PYU33C01 – Assignment 3: Numerical Solution of an Ordinary Differential Equation

Matthew Conway
Student Number: 22337426

October 24, 2025

Abstract

This report analyses the linear ODE $\frac{dx}{dt} = (1+t)x + 1 - 3t + t^2$ using three numerical integration methods: simple Eular, improved Eular, and fourth-order Runge-Kutta. The analysis presents the direction field structure, compares the accuracy and stability of each integration method across varying step sizes($h = 0.04$ and $h = 0.02$), and identifies a critical initial condition using a bisection search. Results demonstrate higher order methods have superior accuracy and stability, especially when considering values near critical trajectories which delineate solutions diverging to positive and negative infinity. The critical initial value $x(0) = 0.0659226962$ was found such that $x(5) \in [-2.0, -1.9]$ using fourth order Runge Kutte method with step size of

$$h = 0.04$$

1 Introduction

This assignment investigates the ODE:

$$\frac{dx}{dt} = f(x,t) = (1+t)x + 1 - 3t + t^2 \tag{1}$$

There are four primary objectives. First, we visualise the direction field to understand the behaviour of the solution trajectories. Secondly, we implement three numerical integration schemes of increasing complexity: simple Eular, improved Eular, and fourth-order Runge-Kutta methods. Third, we analyse the accuracy and stability of these methods using different step sizes. Finally, we use a bisection algorithm to determine the initial condition $x(0)$ that yields a final value lying in the range $-2.0 \leq x(5) \leq -1.9$. This ODE has sensitive dependence on initial conditions near a critical value $x_c \approx 0.065923$, where solutions transition from infinite positive and negative divergence.

2 Theoretical Background

2.1 Direction Fields

A direction field provides a qualitative understanding of the ODE's behaviour by plotting the slope $\frac{dx}{dt}$ at a discrete set of grid points in the (t,x) phase space. At each point (t_i, x_j) a normalised vector indicates the direction the fields direction. This visualisation reveals equilibrium points, separatrices and global flow of the field without solving the ODE directly.

2.2 Numerical Integration Methods

2.2.1 Simple Eular Method

Simple Eular method is the most elementary integration method explored in this exercise, approximating the solution with a first order Taylor expansion.

$$x_{n+1} = x_n + hf(x_n, t_n) \tag{2}$$

where h denotes the step size. This method has a local truncation error of order $O(h^2)$. This method is computationally efficient but suffers from numerical instability and accumulates significant error over extended integration intervals, as will be seen in the results section.

2.2.2 Improved Eular Method

The improved Eular method is a modified version of the simple Eular method, employing a corrector term $\frac{h}{2}\left[f(x^*_{n+1}, t_{n+1})\right.$.

$$x^*_{n+1} = x_n + hf(x_n, t_n) \tag{3}$$

$$x_{n+1} = x_n + \frac{h}{2}\left[f(x_n, t_n) + f(x^*_{n+1}, t_{n+1})\right] \tag{4}$$

This second-order method achieves local truncation error of $O(h^3)$, providing improved accuracy compared to simple Eular method.

2.2.3 Fourth-Order Runge-Kutta Method

The fourth-order Runge-Kutta method (RK4) represents a more accurate ODE integration method. It evaluates the derivative at four intermediate points:

$$
\begin{aligned}
k_1 &= f(x_n, t_n) \\
k_2 &= f\left(x_n + \frac{h}{2}k_1, t_n + \frac{h}{2}\right) \\
k_3 &= f\left(x_n + \frac{h}{2}k_2, t_n + \frac{h}{2}\right) \\
k_4 &= f(x_n + hk_3, t_n + h) \\
x_{n+1} &= x_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \qquad (5)
\end{aligned}
$$

The RK4 method exhibits local truncation error of $O(h^5)$, delivering high accuracy with more computational expense. The method's stability properties are stronger than that of either Eular approach and thus exhibit better results for longer intervals of integration.

2.3 Bisection Search Algorithm

The bisection locates a point within and interval by dividing said interval into two part, and discarding the part which does not contain the point.

