Double Pendulum

Based on Freeball Equations of Motion for the Double Pendulum

Written by Daniel Volinski at danielvolinski@yahoo.es

(%i2) `info:build_info()$info@version;`

(%o2)

5.38.1

(%i2) `reset()$kill(all)$`

(%i1) `derivabbrev:true$`

(%i2) `ratprint:false$`

(%i3) `fpprintprec:5$`

(%i4) `if get('draw,'version)=false then load(draw)$`

(%i5) `wxplot_size:[1024,768]$`

(%i6) `if get('optvar,'version)=false then load(optvar)$`

(%i7) `if get('rkf45,'version)=false then load(rkf45)$`

(%i8) `declare(trigsimp,evfun)$`

(%i9) `declare(t,mainvar)$`

1 Settings

Assume:

- Point masses
- Massless, rigid rods
- Gravity is present

(%i10) `declare([m_1,m_2,l_1,l_2,g],constant)$`

(%i11) `assume(m_1>0,m_2>0,l_1>0,l_2>0,g>0)$`

(%i12) `params:[m_1=1,m_2=2,l_1=2,l_2=3,g=9.8]$`

(%i13) `τ:17$`

Generalized coordinates

(%i14) `ζ:[θ_1,θ_2]$`

(%i15) `depends(ζ,t)$`

(%i16) `dim:length(ζ)$`

Kinematic Constraints

(%i18)
```
x_1:l_1*sin(θ_1)$
y_1:-l_1*cos(θ_1)$
```

(%i20)
```
x_2:l_1*sin(θ_1)+l_2*sin(θ_2)$
y_2:-l_1*cos(θ_1)-l_2*cos(θ_2)$
```

Velocities

(%i21) `ldisplay(v_1:[diff(x_1,t),diff(y_1,t)])$`

$$v_1 = [l_1 \cos(\theta_1) \left(\dot{\theta}_1\right), l_1 \sin(\theta_1) \left(\dot{\theta}_1\right)] \tag{\%t21}$$

(%i22) `ldisplay(v_2:[diff(x_2,t),diff(y_2,t)])$`

$$v_2 = [l_2 \cos(\theta_2) \left(\dot{\theta}_2\right) + l_1 \cos(\theta_1) \left(\dot{\theta}_1\right), l_2 \sin(\theta_2) \left(\dot{\theta}_2\right) + l_1 \sin(\theta_1) \left(\dot{\theta}_1\right)] \tag{\%t22}$$

2 Lagrangian Formalism

(%i23) `kill(labels)$`

Kinetic Energy

(%i1) `ldisplay(T:`$\frac{1}{2}$`*m_1*(v_1.v_1)+`$\frac{1}{2}$`*m_2*(v_2.v_2))$`

$$T = \frac{m_2\left(\left(l_2\sin(\theta_2)\left(\dot{\theta}_2\right)+l_1\sin(\theta_1)\left(\dot{\theta}_1\right)\right)^2+\left(l_2\cos(\theta_2)\left(\dot{\theta}_2\right)+l_1\cos(\theta_1)\left(\dot{\theta}_1\right)\right)^2\right)}{2} + \frac{m_1\left({l_1}^2\sin(\theta_1)^2\left(\dot{\theta}_1\right)^2+{l_1}^2\cos(\theta_1)^2\left(\dot{\theta}_1\right)^2\right)}{2}$$

Potential Energy

(%i2) `ldisplay(V:m_1*g*y_1+m_2*g*y_2),expand$`

$$V = -g\,l_2 m_2\cos(\theta_2) - g\,l_1 m_1 \cos(\theta_1) - g\,l_1 m_1\cos(\theta_1) \tag{\%t2}$$

Lagrangian

(%i3) `ldisplay(L:T-V)$`

$$L = \frac{m_2\left(\left(l_2\sin(\theta_2)\left(\dot{\theta}_2\right)+l_1\sin(\theta_1)\left(\dot{\theta}_1\right)\right)^2+\left(l_2\cos(\theta_2)\left(\dot{\theta}_2\right)+l_1\cos(\theta_1)\left(\dot{\theta}_1\right)\right)^2\right)}{2} + g\,l_2 m_2\cos(\theta_2) + \frac{m_1\left({l_1}^2\sin(\theta_1)^2\left(\dot{\theta}_1\right)^2+{l_1}^2\cos(\theta_1)^2\left(\dot{\theta}_1\right)^2\right)}{2} + g\,l_1 m_2\cos(\theta_1) + g\,l_1 m_1\cos(\theta_1)$$

Momentum Conjugate

(%i4) `ldisplay(P_1:diff(L,'diff(`θ`_1,t)))$`

$$P_1 = (m_2(2l_1\sin(\theta_1)\left(l_2\sin(\theta_2)\left(\dot{\theta}_2\right)+l_1\sin(\theta_1)\left(\dot{\theta}_1\right)\right)+2l_1\cos(\theta_1)\left(l_2\cos(\theta_2)\left(\dot{\theta}_2\right)+l_1\cos(\theta_1)\left(\dot{\theta}_1\right)\right)))/2 + \frac{m_1\left(2{l_1}^2\sin(\theta_1)^2\left(\dot{\theta}_1\right)+2{l_1}^2\cos(\theta_1)^2\left(\dot{\theta}_1\right)\right)}{2}$$

(%i5) `ldisplay(P_2:diff(L,'diff(`θ`_2,t)))$`

$$P_2 = (m_2(2l_2\sin(\theta_2)\left(l_2\sin(\theta_2)\left(\dot{\theta}_2\right)+l_1\sin(\theta_1)\left(\dot{\theta}_1\right)\right)+2l_2\cos(\theta_2)\left(l_2\cos(\theta_2)\left(\dot{\theta}_2\right)+l_1\cos(\theta_1)\left(\dot{\theta}_1\right)\right)))/2$$

Generalized Forces

(%i6) `ldisplay(F_1:diff(L,θ_1))$`

$$F_1 = (m_2(2l_1\cos(\theta_1)\left(\dot{\theta}_1\right)\left(l_2\sin(\theta_2)\left(\dot{\theta}_2\right)+l_1\sin(\theta_1)\left(\dot{\theta}_1\right)\right) - 2l_1\sin(\theta_1)\left(\dot{\theta}_1\right)\left(l_2\cos(\theta_2)\left(\dot{\theta}_2\right)+l_1\cos(\theta_1)\left(\dot{\theta}_1\right)\right)))/2 - g\,l_1 m_2\sin(\theta_1) - g\,l_1 m_1\sin(\theta_1)$$

(%i7) `ldisplay(F_2:diff(L,θ_2))$`

$$F_2 = (m_2(2l_2\cos(\theta_2)\left(\dot{\theta}_2\right)\left(l_2\sin(\theta_2)\left(\dot{\theta}_2\right)+l_1\sin(\theta_1)\left(\dot{\theta}_1\right)\right) - 2l_2\sin(\theta_2)\left(\dot{\theta}_2\right)\left(l_2\cos(\theta_2)\left(\dot{\theta}_2\right)+l_1\cos(\theta_1)\left(\dot{\theta}_1\right)\right)))/2 - g\,l_2 m_2\sin(\theta_2)$$

Euler-Lagrange Equation

(%i8) `aa:el(L,ζ,t)$`

(%i11) `bb:ev(aa,eval,diff)$`

(%i12) `bb[1]:subst([k[0]=-E],-bb[1])$`

Solve for second derivative of coordinates

(%i13) `linsol:linsolve(part(bb,[2,3]),diff(ζ,t,2))$`

(%i14) `map(ldisp,linsol:factor(trigreduce(linsol)))$`

$$\ddot{\theta}_1 = -(g\,m_2\sin(2\theta_2-\theta_1) + l_1 m_2\left(\dot{\theta}_1\right)^2\sin(2(\theta_2-\theta_1)) + 2l_2 m_2\left(\dot{\theta}_2\right)^2\sin(\theta_2-\theta_1) - g\,m_2\sin(\theta_1) - 2g\,m_1\sin(\theta_1))/(l_1(m_2\cos(2(\theta_2-\theta_1)) - m_2 - 2m_1))$$

$$\ddot{\theta}_2 = (l_2 m_2\left(\dot{\theta}_2\right)^2\sin(2(\theta_2-\theta_1)) + 2l_1 m_2\left(\dot{\theta}_1\right)^2\sin(\theta_2-\theta_1) + 2l_1 m_1\left(\dot{\theta}_1\right)^2\sin(\theta_2-\theta_1) + g\,m_2\sin(\theta_2-2\theta_1) + g\,m_1\sin(\theta_2-2\theta_1) + g\,m_2\sin(\theta_2) + g\,m_1\sin(\theta_2))/(l_2(m_2\cos(2(\theta_2-\theta_1)) - m_2 - 2m_1))$$

3 Hamiltonian Formalism

(%i16)
```
kill(labels)$
```

Legendre Transformation

(%i1)
```
Legendre:linsolve([p_1=P_1,p_2=P_2],['diff(θ_1,t),'diff(θ_2,t)])$
```

Hamiltonian

(%i2)
```
H:ev(p_1*'diff(θ_1,t)+p_2*'diff(θ_2,t)-L,Legendre)$
```

Equations of Motion

(%i3)
```
Hq:makelist(Hq[i],i,1,2*dim)$
```

(%i7)
```
Hq[1]:'diff(θ_1,t)=diff(H,p_1)$
Hq[2]:'diff(θ_2,t)=diff(H,p_2)$
Hq[3]:'diff(p_1,t)=-diff(H,θ_1)$
Hq[4]:'diff(p_2,t)=-diff(H,θ_2)$
```

