P346 PROJECT REPORT
Adaptive step-size Runge-Kutta method

Pabitra Ranjan Giri

Roll - 2011103
SPS,NISER

Abstract

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.The Runge–Kutta methods are a family of implicit and explicit iterative methods, which include the Euler method, used in temporal discretization for the approximate solutions of simultaneous nonlinear equations.These methods were developed around 1900 by the German mathematicians Carl Runge and Wilhelm Kutta. 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 accuracy, is discussed in detail. Also, I tested a modified RK45 code for the same problems.

1 Introduction

The RK4 method (multiple-step 4th order Runge-Kutta) has no flexibility while it performs computations. The step size, h ,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. The result differs depending on the step size. We can decrease the step size values to reduce error, but it is extremely time consuming in case of many body problems and multiple variables. This issue can be solved by using a code which dynamically changes the step size for the numerical differential equation solver, when given a fixed amount of accuracy for the solution. We will now discuss such a method that allows us to do the same, which is known as the Runge-Kutta-Fehlberg (RK45) method. Now onto to the modified RK45 code, this algorithm also uses the same idea of dynamic step-size, but instead of reducing the number of steps, it focuses on precision . This enables to achieve more precision without having to use higher order terms(i.e. k1,k2....) in the Runge-Kutta equations.

2 Working principle

The method described by Fehlberg, is more efficient compared to classic RK4 algorithm. Adaptive methods such as Fehlberg's algorithm(also known as RK45) are designed to produce an estimate of the local truncation error of a single Runge–Kutta step. This is done by having two methods, one with order p and one with order p-1. These methods they have common intermediate steps. Using this method estimating the error has little or negligible computational cost compared to a step with the higher-order method. During the integration, the step size is adapted such that the estimated error stays below a user-defined threshold: If the error is too high, a step is repeated with a lower step size; if the error is much smaller, the step size is increased to save time. The code determines the optimal step size for each step, which saves computation time. RKF45 is a 4th order method, but it has a 5th order error estimator. With the performance of an extra calculation, the solution's error can be estimated and regulated accordingly by using the higher-order embedded method that allows for a step size to be determined dynamically.

The modified RK45 reduces the step size if it encounters rapid change in values of the differential equation, and increases if the change in values is not much. It changes the step-size until a specific range of values which is determined by the previous step values and iteration values of the differential equation.

3 Derivations

In general, (p−1)th order adaptive methods require a pth order step in order to compute the local truncation error for a step in question. Since these two steps occur in a single, interwoven step for a loop, a lower computational cost is entailed that what would've been entailed for a method of a higher order. During the execution of the integration, the step size is changed in such a way so as to keep the estimated error below a pre-determined threshold. If the error is higher than permissible, the step is repeated with a lower step size. if the error is small, the step size is increased to reduce the computational cost. This results in the maintenance of an optimal step size, which saves time spent in computation. Hence, additional effort is not needed to find an appropriate step size for a given problem.

The higher order step is computed as,

$$y_{n+1} = y_n + h\Sigma_{i=1}^{s} b_i k_i$$

We also know that the lower order step is computed as,

$$y'_{n+1} = y_n + h\Sigma_{i=1}^{s} b'_i k_i$$

Here, k_i are the same as for the higher order method. Hence, we get the error between the two for the (n+1)th step as,

$$e_{n+1} = y_{n+1} - y'_{n+1} = h\Sigma_{i=1}^{s}(b'_i - b_i)k_i$$

And here, e is the higher order estimator. For this family of adaptive methods, it is seen that the general Butcher tableau is given as,

$$\begin{array}{c|cccccc}
0 & & & & & \\
c_2 & a_{2,1} & & & & \\
c_3 & a_{2,1} & a_{3,2} & & & \\
. & . & . & & & \\
. & . & & . & & \\
. & . & & & . & \\
c_s & a_{s,1} & a_{s,2} & \dots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \dots & b_{s-1} & b_s \\
 & b'_1 & b'_2 & \dots & b'_{s-1} & b'_s
\end{array}$$

