www.simiode.org
SIMIODE A SYSTEMIC INITIATIVE FOR MODELING INVESTIGATIONS & OPPORTUNITIES WITH DIFFERENTIAL EQUATIONS

STUDENT VERSION

Numerical Methods of Various Orders: A Comparison

Swarn Singh
Department of Mathematics
Sri Venkateswara College, University of Delhi
New Delhi 110021 INDIA
sswarn2005@yahoo.co.in, ssingh@svc.ac.in

Abstract: It is not always possible to solve a differential equation analytically.This material makes an effort to teach the basics of numerical methods for first order differential equations by following graphical and numerical approaches. Here we also discuss the order of accuracy of the methods and compare their CPU times.

INTRODUCTION

BACKGROUND

When it is not possible to find analytic solutions for a differential equation, numerical methods are often used. When an analytic solution of a differential equation is possible, but it is time-consuming, then the approximation obtained with a numerical solution could be appropriate. To solve a given differential equation, we replace the derivatives by finite differences, based on the limit definition of the derivative. The methods we offer provide algebraic equations for implementation on computers.

METHODS AND IMPLEMENTATION

The steps involved in finite difference methods are:

- discretize the continuous domain into a grid,
- approximate the derivatives by finite differences, based on the limit definition of the derivative,
- substitute the approximations in the differential equation to obtain algebraic equations, and
- solve the resulting system of finite difference equations.

Before proceeding further, let us first clarify what we mean by accuracy and order of accuracy. We first consider absolute error:

$$\text{Absolute error} = |\text{analytic solution} - \text{numerical solution}|.$$

A numerical method is said to be *r-th order (or order r)* if there is a number C independent of h such that the error $E(h)$ behaves like

$$E(h) \approx Ch^r$$

Hence, the larger the order of accuracy, the faster the error reduces as h decreases. One can also find the order of the method using the formula

$$r = \frac{\ln(E(h_2)/E(h_1))}{\ln(h_2/h_1)}\,,$$

where h_1 and h_2 are two small step sizes for numerical solutions.

Consider the initial value problem

$$y' = f(x, y), y(0) = y_0\,.$$

To approximate the solution of the initial value problem we use the following methods based on finite differences.

1. Euler's method

Divide the solution domain $[a, b]$ into n subintervals. The idea behind Euler's method is to use the tangent line to the solution at the independent variables x_{i+1} as intervals in $[a, b]$. To find the coordinates of the points in our numerical solution, we use the initial condition as our starting point and generate the rest of the solution by using the iterative formulae,

$$\begin{aligned} x_{i+1} &= x_i + h \qquad (1)\\ y_{i+1} &= y_i + h * f(x_i, y_i)\,. \end{aligned}$$

Terminate this process when we have reached the right hand end of the desired interval.

Let us work on a simple numerical method example completely by hand first.

$$y' = x + 2y, \quad y(0) = 0 \qquad (2)$$

to find the value for the approximate numerical solution for (2) at $x = 1$.

Incidentally, the analytic solution of (2) can be found using integrating factors or a computer algebra system to be

$$y(x) = \frac{1}{4}\left(-2x + e^{2x} - 1\right)\,, \qquad (3)$$

and we can compare both analytic and numerical solutions.

We now apply Euler's method to estimate a solution in $[0, 1]$. Here $f(x, y) = x + 2y$ and the initial condition tells us the values of the coordinates of our starting point: $x_0 = 0, y_0 = 0$.

We use Euler's method formulae, (1), to generate values for x_1 and y_1 using steps of size $h = 0.25$.

For the first iteration, when $i = 0$ we have $x_1 = x_0 + h$ or $x_1 = x_0 + 0.25 = 0.25$ so, $y_1 = y_0 + hf(x_0, y_0) = 0$. Thus, the numerical solution at $x_1 = 0.25$ goes through the point $(0.25, 0)$. When you get to your estimate for $y(0.25)$ how does it compare to the analytic solution from (3), i.e. $y(0.25) = 0.03718$?

Now for the second iteration $x_2 = x_1 + h$ and $x_2 = 0.25 + 0.25 = 0.5$ so, $y_2 = y_1 + hf(x_1, y_1) = 0 + 0.25 * (x_1 + 2 * y_1) = 0.25 * 0.25 = 0.0625$. So the numerical solution at $x_2 = 0.5$ goes next through the point $(0.5, 0.0625)$. This compares with the analytic solution from (3), i.e. $y(0.5) = 0.17957$?

We now move on to get the third point in the solution (x_3, y_3) and go further to our final point $(x_4 = 1, y_4 = ?)$. When you get to your estimate, y_4, for $y(1)$ how does it compare to the analytic solution from (3), i.e. $y(1) = 1.09726$? We compare our numerical estimate approach with the analytic solution graphically in Figure 1 and note the increasing errors in our approximation as x increases.

