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 RK4 either went on par with, or outperformed Predictor-Corrector method.

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, we are able to find slopes of the curves and apply a sort of linearization on them to find an approximation for the next value. While the complexity of some of these 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: (1) calculate the slope of the line

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

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

In the following sections are the error plots of implemented methods based on some examples from the book.

3.1 Example 1

Ex. 3, pg. 263 [1]

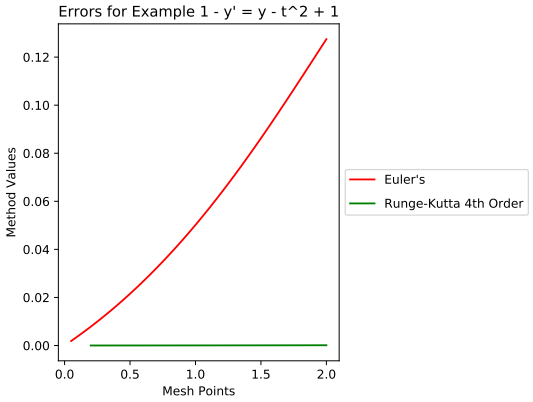

Figure 1: Accuracy of Euler vs RK4 when the former has four times smaller step-size. Ex. 1

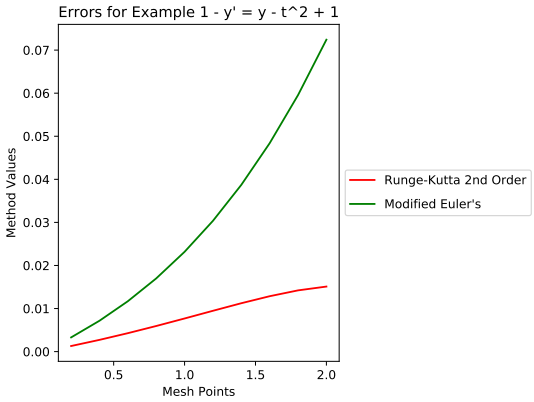

Figure 2: Accuracy of RK2 vs. Modified Euler's method. Ex. 1

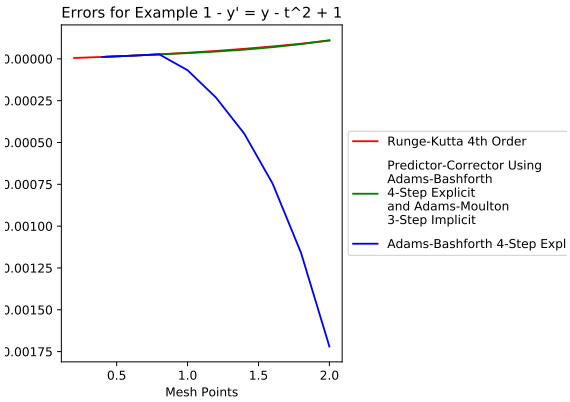

Figure 3: Accuracy of RK4 vs 4th-order Predictor-Corrector vs. Adams-Bashforth 4-step explicit.

3.2 Example 2

5.3. Exercise 10, pg. 282 [1]

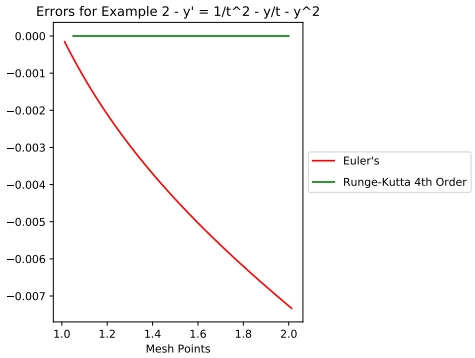

Figure 4: Accuracy of Euler vs. RK4 when the former has four times smaller step-size. Ex. 2

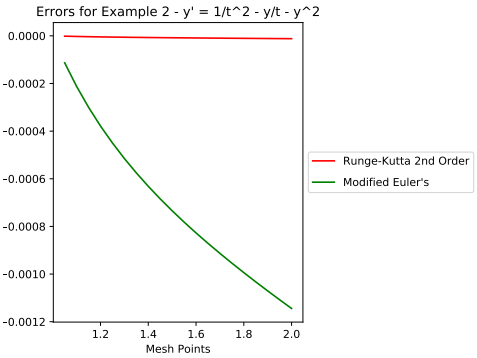

Figure 5: Accuracy of RK2 vs. Modified Euler's method. Ex. 2

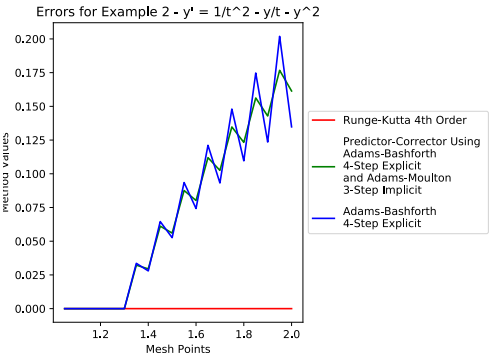

Figure 6: Accuracy of RK4 vs. 4th-order Predictor-Corrector vs. Adams-Bashforth 4-step explicit.

