Mass-Spring System ODE Solver: A Comparative Study of Numerical Methods

Prakriti Singh

September 28, 2024

Abstract

This report presents an analysis of a mass-spring system using two numerical methods: Euler's method and the 4th-order Runge-Kutta method. We investigate the system's behavior for both series and parallel spring configurations, comparing our implemented algorithms against analytical solutions and external numerical solvers (scipy). The study aims to validate the implementations, assess their accuracy, and explore their limitations.

1 Introduction

Mass-spring systems are fundamental in physics, serving as a basis for understanding oscillatory motion and more complex mechanical systems. This project focuses on solving the ordinary differential equations (ODEs) that describe the motion of a mass connected to two springs, which can be arranged either in series or parallel.

1.1 Physical Problem Description

We consider a system consisting of a mass m connected to two springs with spring constants k_1 and k_2. The springs can be arranged in two configurations:

1. Series: The springs are connected end-to-end, with the mass attached to the free end of the second spring.

2. Parallel: Both springs are attached directly to the mass on one end and a fixed point on the other.

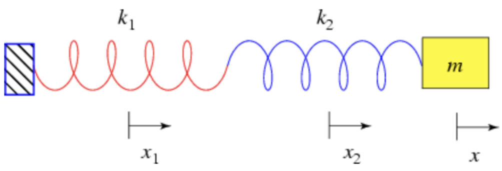

Figure 1: Series

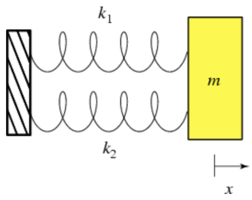

Figure 2: Parallel

The equation of motion for this system is given by:

$$m\frac{d^2x}{dt^2} = -k_{eq}x \tag{1}$$

where k_{eq} is the equivalent spring constant, which depends on the configuration:

- For series: $\frac{1}{k_{eq}} = \frac{1}{k_1} + \frac{1}{k_2}$
- For parallel: $k_{eq} = k_1 + k_2$

1.2 Numerical Methods Implemented

To solve this ODE, we have implemented two numerical methods:

1. Euler's Method
2. 4th-order Runge-Kutta Method (RK4)

These methods will be compared against the analytical solution and scipy for accuracy and performance check.

2 Methodology

2.1 Euler's Method

Euler's method is a first-order numerical procedure for solving ODEs. For our second-order ODE, we use a system of two first-order ODEs:

$$\frac{dx}{dt} = v \tag{2}$$

$$\frac{dv}{dt} = -\frac{k_{eq}}{m}x \tag{3}$$

The update equations for each time step are:

$$x_{n+1} = x_n + v_n \Delta t \tag{4}$$

$$v_{n+1} = v_n - \frac{k_{eq}}{m} x_n \Delta t \tag{5}$$

2.2 4th-order Runge-Kutta Method (RK4)

The 4th-order Runge-Kutta method (RK4) is a higher-order numerical integration technique that provides improved accuracy compared to simpler methods like Euler's method. For our second-order ODE, we again use a system of two first-order ODEs:

$$\frac{dx}{dt} = v \tag{6}$$

$$\frac{dv}{dt} = -\frac{k_{eq}}{m} x \tag{7}$$

The RK4 method calculates four intermediate slopes (k for position, l for velocity) at different points within each time step. These slopes are then combined to update the position and velocity. The process for each time step is as follows:

$$k_1 = v_n \Delta t \tag{8}$$

$$l_1 = -\frac{k_{eq}}{m} x_n \Delta t \tag{9}$$

$$k_2 = (v_n + \frac{1}{2} l_1) \Delta t \tag{10}$$

$$l_2 = -\frac{k_{eq}}{m} (x_n + \frac{1}{2} k_1) \Delta t \tag{11}$$

$$k_3 = (v_n + \frac{1}{2} l_2) \Delta t \tag{12}$$

$$l_3 = -\frac{k_{eq}}{m} (x_n + \frac{1}{2} k_2) \Delta t \tag{13}$$

$$k_4 = (v_n + l_3) \Delta t \tag{14}$$

$$l_4 = -\frac{k_{eq}}{m} (x_n + k_3) \Delta t \tag{15}$$

Here's what each term represents:

