

Computergestützte Mathematik zur linearen Algebra – 10. Übungsblatt

Aufgabe 37: (*Determinanten berechnen*)

Auf der Homepage finden Sie die Datei `plusminusMatrix.py`, in der ein String der Länge 27^2 bestehend aus '+' und '-' angegeben ist.

- (a) Verwenden Sie die Funktion `string2ones_it` aus Aufgabe 9 (Musterlösung ist auf der Homepage) um den String in eine Matrix A der Größe 27×27 mit den Einträgen -1 und 1 umzuwandeln.
- (b) Beweisen Sie per Induktion, dass die Determinante jeder $(n \times n)$ -Matrix mit $n \geq 2$, die nur die Einträge 1 und -1 enthält, gerade ist.
- (c) Berechnen Sie die Determinante der Matrix A aus (a) auf die drei verschiedenen Arten:
 - (i) mit `numpy.linalg.det`,
 - (ii) mit Hilfe Ihrer LR-Zerlegungen aus Aufgabe 35, indem Sie das Produkt der Diagonalelemente von R bilden (Warum lässt sich so, bis auf das Vorzeichen, die gesuchte Determinante ausrechnen?),
 - (iii) symbolisch mit `sympy.det`. Hierzu müssen Sie die Matrix mit Hilfe von `sympy.Matrix(A)` zunächst in eine symbolische Matrix umwandeln.

Welches Ergebnis ist richtig?

- (d) Überlegen Sie sich, ob Ihre Funktion `mydet` aus Aufgabe 29 für die Matrix A das richtige Ergebnis liefern würde. Warum sollten Sie das in der Praxis besser nicht ausprobieren? Was ist für eine $(n \times n)$ -Matrix der Aufwand zur Berechnung der Determinante über den Laplaceschen Entwicklungssatz?

Aufgabe 38: (*Grenzen des Computers*)

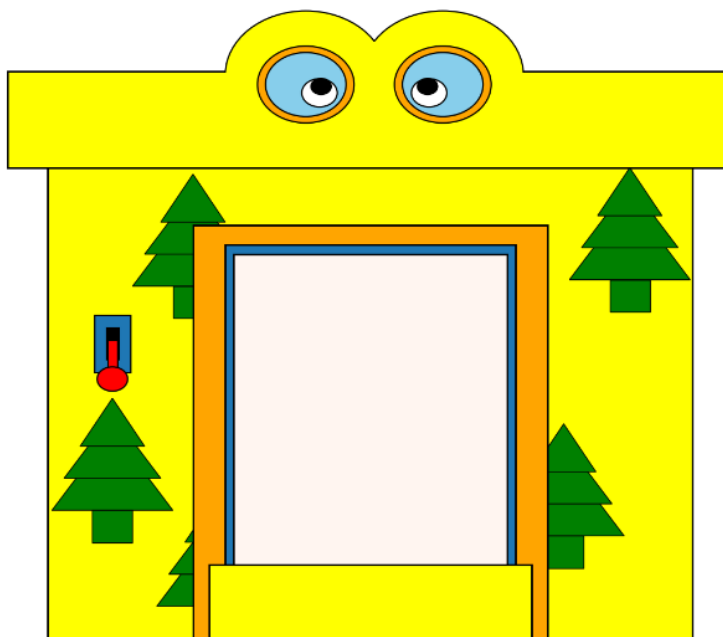
- (a) Definieren Sie die Variable `a=1.2 · 1034`. Was erwarten Sie bei folgender Rechnung: `10*a-9*a-a`? Überprüfen Sie Ihre Vermutung mit PYTHON. Wieso kommt dieses Ergebnis zustande?
- (b) Definieren Sie die Variable `b=12 · 1033`. Was erwarten Sie bei folgender Rechnung: `10*b-9*b-b`? Überprüfen Sie Ihre Vermutung mit PYTHON. Wieso kommt dieses Ergebnis zustande? Steht es im Widerspruch zu (a)?
- (c) Seien $p(x) = \frac{1}{2}(x-1)^7$ und $q(x) = \frac{1}{2}x^7 - \frac{7}{2}x^6 + \frac{21}{2}x^5 - \frac{35}{2}x^4 + \frac{35}{2}x^3 - \frac{21}{2}x^2 + \frac{7}{2}x - \frac{1}{2}$. Man kann leicht überprüfen, dass $p(x) = q(x)$ gilt.
Plotten Sie p und q einmal in einen gemeinsamen Plot über dem Intervall $[0.9, 1.1]$ und einmal in einen gemeinsamen Plot über dem Intervall $[0.99, 1.01]$. Sind die Ergebnisse wie erwartet? Was ist der Grund dafür? (Diese Teilaufgabe muss nicht mit der Polynomklasse gelöst werden.)