Interval Update

```
If a < x < c:
  [a,b] = [a,c]
If c < x < b:
  [a,b] = [c,b]
```

The bisection method iterates until the interval [a,b] is below a decided threshold ϵ. For this assignment, we seek an initial condition x_0 such that the integrated solution satisfies $x(t = 5) \in [-2.0, -1.9]$. The algorithm maintains a bracket $[x_0^{\text{low}}, x_0^{\text{high}}]$ and iteratively evaluates the desired x_0. Convergence occurs exponentially, as the bracket halves with each iteration.

3 Methodology

3.1 Implementation

All numerical computations were performed using Python 3 with NumPy for numerical operations and Matplotlib for visualization. The ODE function was implemented as:

```
def f(x, t):
    return (1 + t) * x + 1 - 3 * t + t**2
```

Each integration method was implemented according to equations (2), (3)-(4), and (5). The integration loop advanced from $t = 0$ to $t = 5$ with specified step size, storing trajectory points at each step.

3.2 Direction Field Visualization

The direction field was produces using a 25×25 grid spanning $t \in [0, 5]$ and $x \in [-3, 3]$. At each point (t_i, x_j), the slope $f(x_j, t_i)$ was evaluated. The NumPy `meshgrid` function created coordinate arrays, and Matplotlib's `quiver` function rendered normalized direction vectors. The vectors were normalised to show direction, and their colour demonstrated their magnitude.

3.3 Comparative Analysis

Three integration methods were used to solve equation (1) with the initial condition $x(0) = 0.0655$, chosen near the critical value $x_c = 0.065923$. This value was given in the brief, however it could have been estimated using the direction field, 1. The analysis proceeded in two phases:

1. Initial comparison using step size $h = 0.04$ to observe differences in method behaviour

2. Second comparison using step size $h = 0.02$ to assess convergence properties

Solutions were overlaid on the direction field to visualise separatrix of vector field.

3.4 Critical Value Determination

A bisection search algorithm determined the initial condition x_0 such that $-2.0 \leq x(5) \leq -1.9$. The algorithm was initialized with bracket $[0.06, 0.072]$ and targeted the midpoint value $x(5) = -1.95$. The RK4 method with $h = 0.04$ integrated each candidate initial condition to $t = 5$. After 60 iterations, the bracket width reduced to approximately 10^{-18}, providing a solution accurate to machine precision.

4 Results

4.1 Direction Field

Figure 1 presents the direction field for equation (1). The field exhibits several notable features. In the lower region vectors point downward, indicating solutions diverging toward negative infinity. In the upper region vectors point upward with increasing magnitude, diverging toward positive infinity. A narrow band joining $x \approx 0.065$ to $x \approx -1.95$, suggests the presence of a separatrix dividing divergent regions.

