

# Simulation technischer Systeme mit Python

Prüfung im Sommersemester 2023

Bitte tragen Sie Nachnamen, Vornamen und Matrikelnummer, Hörsaal und Platznummer deutlich lesbar in die vorgesehenen Felder ein.

Name	Vorname	Matrikelnummer	Hörsaal	Platz-Nr.
------	---------	----------------	---------	-----------

Punkte	Aufgabe 1	Aufgabe 2	Aufgabe 3	Erreichte Pkte.	Note:	
					bestanden?	
					ja	nein

Ich versichere hiermit, dass ich mich geistig und körperlich in der Lage befinde, die Prüfung abzulegen (d. h. prüffähig bin).

☐ Ja

☐ Nein

\_\_\_\_\_  
Unterschrift Studierende/r

## Abschlussprüfung – SoSe 23

### Hinweise zur Prüfung

*Die Prüfung ist bestanden, wenn mindestens 50% der maximal erreichbaren Punkte erzielt werden. Zur Bearbeitung der Prüfung stehen 100 Minuten Zeit zur Verfügung.*

*In jeder Abbildung müssen alle Achsenbeschriftungen sinnvoll gewählt werden.*

Hinweise zur Bearbeitung:

- Es sind alle nicht kommunikativen Hilfsmittel (z. B. Taschenrechner, Skript, Übungsunterlagen, Vorlesungsmitschrift) zugelassen. Chatbots wie ChatGPT sind nicht zugelassen. Die Arbeiten müssen selbstständig bearbeitet werden! Es dürfen keine großen Code-Blöcke per copy/paste eingefügt werden.
- Im Stud.IP-Ordner „Prüfungsaufgaben“ finden Sie außer dieser Aufgabenstellung zusätzliche Dateien, die für die Lösung der Aufgaben benötigt werden.
- Sorgen Sie am Ende der Prüfung eigenverantwortlich für das ordnungsgemäße Hochladen Ihrer Lösungen über den Stud.IP-Ordner „Abgabe Prüfungen“ im Prüfungskurs! Bitte benennen Sie Ihre Dateien nach folgendem Schema: „FAMILIENNAME\_VORNAME\_MATRIKELNUMMER\_AufgabeX.py“, wobei X die Aufgabennummer bedeutet. Bitte laden sie jede Aufgabe einzeln und nicht als .zip hoch.
- Bitte beachten Sie, dass die **Bearbeitungszeit 100 Minuten** beträgt! Bitte sorgen Sie eigenverantwortlich dafür, dass Sie Ihre Lösungen rechtzeitig vor Ende der Abgabefrist **(16:10 Uhr)** über Stud.IP hochladen!
- Lesen Sie vor der Bearbeitung der Aufgaben die Aufgabenstellung aufmerksam durch!
- Für die ordnungsgemäße Abgabe Ihrer Arbeit sind Sie selbst verantwortlich.

Diese Aufgabenstellung umfasst 5 Seiten. Die maximal erreichbare Punktezahl beträgt 60 Punkte.

## Aufgabe 1: Plotten und Fitten (20 Punkte)

Die Opioidkrise, eine ernsthafte gesundheitliche Herausforderung, hat in den USA zu einer alarmierenden Zunahme von Drogentodesfällen geführt. In dieser Aufgabe werden Sie Daten zu Drogentodesfällen in den USA analysieren und ein Vorhersagemodell erstellen, um Einblicke in die Entwicklung der Todesfälle im Zusammenhang mit Fentanyl-Überdosierungen zu gewinnen.

- Importieren Sie die Daten zu den Drogentodesfällen in den USA von 1990 bis 2020. Diese Daten sind in der Datei "**od\_deaths\_usa.csv**" vorhanden.
- Erstellen Sie einen Plot, auf dem die Gesamtzahl der Todesfälle und die Anzahl der mit Fentanyl zusammenhängenden Todesfälle durch Drogenüberdosis dargestellt werden. Verwenden Sie für die Gesamtzahl der Todesfälle eine blaue Linie und für die fentanylbezogenen Todesfälle eine rote Linie. Beschriften Sie die Achsen, fügen Sie eine Legende hinzu und aktivieren Sie das Gitter.
- Passen Sie die folgende Funktion an die Daten an, indem Sie die Methode **curve\_fit** aus der Bibliothek **scipy.optimize** verwenden. Verwenden Sie als Startwerte  $a = 1$  und  $b = 2000$ :  
$$f(x, a, b) = \exp(a \cdot (x - b)).$$
- Fügen Sie das Vorhersagemodell in derselben Farbe wie die entsprechende Kurve als gestrichelte Linie in den Plot ein. Verwenden Sie den Zeitraum von 1975 bis 2025 für die Vorhersage. Geben Sie die Parameterwerte  $a$  und  $b$  des Modells in der Legende an.
- Basierend auf Ihrem Vorhersagemodell: Wie viele Todesfälle im Zusammenhang mit Fentanyl-Überdosierungen werden in den USA im Jahr 2023 und im Jahr 2030 erwartet? Geben Sie Ihre Antwort als formatierten String aus.
- In welchem Jahr werden voraussichtlich mehr Todesfälle aufgrund von Fentanyl-Überdosierungen als aufgrund von Gesamtüberdosierungen auftreten? Berücksichtigen Sie dabei den Zeitraum von 2010 bis 2025.

