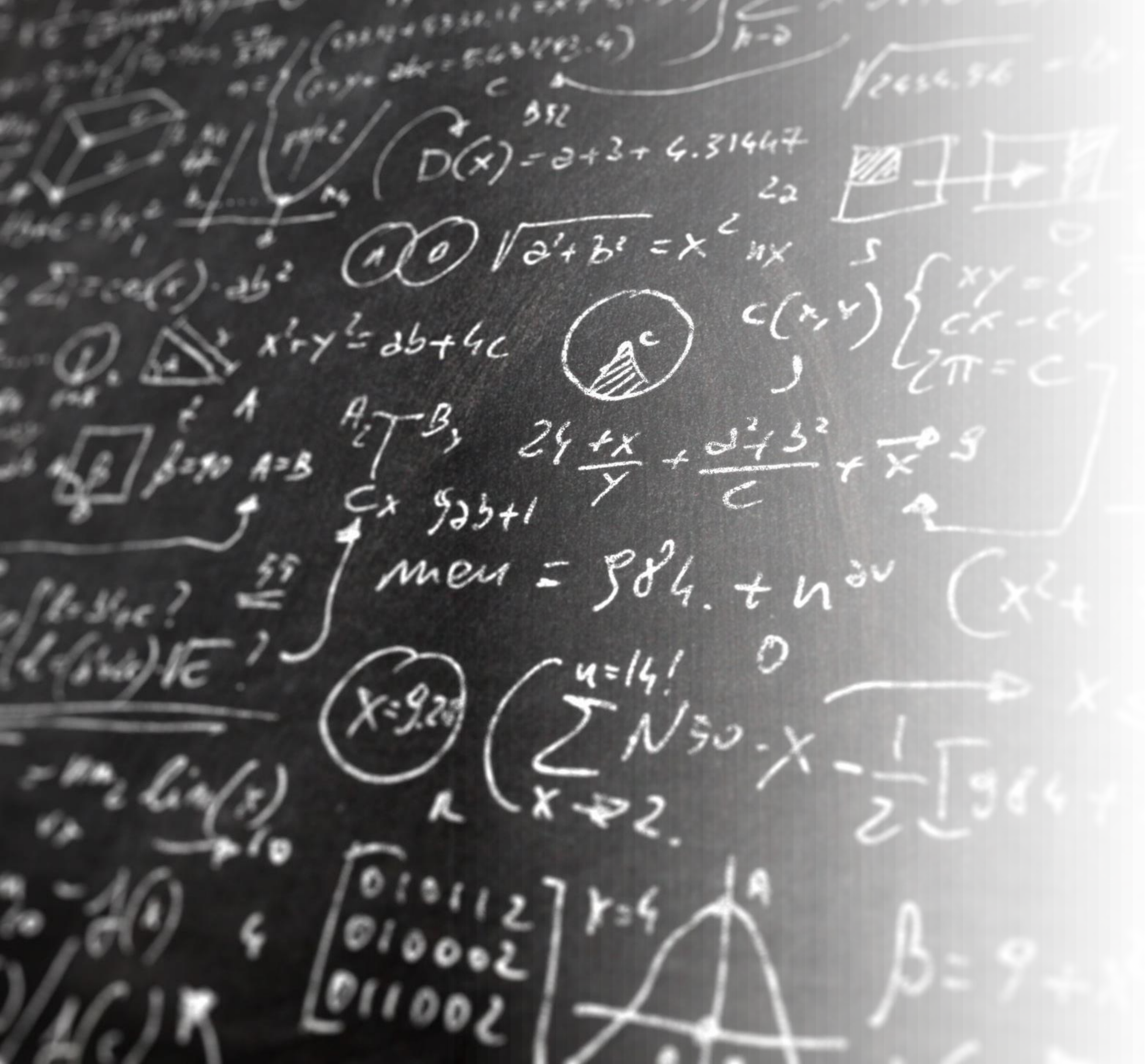
Theoretical Physics Group Project
Chaos in the Double Pendulum

Individual Presentation

2547473m



Introduction & Overview

1. Pendulum motions and modelling
2. Simple pendulum assistance/researches
3. Double pendulum
   - Equation of motion
   - Solution plan
   - Code of the simulation (outline)
   - Code of the automated plot

