Numerical Solutions and Analytical Solutions of First Order Ordinary Differential Equation by Taylor's Series Method and Runge-Kutta Method

Khaing Khaing Thant
University of Computer Studies (Monywa)
khaingkhaing2019m@gmail.com

Mar Mar Myint
Technological University (Taungoo)
marmyint17@gmail.com

Lei Lei Mon
University of Computer Studies (Thaton)
lailaimon95@gmail.com

Abstract

Taylor's Series method and Runge-Kutta method are presented. Taylor's Series method is explained with one example. Runge-Kutta method is explained with one example. Numerical solutions and analytical solutions of the ordinary differential equation by Taylor's Series method at $\boldsymbol{h = 0.001}$ and $\boldsymbol{h = 0.01}$ are computed by using Matlab and compared. And then numerical solutions and analytical solutions of the ordinary differential equation by Runge-Kutta method at $\boldsymbol{h = 0.001}$ and $\boldsymbol{h = 0.01}$ are computed by using Matlab and compaired. Moreover, analytical solutions of the ordinary differential equation from $\boldsymbol{x = 0}$ to $\boldsymbol{x = 0.1}$ and from $\boldsymbol{x = 0}$ to $\boldsymbol{x = 0.01}$ are drawn as the figures by using Matlab. The error at $\boldsymbol{h = 0.001}$ and the error at $\boldsymbol{h = 0.01}$ are calculated. The error at $\boldsymbol{h = 0.001}$ is smaller than the error at $\boldsymbol{h = 0.01}$.

1. Introduction

Taylor's series method and Runge-Kutta method are introduced,[7]. Taylor's series method is presented by [2], [3], [5], [6], [7], [8] and Runge-Kutta method is presented by [2], [4], [5], [6], [7], [8]. Many ordinary differential equations can be solved by analytical methods discussed earlier giving closed form solutions that is expressing $\mathbf{y}$ in terms of a finite number of elementary functions of $\mathbf{x}$. However, a majority of differential equations appearing in physical problems cannot be solved analytically. Thus, it becomes imperative to discuss their solution by numerical methods. Two methods for obtaining numerical solutions of first order and first degree ordinary differential equations are discussed.

2. Taylor's Series Method and Runge-Kutta Method

To improve on the speed of convergence of Euler's method, we look for approximations to $\mathrm{Y}(x_{n+1})$ that are more accurate than the approximation $\mathrm{Y}(x_{n+1}) \approx \mathrm{Y}(x_n) + h\,\mathrm{Y}'(x_n)$, which lead to Euler's method. Since this is a linear Taylor polynomial approximation, it is natural to consider higher-order Taylor approximations. Doing this will lead to a family of methods, are called the Taylor method, depending on the order of the Taylor approximation being used. In deriving a Taylor method, we need higher-order derivatives of the true solution, and we obtain them using the solution itself by differentiating the differential equation. Such expressions for higher-order derivatives are usually time-consuming. The idea of Runge–Kutta method is to use combinations of compositions of the right-side function of the equation to approximate the derivative terms to a required order. The resulting Runge–Kutta method are among the most popular methods in solving initial value problems, [7].

2.1. Taylor's Series Method

Consider an initial value problem described by

$$y' = f(x, y)\,,\ y(x_0) = y_0. \tag{1}$$

The Taylor's expansion of the function $y(x)$ in the neighbourhood of (x_0, y_0) is given by

$$y(x) = y_0 + (x - x_0)y'(x_0) + \frac{(x-x_0)^2}{2!}y''(x_0) + \frac{(x-x_0)^3}{3!}y'''(x_0) + \dots \tag{2}$$

where dashes denote differentiation with respect to x.

Differentiating equation (1) successively with respect to x, we get

$$\left.\begin{aligned} y'' &= \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial x} = \left(\frac{\partial}{\partial x} + f\frac{\partial}{\partial y}\right)f \\ y''' &= \left(\frac{\partial}{\partial x} + f\frac{\partial}{\partial y}\right)\left(\frac{\partial f}{\partial x} + f\frac{\partial}{\partial y}f\right) \\ &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial f}{\partial x}\frac{\partial f}{\partial y} + 2f\frac{\partial^2 f}{\partial x \partial y} + f\left(\frac{\partial f}{\partial y}\right)^2 + f^2\frac{\partial^2 f}{\partial y^2} \end{aligned}\right\} \tag{3}$$

and so on. Putting $x = x_0$ and $y = y_0$ in the expressions for $y', y'', y''', \dots$, and substituting them in equation (2), we get a power series for $y(x)$ in power at $(x - x_0)$

$$y(x) = y_0 + (x - x_0)y'(x_0) + \frac{(x-x_0)^2}{2!}y''(x_0) + \frac{(x-x_0)^3}{3!}y'''(x_0) + \cdots$$

