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. 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 discovered that Predictor-Corrector method outperforms RK4 in all examples as well.

1 Introduction

This is an exploratory report 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, Modified Euler, Midpoint Method, Runge-Kutta 4, Adams-Bashforth 4-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, we take things such as computational costs and memory costs into account.

2 Methods in this study

All numerical methods that we have covered so far have a similar theme. With differential equations and initial values, they are able to find slopes of the curves and apply a sort of linearization on it to find an approximation for the next value. While the complexity of developing some of these specific numerical methods has increased, this basic premise seems to have remained.

2.1 Euler's Method

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

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

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 $i = 0, 1, ..., N-1$ and $h = \frac{b-a}{N}$.

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 Euler's method that uses 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 Runge-Kutta method

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 $O(h^2)$.

The most cost-effective Runge-Kutta method is one with 4-th order local truncation error, commonly called RK4. It has the following scheme:

$$w_0 = \alpha$$

$$k_1 = f(t_i, w_i)$$

$$k_2 = f(t_i + \frac{h}{2}, w_i + \frac{h}{2}k_1)$$

$$k_3 = f(t_i + \frac{h}{2}, w_i + \frac{h}{2}k_2)$$

$$k_4 = f(t_{i+1}, w_i + hk_3)$$

$$w_{i+1} = w_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

As can be seen, each step requires four evaluations. But the exponential increase in order of local truncation error makes it a quite worthy computational investment.

2.4 Adams-Bashforth Four-step explicit method

This method is of the family of multistep methods. Multistep methods rely not only on one previous data point, but on multiple older evaluations. They require storage of previous data points to use for evaluation of next one, but since the previous data points are stored, they normally only require a single evaluation of function $f(t_i, w_i)$. Adams-Bashforth Four Step explicit method uses four previous data points to evaluate the next one. Its scheme is:

$$w_0 = \alpha, \qquad w_1 = \alpha_1, \qquad w_2 = \alpha_2, \qquad w_3 = \alpha_3$$

$$w_{i+1} = w_i + \frac{h}{24}(55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f(t_{i-2}, w_{i-2}) - 9f(t_{i-3}, w_{i-3}))$$

And local truncation error is: $\tau_{i+1}(h) = \frac{251}{720}y^{(5)}(\mu_i)h^4$

2.5 Predictor-Corrector method

Predictor-Corrector methods utilize the scheme of multistep implicit evaluations. Multistep implicit methods such as Adams-Moulton 3-step, predict the next value based on the next value. Predictor-Corrector methods utilize this quirk by first predicting the value using some explicit method of similar order and then correcting it with the implicit method. Example we explore here is the Adams 4th-order Predictor-Corrector method. Since implicit and explicit methods rely on the same data points, the additional correction with implicit method does not require much more resources. Given w_0, w_1, w_2, w_3 here are its steps:

$$w_{i+1,p} = w_i + \frac{h}{24}(55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f(t_{i-2}, w_{i-2}) - 9f(t_{i-3}, w_{i-3}))$$

$$w_{i+1} = w_i + \frac{h}{24}(9f(t_{i+1}, w_{i+1,p}) + 19f(t_i, w_i) - 5f(t_{i-1}, w_{i-1} + f(t_{i-2}, w_{i-2}))$$

Since both implicit and explicit methods have error of order $O(h^4)$ Adams Predictor-Corrector local truncation error is of that same order as well.

3 Numerical experiments

3.1 Example 1

Basic evaluation of example with each method providing a table of evaluation and short statement on the type.
Error plot consisting of all methods corrected for computational and memory costs.
Error plots of methods paired up by local truncation error where step sizes are the same(excluding Euler)
This will include RK2(Midpoint) vs Modified and AB-4Step vs RK4 vs PC

Short remarks are allowed between each plot, but are not necessary. Mosto f the discussion will be in the discussion part

3.2 Example 2

Basic evaluation of example with each method providing a table of evaluation and short statement on the type.
Error plot consisting of all methods corrected for computational and memory costs.

Error plots of methods paired up by local truncation error where step sizes are the same(excluding Euler)
This will include RK2(Midpoint) vs Modified and AB-4Step vs RK4 vs PC

3.3 Example 3

Basic evaluation of example with each method providing a table of evaluation and short statement on the type.
Error plot consisting of all methods corrected for computational and memory costs.
Error plots of methods paired up by local truncation error where step sizes are the same(excluding Euler)
This will include RK2(Midpoint) vs Modified and AB-4Step vs RK4 vs PC

4 Discussion

4.1 Performance after step-size adjusted for number of evaluations

methods with local truncation error of higher order seem to perform better even after the step size has been adjusted for number of evaluations. This is due to the exponential growing faster than multiplication

4.2 RK2 vs Modified Euler

The summary of what error tables reveal and why one may be better than the other

4.3 RK4 vs PC(mainly)

Which one does better based on error tables. Advantages of one over the other depending on situation

5 Summary