4 Reduce Order

(%i8)
```
kill(labels)$
```

(%i2)
```
ξ:[Θ_1,Θ_2]$
depends(ξ,t)$
```

(%i4)
```
gradef(θ_1,t,Θ_1)$
gradef(θ_2,t,Θ_2)$
```

Euler-Lagrange Equations

(%i5)
```
aa:el(L,ζ,t)$
```

(%i8)
```
bb:ev(aa,eval,diff)$
```

(%i9)
```
bb[1]:subst([k[0]=-E],-bb[1])$
```

Solve for second derivative of coordinates

(%i10)
```
linsol:linsolve(part(bb,[2,3]),diff(ζ,t,2))$
```

(%i11)
```
map(ldisp,linsol:factor(trigreduce(linsol)))$
```

$$\dot{\Theta}_1 = -(g\, m_2 \sin(2\theta_2 - \theta_1) + l_1 m_2 {\Theta_1}^2 \sin(2(\theta_2 - \theta_1)) + 2 l_2 m_2 {\Theta_2}^2 \sin(\theta_2 - \theta_1) - g\, m_2 \sin(\theta_1) - 2g\, m_1 \sin(\theta_1))/(l_1 (m_2 \cos(2(\theta_2 - \theta_1)) - m_2 - 2m_1))$$

$$\dot{\Theta}_2 = (l_2 m_2 {\Theta_2}^2 \sin(2(\theta_2 - \theta_1)) + 2 l_1 m_2 {\Theta_1}^2 \sin(\theta_2 - \theta_1) + 2 l_1 m_1 {\Theta_1}^2 \sin(\theta_2 - \theta_1) + g\, m_2 \sin(\theta_2 - 2\theta_1) + g\, m_1 \sin(\theta_2 - 2\theta_1) + g\, m_2 \sin(\theta_2) + g\, m_1 \sin(\theta_2))/(l_2 (m_2 \cos(2(\theta_2 - \theta_1)) - m_2 - 2m_1))$$

Numerical solution (Lagrangian)

(%i13) `kill(labels)$`

(%i7)
```
funcs:[θ_1,θ_2,Θ_1,Θ_2]$ldisplay(funcs)$
initial:[π/7,π/9,0,0]$ldisplay(initial)$
odes:append(ξ,map('rhs,linsol))$
interval:[t,0,τ]$ldisplay(interval)$
```

$$funcs = [\theta_1, \theta_2, \Theta_1, \Theta_2] \tag{\%t2}$$

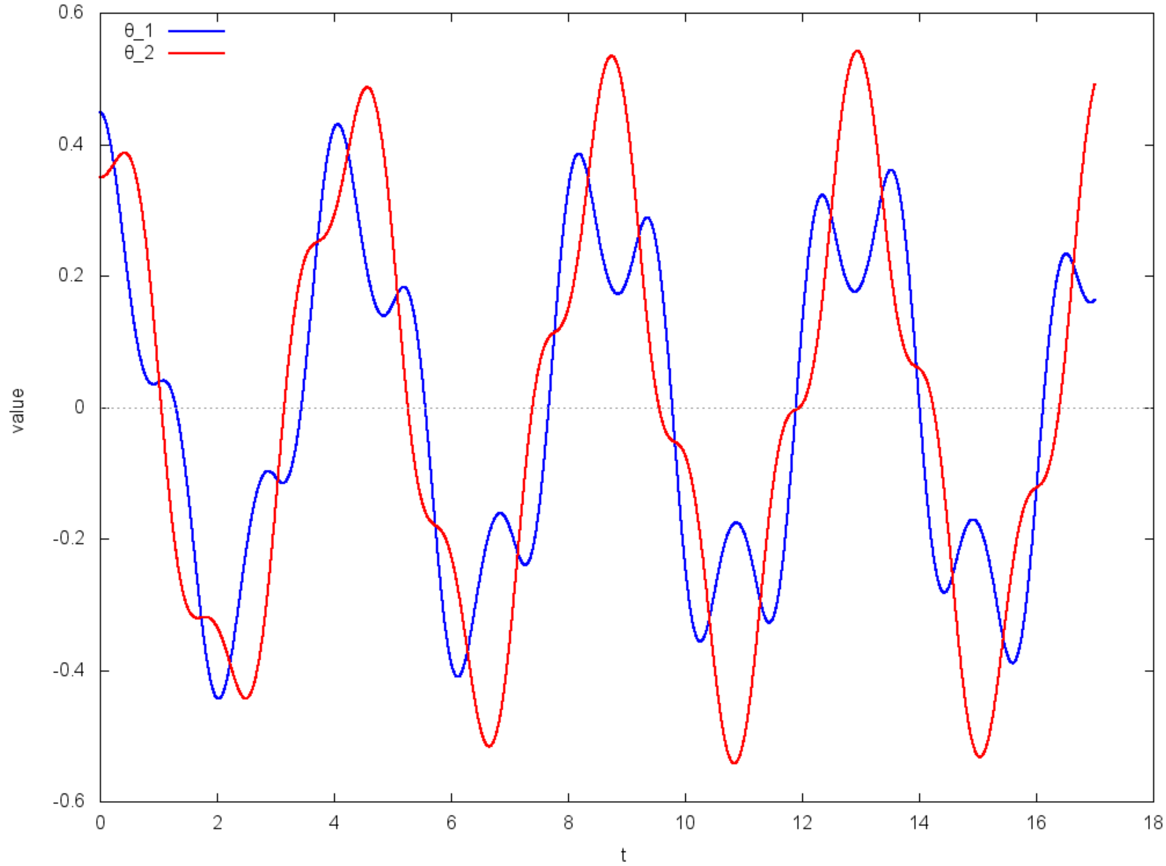
$$initial = \left[\frac{\pi}{7}, \frac{\pi}{9}, 0, 0\right] \tag{\%t4}$$

$$interval = [t, 0, 17] \tag{\%t7}$$

(%i8) `rksol:rkf45(odes,funcs,initial,interval, absolute_tolerance=1E-10,report=true),params$`

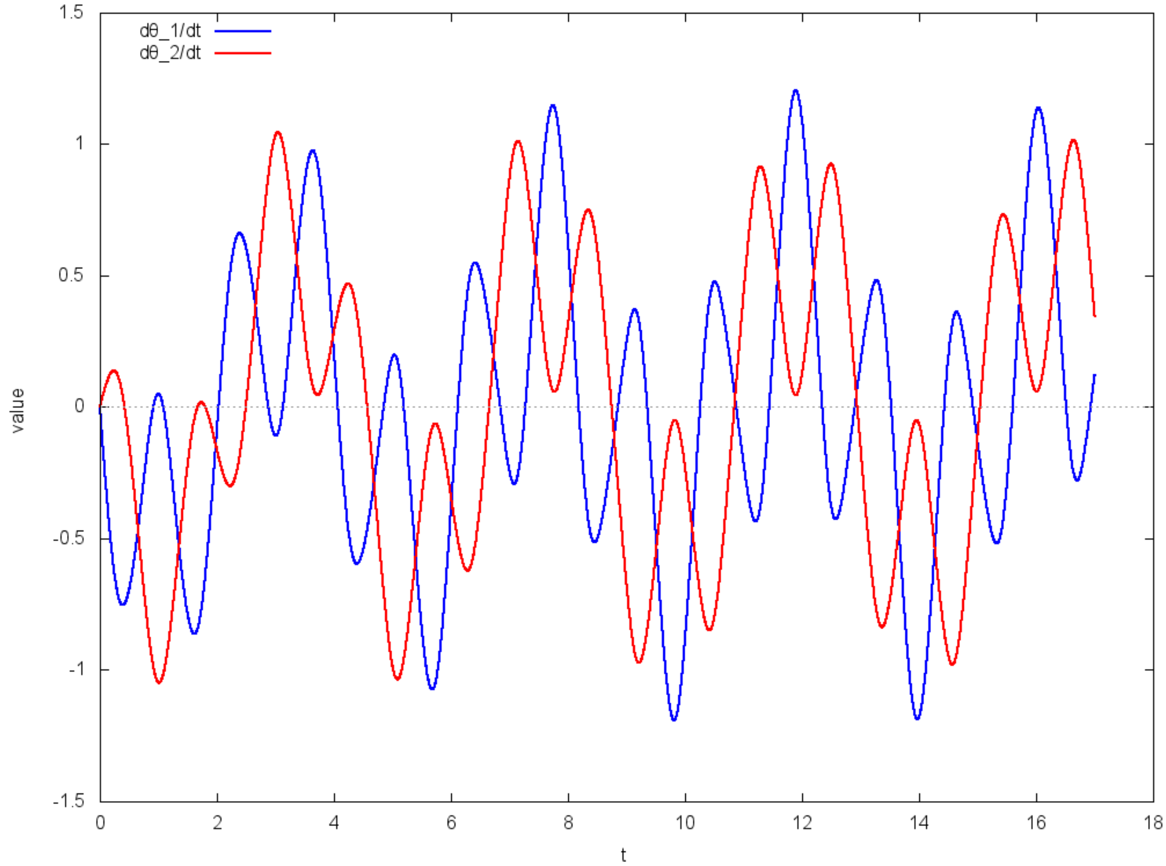
Info: rkf45:
Integration points selected:6019
Total number of iterations:6020
Bad steps corrected:2
Minimum estimated error:$4.1407 10^{-14}$
Maximum estimated error:$5.6824 10^{-11}$
Minimum integration step taken:$4.9929 10^{-4}$
Maximum integration step taken:0.0040347

(%i9) `wxplot2d([[discrete,map(lambda([u],part(u,[1,2])),rksol)], [discrete,map(lambda([u],part(u,[1,3]` `[style,[lines,2]],[xlabel,"t"],[ylabel,"value"], [legend,"`θ`_1","`θ`_2"],[gnuplot_preamble,"set` `key top left"])$`



