Institut für Theoretische Physik
der Universität zu Köln

Prof. Dr. Simon Trebst
Michael Becker, Peter Bröcker

Computerphysik

Übungsblatt 0

SS 2013

Website: http://www.thp.uni-koeln.de/trebst/Lectures/2013-CompPhys.html
Abgabedatum: *Anwesenheitsübung erste Semesterwoche*

Infos zum Übungsbetrieb

Punktevergabe

Auf den folgenden Übungsblättern werden Sie dreierlei Aufgabentypen finden. Die erste Gruppe besteht aus **Programmieraufgaben**, die Ihnen den Einstieg in Python so einfach wie möglich machen sollen. Für diese Aufgaben werden keine Punkte vergeben. Die zweite Art Aufgaben sind **Pflichtaufgaben**, die sich um die in der Vorlesung besprochenen numerischen Methoden drehen. Für diese werden Punkte vergeben von denen Sie am Ende mindestens 50% erreicht haben müussen um zur Klausur zugelassen zu werden. Alle Programmierkenntnisse, die Sie zum Lösen dieser Aufgaben benötigen, werden Sie durch die Aufgaben aus der ersten Gruppe erlernen. Die dritte Gruppe besteht aus **weiterführenden Anwendungen** der erlernten Kenntnisse. Diese *optionalen* Aufgaben bauen auf den Pflichtaufgaben auf und illustrieren vor allem die vielseitigen, praktischen Anwendungen in der Physik. Durch deren Bearbeitung können Sie außerdem **Bonuspunkte** sammeln.

Abgabemodus

Die Abgabe erfolgt **zweiwöchentlich**. Sie haben also für die Bearbeitung der Aufgaben zwei Wochen Zeit, was unter anderem die Möglichkeit eröffnet, in den Übungsgruppen Fragen zu den aktuellen Aufgaben stellen zu können.

---

1. Hello world

Schreiben Sie ein sogenanntes **Hello world** Programm in Python, d.h. ein Programm, welches die einfach Ausgabe “Hello world!” erzeugt.

2. Eulersche Zahl

Schreiben Sie ein Programm zur Berechnung der **Eulerschen Zahl** e,

$$e = \sum_{k=0}^{\infty} \frac{1}{k!}$$

indem Sie *iterativ* die ersten n Terme dieser Reihe aufsummieren, d.h. berechnen Sie

$$e(n) = \sum_{k=0}^{n} \frac{1}{k!}$$

für $n = 1, 2, 3, \ldots, 32$. Untersuchen Sie die Konvergenz dieser Reihe, indem Sie die Abweichung $e - e(n)$ auftragen gegen n. Für welche Werte von n ist die Abweichung kleiner als 10^0, 10^{-1}, 10^{-2}, 10^{-3}?

3. Dezimalzahlen

Erstellen Sie ein Programm bestehend aus dem folgenden Code und führen Sie es aus. Hätten Sie das Ergebnis erwartet?

```
1  a = 0.1
2  b = 0.2
3  c = 0.3
4
5  print (a + b) == c
```

Überlegen Sie wie Sie dieses Problem lösen können.

4. Funktionen plotten

Um Daten am Computer grafisch darzustellen gibt es verschiedene Möglichkeiten. In python bietet sich das Paket *matplotlib* an. Die Dokumentation finden Sie auf der Website des Projekts *http://matplotlib.org*.

Lesen Sie den folgenden Quellcode und machen Sie sich klar was passiert.

```
1  import matplotlib.pylab as plt
2  import numpy as np
3
4  plt.figure(figsize=(10,6))
5
6  for d in [4, 20, 500]:
7      x_values = np.linspace(0, 4, d)
8      y_values = [ np.sqrt(x) for x in x_values ]
9      plt.plot(x_values, y_values, label=str(d))
10
11 plt.xlabel("x");
12 plt.ylabel("y");
13 plt.legend()
14
15 plt.savefig("test.pdf")
16 plt.show()
```

5. Zweidimensionale Daten

Mehrdimensionale Abbildungen haben Sie spätestens im Laufe Ihres Studiums schon kennengelernt. Was sich per Hand schwer zeichnen lässt, ist am Computer oft leicht zu realisieren. Sie haben außerdem die Wahl zwischen verschiedenen Darstellungen, die je nach Art der darzustellenden Daten unterschiedlich gut geeignet sein können. Wie in der vorherigen Aufgabe haben wir einen fertigen Code (http://www.thp.uni-koeln.de/trebst/Lectures/CompPhys-2013/plotting_2d.py) zusammengestellt mit dem Sie experimentieren können. Versuchen Sie zum Beispiel die Maschenweite des Gitternetzes zu verändern. Wie verändern sich die Variablen *X, Y* durch den Befehl *meshgrid*? Besuchen Sie auch einmal die Homepage *matplotlib.org* und finden Sie heraus, wie sie die Achsen des ersten Bildes so anpassen können, dass die korrekten Einheiten gezeigt werden. Im Code sind schon Hinweise darauf enthalten, wie man zwei Graphen in einem Bild darstellt. Vereinen Sie die beiden Konturplots und speichern Sie das resultierende Bild als PDF.

```
1  from mpl_toolkits.mplot3d import axes3d
2  import matplotlib.pylab as plt
3  import numpy as np
4
5  def f_plus(x,y):
6      return np.sqrt(3+2*np.cos(np.sqrt(3)*y)
7                     +4*np.cos(x)*np.cos(np.sqrt(3)/2*y))
8
9  def f_minus(x,y):
10     return -np.sqrt(3+2*np.cos(np.sqrt(3)*y)
11                     +4*np.cos(x)*np.cos(np.sqrt(3)/2*y))
12
13 X = np.arange(-3.14, 3.14, 0.1)
14 Y = np.arange(-3.14, 3.14, 0.1)
15
16 Z_plus = [[f_plus(x,y) for x in X] for y in Y]
17 Z_minus = [[f_minus(x,y) for x in X] for y in Y]
18
19 fig = plt.figure()
20 plt.subplot(111)
21 plt.imshow(Z_plus)
22 plt.show()
23
24 X,Y = np.meshgrid(X,Y)
25
26 fig = plt.figure()
27 plt.subplot(111)
28 plt.contour(X, Y, Z_plus, vmin=0, vmax=3)
29 plt.show()
30
31 fig = plt.figure()
32 ax = fig.add_subplot(111, projection='3d')
33 ax.plot_surface(X, Y, Z_plus, cmap='coolwarm', linewidth=0,
34                 antialiased=True, vmin=-3, vmax=3, rstride=1, cstride=1)
35 ax.plot_surface(X, Y, Z_minus, cmap='coolwarm', linewidth=0,
36                 antialiased=True,vmin=-3, vmax=3, rstride=1, cstride=1)
37 plt.show()
38
39 fig = plt.figure()
40 ax = fig.add_subplot(111, projection='3d')
41 ax.plot_wireframe(X, Y, Z_plus, color='black', rstride=2, cstride=2)
42 ax.plot_wireframe(X, Y, Z_minus, color='black', rstride=2, cstride=2)
43 plt.show()
```