

```

(* MECANICA TEORICA / FC-UABC / PROF. HECTOR ACEVES *)
(* PENDULO ROTANTE *)

(* Cargar primero el paquete de Calculo Variacional *)
Needs["Calculus`VariationalMethods`"]
(* Si se quiere saber que hace poner: ?EulerEquations *)

(* Limpiar todas las variables *)
ClearAll["Global`*"]
Off[General::spell]

Null3

(* La energia cinetica,
potencial y el Lagrangiano. La energia potencial es solo gravitacional *)
T =  $\frac{1}{2} m (x'[t]^2 + y'[t]^2)$ ;
Vgr = -m g y[t];
polarrule =
  {x -> ((a Cos[Ω #] + b * Sin[θ[#]]) &), y -> ((-a Sin[Ω #] + b * Cos[θ[#]]) &)};
Vx = x'[t] /. polarrule
Vy = y'[t] /. polarrule
Kinetic = T /. polarrule // Simplify
Potential = Vgr /. polarrule
L = T - Vgr /. polarrule // Simplify

-  $\frac{1}{4} g m^3 \text{Null}^3 (b \text{Cos}[\theta[t]] - a \text{Sin}[t \Omega]) (-a \Omega \text{Sin}[t \Omega] + b \text{Cos}[\theta[t]] \theta'[t])$ 
 $(-a \Omega \text{Cos}[t \Omega] - b \text{Sin}[\theta[t]] \theta'[t]) (a^2 \Omega^2 - 2 a b \Omega \text{Sin}[t \Omega - \theta[t]] \theta'[t] + b^2 \theta'[t]^2)$ 
 $(2 b g \text{Cos}[\theta[t]] + a (a \Omega^2 - 2 g \text{Sin}[t \Omega]) - 2 a b \Omega \text{Sin}[t \Omega - \theta[t]] \theta'[t] + b^2 \theta'[t]^2)$ 

(* Encontramos la ecuacion para θ[t] *)
EQN = Simplify[EulerEquations[
   $\frac{1}{2} m (2 b g \text{Cos}[\theta[t]] + a (a \Omega^2 - 2 g \text{Sin}[t \Omega]) - 2 a b \Omega \text{Sin}[t \Omega - \theta[t]] \theta'[t] + b^2 \theta'[t]^2)$ ,
  θ[t], t] ]

b m (-a Ω2 Cos[t Ω - θ[t]] + g Sin[θ[t]] + b θ''[t]) == 0

eqMotion =
  Solve[EQN, θ''[t]] /. Rule -> Equal // Simplify // Flatten // ExpandAll // ColumnForm

θ''[t] ==  $\frac{a \Omega^2 \text{Cos}[t \Omega - \theta[t]]}{b} - \frac{g \text{Sin}[\theta[t]]}{b}$ 

```

```
(* Graficamos una solucion, con condiciones iniciales (initialc)
tales que el pendulo no suba mas que la posicion de su punto de
suspension. Se obtienen las ecuaciones de movimiento con las condiciones
iniciales (neqc) y luego se resuelven numericamente (nSolc). Despues se
define funcion interpolante  $\theta_c$  para poder graficarla posteriormente*)
valuesc = {a → 1, b → 2, g → 9.8, m → 0.5,  $\Omega$  → (0.5 * Sqrt[9.8 / 2]) }
 $\tau$  = 1 / Sqrt[g / b] /. valuesc
initialc = { $\theta$ [0] ==  $\pi/2$ ,  $\theta'$ [0] == 0.0} /. valuesc;
neqc = {eqMotion, initialc} /. Rule → Equal /. valuesc // Flatten

{a → 1, b → 2, g → 9.8, m → 0.5,  $\Omega$  → 1.1068}

0.451754

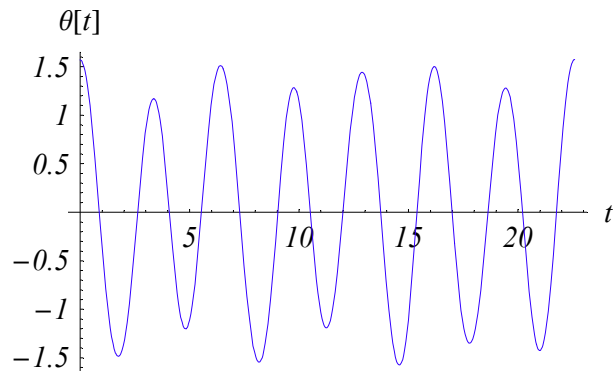
{ $\theta''$ [t] == 0.6125 Cos[1.1068 t -  $\theta$ [t]] - 4.9 Sin[ $\theta$ [t]],  $\theta$ [0] ==  $\frac{\pi}{2}$ ,  $\theta'$ [0] == 0.}
```

```
nSolc = NDSolve[{ $\theta''$ [t] == 0.6125 Cos[1.1068 t -  $\theta$ [t]] - 4.9 Sin[ $\theta$ [t]],
 $\theta$ [0] == 1.5708,  $\theta'$ [0] == 0.0},  $\theta$ [t], {t, 0, 100 *  $\tau$ }, MaxSteps → 3000] // Flatten;
 $\theta_c$ [t_] =  $\theta$ [t] /. nSolc;
```

