Modelado dinámico de un móvil sujeto a $y = f(x)$: Hamiltoniano

Presentación en vídeo: http://personales.upv.es/asala/YT/V/mcm6ha.html

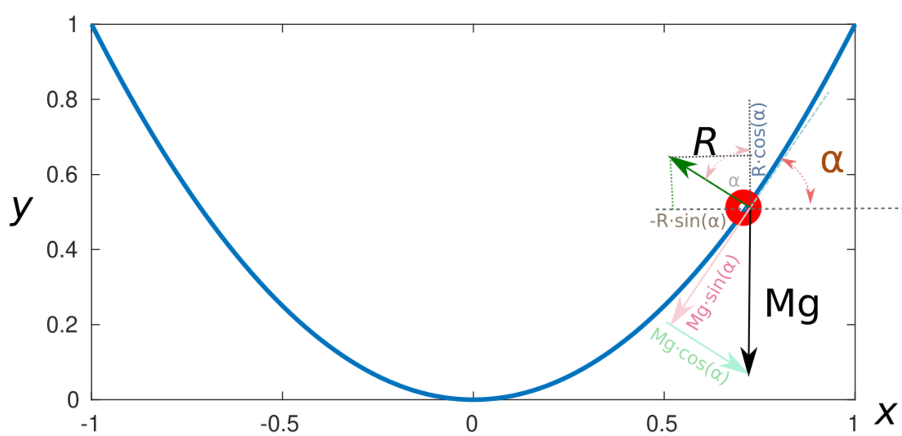
Este código funcionó sin errores en Matlab **R2021a**

Objetivo: obtener, por distintas metodologías (todas ellas equivalentes), las ecuaciones del movimiento de un cuerpo (puntual) que desliza sobre una curva $y = f(x)$ dada. En este caso, este material se centra en el Hamiltoniano (usamos Euler-Lagrange de preliminares).

Tabla de Contenidos

Planteamiento del problema

Movimiento restringido a la ligadura $y = f(x)$. Posible fricción y fuerzas externas.




```
caso_concreto=true %saldrán ecuaciones de mov. en función de f y sus derivadas
```

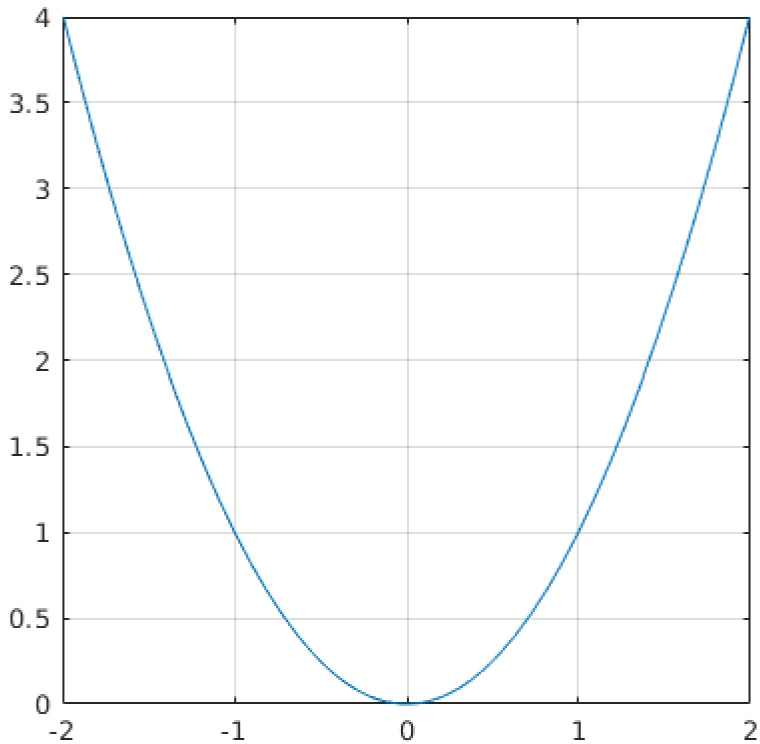
```
caso_concreto = logical

   1
```

```
syms x real
syms v_x a_x  real
if(~caso_concreto)
    syms M g b real
    syms f(x) %función simbólica genérica
else %ejemplos concretos
    M=1;g=10;
```

```
    b=0.25; %coeficiente de fricción
    %f(x)=tan(pi/6)*x+1 %recta
    f(x)=x^2 %parábola
    %f(x)=2-sqrt(2^2-x^2) %círculo radio 2
    %f(x)=sin(x*pi/4)^2*2
    fplot(f,[-2,2]), axis equal, grid on
end
```

