Exercise 4

Christopher Breitfeld, Linus Bölte, Henry Krämerkämper

November 14, 2023

In a lab experiment the following values are measured:

Ψ / °	Asymmetry	Ψ / °	Asymmetry	Ψ / °	Asymmetry
0	−0.032	30	0.010	60	0.057
90	0.068	120	0.076	150	0.080
180	0.031	210	0.005	240	−0.041
270	−0.090	300	−0.088	330	−0.074

The asymmetry values have a measurement value of ±0.011. The theory says that the asymmetry is described by an ansatz of the form

$$f(\psi) = A_0 \cos\psi + \delta.$$

1 a

Start with the ansatz

$$f(\psi) = a_1 f_1(\psi) + a_2 f_2(\psi) = a_1 \cos(\psi) + a_2 \sin(\psi)$$

and write down the design matrix **A**.

$\rightarrow$ Designmatrix $A = \begin{bmatrix} f_1(x_1) & \cdots & f_n(x_1) \\ \vdots & & \vdots \\ f_1(x_n) & \cdots & f_n(x_n) \end{bmatrix} \Rightarrow$ hier: $A = \begin{bmatrix} \cos(\Psi_1) & \sin(\Psi_1) \\ \vdots & \vdots \\ \cos(\Psi_{12}) & \sin(\Psi_{12}) \end{bmatrix}$ mit den Messwerten $\Psi_1, \ldots, \Psi_{12}$ aus der Tabelle oben.

```
import numpy as np
# muss damit numpy nicht weint in rad umgewandelt werden
psi = np.radians(np.arange(0, 331, 30))
```

```
measurement = np.array([-0.032, 0.010, 0.057, 0.068, 0.076, 0.080, 0.031, 0.005, -
0.041, -0.090, -0.088, -0.074])
measurement_error = 0.011

A = np.column_stack([np.cos(psi), np.sin(psi)])
print(f"A = {A}")
```

```
A = [[ 1.00000000e+00  0.00000000e+00]
 [ 8.66025404e-01  5.00000000e-01]
 [ 5.00000000e-01  8.66025404e-01]
 [ 6.12323400e-17  1.00000000e+00]
 [-5.00000000e-01  8.66025404e-01]
 [-8.66025404e-01  5.00000000e-01]
 [-1.00000000e+00  1.22464680e-16]
 [-8.66025404e-01 -5.00000000e-01]
 [-5.00000000e-01 -8.66025404e-01]
 [-1.83697020e-16 -1.00000000e+00]
 [ 5.00000000e-01 -8.66025404e-01]
 [ 8.66025404e-01 -5.00000000e-01]]
```

2 b

Calculate the solution vector $\hat{a}$ for the parameters using the method of the least squares.
The solution vector $\hat{a}$ can be calculated as $\hat{a} = (\mathbf{A^T} \cdot \mathbf{A})^{\mathbf{-1}} \mathbf{A^T y}$:

```
a = np.linalg.inv(A.T @ A) @ A.T @ measurement
print(f"a = {a}")
```

```
a = [-0.0375063   0.07739978]
```

3 c

Calculate the covariance matrix $\mathbf{V}[\mathbf{\hat{a}}]$ as well as the errors of a_1 and a_2 and the correlation coefficient.
The covariance matrix of $\hat{a}$ is given by $\mathbf{V}[\mathbf{\hat{a}}] = \sigma^{\mathbf{2}}(\mathbf{A^T} \cdot \mathbf{A})^{\mathbf{-1}}$, where σ denotes the measurement error.

```
V = measurement_error**2 * np.linalg.inv (A.T @ A)
print(f"V = {V}")
```

```
V = [[ 2.01666667e-05 -1.31944981e-21]
 [-1.31944981e-21  2.01666667e-05]]
```

The error of a_1 and a_2 can be calculated via the diagonal elements of the covariance matrix:

```
a_error = np.sqrt(np.diag(V))
print(f"a_error ={a_error}")
```

```
a_error =[0.00449073 0.00449073]
```

4 d

Calculate A_0 and δ, their error and the correlation of a_1 and a_2.

