

**Projektarbeit zur grafischen Beschreibung des
Dreikörperproblems**
durch numerisches lösen der Bewegungsgleichungen
Joachim N.
WS 07/08

+ Einführendes Beispiel zum lösen einer DGL n-ter Ordnung

- Dokumentation zum Dreikörperproblem (Sonne, Erde, Mond)

Im Folgenden wird das Vorgehen zur Graphischen Darstellung des Dreikörperproblems (DKP) behandelt:

- Einführung:

Dreikörperproblem

Das Dreikörperproblem der Himmelsmechanik besteht darin, eine Lösung für den Bahnverlauf von drei Körpern unter dem Einfluss ihrer gegenseitigen Anziehung (Gravitation) zu finden. Das Dreikörperproblem galt seit den Entdeckungen von Johannes Kepler und Nikolaus Kopernikus als eines der schwierigsten mathematischen Probleme, mit dem sich im Laufe der Jahrhunderte viele bekannte Mathematiker wie Leonhard Euler, Joseph-Louis Lagrange, Thorvald Nicolai Thiele, George William Hill und Henri Poincaré beschäftigten. Das Zweikörperproblem ist durch die Keplerschen Gesetze streng lösbar. Dagegen sind die Integrale im Fall von $n > 2$ Himmelskörpern keine algebraischen Integrale mehr (Satz von Bruns bzw. Poincaré) und sind nicht mehr mit elementaren Funktionen lösbar.

- Ansatz:

Gravitationsfeld

Gemäß der newtonschen Gravitationstheorie besitzt jeder Körper, der über eine schwere Masse verfügt, ein Gravitationsfeld.

Newtonsches Gravitationsgesetz

Das newtonsche Gravitationsgesetz besagt, dass sich die Gravitationskraft $\mathbf{F}(\mathbf{r})$, mit der sich zwei Massenpunkte der Massen \mathbf{m} und \mathbf{M} anziehen, proportional zu den beiden Massen und umgekehrt proportional zum Quadrat ihres Abstandes \mathbf{r} verhält. Für das Gravitationsfeld gilt

$$> \mathbf{F} = -G \cdot \mathbf{M} \cdot \mathbf{m} / \text{abs}(\mathbf{r})^2 \cdot \mathbf{r} / \text{abs}(\mathbf{r});$$

F die Kraft zwischen den Massenpunkten,
M die Masse des ersten Massenpunktes,

G die Gravitationskonstante
 m die Masse des zweiten Massenpunktes,
 r beschreibt dabei die Richtung,
 |r| den Betrag des Richtungsvektors

$$F = - \frac{G M m r}{|r|^3}$$

> **F:=m*a(t);**
F:=diff(v(t),t)*m;
F:=diff(v(t),t)*m=diff(r(t),t,t)*m;

$$F := m a(t)$$

$$m a(t) = \left(\frac{d}{dt} v(t) \right) m$$

$$F := \left(\frac{d}{dt} v(t) \right) m = \left(\frac{d^2}{dt^2} r(t) \right) m$$

Beim DKP wirkt auf jeden Himmelskörper die Summe der beiden verbleibenden Gravitationspotentiale.

Daraus folgt die Form der Bewegungsgleichung für Sonne, Erde und Mond:

> **DGL_Sonne:=diff(r[Sonne](t),t,t)=-G*M[Erde]*r[es]/abs(r[es])^3-G*M[Mond]*r[ms]/abs(r[ms])^3;**
DGL_Erde:=diff(r[Erde](t),t,t)=-G*M[Sonne]*r[se]/abs(r[se])^3-G*M[Mond]*r[em]/abs(r[em])^3;
DGL_Mond:=diff(r[Mond](t),t,t)=-G*M[Sonne]*r[ms]/abs(r[ms])^3-G*M[Erde]*r[me]/abs(r[me])^3;

