EBERHARD KARLS UNIVERSITÄT TÜBINGEN

MATHEMATISCH-NATURWISSENSCHAFTLICHE FAKULTÄT
Fachbereich Physik

Basic Physics Course with MATLAB's Symbolic Toolbox and Live Editor

Kurt Bräuer

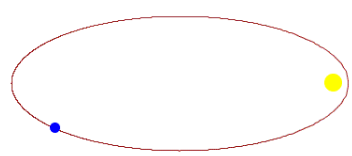
4.2 Keplerian orbits (Computational example)

A body with mass *m* moves in the

Gravitational potential $V(r) = \frac{\alpha}{r}$

The movement takes place in the x-y plane, so that it can be described with

Polar coordinates $\vec{r} = \begin{pmatrix} r\cos\varphi \\ r\sin\varphi \end{pmatrix}$



```
clear all
syms alpha m p_r p_phi r phi positive
syms E
assume(E<0)
Par=[E==-2/25 m==1 alpha==1 p_phi==1]
```

Par =

$$\left(\mathrm{E} = -\frac{2}{25} \quad m = 1 \quad \alpha = 1 \quad p_\varphi = 1\right)$$

```
r_v=[r*cos(phi);r*sin(phi)]
```

r_v =

$$\begin{pmatrix} r\cos(\varphi) \\ r\sin(\varphi) \end{pmatrix}$$

1 Ground vectors and metric

Jacobian *J*, metric *G* and contravariant metric G_i

$$J = \left(\frac{\partial x_i}{\partial q_j}\right),\ G = J'J,\ G_i = G^{-1}$$

```
J=jacobian(r_v,[r,phi])
```

```
J =
```

$$\begin{pmatrix} \cos(\varphi) & -r\sin(\varphi) \\ \sin(\varphi) & r\cos(\varphi) \end{pmatrix}$$

```
G=simplify(transpose(J)*J)
```

```
G =
```

$$\begin{pmatrix} 1 & 0 \\ 0 & r^2 \end{pmatrix}$$

```
G_i=G^(-1)
```

```
G_i =
```

$$\begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{r^2} \end{pmatrix}$$

2 Hamiltonian

$$H = \frac{1}{2m}(p_r, p_\varphi)G_i\begin{pmatrix} p_r \\ p_\varphi \end{pmatrix} + V(r)$$

```
p_v=[p_r;p_phi]
```

```
p_v =
```

$$\begin{pmatrix} p_r \\ p_\varphi \end{pmatrix}$$

```
H=1/(2*m)*(p_v'*G_i*p_v)-alpha/r;
H=expand(H)
```

```
H =
```

$$\frac{{p_r}^2}{2\,m}-\frac{\alpha}{r}+\frac{{p_\varphi}^2}{2\,m\,r^2}$$

3 Radial momentum

$E = H(p_r, \ldots) \;\; \to \;\; p_r(E, \ldots)$

```
E==H
```

```
ans =
```

$$\mathrm{E}=\frac{{p_r}^2}{2\,m}-\frac{\alpha}{r}+\frac{{p_\varphi}^2}{2\,m\,r^2}$$

```
p_rs=solve(ans,p_r,'IgnoreProperties', true')
```

```
p_rs =
```

$$\begin{pmatrix}\dfrac{\sqrt{-{p_\varphi}^2+2\,\mathrm{E}\,m\,r^2+2\,\alpha\,m\,r}}{r}\\ -\dfrac{\sqrt{-{p_\varphi}^2+2\,\mathrm{E}\,m\,r^2+2\,\alpha\,m\,r}}{r}\end{pmatrix}$$

4 Aphelion and perihelion (reversal points)

$p_r(r) = 0 \;\; \to \;\; r_{ap}, \; r_{pe}$

```
r_ape=solve(p_rs(2),r,'IgnoreProperties', true);
[r_ap_,ind]=sort((sube(r_ape,Par))); %Sorting
r_ap=r_ape(ind)
```

```
r_ap =
```

$$\begin{pmatrix}-\dfrac{\alpha\,\sqrt{m}-\sqrt{m\,\alpha^2+2\,\mathrm{E}\,{p_\varphi}^2}}{2\,\mathrm{E}\,\sqrt{m}}\\ -\dfrac{\alpha\,\sqrt{m}+\sqrt{m\,\alpha^2+2\,\mathrm{E}\,{p_\varphi}^2}}{2\,\mathrm{E}\,\sqrt{m}}\end{pmatrix}$$

```
double(sube(r_ap,Par))
```

```
ans = 2×1
    0.5218
   11.9782
```

5 Keplerian orbits

Azimuth angle $\varphi(r) + \frac{\pi}{2} = \int d\varphi = \int \frac{m\dot{r}}{G_{rr}^{-1} p_r(r)} \dot{\varphi} dt = \int \frac{p_\varphi}{r^2 p_r(r)} dr$

Therefore: $r(\varphi) = \frac{p_\varphi^2}{m\alpha + A\cos(\varphi)}, \; \left(A \equiv \sqrt{\alpha^2 m^2 + 2Emp_\varphi}\right)$

Integrand $\frac{p_\varphi}{r^2 p_r(r)}$

```
IG=p_phi/(r^2*p_rs(1))
```

IG =

$$\frac{p_\varphi}{r\,\sqrt{-{p_\varphi}^2 + 2\,\mathrm{E}\,m\,r^2 + 2\,\alpha\,m\,r}}$$

Integration and substraction of the constant of integration (integal value in the perihel)

```
int(IG,r)
```

ans =

$$\log\left(\alpha\,m - \frac{{p_\varphi}^2}{r} + \frac{p_\varphi\,\sqrt{-{p_\varphi}^2 + 2\,\mathrm{E}\,m\,r^2 + 2\,\alpha\,m\,r}\;\mathrm{i}}{r}\right)\mathrm{i}$$

```
phi_int=ans-sube(ans,r==r_ap(1))
```

phi_int =

$$\log\left(\alpha\,m - \frac{{p_\varphi}^2}{r} + \frac{p_\varphi\,\sqrt{-{p_\varphi}^2 + 2\,\mathrm{E}\,m\,r^2 + 2\,\alpha\,m\,r}\;\mathrm{i}}{r}\right)\mathrm{i} - \log\left(\alpha\,m + \frac{2\,\mathrm{E}\,\sqrt{m}\,{p_\varphi}^2}{\sigma_1} - \frac{2\,\mathrm{E}\,\sqrt{m}\,p_\varphi\,\sqrt{\frac{{\sigma_1}^2}{2\,\mathrm{E}} - {p_\varphi}^2 - \frac{\alpha\,\sqrt{m}\,\sigma_1}{\mathrm{E}}}\;\mathrm{i}}{\sigma_1}\right)$$

where

$$\sigma_1 = \alpha\,\sqrt{m} - \sqrt{m\,\alpha^2 + 2\,\mathrm{E}\,{p_\varphi}^2}$$

Inverting to the form $r(\varphi) = \frac{p}{a + \epsilon\cos(\varphi)}$ (Kepler-orbits)

```
cos(phi+pi)==simplify(cos(phi_int),'Steps',100)
```

ans =

$$-\cos(\varphi) = \frac{{p_\varphi}^2 - \alpha\,m\,r}{\sqrt{m}\;r\;\sqrt{m\,\alpha^2 + 2\,\mathrm{E}\,{p_\varphi}^2}}$$

```
er=isolate(ans,r)
```

```
er =
```

$$r = \frac{{p_{\varphi}}^2}{\alpha\, m - \sqrt{m}\, \cos(\varphi)\, \sqrt{m\, \alpha^2 + 2\, \mathrm{E}\, {p_{\varphi}}^2}}$$

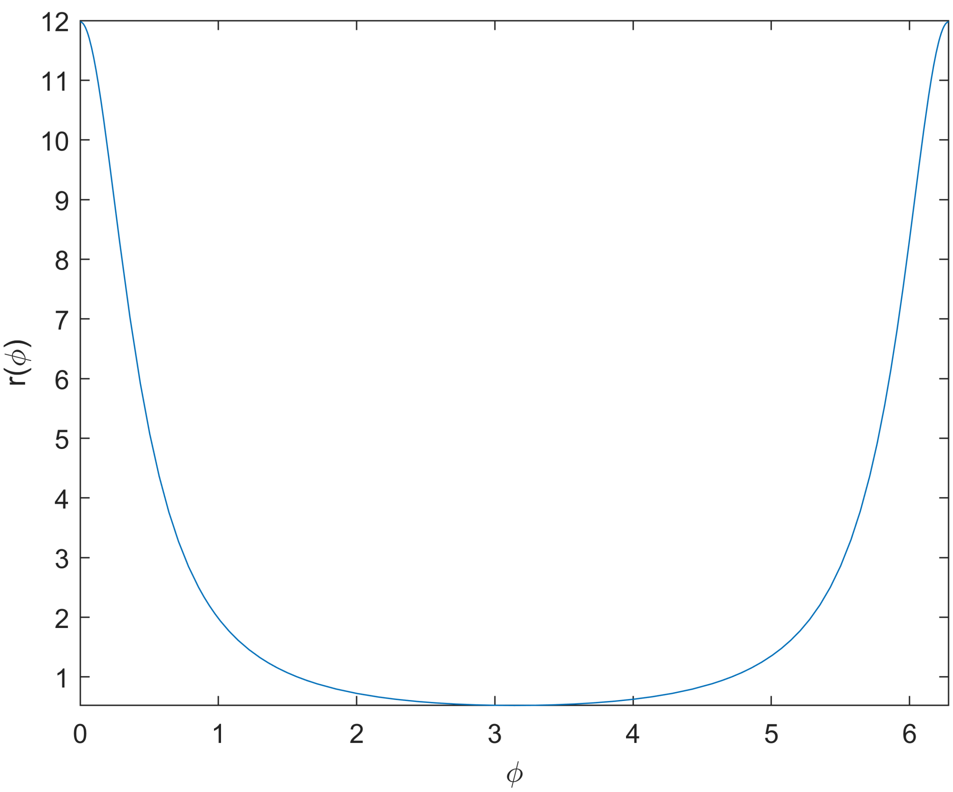
```
erP=r==simplify(sube(rhs(er),Par))
```

```
erP =
```

