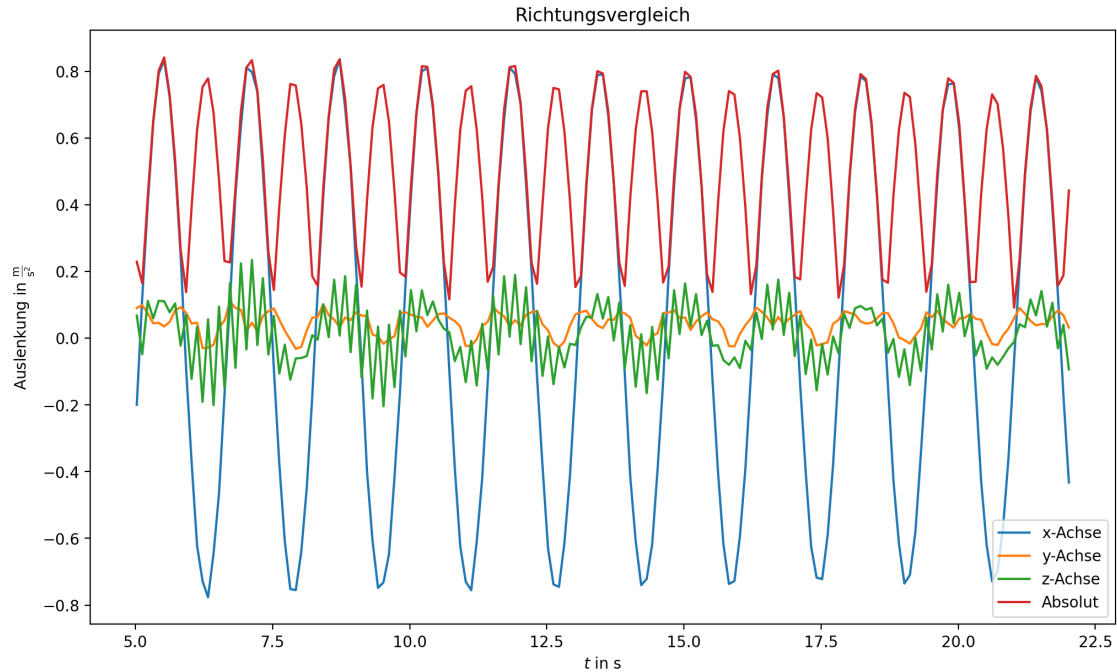
```
[5]: %m data_raw.csv: [!os.path.getsize('data_raw.csv')/10**6:.2f!] MB
      %m data_down_sampled.csv: [!os.path.getsize('data_down_sampled.csv')/10**6:.2f!
      ↪] MB
```

```
[5]: data_raw.csv: 2.43 MB
```

```
[5]: data_down_sampled.csv: 0.01 MB
```

```
[6]: data = np.genfromtxt('data_down_sampled.csv', delimiter=',')[1:]
      t,x,y,z,a = [data[:,i] for i in range(5)]
      for i in range(4):
          plt.plot(t, (x,y,z,a)[i],
          ↪label=("x-Achse", "y-Achse", "z-Achse", "Absolut")[i])
          plt.title("Richtungsvergleich")
          plt.xlabel("$t$ in s")
          plt.ylabel("Auslenkung in $\mathrm{\frac{m}{s^2}}$")
      plt.legend()
      plt.show()
```

```
[6]:
```



2.2 Aufgabe 2: Mathematische Pendel

Hinweise zu allen hier durchzuführenden Messungen finden Sie in der Datei [Hinweise-Aufgabe-2.md](#).

Zur Bestimmung von g unterlegen wir zunächst das Modell eines [mathematischen Pendels](#), woraus sich g wie folgt ableiten lässt:

$$g = \frac{4\pi^2}{T^2} \ell,$$

wobei ℓ der Länge des Pendels entspricht. Als Referenzwert für alle weiteren Messungen können Sie den Wert

$$g_{\text{exp}} = (9.809599 \pm 0.000034) \text{ m/s}^2$$

verwenden. Dieser Wert wurde aus der [Global Gravity Database des Bureau Gravimetric International \(BGI\)](#) für die Stadt Mannheim (bei $49,49^\circ$ nördlicher Breite und $8,53^\circ$ westlicher Länge auf einer Referenzhöhe von 101 m) ausgelesen.