## Aufgabe 2: Modellierung eines Räuber-Beute-Systems (20 Punkte)

Das Räuber-Beute-Modell beschreibt die Interaktion zwischen zwei Populationen, wobei eine als Räuber und die andere als Beute agiert. In dieser Aufgabe betrachten wir das Modell mit Füchsen als Räubern und Hasen als Beute. Das Modell kann durch folgende Differentialgleichungen beschrieben werden:

$$\frac{dH}{dt} = aH - bFH \qquad \frac{dF}{dt} = -cF + dHF$$

Dabei ist  $H$  die Populationsgröße der Hasen,  $F$  die Populationsgröße der Füchse,  $a$  die Wachstumsrate der Hasen,  $b$  die Rate, mit der Füchse Hasen fressen,  $c$  die natürliche Sterberate der Füchse und  $d$  die Rate, mit der Füchse sich ernähren und vermehren.

- Lösen Sie das Differentialgleichungssystem mithilfe von **solve\_ivp** aus der **scipy.integrate** Bibliothek. Verwenden Sie die Anfangsbedingungen  $H_0 = 100$  und  $F_0 = 20$ . Berechnen Sie die Populationsentwicklung für einen Zeitraum von 0 bis 100 Jahren. Verwenden Sie die gegebenen Parameter  $a = 0.03$ ,  $b = 0.001$ ,  $c = 0.1$  und  $d = 0.002$ . Nutzen Sie den **Radau-Solver** zur Lösung.
- Plotten Sie das Ergebnis in einem Subplot mit zwei übereinander liegenden Plots. Im oberen Plot soll die Entwicklung der Hasenpopulation  $H$  als blaue Linie und die Entwicklung der Fuchspopulation  $F$  als rote Linie dargestellt werden. Im unteren Plot soll die Veränderung der Populationsgrößen über der Zeit dargestellt werden, wobei die Änderung der Hasenpopulation als blau gestrichelte Linie und die Änderung der Fuchspopulation als rot gestrichelte Linie dargestellt werden soll. Aktivieren Sie bei allen Plots das Gitter.
- Ermitteln Sie den Einfluss der Wachstumsrate der Hasen auf das Räuber-Beute-System. Variieren Sie die Wachstumsrate  $a$  im Bereich von 0.01 bis 0.1 mit einem Intervall von 0.01. Für jede Wachstumsrate sollen die Populationsentwicklungen der Hasen und Füchse über einen Zeitraum von 0 bis 15 Jahren berechnet werden.
- Plotten Sie die Endwerte der Populationsgrößen von Hasen und Füchsen über der Wachstumsrate  $a$ . Verwenden Sie X-Markierungen für die Datenpunkte und verbinden Sie sie mit einer gestrichelten Linie.

### Aufgabe 3: Objekt-Orientierte Programmierung (20 Punkte)

Für die Vorauslegung von Rotorblättern von Windkraftanlagen sollen Sie ein Programm zur Optimierung von Balken schreiben. Dafür benötigen Sie eine Klasse, die die mechanischen und geometrischen Eigenschaften eines Balkens verarbeitet. Die Rotorblätter sollen näherungsweise als Einseitig eingespannte Balken unter Linienlast modelliert werden. Für die Optimierung sind neben den geometrischen Parametern die maximale Spannung im Balken sowie die maximale Durchbiegung entscheidend.

Gegeben sind die folgenden Formeln:

$$w_{max} = \frac{ql^4}{8EI} \quad M_b = -\frac{ql^2}{2}$$

$$\sigma_{max} = \frac{M_b}{W_b} \quad W_b = \frac{I}{z_{max}}$$

- Definieren Sie eine Klasse **Beam** mit der magischen Methode `__init__`, die die Parameter **modulus**, **inertia**, **length**, **z\_max** und **load** entgegennimmt. Speichern Sie alle Parameter in geschützten Attributen.
- Definieren Sie die Property **inertia** mit einem Setter, um auf das geschützte Attribut `_inertia` zuzugreifen.
- Implementieren Sie die Properties **sigma\_max** und **w\_max**, die die maximale Spannung im Balken  $\sigma_{max}$  als Verhältnis von  $M_b$  und  $W_b$  sowie die Enddurchbiegung  $w_{max}$  des Balkens abhängig vom aktuellen Flächenträgheitsmoment  $I$  zurückgeben.
- Definieren Sie die Funktion **inertia\_i\_beam**, die das Flächenträgheitsmoment eines I-Trägers in Abhängigkeit von der Breite  $b$ , der Höhe  $h$ , der Stegbreite  $t$  und der Gurthöhe  $a$  berechnet.
- Verwenden Sie die gegebenen Parameter  $b = 0.3m$ ,  $h = 1.5m$ ,  $t = 0.02m$  und  $a = 0.05m$  um das Flächenträgheitsmoment zu berechnen. Instanzieren Sie ein Objekt der Klasse **Beam** mit einer Länge und  $l = 80m$ , einem Modul  $E = 90GPa$  und einer Linienlast von  $q = 1500N/m$ . Nutzen Sie das Beam-Objekt, um die maximale Spannung  $\sigma_{max}$  und die maximale Durchbiegung  $w_{max}$  des Balkens in einem formatierten String auszugeben.