Ecuación del movimiento de una partícula en una circunferencia

Based on Física Hasta Que Te Aburras MECÁNICA LAGRANGIANA (EJERCICIO #4) [Ecuación del movimiento de una partícula en una circunferencia]

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

(%i10) orderless(m,g,R)$

(%i11) declare([m,g,R],constant)$

(%i12) assume(m>0,g>0,R>0)$

(%i13) params:[m=1.0,g=9.8,R=5.0]$

(%i14) τ:60$

Generalized coordinates

(%i15) depends(ϕ,t)$

Geometry

(%i16) r:R*[cos(ϕ),sin(ϕ)]$

(%i17) v:diff(r,t);

$$[-R\sin\left(\phi\right)\left(\phi_t\right), R\cos\left(\phi\right)\left(\phi_t\right)] \tag{v}$$

2 Lagrangian Formalism

Kinetic Energy

(%i18) `ldisplay(T:trigsimp(1/2*m*v.v))$`

$$T = \frac{m\,R^2\,(\phi_t)^2}{2} \tag{\%t18}$$

Potential Energy

(%i19) `ldisplay(V:m*g*R*sin(ϕ))$`

$$V = mgR\,\sin(\phi) \tag{\%t19}$$

Lagrangian

(%i20) `ldisplay(L:expand(T-V))$`

$$L = \frac{m\,R^2\,(\phi_t)^2}{2} - mgR\,\sin(\phi) \tag{\%t20}$$

Momentum Conjugate

(%i21) `ldisplay(P:ev(diff(L,'diff(ϕ,t))))$`

$$P = m\,R^2\,\,(\phi_t) \tag{\%t21}$$

(%i22) `linsolve(p=P,diff(ϕ,t));`

$$\left[\phi_t = \frac{p}{m\,R^2}\right] \tag{\%o22}$$

Generalized Forces

(%i23) `ldisplay(F:diff(L,ϕ))$`

$$F = -mgR\,\cos(\phi) \tag{\%t23}$$

Euler-Lagrange Equation

(%i24) `aa:el(L,ϕ,t)$`

(%i26) `bb:ev(aa,eval,diff)$`

(%i27) `bb[1]:subst([k[0]=-E],-bb[1])$`

(%i28) `bb[2]:lhs(bb[2])-rhs(bb[2])=0$`

Conservation Laws

(%i29) `bb[1];`

$$\frac{m\,R^2\,(\phi_t)^2}{2} + mgR\,\sin(\phi) = E \tag{\%o29}$$

Equations of Motion

(%i30) `bb[2];`

$$m\,R^2\,\left(\phi_{tt}\right)+mgR\,\cos\left(\phi\right)=0 \tag{\%o30}$$

Solve for second derivative of coordinates

(%i31) `linsol:linsolve(bb[2],diff(`ϕ`,t,2));`

$$\left[\phi_{tt}=-\frac{g\,\cos\left(\phi\right)}{R}\right] \tag{linsol}$$

Check Conservation of Energy

(%i32) `lhs(bb[1]);`

$$\frac{m\,R^2\,\left(\phi_t\right)^2}{2}+mgR\,\sin\left(\phi\right) \tag{\%o32}$$

(%i33) `subst(linsol,diff(lhs(bb[1]),t)),fullratsimp;`

$$0 \tag{\%o33}$$

Equilibrium points

(%i34) `bb[2],[diff(`ϕ`,t)=0,diff(`ϕ`,t,2)=0];`

$$mgR\,\cos\left(\phi\right)=0 \tag{\%o34}$$

(%i35) `sol:solve(%,cos(`ϕ`));`

$$\left[\cos\left(\phi\right)=0\right] \tag{sol}$$

Small angles approximation

(%i36) `depends(`ϵ`,t)$`

(%i37) `ldisplay(C`ϕ`:ratdisrep(taylor(cos(`ϵ`-`π`/2),`ϵ`,0,1)))$`

$$C\phi=\epsilon \tag{\%t37}$$

(%i38) `ldisplay(S`ϕ`:ratdisrep(taylor(sin(`ϵ`-`π`/2),`ϵ`,0,1)))$`

$$S\phi=-1 \tag{\%t38}$$

(%i39) `bb[2],[`ϕ`=`ϵ`-`π`/2],diff,eval;`

$$m\,R^2\,\left(\epsilon_{tt}\right)+mgR\,\sin\left(\epsilon\right)=0 \tag{\%o39}$$