2.2.1 Aufgabe 2.1: Referenzmessung von T

- Bestimmen Sie **einen einzelnen Wert** für die Periode T in den Daten.
- Überlegen Sie sich eine sinnvolle Unsicherheit ΔT und ermitteln Sie g mit entsprechender Unsicherheit $\Delta g^{(2.1)}$. Bestimmen Sie $\Delta g^{(2.1)}$ mit Hilfe linearer [Fehlerfortpflanzung](#) nach Gauß. Berücksichtigen Sie dabei auch die Unsicherheit $\Delta \ell$.

- Vergleichen Sie Ihr Ergebnis, im Rahmen der Unsicherheiten, mit g_{exp} .

```
[7]: maxima = sci.signal.find_peaks(x)[0]

T = t[maxima[1]]-t[maxima[0]] # Abstand der ersten beiden Maxima
ΔT = t[maxima[0]+1]-t[maxima[0]] # Abstand zwischen zwei Datenpunkten

l = 0.625
Δl = 0.0005

Δg21 = np.sqrt((8*(np.pi**2)*l*(T**-3)*ΔT)**2 + (4*(np.pi**2)*(T**-2)*Δl)**2 )
g21 = 4*np.pi**(2)*l*T**(-2)

%m $g^{(2.1)} = [!g21:.2f!] \mathrm{\frac{m}{s^2}}$
%m $\Delta g^{(2.1)} = [!Δg21:.2f!] \mathrm{\frac{m}{s^2}}$
%m Der Referenzwert liegt im Rahmen der Unsicherheit der Messung, allerdings
    ↳ ist die Unsicherheit recht groß.
%m Hier wäre es natürlich besser die Abstände aller Minima und Maxima zu
    ↳ mitteln und die Unsicherheit aus der Verteilung dieser Abstände zu bestimmen.
    ↳ So ist das Ergebnis praktisch Nutzlos.

%m (Für diese Aufgabe wurde mangels genauerer Angaben angenommen, dass die
    ↳ angegebenen Unsicherheit von $l$ sich auf eine für die Anwendung annähernd
    ↳ normalverteilte Größe bezieht. Sollte diese Annahme nicht halten ist die
    ↳ Anwendung von Gaußsche-Fehlerfortpflanzung natürlich nicht richtig.)
```

[7]: $g^{(2.1)} = 10.98 \frac{\text{m}}{\text{s}^2}$

[7]: $\Delta g^{(2.1)} = 1.46 \frac{\text{m}}{\text{s}^2}$

[7]: Der Referenzwert liegt im Rahmen der Unsicherheit der Messung, allerdings ist die Unsicherheit recht groß.

[7]: Hier wäre es natürlich besser die Abstände aller Minima und Maxima zu mitteln und die Unsicherheit aus der Verteilung dieser Abstände zu bestimmen. So ist das Ergebnis praktisch Nutzlos.

[7]: (Für diese Aufgabe wurde mangels genauerer Angaben angenommen, dass die angegebenen Unsicherheit von 0.625\$ sich auf eine für die Anwendung annähernd normalverteilte Größe bezieht. Sollte diese Annahme nicht halten ist die Anwendung von Gaußsche-Fehlerfortpflanzung natürlich nicht richtig.)

2.2.2 Aufgabe 2.2: Erste Verbesserung der Methodik

- Bestimmen Sie T aus einer Anpassung an alle Datenpunkte, bestehend aus den Wertepaaren $(t, \varphi(t))$. Berücksichtigen Sie dabei die Unsicherheiten sowohl auf t , als auch auf $\varphi(t)$. Angaben zu diesen Unsicherheiten können Sie z.B. der Datei [Datenblatt.md](#) entnehmen. Stellen Sie die Daten und das angepasste Modell geeignet graphisch dar.
- Notieren Sie die folgenden wichtigen Ausgaben der Anpassung:
 - Qualität der Anpassung (quantifiziert durch die Größe $\hat{\chi}^2/n_{\text{dof}}$)

