

Die Kepler Gleichung, Ausblick auf das Dreikörperproblem

Martin Lehner, Rodersdorf, Schweiz

9. Juni 2026

Inhaltsverzeichnis

1	Vorbemerkungen	1
2	Ellipsen und die Keplersche Gleichung	2
2.1	Die Mittelpunktsgleichung der Ellipse	2
2.2	Drehimpulserhaltung und Herleitung der Keplerschen Gleichung	3
2.3	Resultate und Lösungshinweise, Teil 1	6
2.4	Numerische Lösung der Keplerschen Gleichung mit einem Python- Programm	6
3	Einfaches (zweidimensionales) Beispiel zum Dreikörperpro- blem	9
4	Schlusswort	12
5	Verzeichnisse und Quellenangaben	13

1 Vorbemerkungen

In dieser kurzen Abhandlung geht es um die Mittelpunktsgleichung der Ellipse und um die zeitliche Bewegung eines Himmelskörpers beim sogenannten Zweikörperproblem. Die Unterrichtseinheit richtet sich nach dem Niveau der Schweizer Gymnasien im Schwerpunktfach PAM. Am Schluss des Dokuments findet man eine kleine Einführung in den einfachsten Fall des Dreikörperproblems.

2 Ellipsen und die Keplersche Gleichung

2.1 Die Mittelpunktsgleichung der Ellipse

In diesem Beitrag beschränken wir uns auf die einfachste Lage der Ellipsen im Koordinatensystem: Der Mittelpunkt M der Ellipse soll im Nullpunkt $(0 | 0)$ liegen. Die Ellipsengleichung 1 ergibt sich in dem eine axiale Affinität (mit der x -Achse als Affinitätsachse und dem Affinitätsverhältnis b/a) auf einen Kreis mit der Gleichung $x^2 + y^2 = a^2$ (Radius a) angewendet wird (siehe Aufgabe 5). a und b heissen grosse bzw. kleine Halbachse, den Abstand c zwischen Mittelpunkt und einem der Brennpunkte nennt man lineare Exzentrizität.

Johannes Kepler veröffentlichte im Jahr 1609 die bahnbrechende Entdeckung, dass sich die Planeten auf Ellipsenbahnen bewegen, wobei sich die Sonne in einem der beiden Brennpunkte $F_{1,2}$ befindet. Kepler gelangte durch sorgfältige (mathematische) Auswertung der Beobachtungsdaten des dänischen Astronomen Tycho Brahe zu seinen Erkenntnissen.

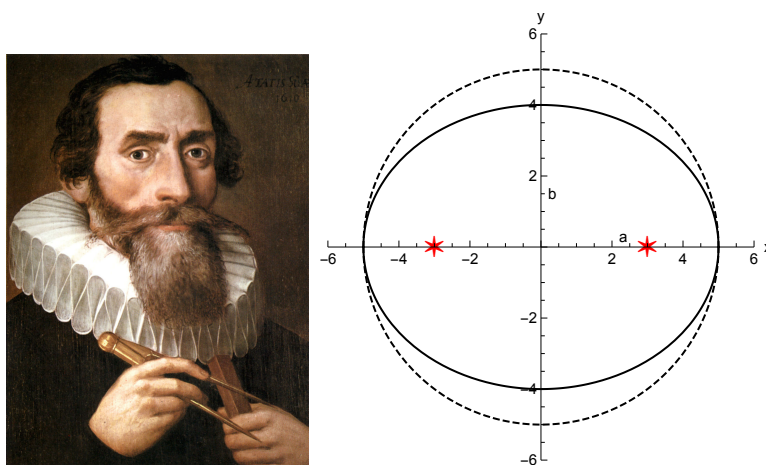


Abbildung 1:

Links: Portrait von Johannes Kepler (1610), Rechts: Ellipse mit $a = 5$ und $b = 4$, die Sterne markieren die beiden Brennpunkte $F_{1,2}$.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad \text{mit} \quad c^2 = a^2 - b^2 \quad (1)$$

Gemäss der zweiten Gleichung in 1 ist a auch die Entfernung zwischen den Nebenseiteln und den Brennpunkten.

Aufgabe 1.1) Bestimmen Sie a , b und die Koordinaten der Brennpunkte

für $x^2 + 4y^2 = 100$. Stellen Sie die Ellipse im Koordinatensystem graphisch dar.

