PR Grundpraktikum I WS 22/23

Hausübung zu Fehlerrechnung und Plots
20.10.2022

Abgabe: 27.10.2022 14:00 Uhr

1) Richtiges Runden [1 Pkt.]

Welche der folgenden Möglichkeiten stellt eine Messung mit Messwert: 1.2736 m und einer Unsicherheit von 0.25034 m korrekt dar?

a) $(1.274 \pm 0.250)\,\mathrm{m}$

b) (1.3 ± 0.3)

c) $1273 \pm 250\,\mathrm{mm}$

d) $1.3(3)\,\mathrm{m}$

e) $1.3(25)\,\mathrm{m}$

f) $(1.3 \pm 0.25034)\,\mathrm{m}$

Erkläre warum die anderen Möglichkeiten falsch sind.

2) Fehlerfortpflanzung [3 Pkte.]

Zwei Größen x und y werden mit statistischen Unsicherheiten bestimmt:

$$x = (17.4 \pm 0.3)\,\mathrm{V}$$
$$y = (9.3 \pm 0.7)\,\mathrm{V}$$

Berechne die folgenden abgeleiteten Größen z und deren Unsicherheiten δz:

a) $z = x - y$

b) $z = 12x + 3y$

c) $z = 5xy$

d) $z = \frac{y^3}{x^2}$

e) $z = x^2 + 3y^2$

f) $z = \arcsin y/x$

g) $z = \sqrt{3xy}$

h) $z = \ln\left(\frac{y}{x}\right)$

i) $z = \frac{x}{y^2} + \frac{y}{x^2}$

j) $z = 2\sqrt{\frac{y}{x}}$

3) Beispiel: Bestimmung der Fallbeschleunigung g [2 Pkte.]

Betrachte ein Geschoss welches horizontal abgefeuert wird. Mithilfe von zwei horizontal aufgestellten Lichtschranken an den Orten $x_1 = (5.000 \pm 0.001)\,\mathrm{m}$ und $x_2 = (17.000 \pm 0.001)\,\mathrm{m}$ wird die Flugdauer $t_x = (77283.5 \pm 0.1)\,\mu\mathrm{s}$ bestimmt. Bestimme den Wert und den damit verbundenen Fehler der Geschwindigkeit des Geschosses (unter der Annahme einer konstanten Geschwindigkeit).

Nun wird das Geschoss vertikal nach oben abgefeuert und zwar mit derselben Geschwindigkeit wie zuvor. Dieselben Lichtschranken sind nun vertikal an zwei Positionen $z_1 = 0\,\mathrm{m}$ (direkt bei der Kanone) und $z_2 = (20.000 \pm 0.001)\,\mathrm{m}$ aufgestellt. Die Zeitmessung ergibt nun eine Flugdauer von $t_z = (129335.3 \pm 0.1)\,\mu\mathrm{s}$. Kombiniere alle aufgenommenen Messungen um einen Ausdruck für die Fallbeschleunigung g und die damit verbundene Unsicherheit zu finden.

Ist dies ein gutes Experiment um die Fallbeschleunigung zu bestimmen? Wäre es besser in eine genauere Zeitmessung oder eine genauere Distanzmessung zu investieren um die Unsicherheit zu reduzieren? Wie könnte man sehr einfach das Experiment ändern, um die Genauigkeit verbessern?

4) Plotten von Daten mit linearem Fit [3 Pkte.]

Ein Student bekommt einen Kupferdraht mit Länge 1 m. Er schließt eine Batterie an den Draht und misst die Potenzialdifferenz zwischen dem negativen Ende und verschiedenen Positionen x_i entlang des Drahtes. Die Unsicherheit jeder Messung des Potenzials beträgt 0.05 V. Die Position wird mittels eines Lineals bestimmt. Der resultierende Messfehler von 1 mm kann hier vernachlässigt werden. Die folgende Tabelle zeigt den aufgenommenen Datensatz.

