# Spezifische_Waermekapazitaet

June 17, 2024

## 1 Fakultät für Physik

### 1.1 Physikalisches Praktikum P2 für Studierende der Physik

Versuch P2-33, 34, 35 (Stand: April 2024)

Raum F1-08

## 2 Spezifische Wärmekapazität

Name: Vrkic Vorname: Tin E-Mail: uyvpq@student.kit.edu

Name: Nock Vorname: Mika E-Mail: uttzi@student.kit.edu

Gruppennummer: Mo-32

Betreuer: Asya Can

Versuch durchgeführt am: 17.06.24

**Beanstandungen zu Protokoll Version _____:**

Testiert am: _____ Testat: _____

## 3 Durchführung

**Die Anleitung zu diesem Versuch finden Sie hier.**

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import kafe2
from uncertainties import ufloat, unumpy as unp
import pathlib

def weighted_mean(array):
    return
```

[9]:
```
# erstellen einer Funktion für kafe2 Fits
def fit_funktion(xy_data, model_function, xy_error, xy_label, title,␣
 ↪constraint=[], add_error=True):
    xy_data = kafe2.XYContainer(xy_data[0], xy_data[1])
    xy_data.label = title
    fit = kafe2.XYFit(xy_data = xy_data, model_function = model_function)
    if add_error:
        fit.add_error(axis = 'x', err_val = xy_error[0])
        fit.add_error(axis = 'y', err_val = xy_error[1])
    for i in range(len(constraint)):
        fit.add_parameter_constraint(name = constraint[i][0], value =␣
 ↪constraint[i][1], uncertainty = constraint[i][2])
    fit.do_fit()
    plot = kafe2.Plot(fit)
```

```
    plot.x_label, plot.y_label = xy_label[0], xy_label[1]

    return fit.parameter_values, fit.parameter_errors, plot
```

## 3.1 Aufgabe 1: Spezifische Wärmekapazität aus Mischtemperatur

**Hinweise zu Aufgabe 1 finden in der Datei Hinweise-Versuchsdurchfuehrung.md.**

- Bestimmen Sie so genau wie mit den verfügbaren Mitteln möglich, die spezifische Wärmekapazität $c_X$ von Aluminium und noch eines weiteren Metalls, durch das Herstellen von Mischtemperaturen in einem Glas-Kalorimeter.
- Bearbeiten Sie hierzu die folgenden Aufgaben.

### 3.1.1 Aufgabe 1.1: Planung des Messvorhabens

- Beschreiben Sie das von Ihnen geplante Messvorhaben und den dazu verwendeten Aufgbau in eigenen Worten.
- Führen Sie ggf. geeignete Vergleichs- und Kalibrationsmessreihen durch, um sich von der Eignung der Methode zu überzeugen oder Korrekturen oder Unsicherheiten abzuschätzen.

---

In dieser Aufgabe soll ein Aufbau bestimmt werden, mit dem die spezifische Wärmekapazität $c_M$ von verschiedenen Metallen gemessen werden kann. Hierbei sollen verschiedene Messfehler verringert werden.

Beim vermessenen Stoff sollte es sich um eine Form handeln, die möglichst gut Wärme mit ihrer Umgebung, also dem Wasser im Kalorimeter, austauschen kann. Das führt dazu, das sich möglichst schnell ein Gleichgewicht einstellt und wenig Wärme an die Umgebung verloren geht. Es wird also ein Granulat einem festen Körper vorgezogen, obwohl sich hier mögliche Probleme beim Erhitzen und bei der genauen Bestimmung der Masse ergeben.

Die Flüssigkeit im Kalorimeter hätte an der Öffnung einer Vergleichsweise große, unisolierte Kontaktoberfläche. Um die Verluste der Wärme möglichst gering zu halten sollte sich die Flüssigkeit bei Raumtemperatur und das Metall bei höheren Temperaturen befinden.

Bei der Flüssigkeit sollte es sich um Wasser handeln, da es aufgrund seiner hohen Wärmekapazität temperaturstabilder gegenüber der Umgebung ist als andere Stoffe.

### 3.1.2 hier fehlen noch Ausarbeitungen

---

### 3.1.3 Aufgabe 1.2: Durchführung des Messprogramms und Auswertung

- Führen Sie eine Messreihe geeigneter Länge für Aluminium und ein weiteres Metall Ihrer Wahl durch.
- Bestimmen Sie für Ihre Auswertung Unsicherheiten auf alle Parameter, die die Messung Ihrer Meinung nach beeinflussen können, pflanzen Sie diese auf das Ergebnis fort und vergleichen Sie mit der Erwartung.

---

[10]:
```
# Messungen Aluminium
T_h20_1 = unp.uarray([],[])
T_al = unp.uarray([],[])
T_mix_1 = ([],[])
m_h20_1 = unp.uarray([],[])
m_al = unp.uarray([],[])