Hence, we can write out the Butcher tableau for RKF45 as,

$$\begin{array}{c|cccccc}
0 & & & & & & \\
1/4 & 1/4 & & & & & \\
3/8 & 3/32 & 9/32 & & & & \\
12/13 & 1932/2197 & 7200/2197 & 7296/2197 & & & \\
1 & 439/216 & 8 & 3680/513 & -845/4104 & & \\
1/2 & 8/27 & 2 & 3544/2565 & 1859/4104 & 11/40 & \\
\hline
 & 16/135 & 0 & 6656/12825 & 28561/56430 & 9/50 & 2/55 \\
 & 25/216 & 0 & 1408/2565 & 2197/4104 & 1/5 & 0
\end{array}$$

These coefficients are applied in the code to find the increments and the summations of the weights for the increments, which is then used to compute the appropriate adjustment for the step size. The truncation error (E), which is the 5th order estimator, is calculated using the coefficients obtained in the embedded pair (the lower two rows, or weights) as,

From the truncation error, we compute the new step size as,

Here, the k factor can be between 0.8 and 1.0, depending upon the amount of fluidity required in changing the step size. If E $> \epsilon$, then we replace h with h_{new} and repeat the step. But if E $\leq \epsilon$, then the step is completed successfully and we replace h with h_{new} for the following step.

4 Solving Problems and results

4.1 1st order ODE

The ODE for the first testing is,

$$\frac{dy}{dx} = \frac{y}{2} + 2\sin 3x$$

The general solution for the differential equation is,

$$y = \frac{(-24\cos(3x) - 4\sin(3x) + 24\exp(x/2))}{37} + c$$

Using boundary condition, y(0)=0.1 ,
The actual solution for the differential equation is,

$$y = \frac{(-24\cos(3x) - 4\sin(3x) + 24\exp(x/2))}{37} + 0.1\exp(x/2)$$

Value of y, for x=5,
Theoretical solution: 9.542876790796473

Classic RK4:
No. of steps: 52
Estimated value: 10.140566385219437

RK45:
No. of steps: 48
Estimated value: 10.21499477469368

Modified RK45:
No. of steps: 68
Estimated value: 9.542738027559297

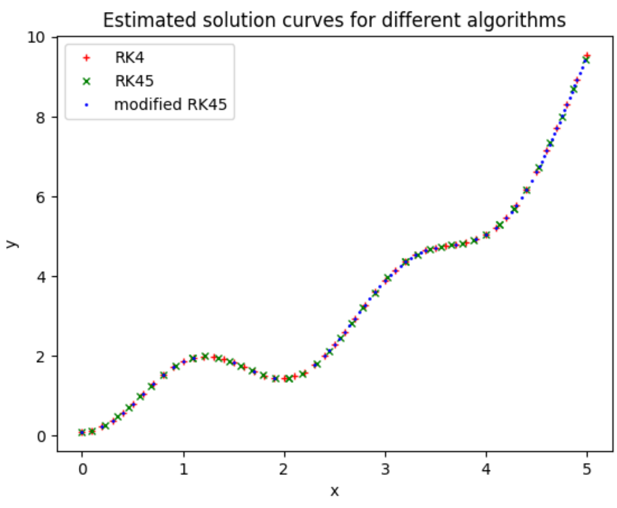

Figure 1

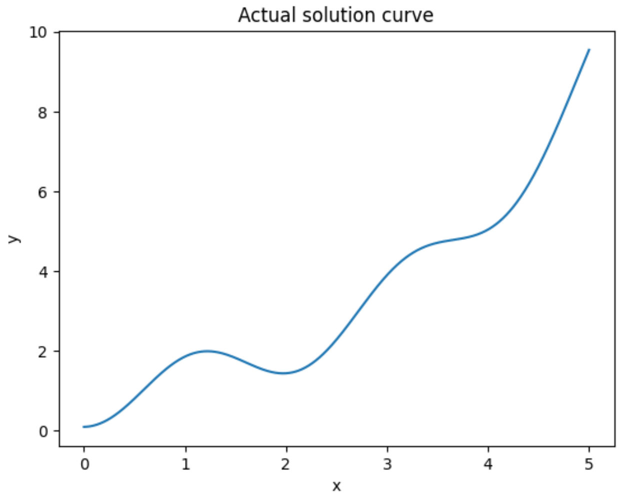

Figure 2

4.2 RLC circuit problem

Question:
Find the charge on the capacitor in an RLC series circuit where L=5/3 H, R=10 , C=1/30 F, and E(t)=300 V. Assume the initial charge on the capacitor is 0 C and the initial current is 9 A. What happens to the charge on the capacitor over time?