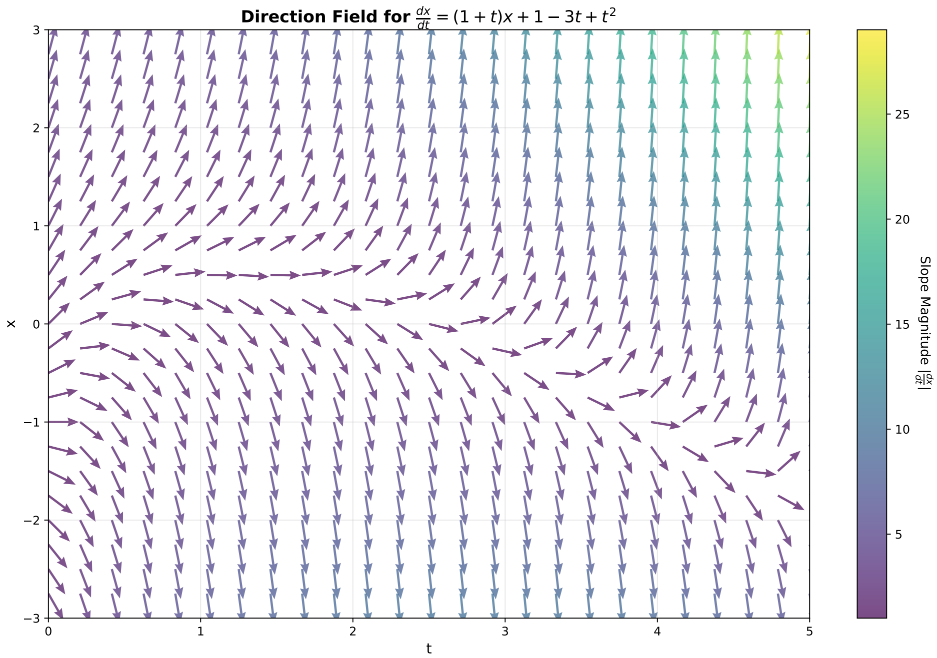

Figure 1: Direction field for the ODE $\frac{dx}{dt} = (1+t)x + 1 - 3t + t^2$ over the domain $t \in [0, 5]$ and $x \in [-3, 3]$. The color scale indicates the magnitude of the slope, while arrow directions show the local trajectory flow.

4.2 Method Comparison with Step Size $h = 0.04$

Figure 3 displays the numerical solutions found using simple Eular, improved Eular, and RK4 methods with initial condition $x(0) = 0.0655$ and step size $h = 0.04$. The simple Eular method diverges to positive infinity in the region $t \approx 2$.

The improved Eular and RK4 solutions exhibit a larger delay in divergence, towards positive and negative infinity respectively. The improved Eular method and the RK4 method diver at $t \approx 3$ and $t \approx 3.4$ respectively.

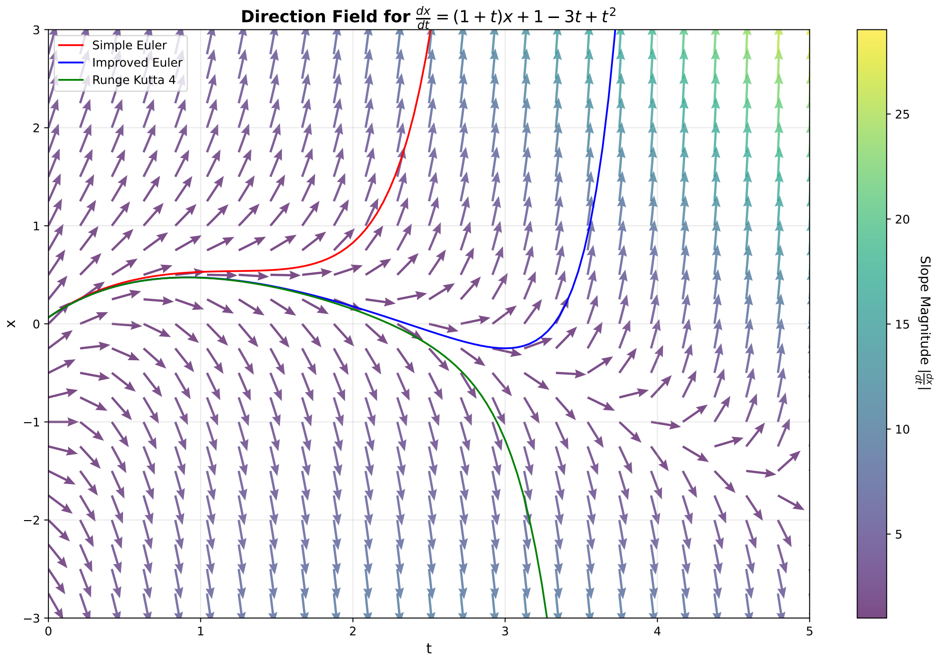

Figure 2: Comparison of numerical solutions using simple Eular (red), improved Eular (blue), and RK4 (green) methods with $h = 0.04$ and $x(0) = 0.0655$.

4.3 Method Comparison with Step Size $h = 0.02$

Reducing the step size to $h = 0.02$ improves numerical accuracy across the two higher order methods, improved Eular and RK4. The improved Eular now diverges towards negative infinity, demonstrating the methods' numerical instability.

