Computational Physics (PHYS6350)

Lecture 13: Classical mechanics problems

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j} = 0$$

February 27, 2025

Instructor: Volodymyr Vovchenko (vvovchenko@uh.edu)

Course materials: https://github.com/vlvovch/PHYS6350-ComputationalPhysics/tree/spring2025

Three-body problem

Exercise 8.10 (M. Newman, *Computational Physics*)

Three stars interacting through gravitational force

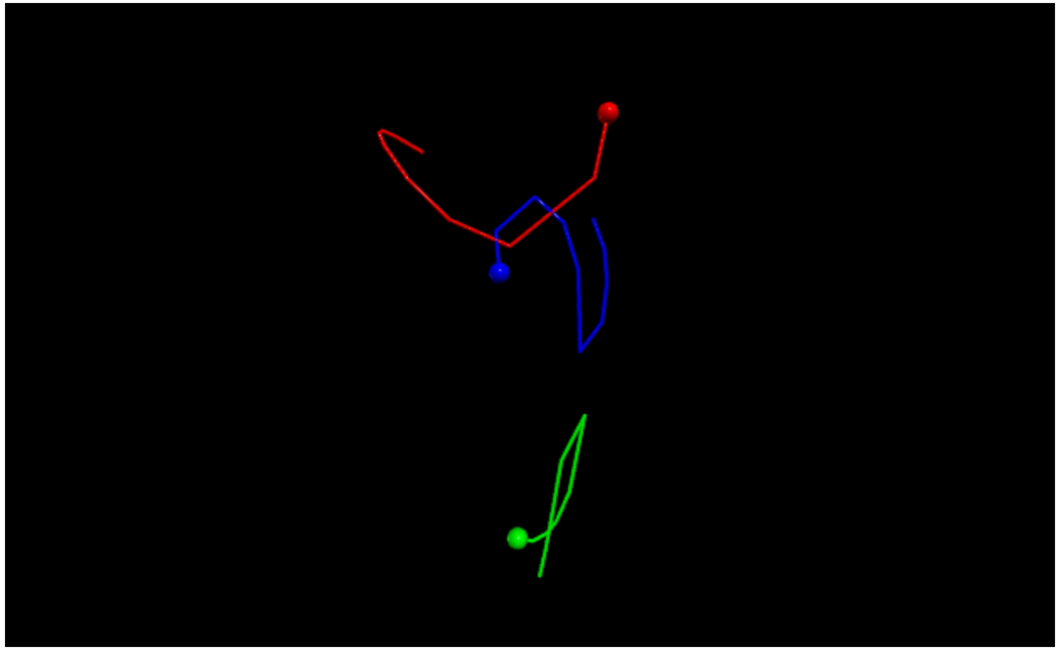
The equations of motion are

$$\frac{d^2\mathbf{r}_1}{dt} = Gm_2\frac{\mathbf{r}_2 - \mathbf{r}_1}{|\mathbf{r}_2 - \mathbf{r}_1|^3} + Gm_3\frac{\mathbf{r}_3 - \mathbf{r}_1}{|\mathbf{r}_3 - \mathbf{r}_1|^3},$$

$$\frac{d^2\mathbf{r}_2}{dt} = Gm_1\frac{\mathbf{r}_1 - \mathbf{r}_2}{|\mathbf{r}_1 - \mathbf{r}_2|^3} + Gm_3\frac{\mathbf{r}_3 - \mathbf{r}_2}{|\mathbf{r}_3 - \mathbf{r}_2|^3},$$

$$\frac{d^2\mathbf{r}_3}{dt} = Gm_1\frac{\mathbf{r}_1 - \mathbf{r}_3}{|\mathbf{r}_1 - \mathbf{r}_3|^3} + Gm_2\frac{\mathbf{r}_2 - \mathbf{r}_3}{|\mathbf{r}_2 - \mathbf{r}_3|^3}.$$

Name	Mass	x	y
Star 1	150.	3	1
Star 2	200.	-1	-2
Star 3	250.	-1	1



Cannot be solved analytically!

Take $G = 1$ (dimensionless)

Initially at rest and move in plane $z = 0$

Initial coordinates: $\boldsymbol{r}_1 = (3,1)$, $\boldsymbol{r}_2 = (-1,-2)$, $\boldsymbol{r}_3 = (-1,1)$

Three-body problem

```python
def fthreebody(xin, t):
    global f_evaluations
    f_evaluations += 1

    x1 = xin[0]
    y1 = xin[1]
    x2 = xin[2]
    y2 = xin[3]
    x3 = xin[4]
    y3 = xin[5]

    r12 = np.sqrt((x1-x2)**2 + (y1-y2)**2)
    r13 = np.sqrt((x1-x3)**2 + (y1-y3)**2)
    r23 = np.sqrt((x2-x3)**2 + (y2-y3)**2)


    return np.array([xin[6],xin[7],xin[8],xin[9],xin[10],xin[11],
                     G * m2 * (x2 - x1) / r12**3 + G * m3 * (x3 - x1) / r13**3,
                     G * m2 * (y2 - y1) / r12**3 + G * m3 * (y3 - y1) / r13**3,
                     G * m1 * (x1 - x2) / r12**3 + G * m3 * (x3 - x2) / r23**3,
                     G * m1 * (y1 - y2) / r12**3 + G * m3 * (y3 - y2) / r23**3,
                     G * m1 * (x1 - x3) / r13**3 + G * m2 * (x2 - x3) / r23**3,
                     G * m1 * (y1 - y3) / r13**3 + G * m2 * (y2 - y3) / r23**3
                    ]
                   )
```