# Messungen
T_h20_2 = unp.uarray([],[])
T_ = unp.uarray([],[])
T_mix_2 = unp.uarray([],[])
m_h20_2 = unp.uarray([],[])
m_ = unp.uarray([],[])

# calculating cm
def get_cm(T_h20,T_mix,T_m,m_h20,m_m):
    cm =  4.1815 * (m_h20(T_h20-T_mix)) * (m_m(T_m-T_mix))
    return weighted_mean()

# printing the results
print(f'Die spezifische Wärmekapazität von Aluminium beträgt {4.1815 *␣
 ↪(m_h20_1*(T_h20_1-T_mix_1)) * (m_al*(T_al-T_mix))}')
print(f'Die spezifische Wärmekapazität von ... beträgt␣
 ↪{get_cm(T_h20_2,T_mix_2,T_,m_h20_2,m_)}')
```

```
Die spezifische Wärmekapazität von Aluminium beträgt []
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/home/mika/Dokumente/P2/Spezifische_Waermekapazitaet/
 ↪Spezifische_Waermekapazitaet.ipynb Cell 11 line 2

    <a href='vscode-notebook-cell:/home/mika/Dokumente/P2/
 ↪Spezifische_Waermekapazitaet/Spezifische_Waermekapazitaet.
 ↪ipynb#X26sZmlsZQ%3D%3D?line=19'>20</a> # printing the results
    <a href='vscode-notebook-cell:/home/mika/Dokumente/P2/
 ↪Spezifische_Waermekapazitaet/Spezifische_Waermekapazitaet.
 ↪ipynb#X26sZmlsZQ%3D%3D?line=20'>21</a> print(f'Die spezifische Wärmekapazität␣
 ↪von Aluminium beträgt {4.1815 * (m_h20_1*(T_h20_1-T_mix_1)) *␣
 ↪(m_al*(T_al-T_mix))}')
---> <a href='vscode-notebook-cell:/home/mika/Dokumente/P2/
 ↪Spezifische_Waermekapazitaet/Spezifische_Waermekapazitaet.
 ↪ipynb#X26sZmlsZQ%3D%3D?line=21'>22</a> print(f'Die spezifische Wärmekapazität␣
 ↪von … beträgt {get_cm(T_h20_2,T_mix_2,T_,m_h20_2,m_)}')


/home/mika/Dokumente/P2/Spezifische_Waermekapazitaet/
 ↪Spezifische_Waermekapazitaet.ipynb Cell 11 line 1

    <a href='vscode-notebook-cell:/home/mika/Dokumente/P2/
 ↪Spezifische_Waermekapazitaet/Spezifische_Waermekapazitaet.
 ↪ipynb#X26sZmlsZQ%3D%3D?line=15'>16</a> def get_cm(T_h20,T_mix,T_m,m_h20,m_m):
```

```
---> <a href='vscode-notebook-cell:/home/mika/Dokumente/P2/
 ↪Spezifische_Waermekapazitaet/Spezifische_Waermekapazitaet.
 ↪ipynb#X26sZmlsZQ%3D%3D?line=16'>17</a>      cm =  4.1815 * (m_h20(T_h20-T_mix)
 ↪* (m_m(T_m-T_mix))
     <a href='vscode-notebook-cell:/home/mika/Dokumente/P2/
 ↪Spezifische_Waermekapazitaet/Spezifische_Waermekapazitaet.
 ↪ipynb#X26sZmlsZQ%3D%3D?line=17'>18</a>      return weighted_mean()