(%i40) `%,[cos(ϵ-π/2)=Cϕ,sin(ϵ-π/2)=Sϕ],expand;`

$$m\,R^2\,\left(\epsilon_{tt}\right)+mgR\epsilon=0 \tag{\%o40}$$

Solve for second derivative of coordinates

(%i41) `linsol:linsolve(%,diff(ϵ,t,2));`

$$\left[\epsilon_{tt}=-\frac{g\epsilon}{R}\right] \tag{linsol}$$

3 Hamiltonian Formalism

Legendre Transformation

(%i42)
```
Legendre:linsolve([p=P],['diff(ϕ,t)]);
```

$$\left[\phi_t = \frac{p}{m\,R^2}\right] \tag{Legendre}$$

Hamiltonian

(%i43)
```
ldisplay(H:ev(p*'diff(ϕ,t)-L,Legendre))$
```

$$H = mgR\,\sin\left(\phi\right) + \frac{p^2}{2m\,R^2} \tag{\%t43}$$

Equations of Motion

(%i44)
```
Hq:makelist(Hq[i],i,1,2)$
```

(%i46)
```
Hq[1]:'diff(ϕ,t)=diff(H,p)$
Hq[2]:'diff(p,t)=-diff(H,ϕ)$
```

(%i47)
```
map(ldisp,Hq)$
```

$$\phi_t = \frac{p}{m\,R^2} \tag{\%t47}$$

$$p_t = -mgR\,\cos\left(\phi\right) \tag{\%t48}$$

4 Reduce Order

(%i49)
```
depends(Φ,t)$
```

(%i50)
```
gradef(ϕ,t,Φ)$
```

Euler-Lagrange Equations

(%i51)
```
aa:el(L,ϕ,t)$
```

(%i53)
```
bb:ev(aa,eval,diff)$
```

(%i54)
```
bb[1]:subst([k[0]=-E],-bb[1])$
```

(%i55)
```
bb[2]:lhs(bb[2])-rhs(bb[2])=0$
```

Conservation Laws

(%i56)
```
bb[1];
```

$$mgR\,\sin\left(\phi\right)+\frac{m\,R^2\,\Phi^2}{2}=E \tag{\%o56}$$

Equations of Motion

(%i57)
```
bb[2];
```

$$mgR\,\cos\left(\phi\right)+m\,R^2\,\left(\Phi_t\right)=0 \tag{\%o57}$$

Solve for second derivative of coordinates

(%i58)
```
linsol:linsolve(bb[2],diff(ϕ,t,2));
```

$$\left[\Phi_t=-\frac{g\,\cos\left(\phi\right)}{R}\right] \tag{linsol}$$

Numerical solution (Lagrangian)

(%i59) `kill(labels)$`

(%i8)
```
funcs:[ϕ,Φ]$ldisplay(funcs)$
initial:[π/5,0]$ldisplay(initial)$
odes:[Φ,rhs(linsol[1])]$ldisplay(odes)$
interval:[t,0,τ]$ldisplay(interval)$
```

$$funcs = [\phi, \Phi] \tag{\%t2}$$

$$initial = \left[\frac{\pi}{5}, 0\right] \tag{\%t4}$$

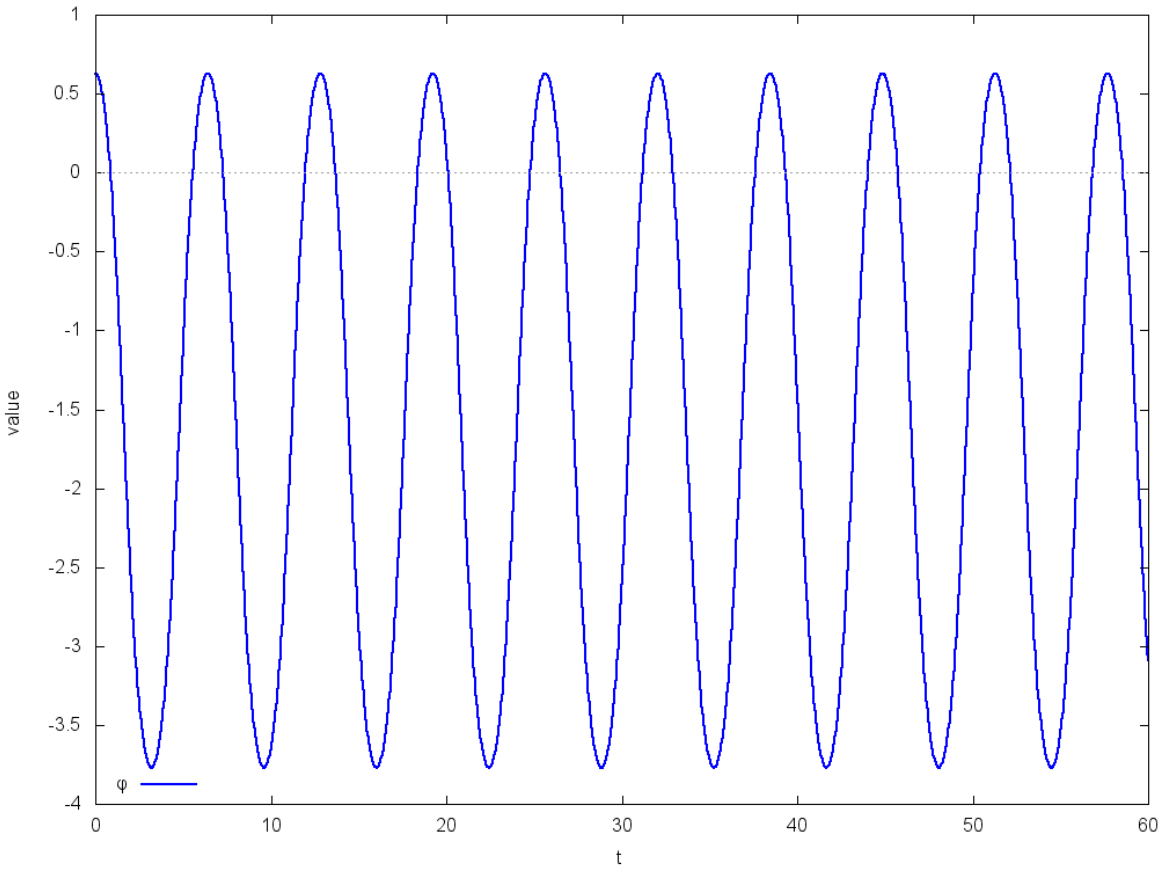
$$odes = \left[\Phi, -\frac{g\ \cos\left(\phi\right)}{R}\right] \tag{\%t6}$$

$$interval = [t, 0, 60] \tag{\%t8}$$

(%i9) `rksol:rkf45(odes,funcs,initial,interval, absolute_tolerance=1E-8,report=true),params$`

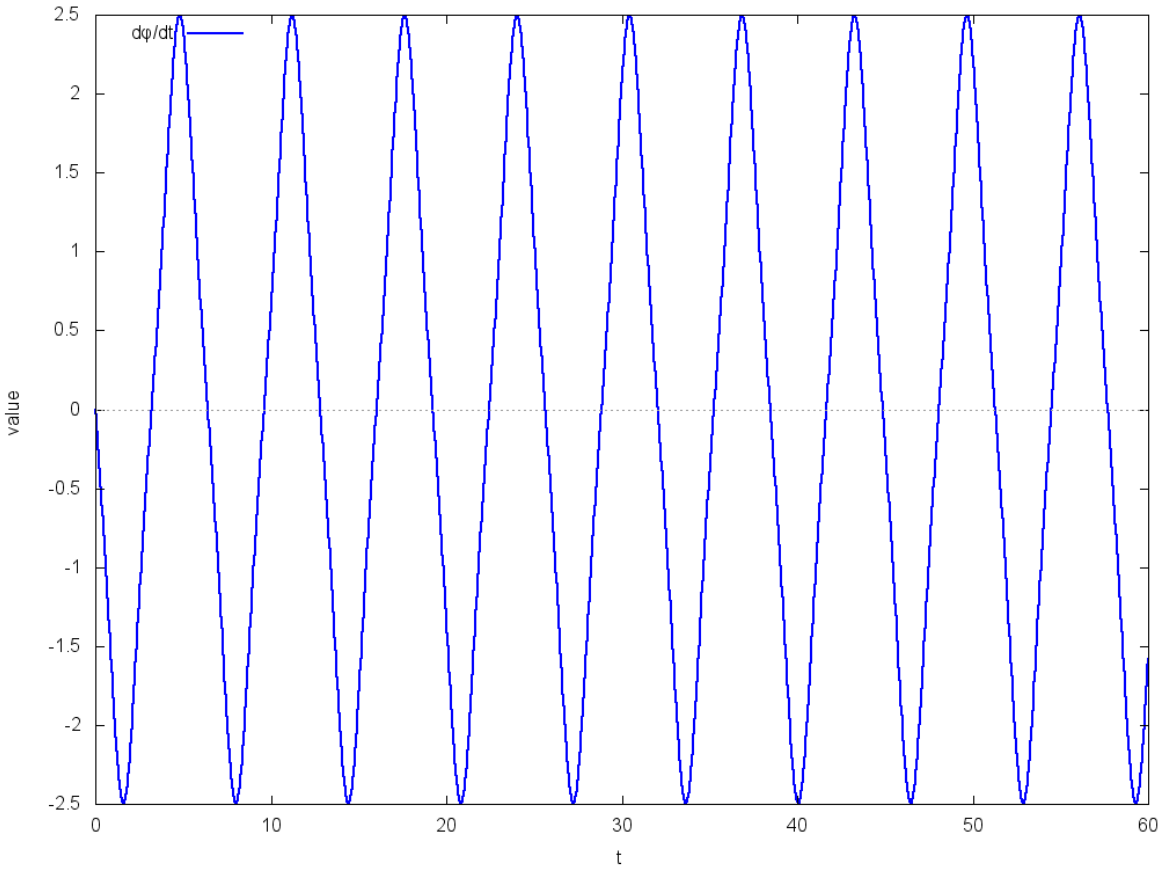
Info: rkf45:
Integration points selected:1833
Total number of iterations:1834
Bad steps corrected:2
Minimum estimated error:2.465810^{-9}
Maximum estimated error:9.30810^{-9}
Minimum integration step taken:0.024146
Maximum integration step taken:0.044882

(%i10) `wxplot2d([discrete,map(lambda([u],part(u,[1,2])),rksol)], [style,[lines,2]],[xlabel,"t"],[ylabel`
`[legend,"$\phi$"],[gnuplot_preamble,"set key bottom left"])$`