f(x) = x^2



Notación

La posición horizontal, velocidad horizontal y aceleración horizontal las denominaremos, respectivamente:

$$x, \qquad v_x := \frac{dx}{dt}, \qquad a_x := \frac{dv_x}{dt} = \frac{d^2x}{dt^2}$$

Cinemática

```
r=[x; f(x)] %vector de posición r=[x; y];
```

r =

$$\begin{pmatrix} x \\ x^2 \end{pmatrix}$$

```
veloc=diff(r,x)*v_x
```

veloc =

$$\begin{pmatrix} v_x \\ 2\,v_x\,x \end{pmatrix}$$

Dinámica

La fricción será un coeficiente multiplicado por el componente tangencial de la velocidad.

```
F_friccion_vector=-b*veloc; %coef * veloc. tangencial...
```

***Es un modelo simplista**: podríamos poner término de Coulomb (constante·signo velocidad tangencial) proporcional a fza normal, hasta variaciones de b según dicha fuerza normal... pero lo dejamos estar por simplicidad (modelos de fricción no lineal, dinámicos, lubricación, etc. no son objetivo de este material).

Euler-Lagrange 2nd kind (cinemática en forma paramétrica)

Indicado cuando se dispone de una expresión paramétrica de todas las posiciones y velocidades intervinientes en la energía cinética y potencial en función de unas coordenadas $(q_1, \ldots, q_N)$ -- una única $q_1 \equiv x$ en este caso concreto --, y sus derivadas $V(q)$, $T(q, \dot{q})$.

El sistema se mueve en la trayectoria $(x, f(x))$, esto es, tiene 1 Grado de Libertad donde $q \equiv x$.

El modelo resulta más sencillo (1 ecuación, no interviene el multiplicador de Lagrange λ), pero no calcula las fuerzas de reacción porque desconoce que "realmente" está en un mundo de "dimensión mayor a 1".

Usando la cinemática en función de v_x tenemos:

```
T=simplify(0.5*M*(veloc(1)^2+veloc(2)^2)) %energía cinética
```

T =

$$\frac{{v_x}^2\,(4\,x^2+1)}{2}$$

```
V=M*g*r(2) %energía potencial
```

V = $10\,x^2$

```
L=T-V %Lagrangiano
```

L =

$$\frac{{v_x}^2\,(4\,x^2+1)}{2} - 10\,x^2$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial v_x}\right)-\frac{\partial L}{\partial x}=Q$$

siendo Q la fuerza generalizada dada por el trabajo de las fuerzas externas cuando se desplace virtualmente δx:

```
syms F_externa_x F_externa_y real
Qext=F_externa_x*1+F_externa_y*diff(f,x)
```

Qext(x) = $F_{\mathrm{externa},x}+2\,F_{\mathrm{externa},y}\,x$

```
Q_friccion=F_friccion_vector(1)*1+F_friccion_vector(2)*diff(f,x)
```

Q_friccion(x) =

$$-v_x\,x^2-\frac{v_x}{4}$$

```
Q=Qext+Q_friccion
```

Q(x) =

$$-v_x\,x^2+2\,F_{\mathrm{externa},y}\,x+F_{\mathrm{externa},x}-\frac{v_x}{4}$$

```
derivada_de_p=jacobian( diff(L, v_x),[x  v_x])*[ v_x; a_x];
acelx_EulerLagrange=simplify(solve(derivada_de_p-diff(L,x)==Q , a_x),50)
```

acelx_EulerLagrange =

$$\frac{F_{\mathrm{externa},x}-x\,\left(4\,{v_x}^2-2\,F_{\mathrm{externa},y}+20\right)}{4\,x^2+1}-\frac{v_x}{4}$$