Aufgabe 1.2) Eine Ellipse mit Mittelpunkt $(0 | 0)$ geht durch die Punkte $P(2 | 2)$ und $Q(5 | 1)$. Bestimmen Sie die Gleichung der Ellipse, die Halbachsen und die numerische Exzentrizität $\epsilon = c/a$.

Aufgabe 1.3) Berechnen Sie den Flächeninhalt des Quadrates, das der Ellipse mit den Halbachsen a und b einbeschrieben werden kann (formale Lösung).

Aufgabe 1.4) Ein Meridianschnitt der Erde kann näherungsweise als Ellipse mit $b/a = 299/300$ betrachtet werden. Berechnen Sie die numerische Exzentrizität ϵ dieser Ellipse.

Aufgabe 1.5) Leiten Sie wie die Mittelpunkts Gleichung der Ellipse via axiale Affinität aus der Kreisgleichung her. Zeigen Sie auch, dass für die Gleichung der Tangente im Ellipsenpunkt (x_o/y_o) gilt $\frac{x_o \cdot x}{a^2} + \frac{y_o \cdot y}{b^2} = 1$. Bestimmen Sie die Geradengleichung der Tangente an die Ellipse aus Aufgabe 1.2 im Punkt $P(2 | 2)$.

Aufgabe 1.6) Gegeben sind $A(2 | 0)$ und $B(2 | 0)$. Wir betrachten alle Dreiecke ABC mit $\tan(\alpha) \cdot \tan(\beta) = 0.5$. Auf welcher Kurve liegen die möglichen Eck-Punkte C ?

Aufgabe 1.7*) Beweise die Abstandseigenschaft $PF_1 + PF_2 = 2a$. P ist ein beliebiger Punkt der Ellipse.

Hinweise: Diese Gleichung begründet die bekannte 'Schnurkonstruktion einer Ellipse'. Bei dieser Aufgabe würde sich eventuell der Einsatz eines symbolischen Algebraprogramms lohnen. Von Hand geht es auch, ist aber etwas mühsam.

2.2 Drehimpulserhaltung und Herleitung der Keplerschen Gleichung

Die zeitliche Bewegung eines Planeten auf einer Ellipsenbahn wird durch das berühmte zweite Keplersche Gesetz beschrieben: Die Fahrstrahlen der Planeten überstreichen in gleichen Zeiträumen die gleichen Flächen. Aus heutiger Sicht ist dies nichts anderes als die Drehimpulserhaltung bezogen auf die als ruhend angeschaute Sonne. Der näher bei der Sonne liegende Hauptseitel wird Perihel, der weiter entfernte Aphel genannt.

Die Begründung der Ellipsenbahnen aus dem Gravitationsgesetz wird hier nicht behandelt. Kepler selbst kannte übrigens das Newtonsche Gravitations-

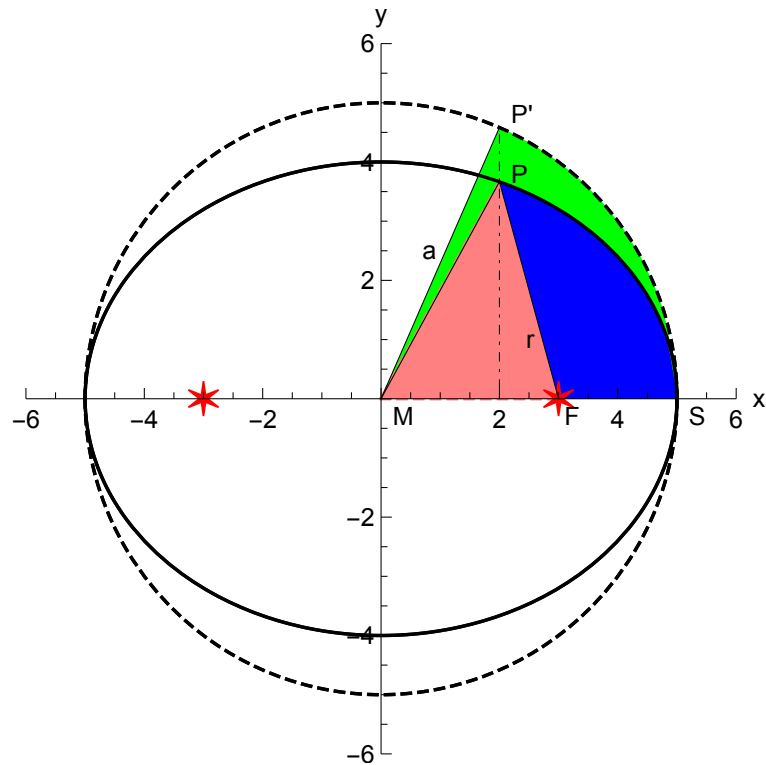


Abbildung 2: Ellipse mit $a = 5$ und $b = 4$, die Bedeutung der verschiedenen angefärbten Flächen wird im Text besprochen. Wichtig ist der Winkel $E = \angle(SMP')$ (exzentrische Anomalie), auch der Polarwinkel $\varphi = \angle(SFP)$ ist oft nützlich.

