

> **restart:**

Einführung

Am 8. Juni 2004 wird sich um ca. 7.30 Uhr die Venus vor die Sonne schieben (Venustransit). Im 'Physik mit dem Computer'--Kurs (SS 2004) nahmen wir das zum Anlaß um uns die Planetenbahnen und speziell die Eigenschaften des Venustransits durch Computersimulationen zu verdeutlichen. Wir gehen zunächst von den Newtonschen Gleichungen der Gravitationstheorie aus, und lösen die Bewegungsgleichung der Venus und der Erde in zwei und drei Dimensionen.

Das Originalfile file stammt von:
Dr. Michael Komma, Tübingen
<http://www.mikomma.de>

Von mir -Dr. W. Tenten- sind einige Dinge hinzugefügt und einige Dinge etwas modifiziert worden. An den Gleichungssystemen wurde nichts verändert. Der Kollege hat weitere sehr schöne MAPLE Programme auf seiner homepage stehen.

Die verwendeten Ephemeriden stammen aus:
<http://www.solarviews.com/germ/data2.thm>

Planetenbewegung in zwei Dimensionen

Die Bewegungsgleichung eines Massenpunktes (Venus, Erde) im Gravitationsfeld der Sonne ist in zwei Dimensionen (die Bewegung soll sich vollkommen in einer Ebene - der z-Ebene- vollziehen) wie folgt gegeben: wobei M die Masse der Sonne und G die Gravitationskonstante ist.

```
> DGLx := (G, M) -> diff(x(t), t, t) = -G*M/(sqrt(x(t)^2+y(t)^2))^3 * x(t);  
DGLy := (G, M) -> diff(y(t), t, t) = -G*M/(sqrt(x(t)^2+y(t)^2))^3 * y(t);
```

$$DGLx := (G, M) \rightarrow \frac{d^2}{dt^2} x(t) = - \frac{G M x(t)}{\sqrt{x(t)^2 + y(t)^2}^3}$$

$$DGLy := (G, M) \rightarrow \frac{d^2}{dt^2} y(t) = - \frac{G M y(t)}{\sqrt{x(t)^2 + y(t)^2}^3}$$

Wir legen die Parameter M und G fest, wobei wir die Einheiten M(Kg) und G(km^3/(Kg*Tage^2)) wählten:

```
> G := 6.6742*10^(-11)/1000^3*(60*60*24)^2;  
M := 1.989*10^30;
```

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

$$M := 0.1989000000 \cdot 10^{31}$$

Die Anfangswerte sind für die Planetenbewegungen von entscheidender Bedeutung. Wir wählen für den Erd- und Venusabstand von der Sonne und für die jeweiligen Geschwindigkeiten die durchschnittlichen Jahreswerte; die Einheiten sind (Km) bzw. (Km/Tage).

```
> Ax_Erde:=149.5*10^6;  
Ay_Erde:=0;  
Adx_Erde:=0;  
Ady_Erde:=29.76*60*60*24;
```

```
Ax_Venus:=108.21*10^6;  
Ay_Venus:=0;  
Adx_Venus:=0;  
Ady_Venus:=35.02*60*60*24;
```

```
Ax_Merkur:= 57.91*10^6;  
Ay_Merkur:=0;  
Adx_Merkur:=0;  
Ady_Merkur:=47.87*60*60*24;
```

```
Ax_Mars:= 227.94*10^6;  
Ay_Mars:=0;  
Adx_Mars:=0;  
Ady_Mars:=24.13*60*60*24;
```

```
Ax_Jupiter:= 778.33*10^6;  
Ay_Jupiter:=0;  
Adx_Jupiter:=0;  
Ady_Jupiter:=13.06*60*60*24;
```

$Ax_{Erde} := 0.1495000000 \cdot 10^9$

$Ay_{Erde} := 0$

$Adx_{Erde} := 0$

$Ady_{Erde} := 0.257126400 \cdot 10^7$

$Ax_{Venus} := 0.1082100000 \cdot 10^9$

$Ay_{Venus} := 0$

$Adx_{Venus} := 0$

$Ady_{Venus} := 0.302572800 \cdot 10^7$

$Ax_{Merkur} := 0.5791000000 \cdot 10^8$

$Ay_{Merkur} := 0$

$Adx_{Merkur} := 0$

$Ady_Merkur := 0.413596800 \cdot 10^7$

$Ax_Mars := 0.2279400000 \cdot 10^9$

$Ay_Mars := 0$

$Adx_Mars := 0$

$Ady_Mars := 0.208483200 \cdot 10^7$

$Ax_Jupiter := 0.7783300000 \cdot 10^9$

$Ay_Jupiter := 0$

$Adx_Jupiter := 0$

$Ady_Jupiter := 0.112838400 \cdot 10^7$

Numerisches Lösen des Differentialgleichungs-Systems:

```
> Loes_Erde:=dsolve({DGLx(G,M), DGLy(G,M), x(0)=Ax_Erde, y(0)=Ay_Erde, D(x)(0)=Adx_Erde, D(y)(0)=Ady_Erde}, {x(t), y(t)}, type=numeric, output=listprocedure):
Loes_Venus:=dsolve({DGLx(G,M), DGLy(G,M), x(0)=Ax_Venus, y(0)=Ay_Venus, D(x)(0)=Adx_Venus, D(y)(0)=Ady_Venus}, {x(t), y(t)}, type=numeric, output=listprocedure):
Loes_Merkur:=dsolve({DGLx(G,M), DGLy(G,M), x(0)=Ax_Merkur, y(0)=Ay_Merkur, D(x)(0)=Adx_Merkur, D(y)(0)=Ady_Merkur}, {x(t), y(t)}, type=numeric, output=listprocedure):
Loes_Mars:=dsolve({DGLx(G,M), DGLy(G,M), x(0)=Ax_Mars, y(0)=Ay_Mars, D(x)(0)=Adx_Mars, D(y)(0)=Ady_Mars}, {x(t), y(t)}, type=numeric, output=listprocedure):
Loes_Jupiter:=dsolve({DGLx(G,M), DGLy(G,M), x(0)=Ax_Jupiter, y(0)=Ay_Jupiter, D(x)(0)=Adx_Jupiter, D(y)(0)=Ady_Jupiter}, {x(t), y(t)}, type=numeric, output=listprocedure):
```

x(t) und y(t) für die Erd- und Venusbewegung:

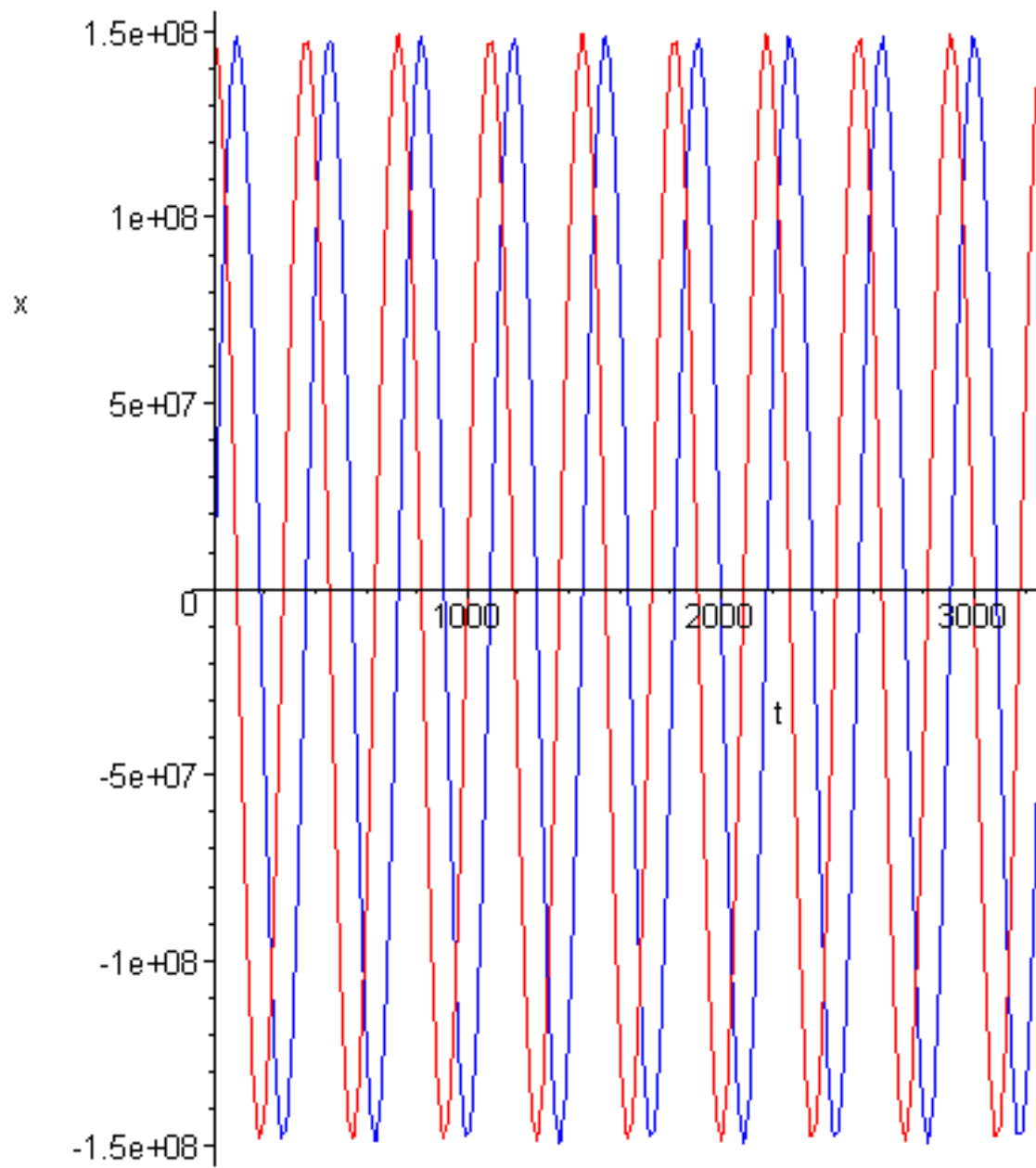
```
> with(plots):
```

Warning, the name changecoords has been redefined

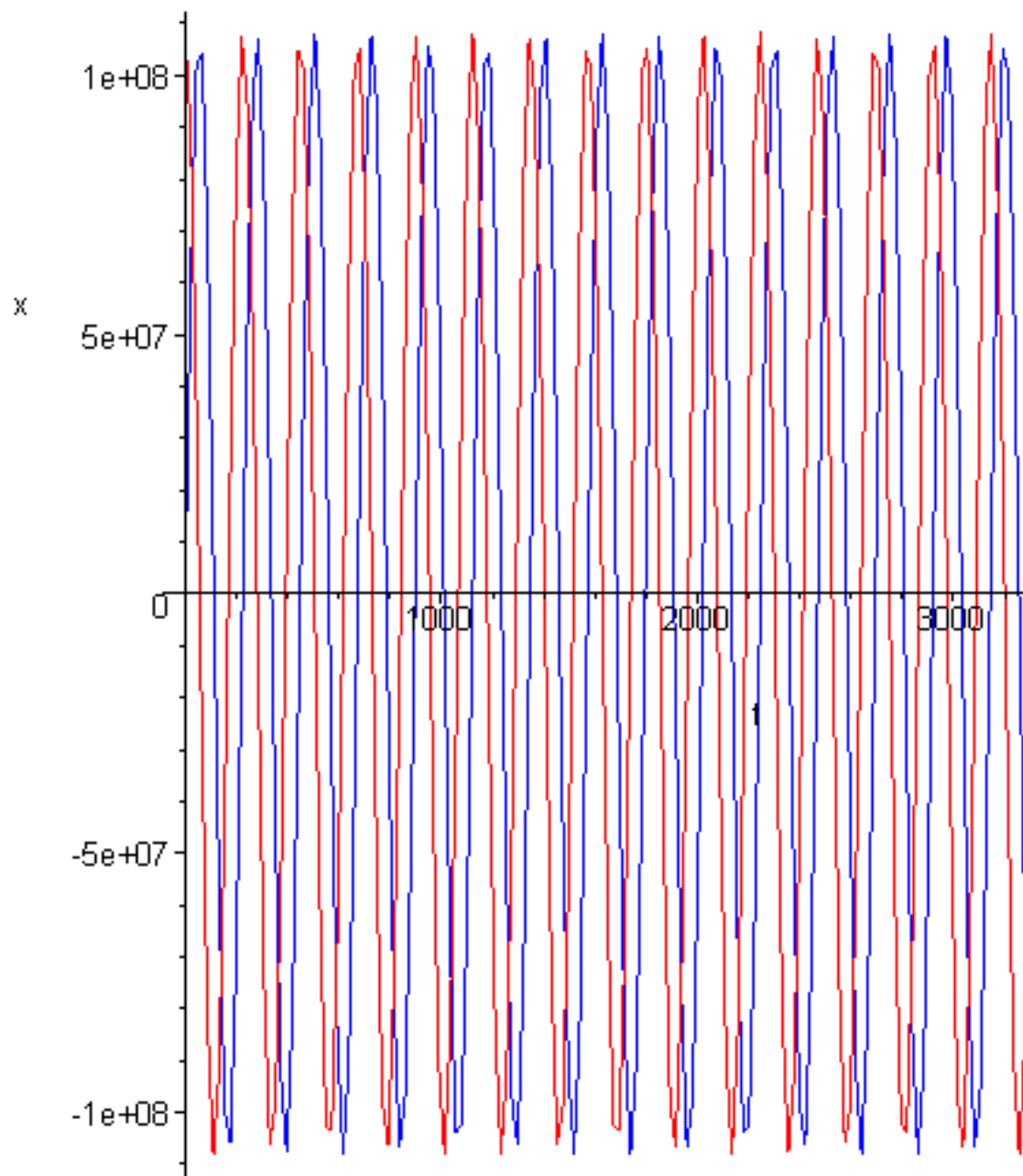
```
> PA1:=odeplot(Loes_Erde, [t, x(t)], 0..365*12, color=red, numpoints=200):
PB1:=odeplot(Loes_Erde, [t, y(t)], 0..365*12, color=blue, numpoints=200):
display(PA1, PB1, title='Erdebewegung');
PA2:=odeplot(Loes_Venus, [t, x(t)], 0..365*12, color=red, numpoints=200):
PB2:=odeplot(Loes_Venus, [t, y(t)], 0..365*12, color=blue, numpoints=200):
display(PA2, PB2, title='Venusbewegung');
PA3:=odeplot(Loes_Merkur, [t, x(t)], 0..365*12, color=red, numpoints=200):
PB3:=odeplot(Loes_Merkur, [t, y(t)], 0..365*
```

```
12,color=blue,numpoints=200):  
display(PA3,PB3,title='Merkurbewegung');  
PA4:=odeplot(Loes_Mars,[t,x(t)],0..365*  
12,color=red,numpoints=200):  
PB4:=odeplot(Loes_Mars,[t,y(t)],0..365*  
12,color=blue,numpoints=200):  
display(PA4,PB4,title='Marsbewegung');  
PA5:=odeplot(Loes_Jupiter,[t,x(t)],0..365*  
12,color=red,numpoints=200):  
PB5:=odeplot(Loes_Jupiter,[t,y(t)],0..365*  
12,color=blue,numpoints=200):  
display(PA5,PB5,title='Jupiterbewegung');
```

