

2Körperproblem

November 21, 2022

0.1 Numerische Lösung eines Zweikörperproblems

Notebook erstellt am 20.11.2022 von C. Rockstuhl, überarbeitet von Y. Augenstein

Wir wollen in diesem Notebook das Zweikörperproblem lösen, wie es im Rahmen der Planetenbewegung diskutiert wurde. Es gibt verschiedene Möglichkeiten, sich diesem Thema zu nähern. Wir würden hier ganz allgemein zwei Körper betrachten, die sich gegenseitig über die Gravitationskraft anziehen und die sich in einem Inertialsystem befinden. Die beiden Körper sind charakterisiert durch ihre Massen m_1 und m_2 . Es gibt darüberhinaus keine weiteren Kräfte, welche auf diese beiden Körper wirken.

0.1.1 Setting

Geometrische Größen, die das System beschreiben, sind die absoluten Raumkoordinaten der beiden Körper

$$\mathbf{r}_1 = x_1 \mathbf{e}_x + y_1 \mathbf{e}_y + z_1 \mathbf{e}_z$$

und

$$\mathbf{r}_2 = x_2 \mathbf{e}_x + y_2 \mathbf{e}_y + z_2 \mathbf{e}_z \quad .$$

Darüberhinaus können wir auch noch die Schwerpunktskoordinate definieren

$$\mathbf{R} = \frac{m_1 \mathbf{r}_1 + m_2 \mathbf{r}_2}{m_1 + m_2}$$

und die Relativkoordinate

$$\mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1 \quad .$$

Die von Körper 2 auf Körper 1 wirkende Gravitationskraft ist gegeben als

$$\mathbf{F}_{12} = G \frac{m_1 m_2}{|\mathbf{r}|^2} \frac{\mathbf{r}}{|\mathbf{r}|}$$

und die von Körper 1 auf Körper 2 wirkende Gravitationskraft ist gerade genau entgegengesetzt und gleich groß (*actio=reactio*), $\mathbf{F}_{12} = -\mathbf{F}_{21}$

$$\mathbf{F}_{21} = -G \frac{m_1 m_2}{|\mathbf{r}|^2} \frac{\mathbf{r}}{|\mathbf{r}|} \quad .$$

Die von uns zu lösenden Bewegungsgleichungen lauten also:

$$\begin{aligned} m_1 \ddot{\mathbf{r}}_1 &= \mathbf{F}_{12} \\ m_2 \ddot{\mathbf{r}}_2 &= \mathbf{F}_{21} \quad . \end{aligned}$$

Nach Division durch die Masse können wir die entsprechenden Bewegungsgleichungen für die Ortskoordinate konkretisieren zu

$$\begin{aligned} \ddot{\mathbf{r}}_1 &= G m_2 \frac{\mathbf{r}}{|\mathbf{r}|^3} \\ \ddot{\mathbf{r}}_2 &= -G m_1 \frac{\mathbf{r}}{|\mathbf{r}|^3} \quad . \end{aligned}$$

Beachten Sie, dass es sich hierbei um insgesamt 6 gekoppelte Differentialgleichungen handelt, die wir im Folgenden numerisch lösen müssen. In Komponenten aufgeschrieben lauten diese

$$\begin{aligned} \ddot{x}_1 &= G m_2 \frac{x_2 - x_1}{|\mathbf{r}|^3} \\ \ddot{y}_1 &= G m_2 \frac{y_2 - y_1}{|\mathbf{r}|^3} \\ \ddot{z}_1 &= G m_2 \frac{z_2 - z_1}{|\mathbf{r}|^3} \\ \ddot{x}_2 &= -G m_1 \frac{x_2 - x_1}{|\mathbf{r}|^3} \\ \ddot{y}_2 &= -G m_1 \frac{y_2 - y_1}{|\mathbf{r}|^3} \\ \ddot{z}_2 &= -G m_1 \frac{z_2 - z_1}{|\mathbf{r}|^3} \quad . \end{aligned}$$

0.1.2 Numerische Differenzierung

Genauso wie in früheren Beispielen müssen wir als Erstes diese Differentialgleichungen zweiter Ordnung in Differentialgleichungen erster Ordnung überführen. Wir wählen hier wieder den Ort und die Geschwindigkeit der beiden Körper als die Größen aus, die wir durch Differentialgleichungen erster Ordnung beschreiben werden. Wir werden eine Vektornotation wählen, in der die zu beschreibenden Größen in einen einzelnen Vektor $\mathbf{r}$ geschrieben werden. Dieser Vektor ist dann definiert als

$$\mathbf{r} = \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \dot{\mathbf{r}}_1 \\ \dot{\mathbf{r}}_2 \end{pmatrix} .$$

Die zeitliche Entwicklung wird dann durch folgende Differentialgleichung beschrieben:

$$\frac{d\mathbf{r}}{dt} = \dot{\mathbf{r}} = \begin{pmatrix} \dot{\mathbf{r}}_1 \\ \dot{\mathbf{r}}_2 \\ Gm_2 \frac{\mathbf{r}}{|\mathbf{r}|^3} \\ -Gm_1 \frac{\mathbf{r}}{|\mathbf{r}|^3} \end{pmatrix} .$$

Es ist diese Differentialgleichung, die wir jetzt im Folgenden implementieren möchten. Auch hier ist wieder offensichtlich, dass wir genauso viele Anfangsbedingungen benötigen wie wir Differentialgleichungen erster Ordnung haben. Wir benötigen also den Anfangsort und die Anfangsgeschwindigkeit der beiden Massen.