$$DGL_Sonne := \frac{d^2}{dt^2} r_{Sonne}(t) = - \frac{G M_{Erde} r_{es}}{|r_{es}|^3} - \frac{G M_{Mond} r_{ms}}{|r_{ms}|^3}$$

$$DGL_Erde := \frac{d^2}{dt^2} r_{Erde}(t) = - \frac{G M_{Sonne} r_{se}}{|r_{se}|^3} - \frac{G M_{Mond} r_{em}}{|r_{em}|^3}$$

$$DGL_Mond := \frac{d^2}{dt^2} r_{Mond}(t) = - \frac{G M_{Sonne} r_{ms}}{|r_{ms}|^3} - \frac{G M_{Erde} r_{me}}{|r_{me}|^3}$$

Dieses Differentialgleichungssystem 2.Ordnung ist analytisch nicht lösbar. Im folgenden Worksheet wird die numerische Lösung mit [dsolve,numeric] demonstriert.

Weiterhin werden die Ergebnisse mit Plot und Animation dargestellt.

Verwendete Parameter:

Bei der Wahl der Parameter wurde stets auf sinnvolle und einfache Beschreibung Wert gelegt.

Die Einheiten sind qualitativ auf wahre Werte bezogen.

Als Einheiten wurden Km, Kg, Tage verwendet.

Alle Parameter, sowie Einheiten sind direkt im Worksheet kommentiert und beschrieben.

Ab "#" steht in der gesamten Zeile Kommentar.

- Variationen im bestehenden Programm:

- Die Programmstruktur ist so angelegt, dass einfache Variationen möglich sind:
- Durch "#" am Zeilenanfang können einzelne Parameter und Strukturen auskommentiert werden.
- Einzelne Parameter sind doppelt beschrieben und bis auf den aktiven auskommentiert.
- Dadurch lassen sich Szenarien einfach durchspielen (Bsp.: Anfangsgeschwindigkeit der Erde [Ady_Erde] > Fluchtgeschwindigkeit).

+ Planetenbewegung in 2D Sonne und Erde

+ Planetenbewegung in 2D Erde und Mond (exzentrisch)

- Planetenbewegung in 2D Sonne, Erde und Mond (vollständig kommentiert)

```
[ > restart;  
[ > Digits;
```

10

- Parameter:

*M(Kg) und G(km³/(Kg*Tage²))*

```
> G:=6.6742*10^(-11)/1000^3*(60*60*24)^2;  
MS:=1.989*10^30; #Sonnenmasse  
ME:=5.974*10^24; #Erdmasse  
MM:=7.352*10^22; #Mondmasse  
Te:=365+(6/24)+(9/60/24)+(9.5/60/60/24); #siderisches  
Jahr: 365d 6h 9m 9.5s  
Tm:=27+(7/24)+(43/60/24)+(12/60/60/24); #siderischer  
Monat: 27d 7h 43m 12s  
R_se:=1.495*10^8; #Abstand Sonne_Erde  
R_em:=384.4*10^3; #Abstand Erde_Mond  
w_se:=2*Pi/Te; #Winkelgeschwindigkeit der Erde  
w_em:=2*Pi/Tm; #Winkelgeschwindigkeit des Mondes  
Hilfsvariablen:  
xs:=R_se/(MS/(ME+MM)+1); #Koordinate der Sonne, wenn der  
Schwerpunkt im Zentrum liegt  
ye:=R_em/((ME/MM)+1); #Abstand des Schwerpunktsystems  
Erde-Mond ~ 4600km  
simplify(%);
```

$G := 0.4982263603 \cdot 10^{-9}$

$$\begin{aligned}
MS &:= 0.1989000000 \cdot 10^{31} \\
ME &:= 0.5974000000 \cdot 10^{25} \\
MM &:= 0.7352000000 \cdot 10^{23} \\
Te &:= 365.2563600 \\
Tm &:= \frac{16393}{600} \\
R_{se} &:= 0.1495000000 \cdot 10^9 \\
R_{em} &:= 384400.0 \\
w_{se} &:= 0.005475606228 \pi \\
w_{em} &:= \frac{1200 \pi}{16393} \\
xs &:= 454.5507748 \\
ye &:= 4673.169828 \\
&4673.169828
\end{aligned}$$

- DGL Sonne, Erde, Mond:

>

Die DGL wurden jeweils in x- und y-Richtung beschrieben.