Método del Hamiltoniano

```
syms p real
H1=-L+p*v_x %Hamiltoniano Transf. Legendre (no vale todavía)
```

H1 =

$$p\,v_x-\frac{{v_x}^2\,(4\,x^2+1)}{2}+10\,x^2$$

Necesitamos expresar el Hamiltoniano en las coordenadas "canónicas" (x, p). Para ello, usamos:

```
this_is_p=diff(L,v_x)
```

this_is_p = $v_x\,(4\,x^2+1)$

de modo que despejando v_x tenemos:

```
this_is_vx=solve(p==this_is_p,v_x)
```

```
this_is_vx =
```

$$\frac{p}{4\,x^2+1}$$

y ya en coordenadas canónicas requeridas por la teoría, sale:

```
H=simplify(subs(H1,v_x,this_is_vx),50) %Hamiltonian
```

```
H =
```

$$\frac{p^2}{8\,x^2+2}+10\,x^2$$

La energía total mecánica es el Hamiltoniano en este caso (Lagrangiano no dependiente explícitamente de t, $\frac{\partial L}{\partial t}=0$):

```
EnergiaMecanica=simplify(subs(T+V,v_x,this_is_vx),50)
```

```
EnergiaMecanica =
```

$$\frac{p^2}{8\,x^2+2}+10\,x^2$$

La primera ec movimiento es $\dot{p}=-\frac{\partial H}{\partial q}$, la segunda $\dot{q}=\frac{\partial H}{\partial p}$.

Incorporando fuerzas externas y fricción (caso no conservativo), modificamos la ecuación de $\dot{p}$ a:

$$\dot{p}=-\frac{\partial H}{\partial q}+Q$$

```
Q_Hamilton=simplify(subs(Q,v_x,this_is_vx)) %gastamos el Q de Euler-Lagrange
```

```
Q_Hamilton(x) =
```

$$F_{\mathrm{externa},x}-\frac{p}{4}+2\,F_{\mathrm{externa},y}\,x$$

```
dot_p=simplify(-diff(H,x)+Q_Hamilton, 50)
```

```
dot_p(x) =
```

$$F_{\mathrm{externa},x}-\frac{p}{4}-20\,x+2\,F_{\mathrm{externa},y}\,x+\frac{4\,p^2\,x}{(4\,x^2+1)^2}$$

```
dot_x=simplify(diff(H,p))
```

```
dot_x =
```

$$\frac{p}{4\,x^2+1}$$

Con ello estaría el modelo completo, y podría integrarse/simularse.

Para comparar con modelos anteriores Euler-Lagrange, vamos a calcular la segunda derivada de x:

```
d2xdt=simplify(jacobian(dot_x,[x p])*[dot_x;dot_p],200);
```

y expresarla en función de x, y v_x en vez de en las coordenadas "canónicas" (x, p):

```
d2xdt_enfuncionde_x_v_x=simplify(subs(d2xdt,p,this_is_p),50)
```

d2xdt_enfuncionde_x_v_x(x) =

$$\frac{F_{\mathrm{externa},x} - x\,\left(4\,{v_x}^2 - 2\,F_{\mathrm{externa},y} + 20\right)}{4\,x^2+1} - \frac{v_x}{4}$$

Como última observación, el Hamiltoniano ($\equiv$ Energía en este caso) se conserva sin fuerza externa ni fricción:

```
dHdt=simplify(diff(H,x)*dot_x+diff(H,p)*dot_p,50)
```

