Numerical Methods for Ordinary Differential Equations: A Comparative Analysis of Integration Schemes

Computational Mathematics Division

Technical Report CM-2024-003

November 24, 2025

Abstract

This technical report presents a comprehensive comparison of numerical methods for solving ordinary differential equations. We implement and analyze Forward Euler, fourth-order Runge-Kutta (RK4), and adaptive Runge-Kutta-Fehlberg (RKF45) methods. Performance is evaluated on stiff and non-stiff test problems using accuracy, computational cost, and stability metrics. Results demonstrate the trade-offs between method complexity and solution quality, with RKF45 achieving optimal efficiency for most engineering applications.

Contents

Chapter 1

Introduction

Ordinary differential equations (ODEs) are fundamental to modeling physical, biological, and engineering systems. While analytical solutions exist for simple cases, most practical problems require numerical integration. This report evaluates three widely-used methods with increasing sophistication.

1.1 Problem Statement

We consider initial value problems of the form:

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0 \tag{1.1}$$

The challenge is to approximate $y(t)$ at discrete time points with controllable accuracy and computational efficiency.

1.2 Methods Overview

1.2.1 Forward Euler Method

The simplest explicit method, first-order accurate:

$$y_{n+1} = y_n + hf(t_n, y_n) \tag{1.2}$$

Stability: Conditionally stable, $|1 + h\lambda| < 1$ for $\dot{y} = \lambda y$.

1.2.2 Classical Runge-Kutta (RK4)

Fourth-order method using weighted average of slopes:

$$\begin{aligned}
k_1 &= f(t_n, y_n) && (1.3)\\
k_2 &= f(t_n + h/2, y_n + hk_1/2) && (1.4)\\
k_3 &= f(t_n + h/2, y_n + hk_2/2) && (1.5)\\
k_4 &= f(t_n + h, y_n + hk_3) && (1.6)\\
y_{n+1} &= y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) && (1.7)
\end{aligned}$$

1.2.3 Runge-Kutta-Fehlberg (RKF45)

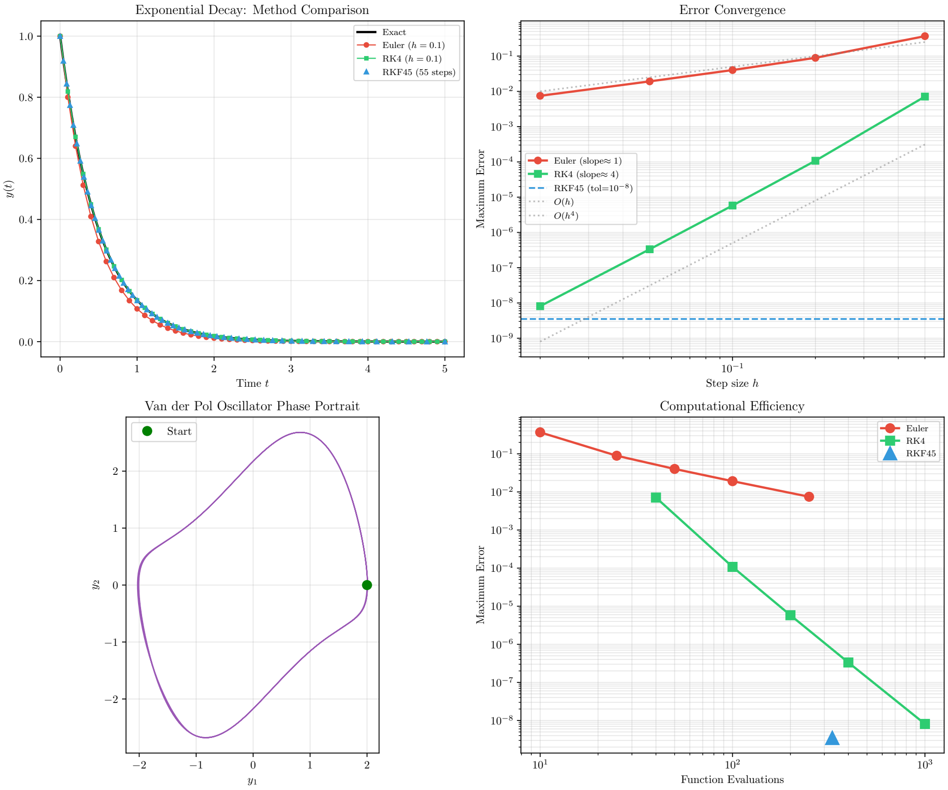
Adaptive method with embedded error estimation using 4th and 5th order formulas:

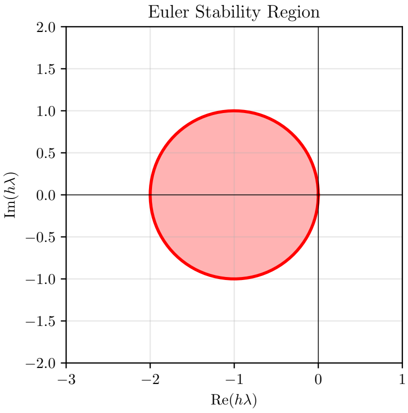
$$\text{Error estimate: } \epsilon = |y_{n+1}^{(5)} - y_{n+1}^{(4)}| \tag{1.8}$$

