Institut für Theoretische Physik
der Universität zu Köln

Prof. Dr. Simon Trebst
Michael Becker, Peter Bröcker

Computerphysik

Übungsblatt 5

SS 2013

Website: http://www.thp.uni-koeln.de/trebst/Lectures/2013-CompPhys.html
Abgabedatum: Montag, 27. Mai 2013 vor Beginn der Vorlesung

20. Einfache Animationen

Programmiertechniken

Oft ist es bei der Visualisierung des zeitlichen Ablaufs von physikalischen Prozessen hilfreich, diese als kleine **Animationen** darzustellen. In dieser Aufgabe wollen wir Ihnen einige elementare Animationstechniken in Python zeigen, welche Sie dann bei den übrigen Übungsaufgaben zum Einsatz bringen können.

In einem ersten Beispiel (siehe animation_simple.py) betrachten wir eine einfache Animation, welche das Zeichnen einer bunten Linie nach dem Prinzip des Daumenkinos visualisiert: Nach einer kurzen Pause wird das *gesamte* Bild durch eine neuere Version ersetzt. Im unserem konkreten Fall wird dabei jedes Mal ein weiterer Punkt der Linie zum neuen Bild hinzugefügt.

```
import matplotlib.pyplot as plt
import numpy as np

resolution = 80
pixels = np.zeros((resolution, resolution))

p = plt.imshow(pixels, interpolation='none')
p.set_cmap('spectral') # Andere nette Colormaps sind "hot", "gray", etc.
plt.clim(0.0, 1.0) # Setze das Intervall fuer die Farbwerte
plt.colorbar() # Zeichne eine Legende fuer die Farben

# Zeichne diagonale Linie von (0, 0) bis (resolution-1, resolution-1)
for i in range(resolution):
    pixels[i,i] = float(i)/resolution # Setze den Farbwert bei (i, i)
    p.set_data(pixels) # Erneuere die Plot-Daten
    plt.pause(0.05) # Pause in Sekunden

# Dieser show-Befehl sorgt dafuer, dass das Fenster am Ende
# der Programmausfuehrung nicht automatisch geschlossen wird.
plt.show()
```

Ein weiterer Animationseffekt, den Sie etwa bei der Visualisierung der Mondbahnen in Aufgabe 22 verwenden können, entsteht, wenn sie bei der Animation eines Punktes in jedem Schritt den Farbwert der bereits vorhandenen Punkte etwas verringern. Dazu können wir eine der speziellen

Eigenschaften der NumPy-Listen verwenden – bei diesen können arithmetische Operationen auf alle Listeneinträge simultan angewandt werden. Ein Beispiel geben wir Ihnen hier (siehe auch animation_tail.py) :

```
import matplotlib.pyplot as plt
import numpy as np

resolution = 80
# Hier erzeugen wir ein NumPy-Array gefuellt mit Nullen.
pixels = np.zeros((resolution, resolution))

p = plt.imshow(pixels, interpolation='none')
p.set_cmap('hot')
plt.clim(0.0, 1.0)
plt.colorbar()

x, y = -30, -30
dx, dy = 1, 0

for t in range(1,1000):
    pixels *= 0.9 # Hier wird jeder Eintrag der Liste
                  # mit 0.9 multipliziert. Dies funktioniert
                  # allerdings nur bei NumPy-Arrays.

    pixels[40+y,40+x] = 1.0 # Setze den neuen Punkt

    if t % 60 == 0: # Wir drehen die Richtung um 90 Grad
        dx, dy = -dy, dx

    x += dx
    y += dy

    p.set_data(pixels) # Erneuere die Plot-Daten
    plt.pause(0.001) # Pause in Sekunden

plt.show()
```

21. Lorenz-Attraktor *5 Punkte*

Für die Simulation eines einfachen **Wetter-Modells** hat der Meteorologe Edward Lorenz 1963 eine Beschreibung von Luftströmungen entworfen. Dazu hat er ein Gleichungssystem von drei gekoppelten Differentialgleichungen betrachtet:

$$\dot{X}(t) = a\,(Y(t) - X(t)) \tag{1}$$
$$\dot{Y}(t) = X(t)\,(b - Z(t)) - Y(t) \tag{2}$$
$$\dot{Z}(t) = X(t)Y(t) - cZ(t). \tag{3}$$