gesetz noch nicht, sondern hat die Ellipsen aus den (für die damalige Zeit ausgezeichneten) Beobachtungsdaten von Tycho Brahe abgeleitet. Die allgemeine Begründung der Kegelschnittbahnen für das Zweikörperproblem ist auf gymnasialem Niveau (knapp) möglich und erfolgt normalerweise unter Verwendung von Polarkoordinaten.

Wir beschränken uns auf den Fall, in dem die Masse des Zentralkörpers viel grösser ist als die Masse des umlaufenden Planeten, so dass wir die Bewegung der Sonne vernachlässigen können. Genau genommen bewegt sich der Zentralkörper ebenfalls auf einer (kleinen) Ellipse (mit der gleichen numerischen Exzentrizität und der gleichen Umlaufzeit um den gemeinsamen Schwerpunkt).

Aufgabe 2.1) a) Diskutieren Sie die Variation der Planetengeschwindigkeit auf der Ellipsenbahn. Sind die Überlegungen gemäss Flächensatz qualitativ

im Einklang mit der Energie-Erhaltung?

b) Leiten Sie das zweite Keplersche Gesetz aus dem Prinzip der Drehimpulserhaltung her. Tipp: Die genaue Form des Gravitationsgesetzes braucht man hier nicht, es reicht zu wissen, dass eine Zentralkraft vorliegt.

Nun soll die Keplersche Gleichung hergeleitet werden, welche erlaubt die Position des Planeten zu einer vorgegebenen Zeit zu berechnen. Dabei spielt die krummlinig berandete (in Figur 2 blau gefärbt) Fläche eine wichtige Rolle. Wir werden aber hier keine Integrale benutzen, sondern diesen Flächeninhalt im Rahmen der klassischen Geometrie ausdrücken. Der Winkel $E = \angle(SMP')$ heisst nach Kepler 'exzentrische Anomalie' E . Für den Flächeninhalt des entsprechenden Kreissektors gilt $A_{SMP'} = \frac{E}{2\pi} \pi a^2$. (Alle Winkel werden im Bogenmass ausgedrückt.) Via Affinitätsverhältnis b/a und durch Subtraktion des Flächeninhalts des (roten) Dreiecks FMP folgt also für die überstrichene (blaue) Fläche $A_{SFP} = \frac{b}{a} \frac{E}{2} a^2 - \frac{1}{2} ca \sin(E) \frac{b}{a}$. Mit $c = \epsilon a$ und einigen Vereinfachungen ergibt sich somit $A_{SFP} = \frac{ab}{2} [E - \epsilon \sin(E)]$. Andererseits ist die überstrichene Fläche seit dem Periheldurchgang $A_{SFP} = \frac{b}{a} \pi a^2 \frac{t}{T}$. (T ist die Umlaufzeit, wogegen t die Zeit seit dem Periheldurchgang bezeichnet.) Nun erhalten wir also durch Kombination der beiden Ausdrücke für A_{SFP} die Keplersche Gleichung

$$E - \epsilon \sin(E) = 2\pi \frac{t}{T} = M. \quad (2)$$

Der Winkel M wird mittlere Anomalie genannt, da er einem gleichförmig auf einer Kreisbahn umlaufenden Planeten entspricht. Die Keplersche Gleichung muss durch ein numerisches Verfahren (z.B. Bisektion) gelöst werden.

Folgende Gleichungen, welche leicht hergeleitet werden können sind oft nützlich:

$$r = a - c \cos(E), \quad \sin(\varphi) = \frac{b \sin(E)}{r} \quad (3)$$

Bei der Berechnung des Winkels φ muss die Fallunterscheidung $x_P > c$ bzw. $x_P < c$ beachtet werden.