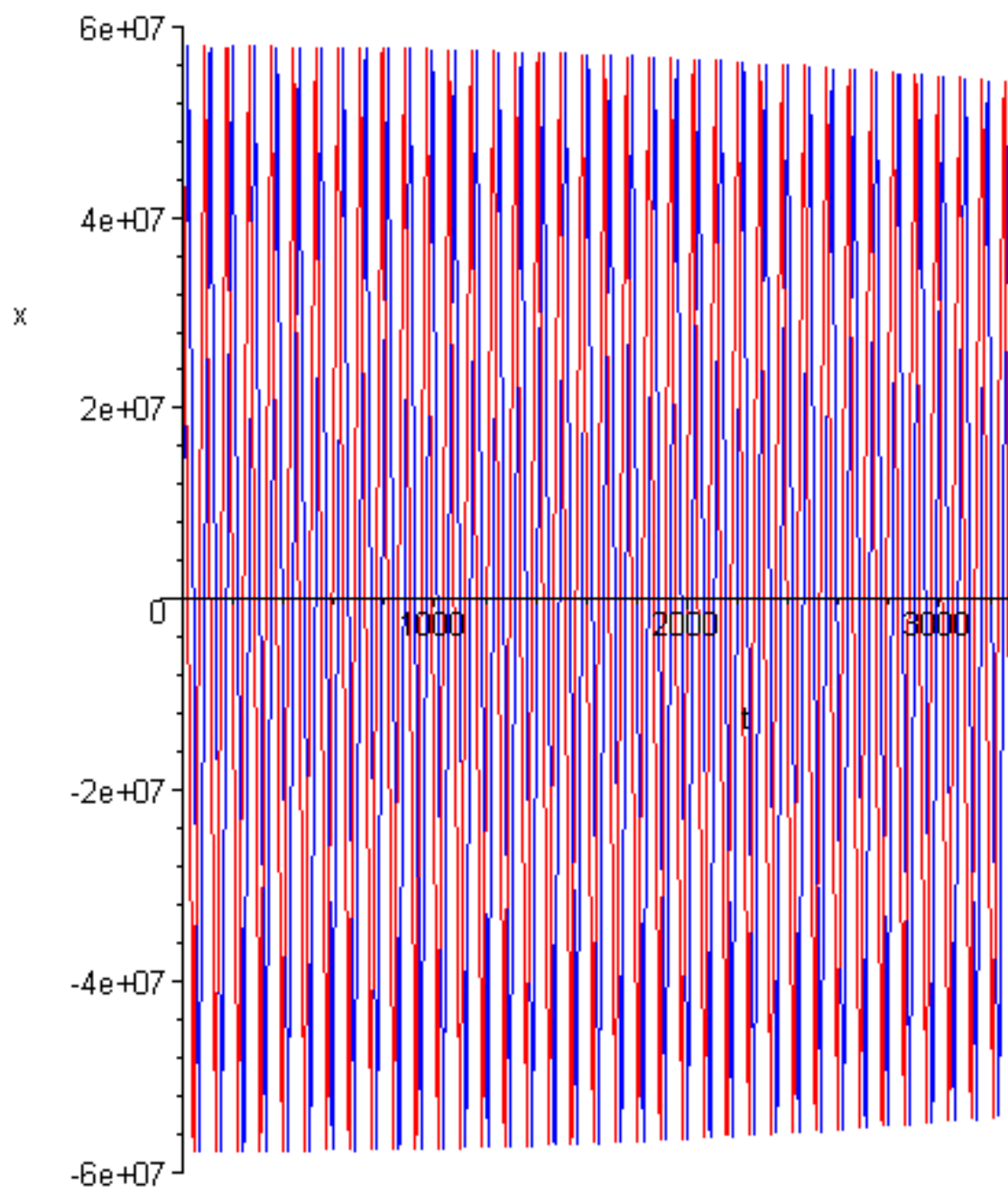
Erdebewegung



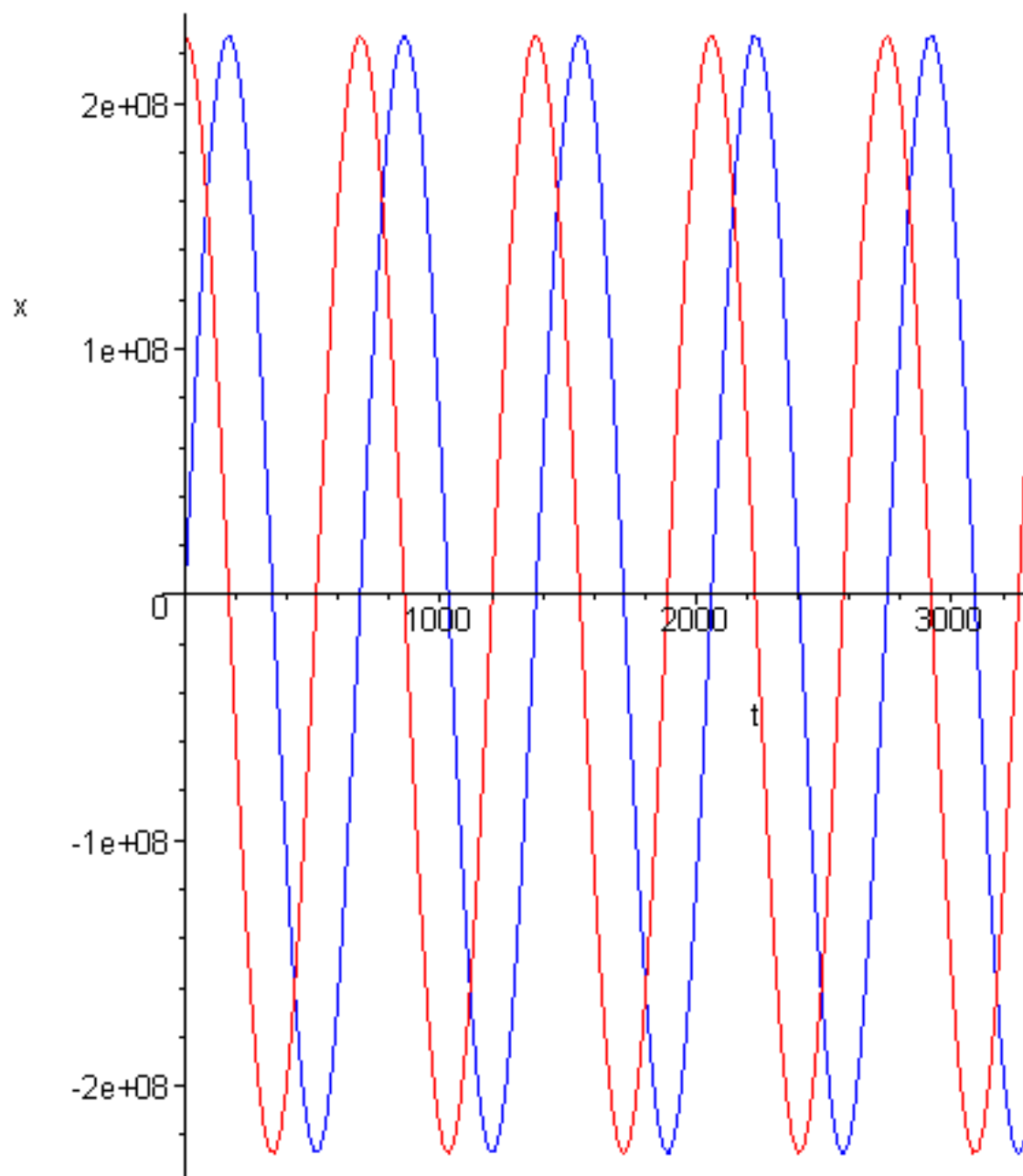
Venusbewegung



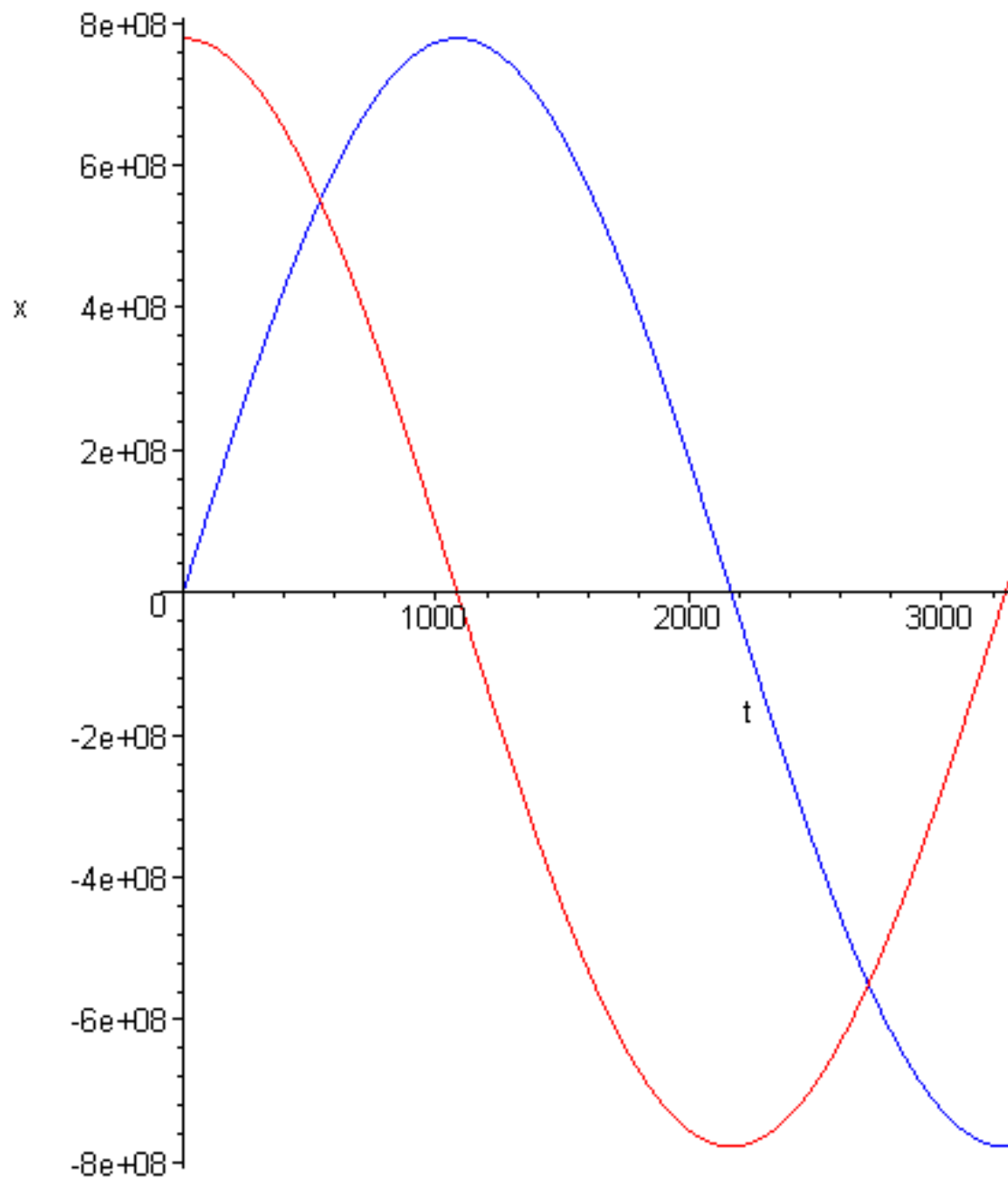
Merkurbewegung



Marsbewegung



Jupiterbewegung



Um die Bewegungen der Planeten im $y(x)$ -Diagramm zu veranschaulichen, definieren wir zunächst die Lösungsfunktionen und stellen die Bewegung dann dar.

```
> X_Erde:=eval(x(t), Loes_Erde):  
Y_Erde:=eval(y(t), Loes_Erde):  
X_Venus:=eval(x(t), Loes_Venus):  
Y_Venus:=eval(y(t), Loes_Venus):  
X_Merkur:=eval(x(t), Loes_Merkur):  
Y_Merkur:=eval(y(t), Loes_Merkur):  
X_Mars:=eval(x(t), Loes_Mars):  
Y_Mars:=eval(y(t), Loes_Mars):
```