Now to solve this question we have to write the differential equation for the charge on the capacitor,

$$L\frac{d^2q}{dt^2} + R\frac{dq}{dt} + \frac{1}{C}q = E(t)$$

$$\frac{d^2q}{dt^2} + 6\frac{dq}{dt} + 18q = 180$$

The general solution to the complementary equation is

$$e^{-3t}(c_1\cos(3t) + c_2\sin(3t))$$

Assume a particular solution of the form q_p=A , where A is a constant. Using the method of undetermined coefficients, we find A=10 . So,

$$q(t) = e^{-3t}(c_1\cos(3t) + c_2\sin(3t)) + 10$$

Applying the initial conditions q(0)=0 and i(0)=((dq)/(dt))(0)=9, we find c_1=10 and c_2=7. So the charge on the capacitor is

$$q(t) = -e^{-3t}(10\cos(3t) + 7\sin(3t)) + 10$$

Looking closely at this function, we see the first two terms will decay over time (as a result of the negative exponent in the exponential function). Therefore, the capacitor eventually approaches a steady-state charge of 10 C.

We solved for t = 10 s,
Theoretical solution: 10.000000000000503

Classic RK4:
No. of steps: 1002
Estimated value: 10.00000000000046

RK45:
No. of steps: 845
Estimated value: 9.999999993011

Modified RK45:
No. of steps: 372
Estimated value: 10.063741173776496

For more accuracy using the modified RK45 , we changed the error limits,but the number of steps exploded,and yet less accurate then original RK45.
Modified RK45:
No. of steps: 80645
Estimated value: 10.0000759381297

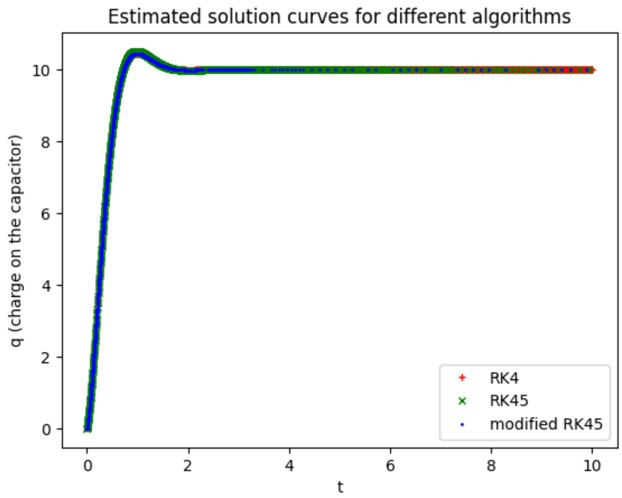

Figure 3

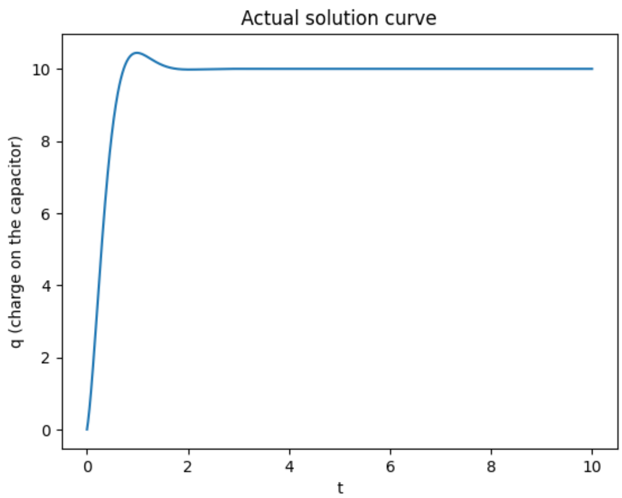

Figure 4

5 Conclusion

- The RK45 code reduces the number of steps required to find the solution, but the no. of steps reduced is not much, rather it consumes time due to more calculations per step.
- The modified RK45 as expected increases the number of steps, and the it gives a very accurate result in first problem, but does not work properly for the second problem.

6 References

- Applications of Second-Order Differential Equations , Gilbert Strang Edwin "Jed" Herman OpenStax https://math.libretexts.org/Bookshelves/Calculus/Book%3A_Calculus_(OpenStax)/17%3A_Second-Order_Differential_Equations/17.03%3A_Applications_of_Second-Order_Differential_Equations
- Wikipedia https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods