Two-Body Orbital Dynamics: A Numerical Investigation

Comparing RK4 and Adaptive RK45 Methods for Gravitational Systems

---

Student: Aditya Agrawal

Roll No: 2311010

Course: Computational Physics

Supervisor: Dr. Subhashish Basak

Date: 17 November 2025

Abstract

This project investigates the numerical solution of the gravitational two-body problem using two different integration schemes: the classical fourth-order Runge-Kutta (RK4) method and the adaptive Runge-Kutta-Fehlberg (RK45) method. Through systematic comparison of energy conservation, computational efficiency, and long-term stability, I found that while RK4 provides slightly better energy conservation (7.47×10■■ vs 9.28×10■■ relative drift), the adaptive RK45 method achieves comparable accuracy using 7.2 times fewer integration steps. This investigation reveals the trade-offs between computational efficiency and numerical precision in orbital mechanics simulations.

---

1. Problem Description and Motivation

1.1 The Challenge

The mathematical challenge lies in solving the coupled system of second-order differential equations:

$$m_i rac{d^2\mathbf{r}_i}{dt^2} = \sum_{j

eq i} rac{Gm_im_j(\mathbf{r}_j - \mathbf{r}_i)}{|\mathbf{r}_j - \mathbf{r}_i|^3}$$

While analytical solutions exist for this problem, but I wanted to understand how numerical methods perform and whether modern adaptive techniques offer advantages over classical fixed-step

approaches.

1.2 But....Why This Matters!?

Understanding numerical integration accuracy is crucial because:

- Small errors accumulate over long time periods(say 1000-2000 oscillations!)

- Energy conservation serves as a reliability indicator

- Different methods have varying computational costs

---

2. Numerical Methodology and Implementation

2.1 Initial Approach and Learning Curve

I started by implementing a basic Euler method, but quickly realized it was too inaccurate for orbital simulations - the orbit would spiral outward due to energy accumulation! This led me to research higher-order methodslike RK4

2.2 Fourth-Order Runge-Kutta (RK4) Implementation

Then I applied Rk4 method(which was already discussed in class)....which goes as:

```
def rk4_step(self, state, dt):
    k1 = self.derivatives(state)
    k2 = self.derivatives(state + 0.5*dt*k1)
    k3 = self.derivatives(state + 0.5*dt*k2)
    k4 = self.derivatives(state + dt*k3)
    return state + (dt/6.0)*(k1 + 2*k2 + 2*k3 + k4)
```

But I encountered Some Challenges...like:

- Choosing the right time step as too large caused instability, too small was inefficient :( )

- Handling the gravitational singularity when bodies get too close

2.3 Adaptive RK45 Implementation

Reading about modern numerical methods, I learned about adaptive step-size control. The RK45 method computes both 4th and 5th order solutions, and then uses difference to estimate local error and adjust the step size automatically.

Instead of guessing the optimal step size, the algorithm adapts based on the local dynamics - using smaller steps during rapid changes (like periapsis passage) and larger steps during smooth motion.

```
def rk45_step(self, state, dt, tolerance=1e-6):
    # Compute 4th and 5th order solutions using Cash-Karp coefficients
    # ... [implementation details]

    error = np.linalg.norm(y5 - y4)
```

```
dt_optimal = safety * dt * (tolerance / error)**(1/5)

if error <= tolerance:
    return y5, dt, dt_optimal, error  # Accept step
else:
    return state, 0.0, dt_optimal, error  # Reject and retry
```

2.4 Test Case Design

Here I chose an elliptical orbit (eccentricity = 0.3) because(obiviously) it is more challenging than circular orbits and represents realistic orbital scenarios

System parameters:

- Primary mass: m■ = 5.0 (dimensionless units)
- Secondary mass: m■ = 1.0
- Initial separation: 1.0
- Integration time: 15.0 time units (~2.4 orbital periods)

2.5 Energy Conservation Analysis

$$ ext{Relative Error} = rac{|E(t) - E(0)|}{|E(0)|}$$

NOTE:---> This provides a sensitive measure of accumulated numerical errors over time.

---

3. Results and Discussion

3.1 Numerical Stability Comparison

Method	Integration Steps	Execution Time	Energy Drift	Acceptance Rate
RK4	3,000	0.022s	7.47×10■■	N/A
RK45	418	0.013s	9.28×10■■	83.6%

Observations:

- RK4 achieves about 12× better energy conservation
- RK45 takes 7.2× fewer steps
- Both methods maintain excellent conservation (<10■■)
- RK45's step acceptance rate of 83.6% indicates good error control

3.2 Unexpected Findings

What surprised me: I initially expected the adaptive method to be more accurate, but RK4 actually preserved energy better. However, this came at the cost of using many more integration steps. THis may happen because RK4's smaller, uniform time steps provide more consistent accuracy, while RK45 trades some precision for computational efficiency through larger adaptive steps.

Plus.....in a long run both behaved more or less same

---

5. Conclusions and Future Work

5.1 Key Findings and Personal Learning Outcomes

Through this investigation, I discovered that:

- Both RK4 and RK45 methods are suitable for orbital mechanics
- Fixed-step RK4 provides higher precision at computational cost
- Adaptive RK45 offers excellent efficiency with acceptable accuracy
- Energy conservation serves as a reliable method validation tool

Through this project , I learned that :

- Numerical analysis and error propagation
- Adaptive algorithms and computational intelligence
- Conservation laws as validation tools
- The importance of method selection in computational physics

The most valuable insight was learning : "when and why to use different approaches"!

---

6. References

Goldstein, H., Poole, C., & Safko, J. (2002). *Classical Mechanics*. 3rd ed. Addison Wesley.

SciPy Documentation - Runge-Kutta Methods.
https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html

Computational Physics Course Notes - Prof. Dr. Subhashish. Basak,NISER (2025)