$$r = -\frac{1}{\frac{\sqrt{21}\, \cos(\varphi)}{5} - 1}$$

Graphics

```
fplot(sube(r,erP),[0 2*pi])
xlabel('\phi')
ylabel('r(\phi)')
```



6 Period

$$T = 2 \int_{t(r_{pe})}^{t(r_{ah})} dt = 2 \int_{t(r_{pe})}^{t(r_{ah})} \frac{m\dot{r}}{G_{rr}^{-1} p_r(r)} dt = 2 \int_{r_{pe}}^{r_{ah}} \frac{m}{p_r(r)} dr$$

```
T=2*m*int(1/p_rs(1),r,r_ap(1),r_ap(2))
```

with parameters

```
TP=real(sube(T,Par))
TP=double(TP)
```

7 Numerical solution

Equation of motion: $\frac{d}{dt}\phi = \frac{\partial}{\partial p_\phi} H(t)$

```
syms phi(t)
dphi_dt=diff(H,p_phi)
sube(dphi_dt,[Par,erP])
sube(ans,phi==phi(t))
odeFunction(ans,phi(t))
[tn,phi_s] = ode45(ans,linspace(0,TP,500),0);
```

8 Graphics

```
subplot(1,2,1)
plot(tn,phi_s,[0 100],2*pi*[1 1],':k')
xlabel('Time t')
ylabel('Angle \phi')
title('Azimuth \phi(t)')
axis square
subplot(1,2,2)
r_s=matlabFunction(sube(r,erP));
rx_s=matlabFunction(r_v(1));
ry_s=matlabFunction(r_v(2));

x=rx_s(phi_s,r_s(phi_s));
y=ry_s(phi_s,r_s(phi_s));
plot(x,y,'r')
hold on
rectangle('Position',0.5*[-1 -1 2 2],'Curvature',[1 1],'FaceColor','y')
k=8;
plot(x(k),y(k),'ob','MarkerSize',5);
xlabel('x-direction')
ylabel('y-direction')
title('Keplerian orbit')
axis equal
axis([-2 13 -3 3])
hold off
```

9 Animation $\rightarrow$ 'Keplerian_Orbit.gif'

```
figure
hold on
plot(x,y,'g')

rectangle('Position',0.4*[-1 -1 2 2],'Curvature',[1 1],'FaceColor','y','EdgeColor','r');
PM=rectangle('Position',0.25*[-1 -1 2 2]+[x(1) y(1) 0 0],'Curvature',[1 1],'FaceColor','b');
axis equal
axis off
axis([-.7 12.5 -3 3])

clear im
im{1}=frame2im(getframe);
cnt=1;
for n=3:4:size(x,1)
    PM.Position=0.25*[-1 -1 2 2]+[x(n) y(n) 0 0];
    im{cnt}=frame2im(getframe);
    cnt=cnt+1;
end
```

```
FiNa='Keplerian_Orbit.gif';
n=1;
[A,map]=rgb2ind(im{n},16);
imwrite(A,map,FiNa,'gif','LoopCount',Inf,'DelayTime',0);
for n=2:size(im,2)
    [A,map]=rgb2ind(im{n},16);
    imwrite(A,map,FiNa,'gif','WriteMode','append','DelayTime',0);
end
```