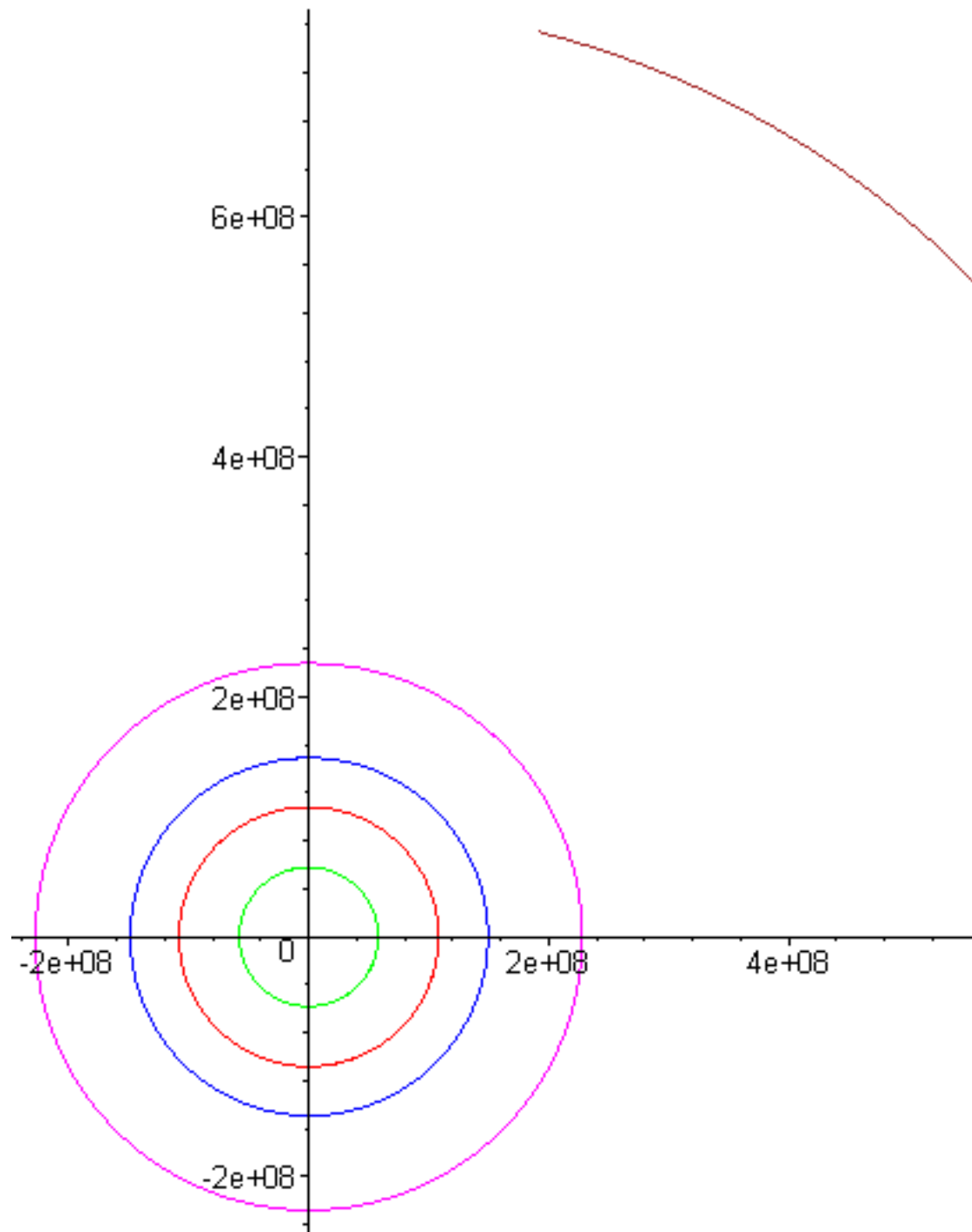
```

X_Jupiter:=eval(x(t),Loes_Jupiter):
Y_Jupiter:=eval(y(t),Loes_Jupiter):

> P_Erde:=listplot([seq([X_Erde(t),Y_Erde(t)],t=0..2.5*
365)],color=blue,scaling=constrained):
P_Venus:=listplot([seq([X_Venus(t),Y_Venus(t)],t=0..2.5*
365)],color=red,scaling=constrained): P_Merkur:=listplot
([seq([X_Merkur(t),Y_Merkur(t)],t=0..2.5*
365)],color=green,scaling=constrained): P_Mars:=listplot
([seq([X_Mars(t),Y_Mars(t)],t=0..2.5*
365)],color=magenta,scaling=constrained): P_Jupiter:
=listplot([seq([X_Jupiter(t),Y_Jupiter(t)],t=0..2.5*
365)],color=brown,scaling=constrained):

display(P_Erde,P_Venus,P_Merkur,P_Mars,P_Jupiter);

```



Im folgenden werden die Planetenbewegungen animiert:

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

Warning, the name arrow has been redefined

```
> frames:=70:
```

```
tend:=365*10:
```

```
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)):
```

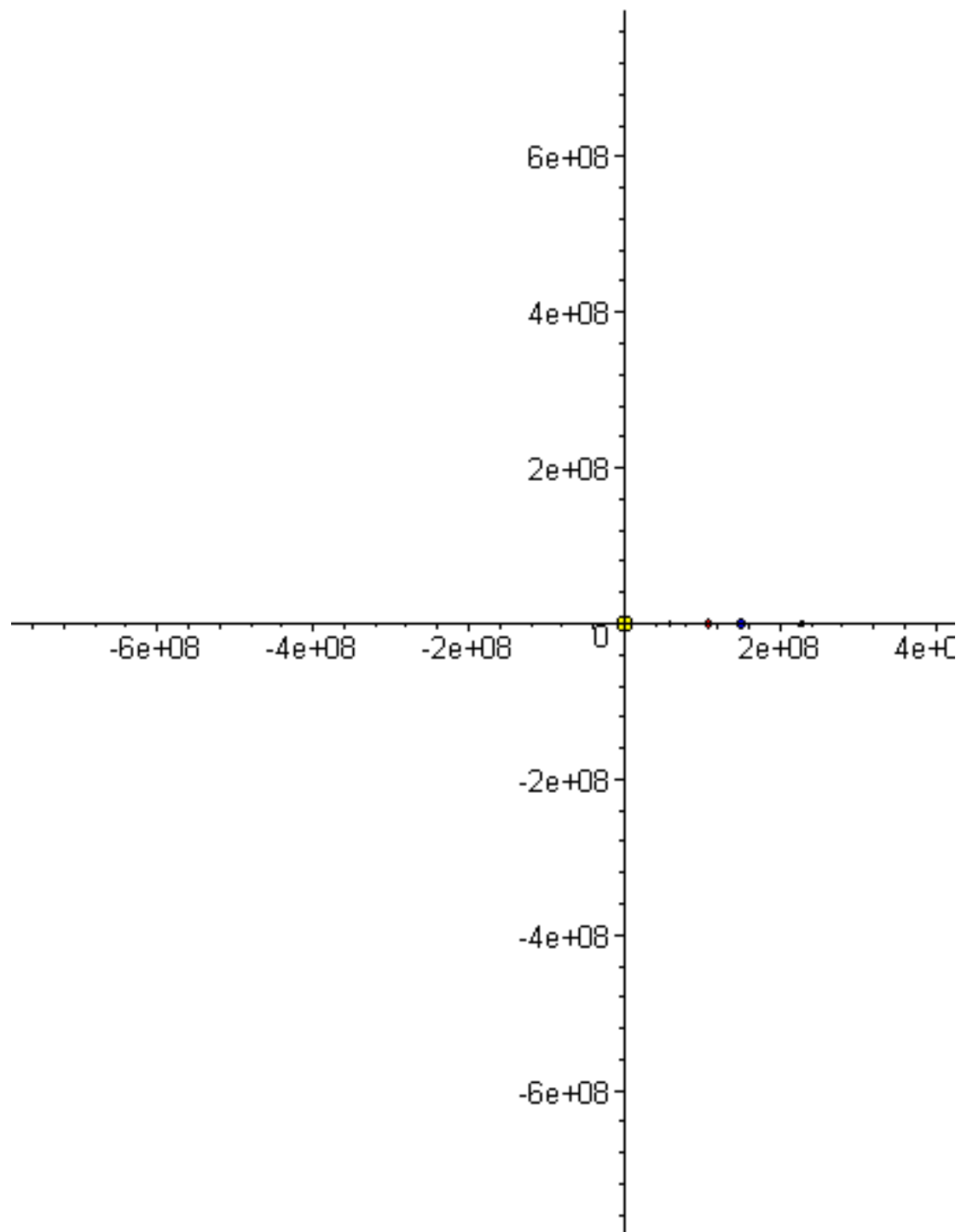
```
Venus[i]:=display(disk([X_Venus(i*tend/frames),Y_Venus
```

```

(i*tend/frames)], 0.45*10^7, color=red)):
Merkur[i]:=display(disk([X_Merkur(i*tend/frames), Y_Merkur
(i*tend/frames)], 0.25*10^7, color=green)):
Mars[i]:=display(disk([X_Mars(i*tend/frames), Y_Mars
(i*tend/frames)], 0.40*10^7, color=magenta)):
Jupiter[i]:=display(disk([X_Jupiter
(i*tend/frames), Y_Jupiter(i*tend/frames)], 1.0*10^
7, color=brown)):

Ani[i]:=display({Erde[i], Merkur[i], Venus[i], Mars
[i], Jupiter[i], Sonne});
od:
> display([seq(Ani[i], i=0..frames)], insequence=true);

```



```
>
> with(linalg):
V:=(x,y)->-G*M/sqrt(x^2+y^2);
FV:=grad(-V(x,y),vector([x,y]));
```

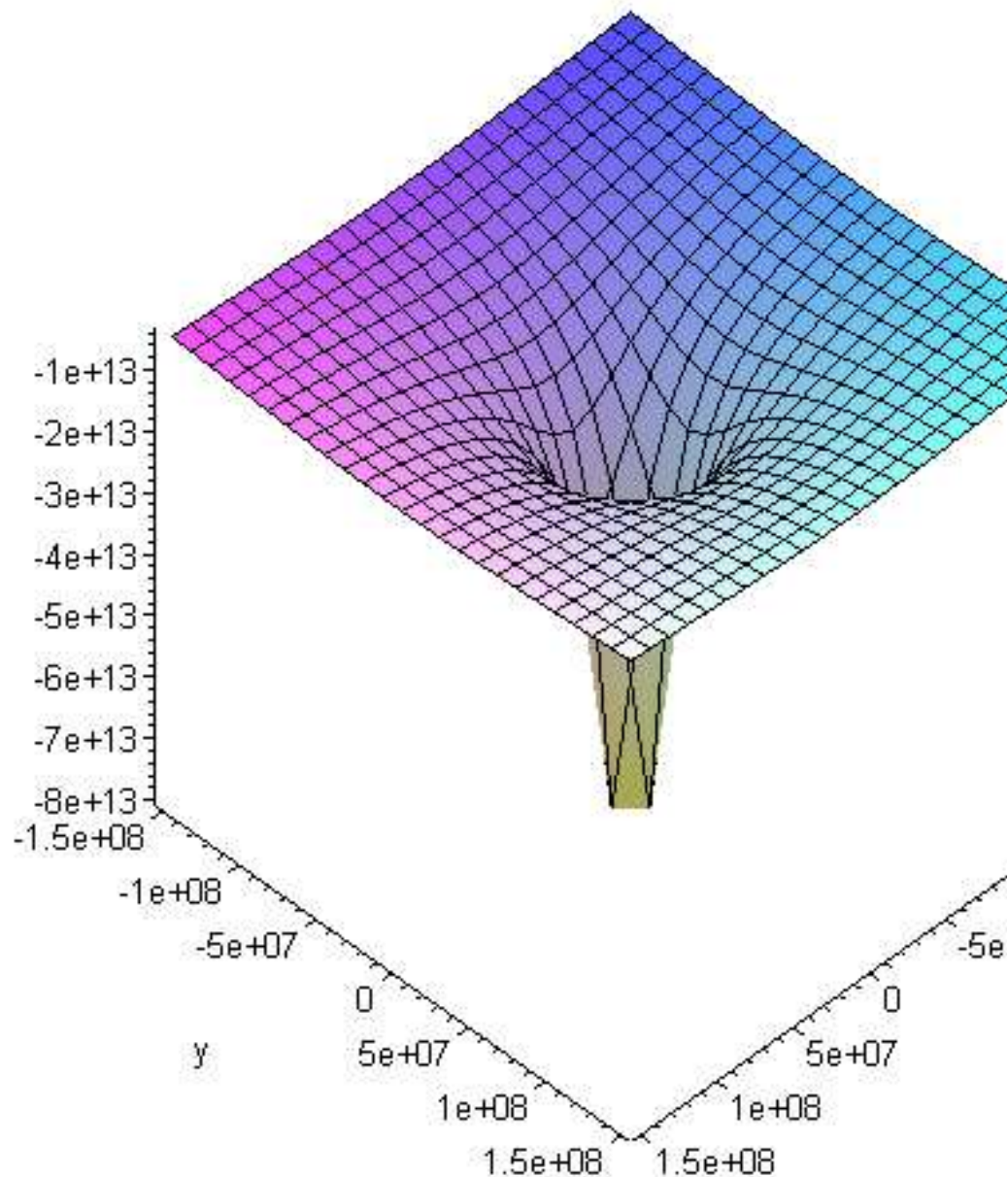
Warning, the protected names norm and trace have been redefined and unprotected

$$V := (x, y) \rightarrow -\frac{G M}{\sqrt{x^2 + y^2}}$$

$$FV := \left[-\frac{0.9909722306 \cdot 10^{21} x}{(x^2 + y^2)^{(3/2)}}, -\frac{0.9909722306 \cdot 10^{21} y}{(x^2 + y^2)^{(3/2)}} \right]$$

```
> plot3d(V(x,y), x=-150*10^6..150*10^6, y=-150*10^6..150*10^
6, axes=framed, title='Gravitationspotential');
Pa:=contourplot(V(x,y), x=-150*10^6..150*10^6, y=-150*10^
6..150*10^6, contours=
40, filled=true, title='Hoehenlinien'):
Pb:=fieldplot(FV, x=-3..3, y=-3..3, arrows=THICK):
```

