P346 Project Report
Adaptive Step Size Runge-Kutta Method

Ratul Das[1,*]

[1] School of Physical Sciences, National Institute of Science Education and Research, HBNI, Jatni, Khordha, Odisha-752050, India

(Dated: December 2, 2021)

The classic Runge-Kutta method (RK4) is a very useful tool in finding numerical solutions to ordinary differential equations accurately, while having a reasonable computational expense. This, along with other numerical methods, were developed around 1900 by Carl Runge and Wilhelm Kutta. However, the method's efficiency is heavily dependent upon the step size it uses. In this project, a version of the RK4 method that can appropriately change its step size, given a specific tolerance value, is discussed in detail.

I. INTRODUCTION

The RK4 method (multiple-step 4th order Runge-Kutta) is found to have no flexibility while it performs computations. The step size, h, once chosen, remains constant during the execution of code for a given problem. There are issues in obtaining accurate solutions when the step size is either too large, or too small.

We know from our coursework in P346 that choosing smaller step sizes and higher order methods are two possible routes to reducing the size of the error we get in our solution. However, one additional way of circumventing this binary restriction is to be able to dynamically change our step size, as required in our problem. We will now discuss such a method that allows us to do that.

II. WORKING PRINCIPLE

III. TEST SCENARIOS

A. Solution for a first-order ODE

We pick a differential equation whose solution curve has at least one sharp bend.

$$\frac{dy}{dt} = t^2 - \frac{3y}{t} \tag{1}$$

The general solution to the differential equation is obtained as,

$$y = \frac{t^3}{6} - \frac{C}{t^3} \tag{2}$$

Solving for $y(1) = \frac{1}{2}$, we see that $C = \frac{1}{3}$. Hence, the solution curve is,

$$y = \frac{t^3}{6} - \frac{1}{3t^3} \tag{3}$$

We run our test by setting the boundary condition as $y(0.1) = 333.33$, which leads to approximately the same solution curve as in equation (3).

B. Earth Hole problem

C. Predator-Prey system

IV. RESULTS

A. Solution for a first-order ODE

The algorithm performs well in this case. It responds to the problem by shortening the step size at the inflection point. It also gradually expands the step size as the curve moves away from the inflection point in the graph of the function. At the end, it is seen that the step size contracts to meet the limit of $t = 10$ that has been set in the parameters for the code.

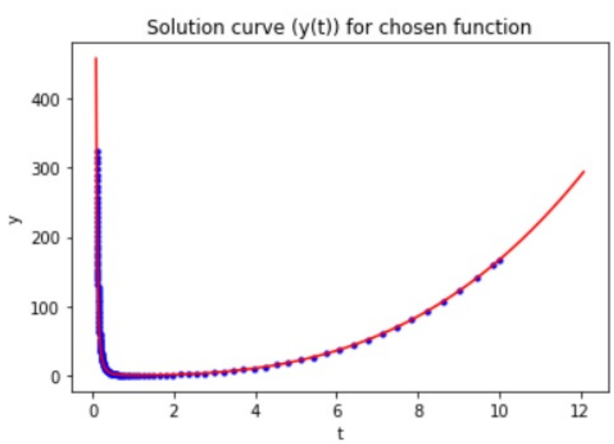

FIG. 1. Solution in Scenario 1 using RKF45

* ratul.das@niser.ac.in

B. Earth Hole problem

It was seen that the computed solution is highly accurate as compared to the analytical solution. The modification made to the code successfully allows for the prevention of the division by zero error when Δ is to be calculated. We also successfully verify the solution to the problem that we obtained while obtaining the analytical solution - it takes almost 5063 seconds for the body dropped from one end of the Earth to return to the same end again. The order of error is seen to be less than 1 part in 10^5 if the position of the body at $t = 5063$s is checked.

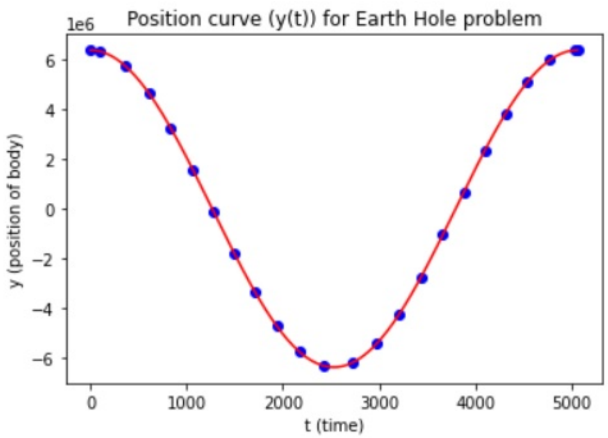

FIG. 2. Solution in Scenario 2 using RKF45

Again, we observe that the step size changes marginally to become smaller when reaching the other side of the Earth, since the position curve experiences an inflection point during that time. It remains large when the body is moving the fastest (near the centre of the Earth).

C. Predator-Prey system

Similar to [2], the birth rate (a) and death rate (c) parameters were varied and the results were recorded for various cases. It was seen that the maximum populations of the prey and predator species increased dramatically with each "period", as opposed to a much slower increase. It is difficult to tell if this is a more accurate or a less accurate solution that the one provided in Juarlin's paper, since no analytical solution exists for this system of equations. However, we can clearly see how the step size changes multiple times during a single period to accommodate for both the prey and predator solution curves.

V. CONCLUSION

We hence come to the conclusion that the Runge-Kutta-Fehlberg modification to the original RK4 algorithm offers some valuable computational discounts in some cases, while resulting in a marginally higher processing time per step. It is a wise idea to compare the performance for such numerical algorithms before using them on much larger data sets, to check the suitability for a given algorithm to a particular problem.

ACKNOWLEDGMENTS

I would like to convey my thanks to Dr Subhashis Basak and to Dr Gunda for their efforts in simplifying the concepts involved in the experiment and providing the measurements for the experiment that were utilised in this report. I would also like to thank my batchmates for the discussions I had with them regarding the theory in this report and also for presenting their queries to Dr Gunda, who answered them promptly and satisfactorily for the whole batch. I would like to thank the developers of Matplotlib for the code that has been used in creating the plots in Figure 5 and 6.

[1] E. Fehlberg, "Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems", NASA Technical Reports, TR R-315 (1969).

[2] E. Juarlin, "Solution of Simple Prey-Predator Model by Runge-Kutta Method", J. Phys.: Conf. Ser **1341** 062024 (2019).

[3] I. Bishop, S. Xian, and S. Feller, Phys. Teach. **57**, 442 (2019).

[4] Student's Sheet for Elementary charge and Millikan experiment, Phywe Series of Publications, Phywe Systeme GmbH (2017).