

# Lagrangian and Hamiltonian Formulations for Simulating a Double Compound Pendulum with Uniform Rods

Maurice-León Schwinger

The kinetic and potential energies are given by

$$T = \frac{1}{2}m(\dot{x}_1^2 + \dot{y}_1^2 + \dot{x}_2^2 + \dot{y}_2^2) + \frac{1}{2}I(\dot{\varphi}_1^2 + \dot{\varphi}_2^2) \quad (1)$$

$$U = mg(y_1 + y_2) \quad (2)$$

The center of mass for each pendulum is located at its center. Thus

$$\dot{x}_1^2 + \dot{y}_1^2 = \frac{\dot{\varphi}_1^2 l^2}{4}(\sin^2(\varphi_1) + \cos^2(\varphi_1)) \quad (3)$$

$$= \frac{\dot{\varphi}_1^2 l^2}{4} \quad (4)$$

$$\dot{x}_2^2 + \dot{y}_2^2 = l^2((\dot{\varphi}_1 \cos(\varphi_1) + \frac{1}{2}\dot{\varphi}_2 \cos^2(\varphi_2)) + (\dot{\varphi}_1 \sin(\varphi_1) + \frac{1}{2}\dot{\varphi}_2 \sin^2(\varphi_2))) \quad (5)$$

$$= l^2(\dot{\varphi}_1^2 + \frac{1}{4}\dot{\varphi}_2^2 + \dot{\varphi}_1 \dot{\varphi}_2 \cos(\varphi_1) \cos(\varphi_2) + \dot{\varphi}_1 \dot{\varphi}_2 \sin(\varphi_1) \sin(\varphi_2)) \quad (6)$$

$$= l^2(\dot{\varphi}_1^2 + \frac{1}{4}\dot{\varphi}_2^2 + \dot{\varphi}_1 \dot{\varphi}_2 \cos(\varphi_1 - \varphi_2)) \quad (7)$$

Plugging in the coordinates with respect to the angle as well as the moment of inertia for a thin rod rotated about its axis we get

$$T = \frac{m}{2} \left( \frac{\dot{\varphi}_1^2 l^2}{4} + l^2(\dot{\varphi}_1^2 + \frac{1}{4}\dot{\varphi}_2^2 + \dot{\varphi}_1 \dot{\varphi}_2 \cos(\varphi_1 - \varphi_2)) \right) + \frac{1}{2} \frac{1}{12} m l^2 (\dot{\varphi}_1^2 + \dot{\varphi}_2^2) \quad (8)$$

$$U = -mgl(\frac{1}{2} \cos(\varphi_1) + \cos(\varphi_1) + \frac{1}{2} \cos(\varphi_2)) \quad (9)$$

The Lagrangian  $L$  is given by

$$L = T - U \quad (10)$$

$$= \frac{ml^2}{2} \left( \frac{\dot{\varphi}_1^2}{4} + \dot{\varphi}_1^2 + \frac{1}{4}\dot{\varphi}_2^2 + \dot{\varphi}_1 \dot{\varphi}_2 \cos(\varphi_1 - \varphi_2) \right) + \frac{1}{24} ml^2 (\dot{\varphi}_1^2 + \dot{\varphi}_2^2) + \frac{mgl}{2} (3 \cos(\varphi_1) + \cos(\varphi_2)) \quad (11)$$

$$= \frac{ml^2}{24} (15\dot{\varphi}_1^2 + 3\dot{\varphi}_2^2 + 12\dot{\varphi}_1 \dot{\varphi}_2 \cos(\varphi_1 - \varphi_2)) + \frac{1}{24} ml^2 (\dot{\varphi}_1^2 + \dot{\varphi}_2^2) + \frac{mgl}{2} (3 \cos(\varphi_1) + \cos(\varphi_2)) \quad (12)$$

$$= \frac{ml^2}{6} (4\dot{\varphi}_1^2 + \dot{\varphi}_2^2 + 3\dot{\varphi}_1 \dot{\varphi}_2 \cos(\varphi_1 - \varphi_2)) + \frac{mgl}{2} (3 \cos(\varphi_1) + \cos(\varphi_2)) \quad (13)$$

Using the Euler-Lagrange equations we could solve for the Equations of Motion in the form of Lagrangian mechanics. But these we yield a second-order differential equation. Solving for the Equations of Motion in Hamilton mechanics is better for simulating, because you only get first-order differential equations, which can more be solved more accurately numerically. Therefore we perform a Legendre-Transform on the Lagrangian, which turns it into the Hamiltonian.