Datenpunkt	Position x_i (cm)	Potentialdifferenz V_i (Volt)
1	5.0	0.26
2	10.0	0.42
3	15.0	0.59
4	20.0	0.79
5	25.0	0.97
6	30.0	1.11
7	35.0	1.28
8	40.0	1.49
9	45.0	1.63
10	50.0	1.80

Wir wollen nun eine lineare Funktion $y(x)$ finden, welche die Abhängigkeit der Spannung V von der Länge x am besten beschreibt.

a) Verwende die Methode der kleinsten Quadrate um die Koeffizienten der linearen Funktion $y = ax + b$ und deren Unsicherheit zu bestimmen. Hierfür definieren wir eine

Anpassungsgüte $\chi^2 = \sum_i (y_i - y(x_i))^2 / \sigma_i^2$. Die Werte von a und b sollen nun so bestimmt werden, dass χ^2 minimal wird. Um diese Minimierung durchzuführen, berechne die Ableitungen von χ^2 nach den zwei Parametern und setze diese gleich Null. Löse das Gleichungssystem für a und b.

b) Verwende die Daten für die Position und Spannung (also y_i) aus der Tabelle um die Werte von a und b für dieses Beispiel zu bestimmen.

c) Verwende ein Programm (z.B. Python, Mathematica, Matlab) um die Daten zu plotten und zu fitten (Tipps für eine Implementierung in Python und Matlab findet ihr im Anhang). Vergleiche das Ergebnis mit den zuvor manuell berechneten Werten.

d) Bestimme die Unsicherheiten von a und b durch Anwenden der Fehlerfortpflanzung (wie in Übung 2) auf die Ergebnisse in den Punkten a) und b).
Zum Beispiel: da nur die y_i Werte Fehler haben, ist die Varianz von a gegeben durch $\sigma_a^2 = \sum_i \sigma_i^2 (\partial a/\partial y_i)^2$. Analog für b.

e) Vergleiche das Ergebnis in d) mit den Fehlern von a und b, welche vom Programm berechnet wurden.

f) Wenn es horizontale Fehlerbalken gäbe, müsste man diese auch für die Anpassungsgüte berücksichtigen, d.h. $\chi^2 = \sum_i \frac{(y_i - y(x_i))^2}{a^2\sigma_{xi}^2 + \sigma_{yi}^2}$. Will man diese Größe nun Minimieren gibt es keinen analytischen Ausdruck mehr für a und b. Wie könnte man trotzdem ein Fehlerbudget für a und b abschätzen? Seid kreativ!

5) Korrelierte Variablen [1 Pkt.]

Meistens weisen die Ergebnisse eines von einem Computerprogramm berechneten Fits Korrelationen zwischen den bestimmten Parametern β_i auf. In diesem Fall muss die Formel der Fehlerfortpflanzung erweitert werden. Für eine Größe $z(\beta_i)$, die als Funktion von den bestimmten Parametern abhängt, erhält man dann die Unsicherheit

$$(\delta z)^2 = \mathbf{g}^T \mathbf{V} \mathbf{g} \tag{1}$$

mit dem Gradienten $\mathbf{g}$ der Funktion $z(\beta_i)$ und der Kovarianzmatrix $\mathbf{V}$. Die Elemente von $\mathbf{g}$ sind die partiellen Ableitungen von $z(\beta_i)$ nach den einzelnen Parametern $g_i = \partial z/\partial \beta_i$. Wenn alle Nebendiagonaleinträge der Kovarianzmatrix gleich Null sind, dann zeigen die Parameter keine Korrelation (d.h. sie sind unabhängig) und Gleichung (1) vereinfacht sich wieder zu der typischen Formel für die Fehlerfortpflanzung von statistischen Unsicherheiten.