Run from $t = 0$ to 5

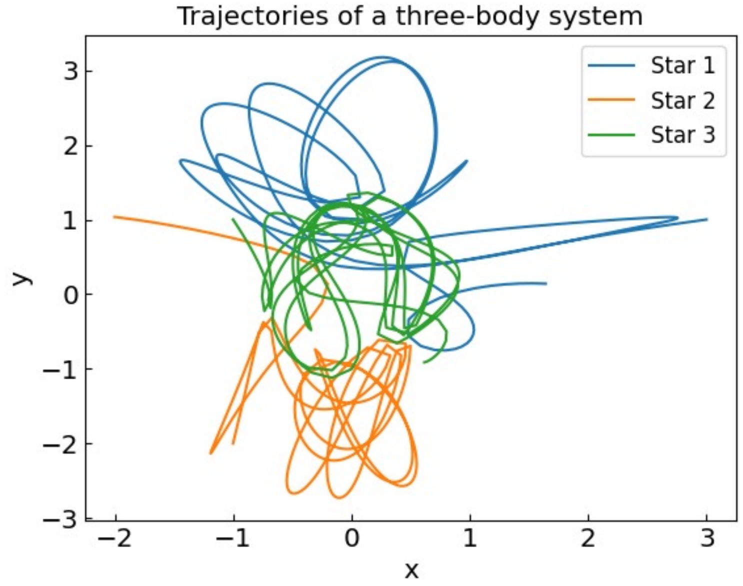
Three-body problem

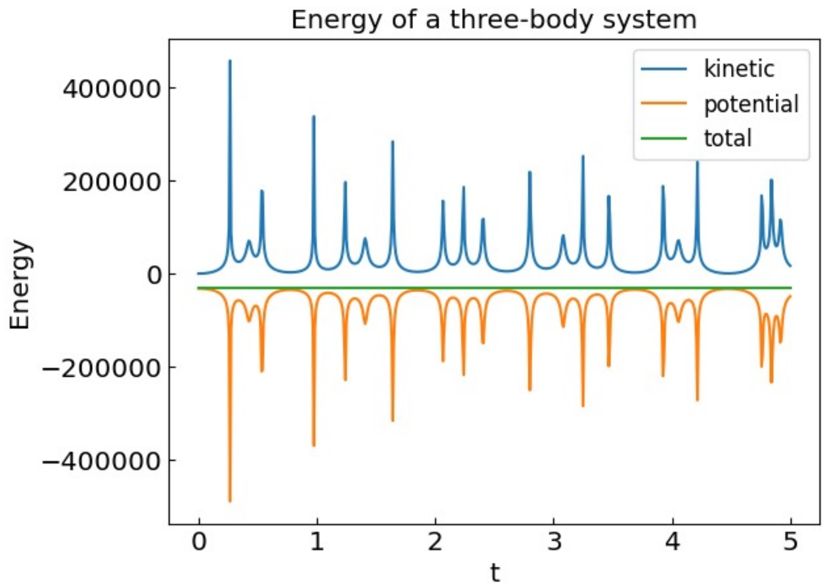
```python
def fthreebody(xin, t):
    global f_evaluations
    f_evaluations += 1

    x1 = xin[0]
    y1 = xin[1]
    x2 = xin[2]
    y2 = xin[3]
    x3 = xin[4]
    y3 = xin[5]

    r12 = np.sqrt((x1-x2)**2 + (y1-y2)**2)
    r13 = np.sqrt((x1-x3)**2 + (y1-y3)**2)
    r23 = np.sqrt((x2-x3)**2 + (y2-y3)**2)


    return np.array([xin[6],xin[7],xin[8],xin[9],xin[10],xin[11],
                     G * m2 * (x2 - x1) / r12**3 + G * m3 * (x3 - x1) / r13**3,
                     G * m2 * (y2 - y1) / r12**3 + G * m3 * (y3 - y1) / r13**3,
                     G * m1 * (x1 - x2) / r12**3 + G * m3 * (x3 - x2) / r23**3,
                     G * m1 * (y1 - y2) / r12**3 + G * m3 * (y3 - y2) / r23**3,
                     G * m1 * (x1 - x3) / r13**3 + G * m2 * (x2 - x3) / r23**3,
                     G * m1 * (y1 - y3) / r13**3 + G * m2 * (y2 - y3) / r23**3
                    ]
                   )
```





Run from t = 0 to 5

final project idea(?): simulate the solar system

Lagrangian mechanics

In Lagrangian mechanics, a classical system with N degrees of freedom is characterized by generalized coordinates q_j that obey Euler-Lagrange equations of motion:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j} = 0, \qquad j = 1 \dots N$$