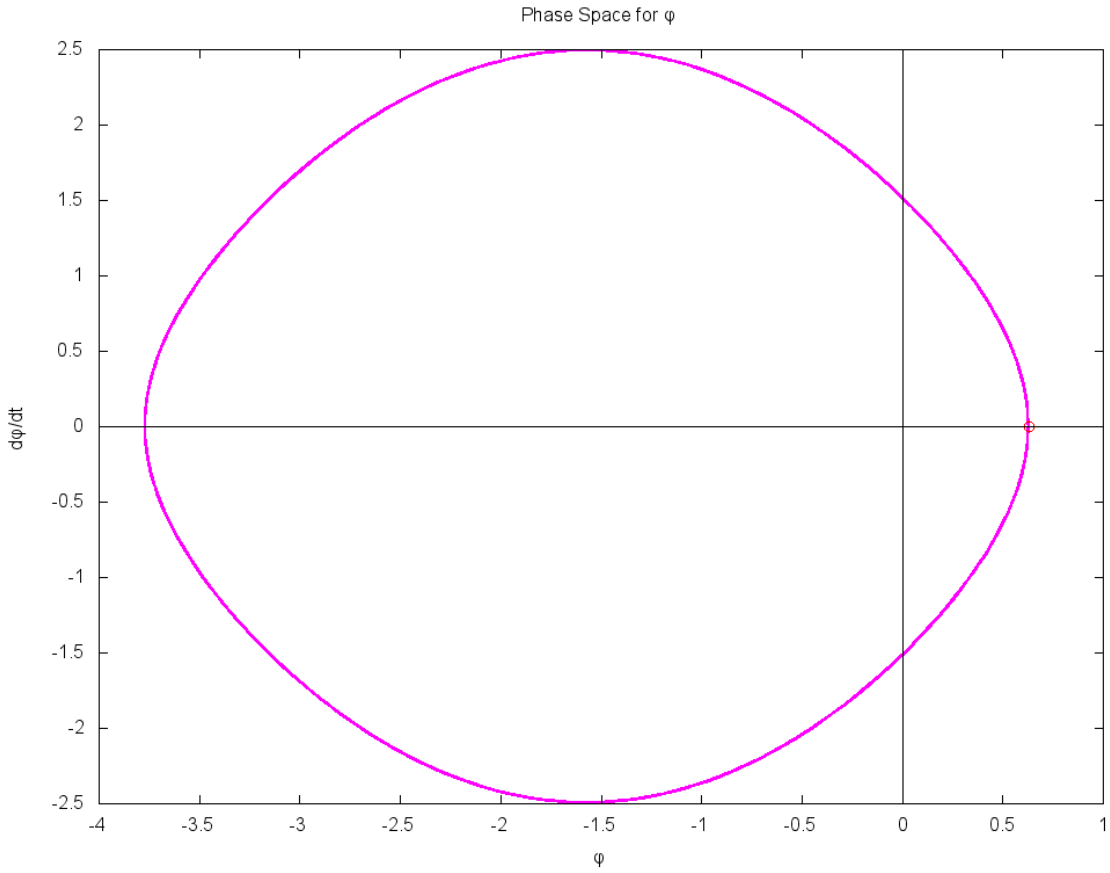
(%t10)

(%i11) `wxplot2d([discrete,map(lambda([u],part(u,[1,3])),rksol)], [style,[lines,2]],[xlabel,"t"],[ylabel`
`[legend,"dϕ/dt"],[gnuplot_preamble,"set key top left"])$`



(%t11)

(%i12)
```
wxplot2d([[discrete,map(lambda([u],part(u,[2,3])),rksol)], [discrete,[part(initial,[1,2])]]],[ax
[title,"Phase Space for ϕ"],[point_type,circle], [style,[lines,2],[points,3]],[color,magenta,red]
[xlabel,"ϕ"],[ylabel,"dϕ/dt"],[legend,false])$
```



