4123 HW 3

Duncan Wilkie

19 October 2021

1

Call the position of the natural length of the spring at $t = 0$ by x_0. The Lagrangian is then $L = \frac{1}{2}m\dot{x}^2 - \frac{1}{2}k(x - vt - x_0)^2$. The resulting Euler-Lagrange equation is

$$\frac{\partial L}{\partial x} = \frac{d}{dt}\frac{\partial L}{\partial \dot{x}} \Leftrightarrow -k(x - vt - x_0) = m\ddot{x} \Leftrightarrow m\ddot{x} + kx = kvt + kx_0$$

We may solve the homogeneous equation via the ansatz $x = e^{mt}$ as

$$m = \pm i\sqrt{\frac{k}{m}} \Rightarrow x = c_1 e^{ti\sqrt{\frac{k}{m}}} - c_2 e^{-ti\sqrt{\frac{k}{m}}} = C_1 \sin\left(t\sqrt{\frac{k}{m}}\right) + C_2 \cos\left(t\sqrt{\frac{k}{m}}\right)$$

A particular solution to the non-homogeneous case is by inspection $x = vt + x_0$, so the full solution is

$$x(t) = x_0 + vt + C_1 \sin\left(t\sqrt{\frac{k}{m}}\right) + C_2 \cos\left(t\sqrt{\frac{k}{m}}\right)$$

Since $v = x'(t)$ is taken to be constant,

$$v + C_1\sqrt{\frac{k}{m}} \sin\left(t\sqrt{\frac{k}{m}}\right) + C_2\sqrt{\frac{k}{m}} \cos\left(t\sqrt{\frac{k}{m}}\right) = v$$

The only way this is satisfied for all t is if $C_1 = C_2 = 0$. Since we take $x(0) = 0$, this also implies $x_0 = 0$, and the solution to the initial value problem is

$$x(t) = vt$$

The generalized momentum is

$$\frac{\partial L}{\partial \dot{x}} = m\dot{x} = mv$$

These are both constant by assumption, so the generalized momentum is conserved.

2

The contrapositive of this was proven in class, but I'll do it again. If the Lagrangian is symmetric under time translations, it does not depend on time. The Euler-Lagrange first-integral is proved as follows:

Take the multivariable chain rule

$$\frac{dL}{dt} = \frac{\partial L}{\partial t} + \frac{\partial L}{\partial q}\dot{q} + \frac{\partial L}{\partial \dot{q}}\ddot{q}$$

By the Euler-Lagrange equation, $\frac{\partial L}{\partial q} = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}}$, and if L is independent of t this becomes

$$\frac{dL}{dt} = \dot{q}\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} + \frac{\partial L}{\partial \dot{q}}\ddot{q}$$

Noticing that the right hand side is of the form of the product rule, this becomes

$$\frac{dL}{dt} = \frac{d}{dt}\left(\dot{q}\frac{\partial L}{\partial \dot{q}}\right)$$

Integrating with respect to t,

$$L = \dot{q}\frac{\partial L}{\partial \dot{q}} + c \Leftrightarrow \dot{q}\frac{\partial L}{\partial \dot{q}} - L = \text{const}$$

Applying this yields

$$\frac{\partial L}{\partial \dot{q}_i}\dot{q}_i - L = \text{const}$$

The formal definition of the Hamiltonian is $H = \sum_i \frac{\partial L}{\partial \dot{q}_i}\dot{q}_i - L$. L is a constant with respect to time, and the first-integral equation implies that each term of the sum is constant as well. Thus, the Hamiltonian is not time-dependent.

3

The Lagrangian for such a system is

$$L = \frac{1}{2}mv^2 + mgy$$

This does not depend on x, z, or t, and so the quantities p_x, p_z, and E are conserved by Noether's theorem.

4

Applying the Euler-Lagrange first integral,

$$f - y'\frac{\partial f}{\partial y'} = c_1 \Leftrightarrow y\sqrt{1+y'^2} - \frac{yy'^2}{\sqrt{1+y'^2}} = c_1 \Leftrightarrow y + yy'^2 - yy'^2 = c_1\sqrt{1+y'^2}$$

$$\Leftrightarrow y^2 = c_1^2(1+y'^2) \Leftrightarrow 1+y'^2 = \frac{y^2}{c_1^2}$$

From the normal Euler-Lagrange equations,

$$\frac{\partial f}{\partial y} = \frac{d}{dx}\frac{\partial f}{\partial y'} \Leftrightarrow \sqrt{1+y'^2} = \frac{d}{dx}\left[\frac{yy'}{\sqrt{1+y'^2}}\right]$$

$$\Leftrightarrow \sqrt{1+y'^2} = \frac{y'^2}{\sqrt{1+y'^2}} + yy''\left(\frac{1}{\sqrt{1+y'^2}} - \frac{y'^2}{(1+y'^2)^{3/2}}\right)$$