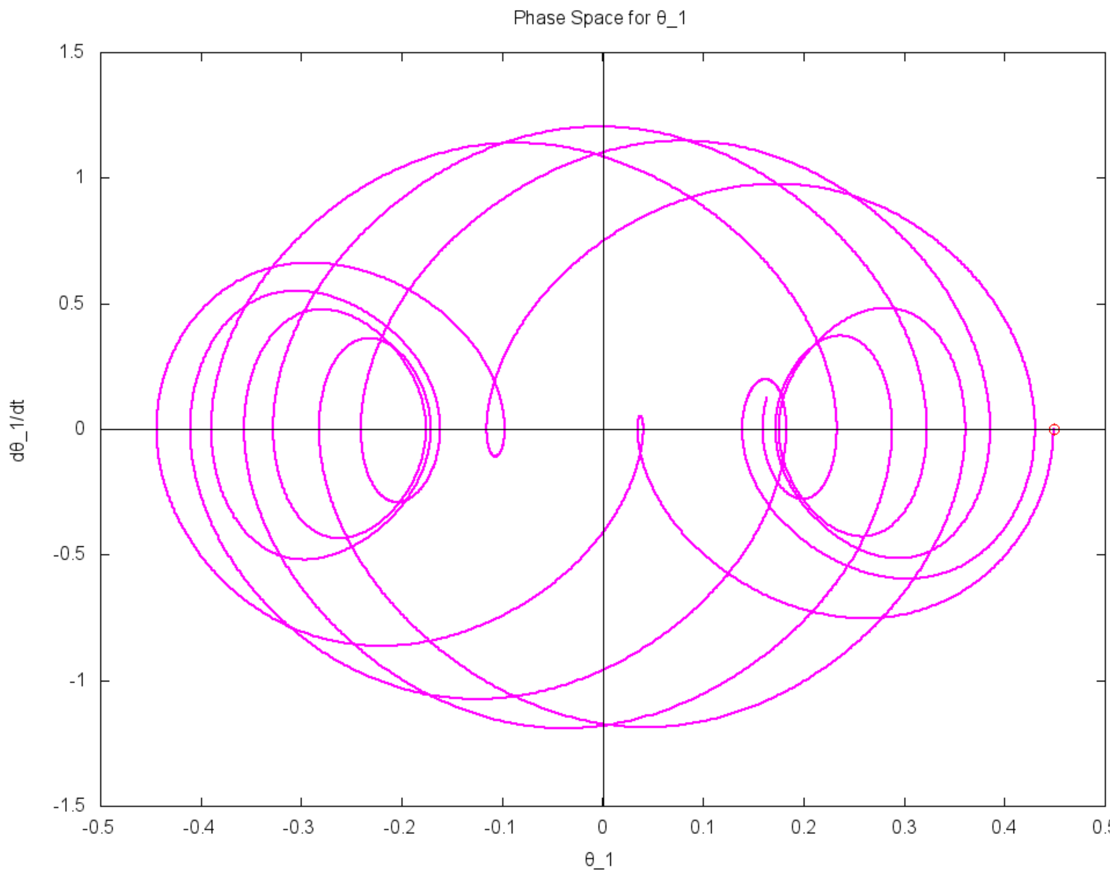
(%t9)

(%i10) wxplot2d([[discrete,map(lambda([u],part(u,[1,4])),rksol)], [discrete,map(lambda([u],part(u,[1,5]
[style,[lines,2]],[xlabel,"t"],[ylabel,"value"], [legend,"dθ_1/dt","dθ_2/dt"],[gnuplot_preamble,"
key top left"])$



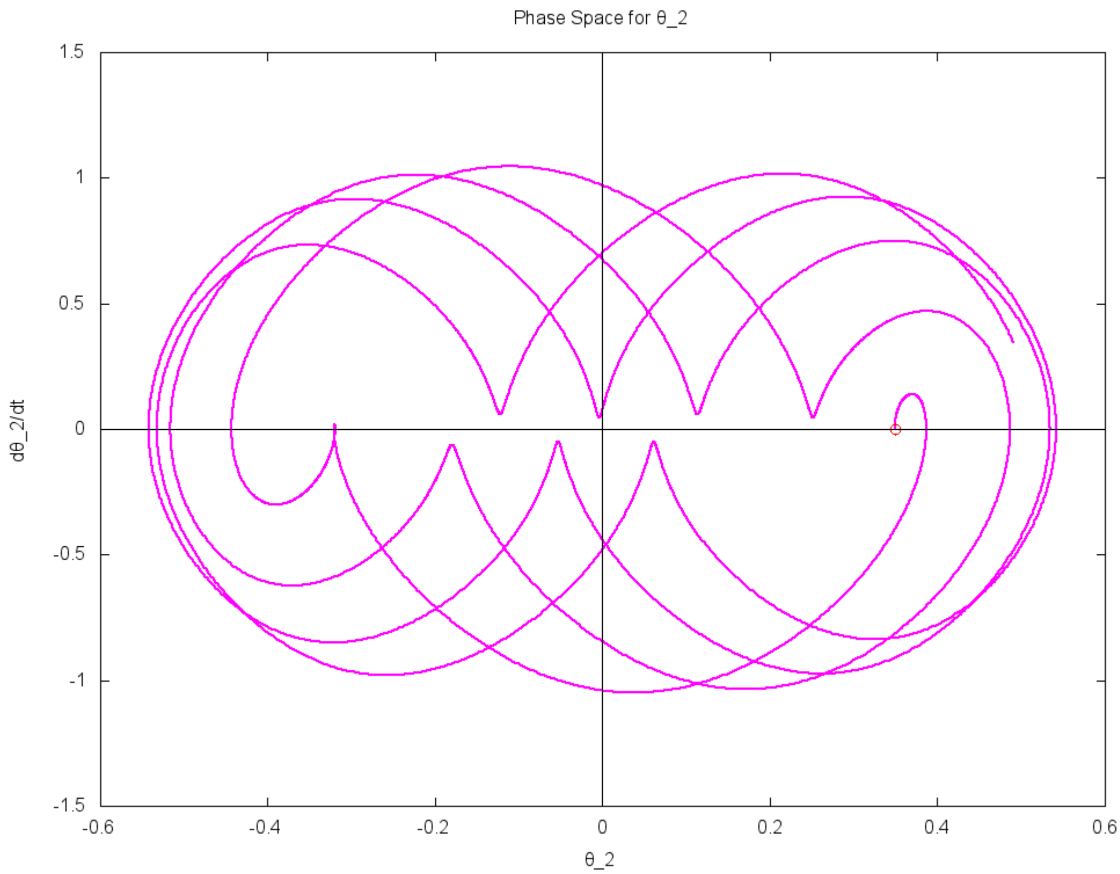
(%t10)

(%i11) `wxplot2d([[discrete,map(lambda([u],part(u,[2,4])),rksol)], [discrete,[part(initial,[1,3])]]],[ax`
`[title,"Phase Space for θ_1"],[point_type,circle], [style,[lines,2],[points,3]],[color,magenta,re`
`[xlabel,"θ_1"],[ylabel,"dθ_1/dt"],[legend,false])$`



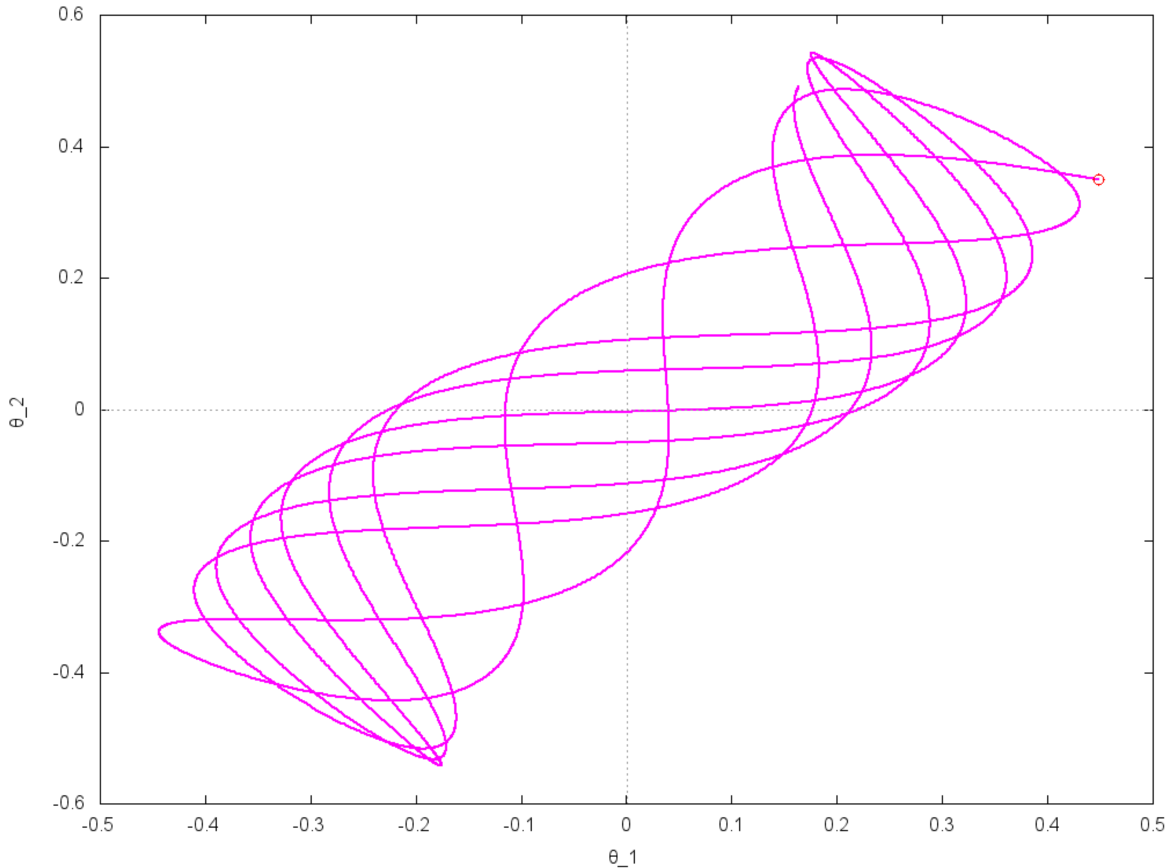
(%t11)

(%i12) wxplot2d([[discrete,map(lambda([u],part(u,[3,5])),rksol)], [discrete,[part(initial,[2,4])]]],[axe
[title,"Phase Space for θ_2"],[point_type,circle], [style,[lines,2],[points,3]],[color,magenta,re
[xlabel,"θ_2"],[ylabel,"dθ_2/dt"],[legend,false])$