With the generalized momenta  $p = \frac{\partial L}{\partial \dot{\varphi}}$  we obtain

$$H = p\dot{\varphi} - L \quad (14)$$

where  $p$  and  $\dot{\varphi}$  are 2-dimensional vectors. First we solve for the generalized momenta.

$$p_1 = \frac{\partial L}{\partial \dot{\varphi}_1} \quad (15)$$

$$= \frac{8ml^2}{6} \dot{\varphi}_1 + \frac{3ml^2}{6} \dot{\varphi}_2 \cos(\varphi_1 - \varphi_2) \quad (16)$$

$$p_2 = \frac{\partial L}{\partial \dot{\varphi}_2} \quad (17)$$

$$= \frac{2ml^2}{6} \dot{\varphi}_2 + \frac{3ml^2}{6} \dot{\varphi}_1 \cos(\varphi_1 - \varphi_2) \quad (18)$$

With

$$K = \begin{pmatrix} \frac{4ml^2}{3} & \frac{ml^2}{2} \cos(\varphi_1 - \varphi_2) \\ \frac{ml^2}{2} \cos(\varphi_1 - \varphi_2) & \frac{ml^2}{3} \end{pmatrix} \quad (19)$$

this can be written as

$$\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = K \begin{pmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{pmatrix} \quad (20)$$

Since

$$\det(K) = m^2 l^4 \left( \frac{4}{9} - \frac{\cos^2(\varphi_1 - \varphi_2)}{4} \right) \quad (21)$$

$$= m^2 l^4 \left( \frac{16 - 9 \cos^2(\varphi_1 - \varphi_2)}{36} \right) \quad (22)$$

$$\geq 0 \quad (23)$$

K is invertible. The inverse matrix of a  $2 \times 2$ -matrix can be computed through

$$K^{-1} = \frac{1}{m^2 l^4 \left( \frac{4}{9} - \frac{\cos^2(\varphi_1 - \varphi_2)}{4} \right)} \begin{pmatrix} \frac{ml^2}{3} & -\frac{ml^2}{2} \cos(\varphi_1 - \varphi_2) \\ -\frac{ml^2}{2} \cos(\varphi_1 - \varphi_2) & \frac{4ml^2}{3} \end{pmatrix} \quad (24)$$

$$= \frac{36}{ml^2(16 - 9 \cos^2(\varphi_1 - \varphi_2))} \begin{pmatrix} \frac{1}{3} & -\frac{1}{2} \cos(\varphi_1 - \varphi_2) \\ -\frac{1}{2} \cos(\varphi_1 - \varphi_2) & \frac{4}{3} \end{pmatrix} \quad (25)$$

Because  $H = T + U$  and  $L = T - U$ ,  $T = \frac{1}{2} p \dot{\varphi}$ . Therefore  $T$  can be written as

$$T = \frac{1}{2} p K^{-1} p \quad (26)$$

The Hamiltonian can now be written as

$$H = \frac{1}{2} p K^{-1} p + U \quad (27)$$

$$= \frac{1}{2} p K^{-1} p - \frac{mgl}{2} (3 \cos(\varphi_1) + \cos(\varphi_2)) \quad (28)$$

$$= \frac{18}{ml^2(16 - 9 \cos^2(\varphi_1 - \varphi_2))} \left( \frac{1}{3} p_1^2 - \cos(\varphi_1 - \varphi_2) p_1 p_2 + \frac{4}{3} p_2^2 \right) - \frac{mgl}{2} (3 \cos(\varphi_1) + \cos(\varphi_2)) \quad (29)$$

$$= \frac{6(p_1^2 - 3 \cos(\varphi_1 - \varphi_2) p_1 p_2 + 4 p_2^2)}{ml^2(16 - 9 \cos^2(\varphi_1 - \varphi_2))} - \frac{mgl}{2} (3 \cos(\varphi_1) + \cos(\varphi_2)) \quad (30)$$