$$\rightarrow f(\varphi) = A_0 \cos(\varphi + \delta) = a_1 \cos(\varphi) + a_2 \sin(\varphi) \qquad \rightarrow \cos(x+y) = \cos(x)\cos(y) - \sin(x)\sin(y)$$

$$\Leftrightarrow A_0 \cos(\delta)\cos(\varphi) - A_0 \sin(\delta)\sin(\varphi) = a_1 \cos(\varphi) + a_2 \sin(\varphi)$$

$$\Rightarrow A_0 \cos(\delta) = a_1 \ , \quad -A_0 \sin(\delta) = a_2$$

$$A_0 = \frac{a_1}{\cos(\delta)} \Rightarrow \frac{-a_1 \sin(\delta)}{\cos(\delta)} = a_2 \Leftrightarrow \frac{\sin(\delta)}{\cos(\delta)} = \frac{-a_2}{a_1} \Leftrightarrow \delta = \arctan\left(-\frac{a_2}{a_1}\right)$$

$$\Rightarrow A_0 = \frac{a_1}{\cos\left(\arctan\left(-\frac{a_2}{a_1}\right)\right)} \quad \rightarrow \text{verwende } \cos\left(\tan^{-1}\left(\frac{-a}{b}\right)\right) = \frac{1}{\sqrt{\frac{a^2}{b^2} - 1}}$$

$$\Rightarrow A_0 = a_1 \sqrt{\frac{a_2^2}{a_1^2} - 1}$$

```
A_0 = a[0] * np.sqrt((a[1]**2 / a[0]**2) -1)
delta = np.arctan(-a[1]/a[0])
print(f"A_0 = {A_0}")
print(f"delta = {delta}")
```

The error of a solution vector $\vec{y} = f(\vec{x})$ is can be computed by using $\mathbf{V}[\tilde{\mathbf{y}}] = \mathbf{J} \cdot \mathbf{V}[\tilde{\mathbf{x}}] \cdot \mathbf{J}^{\mathbf{T}}$. Calculate the Jacobian matrix of $f(\vec{x}) = \begin{bmatrix} A_0 \\ \delta \end{bmatrix}$:

$$f(\vec{x}) = \begin{bmatrix} A_0 \\ \delta \end{bmatrix} = \begin{bmatrix} a_1 \sqrt{\frac{a_2^2}{a_1^2} - 1} \\ \arctan\left(\frac{-a_2}{a_1}\right) \end{bmatrix} = \begin{bmatrix} \sqrt{a_2^2 - a_1^2} \\ \arctan\left(\frac{-a_2}{a_1}\right) \end{bmatrix} \quad \rightarrow \text{Jacobi-Matrix} \quad J = \begin{bmatrix} \frac{\partial f_1}{\partial a_1} & \frac{\partial f_1}{\partial a_2} \\ \frac{\partial f_2}{\partial a_1} & \frac{\partial f_2}{\partial a_2} \end{bmatrix}$$

$$\frac{\partial f_1}{\partial a_1} == \frac{\partial}{\partial a_1} \sqrt{a_2^2 - a_1^2} = -\frac{a_1}{\sqrt{a_2^2 - a_1^2}} , \quad \frac{\partial f_1}{\partial a_2} = \frac{\partial}{\partial a_2} \sqrt{a_2^2 - a_1^2} = \frac{a_2}{\sqrt{a_2^2 - a_1^2}}$$

$$\frac{\partial f_2}{\partial a_1} = \frac{\partial}{\partial a_1} \arctan\left(-\frac{a_2}{a_1}\right) = \frac{a_2}{a_1^2 + a_2^2} , \quad \frac{\partial f_2}{\partial a_2} = \frac{\partial}{\partial a_2} \arctan\left(-\frac{a_2}{a_1}\right) = \frac{-a_1}{a_1^2 + a_2^2}$$

```
J = np.array([[(-a[0]/(np.sqrt(a[1]**2 - a[0]**2))),
               (a[1]/(np.sqrt(a[1]**2 - a[0]**2)))],
              [(a[1]/(a[0]**2 + a[1]**2)),
               (-a[0]/(a[0]**2 + a[1]**2))]])

V_2 = J @ V @ J.T
errors = np.sqrt(np.diag(V_2))
# correlation_coefficient = V_2[1,0] / V_2[0,0]
correlation_coefficient = V_2[1,0] / errors[0]**2

print(f"Fehler von A_0: {errors[0]}")
print(f"Fehler von delta: {errors[1]}")
print(f"Korrelationskoeffizient:{correlation_coefficient}")
```

```
Fehler von A_0: 0.005704736193232627
Fehler von delta: 0.05221269485613977
Korrelationskoeffizient:7.183441097416744
```