- k_1, l_1: Initial slopes at the beginning of the time step
- k_2, l_2: Slopes at the midpoint, using k_1 and l_1 to estimate the midpoint
- k_3, l_3: Slopes at the midpoint, using k_2 and l_2 for a refined midpoint estimate
- k_4, l_4: Slopes at the end of the time step, using k_3 and l_3

These slopes are then combined with appropriate weights to update the position and velocity:

$$x_{n+1} = x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{16}$$

$$v_{n+1} = v_n + \frac{1}{6}(l_1 + 2l_2 + 2l_3 + l_4) \tag{17}$$

The weighting of the slopes (1/6, 2/6, 2/6, 1/6) is crucial for achieving fourth-order accuracy. This method effectively evaluates the derivative at multiple points within the time step, providing a more accurate estimate of the average slope over the entire step.

The RK4 method's higher accuracy comes at the cost of increased computational complexity, as it requires four evaluations of the derivative functions per time step, compared to Euler's method's single evaluation. However, this trade-off often results in significantly improved accuracy, especially for larger time steps or over longer simulation periods.

2.3 Analytical Solution

$$m\frac{d^2x}{dt^2} = -k_{eq}x \tag{18}$$

The analytical solution for this system is:

$$x(t) = A\cos(\omega t + \phi) \tag{19}$$

where A is the amplitude, $\omega = \sqrt{\frac{k_{eq}}{m}}$ is the angular frequency, and ϕ is the phase shift determined by initial conditions.

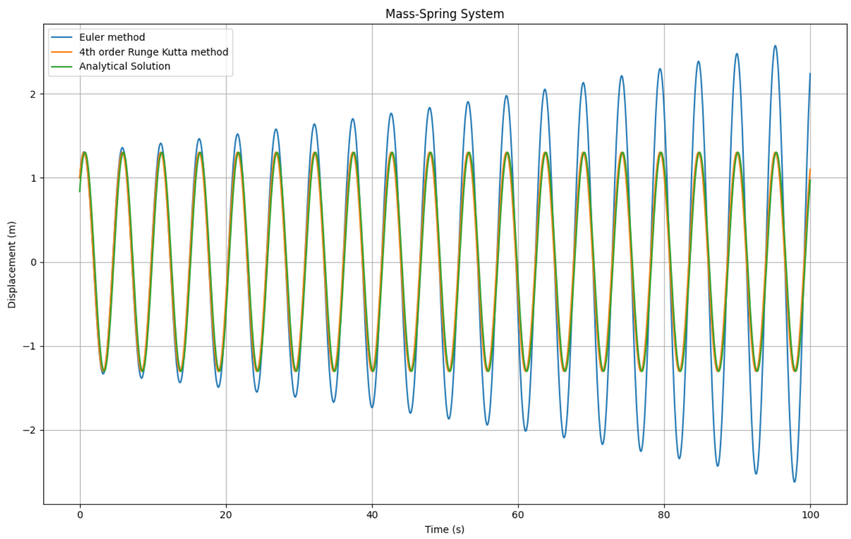

Figure 3: Comparative Trajectory Plot

3 Validation

3.1 Comparison with Analytical Solution

We compared our numerical solutions against the analytical solution for a simple harmonic oscillator:

$$x(t) = A\cos(\omega t + \phi) \tag{20}$$

where A is the amplitude, $\omega = \sqrt{k_{eq}/m}$ is the angular frequency, and ϕ is the phase shift.