L is the non-relativistic Lagrangian which is typically defined as the difference of kinetic and potential energies, $L = T - V$.
One can rewrite these equations using the chain rule:

$$\sum_{i=1}^{N} \frac{\partial^2 L}{\partial \dot{q}_j \partial \dot{q}_i} \ddot{q}_i = -\sum_{i=1}^{N} \frac{\partial^2 L}{\partial \dot{q}_j \partial q_i} \dot{q}_i - \frac{\partial^2 L}{\partial \dot{q}_j \partial t} + \frac{\partial L}{\partial q_j}, \qquad j = 1 \dots N.$$

This is a system of N linear equations for $\ddot{q}_j$. When solved we have

$$\ddot{q}_i = f_i(\{q_j\}, \{\dot{q}_j\}, t).$$

or

$$\frac{dq_i}{dt} = \dot{q}_i,$$
$$\frac{d\dot{q}_i}{dt} = f_i(\{q_j\}, \{\dot{q}_j\}, t),$$

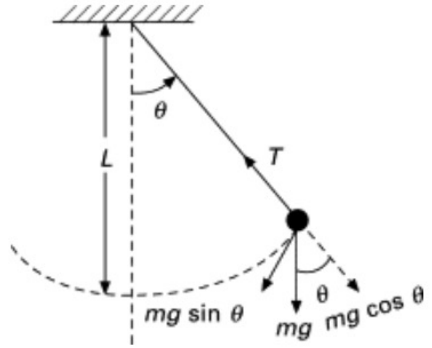
which can be solved using standard methods.