TypeError: 'numpy.ndarray' object is not callable
```

In diesem Versuch wird die Messung nach den Methoden, die in 1.1 erläutert wurden, durchgeführt.

Die aufgenommen Daten werden mithilfe der Formel $c_M = c_{H_2O}\frac{m_{H_2O}(T_{H_2O}-T_{mix})}{m_M(T_{H_2O}-T_{mix})}$ dann zur spezifischen Wärmekapazität umgerechnet und der gewichtete Mittelwert aus den Einzelmessungen gebildet. Hierbei werden die Fehler, die auf jedem einzelnen Wert liegen mit Hilfe des uncertainties-Paketes verarbeitet und in die Berechnung mit eingebracht.

Es ergeben sich somit Werte von:

$$c_M(Aluminium) = \pm\frac{kJ}{kgK}, c_M() = \pm\frac{kJ}{kgK}$$

im Vergleich mit den Literaturwerten:

$$c_M(Aluminium) = \pm\frac{kJ}{kgK}, c_M() = \pm\frac{kJ}{kgK}$$

---

## 3.2 Aufgabe 2: Spezifische Wärmekapazität von Aluminium als Funktion der Temperatur

**Hinweise zu Aufgabe 1 finden in der Datei Hinweise-Versuchsdurchfuehrung.md.**

- Messen Sie $c_{\mathrm{Al}}(T)$ in Abhängigkeit von der Temperatur in einem Bereich zwischen $T = 100 - 300\,\mathrm{K}$.
- Bearbeiten Sie hierzu die folgenden Aufgaben.

### 3.2.1 Aufgabe 2.1: Datennahme

- Beschreiben Sie das VOrgehen in eigenen Worten.
- Bereiten Sie die Datennahme vor, nehmen Sie die Messpunkte auf.

---

**Lösung:**

*Fügen Sie Ihre Lösung zu dieser Aufgabe hier ein. Löschen Sie hierzu diesen kursiv gestellten Text aus dem Dokument. Um Code-Fragmente und Skripte in Python, sowie ggf. bildliche Darstellungen direkt ins Jupyter notebook einzubinden fügen Sie dem notebook eine Code-Zelle zu.*

---

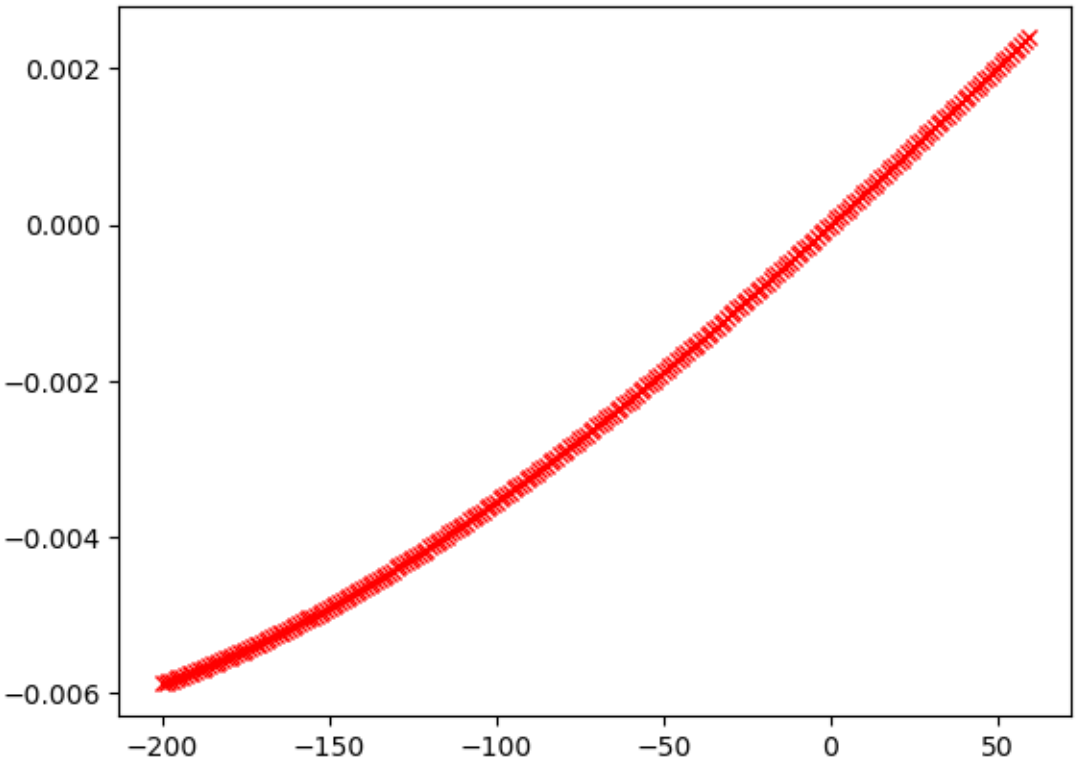
### 3.2.2 Aufgabe 2.2: Kalibration des Thermoelements

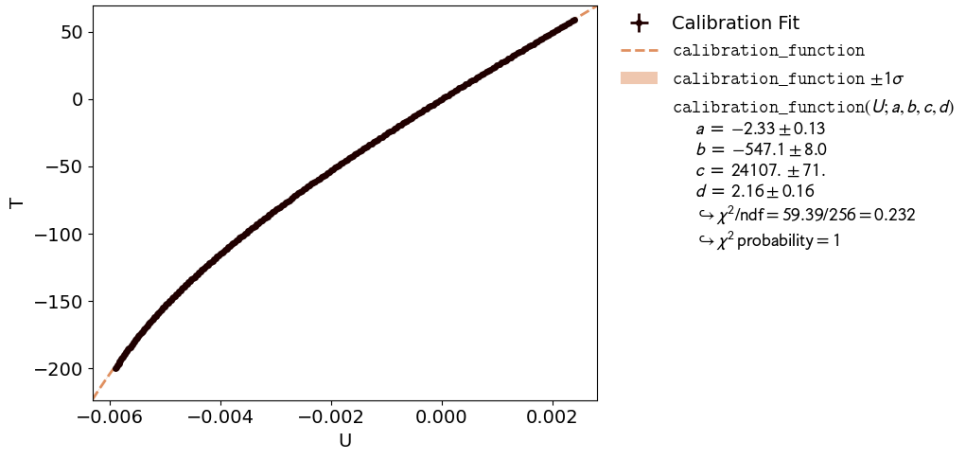
- Kalibrieren Sie das NiCr-Ni-Thermoelement von ursprünglichen Angabe von Volt auf eine Temperaturmessung in Kelvin.
- Sie können diese Kalibration mit Hilfe der Datei calibration.csv vornehmen.

---

[28]:
```
calibration = pd.read_csv(f"{str(pathlib.Path().resolve())}/params/calibration.
 ↪csv", delimiter=",", decimal=".")
calib_T, calib_U = np.array(calibration["T"], dtype=np.float32), np.
 ↪array(calibration["U"], dtype=np.float32) * 10**(-3)
plt.plot(calib_T, calib_U, "rx")
def calibration_function(U, a=-1, b=1, c=1, d=1):
    return a * np.exp(b*U) + c * U + d
calib_data = np.array([calib_U, calib_T])
calib_error = np.array([.016*10**(-3), .05])
calib_label = ["U" , "T"]
calib_title = "Calibration Fit"
calib_fit = fit_funktion(calib_data, calibration_function, calib_error,␣