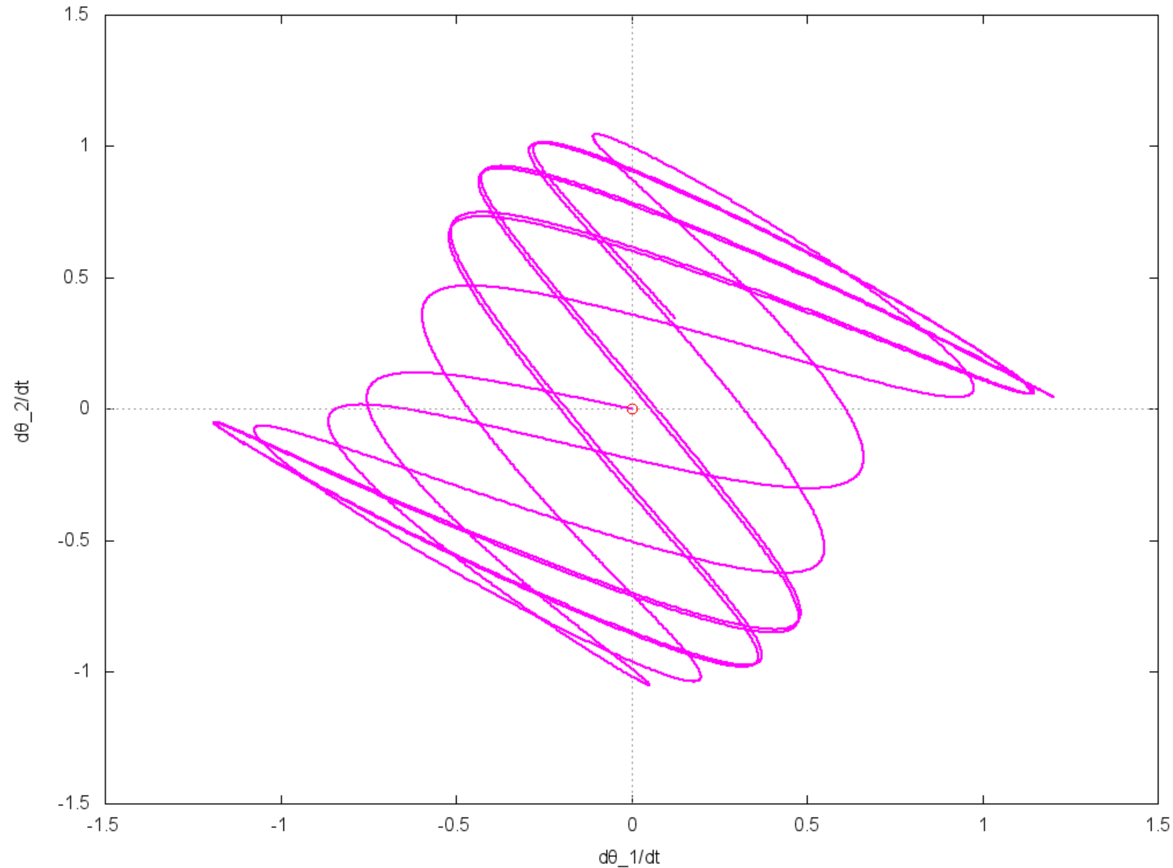
(%t12)

(%i13) wxplot2d([[discrete,map(lambda([u],part(u,[2,3])),rksol)], [discrete,[part(initial,[1,2])]]],[po
[style,[lines,2],[points,3]],[color,magenta,red], [xlabel,"θ_1"],[ylabel,"θ_2"],[legend,false])$



(%t13)

(%i14) wxplot2d([[discrete,map(lambda([u],part(u,[4,5])),rksol)], [discrete,[part(initial,[3,4])]]],[poi
[style,[lines,2],[points,3]],[color,magenta,red], [xlabel,"dθ_1/dt"],[ylabel,"dθ_2/dt"],[legend,f



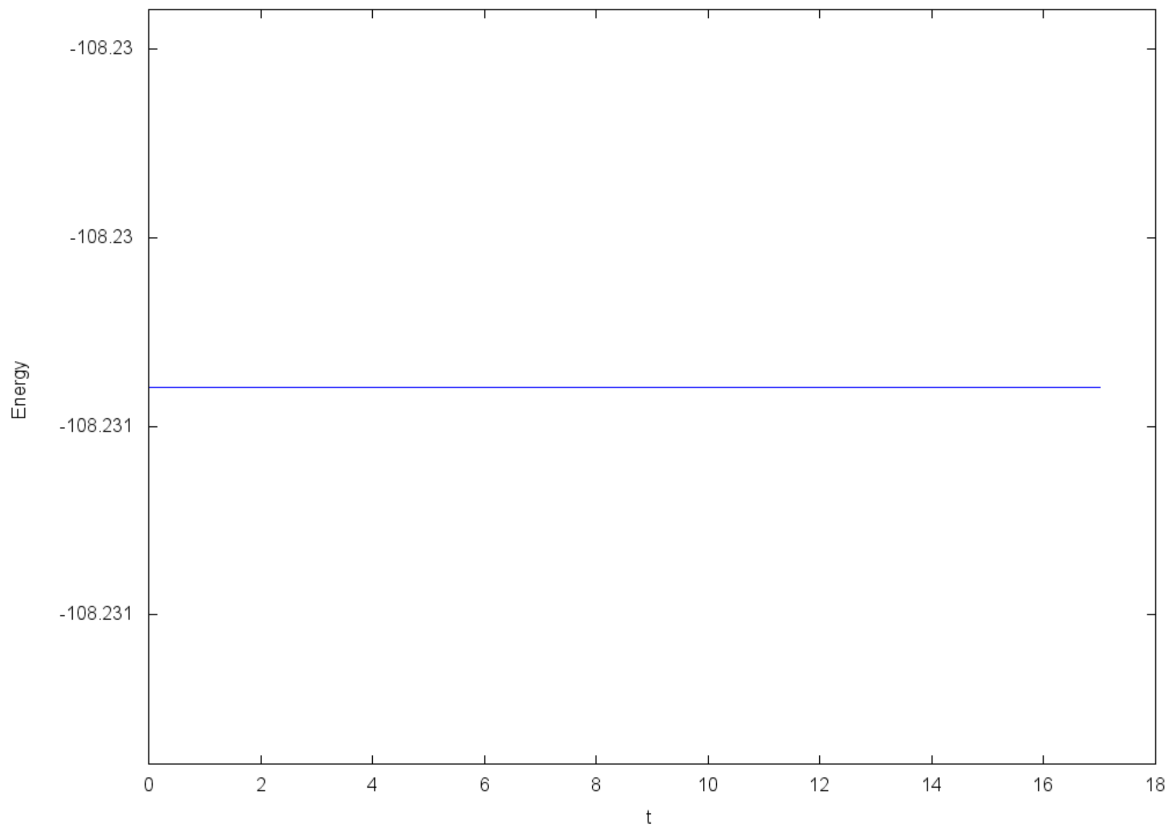
(%t14)

Check Conservation of Energy using the Numerical Data

(%i15) `W:bb[1],map("=",funcs,initial),params,numer,eval;`

$$-108.23 = E \tag{W}$$

(%i16) `wxplot2d([discrete,makelist([first(rkline), ev(lhs(bb[1]),map("=",funcs,rest(rkline)),params)],rk`
`[xlabel,"t"],[ylabel,"Energy"],[y,lhs(W)-0.001,lhs(W)+0.001])$`



(%t16)

Numerical solution (Hamiltonian)

(%i17)
```
kill(labels)$
```

(%i7)
```
funcs:[θ_1,θ_2,p_1,p_2]$ldisplay(funcs)$
initial:[π/7,π/9,0,0]$ldisplay(initial)$
odes:map(rhs,Hq)$
interval:[t,0,τ]$ldisplay(interval)$
```

$$funcs = [\theta_1, \theta_2, p_1, p_2] \tag{\%t2}$$

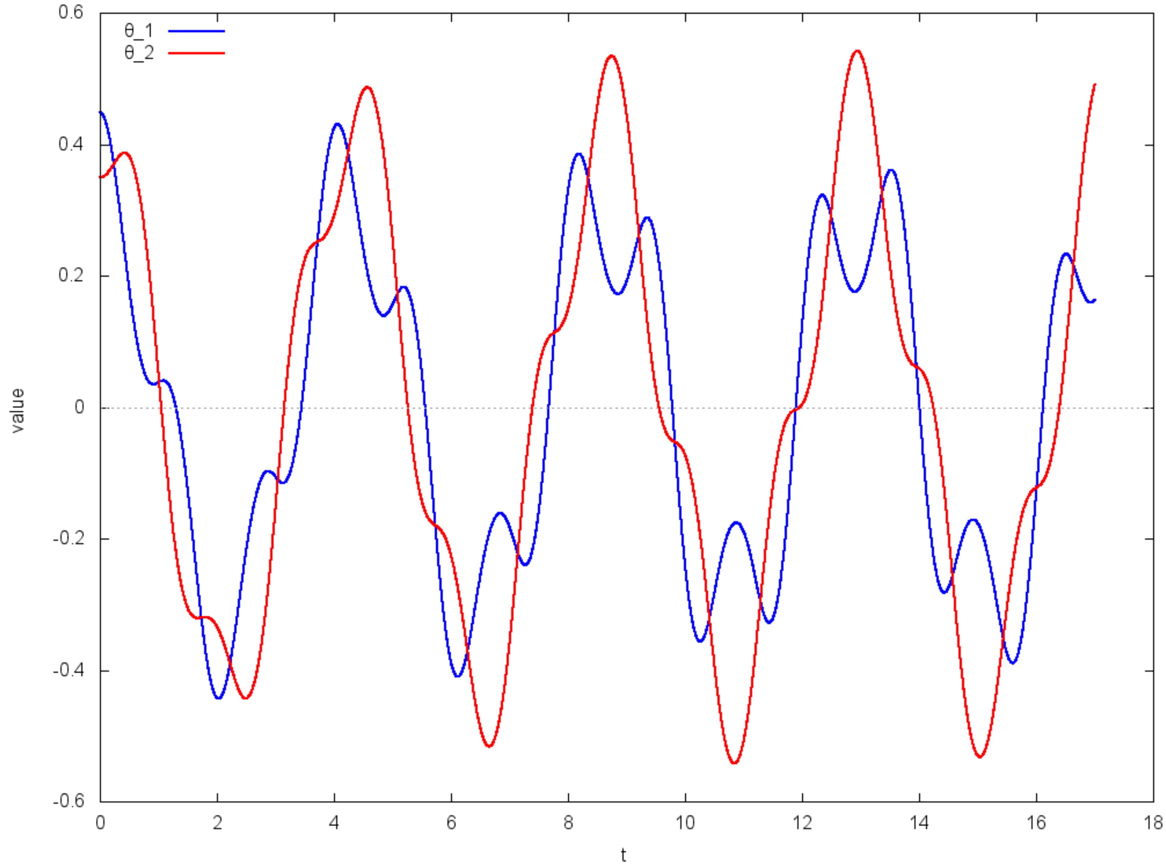
$$initial = \left[\frac{\pi}{7}, \frac{\pi}{9}, 0, 0\right] \tag{\%t4}$$

$$interval = [t, 0, 17] \tag{\%t7}$$

(%i8)
```
rksol:rkf45(odes,funcs,initial,interval, absolute_tolerance=1E-10,report=true),params$
```

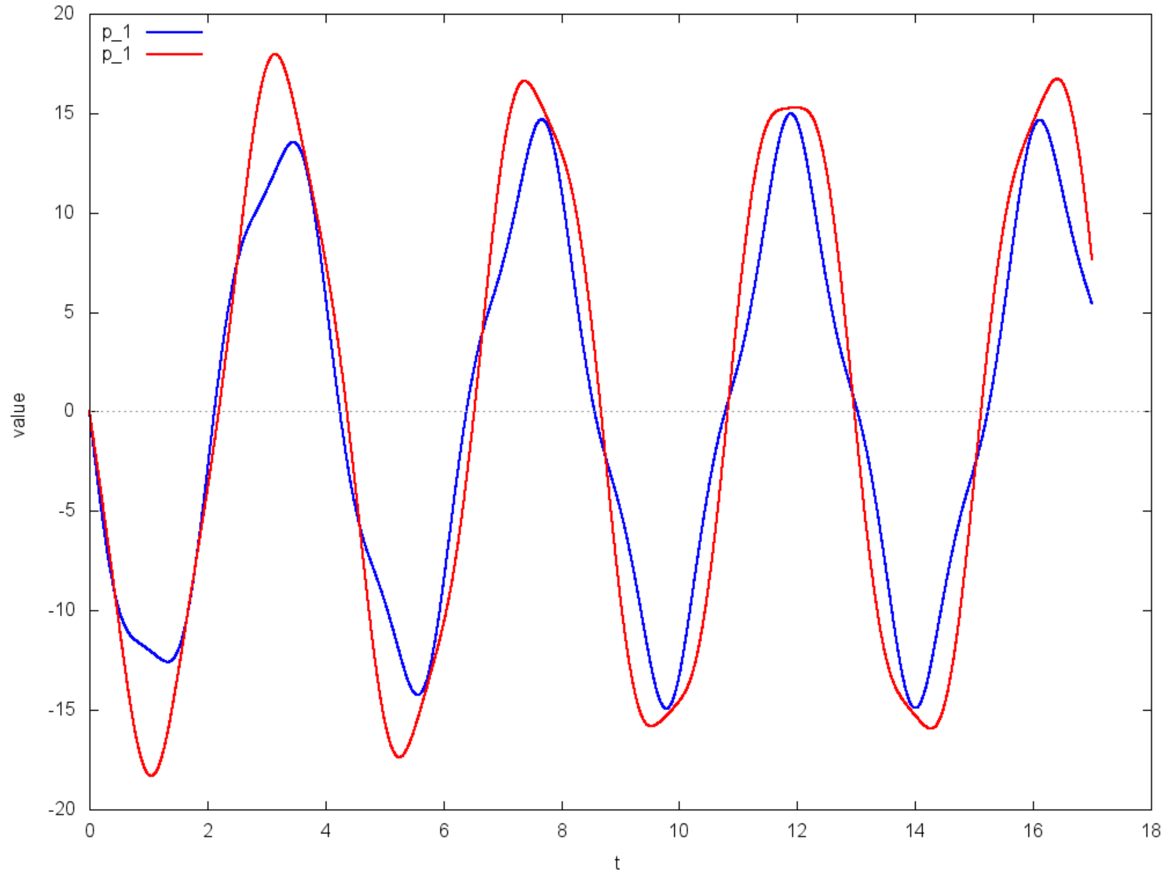
Info: rkf45:
Integration points selected:8049
Total number of iterations:8056
Bad steps corrected:8
Minimum estimated error:$2.3003 10^{-11}$
Maximum estimated error:$9.4399 10^{-11}$
Minimum integration step taken:0.0018079
Maximum integration step taken:0.0044723

(%i9)
```
wxplot2d([[discrete,map(lambda([u],part(u,[1,2])),rksol)], [discrete,map(lambda([u],part(u,[1,3]
[style,[lines,2]],[xlabel,"t"],[ylabel,"value"], [legend,"θ_1","θ_2"],[gnuplot_preamble,"set
key top left"])$
```



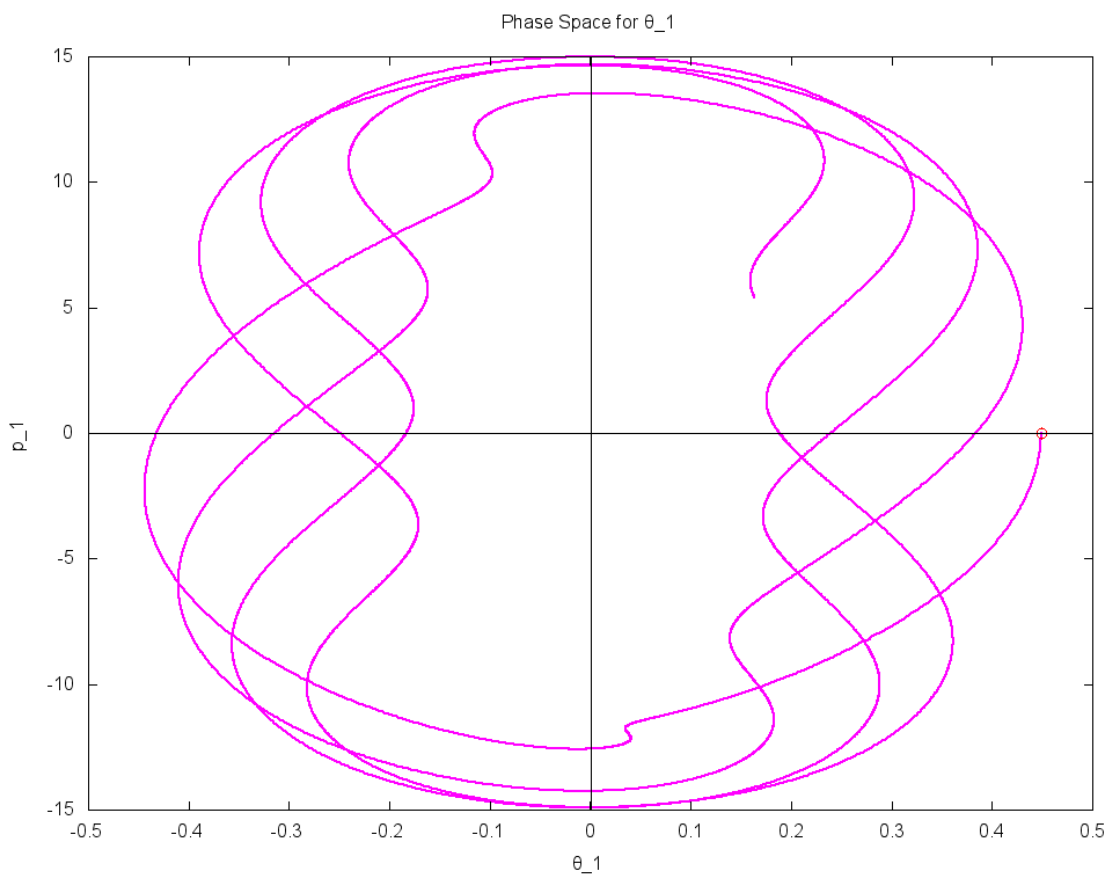
(%t9)

(%i10) wxplot2d([[discrete,map(lambda([u],part(u,[1,4])),rksol)], [discrete,map(lambda([u],part(u,[1,5]
[style,[lines,2]],[xlabel,"t"],[ylabel,"value"], [legend,"p_1","p_2"],[gnuplot_preamble,"set
key top left"])$



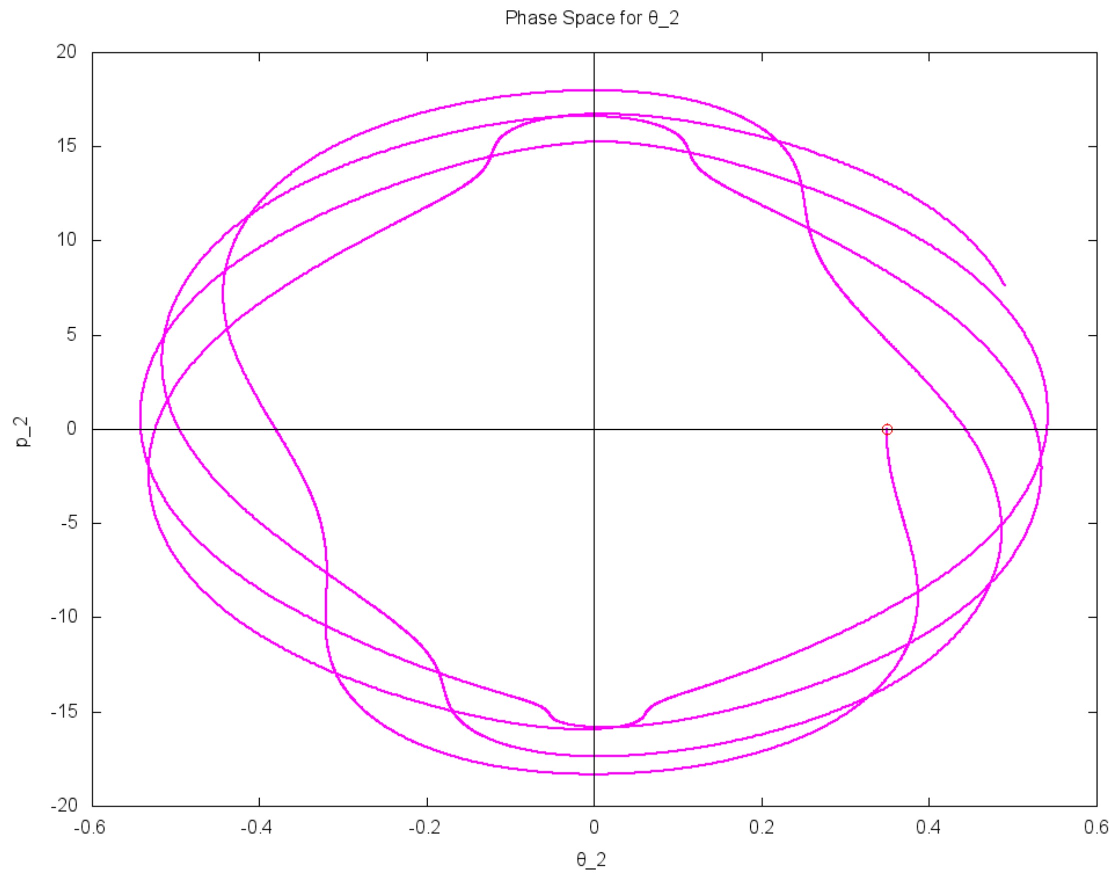
(%t10)