Aufgabe 39: (Fehlermeldungen erzeugen)

Schreiben Sie Code-Schnipsel, die die folgenden Fehlermeldungen erzeugen:

- (a) `NameError: name 'a' is not defined`
- (b) `TypeError: 'list' object is not callable`
- (c) `TypeError: 'int' object is not subscriptable`
- (d) `IndentationError: expected an indented block after 'for' statement`
- (e) `IndentationError: unindent does not match any outer indentation level`
- (f) `SyntaxError: invalid syntax`
- (g) `IndexError: list index out of range`
- (h) `IndexError: index 2 is out of bounds for axis 0 with size 2`
- (i) `TypeError: funktion1() takes 1 positional argument but 2 were given`
- (j) `TypeError: funktion2() missing 1 required positional argument: 'var2'`
- (k) `ValueError: matmul: Input operand 1 has a mismatch in its core dimension 0, with
gufunc signature (n?,k),(k,m?)->(n?,m?) (size 1 is different from 2)`
- (l) `----> 3 plt.plot(x,fx)`
...
...
`ValueError: x and y must have same first dimension, but have shapes (5,) and (4,)`

Hinweis: Bei allen Teilaufgaben wurde der Anfang der Fehlermeldung weggelassen. Bei (l) ist jedoch zusätzlich die Zeilennummer und Zeile, in der der Fehler aufgetreten ist, zu sehen.



Aufgabe 40: (*Weihnachts-Aufgabe - Geschenk von Luca Püttmann*)

Die Weihnachtszeit steht vor der Tür. Der Weihnachtsmann und seine Gehilfen, die Elfen Trixi, Gilfi und Jordi sowie der Eisbär Balbo, sind inmitten der Vorbereitung, die Weihnachtswünsche der Kinder herzustellen. Doch plötzlich fängt die Spielzeug-Maschine an, stark zu rauchen und zu ächzen bis sie völlig streikt. Als Jordi nachsieht, muss er leider feststellen, dass die Mechanik stark beschädigt wurde. Hatten etwa Grantelbart und sein Assistent Gugor mal wieder ihre Hände im Spiel, um Weihnachten zu ruinieren? Nun muss Jordi jedenfalls versuchen, sie wieder schnellstmöglichst funktionstüchtig zu machen.

Helfen Sie Jordi dabei, die Spielzeug-Maschine zu reparieren und somit das Weihnachtsfest zu retten, indem Sie folgendermaßen vorgehen:

- (a) Korrigieren/Vervollständigen Sie den fehlerhaften Code in der *Klasse Spielzeug_Maschine*.
- (b) Stellen Sie nun die durch den Zwischenfall nicht fertig gewordenen Geschenke für *Benôit Mandelbrot* und *Sophie Germain* her.
 - Benôit wünscht sich ein *Fraktal*. Am liebsten möchte er das sogenannte “Apfelmännchen”.
 - Sophie wünscht sich eine möglichst große *Ulam-Spirale*, aber mit den *Germainischen Primzahlen*.

Hinweise:

Die (a) hat 10 Fehler.

Für die (b) ist es nützlich, Folgendes zu wissen:

- `spielzeug = Spielzeug_Maschine(plot_name, name='Name des Kindes', state='on')`
`spielzeug.toy_maker()`,
Bedienweise und Init sowie Ausführung für den Plot
- `inpos = [0.348, 0.211, 0.283, 0.3705]` ist hilfreich, sollte der Plot aus der *gift_area* heraus ragen.

Man beachte: das folgende Format ist einzuhalten: `plot_gift = plot_name(ax, ...)`. Dabei muss `ax` als erstes Argument an die Funktion übergeben werden. Zusätzlich, durch “...” angedeutet, können weitere Argumente folgen, wenn entsprechend so definiert.

Bonusaufgabe: Lassen Sie Ihrer Phantasie freien Lauf und gestalten Sie ein besonderes Geschenk für ein besonders artiges Kind und/oder erweitern Sie die Spielzeugmaschine oder fügen Sie eine Geschenkeinpackmaschine hinzu, all dies z.B. durch Animationen, Plot in 2D oder 3D, Hintergrundmusik, Interaktivität, etc.. Alles ist erlaubt!

In der Vorlesung wird das beste Geschenk prämiert. Schicken Sie hierzu Ihre `.py` oder `.ipynb` Datei mit Ihrem Geschenk unter Angabe Ihres Namens per Mail bis zum Mittwoch, 18.12.24, 23:59 Uhr an schaedle@hhu.de.

Besprechung in den Übungen vom 16.12.2024 bis 20.12.2024.