Die Koordinaten der Massen werden als x- und y-Komponente beschrieben (Bsp.: Perde = vector [xerde , yerde]).

Die Sonne ruht im Zentrum.=> Eine Differentialgleichung für die beschriebene Bahn ist nicht nötig.

```
#DGLxsonne:=0;
```

```
#DGLysonne:=0;
```

DGL der Erde:

```
DGLxerde:=diff(xerde(t),t,t)=
```

```
#0;
```

```
-G*MS/(sqrt(xerde(t)^2+yerde(t)^2))^3
```

```
*xerde(t)
```

```
-G*MM/(sqrt((xmond(t)-xerde(t))^2+(ymond(t)-yerde(t))^2))^3
```

```
*xerde(t);
```

```
DGLyerde:=diff(yerde(t),t,t)=
```

```
#0;
```

```
-G*MS/(sqrt(xerde(t)^2+yerde(t)^2))^3
```

```
*yerde(t)
```

```
-G*MM/(sqrt((xerde(t)-xmond(t))^2+(yerde(t)-ymond(t))^2))^3
```

```
*yerde(t);
```

DGL des Mondes:

```
DGLxmond:=diff(xmond(t),t,t)=  
-G*MS/(sqrt(xmond(t)^2+ymond(t)^2))^3  
*xmond(t)  
-G*ME/(sqrt((xmond(t)-xerde(t))^2+(ymond(t)-yerde(t))^2))^3  
*(xmond(t)-xerde(t));
```

```
DGLymond:=diff(ymond(t),t,t)=  
-G*MS/(sqrt(xmond(t)^2+ymond(t)^2))^3  
*ymond(t)  
-G*ME/(sqrt((xerde(t)-xmond(t))^2+(yerde(t)-ymond(t))^2))^3  
*(ymond(t)-yerde(t));
```

$$DGLxerde := \frac{d^2}{dt^2} xerde(t) = - \frac{0.9909722306 \cdot 10^{21} \cdot xerde(t)}{(xerde(t)^2 + yerde(t)^2)^{(3/2)}} - \frac{0.3662960201 \cdot 10^{14} \cdot xerde(t)}{((xmond(t) - xerde(t))^2 + (ymond(t) - yerde(t))^2)^{(3/2)}}$$

$$DGLyerde := \frac{d^2}{dt^2} yerde(t) = - \frac{0.9909722306 \cdot 10^{21} \cdot yerde(t)}{(xerde(t)^2 + yerde(t)^2)^{(3/2)}} - \frac{0.3662960201 \cdot 10^{14} \cdot yerde(t)}{((-xmond(t) + xerde(t))^2 + (-ymond(t) + yerde(t))^2)^{(3/2)}}$$

$$DGLxmond := \frac{d^2}{dt^2} xmond(t) = - \frac{0.9909722306 \cdot 10^{21} \cdot xmond(t)}{(xmond(t)^2 + ymond(t)^2)^{(3/2)}} - \frac{0.2976404276 \cdot 10^{16} \cdot (xmond(t) - xerde(t))}{((xmond(t) - xerde(t))^2 + (ymond(t) - yerde(t))^2)^{(3/2)}}$$

$$DGLymond := \frac{d^2}{dt^2} ymond(t) = - \frac{0.9909722306 \cdot 10^{21} \cdot ymond(t)}{(xmond(t)^2 + ymond(t)^2)^{(3/2)}} - \frac{0.2976404276 \cdot 10^{16} \cdot (ymond(t) - yerde(t))}{((-xmond(t) + xerde(t))^2 + (-ymond(t) + yerde(t))^2)^{(3/2)}}$$

- Anfangsbedingungen:

Anfangsbedingungen werden jeweils in x- und y-Richtung angegeben.

