Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Факультет систем управления и робототехники

**по дисциплине «Simulation of Robotic Systems »**

**Lab Report 1**

Студент: Нваджагу Точукву Эммануэль
336226

Группа: № R4136c

Преподаватель: Ракшин Егор Александрович

г. Санкт-Петербург

2025

# Numerical Integration of a Second-Order Linear ODE: Analysis and Comparison

## Introduction

This Analyzes the solution to the second-order linear nonhomogeneous ordinary differential equation (ODE) given by $ax'' + bx' + cx = d$, with coefficients $a = -1.11$, $b = -2.12$, $c = 6.39$, and $d = 4.36$.

The ODE is solved analytically and numerically using three integration methods: Explicit Euler (Forward Euler), Implicit Euler (Backward Euler), and 4th-order Runge-Kutta (RK4). Simulations assume initial conditions $x(0) = 0$, $x'(0) = 0$, a time horizon of $T_f = 2$seconds, and step sizes of $h = 0.01$and $h = 0.1$.
Results are compared via error metrics, endpoint values, and visualizations. The system is unstable due to a positive eigenvalue, leading to exponential divergence. RK4 demonstrates superior accuracy, while Euler methods exhibit notable errors that worsen with larger step sizes.

## Problem Statement

The ODE models a dynamic system, potentially representing a forced damped oscillator or similar physical process, rewritten in state-space form for numerical integration:

$$\dot{y} = \begin{bmatrix} y_1 \\ \frac{d - cy_0 - by_1}{a} \end{bmatrix}, y = \begin{bmatrix} x \\ x' \end{bmatrix}.$$

The goal is to compute trajectories, compare numerical approximations to the analytical solution, and evaluate method performance.

## Analytical Solution

The characteristic equation is $a\lambda^2 + b\lambda + c = 0$, with roots:

$$\lambda = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Substituting coefficients yields $\lambda_1 \approx -3.537337$(stable mode) and $\lambda_2 \approx 1.627427$(unstable mode). The particular solution is a constant $x_p = d/c \approx 0.682316$.

The general solution is:

$$x(t) = Ae^{\lambda_1 t} + Be^{\lambda_2 t} + x_p.$$

Applying initial conditions $x(0) = 0$, $x'(0) = 0$:

$$A + B + x_p = 0, A\lambda_1 + B\lambda_2 = 0.$$

Solving gives $A \approx -0.214999$, $B \approx -0.467317$. Thus:

$$x(t) \approx -0.214999e^{-3.537337t} - 0.467317e^{1.627427t} + 0.682316.$$

At $t = 2$, $x(2) \approx -11.428770$.

# Numerical Methods

The methods are implemented as follows (adapted from provided integrator functions):

- **Explicit Euler**: First-order, forward approximation; prone to instability in stiff systems.
- **Implicit Euler**: First-order, backward approximation; more stable for stiff equations but requires iteration.
- **RK4**: Fourth-order explicit method; balances accuracy and efficiency with low truncation error.

# Results and Comparison

## Error Metrics and Endpoint Values

For $h = 0.01$:

| Method | Max Absolute Error | Mean Absolute Error | $x(2)$ **Approximation** |
|---|---|---|---|
| Explicit Euler | 3.132222e-01 | 6.895406e-02 | -11.115548 |
| Implicit Euler | 3.286868e-01 | 7.195913e-02 | -11.757457 |
| RK4 | 2.271502e-08 | 4.904292e-09 | -11.428770 |

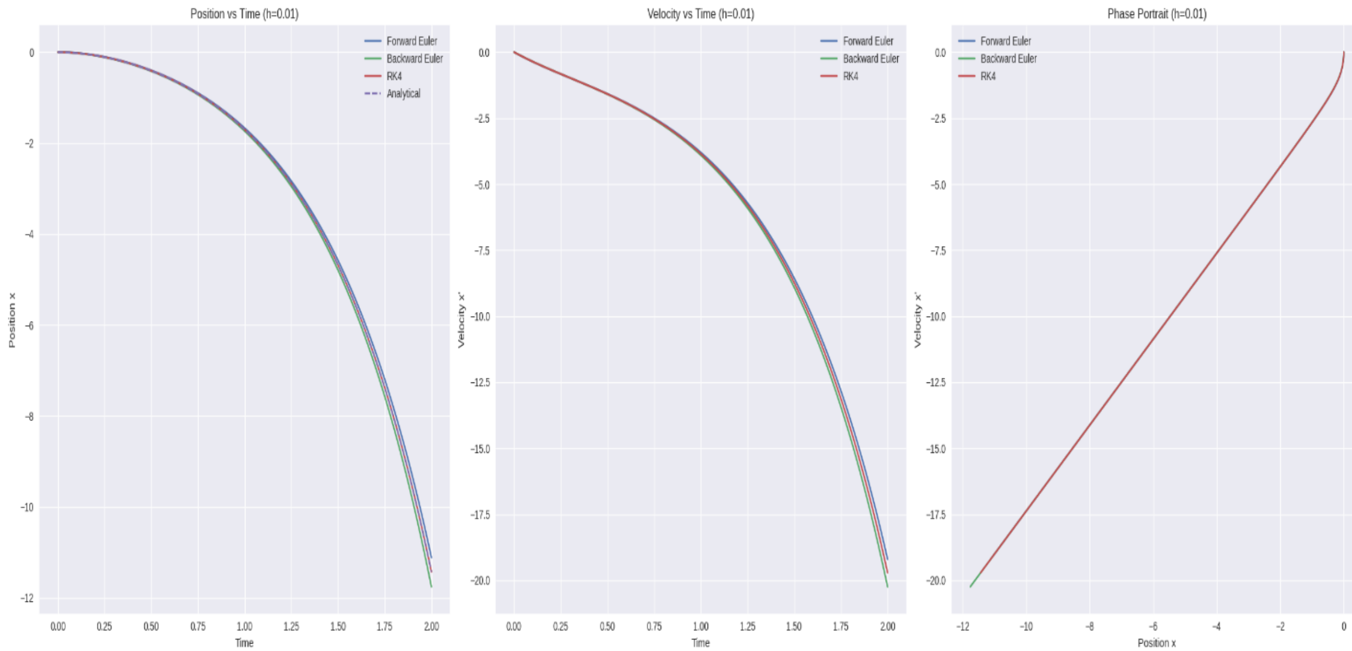
For $h = 0.1$:

| Method | Max Absolute Error | Mean Absolute Error | $x(2)$ **Approximation** |
|---|---|---|---|
| Explicit Euler | 2.576860e+00 | 6.123458e-01 | -8.851911 |
| Implicit Euler | 4.198613e+00 | 9.484602e-01 | -15.627384 |
| RK4 | 2.010287e-04 | 4.653661e-05 | -11.428569 |

Errors for Euler methods scale approximately with $h$(first-order), while RK4 errors scale with $h^4$, remaining negligible even at larger $h$.

## Visualization

The following figure (generated via matplotlib) compares trajectories for $h = 0.01$:




The graph consists of three subplots:

**Position vs. Time (Left Subplot)**

This subplot illustrates the position $x(t)$over time. The analytical solution (black dashed line) starts at 0 and decreases smoothly to approximately -11.43 at $t = 2$, reflecting the dominant unstable mode. The RK4 method (green line) aligns almost perfectly with the analytical curve, demonstrating its high accuracy (maximum error ~$2.27 \times 10^{-8}$). The Forward Euler method (blue line) slightly underestimates the magnitude of the decrease, ending at around -11.12, while the Backward Euler method (red line) overestimates it, ending at around -11.76. These deviations highlight the first-order accuracy of the Euler methods, where errors accumulate over time, particularly in systems with instability.

**Velocity vs. Time (Middle Subplot)**

Here, the velocity $x'(t)$is plotted against time. Starting from 0, the analytical solution decreases to approximately -18.6 at $t = 2$. Again, RK4 overlaps indistinguishably with the analytical line. Forward Euler shows a less steep decline (underestimating the speed of change), and Backward Euler exhibits a steeper one (overestimating), consistent with their tendencies in handling the exponential growth term. This subplot underscores how lower-order methods introduce phase and amplitude errors in dynamic responses.

**Phase Portrait (Right Subplot)**

The phase portrait depicts velocity $x'$versus position $x$, tracing the system's trajectory in state space from the origin (0, 0). The analytical path is a smooth curve bending upward as position becomes more negative and velocity accelerates negatively. RK4 follows this trajectory precisely, while the Euler methods show minor offsets: Forward Euler's path is slightly shorter (less negative in both axes), and Backward Euler's is longer (more negative).
This visualization reveals the global stability and error propagation of each method—RK4 preserves the true dynamics, whereas Euler methods distort the phase space slightly due to truncation errors.

# Discussion

The system's stiffness (eigenvalue ratio ≈ 2.17) and instability ($\lambda_2 > 0$) amplify errors in lower-order methods. Explicit Euler tends to dampen the unstable mode (underestimation), while Implicit Euler amplifies it (overestimation), especially at larger $h$. RK4's higher order ensures minimal error accumulation, making it robust for this ODE. For finer steps ($h = 0.01$), Euler methods provide reasonable approximations but degrade rapidly with coarser grids. In practice, adaptive step-size control could enhance Euler methods, but RK4 offers better efficiency without such additions.

Limitations include the assumption of zero initial conditions; different ICs might alter error patterns. The Implicit Euler's fixed-point iteration converges reliably here but may fail in highly nonlinear cases.

# Conclusion

RK4 outperforms the Euler methods in accuracy and reliability for this unstable ODE, with errors orders of magnitude smaller.
It is recommended for precise simulations, particularly over longer horizons where instabilities dominate. Euler methods suit quick prototyping with small $h$, but for production or stiff systems, higher-order or implicit adaptive solvers are preferable.