General::spell1 : Possible spelling error: new symbol name " $\theta_c$ " is similar to existing symbol " $\theta$ ". MORE...

```
(* Graficamos la variacion temporal de la posicion
y luego de la coordenada angular. Y luego las combinamos para
ver el movimiento de la particula masa en el plano XY *)
```

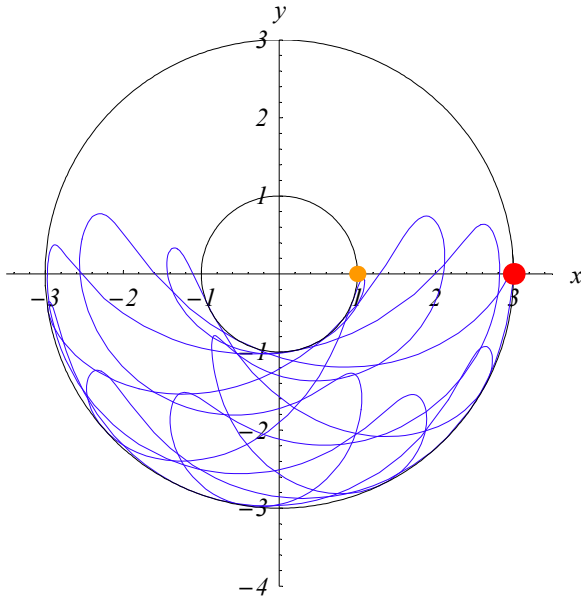
```
Plot[ $\theta_c$ [t] // Evaluate, {t, 0, 50 *  $\tau$ },
PlotStyle → {Hue[.7]} ,
TextStyle → {FontFamily → "Times-Italic", FontSize → 14}, AxesLabel → {"t", " $\theta$ [t]"} ];
```



```

grafical =
  ParametricPlot[ { 1 * Cos[1.1068 * t] + 2 * Sin[θc[t]], 1 * Sin[1.1068 * t] - 2 * Cos[θc[t]] }
    // Evaluate, {t, 0, 50 * τ},
    PlotStyle → Hue[0.7], TextStyle → {FontFamily → "Times-Italic", FontSize → 12},
    AspectRatio → 1, AxesLabel → {"x", "y"}, PlotRange → {{-3.5, 3.5}, {-4, 3}},
    Epilog → {Circle[{0, 0}, #] & /@ {1, 3}, PointSize[0.04], Hue[1],
      Point[{3, 0}], PointSize[0.03], Hue[0.1], Point[{1, 0}]}
  ];