Zusätzlich werden Anfangsbedingungen für das Schwerpunktsystem Erde-Mond definiert

Der Anfangsabstand in x-Richtung

Die Anfangsgeschwindigkeit in y-Richtung

```
> Ax_Sonne:=xs; #Anfangsbedingung der Sonne in x-Richtung
```

```
Ay_Sonne:=0; #Anfangsbedingung der Sonne in y-Richtung
Adx_Sonne:=0; #Anfangsgeschwindigkeit der Sonne in
x-Richtung
Ady_Sonne:=w_se*xs; #Anfangsgeschwindigkeit der Sonne in
y-Richtung
```

```
Ax_Sonne := 454.5507748
```

```
Ay_Sonne := 0
```

```
Adx_Sonne := 0
```

```
Ady_Sonne := 2.488941053  $\pi$ 
```

Ortsvektor des Schwerpunktsystems Erde-Mond

```
Ax_ErdeMond:=- (R_se-xs);
```

```
Ay_ErdeMond:=0;
```

Geschwindigkeit des Schwerpunktsystems Erde-Mond

```
Adx_ErdeMond:=0;
```

```
Ady_ErdeMond:=-w_se*(R_se-xs);
```

```
Ax_ErdeMond := -0.1494995454  $10^9$ 
```

```
Ay_ErdeMond := 0
```

```
Adx_ErdeMond := 0
```

```
Ady_ErdeMond := -818600.6419  $\pi$ 
```

```
> Ax_Erde:=- (R_se-xs); #Anfangsabstand der Erde in
x-Richtung
```

```
Ay_Erde:=ye; #Anfangsbedingung der Erde bei Halbmond
```

```
Adx_Erde:=Adx_ErdeMond+(-w_em*ye); #Anfangsgeschwindigkeit
der Erde in x-Richtung
```

```
#Adx_Erde:=0;
```

```
Ady_Erde:=Ady_ErdeMond+0; #Anfangsgeschwindigkeit der Erde
in y-Richtung
```

```
#Ady_Erde:=10.76*60*60*24;
```

```
#Ady_Erde:=0; #Anfangsgeschwindigkeit = 0
```

```
Ax_Erde := -0.1494995454  $10^9$ 
```

```
Ay_Erde := 4673.169828
```

```
Adx_Erde := -342.0852677  $\pi$ 
```

```
Ady_Erde := -818600.6419  $\pi$ 
```

```
> Ax_Mond:=- (R_se-xs); #Abstand zur Sonne als Zentralkörper
Ay_Mond:=- (R_em-ye);
```

```
Adx_Mond:=Adx_ErdeMond+(w_em*(R_em-ye));
```

```
#Anfangsgeschwindigkeit des Mondes in x-Richtung
```

```
#Adx_Mond:=0;
```

```
Ady_Mond:=Ady_ErdeMond+0; #Anfangsgeschwindigkeit des
Mondes in y-Richtung
```

```
#Ady_Mond:=2*Pi*384.400/27.3217;#Mittlere
```

```
Bahngeschwindigkeit v=s/t des Mondes [km/tag]
```

```
#Ady_Mond:=15.023*60*60*24; #>Fluchtgeschwindigkeit
```

$$Ax_Mond := -0.1494995454 \cdot 10^9$$

$$Ay_Mond := -379726.8302$$

$$Adx_Mond := 27796.75448 \pi$$

$$Ady_Mond := -818600.6419 \pi$$

– Numerisches Lösen des Differentialgleichungs-Systems:

– Loes_ErdeMond:

Lösungsformel für das DGL-System mit dsolve-numeric:

Anfangsbedingungen werden zum numerischen lösen benötigt.