- Die ermittelten Werte mit entsprechenden Unsicherheiten für alle an die Daten angepassten Parameter.
- Berechnen Sie aus den bestimmten Werten für T und ΔT verbesserte Abschätzungen für $g^{(2.2)}$ und $\Delta g^{(2.2)}$. Bestimmen Sie $\Delta g^{(2.2)}$ mit Hilfe linearer Fehlerfortpflanzung nach Gauß. Berücksichtigen Sie dabei auch die Unsicherheit $\Delta \ell$.
- Vergleichen Sie Ihr Ergebnis, im Rahmen der Unsicherheiten, mit g_{exp} .

```
[8]: Δt = 0.0125
Δ = 0.02

def sin_model(t, T=1.6, A=0.8, T0=0):
    return np.sin(t*2*np.pi/T+T0)*A

xy_data = kafe2.XYContainer(x_data=t, y_data=x)
xy_data.add_error(axis='x', err_val=Δt)
xy_data.add_error(axis='y', err_val=Δ)
sin_fit = kafe2.Fit(data=xy_data, model_function=sin_model)

sin_fit.do_fit()
p = kafe2.Plot(sin_fit)
p.plot()
p.show()

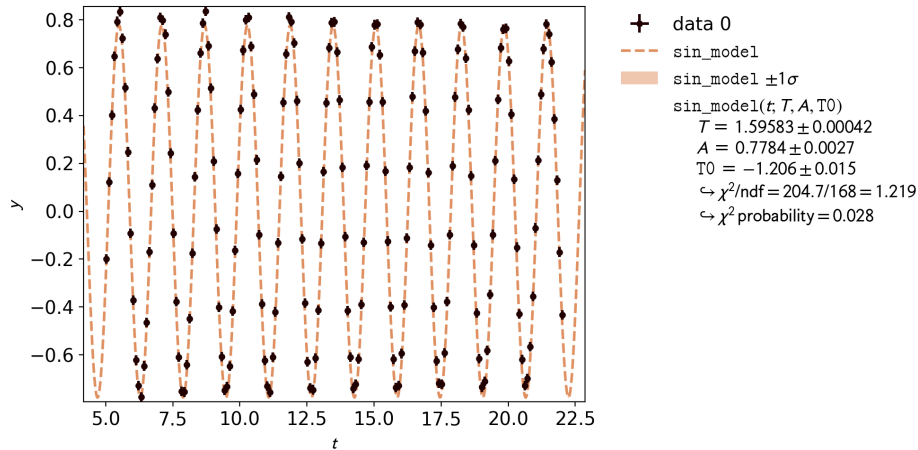
T, A, T0 = sin_fit.parameter_values
ΔT, ΔA, ΔT0 = sin_fit.parameter_errors

Δg22 = np.sqrt((8*(np.pi**2)*1*(T**3)*ΔT)**2 + (4*(np.pi**2)*(T**2)*Δl)**2)
g22 = 4*np.pi**2*1*T**(-2)

%m $g^{\{(2.2)\}}= [!g22:.2f!] \mathrm{\frac{m}{s^2}}$
%m $\Delta g^{\{(2.2)\}} = [!Δg22:.2f!] \mathrm{\frac{m}{s^2}}$
%m Die Unsicherheit ist sehr gering aber der Wert stimmt nicht im Rahmen der
    ↳ Unsicherheit mit dem Literaturwert überein. Das dürfte an der Vereinfachung
    ↳ auf ein mathematisches Pendel zurück zu führen sein. Der Fit hat einen
    ↳ $\frac{\chi^2}{\text{ndf}}$ Wert von [!sin_fit.goodness_of_fit/sin_fit.ndf:.3f!],
    ↳ was recht nah an 1 ist, also einen guten Fit darstellt.

%m (Für diese Aufgabe wurde mangels genauerer Angaben angenommen, dass die
    ↳ angegebenen Unsicherheiten von $T$, $l$ und $\varphi$ sich auf eine für
    ↳ die Anwendung annähernd normalverteilte Größe bezieht. Sollte diese Annahme
    ↳ nicht halten ist die Anwendung von Gaußsche-Fehlerfortpflanzung natürlich
    ↳ nicht richtig.)
```

[8]:



[8]: $g^{(2.2)} = 9.69 \frac{\text{m}}{\text{s}^2}$

[8]: $\Delta g^{(2.2)} = 0.01 \frac{\text{m}}{\text{s}^2}$

[8]: Die Unsicherheit ist sehr gering aber der Wert stimmt nicht im Rahmen der Unsicherheit mit dem Literaturwert überein. Das dürfte an der Vereinfachung auf ein mathematisches Pendel zurück zu führen sein. Der Fit hat einen $\frac{\chi^2}{ndf}$ Wert von 1.219, was recht nah an 1 ist, also einen guten Fit darstellt.