Gravitationspotential



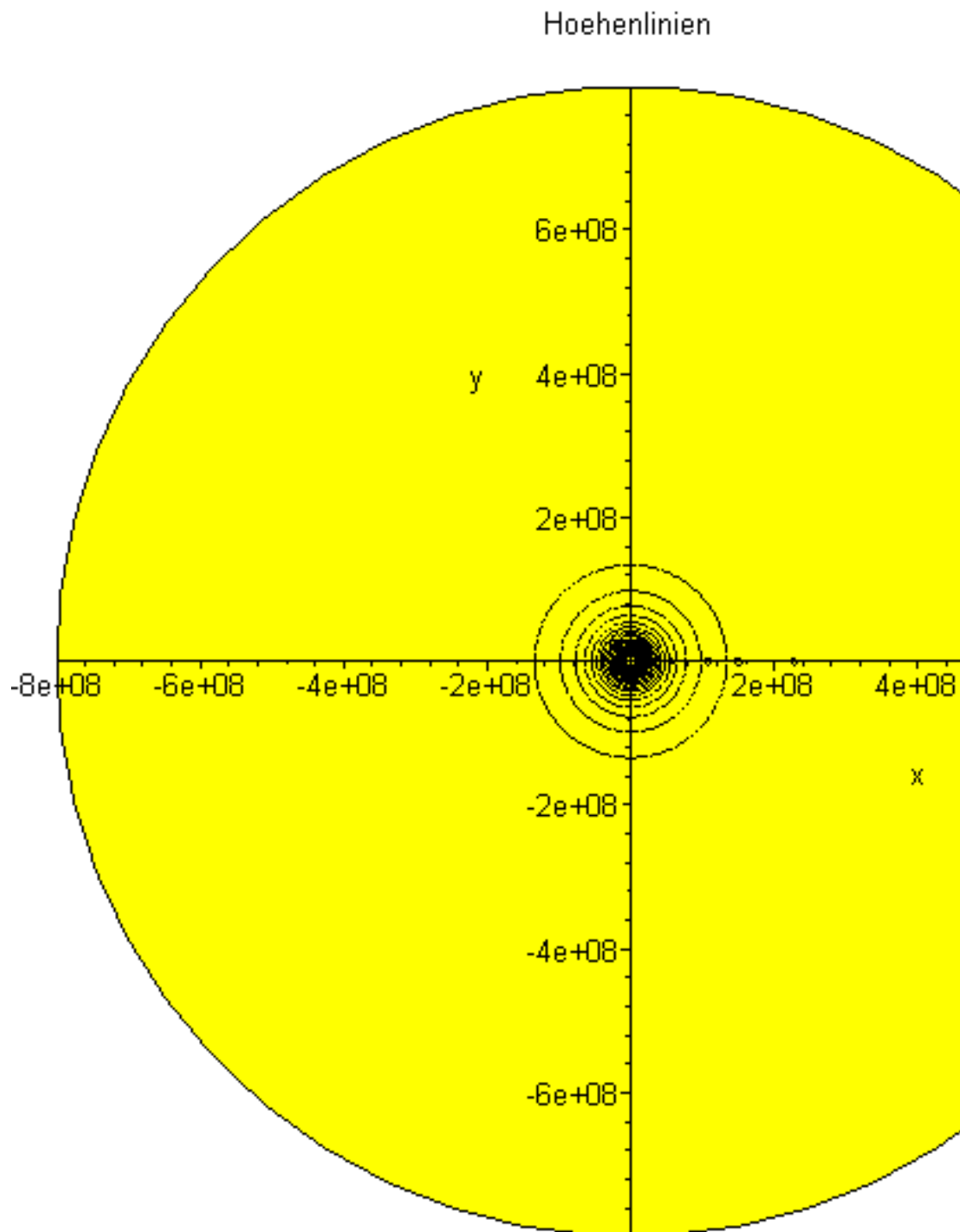
```
> frames:=70:
tend:=365*10:
Sonne:=display(disk([0,0],8*10^8,color=yellow)):
for i from 0 by 1 to frames do
Erde[i]:=display(disk([X_Erde(i*tend/frames),Y_Erde
(i*tend/frames)],0.5*10^7,color=blue)):
Venus[i]:=display(disk([X_Venus(i*tend/frames),Y_Venus
(i*tend/frames)],0.45*10^7,color=red)):
Merkur[i]:=display(disk([X_Merkur(i*tend/frames),Y_Merkur
(i*tend/frames)],0.25*10^7,color=green)):
endfor
```

```

Mars[i]:=display(disk([X_Mars(i*tend/frames),Y_Mars
(i*tend/frames)],0.40*10^7,color=magenta)):
Jupiter[i]:=display(disk([X_Jupiter
(i*tend/frames),Y_Jupiter(i*tend/frames)],1.0*10^
7,color=brown)):
Ani[i]:=display({Erde[i],Venus[i],Merkur[i],Mars
[i],Jupiter[i],Sonne,Pa,Pb});
od:

```

Die folgende animierte Grafik zeigt die Planeten i ihrem Abstand zur Sonne
 > **display([seq(Ani[i],i=0..frames)],insequence=true);**



Da die Exzentrizität der Planetenbewegungen sehr gering ist, verläuft die Bewegung nahe der entsprechenden Höhenlinie des Gravitationspotentials.

Planetenbewegung in drei Dimensionen

Nun soll die Bewegung in drei Dimensionen gelöst werden.

```
> DGLx:=(G,M)->diff(x(t),t,t)=-G*M/(sqrt(x(t)^2+y(t)^2+z(t)^2))^3*x(t);
DGLy:=(G,M)->diff(y(t),t,t)=-G*M/(sqrt(x(t)^2+y(t)^2+z(t)^2))^3*y(t);
DGLz:=(G,M)->diff(z(t),t,t)=-G*M/(sqrt(x(t)^2+y(t)^2+z(t)^2))^3*z(t);
```

$^2))^3 \cdot z(t);$

$$DGLx := (G, M) \rightarrow \frac{d^2}{dt^2} x(t) = - \frac{G M x(t)}{\sqrt{x(t)^2 + y(t)^2 + z(t)^2}^3}$$

$$DGLy := (G, M) \rightarrow \frac{d^2}{dt^2} y(t) = - \frac{G M y(t)}{\sqrt{x(t)^2 + y(t)^2 + z(t)^2}^3}$$

$$DGLz := (G, M) \rightarrow \frac{d^2}{dt^2} z(t) = - \frac{G M z(t)}{\sqrt{x(t)^2 + y(t)^2 + z(t)^2}^3}$$

Wir legen die Parameter M und G fest, wobei wir die Einheiten M(Kg) und G(km³/Kg*Tage²) wählten:

> G:=6.6732*10^{^(-11)}/1000^{^3}*(60*60*24)^{^2};
M:=1.989*10^{^30};

$$G := 0.4981517107 \cdot 10^{-9}$$

$$M := 0.1989000000 \cdot 10^{31}$$

Die Anfangswerte sind für die Planetenbewegungen von entscheidender Bedeutung. Wir wählten für den Erd- und Venusabstand von der Sonne und für die jeweiligen Geschwindigkeiten die durchschnittlichen Jahreswerte; die Einheiten sind (Km) bzw. (Km/Tage). Wir wählten die Bahnbewegung der Erde in der z-Ebene. Die Ebene der Bewegung der Venus ist um 3.24 Grad gegen die Bewegungsebene der Erde geneigt; dies berücksichtigten wir in der Anfangsgeschwindigkeit der Venus.

> Ax_Erde:=149.5*10^{^6};
Ay_Erde:=0;
Az_Erde:=0;
Adx_Erde:=0;
Ady_Erde:=29.76*60*60*24;
Adz_Erde:=0;

Ax_Venus:=108.21*10^{^6};
Ay_Venus:=0;
Az_Venus:=0;
Adx_Venus:=0;
Ady_Venus:=35.02*60*60*24*sin((90-3.394)/360*2*Pi);
Adz_Venus:=-35.02*60*60*24*sin(3.394/360*2*Pi);

Ax_Merkur:= 57.91*10^{^6};
Ay_Merkur:=0;
Az_Merkur:=0;
Adx_Merkur:=0;
Ady_Merkur:=47.87*60*60*24*sin((90-7.004)/360*2*Pi);

```

Adz_Merkur:=-47.87*60*60*24*sin(7.004/360*2*Pi);

Ax_Mars:= 227.94*10^6;
Ay_Mars:=0;
Az_Mars:=0;
Adx_Mars:=0;
Ady_Mars:=24.13*60*60*24*sin((90-1.850)/360*2*Pi);
Adz_Mars:=-24.13*60*60*24*sin(1.850/360*2*Pi);

Ax_Jupiter:= 778.33*10^6;
Ay_Jupiter:=0;
Az_Jupiter:=0;
Adx_Jupiter:=0;
Ady_Jupiter:=13.06*60*60*24*sin((90-1.308)/360*2*Pi);
Adz_Jupiter:=-13.06*60*60*24*sin(1.308/360*2*Pi);

```

$$Ax_{Erde} := 0.1495000000 \cdot 10^9$$

$$Ay_{Erde} := 0$$

$$Az_{Erde} := 0$$

$$Adx_{Erde} := 0$$

$$Ady_{Erde} := 0.257126400 \cdot 10^7$$

$$Adz_{Erde} := 0$$

$$Ax_{Venus} := 0.1082100000 \cdot 10^9$$

$$Ay_{Venus} := 0$$

$$Az_{Venus} := 0$$

$$Adx_{Venus} := 0$$

$$Ady_{Venus} := 0.302572800 \cdot 10^7 \sin(0.4811444444 \pi)$$

$$Adz_{Venus} := -0.302572800 \cdot 10^7 \sin(0.01885555556 \pi)$$

$$Ax_{Merkur} := 0.5791000000 \cdot 10^8$$

$$Ay_{Merkur} := 0$$

$$Az_{Merkur} := 0$$

$$Adx_{Merkur} := 0$$

$$Ady_{Merkur} := 0.413596800 \cdot 10^7 \sin(0.4610888888 \pi)$$

$Adz_Merkur := -0.413596800 \cdot 10^7 \sin(0.03891111112 \pi)$

$Ax_Mars := 0.2279400000 \cdot 10^9$

$Ay_Mars := 0$

$Az_Mars := 0$

$Adx_Mars := 0$

$Ady_Mars := 0.208483200 \cdot 10^7 \sin(0.4897222222 \pi)$

$Adz_Mars := -0.208483200 \cdot 10^7 \sin(0.01027777778 \pi)$

$Ax_Jupiter := 0.7783300000 \cdot 10^9$

$Ay_Jupiter := 0$

$Az_Jupiter := 0$

$Adx_Jupiter := 0$

$Ady_Jupiter := 0.112838400 \cdot 10^7 \sin(0.4927333334 \pi)$

$Adz_Jupiter := -0.112838400 \cdot 10^7 \sin(0.007266666666 \pi)$

Numerisches Lösen des Differentialgleichungs-Systems:

```
> Loes_Erde:=dsolve({DGLx(G,M),DGLy(G,M),DGLz(G,M),x(0)
=Ax_Erde,y(0)=Ay_Erde,z(0)=Az_Erde,D(x)(0)=Adx_Erde,D(y)
(0)=Ady_Erde,D(z)(0)=Adz_Erde},{x(t),y(t),z
(t)},type=numeric,output=listprocedure):
Loes_Venus:=dsolve({DGLx(G,M),DGLy(G,M),DGLz(G,M),x(0)
=Ax_Venus,y(0)=Ay_Venus,z(0)=Az_Venus,D(x)(0)=Adx_Venus,D
(y)(0)=Ady_Venus,D(z)(0)=Adz_Venus},{x(t),y(t),z
(t)},type=numeric,output=listprocedure):
Loes_Merkur:=dsolve({DGLx(G,M),DGLy(G,M),DGLz(G,M),x(0)
=Ax_Merkur,y(0)=Ay_Merkur,z(0)=Az_Merkur,D(x)(0)
=Adx_Merkur,D(y)(0)=Ady_Merkur,D(z)(0)=Adz_Merkur},{x
(t),y(t),z(t)},type=numeric,output=listprocedure):
Loes_Mars:=dsolve({DGLx(G,M),DGLy(G,M),DGLz(G,M),x(0)
=Ax_Mars,y(0)=Ay_Mars,z(0)=Az_Mars,D(x)(0)=Adx_Mars,D(y)
(0)=Ady_Mars,D(z)(0)=Adz_Mars},{x(t),y(t),z
(t)},type=numeric,output=listprocedure):
Loes_Jupiter:=dsolve({DGLx(G,M),DGLy(G,M),DGLz(G,M),x(0)
=Ax_Jupiter,y(0)=Ay_Jupiter,z(0)=Az_Jupiter,D(x)(0)
=Adx_Jupiter,D(y)(0)=Ady_Jupiter,D(z)(0)=Adz_Jupiter},{x
(t),y(t),z(t)},type=numeric,output=listprocedure):
```

x(t), y(t) und z(t) für die Planetenbewegung:

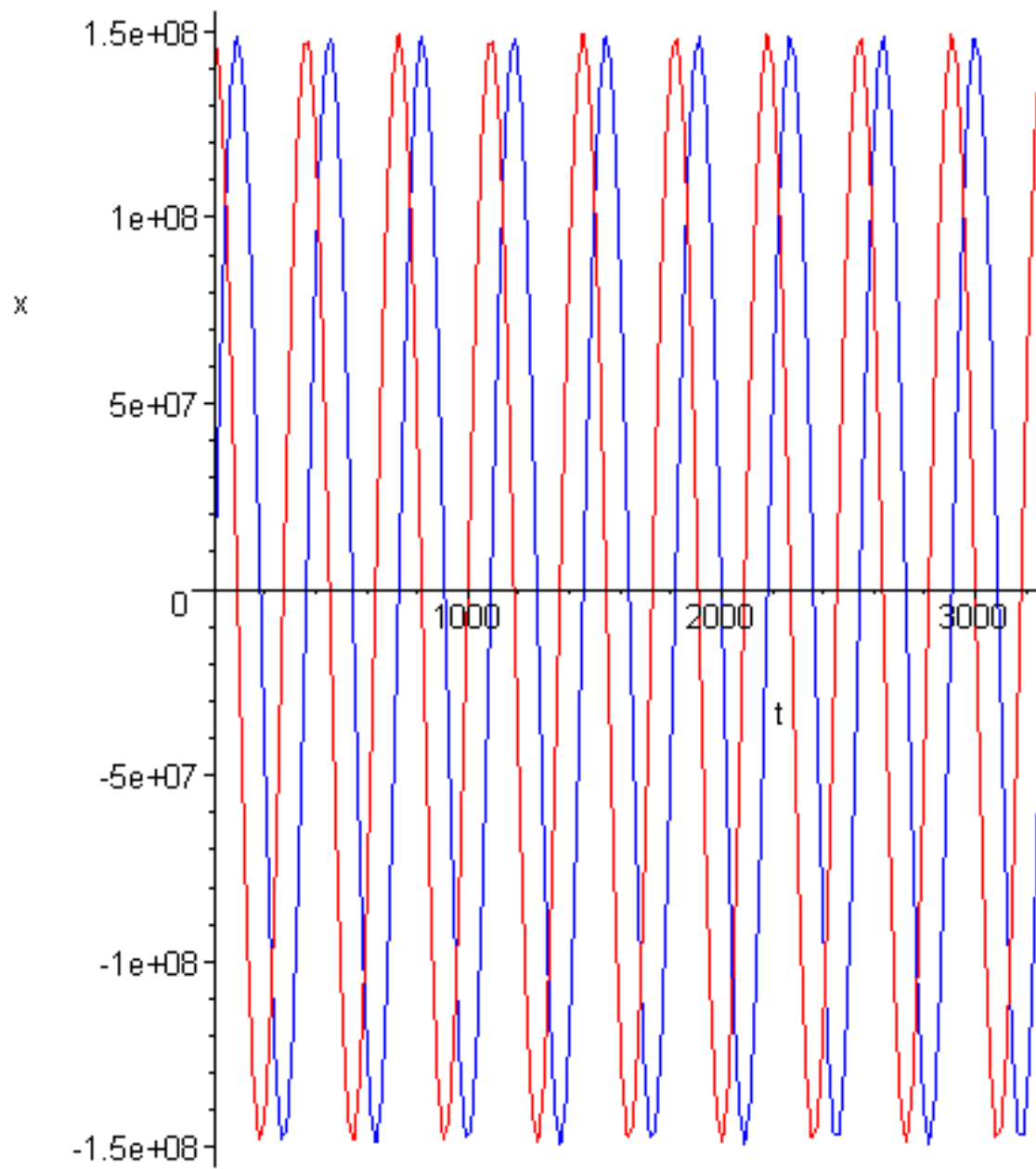
```
> with(plots):
> PA1:=odeplot(Loes_Erde,[t,x(t)],0..365*
```

```

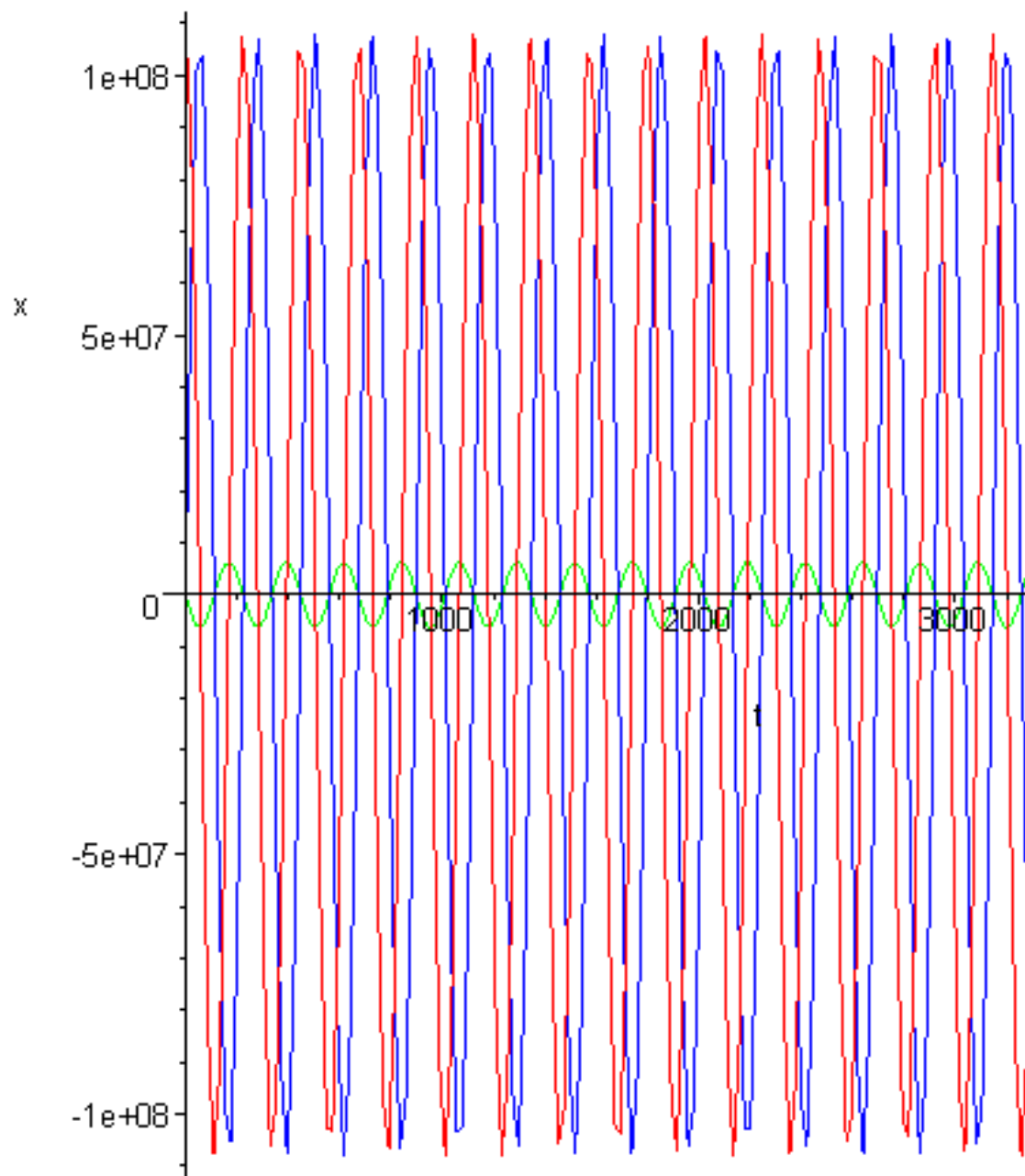
12,color=red,numpoints=200):
PB1:=odeplot(Loes_Erde,[t,y(t)],0..365*
12,color=blue,numpoints=200):
PC1:=odeplot(Loes_Erde,[t,z(t)],0..365*
12,color=green,numpoints=200):
display(PA1,PB1,PC1,title='Erdebewegung');
PA2:=odeplot(Loes_Venus,[t,x(t)],0..365*
12,color=red,numpoints=200):
PB2:=odeplot(Loes_Venus,[t,y(t)],0..365*
12,color=blue,numpoints=200):
PC2:=odeplot(Loes_Venus,[t,z(t)],0..365*
12,color=green,numpoints=200):
display(PA2,PB2,PC2,title='Venusbewegung');
PA3:=odeplot(Loes_Merkur,[t,x(t)],0..365*
12,color=red,numpoints=200):
PB3:=odeplot(Loes_Merkur,[t,y(t)],0..365*
12,color=blue,numpoints=200):
PC3:=odeplot(Loes_Merkur,[t,z(t)],0..365*
12,color=green,numpoints=200):
display(PA3,PB3,PC3,title='Merkurbewegung');
PA4:=odeplot(Loes_Mars,[t,x(t)],0..365*
12,color=red,numpoints=200):
PB4:=odeplot(Loes_Mars,[t,y(t)],0..365*
12,color=blue,numpoints=200):
PC4:=odeplot(Loes_Mars,[t,z(t)],0..365*
12,color=green,numpoints=200):
display(PA4,PB4,PC4,title='Marsbewegung');
PA5:=odeplot(Loes_Jupiter,[t,x(t)],0..365*
12,color=red,numpoints=200):
PB5:=odeplot(Loes_Jupiter,[t,y(t)],0..365*
12,color=blue,numpoints=200):
PC5:=odeplot(Loes_Jupiter,[t,z(t)],0..365*
12,color=green,numpoints=200):
display(PA5,PB5,PC5,title='Jupiterbewegung');