Aufgabe 2.2) Von Merkur ist bekannt: $a = 57.9\text{Gm}$, $\epsilon = 0.206$, $T = 58.65\text{Tage}$. Berechnen M , E , r (Abstand Sonne-Planet) und φ (wahre Anomalie, siehe Skizze) für $t = 10\text{d}$ und 20d nach dem Periheldurchgang. Weiter unten finden Sie einen Vorschlag für ein entsprechendes, kleines Python-Programm.

Aufgabe 2.3) Schauen Sie in der Formelsammlung die Bahndaten des berühmten Halley'schen Kometen nach. Berechnen Sie für einige sinnvolle Zeiten den Abstand zur Sonne. (Periheldurchgang am 5. Februar 1986, den haben Sie

also (wahrscheinlich) verpasst und beim nächsten Mal sind Sie schon nicht mehr ganz jung.)

2.3 Resultate und Lösungshinweise, Teil 1

- 1.1) $a = 10, b = 5, F_{1,2} = (\pm\sqrt{75} \mid 0)$
- 1.2) $\frac{x^2}{32} + \frac{7y^2}{32} = 1, c = \sqrt{\frac{192}{7}}, \epsilon \approx 0.9258$
- 1.3) Flächeninhalt $\frac{4a^2b^2}{a^2+b^2}$ (Winkelhalbierende $y = x$ mit der Ellipse schneiden)
- 1.4) $\epsilon \approx 0.0816$
- 1.5 Herleitung der allgemeinen Tangentengleichung aus Kreistangente via axiale Affinität. Beispiel Tangentengleichung $y = \frac{-1}{7}x + \frac{16}{7}$
- 1.6) Auf der Ellipse $\frac{x^2}{4} + \frac{y^2}{2} = 1$
- 2.2) Merkur

Zeit	E	r	phi	x	y
0.00000	0.00000	45.97260	0.00000	57.90000	0.00000
10.00000	72.64678	54.34252	84.35864	17.26934	54.07932
20.00000	131.58976	65.81732	139.92151	-38.43359	42.37559
30.00000	183.43586	69.80596	182.78815	-57.79593	-3.39559

2.3) Halley (Entfernungen in Astronomischen Einheiten $1.496 \cdot 10^{11}\text{m}$, Zeit in Jahren)

Grosse Halbachse (beliebige Einheiten) 17.8					
Numerische Exzentrizitaet 0.967					
Umlaufzeit (beliebige Einheiten) 75.3					
Zeitschritt 1					
Kleine Halbachse 4.535019431049877					
Zeit	E	r	phi	x	y
0.00000	0.00000	0.58740	0.00000	17.80000	0.00000
1.00000	41.47588	4.90373	142.22902	13.33638	3.00356
2.00000	54.88052	7.89788	151.98678	10.24004	3.70944
3.00000	64.24281	10.32012	156.68574	7.73514	4.08444

2.4 Numerische Lösung der Keplerschen Gleichung mit einem Python-Programm

Wir beschränken hier auf das primitivste aller numerischen Verfahren (Bisektion) zur Lösung von $E - \epsilon \sin(E) = 2\pi \frac{t}{T} = M$. Die linke Seite $E - \epsilon \sin(E)$ steigt im Intervall $[0, 2\pi]$ monoton von 0 bis 2π . Somit besitzt $F[E] = E - \epsilon \sin(E) - 2\pi \frac{t}{T}$ für jede Zeit t mit $0 \leq t \leq T$ nur eine einzige Nullstelle. Wir starten mit dem Intervall $[0, 2\pi]$. In jedem Iterationsschritt wird $F[E_m]$ in der Mitte E_m des aktuellen Intervalls berechnet. Je nach Vorzeichen von $F[E]$ wir mit dem linken oder rechten Teilintervall weiter gefahren. In jedem Schritt wird also die Länge des eingrenzenden Intervalls halbiert. Verständnisfrage: Wie viele Iterationen sind z.B. nötig, um E auf 6 Stellen nach dem Dezimalpunkt genau zu berechnen?