[8]: (Für diese Aufgabe wurde mangels genauerer Angaben angenommen, dass die angegebenen Unsicherheiten von t , l und φ sich auf eine für die Anwendung annähernd normalverteilte Größe bezieht. Sollte diese Annahme nicht halten ist die Anwendung von Gaußsche-Fehlerfortpflanzung natürlich nicht richtig.)

2.2.3 Aufgabe 2.3: Zweite Verbesserung der Methodik

- Bestimmen Sie $g^{(2.3)}$ und $\Delta g^{2.3}$ direkt aus der Anpassung. Formulieren Sie ihre Modellfunktion dazu entsprechend um, führen Sie die Anpassung erneut durch und vergleichen Sie die Ergebnisse für $g^{(2.3)}$ und $\Delta g^{(2.3)}$ mit den Ergebnissen aus Aufgabe 2.2.
- Überlegen Sie, wie Sie in diesem Fall Δl im Ergebnis von $\Delta g^{(2.3)}$ berücksichtigen können.

```
[9]: l = 0.625
      Δl = 0.0005

      def sin_model(t, g=9.81, A=0.8, T0=0, l=1):
          return A*np.cos(np.sqrt(g/l)*t+T0)

      xy_data = kafe2.XYContainer(x_data=t, y_data=x)
      xy_data.add_error(axis='x', err_val=Δt)
      xy_data.add_error(axis='y', err_val=Δ)
      sin_fit = kafe2.Fit(data=xy_data, model_function=sin_model)
```

```

sin_fit.add_parameter_constraint(name='l', value=1, uncertainty=Δl)
    ↪ #Alternative dazu, den Fit dreimal durchzuführen

sin_fit.do_fit()
p = kafe2.Plot(sin_fit)
p.plot()
p.show()

g23, A, T0, _ = sin_fit.parameter_values
Δg23, ΔA, ΔT0, _ = sin_fit.parameter_errors

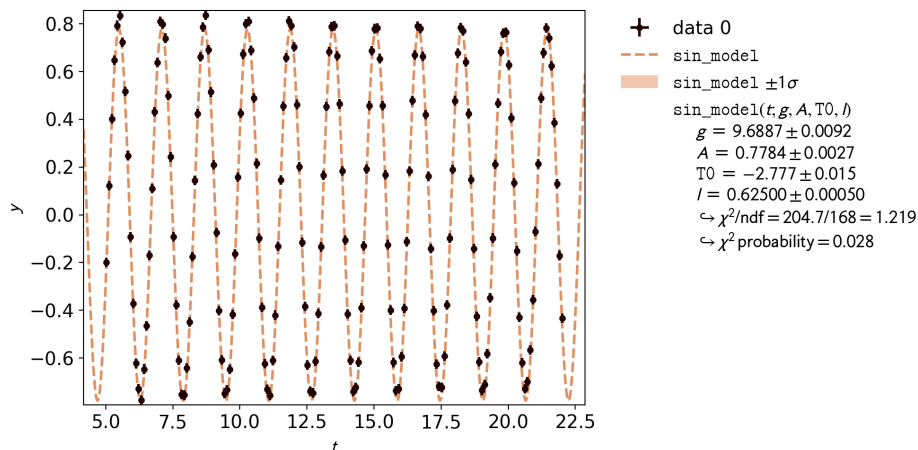
%m $g^{\{2.3\}} = [!g23:.3f!] \mathrm{\frac{m}{s^2}}$
%m $\Delta g^{\{2.3\}} = [!Δg23:.5f!] \mathrm{\frac{m}{s^2}}$
%m Der Fit hat einen $\frac{\chi^2}{ndf}$ Wert von [!sin_fit.goodness_of_fit/
    ↪ sin_fit.ndf:.3f!], was recht nah an 1 ist, also einen guten Fit darstellt.

%m Vergleich der Werte:
%m $g^{\{2.2\}} = g^{\{2.3\}} = [!g23:.3f!] \mathrm{\frac{m}{s^2}}$
%m $\Delta g^{\{2.2\}} = \Delta g^{\{2.3\}} = [!Δg23:.5f!] \mathrm{\frac{m}{s^2}}$

%m (Für diese Aufgabe wurde mangels genauerer Angaben angenommen, dass die
    ↪ angegebenen Unsicherheiten von $t$, $l$ und $\varphi$ sich auf eine für
    ↪ die Anwendung annähernd normalverteilte Größe bezieht. Sollte diese Annahme
    ↪ nicht halten ist die Anwendung von Gaußsche-Fehlerfortpflanzung natürlich
    ↪ nicht richtig.)