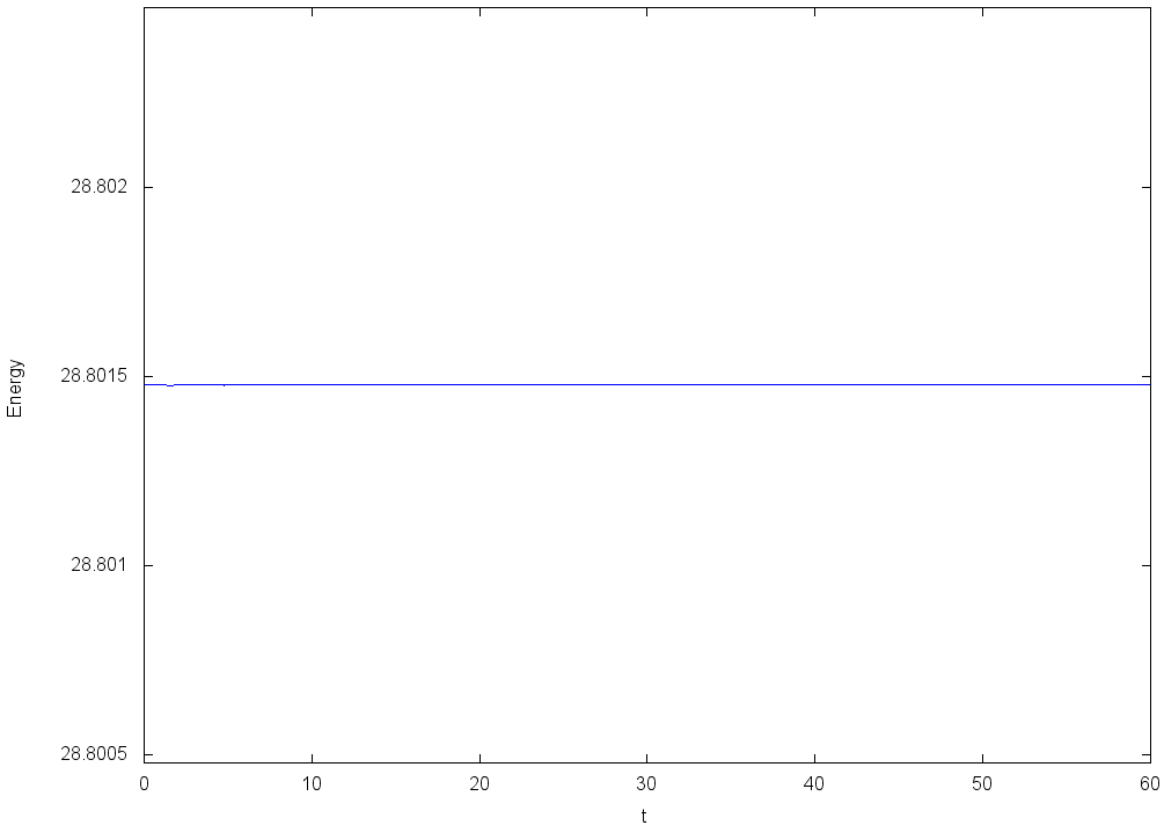
(%t12)

Check Conservation of Energy using the Numerical Data

(%i13)
```
W:bb[1],map("=",funcs,initial),params,numer,eval;
```

$$28.801 = E \qquad \text{(W)}$$

(%i14)
```
wxplot2d([discrete,makelist([first(rkline), ev(lhs(bb[1]),map("=",funcs,rest(rkline)),params)],rl
[xlabel,"t"],[y,lhs(W)-0.001,lhs(W)+0.001],[ylabel,"Energy"])$
```



(%t14)

5 Graphics

(%i15) kill(labels)$

(%i1) wxanimate_framerate:60$

(%i2) wxanimate_autoplay:false$

(%i3) rksol:rk(odes,funcs,initial,[t,0,τ/2,0.1]),params$

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

Create animated GIF file

(%i5)
```
draw(terminal = 'animated_gif, file_name = "Particula en una circunferencia",
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],
[cos(rksol[t][2]),sin(rksol[t][2])]])), t,1,length(rksol))),params$
```