```

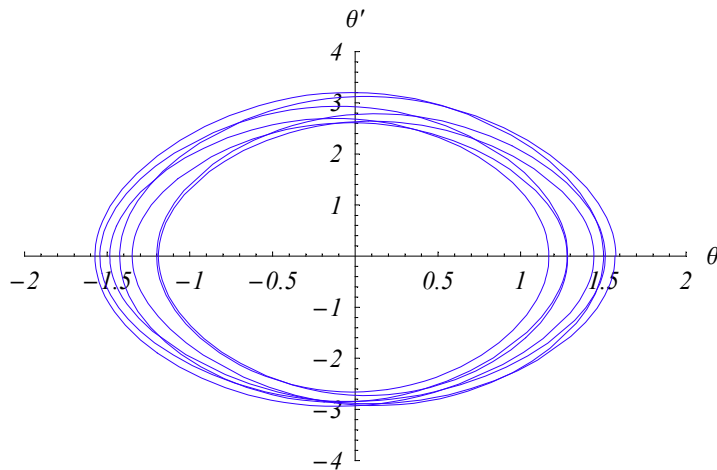


(\* Trayectoria en el espacio fase:  $(\theta, \theta')$  \*)

```

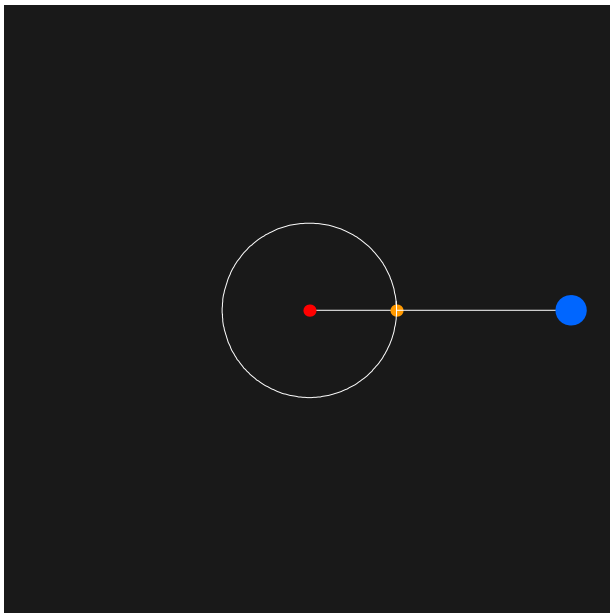
pt1 = ParametricPlot[ {θc[t], θc'[t]} // Evaluate, {t, 0, 50 * τ},
  PlotStyle → {Hue[.7]}, TextStyle → {FontFamily → "Times-Italic", FontSize → 12},
  AxesLabel → {"θ", "θ'"}, PlotRange → {{-2, 2}, {-4, 4}};

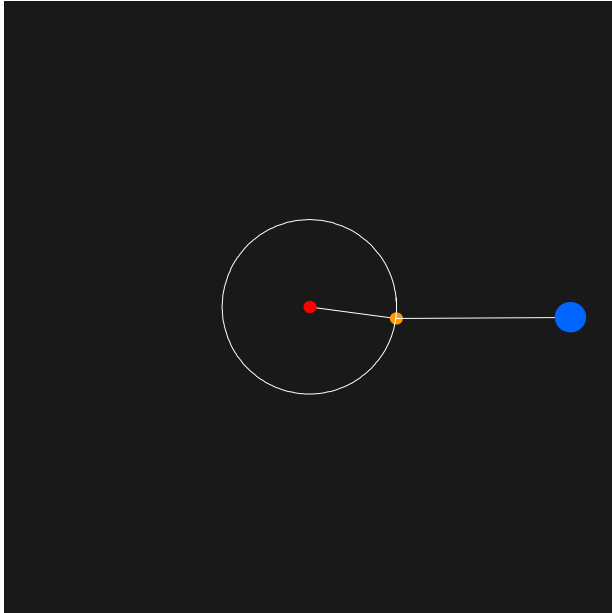
```



```
(* Animacion para la primer condicion inicial,
con X numero de frames. Para sacar ya una movie mas decente cambiar,
por ejemplo frames=80, y el nombre del archivo y su direccion
a donde se quiera. El rango tambien puede cambiarse a 0,
50 tau. Aqui solo se ponen dos cuadros para ilustrar. *)
line[t_] :=
Show[
Graphics[
{ { Line[{ {0, 0}, {1 * Cos[ 1.1068 t], 1 * Sin[1.1068 t] } ] } ,
{ PointSize[0.02], Hue[0.1], Point[ { 1 * Cos[1.1068 t] , Sin[1.1068 t] } ] } },
Line[{ {1 * Cos[ 1.1068 t], 1 * Sin[1.1068 t] },
{ 1 * Cos[1.1068 * t] + 2 * Sin[θc[t]], 1 * Sin[1.1068 * t] - 2 * Cos[θc[t]] } ]},
{ PointSize[0.05], Hue[0.6],
Point[ { 1 * Cos[1.1068 * t] + 2 * Sin[θc[t]], 1 * Sin[1.1068 * t] - 2 * Cos[θc[t]] } ] }
}
},
PlotRange → {{-3.5, 3.5}, {-3.5, 3.5}}, Frame → False, AspectRatio → 1,
TextStyle → {FontFamily → "Times-Italic", FontSize → 12}, Background → GrayLevel[0.1],
Epilog → {Circle[{0, 0}, #] & /@ {1}, PointSize[0.02], Hue[0], Point[{0, 0]}}
]
]

frames = 2;
movie = line[#] & /@ Range[0, 50 * τ, 50 * τ / ( frames - 1)];
Export["c:/test.gif", movie, ConversionOptions →
{"AnimationDisplayTime" → 0.5 , "Loop" → False}, ImageSize → {500, 500}]
```





c:/test.gif