Hier ist ein einfaches Python-Programm, welches die Kepler-Gleichung für eine Anzahl Zeiten löst und auch die Winkel E und φ in Grad, sowie die

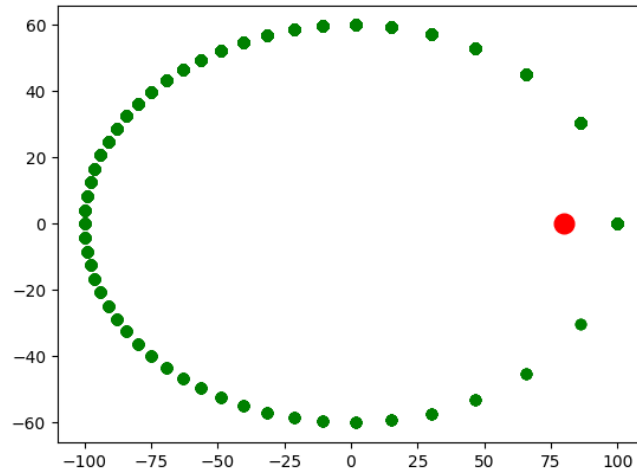


Abbildung 3: Standbild zur Animation der Bewegung eines Himmelskörpers gemäss der Kepler Gleichung.

Koordinaten x und y ausgibt. Die Bisektion erfolgt in einer separaten Prozedur. Auf programmtechnische Raffinessen wir zugunsten der Verständlichkeit verzichtet.

```
# Loesung der Kepler-Gleichung mit dem Bisektionsverfahren
from numpy import *
from pylab import *
from math import *

def kep(eps,ma,pi):
# Loesung der Keplergleichung mit dem Bisektionsverfahren
    xa=0.
    xb=2.*pi
    while (xb-xa)>1.E-8:
        xc=0.5*(xa+xb)
        fxc=xc-eps*sin(xc)-ma
        if fxc<0:
            xa=xc
        else:
            xb=xc
    return xc

print(" Loesung der Kepler-Gleichung")
print(" *****")
istring=input(" Grosse Halbachse (beliebige Einheiten) ")
a=float(istring)
istring=input(" Numerische Exzentrizitaet ")
eps=float(istring)
pi=4.*atan(1.)
u=180./pi
istring=input(" Umlaufzeit (beliebige Einheiten) ")
tum=float(istring)
istring=input(" Zeitschritt ")
dt=float(istring)
nmax=floor(tum/dt)+1
xvek=zeros(nmax,dtype=float)
yvek=zeros(nmax,dtype=float)
c=eps*a
b=sqrt(a**2-c**2)
print(" Kleine Halbachse ",b)
```

```

print(" Zeit      E      r      phi      x      y ")
for n in range(nmax):
    t=float(n)*dt
    ma=2.*pi*t/tum # Mittlere Anomalie
    exan=0.
    r=a-c
    if ma>1.E-3: exan=kep(eps,ma,pi) # Exzentrische Anomalie
    xvek[n]=a*cos(exan)
    yvek[n]=b*sin(exan)
    r=sqrt((xvek[n]-c)**2+yvek[n]**2)
    phi=pi-arccos((c-xvek[n])/r)
    if yvek[n]<0.: phi=2.*pi-phi
    print(f"{t:8.5f}",f"{exan*u:10.5f}",f"{r:10.5f}",f"{phi*u:10.5f}",f"{xvek[n]:10.5f}",f"{yvek[n]:10.5f}")

```

Hier ein Ausgabe-Beispiel:

```

Loesung der Kepler-Gleichung
*****
Grosse Halbachse (beliebige Einheiten) 100.
Numerische Exzentrizitaet 0.8
Umlaufzeit (beliebige Einheiten) 5.
Zeitschritt 0.1
Kleine Halbachse 60.

```

Zeit	E	r	phi	x	y
0.00000	0.00000	20.00000	0.00000	100.00000	0.00000
0.10000	30.38348	30.98724	78.33372	86.26596	30.34710
0.20000	48.98598	47.50050	107.61634	65.62437	45.27294

Nun soll noch eine graphische Darstellung (Animation) der Bewegung zugefügt werden. In einem Koordinatensystem sieht man die Position des Zentralkörpers (im rechten Brennpunkt) und die Position des umlaufenden Körpers jeweils nach jedem Zeitschritt. Die Animation kann entweder auf dem Bildschirm angeschaut werden oder als kepler.gif abgespeichert werden. Zur Betrachtung der gif-Datei reicht ein Browser, ein Beispiel findet man unter <https://medien.martinlehner.ch/kepler.gif>. Für die Animation müssen noch einige Pakete geladen werden., insbesondere matplotlib bietet sehr viele Möglichkeiten. Im Kopf des Programms werden folgende Zeilen zugefügt.

```

import matplotlib.pyplot as plt
import numpy as np
import matplotlib.animation as animation
fig, ax = plt.subplots()