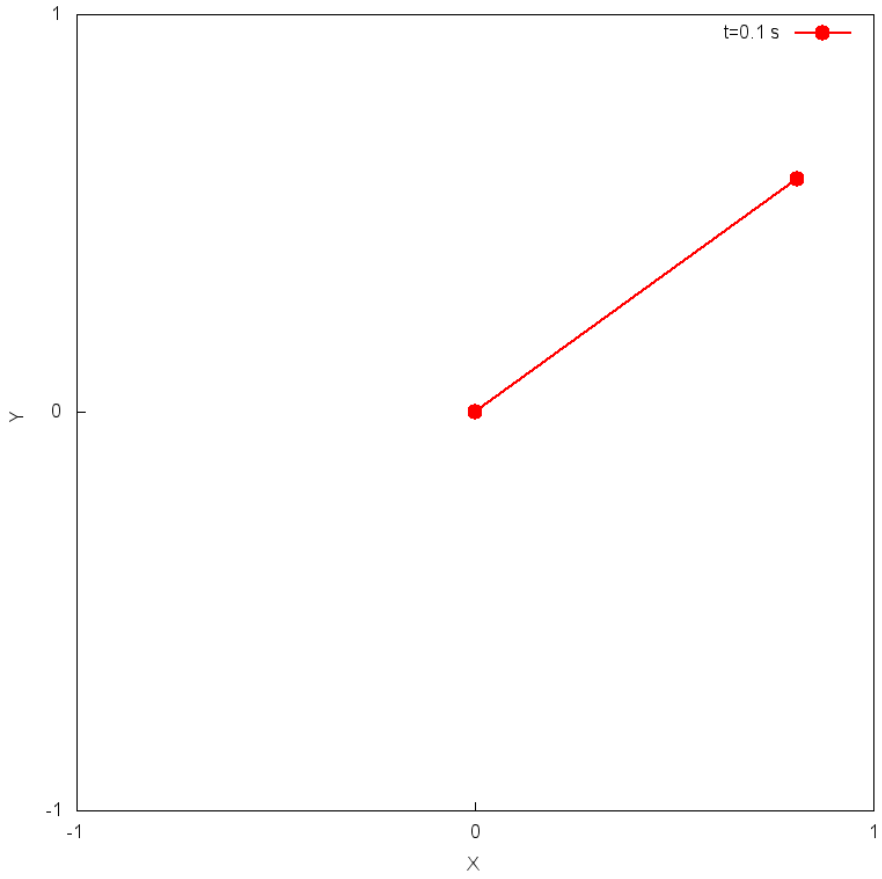
(%i6)
```
time(%);
```

[0.016] (%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],
[cos(rksol[t][2]),sin(rksol[t][2])]])),params$
```

Click the figure to start animation



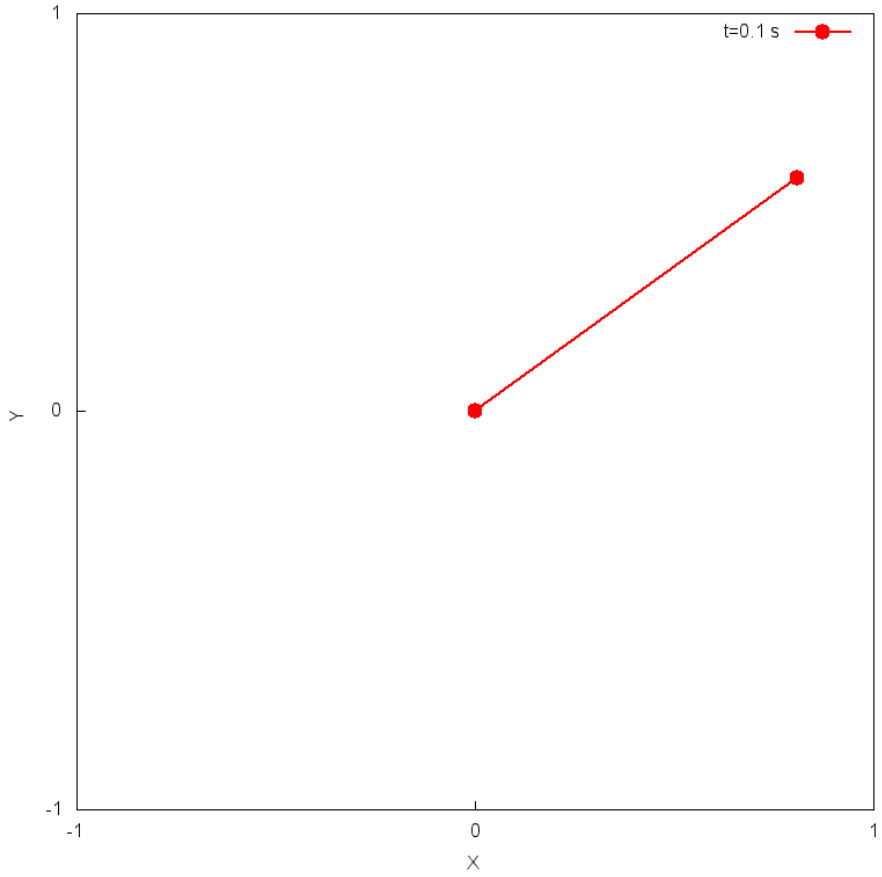
(%t9)

(%i10)
```
time(%);
```

[0.797] (%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],
[cos(rksol[t][2]),sin(rksol[t][2])]])),params$
```

Click the figure to start animation



(%t12)

(%i13)
```
time(%);
```

[0.985] (%o13)