Man stelle sich nun vor, dass nachdem ein linearer Fit mit der Funktion $y = \beta_1 x + \beta_0$ an einen Datensatz angepasst wurde, man den Schnittpunkt der Gerade mit der x-Achse berechnen will. Das Fitprogramm berechnet die besten Werte für β_1 als 3.77 und β_0 als 1.58. Zusätzlich erhält man die Kovarianzmatrix für den Vektor $\mathbf{g} = (\partial z/\partial \beta_0, \partial z/\partial \beta_1)^T$

$$\mathbf{V} = \begin{pmatrix} 0.033 & 0.019 \\ 0.019 & 0.009 \end{pmatrix}.$$

Bestimme die beste Schätzung für den Schnittpunkt und dessen Fehler.

Anmerkung: Eine nützliche Einführung zu Fehlerfortpflanzung mit Korrelationen gibt es von Prof. R. Grimm auf Youtube: `https://www.youtube.com/watch?v=oUBm2ozRpiQ`

Anhang: Daten Plotten und Fitten in Python

Nützliche Bibliotheken

```
import numpy as np  # Importiere Bibliothek f\"ur Numerik
import scipy as sc  # Importiere Bibliothek f\"ur fit
import matplotlib.pyplot as plt # Plot Bibliothek
```

Laden der Daten als numpy array (erlaubt leichtere Manipulationen)

```
Position = np.arange(5., 55., 5) # Erstelle x position
PotDifference = np.array([0.26, 0.42, ...])
errorbars = np.ones(10)*0.05 # Alle Fehler sind gleich
```

Hier sind die Unsicherhieten (errorbars) dieselben für jeden Datenpunkt. Das ist aber nicht immer so. Sie stellen die Gewichtung der einzelnen Datenpunkte für die Minimierung dar (siehe χ^2 oben).

Implementiere die lineare Funktion

```
def linearFunction(x, a, b):
        return a*x + b
```

Die folgende Funktion kann genutzt werden um die Daten zu fitten:

```
sc.optimize.curve_fit()
```

Die Funktion benötigt als Argument die linearFunction und die x-Werte,y-Werte, Anfangswerte (man kann sie einfach 1 setzen, z.B. (1,1)) und Unsicherheiten. Details kann man in der Dokumentation der Funktion finden. Außerdem ist es wichtig das Argument absolute_sigma=True zu geben, da hier die Fehlerwerte als Absolutwerte angegeben sind. Die Funktion gibt die optimierten Werte als ein Numpy Array zurück, sowie die Kovarianzmatrix. Die Unsicherheiten für die Fitparameter sind die Wurzel der Diagonalen der Kovarianzmatrix.

Verwende die folgende Funktion um die Daten und den Fit grafisch darzustellen. Um die Fehlerbalken anzuzeigen wird die plt.errorbars() Funktion benötigt.

```
plt.plot()
```

Anhang: Daten Plotten und Fitten in Matlab

Nützliche Bibliotheken

```
Position =[5.0;10.0;....]
PotDifference =[0.26;0.42;....]
errobars =[0.05;...]
```

Hier sind die Unsicherhieten (errobars) dieselben für jeden Datenpunkt. Das ist aber nicht immer so. Sie stellen die Gewichtung der einzelnen Datenpunkte für die Minimierung dar (siehe χ^2 oben).

Rufe die folgenden Funktionen auf:

```
fitoptions()
```

Hier kann man die Fitmethode (z.B LinearSquareFit), den Ausgangspunkt, Punkte die weggelassen werden sollen und die Gewichtungen (der errobars Vektor) auswählen.

```
fiteval()
```

Hier kann man die Optionen von fitoptions() aufrufen und die Fitfunktion wählen (hier linear).

Hier kann man die Optionen von fitoptions() aufrufen und die Fitfunktion wählen (hier linear).

Die Hauptfunktion, welche die Minimierung durchführt und die Werte der Fitparameter sowie deren Fehler zurückgibt ist:

```
fit()
```

Verwende die folgende Funktion um die Daten und den Fit grafisch darzustellen. Um die Fehlerbalken anzuzeigen wird die errorbars() Funktion benötigt.

```
plot()
```