Maple schreibt die Werte für Richtung und Geschwindigkeit in eine Liste

```
> Loes_ErdeMond:=dsolve( {DGLxerde,DGLyerde,DGLxmond,DGLymond,
xerde(0)=Ax_Erde,
yerde(0)=Ay_Erde,
D(xerde)(0)=Adx_Erde,
D(yerde)(0)=Ady_Erde,
xmond(0)=Ax_Mond,
ymond(0)=Ay_Mond,
D(xmond)(0)=Adx_Mond,
D(ymond)(0)=Ady_Mond},
{xerde(t),yerde(t),xmond(t),ymond(t)},
type=numeric,output=listprocedure);
```

$$Loes_ErdeMond := \left[t = (\text{proc}(t) \dots \text{end proc}), \right.$$

$$xerde(t) = (\text{proc}(t) \dots \text{end proc}), \frac{d}{dt} xerde(t) = (\text{proc}(t) \dots \text{end proc}),$$

$$xmond(t) = (\text{proc}(t) \dots \text{end proc}), \frac{d}{dt} xmond(t) = (\text{proc}(t) \dots \text{end proc}),$$

$$yerde(t) = (\text{proc}(t) \dots \text{end proc}), \frac{d}{dt} yerde(t) = (\text{proc}(t) \dots \text{end proc}),$$

$$ymond(t) = (\text{proc}(t) \dots \text{end proc}), \frac{d}{dt} ymond(t) = (\text{proc}(t) \dots \text{end proc}) \left. \right]$$

Beliebige Punkte der Lösung können angezeigt werden

```
Loes_ErdeMond(0);
```

```
#Loes_ErdeMond(1);
```

```
#Loes_ErdeMond(2);
```

$$\left[t(0) = 0, xerde(t)(0) = -0.14949954540000 \cdot 10^9, \right.$$

$$\left. \left(\frac{d}{dt} xerde(t) \right) (0) = -1074.6925639076, xmond(t)(0) = -0.14949954540000 \cdot 10^9, \right.$$

$$\left. \left(\frac{d}{dt} xmond(t) \right) (0) = 87326.079668007, yerde(t)(0) = 4673.1698280000, \right.$$

$$\left(\frac{d}{dt} \text{yerde}(t) \right) (0) = -0.25717097628169 \cdot 10^7, \text{ymond}(t)(0) = -379726.83020000,$$

$$\left(\frac{d}{dt} \text{ymond}(t) \right) (0) = -0.25717097628169 \cdot 10^7 \Big]$$

```
> with(plots):
```

```
Warning, the previous binding of the name arrow has been removed and it
now has an assigned value
```

Mit odeplot werden die Lösungen für einen bestimmten Zeitraum unter dem jeweiligen Parameter gespeichert.

Zunächst sollen 365 Tage gelistet werden.

```
PxErdeMond:=odeplot(Loes_ErdeMond,[t,xerde(t)+xmond(t)],0..365
,color=red,numpoints=365):
PyErdeMond:=odeplot(Loes_ErdeMond,[t,yerde(t)+ymond(t)],0..365
,color=blue,numpoints=365):
```

```
PxErde:=odeplot(Loes_ErdeMond,[t,xerde(t)],0..365
,color=magenta,numpoints=365):
PyErde:=odeplot(Loes_ErdeMond,[t,yerde(t)],0..365
,color=aquamarine,numpoints=365):
```