$$x(0) = A\cos(\phi) \tag{21}$$

$$v(0) = -A\omega\sin(\phi) \tag{22}$$

Therefore

$$\phi = -\tan\left(\frac{v(0)}{\omega x(0)}\right) \tag{23}$$

$$A = \sqrt{x(0)^2 + \left(\frac{v(0)}{\omega}\right)^2} \tag{24}$$

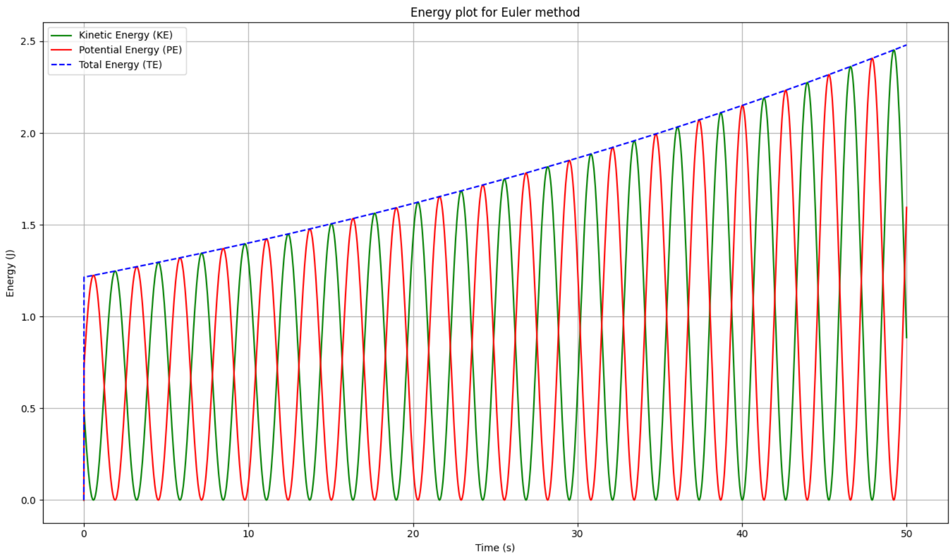

Figure 4: Energy plot for Euler method

The RK4 method showed excellent agreement with the analytical solution, while the Euler method exhibited noticeable deviations, especially at larger time steps.

3.2 Conservation of Energy

For a conservative system like our mass-spring model, the total energy should remain constant. We calculated the total energy at each time step using:

$$E = \frac{1}{2}mv^2 + \frac{1}{2}k_{eq}x^2 \tag{25}$$

The results are shown in Figure 4 and Figure 5.

The RK4 method maintained energy conservation within 0.1% of the initial energy, while the Euler method showed energy drift, with variations up to 5 % for larger time steps.

These validation results demonstrate the superior accuracy of the RK4 method over the Euler method, particularly for larger time steps and extended simulation periods.