```

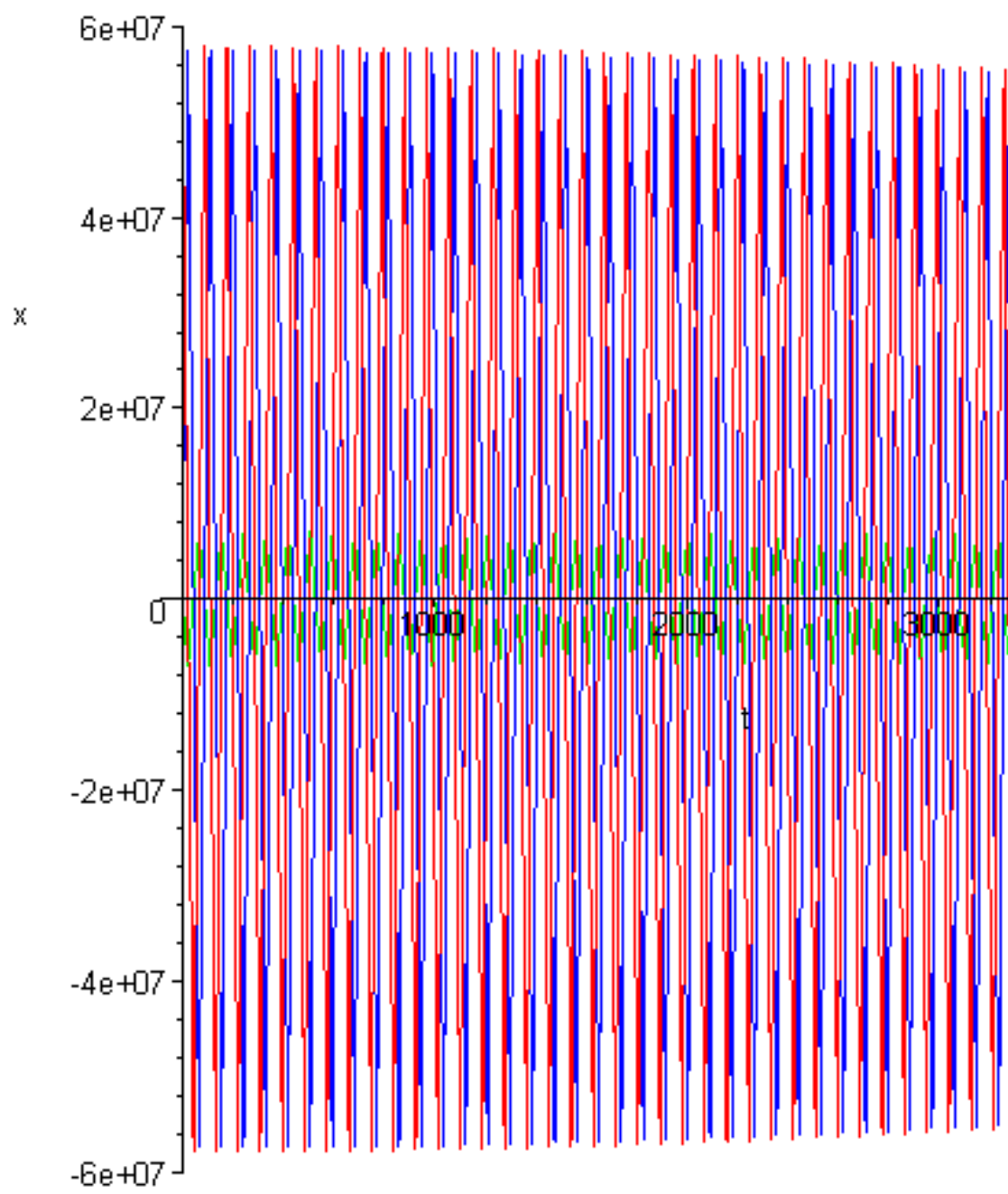
Erdebewegung



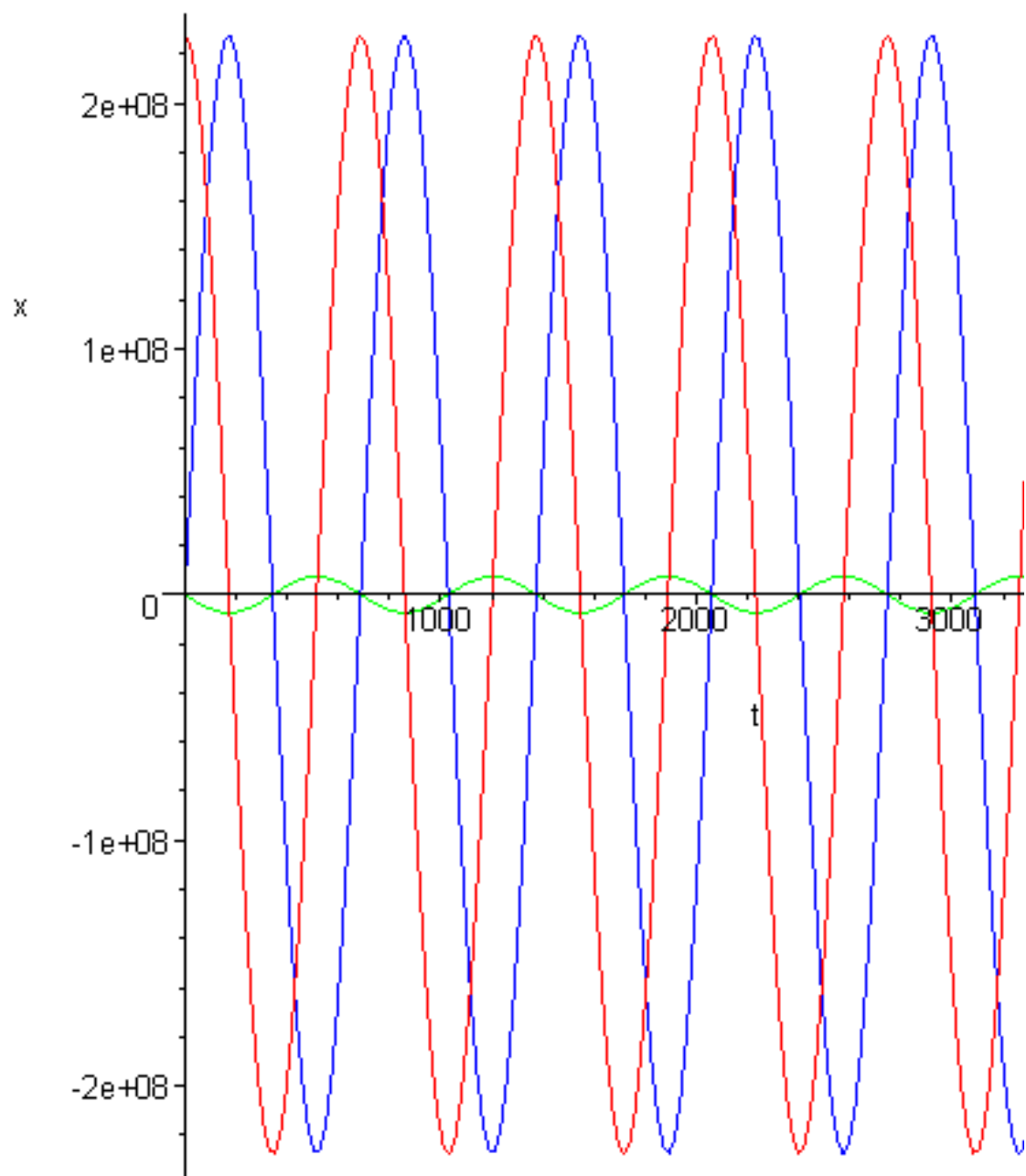
Venusbewegung



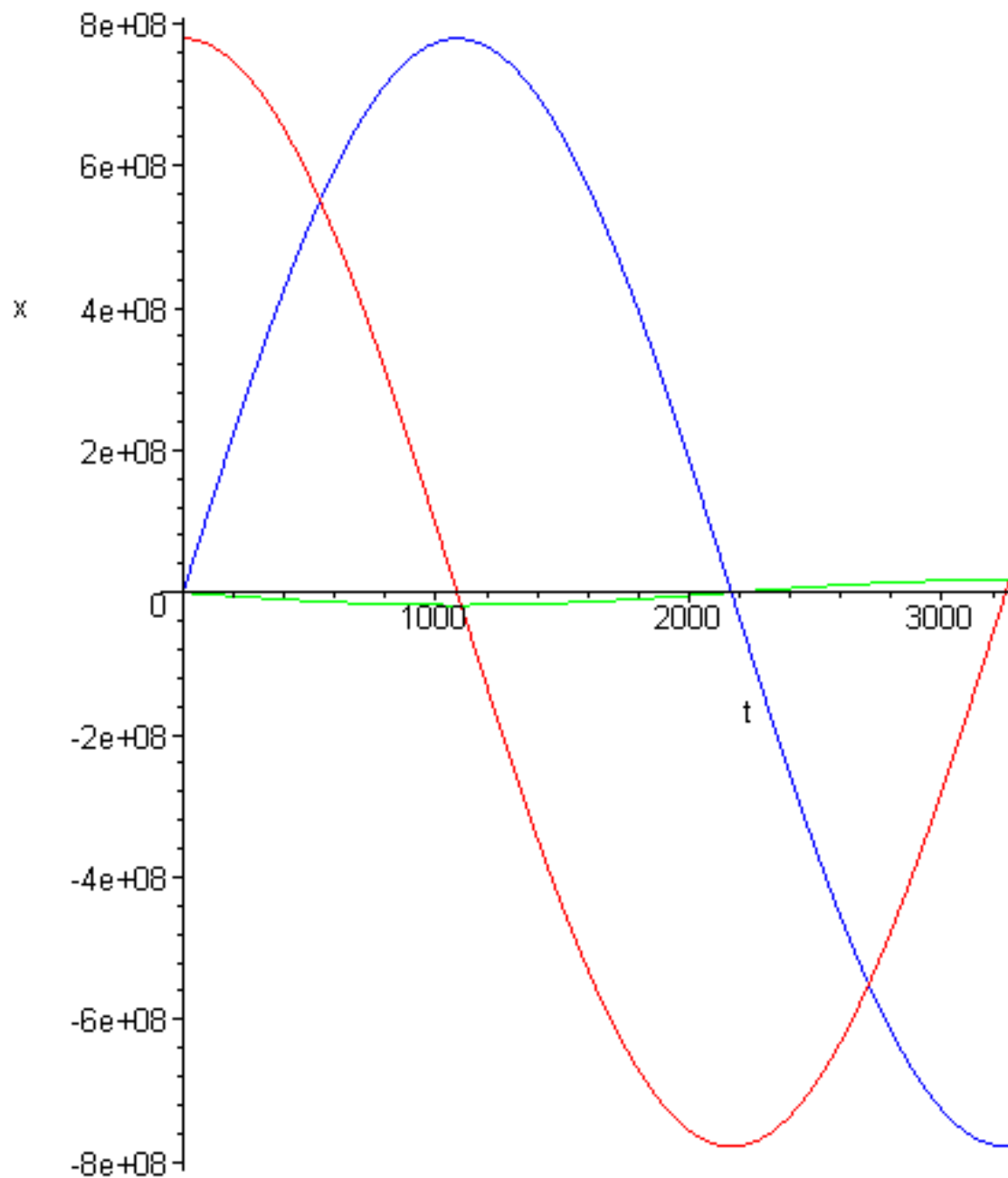
Merkurbewegung



Marsbewegung



Jupiterbewegung



Um die Bewegungen der Planeten im $z(x,y)$ -Diagramm zu veranschaulichen, definieren wir zunächst die Lösungsfunktionen und stellen die Bewegung dann dar.

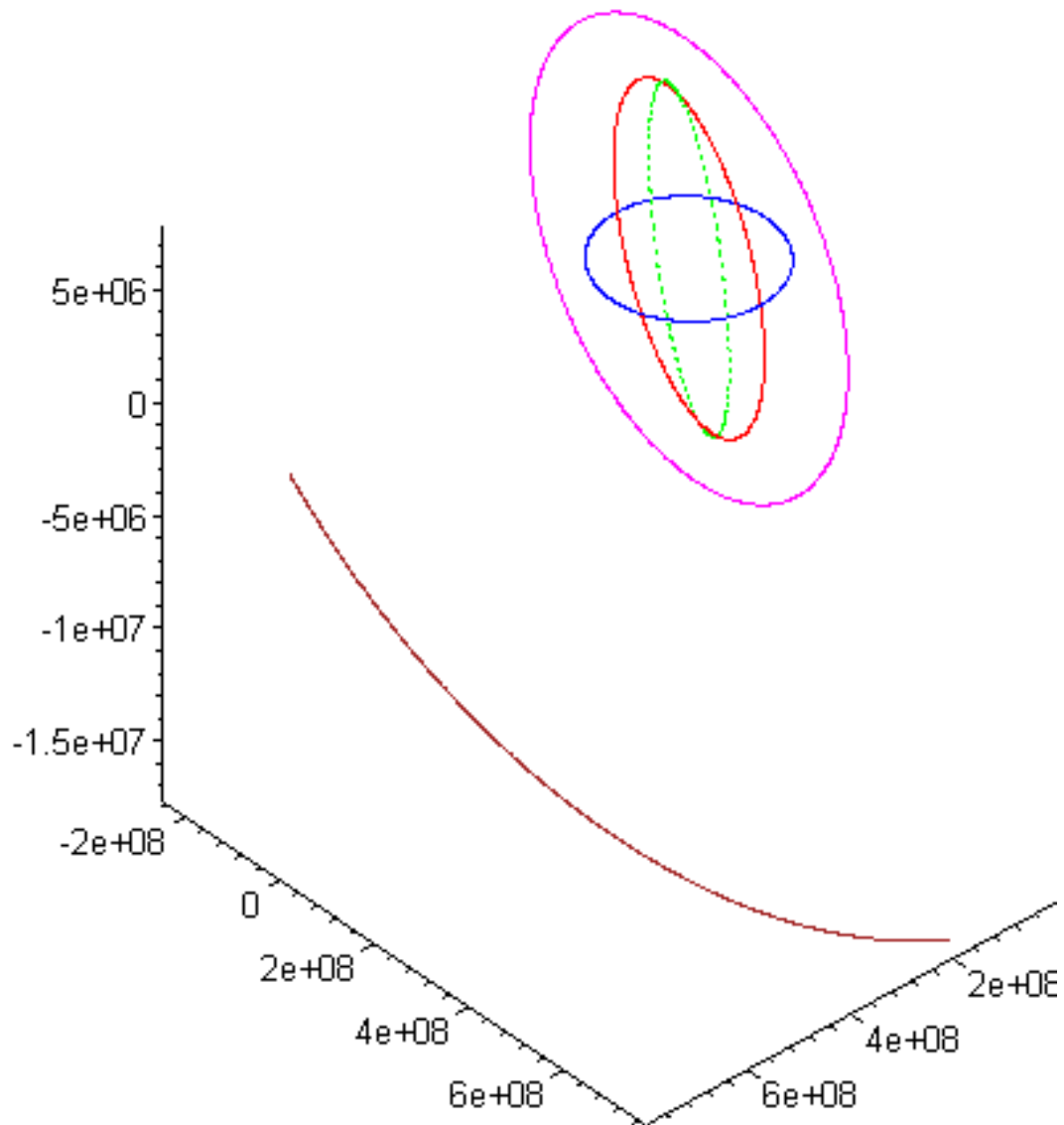
```
> X_Erde:=eval(x(t), Loes_Erde):  
Y_Erde:=eval(y(t), Loes_Erde):  
Z_Erde:=eval(z(t), Loes_Erde):  
X_Venus:=eval(x(t), Loes_Venus):  
Y_Venus:=eval(y(t), Loes_Venus):  
Z_Venus:=eval(z(t), Loes_Venus):  
X_Merkur:=eval(x(t), Loes_Merkur):  
Y_Merkur:=eval(y(t), Loes_Merkur):
```

```

Z_Merkur:=eval(z(t),Loes_Merkur):
X_Mars:=eval(x(t),Loes_Mars):
Y_Mars:=eval(y(t),Loes_Mars):
Z_Mars:=eval(z(t),Loes_Mars):
X_Jupiter:=eval(x(t),Loes_Jupiter):
Y_Jupiter:=eval(y(t),Loes_Jupiter):
Z_Jupiter:=eval(z(t),Loes_Jupiter):

> P_erde:=pointplot3d([seq([X_Erde(t),Y_Erde(t),Z_Erde
(t)],t=0..2.5*365)],color=blue):
P_venus:=pointplot3d([seq([X_Venus(t),Y_Venus(t),Z_Venus
(t)],t=0..2.5*365)],color=red):
P_Merkur:=pointplot3d([seq([X_Merkur(t),Y_Merkur
(t),Z_Merkur(t)],t=0..1.5*365)],color=green):
P_Mars:=pointplot3d([seq([X_Mars(t),Y_Mars(t),Z_Mars
(t)],t=0..2.5*365)],color=magenta):
P_Jupiter:=pointplot3d([seq([X_Jupiter(t),Y_Jupiter
(t),Z_Jupiter(t)],t=0..2.5*365)],color=brown):
display
(P_erde,P_venus,P_Merkur,P_Mars,P_Jupiter,axes=framed,ori
entation=[5,71]);