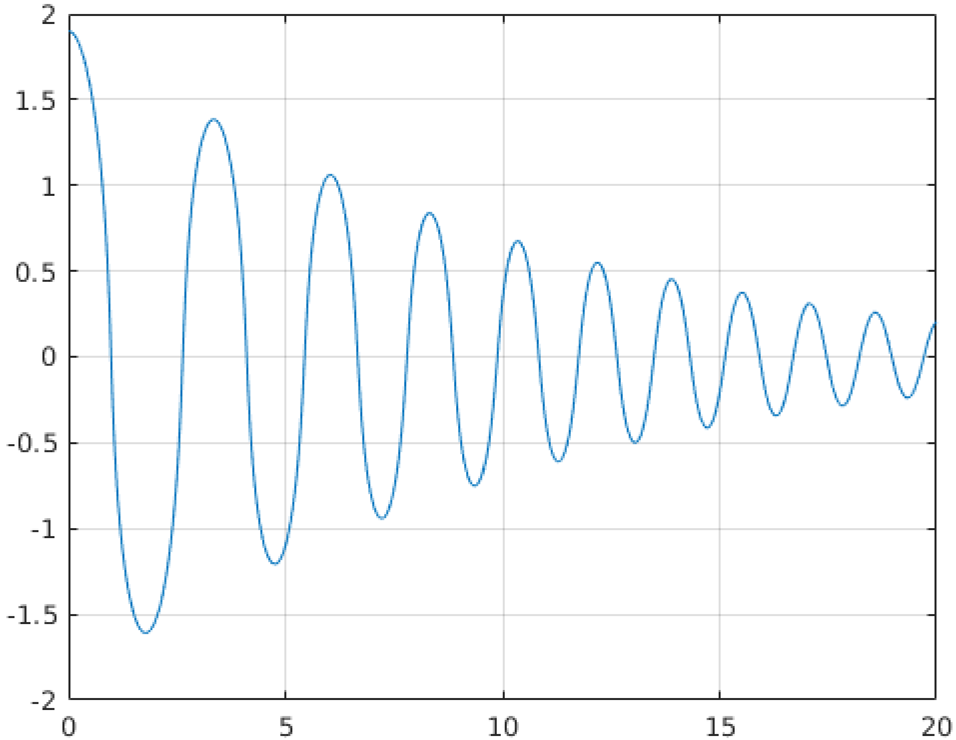
dHdt(x) =

$$\frac{p\,\left(4\,F_{\mathrm{externa},x} - p + 8\,F_{\mathrm{externa},y}\,x\right)}{4\,\left(4\,x^2+1\right)}$$

Simulación temporal

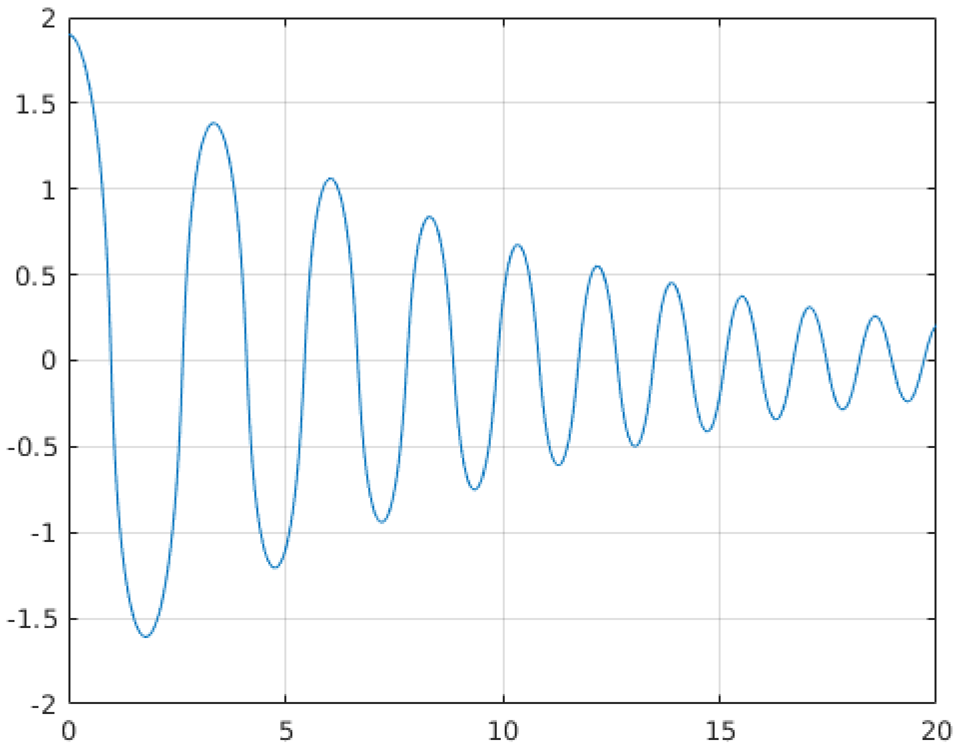
El Euler-Lagrange (estados: {posición y velocidad horizontal}):

```
if(~caso_concreto) %no podemos simular "numéricamente" en simbólico
    return %tampoco podemos resolver las ecuaciones diferenciales no-lineales
end
opts=odeset('AbsTol',1e-5,'RelTol',1e-4);
d2xdt=matlabFunction(acelx_EulerLagrange,'Vars',{x, v_x, F_externa_x,F_externa_y});
simularEL=@(t,estado) [estado(2);d2xdt(estado(1),estado(2),0,0)];
Tfinal=20;
[TiemposEL,EstadosEL]=ode45(simularEL,[0 Tfinal],[1.9;0],opts);
plot(TiemposEL,EstadosEL(:,1)), grid on
```