(%i11) wxplot2d([[discrete,map(lambda([u],part(u,[2,4])),rksol)], [discrete,[part(initial,[1,3])]]],[axe
[title,"Phase Space for θ_1"],[point_type,circle], [style,[lines,2],[points,3]],[color,magenta,re
[xlabel,"θ_1"],[ylabel,"p_1"],[legend,false])$



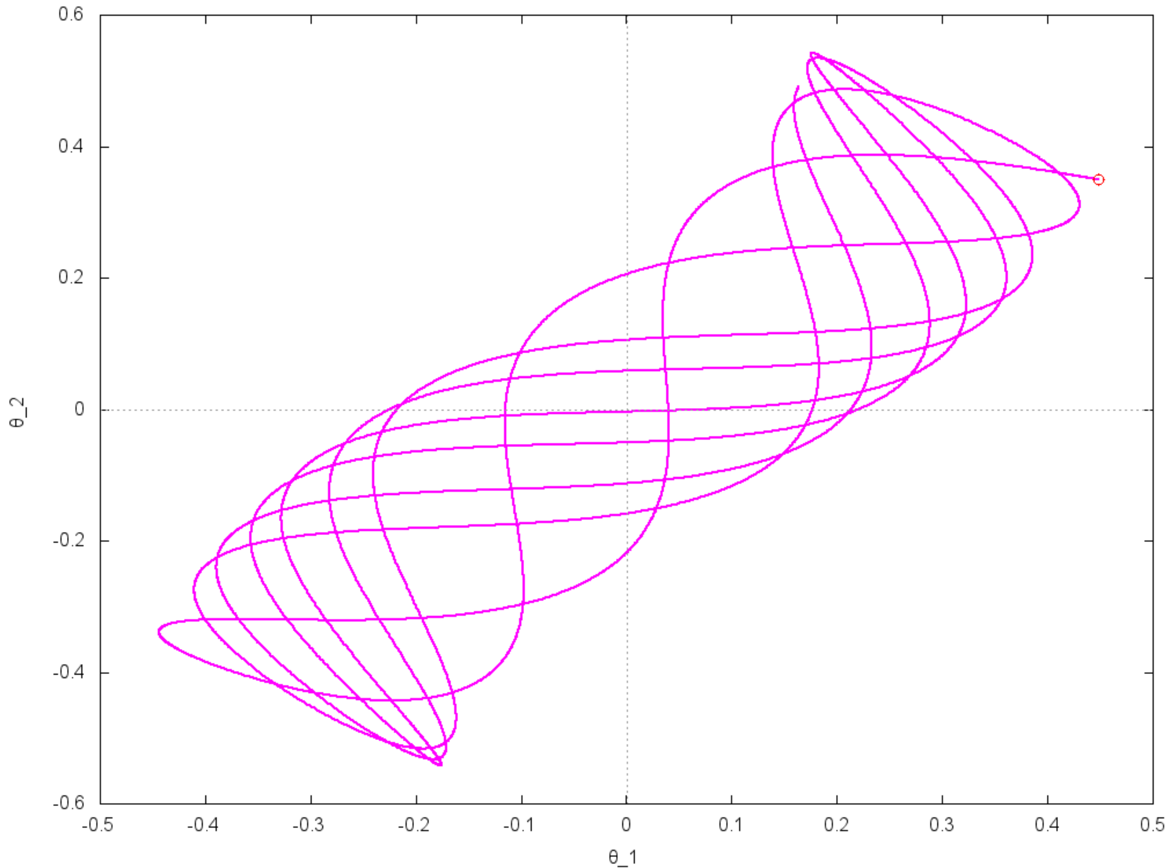
(%t11)

(%i12) wxplot2d([[discrete,map(lambda([u],part(u,[3,5])),rksol)], [discrete,[part(initial,[2,4])]]],[ax
[title,"Phase Space for θ_2"],[point_type,circle], [style,[lines,2],[points,3]],[color,magenta,re
[xlabel,"θ_2"],[ylabel,"p_2"],[legend,false])$



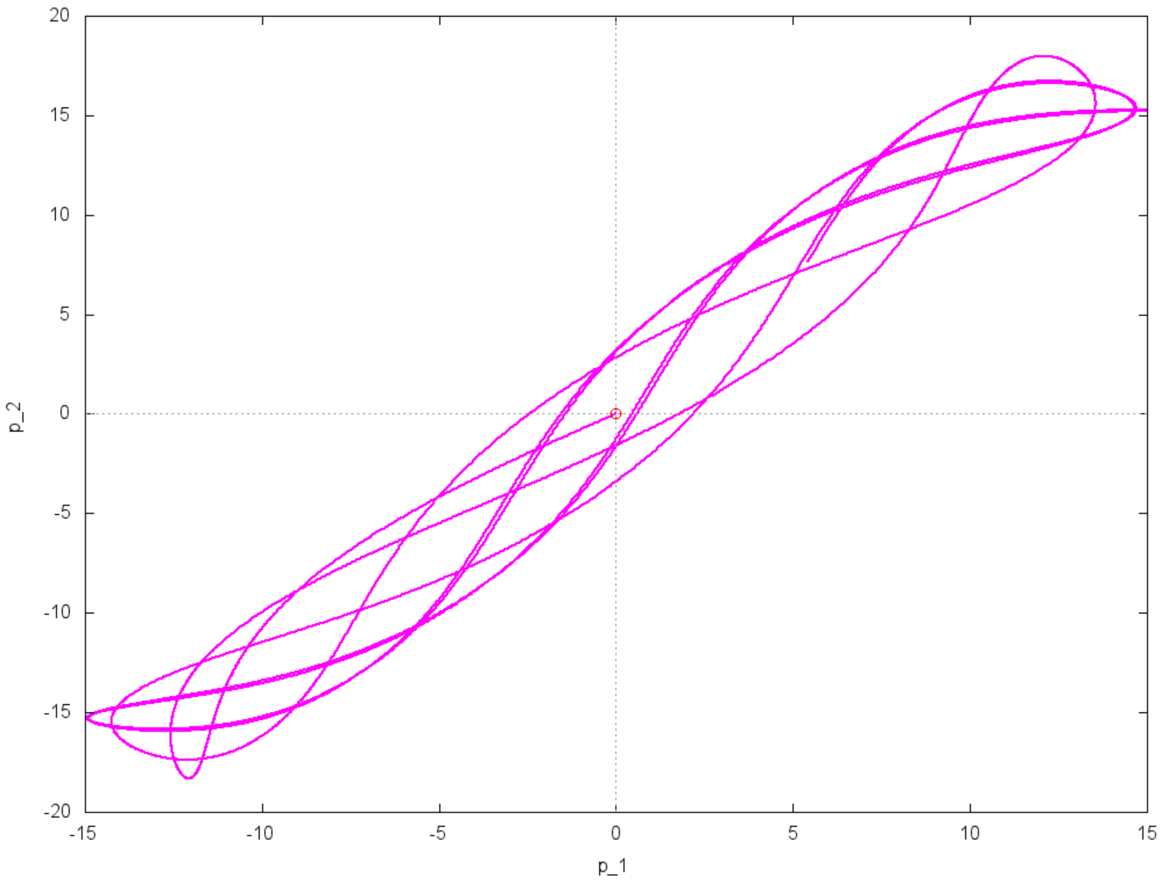
(%t12)

(%i13) wxplot2d([[discrete,map(lambda([u],part(u,[2,3])),rksol)], [discrete,[part(initial,[1,2])]]],[poi
[style,[lines,2],[points,3]],[color,magenta,red], [xlabel,"θ_1"],[ylabel,"θ_2"],[legend,false])$



(%t13)

(%i14) wxplot2d([[discrete,map(lambda([u],part(u,[4,5])),rksol)], [discrete,[part(initial,[3,4])]]],[po
[style,[lines,2],[points,3]],[color,magenta,red], [xlabel,"p_1"],[ylabel,"p_2"],[legend,false])$