$$\Leftrightarrow (1+y'^2)^2 = y'^2 + y'^4 + yy'' + yy'^2y'' - yy'^2y''$$

$$\Leftrightarrow y'^4 + 2y'^2 + 1 = y'^2 + y'^4 + yy''$$

$$\Leftrightarrow y'^2 - yy'' + 1 = 0$$

Substituting the first-integral expression for $1+y'^2$,

$$\frac{y^2}{c_1^2} = yy'' \Leftrightarrow y'' - c^2y = 0$$

Via the ansatz $y = e^{mx}$, $m^2 - c^2 = 0 \Rightarrow y = c_1e^{cx} + c_2e^{-cx}$ is the general solution to the homogeneous equation. Plugging this in to the ODE obtained from the Euler-Lagrange equations,

$$\left(c_1ce^{cx} - c_2ce^{-cx}\right)^2 - \left(c_1e^{cx} + c_2e^{-cx}\right)\left(c_1c^2e^{cx} + c_2c^2e^{-cx}\right) + 1 = 0$$

$$\Leftrightarrow \left(c_1^2c^2e^{2cx} - 2c_1c_2c^2 + c_2^2c^2e^{-2cx}\right) - \left(c_1^2c^2e^{2cx} + 2c_1c_2c^2 + c_2^2c^2e^{-2cx}\right) + 1 = 0$$

$$\Leftrightarrow 1 = 4c_1c_2c^2$$

With the boundary conditions $y(a) = A$ and $y(b) = B$, we obtain the nonlinear system

$$A = c_1e^{ca} + c_2e^{-ca}$$

$$B = c_1e^{cb} + c_2e^{-cb}$$

$$c = \frac{1}{2\sqrt{c_1c_2}}$$

We may write the system as

$$\vec{F} = \begin{bmatrix} xe^{za} + ye^{-za} - A \\ xe^{zb} + ye^{-zb} - B \\ 4xyz^2 - 1 = 0 \end{bmatrix} = 0$$

By the inverse function theorem, this has solutions when

$$J = \begin{bmatrix} \frac{\partial F_1}{\partial x} & \frac{\partial F_2}{\partial x} & \frac{\partial F_3}{\partial x} \\ \frac{\partial F_1}{\partial y} & \frac{\partial F_2}{\partial y} & \frac{\partial F_3}{\partial y} \\ \frac{\partial F_1}{\partial z} & \frac{\partial F_2}{\partial z} & \frac{\partial F_3}{\partial z} \end{bmatrix}$$

is nonsingular, i.e. when

$$
\begin{aligned}
0 \neq &\begin{vmatrix} e^{za} & e^{zb} & 4yz^2 \\ e^{-za} & e^{-zb} & 4xz^2 \\ xae^{za} + yae^{-za} & xbe^{zb} + ybe^{-zb} & -1 \end{vmatrix} \\
&= e^{za}\left[-4xz^2\left(xbe^{zb} + ybe^{-zb}\right) - e^{-zb}\right] - e^{zb}\left[-4xz^2\left(xae^{za} + yae^{-za}\right) - e^{-za}\right] \\
&\quad +4yz^2\left[e^{-za}\left(xbe^{zb} + ybe^{-zb}\right) - e^{-zb}\left(xae^{za} + yae^{-za}\right)\right] \\
&= (4xyz^2(a+b)+1)\left(e^{z(b-a)} - e^{z(a-b)}\right) + 4z^2(a-b)\left(x^2e^{z(a+b)} - y^2e^{-z(a+b)}\right)
\end{aligned}
$$

It is evident this is zero at $z = 0$, and where $a = b$. However, numerical experimentation suggests these are the only places it becomes zero. The following Python script checks the values of the expression for all values of its five arguments between -5 and 5 with step 0.01, and outputs any roots it finds that don't match the two cases listed above. It takes an eternity to run, due to the superexponential nature of checking a Cartesian product of arrays, but produces no output.

```
from numpy import arange
from math import exp
from itertools import product

def f(a, b, x, y, z):
    return 4*x*y*z**2*(a+b)*(exp(-z*(b-a))-exp(z*(a-b))) \
            +4*z**2*(a-b)*(x**2*exp(z*(a+b))-y**2*exp(-z*(a+b))) \
            +exp(z*(b-a))-exp(z*(a-b))

for a, b, x, y, z in product(*[arange(-5, 5, 0.01) for i in range(0,5)]):
    res = f(a, b, x, y, z)
    if abs(res) < 1e-6 and abs(z) > 0.0001 and abs(a-b) > 0.0001:
        print("zero at ", *[round(a, 4) for a in [a,b,x,y,z]])
```

Given that the system appears to be somewhat "nice" in terms of existence, it's reasonable to suspect root-finding will proceed smoothly. The GNU Octave/MATLAB script below is likely to be able to compute the constants for any input initial condition with reasonable choice of initial guess.

```
1;

function y = f(x)

  # Change these to desired initial conditions
  a = 2;
  A = 5;
  b = 3;
  B = 5;

  y = zeros(3, 1);
  y(1) = x(1)*exp(a*x(3)) + x(2)*exp(-a*x(3)) - A;
  y(2) = x(1)*exp(b*x(3)) + x(2)*exp(-b*x(3)) - B;
  y(3) = 4*x(1)*x(2)*x(3)^2-1;

endfunction

# Change [10;10;10] to the desired inital guess [x;y;z] for root-finding
[x, F, converged] = fsolve(@f, [10;10;10])
```