```

Weiter braucht es noch eine Prozedur, welche jeweils die Animation bis zum Zeitschritt mit der Nummer frame aktualisiert:

```

def update(frame):
    # Teilplot bis Position frame
    xvekb = xvek[:frame]
    yvekb = yvek[:frame]
    scat=plt(xvekb,yvekb,'go')
    return (scat)

```

Schliesslich werden die Befehle für die graphische Animation am Schluss des Programms angefügt. Je nach Exzentrizität der Bahn sieht man deutlich wie die Geschwindigkeit des umlaufenden Körpers vom Abstand zum Zentralkörper abhängt.

```

plot(c,0,'ro',markersize=12)
plt.axis([-1.1*a,1.1*a,-1.1*b,1.1*b])
tpause=200 # Wartezeit in ms zwischen Aktualisierungen der Animation
ani = animation.FuncAnimation(fig=fig, func=update, frames=nmax, interval=tpause)
print(" Die Animation wird entweder auf dem Schirm gezeigt oder gespeichert.")
istring=input(" Animation als kepler.gif sichern ? (y/n) ")
if istring=='y':
    # To save the animation using Pillow as a gif
    writer = animation.PillowWriter(fps=5,metadata=dict(artist='Me'),bitrate=1800)
    ani.save('kepler.gif', writer=writer)
if istring=='n':
    plt.show()

```

3 Einfaches (zweidimensionales) Beispiel zum Dreikörperproblem

In folgenden Beispiel geht es um einen als ruhend angesehenen Zentralkörper mit grosser Masse, der von zwei Planeten in der gleichen Ebene umkreist wird. In erster Näherung sind die beiden Ellipsenbahnen unabhängig von einander. Falls sich die beiden umlaufenden Körper im Verlauf der Zeit genügend nahe kommen, kann es zu nicht vernachlässigbaren Bahnstörungen kommen. Unser Sonnensystem gilt als langfristig stabil, allerdings können gelegentlich kleine Planetoiden in Folge der gravitativen Anziehung durch einen schweren Planeten aus dem Sonnensystem katapultiert werden.

Wir packen das Problem mit einem primitiven Python-Programm an. Die Bewegungsgleichungen für die beiden umlaufenden Planeten werden in kartesischen Koordinaten mit einem Runge-Kutta-Verfahren 2. Ordnung gelöst. Hier ein Ausgabe-Beispiel:

```
Ebenes Dreikoerperproblem
*****
1. Planet: Startgeschwindigkeit (km/s) 29.6
Masse in 10**24 kg 6000.0
r01 in Gm 149.6
2. Planet: Startgeschwindigkeit (km/s) 22.2
Masse in 10**24 kg 0.1
r02 in Gm 200.0
t in Tagen      ek1      ep1      ek2      ep2      eges      in 10**30 J
0.000      2628480.186 -5350267.566      24.642      -66.700 -2721829.438
11.574      2630370.744 -5352158.195      24.869      -66.875 -2721829.456
23.148      2635904.922 -5357692.543      25.512      -67.390 -2721829.500
34.722      2644894.902 -5366682.720      26.522      -68.246 -2721829.542
46.296      2657030.082 -5378818.076      27.878      -69.449 -2721829.565
```

Wie man sieht, ist die Gesamtenergie (letzte Spalte) trotz des relativ grossen Zeitschritts von 10000s recht gut erhalten. Beim einfacheren Eulerverfahren müsste der Zeitschritt deutlich kleiner gewählt werden. (Nur jeder 100. Schritt führt zu einer Ausgabezeile.) Der erste Planet entspricht der Erde, ausser dass die Masse tausendfach erhöht wurde, um die Störung auf den Planetoiden drastischer zu gestalten.

In Figur 4 sieht man die beiden Bahnellipsen mit ausgeschalteter Wechselwirkung zwischen den umlaufenden Körpern. Wird nun die gegenseitige Bahnstörung eingeschaltet, so wird der (leichte) Planetoid aus der Bahn katapultiert, während die Wirkung auf den schwereren Planeten vernachlässigbar bleibt (siehe Figur 5).