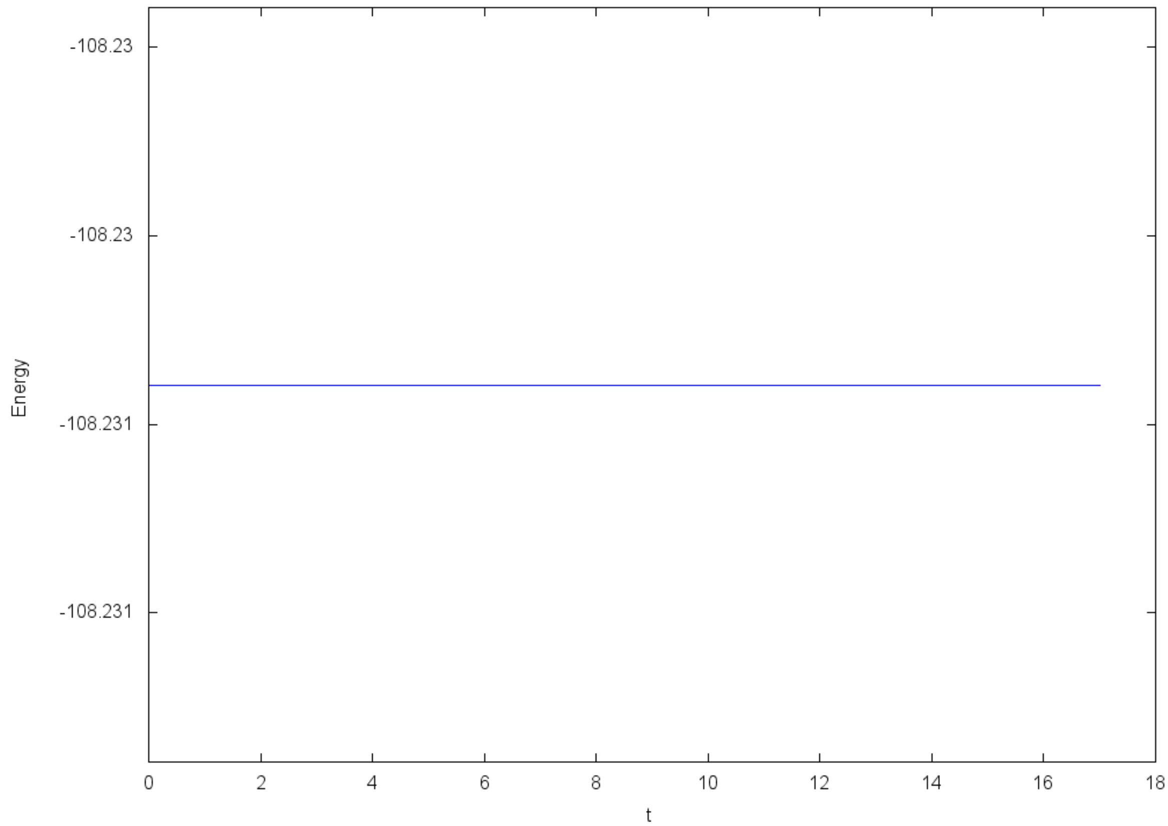
(%t14)

Check Conservation of Energy using the Numerical Data

(%i15) `W:H,map("=",funcs,initial),params,numer,eval;`

$$-108.23 \qquad \text{(W)}$$

(%i16) `wxplot2d([discrete,makelist([first(rkline), ev(H,map("=",funcs,rest(rkline)))],rkline,rksol)], [xlabel,"t"],[ylabel,"Energy"],[y,W-0.001,W+0.001]),params$`



(%t16)

5 Symplectic Integrator

(%i17) kill(labels)$

(%i1) file_search_maxima:cons(sconcat("D:/USERS/wxMaxima/Barton/symplectic_ode/≠≠≠.lisp,mac,mc"),file_

(%i2) load("symplectic_ode.fasl")$

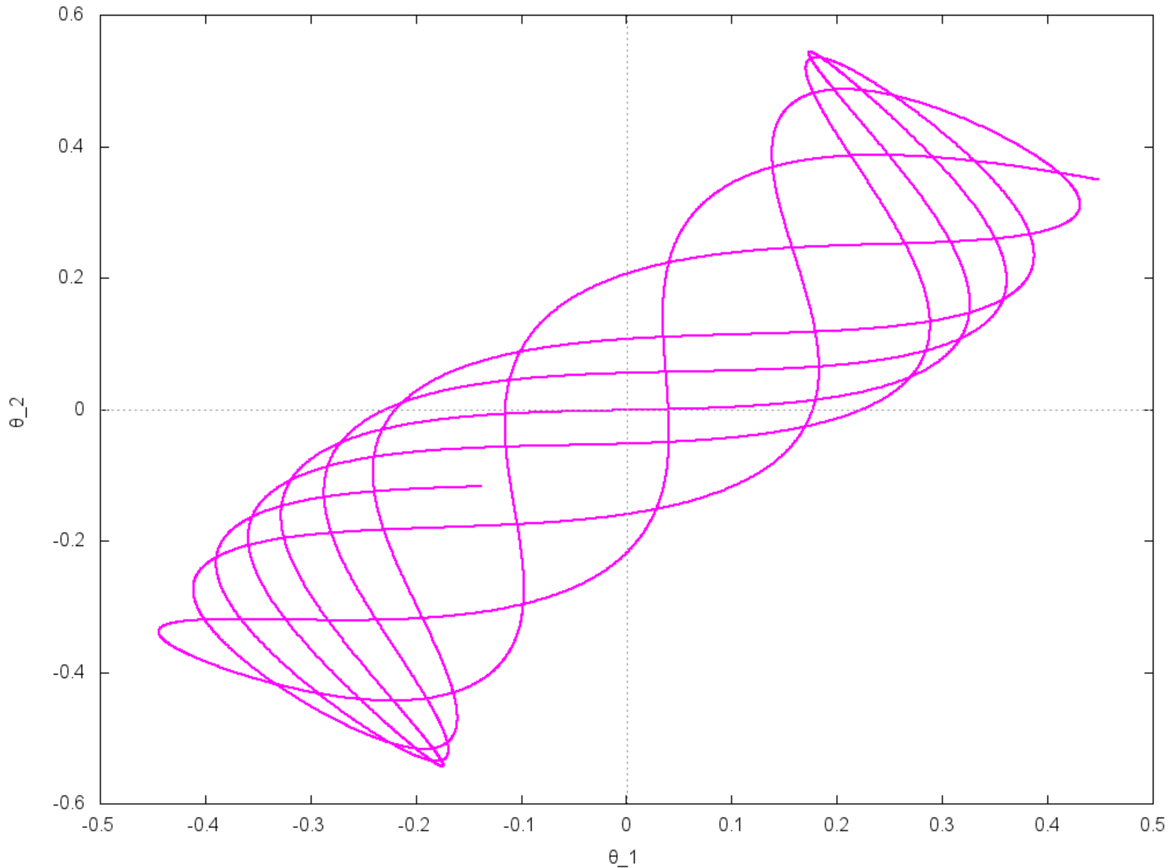
Numerical solution (Symplectic)

(%i5) δ:0.002$N:8000$$\delta$*N;

$$16.0 \quad (\%o5)$$

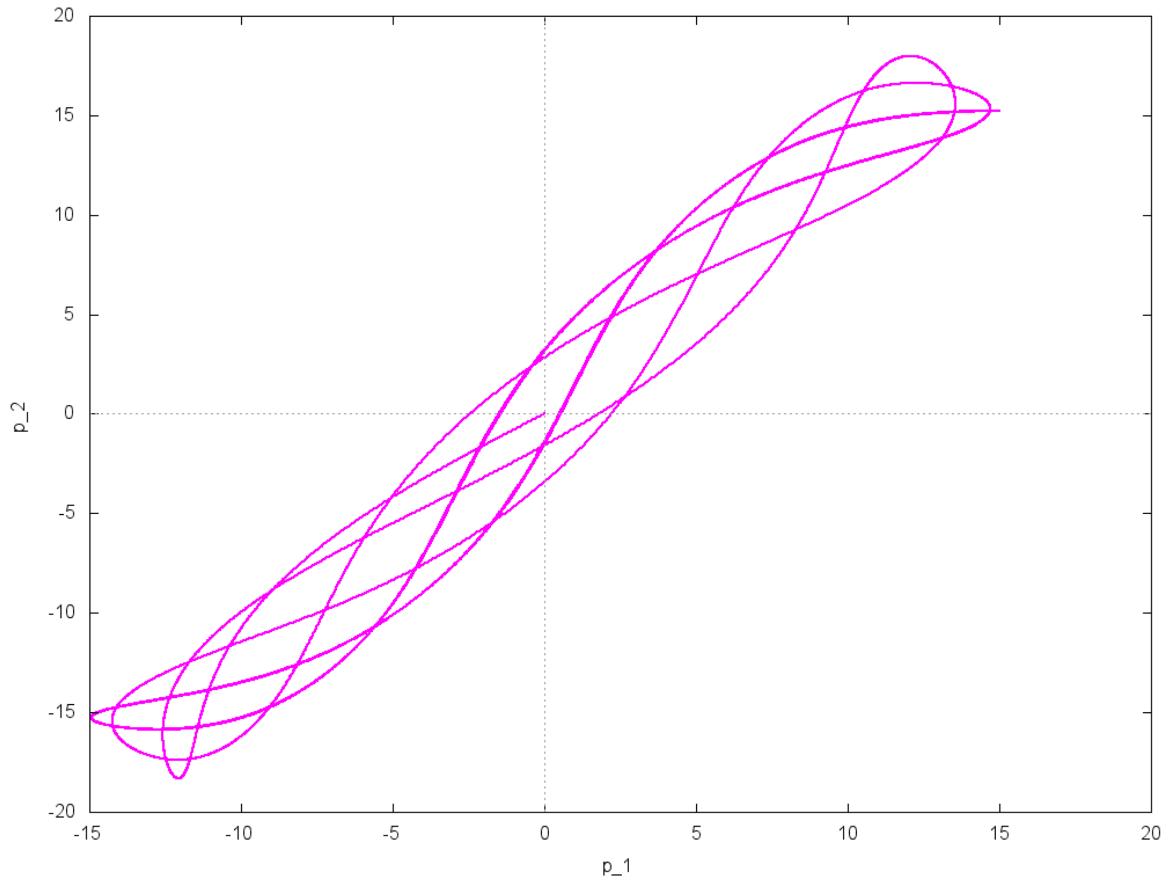
(%i6) [pp,qq]:symplectic_ode(ev(H,params),[p_1,p_2],[θ_1,θ_2], part(initial,[3,4]),part(initial,[1,2]), δ,N,'symplectic_euler,'float)$

(%i7) wxplot2d(cons('discrete,[qq]), [style,[lines,2]],[color,magenta],
[xlabel,"θ_1"],[ylabel,"θ_2"])$



(%t7)

(%i8) wxplot2d(cons('discrete,[pp]), [style,[lines,2]],[color,magenta],
[xlabel,"p_1"],[ylabel,"p_2"])$