3.3 Example 3

No. 3.d. pg. 330 [1]

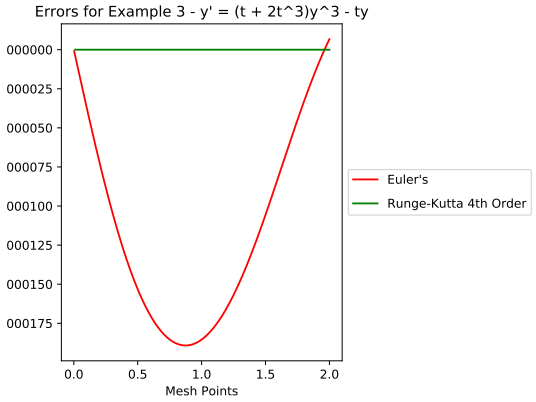

Figure 7: Accuracy of Euler vs RK4 when the former has four times smaller step-size. Ex. 3

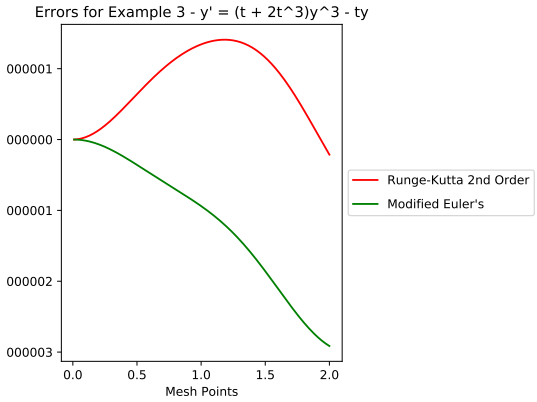

Figure 8: Accuracy of RK2 vs. Modified Euler's method.

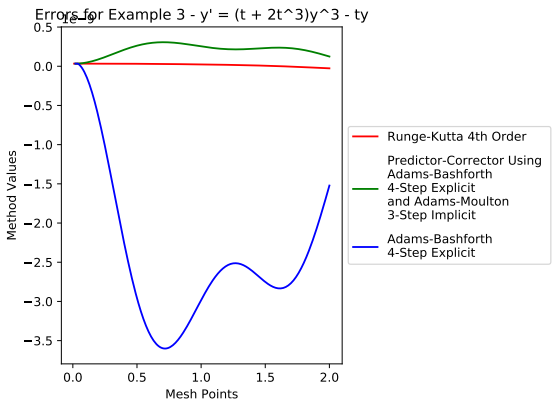

Figure 9: Accuracy of RK4 vs. 4th-order Predictor-Corrector vs. Adams-Bashforth 4-step explicit.

4 Discussion

4.1 Performance after step-size adjustment for number of evaluations

In plots of Euler vs. RK4, Euler method's step size has been reduced by four times in order to match the computational cost of RK4 that performs four function evaluations per step. From the plots, we can see that RK4 still easily outperforms Euler method even after step-size adjustment. Actually, RK4s accuracy is so much higher that in comparison to Euler method it looks like a straight line.

Reason for such staggering difference, is due to the enormous difference in local truncation order of both methods. Euler method has order of $O(h)$ and RK4ı́s order is $O(h^4)$. Exponentiation grows a lot faster than multiplication can which lets us see that at a smaller step-size is not enough for Euler

4.2 RK2 vs. Modified Euler

It is interesting to compare RK2 and Modified Euler's method since they both have similar order for local truncation errors and computational cost. From numerical experiments it can be seen that RK2 performed better in all examples attempted and it did so by a big margin. This most likely has to do with Runge-Kutta's application of weights. Euler just takes two values of the functions and takes an average, while Runge Kutta works more carefully from within. Through iterated evaluations it goes along the function which. This is less error prone than the weighted evaluations from outside of Modified Euler's method.

4.3 RK4 vs. Predictor-Corrector and 4-step Explicit

Originally when I first saw the plots of this, I was surprised for Predictor-Corrector to lose as I thought it performed more evaluatins. But after thinking about results of RK2 vs. Modified Euler's, it is a lot more clear how Runge-Kutta Four beat Predictor-Corrector and Adam's 4-step Explicit methods. They use averages of previous points for their structure. This type of evaluation has resemblance to performing an educated guess. Predictor-Corrector and Adam-Bashforth Four-step explicit methods are both quite efficient in terms of computing power and have a quite good local truncation error size, but RK4's ability to follow along the function always reduced error by orders of magnitude and is an extremely effective method

Additionally, the intermediate size coefficients in Predictor Corrector methods make the method a lot more susceptible to fluctuations. As it can be seen in example two. The equations with the reciprocals can change quickly without much warning. That is why looking back and placing decision based only on the old values is not the best idea for these methods. Each inner evaluation of RK4 gets closer and closer to exact values making it less weak when confronted by the unknown.

5 Summary

With this project, we have implemented various numerical methods. A numerical comparison was performed afterwards based on the examples found in our class' book and slides. We have found out that Runge-Kutta methods generally perform better and are a lot more stable than many other methods we have discovered. Recursive evaluations implemented by Runge-Kutta methods, often allowing them to get to a better evaluation with each step.

6 Reference

[1] R. L. Burden and J. D. Faires. Numerical Analysis 9th edition. Boston, MA: Brooks/Cole, 2011.