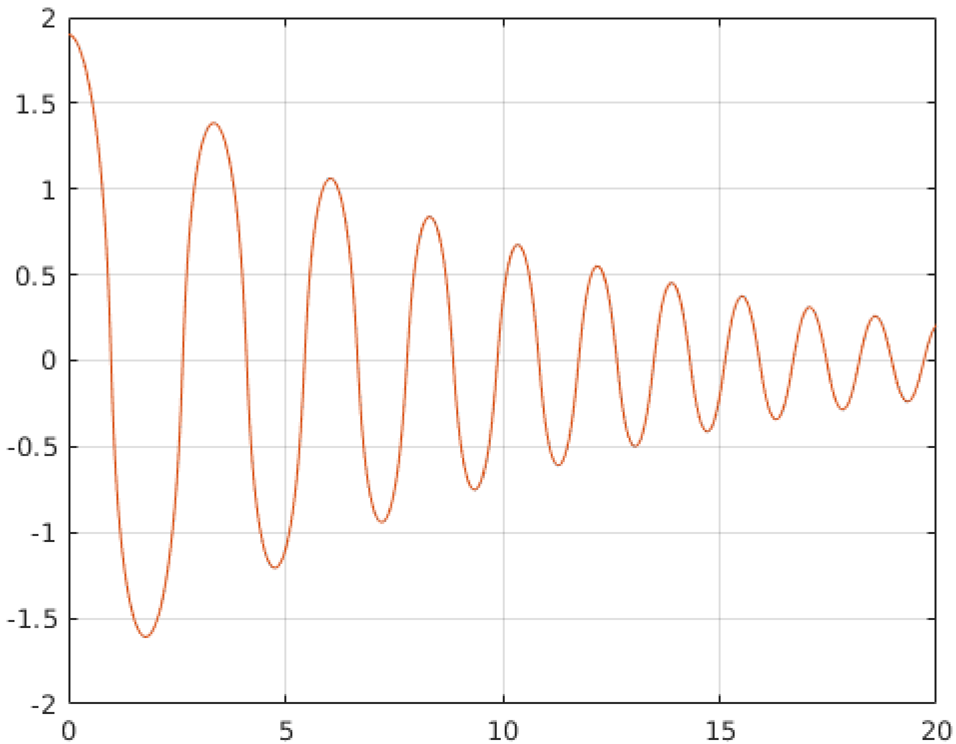
Ecs. Hamilton (estados: {posición horizontal, momentum generalizado}):

```
dpdt=matlabFunction(dot_p,'Vars',{x,p,F_externa_x,F_externa_y});
dxdt=matlabFunction(dot_x,'Vars',{x,p});
simularHamilton=@(t,estado) [dxdt(estado(1),estado(2)); dpdt(estado(1),estado(2),0,0)];
[TiemposH,EstadosH]=ode45(simularHamilton,[0 Tfinal],[1.9;0],opts);
plot(TiemposH,EstadosH(:,1)), grid on
```



Ambas soluciones (Hamilton vs. Euler-Lagrange), por supuesto, dan trayectorias idénticas:

```
plot(TiemposEL,EstadosEL(:,1)), hold on
plot(TiemposH,EstadosH(:,1)), hold off
grid on
```


2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
5
10
15
20