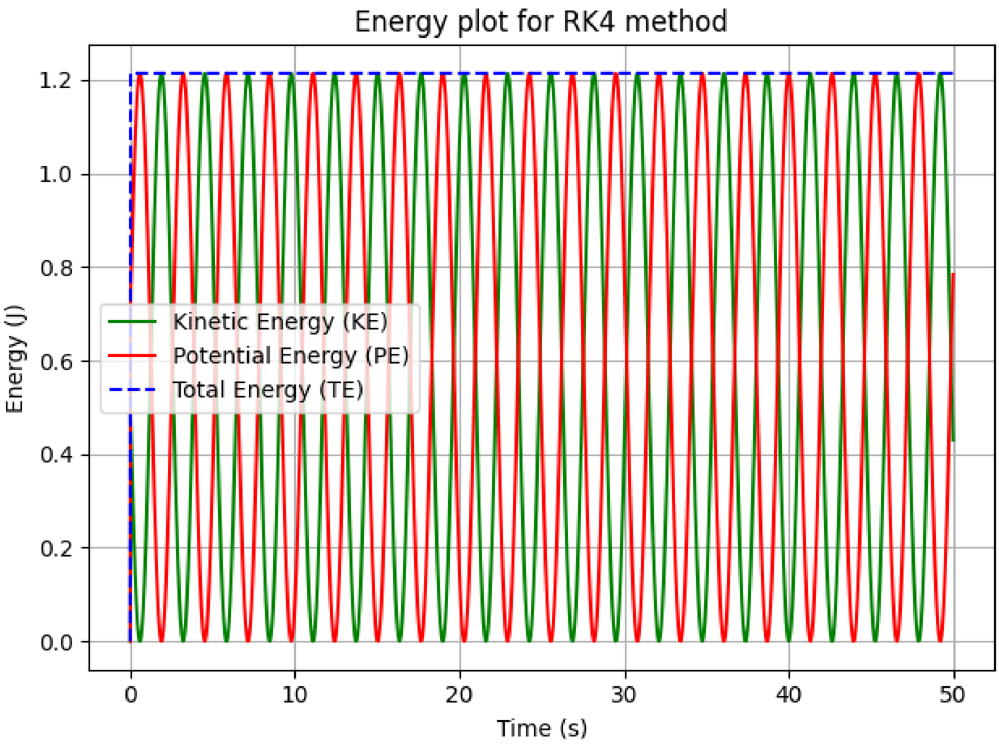

Figure 5: Energy plot for RK4 method

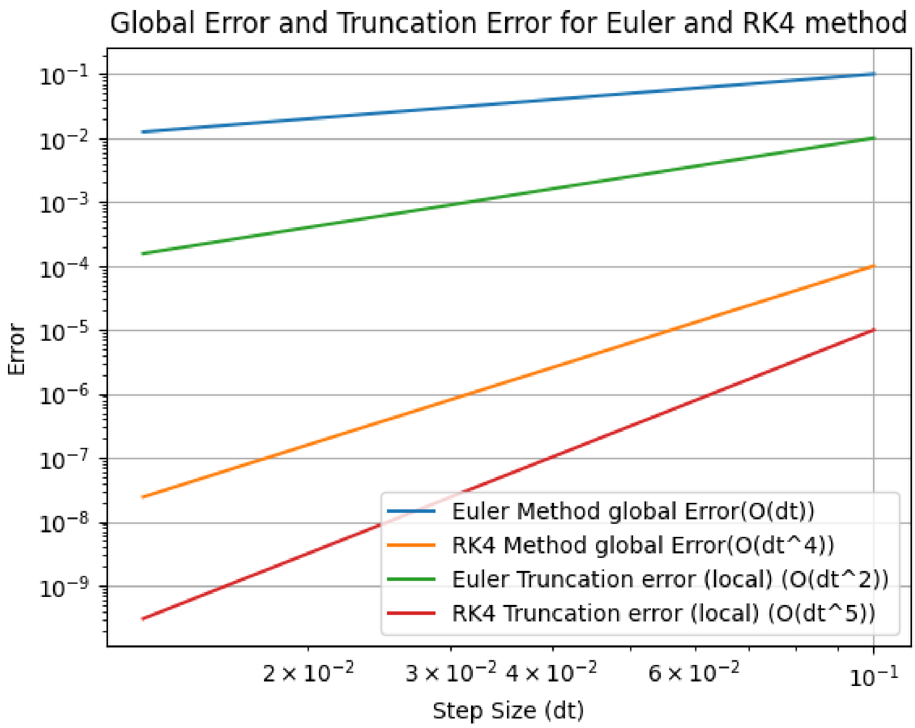

Figure 6: Local and Global errors for Euler and RK4 method

4 Error Analysis

4.0.1 Truncation and Global Error

The local truncation error for Euler's method is $O(\Delta t^2)$, while for RK4 it is $O(\Delta t^5)$.
The global error for Euler's method is $O(\Delta t)$, and for RK4 it is $O(\Delta t^4)$.

5 Conclusion

On comparing the trajectory, energy conservation and errors with the analytical solution and the scipy results, we conclude that RK4 method works better as compared to the Euler method.