```

[9]:



[9]: $g^{(2.3)} = 9.689 \frac{\text{m}}{\text{s}^2}$

[9]: $\Delta g^{(2.3)} = 0.00925 \frac{\text{m}}{\text{s}^2}$

[9]: Der Fit hat einen $\frac{\chi^2}{ndf}$ Wert von 1.219, was recht nah an 1 ist, also einen guten Fit darstellt.

[9]: Vergleich der Werte:

[9]: $g^{22} = g^{23} = 9.689 \frac{\text{m}}{\text{s}^2}$

[9]: $\Delta g^{2,2} = \Delta g^{2,3} = 0.00925 \frac{\text{m}}{\text{s}^2}$

[9]: (Für diese Aufgabe wurde mangels genauerer Angaben angenommen, dass die angegebenen Unsicherheiten von t , l und φ sich auf eine für die Anwendung annähernd normalverteilte Größe bezieht. Sollte diese Annahme nicht halten ist die Anwendung von Gaußsche-Fehlerfortpflanzung natürlich nicht richtig.)

2.3 Aufgabe 3: Schrittweise Erweiterung des Modells

Hinweise zu allen hier durchzuführenden Messungen finden Sie in der Datei [Hinweise-Aufgabe-3.md](#).

Eine offensichtliche Unzulänglichkeit des vorherigen Modells besteht in der Vernachlässigung der endlichen Ausdehnung des Pendels. Wenn Sie das Modell entsprechend erweitern, nimmt die Formel zur Bestimmung von g die folgende Form an:

$$g = \frac{4\pi^2}{T^2} \frac{\Theta}{Ms},$$

wobei Θ und M dem Trägheitsmoment und der Masse der gesamten Pendelkonstruktion (einschließlich aller Montageteile und Smartphone!) und s dem Abstand zwischen dem Schwerpunkt und der Aufhängung des Pendels entsprechen.

2.3.1 Aufgabe 3.1: Erste Erweiterung des Modells

- Modifizieren Sie Ihr Modell, so dass es dem Modell eines [physikalischen Pendels](#) entspricht und machen Sie eine neue Abschätzung für $g^{(3.1)}$ und $\Delta g^{(3.1)}$.
- Schätzen Sie den Einfluss von $\Delta\Theta$, ΔM , und Δs auf $\Delta g^{(3.1)}$ ab und überlegen Sie, wie Sie diese Unsicherheiten geeignet zu einer Gesamtunsicherheit kombinieren können.

```
[10]: M = 0.789
      ΔM = 0.005
      Θ = 0.2352
      ΔΘp = 0.0026
      ΔΘm = 0.0024
      s = 0.473
      Δs = 0.007

      def sin_model(t, T=1.6, A=0.8, T0=0, M=M, s=s, Θ=Θ, l=1):
          return A*np.cos(np.sqrt(4*np.pi*np.pi*Θ/(T*T*M*s))/l)*t+T0

      xy_data = kafe2.XYContainer(x_data=t, y_data=x)
      xy_data.add_error(axis='x', err_val=Δt)
      xy_data.add_error(axis='y', err_val=Δ)
      sin_fit = kafe2.Fit(data=xy_data, model_function=sin_model)
```

```

sin_fit.add_parameter_constraint(name='l', value=l, uncertainty=Δl)
    ↪ #Alternative dazu, den Fit dreimal durchzuführen
sin_fit.add_parameter_constraint(name='M', value=M, uncertainty=ΔM)
sin_fit.add_parameter_constraint(name='s', value=s, uncertainty=Δs)
sin_fit.add_parameter_constraint(name='θ', value=θ, uncertainty=Δθp)

# θ Wert ohne Unsicherheit
sin_fit.do_fit()
p = kafe2.Plot(sin_fit)
p.plot()
p.show()

T31, A, T0, _, _, _, _ = sin_fit.parameter_values
ΔT31, ΔA, ΔT0, _, _, _, _ = sin_fit.parameter_errors

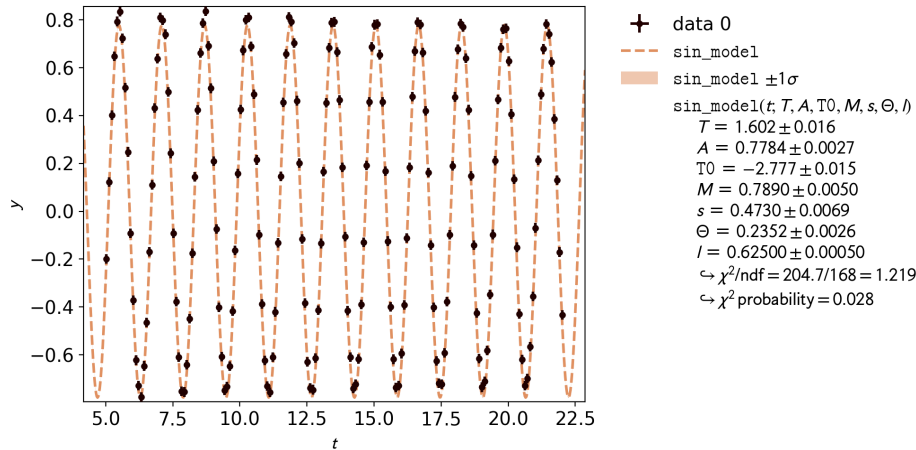
g31 = 4*np.pi**2*θ/(T31**2*M*s)
Δg31 = 4*np.pi**2* \
np.sqrt((1/(T31**2*M*s)*Δθp)**2 \
+(2*θ/(T31**3*M*s)*ΔT31)**2 \
+(θ/(T31**2*M**2*s)*ΔM)**2 \
+(θ/(T31**2*M*s**2)*Δs)**2 )

%m $g^{\{3.1\}} = [!g31:.2f!] \mathrm{\frac{m}{s^2}}$
%m $\Delta g^{\{3.1\}} = [!Δg31:.2f!] \mathrm{\frac{m}{s^2}}$
%m Der Fit hat einen $\frac{\chi^2}{\text{ndf}}$ Wert von [!sin_fit.goodness_of_fit/
    ↪ sin_fit.ndf:.3f!], was recht nah an 1 ist, also einen guten Fit darstellt.

%m (Für diese Aufgabe wurde mangels genauerer Angaben angenommen, dass die
    ↪ angegebenen Unsicherheiten von $t$, $l$, $M$, $s$, $\Theta$ und
    ↪ $\varphi$ sich auf eine für die Anwendung annähernd normalverteilte Größe
    ↪ bezieht. Sollte diese Annahme nicht halten ist die Anwendung von
    ↪ Gaußsche-Fehlerfortpflanzung natürlich nicht richtig.)