Non-linear pendulum

The generalized coordinate is the displacement angle θ

$$x = l\sin(\theta),$$

$$y = -l\cos(\theta),$$



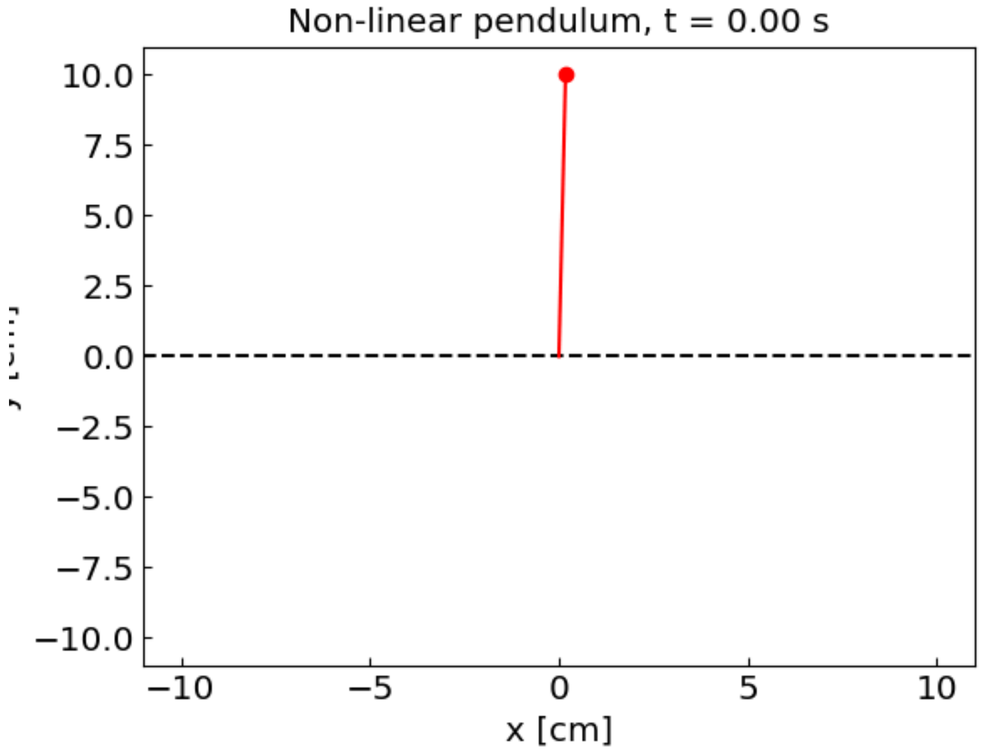
Lagrangian

$$L = T - V = \frac{ml^2\dot{\theta}^2}{2} - mgl\cos(\theta),$$

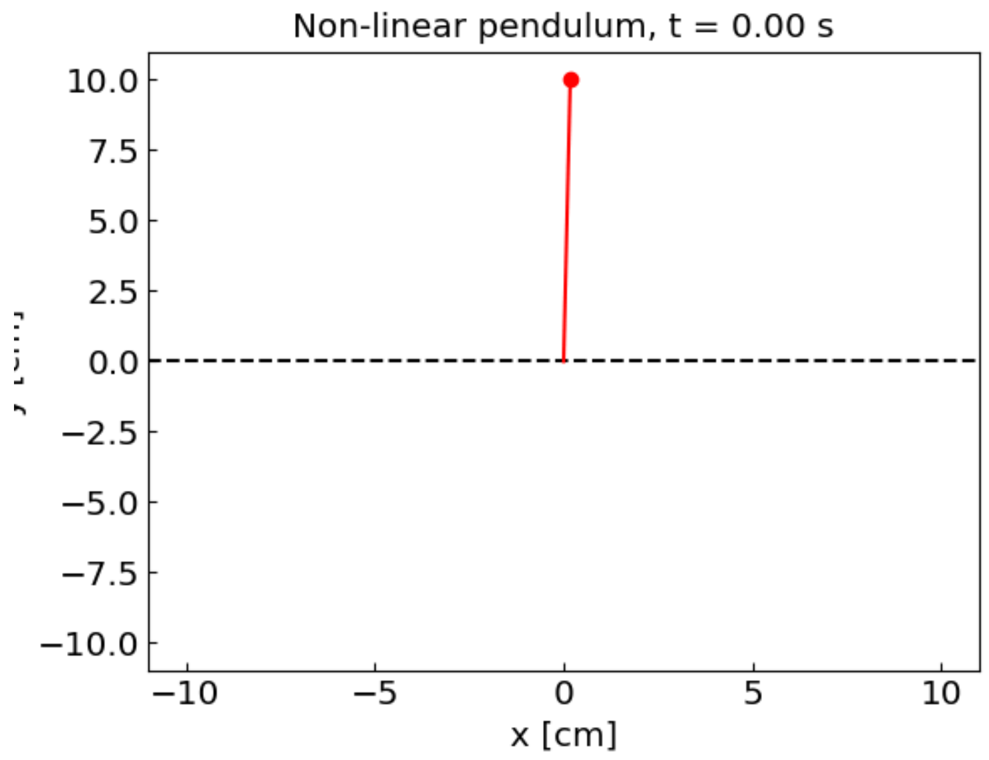
Equations of motion

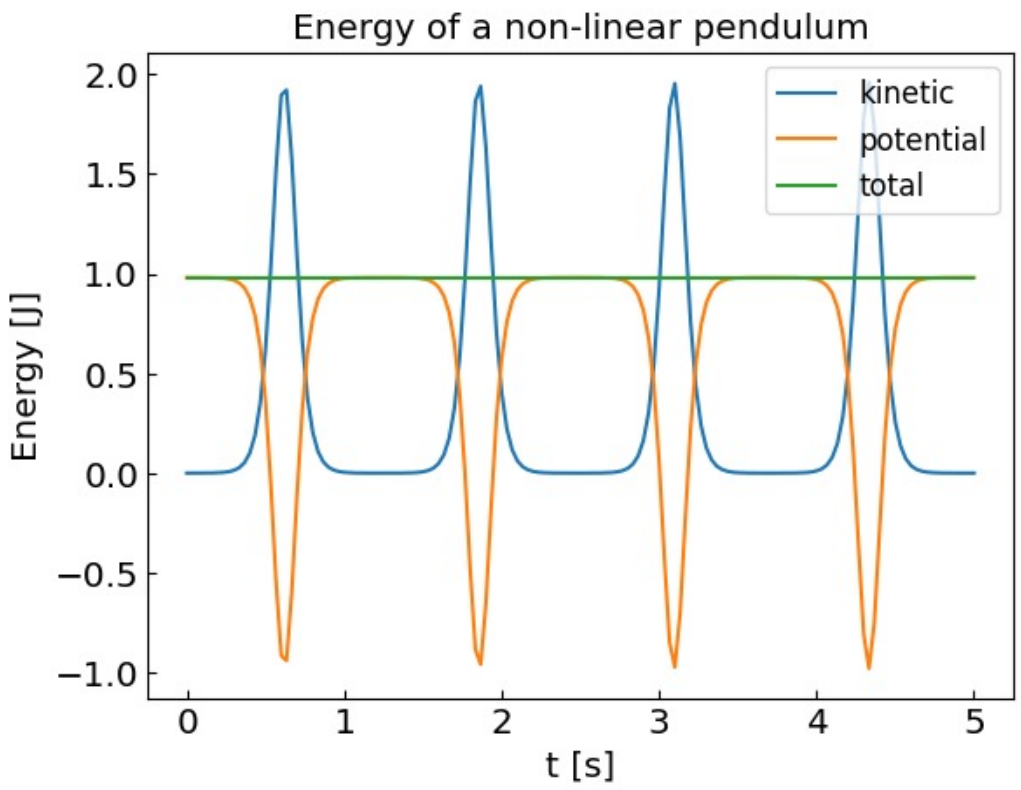
$$ml\ddot{\theta} = -mgl\sin\theta.$$

Non-linear pendulum



Non-linear pendulum



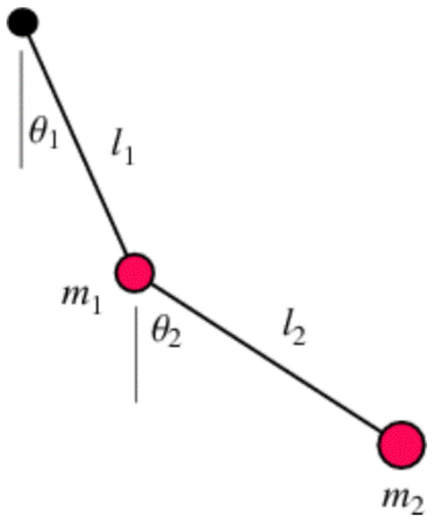


Double pendulum

Double pendulum is the simplest system exhibiting chaotic motion – *deterministic chaos*

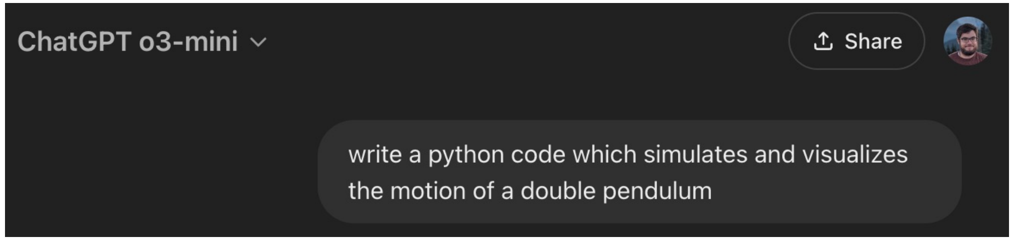
Two degrees of freedom: the displacement angles θ_1 and θ_2

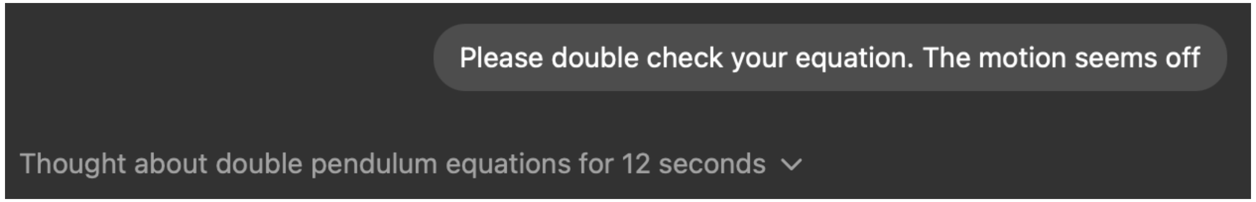
$$
\begin{aligned}
x_1 &= l_1 \sin(\theta_1), \\
y_1 &= -l_1 \cos(\theta_1), \\
x_2 &= l_1 \sin(\theta_1) + l_2 \sin(\theta_2), \\
y_2 &= -l_1 \cos(\theta_1) - l_2 \cos(\theta_2).
\end{aligned}
$$