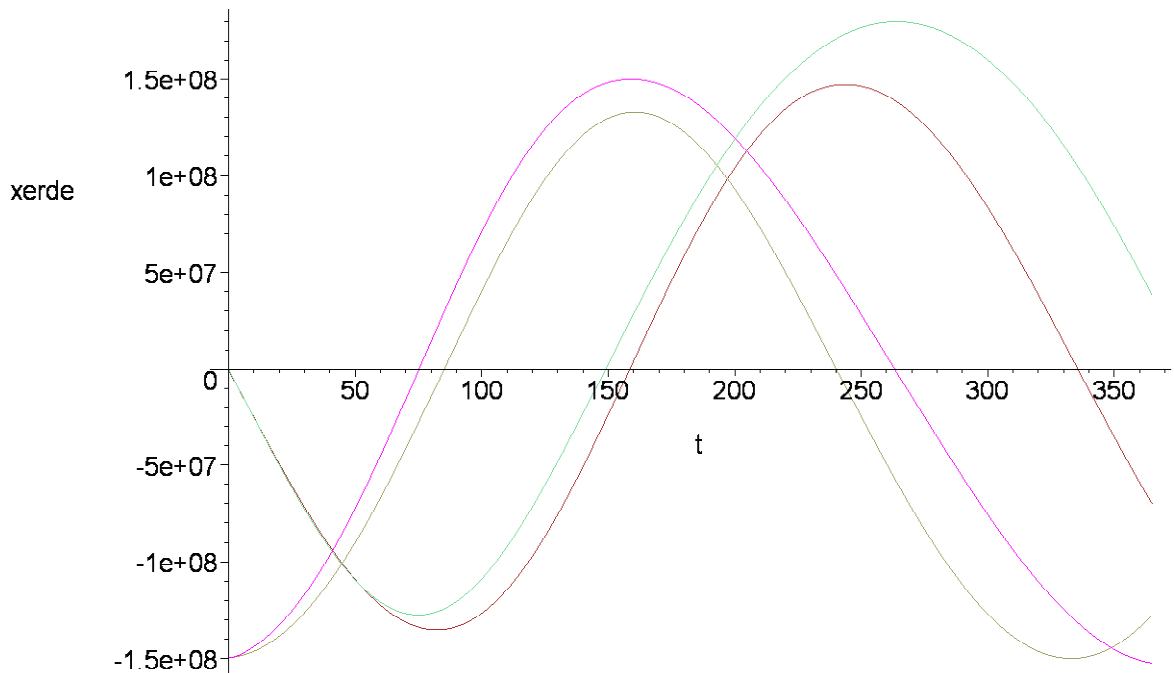
```
PxMond:=odeplot(Loes_ErdeMond,[t,xmond(t)],0..365
,color=khaki,numpoints=365*12):
PyMond:=odeplot(Loes_ErdeMond,[t,ymond(t)],0..365
,color=brown,numpoints=365*12):
```

Zur besseren Kontrolle werden die Ergebnisse in x- und y-Auslenkung über die Zeit dargestellt.

durch Auskommentieren können einzelne Kurven betrachtet ausgeblendet werden.

```
display(
#PxErdeMond,#red
#PyErdeMond,#blue
PxErde,#magenta
PyErde,#aquamarine
PxMond,#khaki
PyMond,#brown
title='Bewegung_in_Einzelkomponenten');
```

Bewegung_in_Einzelkomponenten



[-] Darstellung im y(x)-Diagramm:

Jetzt werden die Koordinaten erneut in eine Liste geschrieben und zur Bahndarstellung im x-y-plot dargestellt.

Um etwaige Bahnverzerrungen, oder nur Ausschnitte darzustellen, kann wiederum die Zeit in Tagen variiert werden.

```
X_Mond:=eval(xmond(t),Loes_ErdeMond);
```

```
Y_Mond:=eval(ymond(t),Loes_ErdeMond);
```

```
X_Erde:=eval(xerde(t),Loes_ErdeMond);
```

```
Y_Erde:=eval(yerde(t),Loes_ErdeMond);
```

```
X_Mond := proc(t) ... end proc
```

```
Y_Mond := proc(t) ... end proc
```

```
X_Erde := proc(t) ... end proc
```