Der folgende Code ist, wie die Ausführungen weiter oben, komplett drei-dimensional aufgeschrieben. Wir würden aber die dritte Raumkoordinate zu null setzen und auch keine Anfangsgeschwindigkeit oder Anfangsposition in dieser Ebene forcieren. Wir können daher effektiv die Bewegung in einer Ebene diskutieren, was die Darstellung erleichtert.

```
[1]: import numpy as np
      from scipy.integrate import solve_ivp
      import matplotlib.pyplot as plt
```

```
[2]: def two_body_problem(t, y, G, m1, m2):
      """
      Die hier beschriebene Differentialgleichung beschreibt die
      ↪Bewegungsgleichung der
      beiden Körper mit den beiden Massen m1 und m2.
      t: diskreter Zeitschritt (wird formal immer benötigt, auch wenn die DGL
      ↪nicht explizit von der Zeit abhängt)
      y: Zustandsvektor, den wir übergeben und dessen Bewegung, also seine
      ↪zeitliche Veränderung, durch die DGL beschrieben wird.
      """
      # Berechne hier den Abstandsvektor zwischen den beiden Massen
      r1 = y[:3]
      r2 = y[3:6]
      v1v2 = y[6:]
      r_mag = np.linalg.norm(r2 - r1)
      c1 = G * m2 * (r2 - r1) / r_mag**3
      c2 = -G * m1 * (r2 - r1) / r_mag**3
      return np.concatenate([v1v2, c1, c2])
```

Nach der Definition der Differentialgleichung, die die zeitliche Entwicklung von Ort und Geschwindigkeit beschreibt, können wir die eigentliche Simulation laufen lassen.

```
[3]: # Gravitationskonstante
G = 6.67259e-20 # (km**3/kg/s**2)

# Zeitintervall (Anfang und Ende), innerhalb dessen wir die DGL lösen
↳ möchten
time_interval = [0, 1000]

# Details zum Körper 1. Wir geben diesem hier zunächst eine sehr hohe Masse
↳ und setzen
# ihn in die Mitte des Koordinatensystem und geben dem Körper keine
↳ Anfangsgeschwindigkeit.
# In guter Näherung können wir so ein System bestehend aus Sonne und Erde
↳ beschreiben.
m1 = 1e26 # Masse (kg)
r10 = np.array([0, 0, 0]) # Anfangsort (km)
v10 = np.array([0, 0, 0]) # Anfangsgeschwindigkeit (km/s)

# Details zum Körper 2
m2 = 1e23 # Masse (kg)
r20 = np.array([3000, 0, 0]) # Anfangsort (km)
v20 = np.array([0, 40, 0]) # Anfangsgeschwindigkeit (km/s)

# [x1 (0), y1 (1), z1 (2), x2 (3), y2 (4), z2 (5), vx1 (6), vy1 (7), vz1
↳ (8), vx2 (9), vy2 (10), vz2 (11)]
y0 = np.concatenate((r10, r20, v10, v20))

y = solve_ivp(two_body_problem, time_interval, y0, method='DOP853', args=(G,
↳ m1, m2), dense_output=True)
```

Nun berechnen wir ein paar relevante Größen (insbesondere die Positionen der beiden Körper) an einer Reihe von uns gewählten Zeitpunkten.

```
[4]: t = np.linspace(*time_interval, 1000)

sol = y.sol(t)
r1 = sol[:3]
r2 = sol[3:6]
```

Alles was jetzt kommt dient nur der Visualisierung.

```
[43]: plt.rcParams["axes.grid"] = True
plt.style.use("dark_background")
```

```

fig = plt.figure(figsize=plt.figaspect(0.5))

ax1 = fig.add_subplot(1, 2, 1)

ax1.plot(*r1[:2], "cyan")
ax1.plot(*r2[:2], "red")
ax1.set_xlabel("X (km)")
ax1.set_ylabel("Y (km)")

ax2 = fig.add_subplot(1, 2, 2, projection="3d")

ax2.plot3D(*r1, "cyan")
ax2.plot3D(*r2, "red")

ax2.set_xlabel("X (km)")
ax2.set_ylabel("Y (km)")
ax2.set_zlabel("Z (km)")

fig.suptitle("Zwei-Körper Problem")
fig.tight_layout()
plt.show()

```