Putting $x = x_1 = x_0 + h$. we get

$$y_1 = y(x_1) = y_0 + \frac{h}{1!}y_0' + \frac{h^2}{2!}y_0'' + \frac{h^3}{3!}y_0''' + \cdots \tag{4}$$

Here $y_0'\,, y_0'', y_0''', \dots$ can be found by using equation (1) and its successive differentiations equation (3) at $x = x_0$. The series (4) can be truncated at any stage if h is small. If we get y_1, we can calculate $y_1'\,, y_1'', y_1''', \dots$ from equation (1) at $x = x_0 + h$. Expanding $y(x)$ by Taylor's series about $x = x_1$,

$$y_2 = y_1 + \frac{h}{1!}y_1' + \frac{h^2}{2!}y_1'' + \frac{h^3}{3!}y_1''' + \cdots \tag{5}$$

Proceeding on, we get

$$y_n = y_{n-1} + \frac{h}{1!}y'_{n-1} + \frac{h^2}{2!}y''_{n-1} + \frac{h^3}{3!}y'''_{n-1} \qquad (6)$$

where $y^r_{n-1}=\left[\frac{d^r}{dx^r}y_{n-1}\right]_{(x_{n-1},y_{n-1})}$.

By taking sufficient number of terms in the above series, the value of y_n can be get, without mush error. If we retain the terms up to h^non the right-hand side of equation (5) the error will be proportional to the $(n+1)$ -th power of step size, h^{n+1} and the Taylor's algorithm is said to be of n-th order. By including more number of terms on the right hand side of equation (6), the error can be reduced further, [2], [3], [5], [6], [7], [8].

2.1.1.Example

Case (i)

We consider the initial value problem $y' = x^2 + y$, $y(0) = 1$ at $x = 0.01, 0.02, 0.03, \dots ,0.1$.

Thus $x_0 = 0, y_0 = 1, x_1 = 0.01, x_2 = 0.02, \dots,\ x_{10} = 0.1$ and $h = 0.01$. The first few derivatives are computed as follows:

$y' = x^2 + y$, thus $y'_0 = x_0^2 + y_0 = 1$.

$y'' = 2x + y'$, thus $y''_0 = 2x_0 + y'_0 = 1$.

$y''' = 2 + y''$, thus $y'''_0 = 2 + y''_0 = 3$.

The third order Taylor's algorithm is considered.

The Taylor series for $y(x)$ near $x = 0$ is

$$y_1 = y_0 + \frac{h}{1!}y'_0 + \frac{h^2}{2!}y''_0 + \frac{h^3}{3!}y'''_0 .$$

$y_1 = 1.0100505$.

By successive approximation, we get

$y_2 = 1.020204018$, $y_3 = 1.030463598$,

$y_4 = 1.040832317$, $y_5 = 1.051313282$,

$y_6 = 1.061909631$, $y_7 = 1.072624534$,

$y_8 = 1.083461192$, $y_9 = 1.094422839$,

$y_{10} = 1.105512741$.

Case (ii)

We consider the initial value problem $y' = x^2 + y$, $y(0) = 1$at $x = 0.001, 0.002, , \dots , 0.01$.

Thus $x_0 = 0, y_0 = 1, h = 0.001$,

$x_1 = 0.001, x_2 = 0.002, \dots , x_{10} = 0.01$. The first few derivatives are computed as follows:

$y' = x^2 + y$, thus $y'_0 = x_0^2 + y_0 = 1$.

$y'' = 2x + y'$, thus $y''_0 = 2x_0 + y'_0 = 1$.

$y''' = 2 + y''$, thus $y'''_0 = 2 + y''_0 = 3$.

The third order Taylor's algorithm is considered.

The Taylor series for $y(x)$ near $x = 0$ is

$$y_1 = y_0 + \frac{h}{1!}y'_0 + \frac{h^2}{2!}y''_0 + \frac{h^3}{3!}y'''_0.$$

$y_1 = 1.001000501$.

By successive approximation, we get

$y_2 = 1.002002005$, $y_3 = 1.003004515$,

$y_4 = 1.004008034$, $y_5 = 1.005012565$,

$y_6 = 1.006018111$, $y_7 = 1.007024675$,

$y_8 = 1.00803226$, $y_9 = 1.009040869$,

$y_{10} = 1.010050505$.