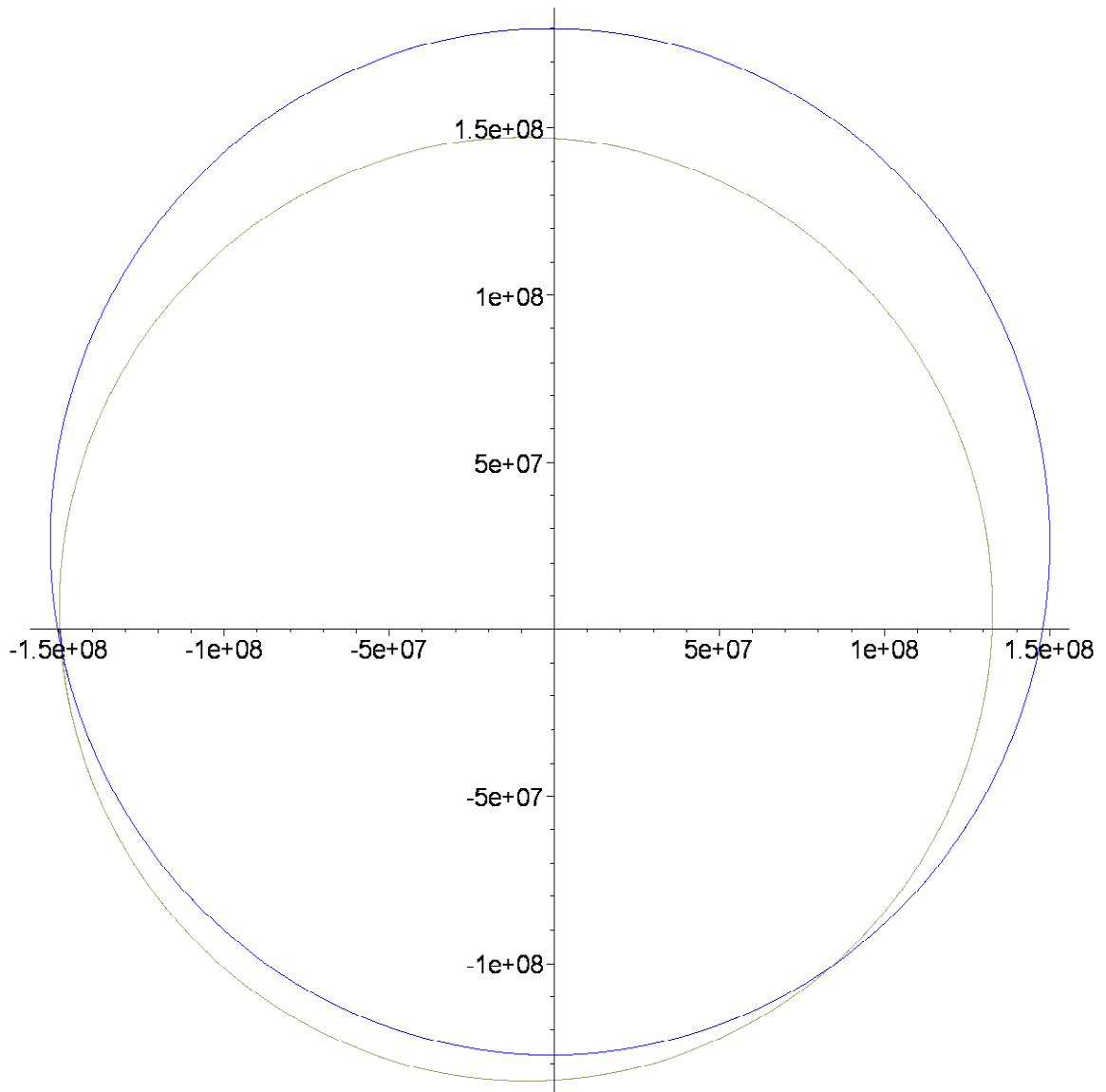
```
Y_Erde := proc(t) ... end proc
```

Auch hier können wieder einzelne Plots auskommentiert werden.

```
> P_Erde:=listplot([seq([X_Erde(t),Y_Erde(t)],t=0..2*365)],color=blue,scaling=constrained):
```

```
P_Mond:=listplot([seq([X_Mond(t),Y_Mond(t)],t=0..2*365)],color=khaki,scaling=constrained):
```

```
display(P_Erde,P_Mond);
```