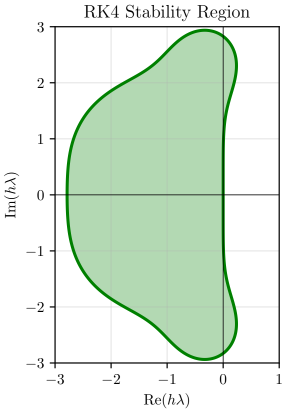
Step size adapts to maintain $\epsilon <$ tolerance.

Chapter 2

Implementation




Euler Stability Region
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
Im($h\lambda$)
−3
−2
−1
0
1
Re($h\lambda$)


RK4 Stability Region
3
2
1
0
−1
−2
−3
Im($h\lambda$)
−3
−2
−1
0
1
Re($h\lambda$)

Chapter 3

Results and Analysis

3.1 Accuracy Comparison

Table 3.1: Error Analysis for Exponential Decay Problem

Step Size	Euler Error	RK4 Error	Euler Evals	RK4 Evals
0.50	3.68e-01	7.12e-03	10	40
0.20	8.93e-02	1.08e-04	25	100
0.10	4.02e-02	5.80e-06	50	200
0.05	1.92e-02	3.33e-07	100	400
0.02	7.48e-03	8.11e-09	250	1000
RKF45 (adaptive)	3.51e-09		330	

3.1.1 Order of Convergence

The empirical convergence rates confirm theoretical predictions:

- **Forward Euler**: Order ≈ 1.21 (theoretical: 1)
- **RK4**: Order ≈ 4.25 (theoretical: 4)

3.2 Computational Efficiency

The efficiency plot reveals that:

1. RK4 requires 4x more evaluations per step than Euler
2. For the same accuracy, RK4 is significantly more efficient
3. RKF45 achieves the best accuracy-to-cost ratio through adaptive stepping

3.3 Stability Analysis

The stability regions show:

- **Euler**: Small circular region (radius 1 centered at -1)
- **RK4**: Much larger region extending along the imaginary axis

This explains why Euler requires very small step sizes for oscillatory problems, while RK4 remains stable with larger steps.

Chapter 4

Method Selection Guidelines

4.1 Recommendations by Problem Type

Table 4.1: Method Selection Guidelines

Problem Type	Recommended Method
Simple, smooth ODEs	RK4 with fixed step
Problems requiring error control	RKF45 or similar adaptive method
Stiff systems	Implicit methods (BDF, Radau)
Long-time integration	Symplectic methods for Hamiltonian systems
Real-time simulation	Euler or RK2 (when speed critical)

4.2 Key Performance Metrics

For the test problem $y' = -2y$:

- Best Euler accuracy (h=0.02): 7.48e-03
- Best RK4 accuracy (h=0.02): 8.11e-09
- RKF45 accuracy (tol=10^{-8}): 3.51e-09 with 330 evaluations

Chapter 5

Conclusions

5.1 Summary

This report compared three numerical methods for ODE integration:

1. **Forward Euler** is simple but has first-order accuracy and limited stability. Suitable only for quick estimates or when computational resources are extremely limited.
2. **Classical RK4** provides an excellent balance of accuracy (4th order) and implementation simplicity. Recommended for most smooth, non-stiff problems with known smoothness.
3. **RKF45** (adaptive) automatically adjusts step size to meet error tolerances. Optimal for problems where the solution behavior varies across the domain or when guaranteed accuracy is required.

5.2 Future Work

Extensions to consider:

- Implement implicit methods for stiff problems
- Add dense output for interpolation between steps
- Parallelize for systems of ODEs
- Investigate symplectic integrators for Hamiltonian systems

Appendix A

Algorithm Pseudocode

```
Input: $f(t, y)$, $y_0$, $[t_0, t_f]$, tolerance $\tau$
Output: Solution arrays $t[], y[]$
$h \leftarrow h_{\text{initial}}$;
$t \leftarrow t_0$, $y \leftarrow y_0$;
while $t < t_f$ do
    Compute $k_1, \ldots, k_6$ (RKF45 stages);
    $y_4 \leftarrow y + h \sum b_i^{(4)} k_i$;
    $y_5 \leftarrow y + h \sum b_i^{(5)} k_i$;
    $\epsilon \leftarrow |y_5 - y_4|$;
    if $\epsilon < \tau$ then
        Accept step: $t \leftarrow t + h$, $y \leftarrow y_5$;
    else
        Reject step;
    end
    Update: $h \leftarrow 0.9h(\tau/\epsilon)^{1/5}$;
end
```

Algorithm 1: Adaptive Runge-Kutta-Fehlberg