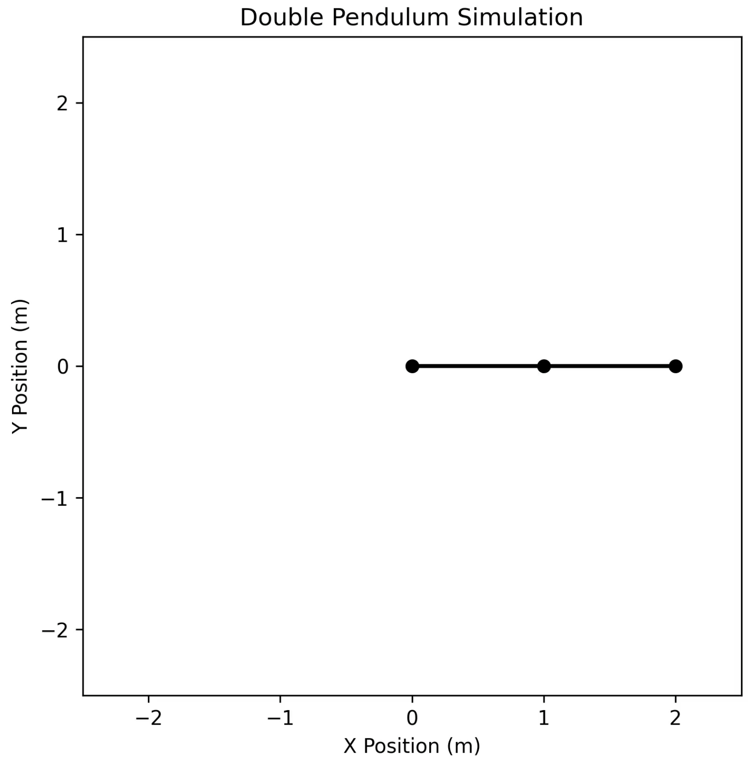


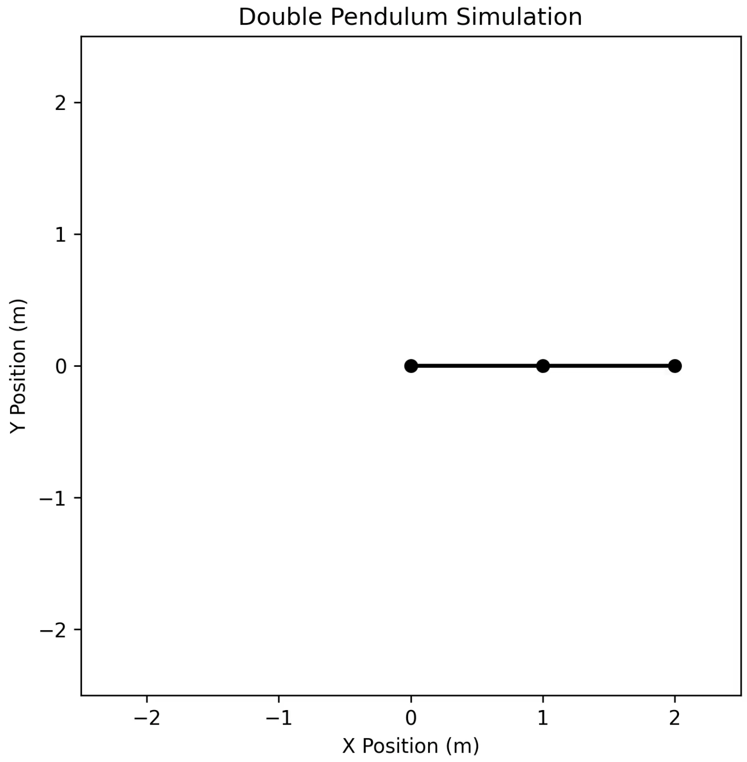
Double pendulum vs ChatGPT

First approach nowadays: try LLM



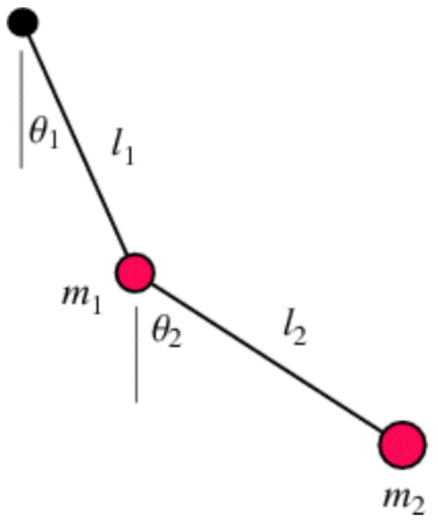






Double pendulum

Double pendulum is the simplest system exhibiting chaotic motion – *deterministic chaos*



Two degrees of freedom: the displacement angles θ_1 and θ_2

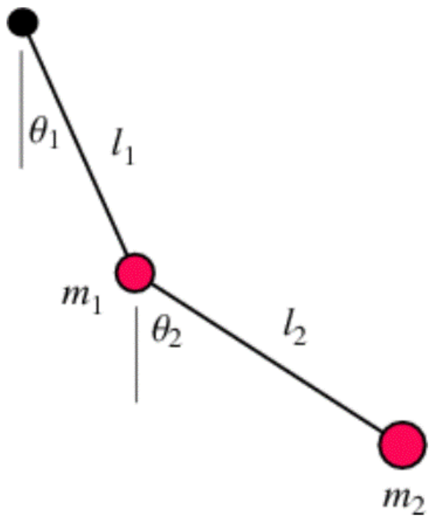
$$\begin{aligned}
x_1 &= l_1 \sin(\theta_1),\\
y_1 &= -l_1 \cos(\theta_1),\\
x_2 &= l_1 \sin(\theta_1) + l_2 \sin(\theta_2),\\
y_2 &= -l_1 \cos(\theta_1) - l_2 \cos(\theta_2).
\end{aligned}$$

$$T = \frac{m_1 \dot{x}_1^2}{2} + \frac{m_2 \dot{x}_2^2}{2} = \frac{1}{2} m_1 l_1^2 \dot{\theta}_1 + \frac{1}{2} m_2 \left[l_1^2 \dot{\theta}_1^2 + l_2^2 \dot{\theta}_2^2 + 2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2), \right]$$ kinetic energy

$$V = m_1 g y_1 + m_2 g y_2 = -(m_1 + m_2) g l_1 \cos(\theta_1) - m_2 g l_2 \cos(\theta_2).$$ potential energy

Double pendulum

Double pendulum is the simplest system exhibiting chaotic motion – *deterministic chaos*



Two degrees of freedom: the displacement angles θ_1 and θ_2

$$
\begin{aligned}
x_1 &= l_1 \sin(\theta_1),\\
y_1 &= -l_1 \cos(\theta_1),\\
x_2 &= l_1 \sin(\theta_1) + l_2 \sin(\theta_2),\\
y_2 &= -l_1 \cos(\theta_1) - l_2 \cos(\theta_2).
\end{aligned}
$$

$$T = \frac{m_1 \dot{x}_1^2}{2} + \frac{m_2 \dot{x}_2^2}{2} = \frac{1}{2} m_1 l_1^2 \dot{\theta}_1 + \frac{1}{2} m_2 \left[l_1^2 \dot{\theta}_1^2 + l_2^2 \dot{\theta}_2^2 + 2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2), \right]$$ kinetic energy

$$V = m_1 g y_1 + m_2 g y_2 = -(m_1 + m_2) g l_1 \cos(\theta_1) - m_2 g l_2 \cos(\theta_2).$$ potential energy