Berechnen Sie mittels des Runge-Kutta-Verfahrens 4. Ordnung eine Lösung für $X(t)$, $Y(t)$ und $Z(t)$ und stellen sie diese für $t \in [0, 100]$ in einem 3D-Plot dar – Sie werden damit den sogenannten **Lorenz-Attraktor** visualisieren. Wählen Sie dabei eine Schrittweite von $\Delta t = 0.01$ und verwenden Sie die folgenden Parameter und Startwerte:

$$a = 10,\ b = 28,\ c = 8/3,$$
$$X(0) = 0,\ Y(0) = 1,\ Z(0) = 1.05.$$

Wie sich erst im Laufe der nächsten Dekaden herausstellen sollte, hat Edward Lorenz mit diesem einfachen Modell einen ganz neuen Forschungszweig gestartet – die sogenannte **Chaos-Forschung**. Lorenz selbst war sich der weitreichenden Implikationen seiner Beobachtung, daß die zeitliche Entwicklung eines relativ einfachen *deterministischen* Systems höchst sensibel von den gewählten Startbedingungen abhängen kann, voll bewußt und hat schon früh den Begriff des **Schmetterlings-Effekts** geprägt. Wer sich zu diesen historischen Entwicklungen weiter informieren möchte, der sei auf diesen Artikel zum 50-jährigen Jubiläum der Entdeckung des Lorenz-Attraktors hingewiesen.

22. Monde auf der Überholspur

5 Punkte

Die Monde **Janus und Epimetheus** umkreisen den Planeten Saturn *koorbital*, d.h. auf nahezu der gleichen Bahn. Bemerkenswerterweise ist dabei der Bahnunterschied zwischen dem inneren und dem äußeren der beiden Monde deutlich kleiner als deren Durchmesser. Alle vier Jahre begegnen sich die beiden Monde, da der innere Mond ein wenig schneller unterwegs ist als der äußere. Wieso es dabei nicht zu einer Kollision kommt, wollen wir in dieser Aufgabe untersuchen.

Dazu wollen wir das **3-Körper-System** aus Saturn, Janus und Epimetheus simulieren. Für die Simulation bewegen wir uns ins Bezugssystem des Saturn. Zwischen zwei Körpern wirkt dann eine Gravitationskraft gemäß

$$\mathbf{F}_{1\leftarrow 2} = -G\frac{m_1 m_2}{|\mathbf{r}_1 - \mathbf{r}_2|^3}\left(\mathbf{r}_1 - \mathbf{r}_2\right), \tag{4}$$

wobei wir die Gravitationskonstante $G = 1$ setzen (wodurch sich die Größen der Parameter natürlich von den realistischen unterscheiden). Berechnen Sie die Bahnen der Monde mittels des **Verlet**-Verfahrens und plotten Sie die Bewegung der Monde als eine Animation. Verwenden Sie folgende Startwerte:

$$\begin{aligned} m_1 &= 1 \\ m_2 &= 4 \\ m_{\text{Saturn}} &= 4\times 10^4 \\ \mathbf{r}_1 &= (-155, 0) \quad \frac{d\mathbf{r}_1}{dt} = (0, -16.1) \\ \mathbf{r}_2 &= (150, 0), \quad \frac{d\mathbf{r}_2}{dt} = (0, 16.3) \end{aligned}$$

Für die Animation sollten Sie den Ausschnitt $x \in (-200, 200), y \in (-200, 200)$ zeichnen. Geben Sie außerdem für jeden Zeitschritt den Abstand der Monde vom Planeten in eine Datei aus. Wenn Sie anschließend diese Abstände vergleichen, sollten sie deutlich sehen können, wie die Monde einer Kollision ausweichen. Erläutern Sie, wie dies von statten geht.