The given linear equation is $y' = x^2 + y$.

Integration factor is $e^{\int(-1)dx} = e^{-x}$ and the solution is

$ye^{-x} = \int e^{-x}x^2dx + C$ or

$y = (-x^2 - 2x - 2) + Ce^x$.

Given that $y(0) = 1$ implies $C = 3$. Hence the exact solution is $y = [-(x + 1)^2 - 1] + 3e^x$.

The values of y ; from $x = 0.01$ to $x = 0.1$ are described in the following Table 1.

Table 1. The values of y; from $x = 0.01$ to $x = 0.1$

	y		
x	$h = 0.01$ Exact solution	$h = 0.01$ Taylor's solution	Error
0.01	1.01005051	1.0100505	0.000000001
0.02	1.02020402	1.020204018	0.000000002
0.03	1.030463602	1.030463598	0.000000004
0.04	1.040832323	1.040832317	0.000000006
0.05	1.051313289	1.051313282	0.000000007
0.06	1.061910964	1.061909631	0.000000009
0.07	1.072624544	1.072624534	0.000000010
0.08	1.083461203	1.083461192	0.000000011
0.09	1.094422851	1.094422839	0.000000012
0.1	1.105512754	1.105512741	0.000000013

The values of y ; from $x = 0.001$ to $x = 0.01$ are described in the following Table 2.

Table 2. The values of y; from $x = 0.001$ to $x = 0.01$

	y		
x	$h = 0.001$ Exact solution	$h = 0.001$ Taylor's solution	Error
0.001	1.00100501	1.001000501	0
0.002	1.002002004	1.002002005	−0.000000001
0.003	1.003004514	1.003004515	−0.000000001
0.004	1.004008032	1.004008034	−0.000000002
0.005	1.005012563	1.005012565	−0.000000002
0.006	1.006018108	1.0060180111	−0.000000003
0.007	1.007024672	1.007024675	−0.000000003
0.008	1.008032257	1.00803226	−0.000000003
0.009	1.009040865	1.009040869	−0.000000004
0.01	1.010050501	1.010050505	− 0.000000004

In this example, $h = 0.001$ and $h = 0.01$ are computed. The error at $h = 0.001$ smaller than the error at $h = 0.01$. Therefore the numerical solutions of $h = 0.001$ are better approximations.

2.2. Runge- Kutta Methods

Runge-Kutta methods are more accurate methods of great practical importance. They do not require the computation of higher order derivatives as in Taylor's series method, rather they require evaluation of $f(x, y)$. They agree with Taylor's series up to the terms of h^r, where r is the different for different methods and is known as the order of Runge-Kutta method. Euler's method, Modified Euler's method and Runge's method are the Runge-Kutta method of first, second and third order respectively. The fourth order Runge-Kutta method is widely used in practice and is often referred to as the Runge-Kutta method only without any reference to the order.

We consider the initial value problem

$$y' = f(x, y) \tag{7}$$

where $y(x_0) = y_0$.

To compute y_1

$$k_1 = hf(x_0, y_0), k_2 = hf(x_0 + \tfrac{1}{2}h, y_0 + \tfrac{1}{2}k_1),$$

$$k_3 = hf(x_0 + \tfrac{1}{2}h, y_0 + \tfrac{1}{2}k_2),$$

$$k_4 = hf(x_0 + h, y_0 + k_3)$$

$$k = \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

then $y_1 = y(x_0 + h) = y_0 + k$.

To compute y_2

$$k_1 = hf(x_1, y_1), k_2 = hf(x_1 + \tfrac{1}{2}h, y_1 + \tfrac{1}{2}k_1),$$

$$k_3 = hf(x_1 + \tfrac{1}{2}h, y_1 + \tfrac{1}{2}k_2),$$

$$k_4 = hf(x_1 + h, y_1 + k_3)$$

$$k = \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

then $y_2 = y_1 + k$ and so on.

It will be notified that the only change in the formulae for different intervals is in the values of x and y. Thus to find k in i-th interval, we should substitute x_{i-1}, y_{i-1} in the expressions for k_1, k_2, k_3 and k_4, [2], [4], [5], [6], [7], [8].

2.2.1.Example

Case (i) We consider the initial value problem

$y' = x^2 + y, y(0) = 1$

at $x = 0.01, 0.02, 0.03, \dots, 0.1,$

using fourth order Runge-Kutta method.

Thus $x_0 = 0, y_0 = 1, x_1 = 0.01, x_2 = 0.02, \dots, x_{10} = 0.1,$ and $h = 0.01$.