The Lagrange equations of motion read

$$
\begin{aligned}
(m_1 + m_2) l_1 \ddot{\theta}_1 + m_2 l_2 \cos(\theta_1 - \theta_2) \ddot{\theta}_2 &= -m_2 l_1 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) - (m_1 + m_2) g \sin(\theta_1),\\
m_2 l_1 \cos(\theta_1 - \theta_2) \ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 &= m_2 l_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) - m_2 g \sin(\theta_2).
\end{aligned}
$$

This is a system of two linear equations for $\ddot{\theta}_{1,2}$ that can be solved straightforwardly.

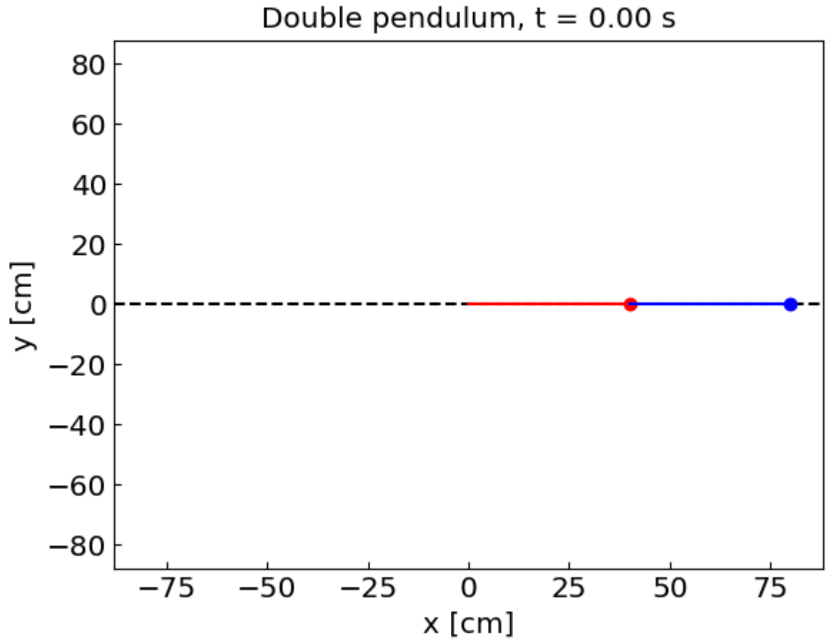
Double pendulum

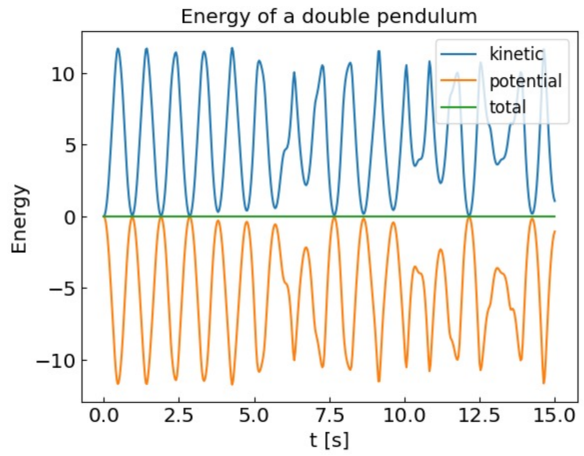
```python
g = 9.81
l1 = 0.4
l2 = 0.4
m1 = 1.0
m2 = 1.0

def fdoublependulum(xin, t):
    global f_evaluations
    f_evaluations += 1

    theta1 = xin[0]
    theta2 = xin[1]
    omega1 = xin[2]
    omega2 = xin[3]
    a1 = (m1 + m2)*l1
    b1 = m2*l2*np.cos(theta1 - theta2)
    c1 = m2*l2*omega2*omega2*np.sin(theta1 - theta2) + (m1 + m2)*g*np.sin(theta1)
    a2 = m2*l1*np.cos(theta1 - theta2)
    b2 = m2*l2;
    c2 = -m2*l1*omega1*omega1*np.sin(theta1 - theta2) + m2*g*np.sin(theta2) # + k
    domega1 = - ( c2/b2 - c1/b1 ) / ( a2/b2 - a1/b1 )
    domega2 = - ( c2/a2 - c1/a1 ) / ( b2/a2 - b1/a1 )
    return np.array([omega1,
                     omega2,
                     domega1,
                     domega2
                    ])
```