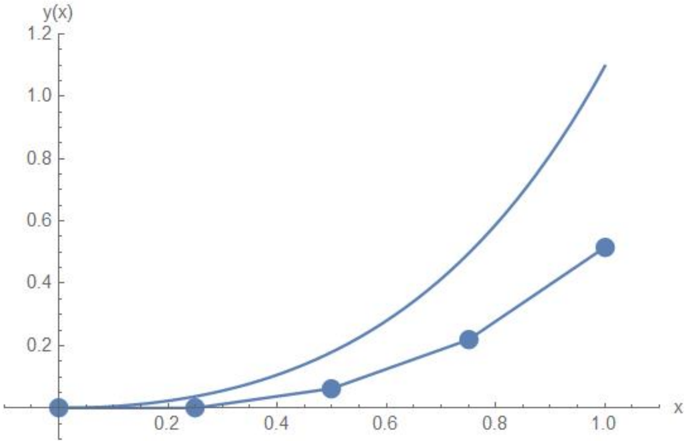

Figure 1. Comparison of numerical estimate approach (straight line segments with iteration points in large dots) with the analytic solution (solid curved line).

Activities

1. Summarize the results of all the calculations in a tabular form, as in Table 1.
2. Plot the graph of the solution.
3. Compare the solution obtained with the analytic solution.

i	x_i	y_i
0		
1		
2		
3		
4		

Table 1. Table of Numerical Solution using Euler's method.

2. Improved Euler's Second Order method

Use the formula from Euler's method to obtain a first approximation to the solution y_{i+1} as

$$y^*_{i+1} = y_i + h * f(x_i, y_i)\,.$$

Now improve this approximation once more by applying Euler's method and using the average of $f(x_i, y_i)$ and $f(x_{i+1}, y^*_{i+1})$ as

$$y_{i+1} = y_i + \frac{1}{2} h * [f(x_i, y_i) + f(x_{i+1}, y^*_{i+1})],$$

Activities

4. Solve the example given in the previous section using improved Euler's method and compile the solution in a tabular form.
5. Plot the graph of the solution.
6. Compare the solution obtained with the analytic solution and compute the absolute error between the numerical and analytic solution.
7. Obtain the order of accuracy of the method.

3. Runge Kutta Third Order method

The Runge Kutta Third Order (RK3) method is as follows.

$$\begin{aligned} y_{i+1} &= y_i + \frac{1}{6}(k_1 + 4 * k_2 + k_3) \quad \text{where} \\ k_1 &= h * f(x_i, y_i), \\ k_2 &= h * f(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}), \text{ and} \\ k_3 &= h * f(x_i + h, y_i - k_1 + 2 * k_2)\,. \end{aligned}$$

Activities

8. Solve the example given in the section 1 using RK3 method and compile the solution in a tabular form.

9. Plot the graph of the solution.
10. Compare the solution obtained with the analytic solution and compute the absolute error.
11. Obtain the order of accuracy of the method.

4. Runge Kutta fourth order method

The Runge Kutta fourth order (RK4) method is as follows.

$$\begin{aligned}
y_{i+1} &= y_i + \frac{1}{6}(k_1 + 2 * k_2 + 2 * k_3 + k_4) \quad \text{where} \\
k_1 &= h * f(x_i, y_i), \\
k_2 &= h * f(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}), \\
k_3 &= h * f(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}), \text{ and} \\
k_4 &= h * f(x_i + h, y_+ k_3)\,.
\end{aligned}$$

Activities

12. Solve the example using RK4 method.
13. Plot the table and graph of the solution.
14. Compare the solution obtained with the analytic solution and compute the absolute error between the numerical and the analytic solution.
15. Obtain the order of accuracy of RK4 method.

COMPARISON OF THE METHODS

Activities

16. Solve the example given in Section 1 with all the four methods and calculate the absolute errors obtained between the analytic and numerical solutions. Make a single table of these.
17. Compare the accuracy of all the methods and check the order of accuracy. Describe the importance of order of accuracy of the methods.
18. For all the methods plot the graphs of error versus iteration.
19. Solve the example by using software packages Matlab, Mathematica, Maple, etc. or see (2).
20. Calculate CPU time taken by each method to get the desired accuracy and complete Table 2.
21. Compare the accuracy and the CPU time taken for all the methods.

Final Activity

Consider the non-linear IVP

$$y' = (\sin^2 y + \cos x)y, \quad y(0) = 1\,.$$

i	x_i	y_i	Error	CPU Time
0				
1				
2				
3				
4				

Table 2. Table of Numerical Solution and CPU time.

Implement the above methods and calculate absolute error and CPU time.

PREREQUISITES

Before starting this activity, it may be helpful (but not required) for students to know how to

- find analytic solutions of ordinary differential equations and
- use software packages such as Matlab/Mathematica/Maple to solve first order ODE's

OBJECTIVES

Upon successful completion of this activity, students will be able to

- know about the importance of learning higher order methods with respect to fewer number of iterations involved to obtain the desired accuracy and
- know about the costs involved in applying higher order methods in terms of CPU time.

REFERENCES

[1] Bradie B. 2007. *A Friendly Introduction to Numerical Analysis*. Delhi INDIA: Pearson Education.

[2] Jain M.K., S.R.K. Iyengear, and R.K. Jain. 1996. *Numerical Methods for Scientific and Engineering Computation*. Delhi INDIA: New Age International Pvt. Ltd.