Animation der Bewegung:

```
[ > with(plottools):
```

```
[ Hier beginnt die Programmstruktur für die Animation der Planetenbewegung.
```

```
[ Für die Sonne wird eine Scheibe im Koordinatenursprung dargestellt.
```

```
[ Die Scheibe für Erde und Mond ist in einer for-Schleife umgesetzt.
```

```
[ Die Größenverhältnisse sind der Anschaulichkeit angepasst.
```

```
[ Variationen:
```

- Anzahl der Frames
- Simulationsdauer (tend)
- Größe und Farbe der Massen
- Auskommentieren einer Masse

```
[ > frames:=200:
```

```
    tend:=365*1:
```

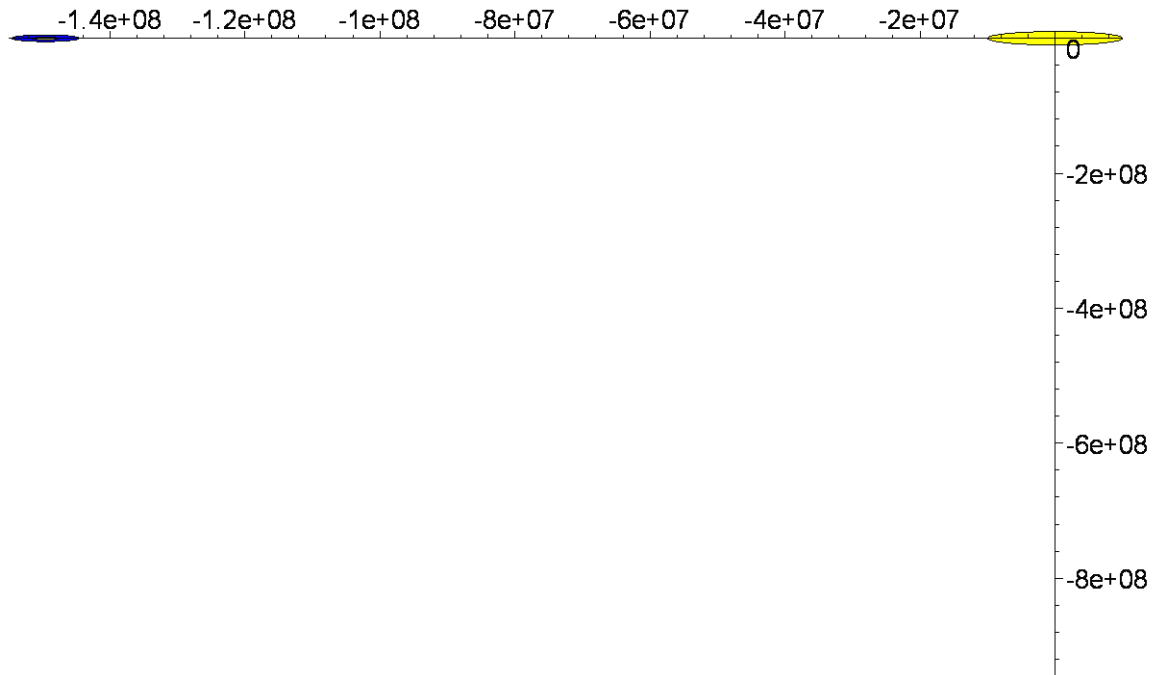
```
    Sonne:=display(disk([0,0],10^7,color=yellow)):
```

```
    for i from 0 by 1 to frames
```

```
    do
```

```
Erde[i]:=display(disk([X_Erde(i*tend/frames),Y_Erde
(i*tend/frames)],0.5*10^7,color=blue)):
Mond[i]:=display(disk([X_Mond(i*tend/frames),Y_Mond
(i*tend/frames)],0.2*10^7,color=khaki)):
Ani[i]:=display({Erde[i],Mond[i],Sonne});
od:
```

```
> display([seq(Ani[i],i=0..frames)],insequence=true);
```



```
[ >
[ >
```