 ↪calib_label, calib_title)
calib_fit[2].plot()
```

[28]:
```
[{'main': {'plots': [{'type': 'data',
     'fit_index': 0,
     'adapter': <kafe2.fit.xy.plot.XYPlotAdapter at 0x2087971da00>,
     'artist': <ErrorbarContainer object of 3 artists>},
    {'type': 'model',
     'fit_index': 0,
     'adapter': <kafe2.fit.xy.plot.XYPlotAdapter at 0x2087971da00>,
     'artist': None},
    {'type': 'model_line',
     'fit_index': 0,
     'adapter': <kafe2.fit.xy.plot.XYPlotAdapter at 0x2087971da00>,
     'artist': [<matplotlib.lines.Line2D at 0x2087c01a100>]},
    {'type': 'model_error_band',
     'fit_index': 0,
     'adapter': <kafe2.fit.xy.plot.XYPlotAdapter at 0x2087971da00>,
     'artist': <matplotlib.collections.PolyCollection at 0x2087c01a310>}],
   'x_range': (-0.0063052999204956, 0.0028093000757507987)}}]
```




### 3.2.3 Aufgabe 2.3: Korrektur des Wärmegangs

Trotz Wärmeisolation nimmt der für die Messung verwendete Aluminium-Hohlzylinder im Verlauf der Messung zusätzlich zur elektrischen Heizleistung Wärme aus der Umgebung auf (**Wärmegang**).

- Sie könnten den Wärmegang durch eine zweite, gleichartige Messung, ohne elektrische Heizung abschätzen (**Nullmessung**) und korrigieren. Eine solche Messung dauert allerdings 24 Stun-

den, sie lässt sich also nicht an einem Versuchstag durchführen.
- Sie können stattdessen die Daten aus der Datei waermegang.csv für diese Korrektur verwenden.

---

**Lösung:**

*Fügen Sie Ihre Lösung zu dieser Aufgabe hier ein. Löschen Sie hierzu diesen kursiv gestellten Text aus dem Dokument. Um Code-Fragmente und Skripte in Python, sowie ggf. bildliche Darstellungen direkt ins Jupyter notebook einzubinden fügen Sie dem notebook eine Code-Zelle zu.*

---

### 3.2.4 Aufgabe 2.4: Bestimmung von $c_{\mathrm{Al}}(T)$

- Bestimmen Sie den Verlauf der spezifischen Wärmekapazität von Aluminium als Funktion von $T$.
- Bestimmen Sie aus diesem Verlauf den Wert für $T = 20^{\circ}\mathrm{C}$ und vergleichen Sie mit Ihrer Erwartung.
- Bestimmen Sie aus diesem Verlauf den Wert für $T_{\mathrm{Mix}}$ und vergleichen Sie mit dem Ergebnis aus **Aufgabe 1.2**.

---

**Lösung:**

*Fügen Sie Ihre Lösung zu dieser Aufgabe hier ein. Löschen Sie hierzu diesen kursiv gestellten Text aus dem Dokument. Um Code-Fragmente und Skripte in Python, sowie ggf. bildliche Darstellungen direkt ins Jupyter notebook einzubinden fügen Sie dem notebook eine Code-Zelle zu.*

---