As first step, we calculate

$k_1 = hf(x_0, y_0) = 0.01,$

$k_2 = hf\left(x_0 + \tfrac{1}{2}h, y_0 + \tfrac{1}{2}k_1\right) = 0.0100525,$

$k_3 = hf\left(x_0 + \tfrac{1}{2}h, y_0 + \tfrac{1}{2}k_2\right) = 0.010050501,$

$k_4 = hf(x_0 + h, y_0 + k_3) = 0.010101505,$

$k = \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 0.010050501$. Using Runge Kutta method, we get

$y_1 = y(x_0 + h) = y_0 + k = 0.010050501$. By successive approximation, we get

$y_2 = 1.02020402, y_3 = 1.030463602,$

$y_4 = 1.040832323, y_5 = 1.05131329,$

$y_6 = 1.0619109641, \quad y_7 = 1.072624545,$

$y_9 = 1.094422845, y_{10} = 1.10551275.$

Case (ii) We consider the initial value problem

$y' = x^2 + y, \ y(0) = 1$

at $x = 0.001, 0.002, \dots, 0.01.$

Thus $x_0 = 0, y_0 = 1, x_1 = 0.001, x_2 = 0.002, \dots,$ $x_{10} = 0.01,$ and $h = 0.001$.

As first step, we calculate

$k_1 = hf(x_0, y_0) = 0.001,$

$$k_2 = hf\left(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_1\right) = 0.00100050025,$$

$$k_3 = hf\left(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_2\right) = 0.001000501,$$

$$k_4 = hf(x_0 + h, y_0 + k_3) = 0.001001501,$$

$$k = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 0.001000583833$$

Using Runge-Kutta method, we get

$$y_1 = y(x_0 + h) = y_0 + k = 1.001000501.$$

By successive approximation, we get

$$y_2 = 1.002002005 ,$$

$$y_3 = 1.003004515, \; y_4 = 1.004008032,$$

$$y_5 = 1.005012565, \; y_6 = 1.006018111,$$

$$y_7 = 1.007024675, \quad y_8 = 1.00803226,$$

$$y_9 = 1.009040869, \; y_{10} = 1.010050505.$$

The values of y ; from $x = 0.01$ to $x = 0.1$ are described in the following Table 3.

Table 3. The values of y ; from $x = 0.01$ to $x = 0.1$

	y		
x	$h = 0.01$ Exact solution	$h = 0.01$ Runge-Kutta solution	Error
0.01	1.010050501	1.010050501	0
0.02	1.02020402	1.02020402	0
0.03	1.030463602	1.030463602	0
0.04	1.040832323	1.040832323	0
0.05	1.051313289	1.05131329	−0.000000001
0.06	1.06190964	1.061909641	−0.000000001
0.07	1.072624544	1.072624545	−0.000000001
0.08	1.083461203	1.083461204	−0.000000001
0.09	1.094422851	1.094422845	0.000000004
0.1	1.105512754	1.10551275	0.000000004

The values of y ; from $x = 0.001$ to $x = 0.01$ are described in the following Table 4.

Table 4. The values of y ; from $x = 0.001$ to $x = 0.01$

	y		
x	$h = 0.001$ Exact solution	$h = 0.001$ Runge -Kutta solution	Error
0.001	1.001000501	1.001000501	0
0.002	1.002002005	1.002002005	0
0.003	1.003004515	1.003004515	0
0.004	1.004008032	1.004008032	0
0.005	1.005012563	1.005012565	−0.000000002
0.006	1.006018108	1.006018111	−0.000000003
0.007	1.007024672	1.007024675	−0.000000003
0.008	1.008032257	1.00803226	−0.000000003
0.009	1.009040865	1.009040869	−0.000000004
0.01	1.010050501	1.010050505	0.000000004

The approximate values and the exact values of y from $x = 0$ to $x = 0.1$ are as shown in the following Figures.

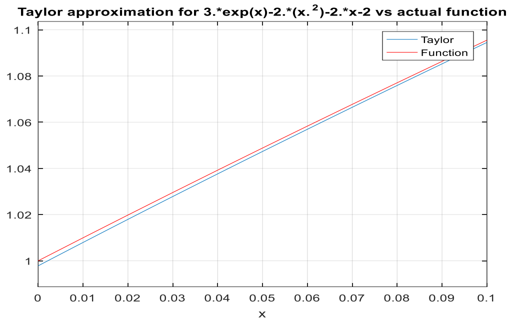

Figure 1. The approximate values and the exact values of y ; from $x = 0$ to $x = 0.1$

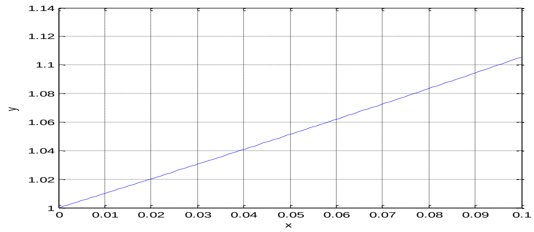

Figure 2. The exact values of y ; from $x = 0$ to $x = 0.1$

The approximate values and the exact values of y from $x = 0.001$ to $x = 0.01$ are as shown in the following Figures.