Simple pendulum

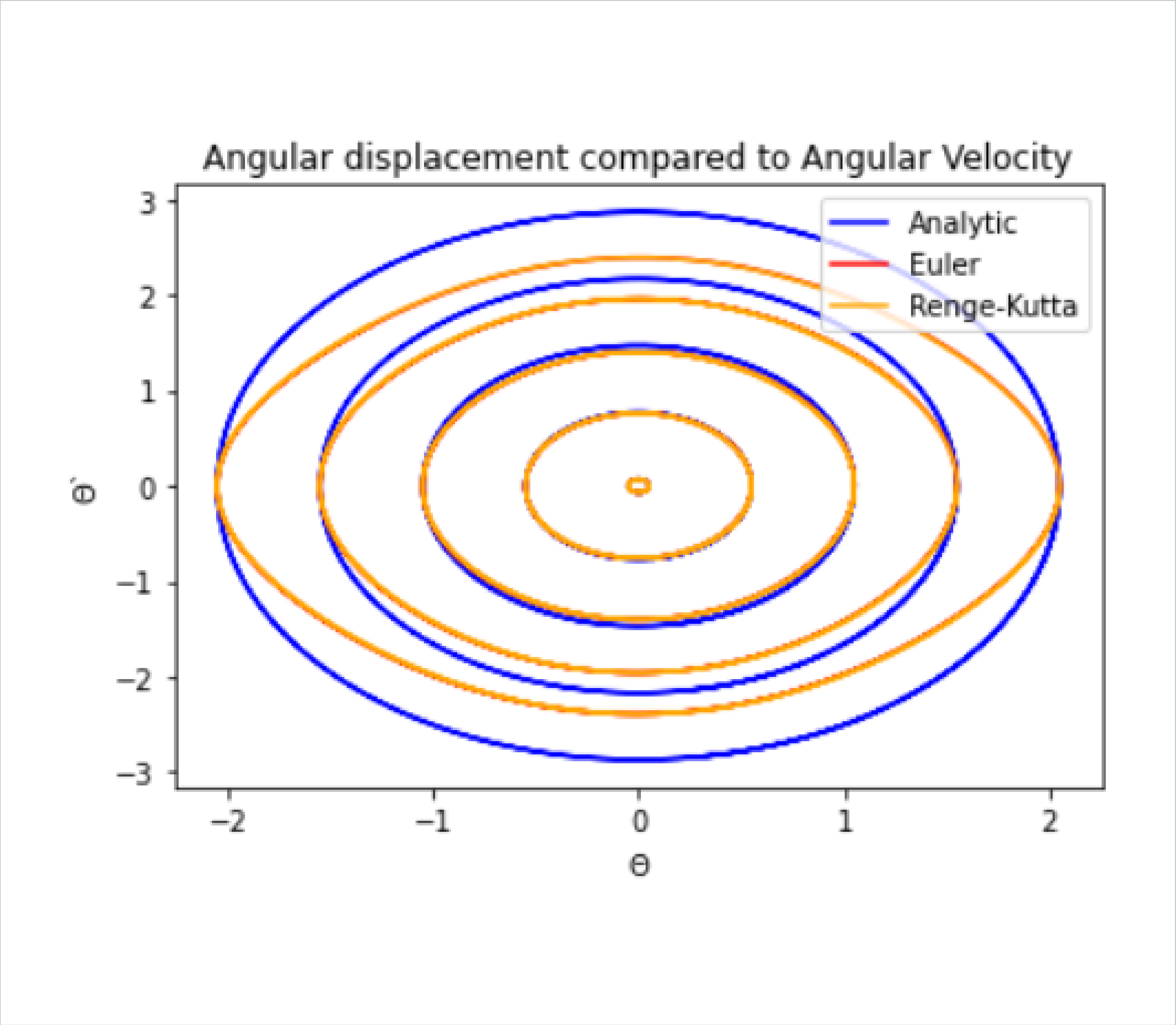
1. Equation of motion
    - Torque consideration
2. Solution derivation
    - Auxiliary Equation
    - Simple Ansatz
3. Analytical Solution

$$\theta = \theta_0 \cos\left(\sqrt{\frac{g}{L}}\,t\right)$$

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\sin(\theta)$$

Simple Pendulum

1. Simulation and Plotting
2. Numerical examples
   - Euler
   - Runge-Kutta



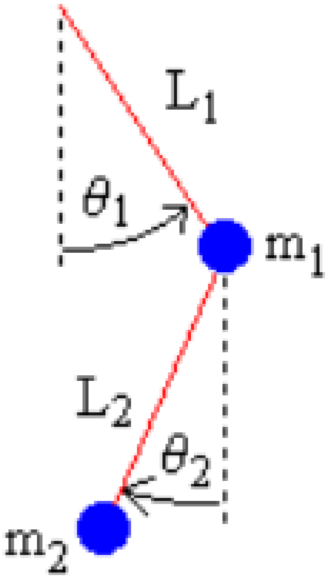
Double Pendulum

$$x_1 = L_1 \sin\theta_1$$

$$y_1 = -L_1 \cos\theta_1$$

$$x_2 = x_1 + L_2 \sin\theta_2$$

$$y_2 = y_1 - L_2 \cos\theta_2$$



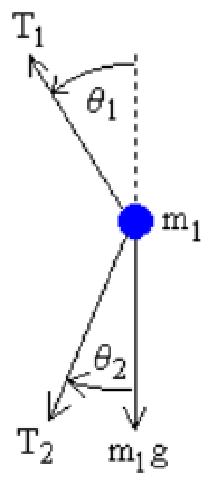
Equation of motion:

1. Separations
   - Upper / Lower
2. Kinematic Analysis
   - Cartesian Coordinates
3. Take Derivatives
   - Acceleration
   - x_1'' y_1'' x_2'' x_2''

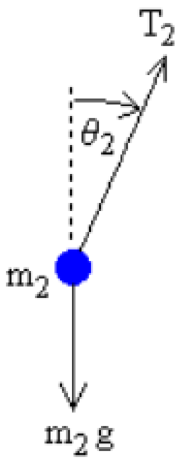
Double pendulum

Equation of motion:

1. Force analysis
   - 4 simultaneous equations
   - + 4 from previous
2. Equation Evaluation
   - Substitute the accelerations
   - Rearrange to eliminate T_2 T_1



upper mass



lower mass

$$m_1 x_1'' = -T_1 \sin\theta_1 + T_2 \sin\theta_2$$

$$m_1 y_1'' = T_1 \cos\theta_1 - T_2 \cos\theta_2 - m_1 g$$

$$m_2 x_2'' = -T_2 \sin\theta_2$$

$$m_2 y_2'' = T_2 \cos\theta_2 - m_2 g$$

$$l_1\left[(m_1+m_2)\left(g\sin(\theta_1)+\ddot{\theta}_1 l_1\right)+\ddot{\theta}_2 l_2 m_2\cos(\theta_1-\theta_2)+\dot{\theta}_2^2 l_2 m_2\sin(\theta_1-\theta_2)\right]=0$$

$$l_2 m_2\left[g\sin(\theta_2)+\ddot{\theta}_1 l_1\cos(\theta_1-\theta_2)+\dot{\theta}_1^2 l_1\left(-\sin(\theta_1-\theta_2)\right)+\ddot{\theta}_2 l_2\right]=0$$

Double Pendulum

1. Equation of Motion
   - Two coupled 2^{nd} order ODEs
   - Relatively complex
   - No simple Ansatz
2. Numerical Analysis

Double pendulum

Numerical Analysis:

1. Solution Plan
   - Runge-Kutta 4
   - Convert into 1st order ODEs
   - $2 \times 2 = 4$ equations

$$\dot{\theta}_1 = \omega_1$$

$$\dot{\theta}_2 = \omega_2$$

$$\omega_1' = \frac{-g\,(2\,m_1 + m_2)\sin\theta_1 - m_2\,g\sin(\theta_1 - 2\,\theta_2) - 2\sin(\theta_1 - \theta_2)\,m_2\,(\omega_2^2\,L_2 + \omega_1^2\,L_1\cos(\theta_1 - \theta_2))}{L_1\,(2\,m_1 + m_2 - m_2\cos(2\,\theta_1 - 2\,\theta_2))}$$

$$\omega_2' = \frac{2\sin(\theta_1 - \theta_2)\,(\omega_1^2\,L_1\,(m_1 + m_2) + g(m_1 + m_2)\cos\theta_1 + \omega_2^2\,L_2\,m_2\cos(\theta_1 - \theta_2))}{L_2\,(2\,m_1 + m_2 - m_2\cos(2\,\theta_1 - 2\,\theta_2))}$$

Double Pendulum

Numerical Analysis:

1. Four Solutions
   - Initial Conditions
   - Decide step-size h
   - Evaluate solution of next instant
2. Specific for Runge-Kutta 4
   - ω_1 & ω_2
   - Identify k-values

$$\omega_2(t_{n+1}) = \omega_2(t_n) + \frac{k_{1_{\omega_2}}}{6} + \frac{k_{2_{\omega_2}}}{3} + \frac{k_{3_{\omega_2}}}{3} + \frac{k_{4_{\omega_2}}}{6}$$

$$\omega_1(t_{n+1}) = \omega_1(t_n) + \frac{k_{1_{\omega_1}}}{6} + \frac{k_{2_{\omega_1}}}{3} + \frac{k_{3_{\omega_1}}}{3} + \frac{k_{4_{\omega_1}}}{6}$$

$$\theta_2(t_{n+1}) = \theta_2(t_n) + h * \omega_2(t_n)$$

$$\theta_1(t_{n+1}) = \theta_1(t_n) + h * \omega_1(t_n)$$

Double Pendulum

Numerical Solution:

1. Identify k-values
   - Slopes at various points
   - Dependent on steps
2. 8 k-values
   - Corresponds to ω_1 & ω_2

$$k_{4_{\omega_1}} = h\omega_1(\theta_1, \theta_2, \omega_1 + \dot{k}_{3_{\omega_1}}, \omega_2 + k_{3_{\omega_2}})$$

$$k_{3_{\omega_1}} = h\omega_1(\theta_1, \theta_2, \omega_1 + \frac{\dot{k}_{2_{\omega_1}}}{2}, \omega_2 + \frac{k_{2_{\omega_2}}}{2})$$

$$k_{2_{\omega_1}} = h\omega_1(\theta_1, \theta_2, \omega_1 + \frac{\dot{k}_{1_{\omega_1}}}{2}, \omega_2 + \frac{k_{1_{\omega_2}}}{2})$$

$$k_{1_{\omega_1}} = h\dot{\omega}_1$$

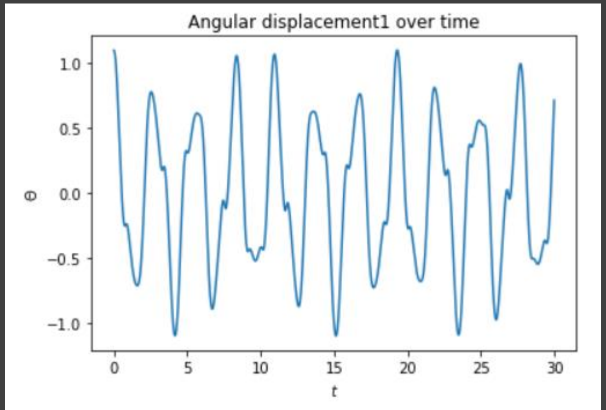
```python
#calculating k4 for all values
k4omega1 = step*kfunc1(con.arg1, con.arg2,
k4omega2 = step*kfunc2(con.arg1, con.arg2,
```

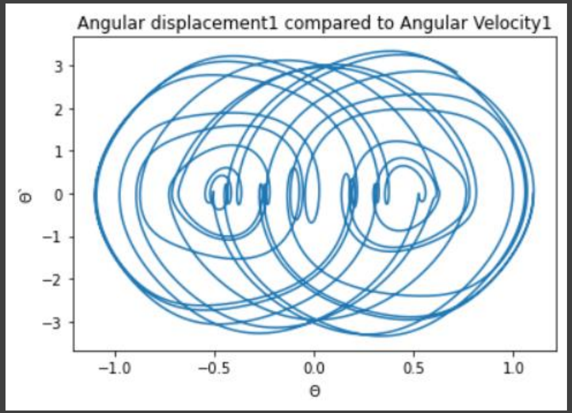
Double Pendulum

1. Simulation code (Outline)
   - Write-in from Plan
   - Create functions of equations
2. Bartosz completed the Code
   - Error fixed
   - Rearranging and Refinements (Classes)

```python
#functions for calculating angular acceleration
def alpha_1(g, l1, l2, m1, m2, th1, th2, om1, om2):
    a1 = (-g*(2*m1 + m2)*np.sin(th1) - m2*g*np.sin(t
    return a1

def alpha_2(g, l1, l2, m1, m2, th1, th2, om1, om2):
    a2 = (2*np.sin(th1 - th2)*((om1**2)*l1*(m1 + m2)
    return a2
```






```python
class Initial_Conditions:
    def __init__(init, arg1, arg2, arg3, arg4, arg5, arg6, arg7, arg8, arg9, arg10, arg11):
        """
        Args: in numeric order specific for Double Pendulum
            g: gravity
            l1: length of upper pendulum in meter
            l2: length of lower pendulum in meter
            m1: attached mass on the upper pendulum in kg
            m2: attached mass on the lower pendulum in kg
            theta1: initial angular space of the upper pendulum theta1, at t = 0
            theta2: initial angular space of the lower pendulum theta2, at t = 0
            omega1: initial angular velocity of the upper pendulum theta1, at t = 0
            omega2: initial angular velocity of the lower pendulum theta2, at t = 0
            s: time step for the simulation
            t: end time of the simulation
        Return:
            array of the above arguments
        """
        init.arg1 = arg1
        init.arg2 = arg2
        init.arg3 = arg3
        init.arg4 = arg4
        init.arg5 = arg5
        init.arg6 = arg6
        init.arg7 = arg7
        init.arg8 = arg8
        init.arg9 = arg9
        init.arg10 = arg10
        init.arg11 = arg11
```

Double Pendulum

1. Automated Plotter
    - Created Python Library
    - Wrote Plotting Notebook
2. Variation Plots
    - Angular displacement/velocity
3. Nathan Weir Refined the Notebook
    - Pendulum trajectories

Additions

- Potential Upgrades:
  1. Animations
- Thank you for listening!