Karlsruher Institut für Technologie
Institut für Theorie der Kondensierten Materie
Dr. Andreas Poenicke
http://comp.physik.kit.edu/Lehre/ERA/

SS 2022

12.04.2022

era@physik.kit.edu

Übungsblatt Nr. 5 zur Vorlesung „Einführung in das Rechnergestützte Arbeiten“

Die Übungen sollen auf dem Jupyter-Server der Fakultät bearbeitet werden. Vorraussetzung dafür ist nur ein Webbrowser und ein Poolraum-Account.
Der Jupyter-Server ist unter `https://jupytermachine.etp.kit.edu` zu erreichen.
Eine Kurzübersicht über die wichtigsten Python Anweisungen und Funktionen ist auf dem Python Spickzettel zusammengefasst.

1 NumPy

NumPy ist ein Python-Modul, dass numerische Berechnungen mit Vektoren und Matrizen deutlich vereinfacht. Die Funktionalität orientiert sich dabei stark an dem Programm Matlab. Wie bei diesem werden dabei hochoptimierte Numerik-Bibliotheken für die performante Berechnung verwendet.

NumPy definiert einen neuen Datentyp NumPy-Arrays (`ndarray`) der Basis der meisten Methoden ist. Da die mathematischen Funktionen aus dem `math` idR. nur mit Skalaren umgehen können, sind analoge Funktionen in NumPy neu implementiert. Auch mathematische Konstanten wie π findet man hier.

1. Binden Sie das Modul `numpy` ein.
2. Berechnen Sie ein `x_max` $= 4\pi$. Generieren Sie einen NumPy-Array `x` der $N = 101$ Werte zwischen 0 und `x_max` hat.
3. Berechnen Sie (in einem Schritt) $\sin(x)^2 + \cos(x)^2$ für alle diese Stützstellen.
4. Numpy bietet viele Funktionen die Eigenschaften von NumPy-Arrays berechnen. So lässt sich z.B. mit einer Funktion die Summe aller Elemente eines Arrays berechnen.
   Berechnen Sie einen weiteren Array `y`=$\sin(x)$ und nutzen Sie diese Summation (d.h. keine Schleife!) und den Array `y` um näherungsweise das Integral von 0 bis x_{max} über die Sinus-Funktion zu berechnen.
   Nutzen Sie dabei in einfachster Näherung für das Integral die Trapezformel

$$\int_a^b f(x) \approx \frac{a-b}{N-1}\left(\frac{f(x_{N-1})+f(x_0)}{2} - \sum_{n=0}^{N-1} f(x_n)\right) .$$

   Wobei N hier die Anzahl der äquidistanten(!) Stützstellen x_n im Intervall $[a, b]$ ist.
   Hinweis: Berechnen Sie für einen Numpy-Array `x` die Funktion `y` $= f($`x`$)$, gilt danach für jedes Element $\mathtt{y}_n$ des Arrays `y` das $\mathtt{y}_n = f(\mathtt{x}_n)$

2 Matplotlib

Matplotlib ist das zweite wichtige Modul, das bei Pythonprogrammen in der Physik regelmäßig zum Einsatz kommt. Diese Bibliothek bietet einfache Möglichkeiten wissenschaftliche Graphen zu erstellen. Matplotlib kann dabei problemlos mit numpy-Arrays umgehen, die beiden Pakete werden daher meist zusammen eingesetzt.

Eine gute Übersicht über die Fähigkeiten des Moduls bietet die Matplotlib-Galerie. Bei allen Beispielen hier wird auch jeweils das Programm gezeigt, mit dem sie erstellt wurden.

1. Binden Sie nun zusätzlich aus `matplotlib` das Untermodul `pyplot` ein.
2. Übernehmen Sie Ihre Funktion `my_sin(x)` vom letzten Blatt (oder aus dem Lösungsvorschlag dazu) und berechnen Sie[1] `y2=my_sin(x)`.
3. Vergleichen Sie diesen Funktionsverlauf mit dem von `math.sin(x)` in dem Sie beide in einem Graphen darstellen.
4. Ergänzen Sie nun den Plot um x- und y-Achsenbeschriftung sowie einer Legende.
5. Als letzter Schritt geben Sie den Graphen nicht auf den Bildschirm aus, sondern speichern Sie den Graphen in eine Datei `vergleich.pdf`.

3 SciPy - Curvefit

Die SciPy-Bibliothek ist eine umfangreiche Sammlung numerischer Methoden[2] aus den verschiedensten Bereichen der Mathematik. Im folgenden benutzen wir hieraus nur die Funktion `curve_fit` mit der man einen theoretischen Funktionsverlauf an experimentelle Daten anpassen („fitten“) kann.

1. Binden Sie aus `scipy.optimize` nur die Funktion `curve_fit` ein.
2. Laden Sie (wie in der Vorlesung gezeigt) die Messdaten aus den beiden Spalten der Datei `data.txt` in die Arrays `x` und `y`.
3. Definieren Sie eine Funktion `linear(x,a,b)` die $y = a \cdot x + b$ berechnet und zurück gibt.
4. Nutzen Sie nun `curve_fit()` um an die Messdaten einen linearen Verlauf anzupassen.
   Bestimmen Sie damit die Steigung a und Abzisse b sowie deren Standardabweichungen.
5. Plotten Sie Daten gemeinsam mit der angepassten linearen Kurve in einem Graphen. Die Daten sollten dabei nicht verbunden sondern durch das Symbol `+` dargestellt werden.
   Hinweis: Zur Berechnung der linearen Kurve können Sie wieder Ihre Funktion `linear(x,a,b)` *verwenden.*

[1] (`x` sollte noch immer das Array aus Aufgabe 1 mit N=101 Werten zwischen 0 und `x_max` sein.)

[2] Unter `https://docs.scipy.org/doc/scipy/reference/` findet man, neben der ausführlichen Dokumentation, zu vielen Methoden eine Einführung mit Beispielen.