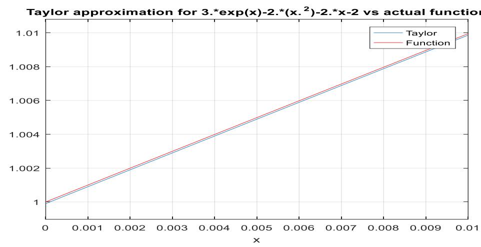

Figure 3. The approximate values and the exact values of y; from $x = 0$ to $x = 0.01$

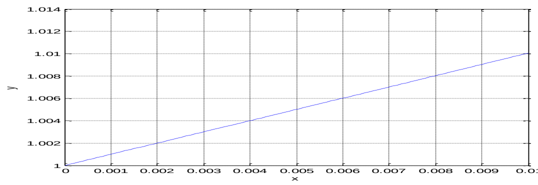

Figure 4. The exact values of y; from $x = 0$ to $x = 0.01$

The approximations from $h = 0.01$ and $h = 0.001$ are computed and drawn the Figures by using Matlab. The error at $h = 0.001$ is smaller than the error at $h = 0.01$ .Therefore the solutions of $h = 0.001$ are better approximations.

3. Conclusion

Two conditions of two examples are computed by third order Taylor's algorithm and fourth order Runge-Kutta method. The values of y are computed by third order Taylor's series method at $h = 0.001$ and $h = 0.01$ in the first example. The values of y are also computed by fourth order Runge-Kutta method at $h = 0.001$ and $h = 0.01$ in the last example. Numerical solutions of ordinary differential equation by fourth order Runge-Kutta method at $h = 0.01$ and at $h = 0.001$ are better approximations than numerical solutions of ordinary differential equation by third order Taylor's series method at $h = 0.01$ and at $h = 0.001$. Fourth order Runge-Kutta method is better than third order Taylor's series method. We observe that the more the exact solution is near, the less the value of h. If the algorithm which is fourth order and greater than fourth order is used in Taylor's series method, the better solutions can be obtained rather than fourth order Runge-Kutta method.

Acknowledgments

We would like to greateful to the committee of UJCIR and the editorial board for permitting the submission of this paper. This research, appreciation is also due to Dr. Moe Moe Hlaing, Professor and Head of department of faculty of computing, University of Computer Studies (Monywa) who has given us various guidelines and knowledges. We would like to deepest thanks to our families for their kind help and encouragement.

References

[1] Aung Myint (M.E.Australia) , *Matlab & Simulink.,ch-1.*

[2] E.C.Zachmanoglou and W. Dale Thoe, An Introduction to Partial Differential Equations with applications, Dover Publications,Inc.,New York, 1986, pp.96-100.

[3] Endre Suli, Numerical Solution of Ordinary Differential Equations, 2014, pp.13-19.

[4] Hans Petter Langtangen.Svein Linge, Finite Difference computing with Partial Differential Equations, Texts in Computational Science and Engineering, ISSN:1611-0994,2017.

[5] J.C. Butcher, Numerical Methods for Ordinary Differential Equations, John Wiley & Sons, England, 2008, pp.93-104, pp.114-118.

[6] Kendall Atkinson, Weimin Han and David Stewart, Numerical Solutions of Ordinary Differential Equations, John Wiley & Sons, Inc,Hoboken, New Jersey, 2009, pp.67-80.

[7] K. Sankara Rao, Numerical Methods for Scientists and Engineers, PHI Learning Private Ltd, New Delhi-110001, 2006, pp.175-179, pp.183-190.

[8] N.P. Bali, Manish Goyal," *A Textbook of Engineering Mathematics"*, Laxmi Publications LTD 113, Golden House, Daryagani, New Delhi 110002, 2007, pp.1396-1404, pp.1415-1424.

[9] Y. Won Yang, Wenwu Cao, Tae-Sang Chung, and John Morris, *Applied Numerical Method Using MATLAB,* John Wiley & Sons, Inc, New York, 2005.