Now we can use the Hamilton equations to get the Equations of Motion as first-order differential equations. With  $\Delta = \varphi_1 - \varphi_2$

$$\frac{d\varphi_1}{dt} = \frac{\partial H}{\partial p_1} \quad (31)$$

$$= \frac{6(2p_1 - 3\cos(\Delta)p_2)}{ml^2(16 - 9\cos^2(\Delta))} \quad (32)$$

$$\frac{d\varphi_2}{dt} = \frac{\partial H}{\partial p_2} \quad (33)$$

$$= \frac{6(8p_2 - 3\cos(\Delta)p_1)}{ml^2(16 - 9\cos^2(\Delta))} \quad (34)$$

$$\frac{dp_1}{dt} = -\frac{\partial H}{\partial \varphi_1} \quad (35)$$

$$= -\frac{18(p_1p_2\sin(\Delta)(16 - 9\cos^2(\Delta)) - 6\sin(\Delta)\cos(\Delta)(p_1^2 + 4p_2^2 - 3p_1p_2\cos(\Delta)))}{ml^2(16 - 9\cos^2(\Delta))^2} - \frac{3mgl\sin(\varphi_1)}{2} \quad (36)$$

$$\frac{dp_2}{dt} = -\frac{\partial H}{\partial \varphi_2} \quad (37)$$

$$= \frac{18(p_1p_2\sin(\Delta)(16 - 9\cos^2(\Delta)) - 6\sin(\Delta)\cos(\Delta)(p_1^2 + 4p_2^2 - 3p_1p_2\cos(\Delta)))}{ml^2(16 - 9\cos^2(\Delta))^2} + \frac{mgl\sin(\varphi_2)}{2} \quad (38)$$

Finally we can simulate a trajectory of the system given initial conditions using Runge-Kutta 4 (RK4).

```

1 import numpy as np
2 from math import *
3
4 import matplotlib.pyplot as plt
5
6 def rk4(f, h, v, param):
7     k_1=f(v, param)
8     k_2=f(v+h/2*k_1, param)
9     k_3=f(v+h/2*k_2, param)
10    k_4=f(v+h*k_3, param)
11    return v+h/6*(k_1+2*k_2+2*k_3+k_4)
12
13 def double_pendulum(v, param=[1,1,9.81]):
14     phi1, phi2, p1, p2 = v
15     m, l, g = np.array(param)
16     cos_delta = cos(phi1-phi2)
17     sin_delta = sin(phi1-phi2)
18     c = (16-9*cos_delta*cos_delta)
19     a = 6/(m*l*c)
20     b = 3*a/c*(p1*p2*sin_delta*c-6*sin_delta*cos_delta*(p1**2+4*p2**2-3*p1*p2*cos_delta))
21     return np.array([
22         a*(2*p1-3*cos_delta*p2),
23         a*(8*p2-3*cos_delta*p1),
24         -b-3*m*g*l*sin(phi1)/2,
25         b+m*g*l*sin(phi2)/2
26     ])
27
28 def generate_trajectory_rk4(f, h, n, v, param):
29     traj = [v]
30     for x in range(n):
31         v = rk4(f, h, v, param)
32         traj.append(v)
33     return np.array(traj)
34
35 traj = generate_trajectory_rk4(double_pendulum, 0.01, 10000, [pi/4,pi/4,1,1], [1,1,9.81])
36 print(traj)
37 plt.plot(traj[:,1],traj[:,2])
38 plt.show()

```

ImplementationRK4.py

It's trivial to see, that the double pendulum stays bounded on the  $\varphi_1$ - $\varphi_2$ -plane. This can be useful when analysing the phase-space of the system.