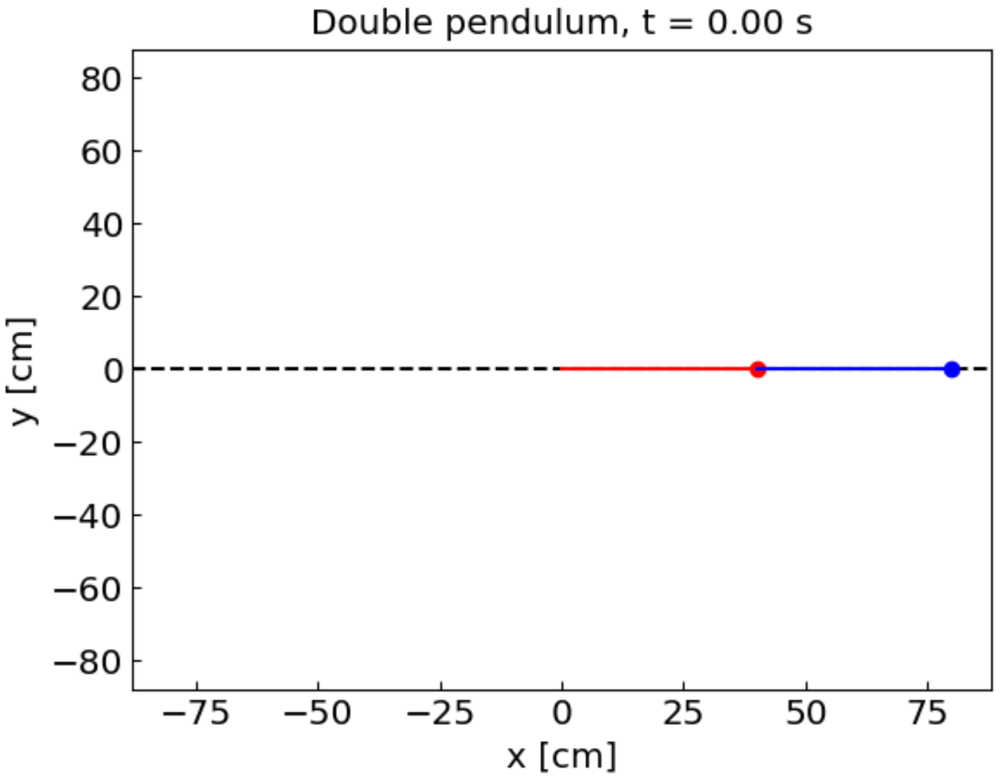
```python
sol = bulirsch_stoer(fpendulum, x0, t0, 1, tend, eps, error_definition_pendulum, 10)
```



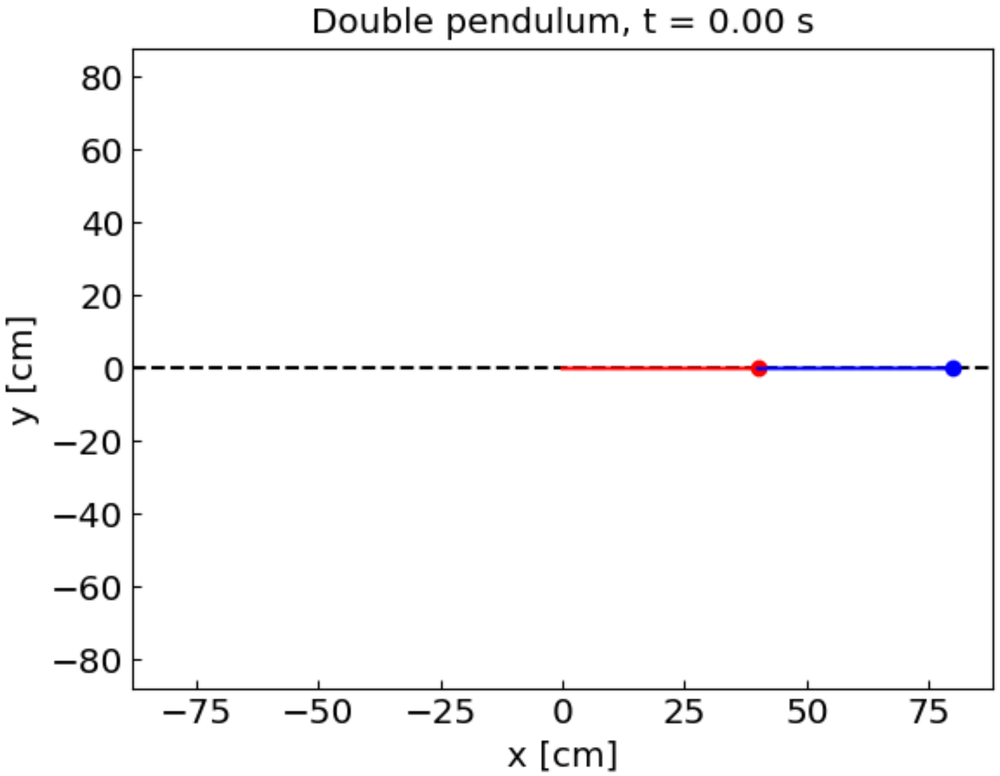


Double pendulum: chaotic behavior

$$\theta_1^0 = \theta_2^0 = \pi/2$$



$$\theta_1^0 = \theta_2^0 = \pi/2 + 10^{-4}$$



Double pendulum: chaotic behavior

$$\theta_1^0 = \theta_2^0 = \pi/2 \qquad\qquad \theta_1^0 = \theta_2^0 = \pi/2 + 10^{-4}$$