Numerical Solution of Ordinary Differential Equations Using Various Methods

---

1. Introduction

Numerical methods are essential for solving ordinary differential equations (ODEs) that lack analytical solutions. Many real-world problems in physics, engineering, and finance require numerical techniques for approximation. In this report, we implement and compare various numerical methods to solve the equation:

$$\frac{dy}{dt} = -0.5y + sin\,(t), y(0) = 1$$

The methods used in this study include:

1. **Forward Euler Method**
2. **Modified Euler Method (Heun's Method)**
3. **Backward Euler Method**
4. **Runge-Kutta 2nd Order (RK2 - Midpoint Method)**
5. **Runge-Kutta 3rd Order (RK3)**
6. **Runge-Kutta 4th Order (RK4)**
7. **Adams-Bashforth 2-Step Method (Explicit)**
8. **Adams-Moulton 2-Step Method (Implicit)**

The objective is to evaluate the accuracy, stability, and computational cost of these methods.

---

2. Implementation in Octave

2.1 Initial Parameters

- $t_0 = 0, t_f = 10$
- $y_0 = 1$
- $Step\ size\ h = 0.1$

2.2 Numerical Methods Implementation

Each method is implemented using iterative schemes. Explicit methods, such as Forward Euler and Runge-Kutta, compute the next value directly, while implicit methods, such as Backward Euler and Adams-Moulton, require solving algebraic equations at each step.

---

3. Results and Discussion

3.1 Forward Euler Method

- The simplest method, using only past values to estimate the next step.
- Easy to implement but accumulates numerical errors quickly.
- Unstable for large step sizes, leading to divergence.

3.2 Modified Euler Method (Heun's Method)

- Uses a predictor-corrector approach, making it more accurate than Forward Euler.

- Reduces error accumulation but does not completely eliminate instability for larger step sizes.
- More computationally expensive than Forward Euler but still relatively efficient.

3.3 Backward Euler Method

- An implicit method that computes the next step using the function evaluated at the future step.
- Always stable, even for large step sizes, making it suitable for stiff equations.
- Requires solving an equation at each step, increasing computational cost.

3.4 Runge-Kutta Methods

- **RK2 (Midpoint Method)**: Uses an intermediate step for better accuracy than Euler.
- **RK3**: A third-order method balancing efficiency and accuracy.
- **RK4**: The most widely used due to its high accuracy and stability.
- Runge-Kutta methods are computationally efficient while maintaining stability, especially RK4.

3.5 Multi-Step Methods

- **Adams-Bashforth (Explicit)**: Uses past values to predict the next step, reducing function evaluations.
- **Adams-Moulton (Implicit)**: Includes a correction step, improving stability over explicit methods.

- Multi-step methods are useful for long-term integration but require initialization using single-step methods.

---

4. Comparison of Methods

Method	Accuracy	Stability	Computational Cost
Forward Euler	Low	Unstable for large hh	Low
Modified Euler	Medium	More stable	Medium
Backward Euler	Medium	Very stable	High
RK2	Higher than Euler	Stable	Medium
RK3	High	Stable	Medium-High
RK4	Very High	Very Stable	High
Adams-Bashforth	High	Conditionally stable	Medium
Adams-Moulton	Very High	Very stable	High

---

5. Conclusion

Among all methods:

- **RK4 and Adams-Moulton provide the best accuracy and stability, making them ideal for most applications.**
- **Forward Euler is simple and computationally cheap but accumulates errors rapidly.**
- **Backward Euler and Adams-Moulton are preferable for stiff equations, as they remain stable for large step sizes.**
- **Multi-step methods like Adams-Bashforth reduce computational overhead by using previous values but require special handling for initialization.**