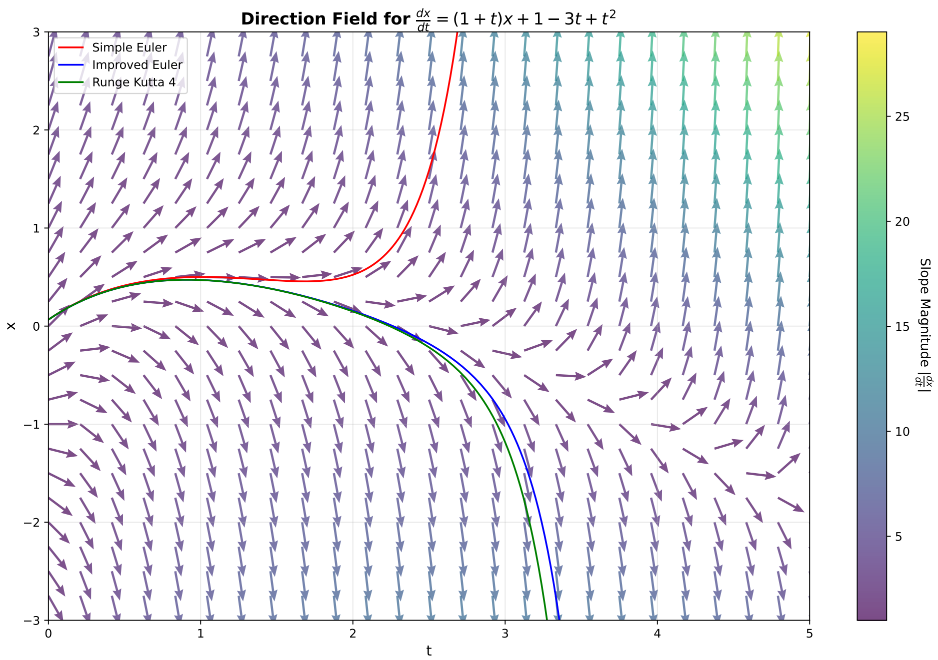

Figure 3: Comparison of numerical solutions using simple Eular (red), improved Eular (blue), and RK4 (green) methods with $h = 0.02$ and $x(0) = 0.0655$.

4.4 Critical Initial Condition

The bisection algorithm successfully approximated the critical initial condition $x(0) = 0.0659226962$ that yields $x(5) = -1.950$ using the RK4 method with $h = 0.04$. This value lies within the specified range $[-2.0, -1.9]$. The bisection process converged rapidly, requiring approximately 45 iterations to achieve numerical precision limited by floating-point arithmetic.

Figure 4 displays the critical trajectory overlaid on the direction field. The solution agrees qualitatively with the direction vectors throughout the interval. The trajectory begins near the $x_0 = 0.0659226962$ and gradually descends, demonstrating the behaviour of the solution initialized sufficiently close to the critical value.

