Team Project Outline in LaTeX Template

Ratislav Krylov Caleb Lewis

Abstract

Many useful differential equations either cannot be solved analytically or do not have analytical solutions. In this paper we explore the power of different numerical analysis methods by implementing them in Python. Methods such as Runge-Kutta Four and Euler have different computational cost, so in order to keep comparison fair the step size is adjusted so that both methods have similar cost. Then the errors of both methods are plotted and compared to see which one performs better on example problems. One of the surprising findings is that Runge-Kutta Two outperformed Modified Euler method, even though both of them have similar computational cost and order of local truncation error. We also descovered that Predictor-Corrector method outperforms RK4 in all examples as well, and the latter method is more computationally costly.

1 Introduction

This is an exploratory paper meant to be our proving and playing ground where we explore methods learned in Numerical Analysis II class. In the class, we have closely covered Euler, Modefied Euler, Midpoint Methid, Runge-Kutta Four, Adams-Bashforth Four-step explicit method, and Adams 4th-order Predictor corrector method, as well as some other methods that are not covered in this paper. While we have done some review of the proofs for Euler method, this paper mainly focuses on numerical exploration of these methods. We implement different techniques using python and compare them on the set of 3 textbook problems. Visual plots of errors are made to make it easier to see how they each compare to each other. To measure up methods fairly things such as computational costs and memory costs are taken into account, though computational cost is weighted more.

2 Methods in this study

2.1 Euler's Method

Euler's method is one of the oldest numerical methods with intuitive rules. Given IVP:

$$y\prime = f(t,y) fort \in [a,b] andy(a) = \alpha$$

We can divide the interval into N equal segments. Then Euler's method on this IVP is:

$$w_0 = \alpha$$

$$w_{i+1} = w_i + hf(t_i, w_i)$$

Where $i = 0, 1, ..., N-1$.

Euler's method local truncation error is:

$$|\tau_{i+h}(h) = \frac{hM}{2} = O(h).$$

Making it change linearly with step-size with the computational cost of one evaluation of $f(t, y)$ per step.

2.2 Modified Euler's Method

Modified Euler's method is a take on the original idea of eulers method and use of two function evaluations:

$$w_0 = \alpha$$

$$w_{i+1} = w_i + \frac{h}{2}(f(t_i, w_i) + f(t_{i+1}, w_i + hf(t_i, w_i)))$$

It has $O(h^2)$ local truncation error.

2.3 Midpoint Method(RK2)

Runge-Kutta methods are a family of methods that see a big boost in comparison to Euler's method. One of them is RK2 with the following scheme:

$$w_0 = \alpha$$

$$w_{i+1} = w_i + hf(t_i + \frac{h}{2}, w_i + \frac{h}{2}f(t_i, w_i))$$

It evaluates the function twice and has a local truncation error of order $O(h^2)$.

2.4 Runge-Kutta method

2.5 Adams-Bashforth Four-step explicit method

2.6 Predictor-Corrector method

3 Numerical experiments

4 Discussion

5 Summary