```



Im folgenden werden die Planetenbewegungen animiert:

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

Warning, the name arrow has been redefined

```
> frames:=70:
```

```
tend:=10*365:
```

```
Sonne:=display(sphere([0,0,0],1.5*10^7,color=yellow)):
```

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

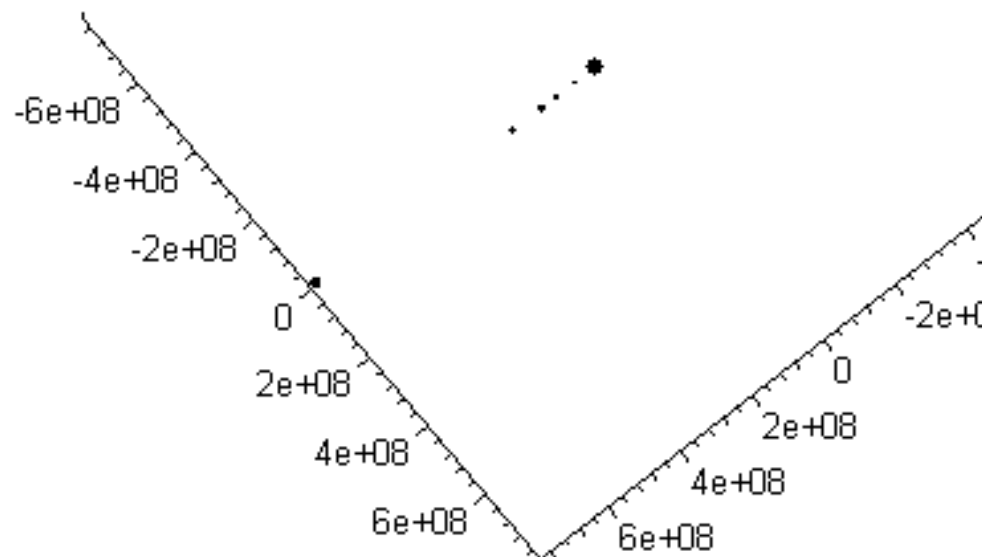
```
Erde[i]:=display(sphere([X_Erde(i*tend/frames),Y_Erde  
(i*tend/frames),Z_Erde(i*tend/frames)],0.6*10^
```

```
7,color=blue)):
```

```

Venus[i]:=display(sphere([X_Venus(i*tend/frames),Y_Venus
(i*tend/frames),Z_Venus(i*tend/frames)],0.5*10^
7,color=red)):
Merkur[i]:=display(sphere([X_Merkur
(i*tend/frames),Y_Merkur(i*tend/frames),Z_Merkur
(i*tend/frames)],0.2*10^7,color=green)):
Mars[i]:=display(sphere([X_Mars(i*tend/frames),Y_Mars
(i*tend/frames),Z_Mars(i*tend/frames)],0.4*10^
7,color=magenta)):
Jupiter[i]:=display(sphere([X_Jupiter
(i*tend/frames),Y_Jupiter(i*tend/frames),Z_Jupiter
(i*tend/frames)],1.0*10^7,color=brown)):
Ani[i]:=display({Erde[i],Venus[i],Merkur[i],Mars
[i],Jupiter[i],Sonne});
od:
10 Erdenjahre vergehen wie im Flug...
> display([seq(Ani[i],i=
0..frames)],insequence=true,scaling=constrained,axes=None
,orientation=[45,75],axes=framed);

```



>

Der Venustransit am 8. Juni 2004

>

Die Wirklichkeit des erdgebundenen Beobachters

Zunächst legen wir einige Größen fest:

> **R_Sonne:=1392000/2:**

R_Erde:=12752/2:

R_Venus:=12228/2:

R_Mond:=3476/2:

A_SV:=108.21*10^6:

A_SE:=149.5*10^6:

A_VE:=A_SE-A_SV:

A_EM:=384400:

Wie groß die Planeten, die Sonne und der Mond einem Beobachter auf der Erde erscheinen, hängt von der Größe der Himmelskörper und von der ihrer Entfernung ab. Wir berechnen nun den jeweiligen Erscheinungswinkel der entsprechenden Himmelskörper:

> EW_S:=evalf(arctan(2*R_Sonne/A_SE)*360/(2*Pi));

EW_V:=evalf(arctan(2*R_Venus/A_VE)*360/(2*Pi));

EW_M:=evalf(arctan(2*R_Mond/A_EM)*360/(2*Pi));

EW_S := 0.5334676947

EW_V := 0.01696809811

EW_M := 0.5180923553

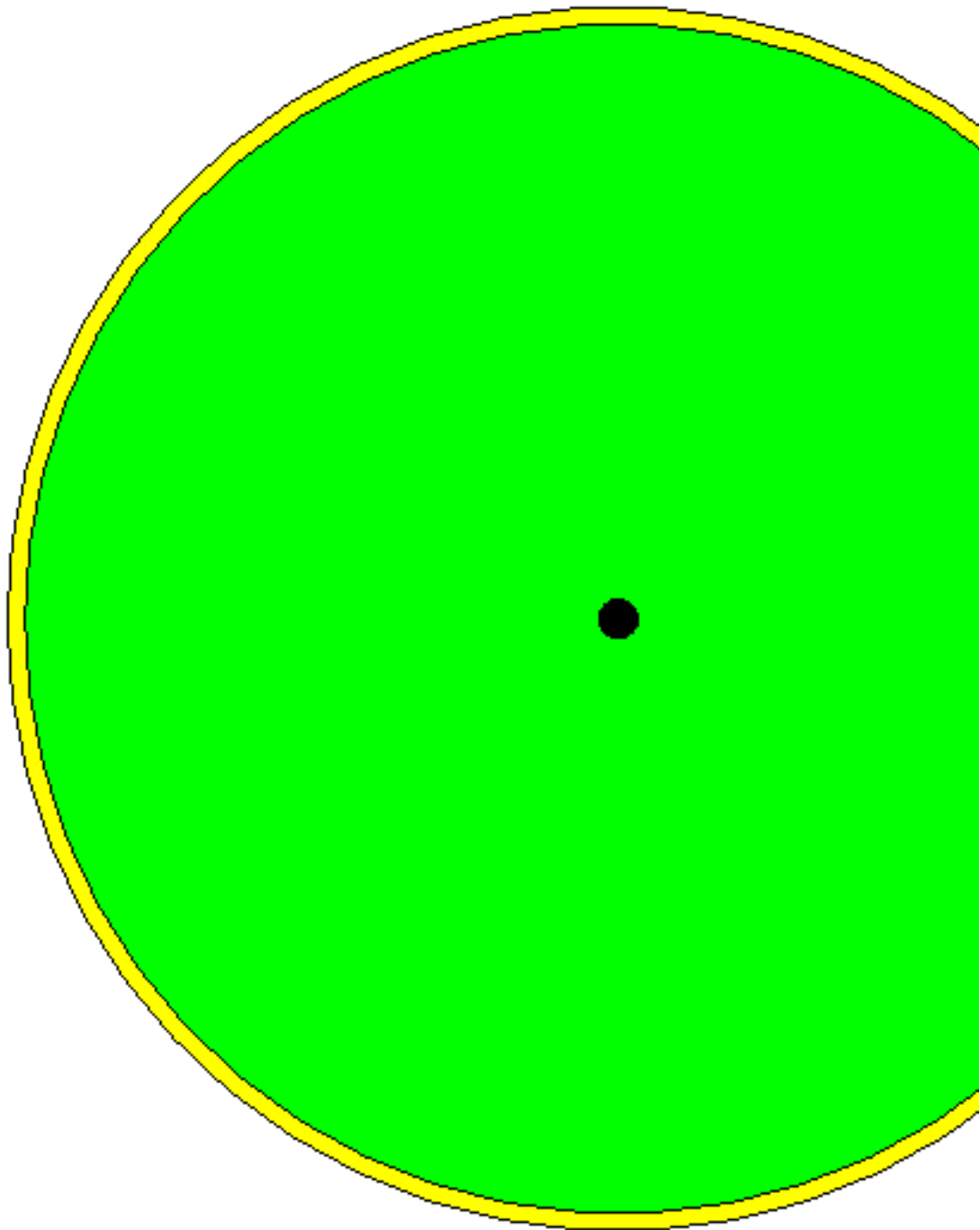
Man kann sich nun die scheinbaren Größenverhältnisse grafisch verdeutlichen (Sonne=gelb, Mond=grün und Venus=schwarz):

> P_S:=display(disk([0,0],tan(EW_S/360*2*Pi)),color=yellow):

P_V:=display(disk([0,0],tan(EW_V/360*2*Pi)),color=black):

P_M:=display(disk([0,0],tan(EW_M/360*2*Pi)),color=green):

display(P_V,P_M,P_S,axes=None);



```

>
> DGLx:=(G,M)->diff(x(t),t,t)=-G*M/(sqrt(x(t)^2+y(t)^2+z
(t)^2))^3*x(t):
DGLy:=(G,M)->diff(y(t),t,t)=-G*M/(sqrt(x(t)^2+y(t)^2+z(t)
^2))^3*y(t):
DGLz:=(G,M)->diff(z(t),t,t)=-G*M/(sqrt(x(t)^2+y(t)^2+z(t)
^2))^3*z(t):
> G:=6.6732*10^(-11)/1000^3*60^2:
Anschr_Rad_S:=evalf(tan(EW_S/360*2*Pi)*A_VE/2);

```

Ansch_Rad_S := 192226.3545

> Ax_erde:=149.5*10^6:

Ay_erde:=0:

Az_erde:=0:

Adx_erde:=0:

Ady_erde:=29.76*60:

Adz_erde:=0:

Ax_venus:=108.21*10^6:

Ay_venus:=-Ansch_Rad_S/1.19:

Az_venus:=-Ansch_Rad_S/1.83:

Adx_venus:=0:

Ady_venus:=34.99*sin(86.76/360*2*Pi)*60:

Adz_venus:=-34.99*sin(3.24/360*2*Pi)*60:

**> Loes_erde:=dsolve({DGLx(G,M),DGLy(G,M),DGLz(G,M),x(0)
=Ax_erde,y(0)=Ay_erde,z(0)=Az_erde,D(x)(0)=Adx_erde,D(y)
(0)=Ady_erde,D(z)(0)=Adz_erde},{x(t),y(t),z
(t)},type=numeric,output=listprocedure);
Loes_venus:=dsolve({DGLx(G,M),DGLy(G,M),DGLz(G,M),x(0)
=Ax_venus,y(0)=Ay_venus,z(0)=Az_venus,D(x)(0)=Adx_venus,D
(y)(0)=Ady_venus,D(z)(0)=Adz_venus},{x(t),y(t),z
(t)},type=numeric,output=listprocedure);**

Loes_erde := $\left[t = (\text{proc}(t) \dots \text{end proc}), x(t) = (\text{proc}(t) \dots \text{end proc}), \right.$
 $\left. \frac{d}{dt} x(t) = (\text{proc}(t) \dots \text{end proc}), y(t) = (\text{proc}(t) \dots \text{end proc}), \right.$

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

Loes_venus := $\left[t = (\text{proc}(t) \dots \text{end proc}), x(t) = (\text{proc}(t) \dots \text{end proc}), \right.$
 $\left. \frac{d}{dt} x(t) = (\text{proc}(t) \dots \text{end proc}), y(t) = (\text{proc}(t) \dots \text{end proc}), \right.$

$$\left. \begin{aligned} \frac{d}{dt} y(t) &= (\text{proc}(t) \dots \text{end proc}), z(t) = (\text{proc}(t) \dots \text{end proc}), \\ \frac{d}{dt} z(t) &= (\text{proc}(t) \dots \text{end proc}) \end{aligned} \right]$$

```
> X_erde:=eval(x(t), Loes_erde):
Y_erde:=eval(y(t), Loes_erde):
Z_erde:=eval(z(t), Loes_erde):
X_venus:=eval(x(t), Loes_venus):
Y_venus:=eval(y(t), Loes_venus):
Z_venus:=eval(z(t), Loes_venus):
```

Wir definieren nun zwei Ortsvektoren. V_ES beschreibt den Vektor von der Erde zur Sonne und V_EV beschreibt den Vektor von der Erde zur Venus.

```
> with(linalg):
> V_EV:=t->vector([X_erde(t)-X_venus(t), Y_erde(t)-Y_venus
(t), Z_erde(t)-Z_venus(t)]);
V_ES:=t->vector([X_erde(t), Y_erde(t), Z_erde(t)]);
```

