

# Computergrundlagen 2025

## Blatt 9: Python

- Abgabetermin für die Lösungen: **21.12.2025, 20 Uhr**
- Bei Fragen wendet euch bitte an eure/n Tutor/in:
  - Mo 11:30: Stephan Haag: `st170833@stud.uni-stuttgart.de`
  - Di 09:45: Julian Hoßbach: `julian.hossbach@icp.uni-stuttgart.de`
  - Mi 14:00: Julian Peters: `julian.peters@icp.uni-stuttgart.de`
  - Do 09:45: Rebecca Stephan: `rebecca.stephan@icp.uni-stuttgart.de`
  - Fr 09:45: Jonas Höpker: `st182335@stud.uni-stuttgart.de`
- Die Übungsaufgaben sollen in der Regel in **Zweiergruppen** bearbeitet werden. Nur in **begründeten Ausnahmefällen** sind Dreiergruppen möglich.
- Die Abgabe der Übungsblätter erfolgt über Ilias.
- Mit Abgabe der Lösungen erklärt Ihr, dass Ihr die Lösung euren Mitstudierenden im Rahmen der Übungsbesprechung vorstellen könnt. Um dies zu überprüfen, muss mindestens zweimal von jedem Teilnehmenden vorgetragen werden. Wenn Ihr das nicht könnt, werden euch die Punkte für die entsprechenden Aufgaben wieder abgezogen.
- **Befehle, die nicht in der Vorlesung besprochen wurden, müssen gegebenenfalls recherchiert werden.**
- **Alle erstellen Skripte (.py and .ipynb) sowie ein mit markown oder Latex erstellter Report (.pdf) sind Teil der Abgabe**

Tipp: Für Abgaben in Python kann der Editor <https://jupyter.org/> hilfreich sein. Hier könnt ihr euren Code *interactive* ausführen sowie mit Markdown Zellen ausführlich kommentieren. Die \*.ipynb\* Dateien können ebenfalls als Lösungen abgegeben werden und sind leicht zu validieren.

## Fitten von Funktionen (3 Punkte)

Die bereitgestellte Datei `data.txt` enthält eine Liste von  $x$ - und  $y$ -Werten einer Funktion, deren analytische Form jedoch nicht bekannt ist. Zusätzlich sind die Funktionswerte mit statistischem Rauschen belegt.

- Lade die Daten in ein NumPy-Array und plote sie.
- Überlege dir eine Fitfunktion und fitte die Daten mithilfe von SciPy. Gib die gefitteten Parameter an.
- Erstelle einen Plot, der die Daten sowie die gefittete Funktion enthält. Achte auf die Achsenbeschriftung und erstelle eine Legende.

## SymPy (4 Punkte)

Betrachte die Funktion einer gedämpften Schwingung:

$$f(t) = A \cdot e^{-\lambda t} \cdot \cos(\omega t)$$

Schreibe ein Python-Skript, das diese Funktion symbolisch analysiert und anschließend visualisiert. Bearbeite dazu die folgenden Teilaufgaben:

1. **Symbolische Definition** Definiere die Variablen  $t, A, \lambda$  und  $\omega$  als SymPy-Symbole. Lege dabei fest, dass  $\lambda$  und  $\omega$  positive, reelle Zahlen sind und  $t$  reell ist (`real=True`, `positive=True`).
2. **Analysis**
  - Berechne symbolisch die erste Ableitung  $f'(t)$ .
  - Berechne symbolisch das unbestimmte Integral  $\int f(t) dt$ .
  - Berechne den Grenzwert der Funktion für  $t \rightarrow \infty$ .
3. **Numerische Konvertierung** Erzeuge mithilfe von `sp.lambdify` aus dem symbolischen Ausdruck für  $f(t)$  und der berechneten Ableitung  $f'(t)$  zwei Python-Funktionen, die NumPy-Arrays verarbeiten können.
4. **Visualisierung** Erstelle einen Plot (mit `matplotlib.pyplot`), der die Funktion  $f(t)$  und ihre Ableitung  $f'(t)$  im Bereich  $t \in [0, 10]$  darstellt.
  - Verwende folgende Parameterwerte:  $A = 2, \lambda = 0.5, \omega = 3$ .
  - Füge eine Legende und Achsenbeschriftungen hinzu.

### Primzahlen (3 Punkte)

Schreibe eine Funktion, um Primzahlen mithilfe des **Sieb des Eratosthenes** zu bestimmen. Als Argument soll die Funktion eine Zahl  $N$  erhalten und alle Primzahlen kleiner oder gleich  $N$  zurückgeben.

Tipp: Für Infos zum Sieb des Eratosthenes kannst du dir den Wikipedia Artikel durchlesen: [https://de.wikipedia.org/wiki/Sieb\\_des\\_Eratosthenes](https://de.wikipedia.org/wiki/Sieb_des_Eratosthenes)