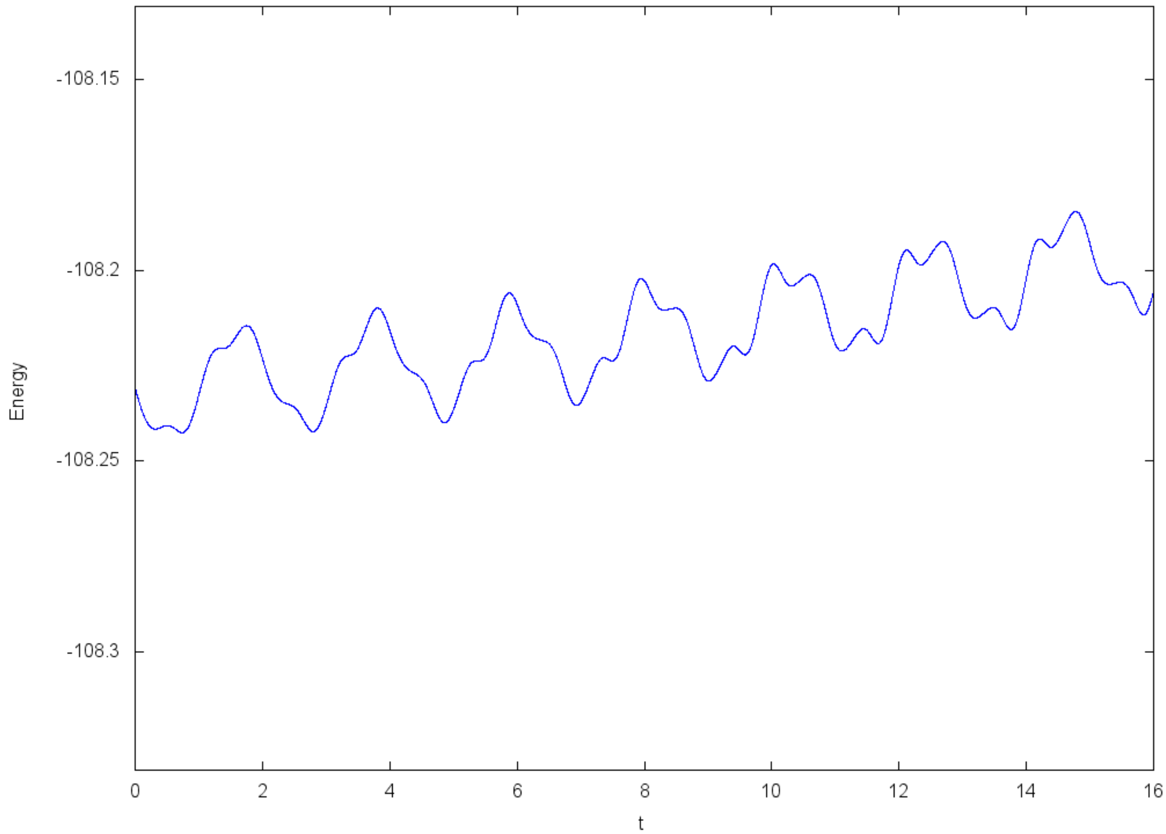


(%t8)

Check Conservation of Energy using the Numerical Data

(%i9)
```
synsol:makelist([δ*(i-1),qq[i][1],qq[i][2],pp[i][1],pp[i][2]],i,1,length(pp))$
```

(%i10)
```
wxplot2d([discrete,makelist([first(synline), ev(H,map("=",funcs,rest(synline)),params)],synline,s
[xlabel,"t"],[ylabel,"Energy"],[y,W-0.1,W+0.1])$
```



(%t10)

6 Graphics

(%i11) `kill(labels)$`

(%i1) `wxanimate_framerate:60$`

(%i2) `wxanimate_autoplay:false$`

(%i3) `rksol:rk(odes,funcs,initial,[t,0,τ,0.1]),params$`

(%i4)
```
set_draw_defaults(proportional_axes = xy, delay = 1, xtics = 1, ytics = 1,
xrange=[-2,2],yrange=[-5,0])$
```

Create animated GIF file

(%i5)
```
draw(terminal = 'animated_gif, file_name = "Double Pendulum", makelist(gr2d(
color = red, point_type = filled_circle, point_size = 2, points_joined = true,
line_width = 2, key = sconcat("t=",float(t)/10," s"), points([[0.0,0.0],
[l_1*sin(rksol[t][2]),-l_1*cos(rksol[t][2])], [l_1*sin(rksol[t][2]) +l_2*sin(rksol[t][3]),
-l_1*cos(rksol[t][2]) -l_2*cos(rksol[t][3])]])), t,1,length(rksol))),params$
```

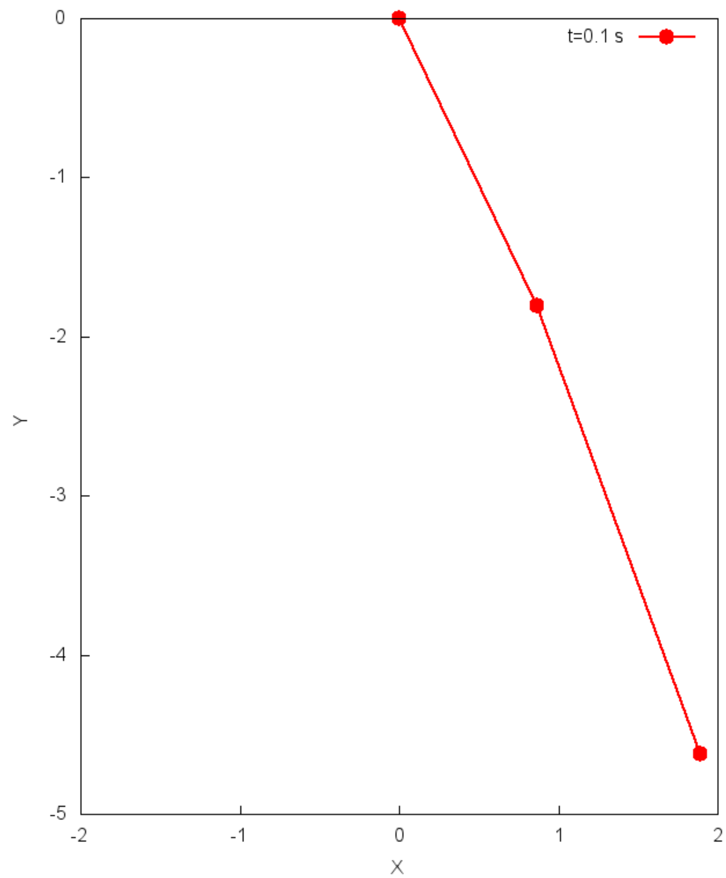
(%i6) `time(%);`

$$[0.015] \tag{\%o6}$$

(%i7) `wxanimate_framerate:30$`

(%i9)
```
print("Click the figure to start animation")$ with_slider_draw( t,makelist(i,i,1,length(rksol)),
color = red, point_type = filled_circle, point_size = 2, points_joined = true,
line_width = 2, key = sconcat("t=",float(t)/10," s"), points([[0.0,0.0],
[l_1*sin(rksol[t][2]),-l_1*cos(rksol[t][2])], [l_1*sin(rksol[t][2]) +l_2*sin(rksol[t][3]),
-l_1*cos(rksol[t][2]) -l_2*cos(rksol[t][3])]])),params$
```

Click the figure to start animation



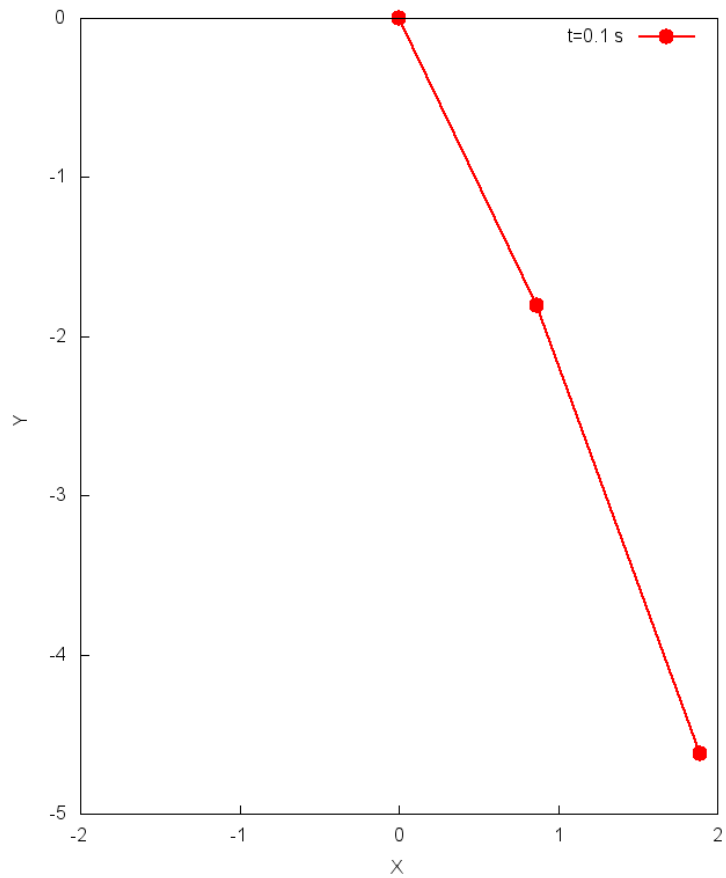
(%t9)

(%i10) `time(%);`

$$[0.485] \qquad (\%o10)$$

(%i12) `print("Click the figure to start animation")$ wxanimate_draw(t,length(rksol), color = red, point_type = filled_circle, point_size = 2, points_joined = true, line_width = 2, key = sconcat("t=",float(t)/10," s"), points([[0.0,0.0], [l_1*sin(rksol[t][2]),-l_1*cos(rksol[t][2])], [l_1*sin(rksol[t][2]) +l_2*sin(rksol[t][3]), -l_1*cos(rksol[t][2]) -l_2*cos(rksol[t][3])]])),params$`

Click the figure to start animation



(%t12)

(%i13) `time(%);`

$$[0.656] \quad (\%o13)$$