```

[10]:



[10]: $g^{(3.1)} = 9.69 \frac{\text{m}}{\text{s}^2}$

[10]: $\Delta g^{(3.1)} = 0.27 \frac{\text{m}}{\text{s}^2}$

[10]: Der Fit hat einen $\frac{\chi^2}{\text{ndf}}$ Wert von 1.219, was recht nah an 1 ist, also einen guten Fit darstellt.

[10]: (Für diese Aufgabe wurde mangels genauerer Angaben angenommen, dass die angegebenen Unsicherheiten von t , l , M , s , Θ und φ sich auf eine für die Anwendung annähernd normalverteilte Größe bezieht. Sollte diese Annahme nicht halten ist die Anwendung von Gaußsche-Fehlerfortpflanzung natürlich nicht richtig.)

2.3.2 Aufgabe 3.2: Zweite Erweiterung der Modells

Das Pendel erfährt in seiner Bewegung zusätzlich eine Dämpfung. Wenn Sie der Messung das Modell einer **linearer gedämpften Schwingung** zugrunde legen verändert sich die Formel zur Bestimmung von g wie folgt:

$$g = \left(\frac{4\pi^2}{T^2} + \frac{1}{\tau^2} \right) \frac{\Theta}{Ms},$$

wobei τ einer Abklingzeit der Schwingungsamplitude in Sekunden entspricht. Wie Sie sehen handelt es sich um eine Korrektur, die die Abschätzung von g zu größeren Werten hin verändert.

Verändern Sie ihr Modell geeignet und beantworten Sie die folgenden Fragen:

- Kann das zugrundeliegende Modell die Daten beschreiben?
- Wie könnten Sie die Hypothese, dass das zugrundeliegende Modell die Daten beschreiben kann, noch besser testen?
- Wie groß ist der Effekt der Korrektur aus der obigen Gleichung auf die Messung von g ?