Hier folgt eine Liste des verwendeten Python-Programms:

```
# Loesung der Bewegungsgleichung nach Runge-Kutta 2. Ordnung
# m2 Masse des Zentralkoerpers, ms,xs,ys Masse und Koordinaten
# des 'anderen' Planeten
def rk2z(x,y,vx,vy,gam,m2,dt,dth,ms,xs,ys,istoe):
    rq=(x**2+y**2)
    r=rq**0.5
    # Halber Vor-Schritt
    a=gam*m2/rq
    wi=math.acos(x/r)
    if y<0: wi=2.*pi-wi
    ax=-a*cos(wi)
```

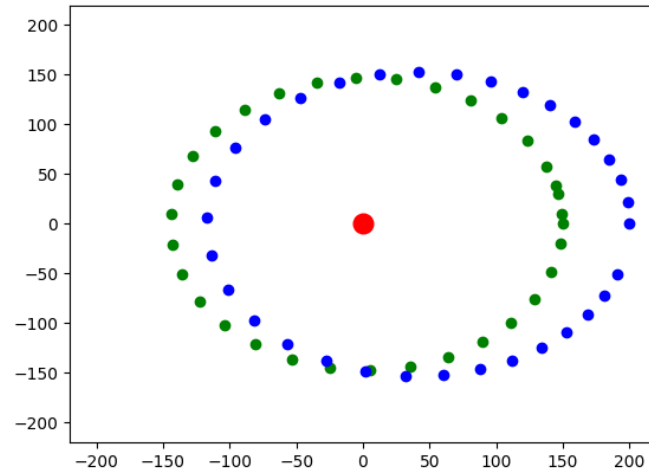


Abbildung 4: Beispiel zum Dreikörperproblem. Die Wechselwirkung zwischen den beiden Planeten ist **weg gelassen**.

```

ay=-a*sin(wi)
# Störung durch anderen Körper
r13q=(x-xs)**2+(y-ys)**2
r13=r13q**0.5
xww=x-xs
yww=y-ys
wis=math.acos(xww/r13)
if yww<0: wis=2.*pi-wis
asg=gam*ms/r13q
asx=-asg*cos(wis)
asy=-asg*sin(wis)
if istoe==1 :
    ax=ax+asx
    ay=ay+asy
# Geschwindigkeit und Ort nach Halbschritt
vx1=vx+ax*dth*0.001
vy1=vy+ay*dth*0.001
x1=x+vx1*dth*0.000001
y1=y+vy1*dth*0.000001
r1q=(x1**2+y1**2)
r1=r1q**0.5
# Ganzer Schritt (entspricht RK 2. Ordnung)
wi=math.acos(x1/r1)
if y<0: wi=2.*pi-wi
a=gam*ms/r1q
ax=-a*cos(wi)
ay=-a*sin(wi)
# Störung durch anderen Körper
r13q=(x1-xs)**2+(y1-ys)**2
r13=r13q**0.5
xww=x1-xs
yww=y1-ys
wis=math.acos(xww/r13)
if yww<0: wis=2.*pi-wis
asg=gam*ms/r13q
asx=-asg*cos(wis)
asy=-asg*sin(wis)
if istoe==1 :
    ax=ax+asx
    ay=ay+asy
# Geschwindigkeit (in 10**-3 m/s) und Ort (in Gm) nach ganzem Schritt
vx=vx+ax*dt*0.001