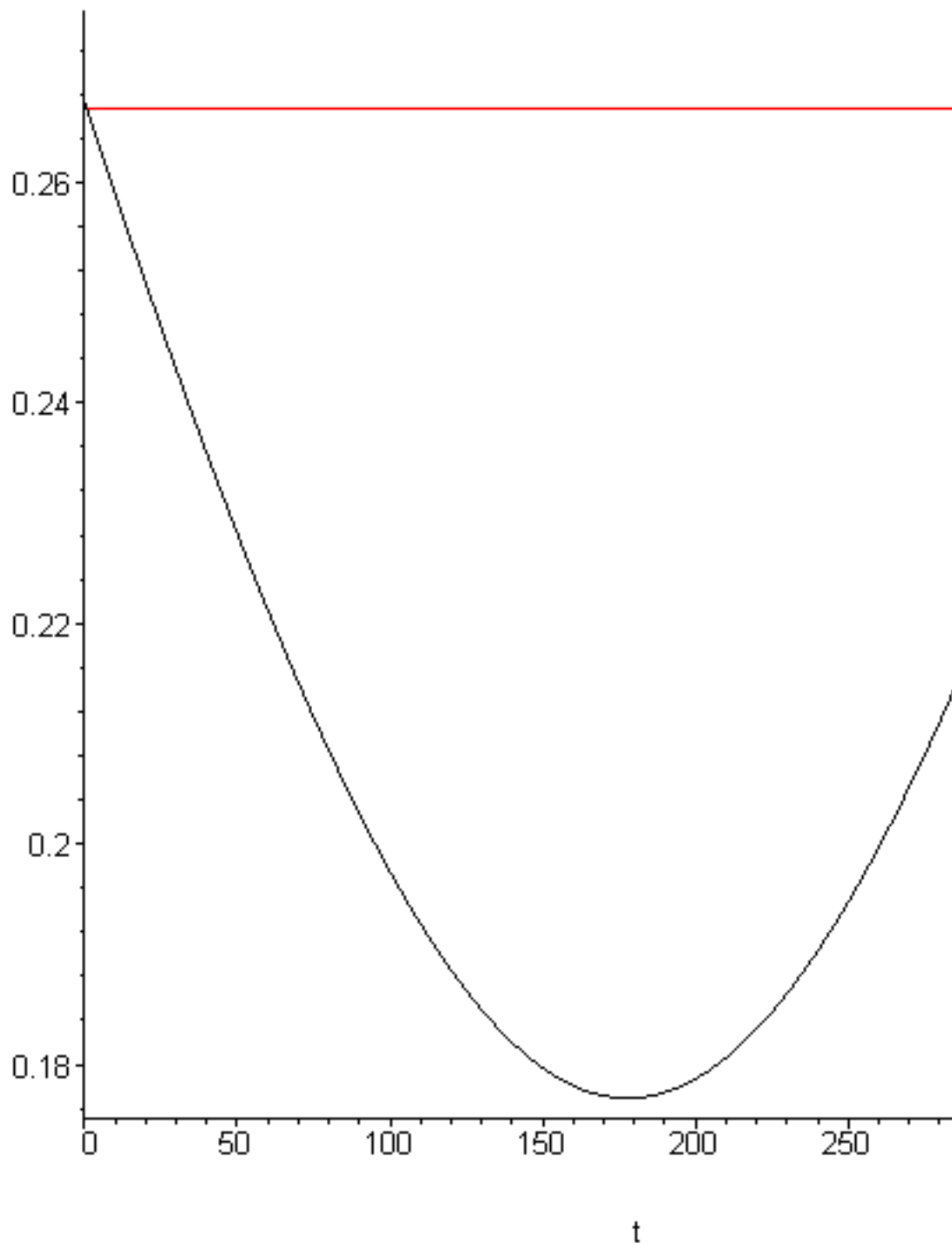
V_EV:=

$t \rightarrow [X_{\text{erde}}(t) - X_{\text{venus}}(t), Y_{\text{erde}}(t) - Y_{\text{venus}}(t), Z_{\text{erde}}(t) - Z_{\text{venus}}(t)]$

$V_{ES} := t \rightarrow [X_{\text{erde}}(t), Y_{\text{erde}}(t), Z_{\text{erde}}(t)]$

Der Venustransit dauert so lange, bis der von den Vektoren V_EV und V_ES eingespannte Winkel kleiner ist, als der Erscheinungswinkel EW_S der Sonne. Man erkennt, dass die Dauer des Venustransiten 2004 ca. 6 Stunden beträgt

```
> Pangl:=listplot([seq([t, evalf(360*angle(V_ES(t), V_EV
(t)))/2/Pi], t=0..6*60+3)]):
PS:=plot(EW_S/2, t=0..6*60+3):
display(Pangl, PS);
```



Wir stellen nun den Venustransit als Animation in der y-z Ebene dar (so wie man ihn von der Erde aus beobachten würde). Hierzu definieren wir zunächst die folgenden Vektoren:

```
> V_EVx:=t->vector([-X_erde(t)-X_venus(t)], -(Y_erde(t)-
Y_venus(t))):
V_EVz:=t->vector([-X_erde(t)-X_venus(t)], Z_erde(t)-
Z_venus(t))):
V_ESx:=t->vector([-X_erde(t), -Y_erde(t)]):
V_ESz:=t->vector([-X_erde(t), Z_erde(t)]):

> Bild_Sonne:=evalf(tan(EW_S/360*2*Pi)/2):
```

```

Bild_Venus:=evalf(tan(EW_V/360*2*Pi)/2):
> Ort:=TEXT([0,Bild_Sonne/1.70],"Beobachtung von meinem
Haus aus gesehen"):
frames:=100:
tend:=6*60+3:
Sonne:=display(disk
([0,0],Bild_Sonne,color=yellow),axes=None):
Time1:=TEXT([0,Bild_Sonne/1.15],"Vergangene Zeit in
Minuten"):
for i from 0 by 1 to frames do
Time2[i]:=TEXT([0,Bild_Sonne/1.3],convert(round
((i*tend)/frames),string)):
if (angle(V_EVx(((i+1)*tend)/frames),V_ESx(((i+1)
*tend)/frames)) > angle(V_EVx((i*tend)/frames),V_ESx
((i*tend)/frames))) then Vorz:=1 else Vorz:=-1 fi;
Venus[i]:=display(disk([Vorz*sin(angle(V_EVx
(i*tend/frames),V_ESx(i*tend/frames))),-sin(angle(V_EVz
(i*tend/frames),V_ESz
(i*tend/frames)))]),Bild_Venus,color=black),axes=None):
Ani[i]:=display({Venus[i],Sonne,Ort,Time1,Time2[i]});
od:

```

Diese Animation illustriert den Verlauf des Venustransits am 8.Juni 2004. Bei uns wird der erste Kontakt der Venus mit der Sonne um ca. 7.03 Uhr stattfinden. Der Venus-Sonne Kontakt endet um ca. 13.07.

```

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

```



>

Spezielle Relativitätstheorie

Da durch die spezielle Relativitätstheorie die Ausbreitungsgeschwindigkeit des Lichts begrenzt ist, ist das von der Erde aus beobachtete Bild der Sonne eigentlich nicht das Wirkliche. Das Licht der Sonne benötigt ca. 8 Minuten 19 Sek. für seinen Weg zur Erde. Wir sehen somit um ca. 7.20 den Eintritt der Venus, obwohl das dann gesehene Bild der Sonne schon vor 8 Minuten 19 Sek. von der Sonne aus zu uns gesandt wurde, und somit ein vergangenes ist. Genau genommen sehen wir zwei unterschiedliche Vergangenheiten in einem Bild, da das Bild der Sonne ca. 8 Minuten alt ist, der Schatten der Venus jedoch nur ca. 2 Minuten 18 Sek. alt ist. Eine Raumzeit-

unabhängige allgemeingültige Wirklichkeit existiert nicht, sondern die Wirklichkeit eines Beobachters hängt von seinem raumzeitlichen Standort ab.

Wir definieren die Lichtgeschwindigkeit c (in km/Min) und berechnen die benötigte Zeit die Licht von der Sonne zur Erde und von der Venus zur Erde benötigt

```
> c:=299792*60;
```

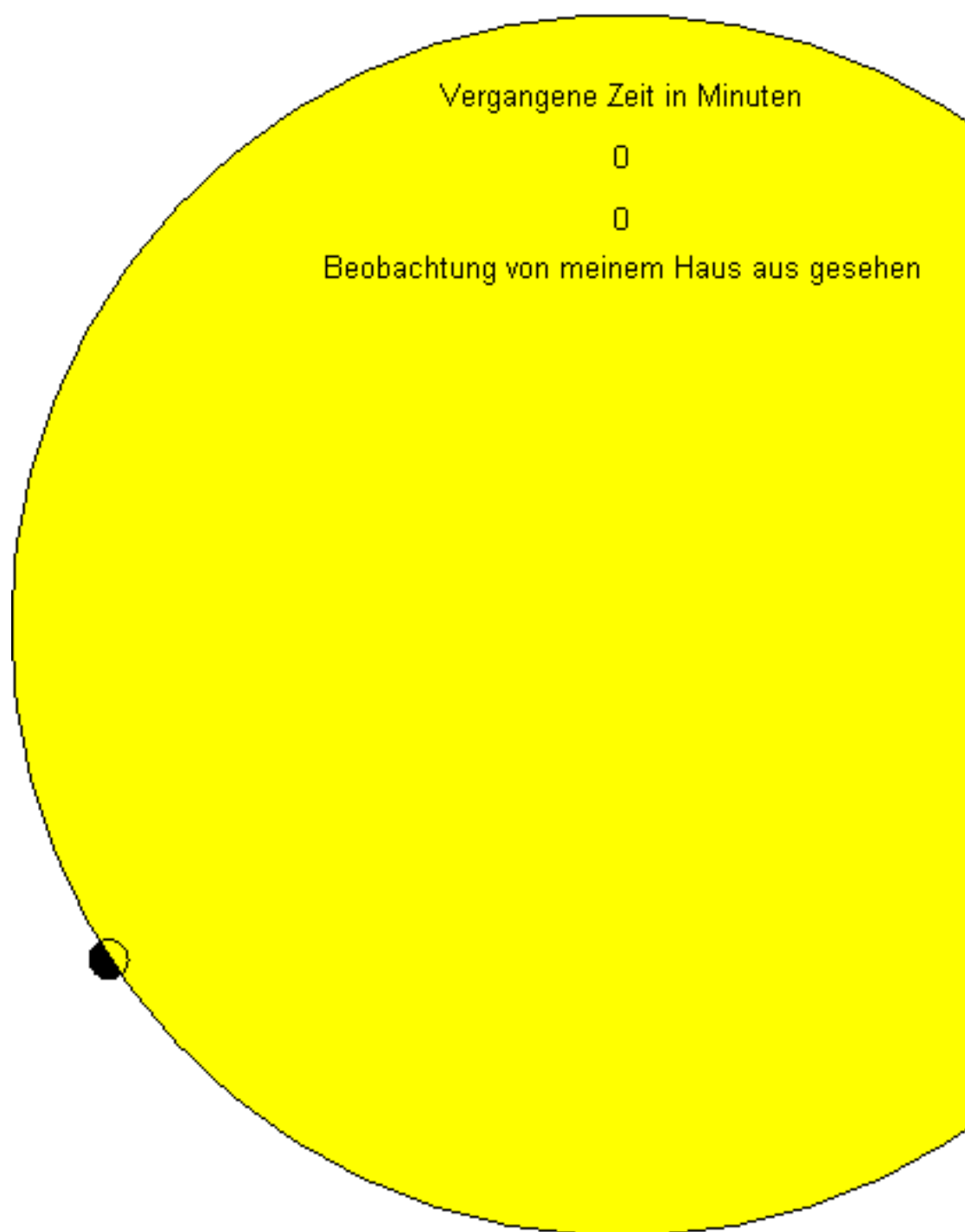
```
c:=17987520
```

```
> t_SE:=evalf(A_SE/c);  
t_VE:=evalf(A_VE/c);
```

```
t_SE:=8.311318069
```

```
t_VE:=2.295480422
```

```
> frames:=100:  
tend:=6*60+3+t_SE tend_rel:=6*60+3:  
Sonne:=display(disk  
([0,0],Bild_Sonne,color=yellow),axes=None):  
Time1:=TEXT([0,Bild_Sonne/1.15],"Vergangene Zeit in  
Minuten"):  
for i from 0 by 1 to frames do  
Time2[i]:=TEXT([0,Bild_Sonne/1.3],convert(round  
((i*tend)/frames),string)):  
Time2_rel[i]:=TEXT([0,Bild_Sonne/1.5],convert(round  
((i*tend_rel)/frames),string)):  
  
if (angle(V_EVx(((i+1)*tend)/frames),V_ESx(((i+1)  
*tend)/frames)) > angle(V_EVx((i*tend)/frames),V_ESx  
((i*tend)/frames))) then Vorz:=1 else Vorz:=-1 fi;  
Venus[i]:=display(disk([Vorz*sin(angle(V_EVx  
(i*tend/frames),V_ESx(i*tend/frames))),-sin(angle(V_EVz  
(i*tend/frames),V_ESz  
(i*tend/frames)))]),Bild_Venus,color=black),axes=None):  
if (angle(V_EVx(((i+1)*tend_rel)/frames),V_ESx(((i+1)  
*tend_rel)/frames)) > angle(V_EVx  
((i*tend_rel)/frames),V_ESx((i*tend_rel)/frames))) then  
Vorz_rel:=1 else Vorz_rel:=-1 fi;  
Venus_rel[i]:=display(disk([Vorz_rel*sin(angle(V_EVx  
(i*tend_rel/frames),V_ESx(i*tend_rel/frames))),-sin  
(angle(V_EVz(i*tend_rel/frames),V_ESz  
(i*tend_rel/frames)))]),Bild_Venus,color=magenta),axes=None):  
Ani[i]:=display({Venus[i],Venus_rel  
[i],Sonne,Ort,Time1,Time2[i],Time2_rel[i]});  
od:  
  
> display([seq(Ani[i],i=  
0..frames)],insequence=true,scaling=constrained);
```



>