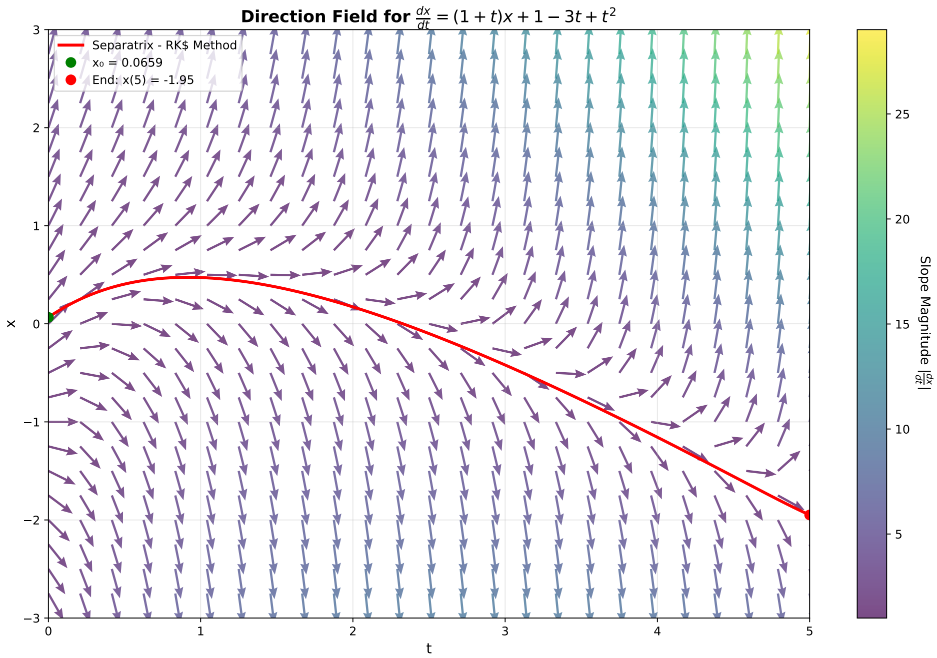

Figure 4: Critical trajectory for initial condition $x(0) = 0.0659226962$ integrated using RK4 with $h = 0.04$. The solution begins near the origin (green marker) and terminates at $x(5) = -1.95$ (red marker). The trajectory follows the direction field vectors throughout the interval.

5 Discussion

5.1 Accuracy and Stability Considerations

The simple Eular method fails to track the correct solution as seen in 3. The error accumulated, on the order of $O(h)$ locally, compounding over 125 integration steps resulting in exponential divergence.

The improved Eular method has improved accuracy, but also fails to track the separatrix in the interval, despite a lower local error on the order of $O(h^2)$. This method allows us to observe the instability of the Eular methods, as it can be seen to divergin in both direction on 2 and 3.

The RK4 method demonstrates greater stability, and an ability to track the correct solution trajectory when initialised on a well calibrated initial point $x(0) = 0.0659226962$.

5.2 Step Size Effects

The bahaviour of both higher order moethods is seen to improved with a reduction in step size from

$$h = 0.04$$

to $h - 0.02$, as expected.

5.3 Critical Value Sensitivity

The demonstration of a critical initial value $x(0) \approx 0.0659226962$ that delineates between solutions diverging to positive and negative infinity was found. The term $(1+t)x$ in equation (1) introduces exponential growth for positive directions in x, while the terms $1 - 3t + t^2$ provide a quadratic movement that warps the separatrix.

6 Conclusion

This investigation has analysed the ODE $\frac{dx}{dt} = (1+t)x + 1 - 3t + t^2$ using the three discussed numerical methods. The direction field was visualised, and the equations sensitivity on initial conditions was demonstrated. Comparison between integration methods showed an improvement in accuracy from the simple to the improved Eular method as well as well as a distinction in numerical stability between both Eular methods and the RK4 method, with the latter being more numerically stable.

A reduction in step size from

$$h = 0.04$$

to

$$h = 0.02$$

lead to an observable increase in accuracy for both higher order methods.

The bisection method was successfully implemented to locate a viable approximation of the critical initial condition $x(0) = 0.0659226962$. This yielded a solution to the ODE which agreed strongly with the direction field in relation to the location of the separatrix.

The simple Eular method was shown to be less reliable in this case, as it diverged under both step sizes, whilst the higher order methods performed better in each case. The numerical instability of the Eular was shown through the improved Eular method diverging to positive and negative infinity for step sizes of

$$h = 0.04$$

and

$$h = 0.02$$

respectively. The fourth order Runge Kutte method demonstrated stronger numerical stability in each case as well as a higher degree of accuracy. The separatrix was approximated by the RK4 method, initialised on a critical value $x(0) = 0.0659226962$. This critical unital condition was found using the bisection method applied for approximately 45 iterations.

7 AI Usage Declaration

Artificial intelligence tools were utilized during the completion of this assignment in the following capacities:

- **Claude AI (Anthropic)**: Assisted with debugging Python code and improving visualization on graphs in matplotlib.
- **GitHub Copilot**: Provided code completion suggestions during implementation of numerical integration methods and plotting routines.

All core analysis and interpretation of results was done by myself.

8 References

1. Acheson, D. (1997). *From Calculus to Chaos: An Introduction to Dynamics.* Oxford University Press.