```

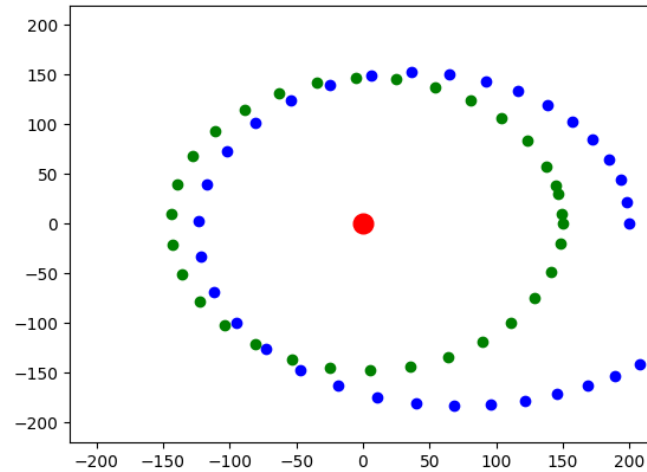


Abbildung 5: Beispiel zum Dreikörperproblem mit Berücksichtigung der Wechselwirkung zwischen den beiden Planeten. Der leichte Planetoid (blau) wird aus der Bahn geworfen.

```

vy=vy+ay*dt*0.001
x=x+vx1*dt*0.000001
y=y+vy1*dt*0.000001
return x,y,vx,vy

print(" Ebenes Dreikörperproblem ")
print(" *****")
v01=29.6
v02=22.2
gam=6.67*10**-5
# Massen in 10**24 kg
m1=6.0*10**3 # Tausenfache Erdmasse
m2=2.0*10**6 # Sonne
m3=0.1 # Planetoid
r01=149.6 # in Gm
r02=200. # in Gm
print(' 1. Planet: Startgeschwindigkeit (km/s) ',v01)
print(' Masse in 10**24 kg ',m1)
print(' r01 in Gm ',r01)
print(' 2. Planet: Startgeschwindigkeit (km/s) ',v02)
print(' Masse in 10**24 kg ',m3)
print(' r02 in Gm ',r02)
# Beschleunigung kommt in m/s**2
pi=4.*atan(1.)
u=180./pi
nmax=3300
nmp=100
nmaxp=nmax//nmp
dt=10000. # in s
dth=0.5*dt
# Ev. Stoerung ausschalten
istoe=1
# istoe=0
xvek1=zeros(nmaxp, dtype=float)
yvek1=zeros(nmaxp, dtype=float)
xvek2=zeros(nmaxp, dtype=float)
yvek2=zeros(nmaxp, dtype=float)
x1=r01
y1=0.
vx1=0.

```

```

vy1=v01
x2=r02
y2=0.
vx2=0.
vy2=v02
print(' t in Tagen      ek1      ep1      ek2      ep2      eges      in 10**30 J  ')

for n in range(nmax):
    t=float(n)*dt
    x1,y1,vx1,vy1=rk2z(x1,y1,vx1,vy1,gam,m2,dt,dth,m3,x2,y2,istoe)
    r1=(x1**2+y1**2)**0.5
    x2,y2,vx2,vy2=rk2z(x2,y2,vx2,vy2,gam,m2,dt,dth,m1,x1,y1,istoe)
    r2=(x2**2+y2**2)**0.5
    #
    if mod(n,nmp)==0:
        np=n//nmp
        vg1=(vx1**2+vy1**2)**0.5
        ek1=0.5*m1*vg1**2      # in 10**30 J
        ep1=-gam*m1*m2/r1*10**3  # in 10**30 J
        eges1=ek1+ep1
        xvek1[np]=x1
        yvek1[np]=y1
        vg2=(vx2**2+vy2**2)**0.5
        ek2=0.5*m3*vg2**2
        ep2=-gam*m2*m3/r2*10**3
        eges2=ek2+ep2
        eges=eges1+eges2
        print(f"{t/86400:8.3f}",f"{ek1:15.3f}",f"{ep1:10.3f}",f"{ek2:10.3f}",f"{ep2:10.3f}",f"{eges:10.3f}")
        xvek2[np]=x2
        yvek2[np]=y2

plot(xvek1,yvek1,'go')
plot(xvek2,yvek2,'bo')
plot(0,0,'ro',markersize=12)
plt.axis([-1.1*r02,1.1*r02,-1.1*r02,1.1*r02])
show()

```

Aufgabe) Untersuchen Sie verschiedene Anordnungen und Startbedingungen, bei denen die Wechselwirkung zwischen den beiden Planeten deutliche Auswirkungen hat. Dankbar wäre auch die Simulation des Systems Sonne, Erde, Erdmond (Bahnneigung der Mondbahn vernachlässigen).

Anspruchsvolle Programmier-Aufgabe) Erweitern Sie das Python-Programm auf den dreidimensionalen Fall.

4 Schlusswort

Das vorliegende Dokument wurde mehrfach im Unterricht eingesetzt und nach meiner Pensionierung leicht überarbeitet. Für Rückmeldungen und Hinweise auf Fehler bin ich dankbar.

Martin Lehner, (Lehrer am Gymnasium Biel-Seeland, Schweiz von 1992 bis 2022), martinlehner@bluewin.ch

5 Verzeichnisse und Quellenangaben

Abbildungsverzeichnis

1	Kepler mit Ellipse, https://upload.wikimedia.org/wikipedia/commons/thumb/d/d4/Johannes_Kepler_1610.jpg/1024px-Johannes_Kepler_1610.jpg	2
2	Ellipse zur Herleitung der Kepler-Gleichung	4
3	Standbild zur Animation der Bewegung auf einer Ellipsenbahn	7
4	Dreikörperproblem mit ausgeschalteter Wechselwirkung	10
5	Dreikörperproblem mit eingeschalteter Wechselwirkung	11