[13]: $M = 0.789$

$\Delta M = 0.005$

$\Theta = 0.2352$

```

Δθp = 0.0026
Δθm = 0.0024
s = 0.473
Δs = 0.007

def cos_model(t, g=9.81, A=0.8, T0=0, M=M ,s=s,θ=θ, =5):
    return A*np.exp(-t/ )*np.cos(np.sqrt(g*M*s/θ - 1/( * ))*t+T0)

xy_data = kafe2.XYContainer(x_data=t, y_data=x)
xy_data.add_error(axis='x', err_val=Δt)
xy_data.add_error(axis='y', err_val=Δ)
cos_fit = kafe2.Fit(data=xy_data, model_function=cos_model)

cos_fit.add_parameter_constraint(name='M', value=M, uncertainty=ΔM)
cos_fit.add_parameter_constraint(name='s', value=s, uncertainty=Δs)
cos_fit.add_parameter_constraint(name='θ', value=θ, uncertainty=Δθp)

cos_fit.do_fit()
p = kafe2.Plot(cos_fit)
p.plot()
p.show()

g32, A, T0, _, _, _, _ = cos_fit.parameter_values
Δg32, ΔA, ΔT0, _, _, Δ, _ = cos_fit.parameter_errors

%m $g^{\{(3.2)\}} = [!g32:.2f!] \mathrm{\frac{m}{s^2}}$
%m $\Delta g^{\{(3.2)\}} = [!Δg32:.2f!] \mathrm{\frac{m}{s^2}}$
%m Der Fit hat einen $\frac{\chi^2}{\mathrm{ndf}}$ Wert von [!cos_fit.goodness_of_fit/
  ↳cos_fit.ndf:.3f!], was sehr nah an 1 ist, also einen besonders guten Fit
  ↳darstellt.

%m Die Güte des Fits weist darauf hin, dass das mit linearer Dämpfung
  ↳erweiterte Modell besonders gut die Daten beschreiben kann.

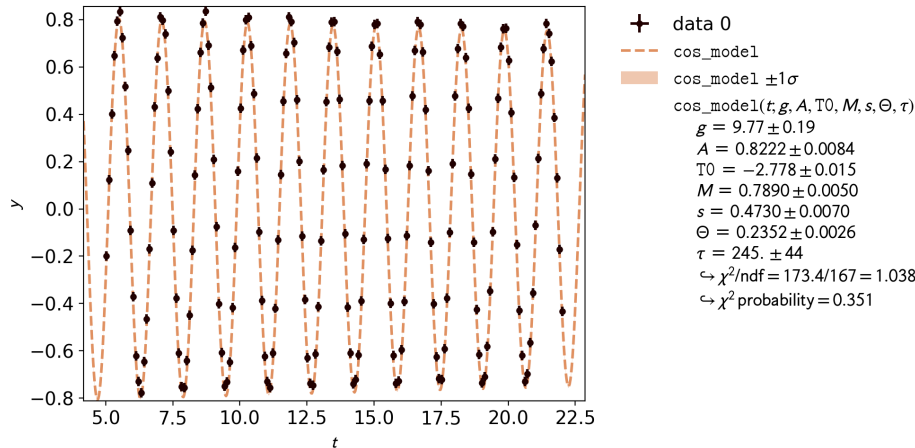
%m Es ließe sich noch genauer testen, wie gut das Modell auf die Daten passt,
  ↳indem man einen Längeren Zeitraum aus den Rohdaten herausnimmt und oder eine
  ↳engere Abtastrate aus den Daten wählt. Ansonsten gibt der Wert von $\chi^2/
  ↳\mathrm{ndf}$ schon einen guten Hinweis auf die Güte des Fits.

%m Das erhaltene Ergebnis für $g^{\{(3.2)\}}$ ist um $[!g32-g31:.2f!]
  ↳\mathrm{\frac{m}{s^2}}$ höher als ohne die Erweiterung.

%m (Für diese Aufgabe wurde mangels genauerer Angaben angenommen, dass die
  ↳angegebenen Unsicherheiten von $$t$, $$M$, $$s$, $\Theta$ und $\varphi$ sich
  ↳auf eine für die Anwendung annähernd normalverteilte Größe bezieht. Sollte
  ↳diese Annahme nicht halten ist die Anwendung von
  ↳Gaußsche-Fehlerfortpflanzung natürlich nicht richtig.)

```

[13] :



[13] : $g^{(3.2)} = 9.77 \frac{\text{m}}{\text{s}^2}$

[13] : $\Delta g^{(3.2)} = 0.19 \frac{\text{m}}{\text{s}^2}$

[13] : Der Fit hat einen $\frac{\chi^2}{ndf}$ Wert von 1.038, was sehr nah an 1 ist, also einen besonders guten Fit darstellt.

[13] : Die Güte des Fits weist darauf hin, dass das mit linearer Dämpfung erweiterte Modell besonders gut die Daten beschreiben kann.

[13] : Es ließe sich noch genauer testen, wie gut das Modell auf die Daten passt, indem man einen Längeren Zeitraum aus den Rohdaten herausnimmt und oder eine engere Abtastrate aus den Daten wählt. Ansonsten geben die werte von χ^2/ndf

[13] : Das erhaltene Ergebnis für $g^{(3.2)}$ ist um $0.08 \frac{\text{m}}{\text{s}^2}$ höher als ohne die Erweiterung.

[13] : (Für diese Aufgabe wurde mangels genauerer Angaben angenommen, dass die angegebenen Unsicherheiten von t , M , s , Θ und φ sich auf eine für die Anwendung annähernd normalverteilte Größe bezieht. Sollte diese Annahme nicht halten ist die Anwendung von Gaußsche-Fehlerfortpflanzung natürlich nicht richtig.)

2.4 Bonusaufgabe: Vom Messen zur Kunst

Ein bewusst formuliertes Modell zur Behandlung der Unsicherheiten, die in eine Parameterschätzung eingehen, ist Bestandteil eines guten statistischen Modells. Die Diskussion über die Berücksichtigung der Unsicherheiten der Modellparameter Θ , M und s , bei der Bestimmung von g und Δg in **Aufgabe 3.1** wirft eine Frage auf, die wir im Rahmen dieses Vorversuchs bisher nur streifen konnten: Wie sind die Unsicherheiten auf die äußeren Parameter der Messung korreliert? Sie können die folgenden Bonusaufgaben bearbeiten, um dieser Frage weiter nachzugehen. Die Bearbeitung ist jedoch nicht verpflichtend.

2.4.1 Bonusaufgabe 1: Korrelierte Unsicherheiten

Jede Variation eines der drei Parameter Θ , M oder s in **Aufgabe 3.1** hat einen nicht-trivialen Einfluss, nicht nur auf g , sondern auch auf die jeweils anderen äußeren Parameter. Durch naive, quadratische Addition von $\Delta\Theta$, ΔM , und Δs unterlegen Sie (vielleicht unterbewusst) die Annahme,

das alle drei Variationen paarweise unabhängig sind. Diese Annahme ist auf jeden Fall falsch! Ein anderes Modell, dass Sie anwenden könnten, wäre zwei oder alle Parameter vollständig zu korrelieren. Was bedeutet diese Annahme für die Variation der Parameter? Denken, Sie dass diese Annahme korrekt ist? Machen Sie einen Vorschlag zur Lösung dieses Problems.

Da Θ aus den Massen M_i und Abständen vom Zentrum s_i berechnet und $\Delta\Theta$ dann aus den jeweiligen Fehlern berechnet wird, ist Θ natürlich mit M_i und s_i korreliert. (M_i und s_i hingegen sollten unkorreliert sein.) Damit sollte sich Θ im Modell vollständig durch M_i und s_i ausdrücken lassen und Θ ließe sich eliminieren. Damit wäre die Korrelation beseitigt.

Ist das nicht gewünscht kann man das Gaußsche Fehlerfortpflanzungsgesetz natürlich auch mit der entsprechenden Kovarianz erweitern. Die allgemeine Formel für die Unsicherheit lautet dann

$$\Delta_y = \sqrt{\sum_{i=1}^m \sum_{k=1}^m \frac{\partial y}{\partial x_i} \frac{\partial y}{\partial x_k} \cdot u(x_i, x_k)} \text{ mit der Kovarianz } u(x_i, x_k) \text{ für die Eingangsgrößen } x_i$$

2.4.2 Bonusaufgabe 2: Experimentelle Verbesserung der Messung

Diskutieren Sie, wie dieser Versuch konzeptionell verbessert werden könnte, um die in Bonusaufgabe 1 diskutierten Probleme von vornherein zu vermeiden.

Entweder sollte das Trägheitsmoment direkt bestimmt werden und nicht aus anderen Größen, die zur Berechnung genutzt werden abgeleitet werden, sodass keine Korrelation entsteht oder es sollten alle gemessenen Werte mit Unsicherheiten direkt in das Modell einfließen ohne vorher Zwischenwerte zu errechnen und Korrelationen zu erzeugen. Es sollten keine Daten vorschnell entsorgt werden.