

# Comparison of Several Numerical Algorithms for Solving Ordinary Differential Equation Initial Value Problem

Zhang Lijuan<sup>1,a,\*</sup>, Guan Tianye<sup>2,b</sup>

<sup>1</sup>College of Mathematics and Statistical, Baicheng Normal University, Baicheng, 137000, China

<sup>2</sup>Jilin Province Water Conservancy and Hydropower Survey Design Institute, Changchun, 130021, China

<sup>a</sup>Email:zhanglj4221@126.com\*, <sup>b</sup>Email:Gye09@163.com

**Keywords:** ordinary differential equation, initial value problem, numerical calculation method

**Abstract:** In this paper, the authors using the MATLAB software to the ordinary differential equation initial value problem, Euler's method, improved Euler method, classical Runge-Kutta method is used for computer programming and implementation, through the comparison of numerical results analysis, Euler method principle simple error is bigger, improved Euler method to calculate the amount is larger, the classical Runge-Kutta method for high precision stability is also very good.

## 1. Introduction

With the rapid development of computer technology, numerical calculation methods have been infiltrated into scientific research and computational interdisciplinary is emerging, we often encountered ordinary differential equation initial value problem to solve in engineering calculation [1] [2] [3], but only a small part of the initial value problems could be solved using elementary solution and the analytical solution. For the most part of them could only be solved by approximate methods [4] [5][6] [7] In the most case we can only use approximate method to get numerical solution. Numerical solution of differential equation is a kind of discrete method. We compare to Euler method, improved Euler method, classical Runge-Kutta method solving initial value of ordinary differential equations by MATLAB.

A general form of the type of problem we consider is

$$\begin{cases} \frac{dy}{dx} = f(x, y), x \in [a, b] \\ y(a) = y_0 \end{cases} \quad (1)$$

First the interval  $[a, b]$  is divided into  $n$  equal parts,  $x_i = a + ih, (i = 0, 1, 2, \dots, n)$ , the step is  $h = x_{i+1} - x_i$ . Then to solve the function  $y(x)$  in a series of discrete equidistant node  $x_0 < x_1 < x_2 < x_3 < \dots < x_n$  to get approximate values  $y_0 < y_1 < y_2 < y_3 < \dots < y_n$ .

## 2. Basic Theory

### 2.1 Euler Method.

Euler's method is the most fundamental and most simple algorithm. According to Taylor's theorem  $y(x)$  in some show,  $x_n$ , we can obtain:

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2!} y''(\xi_n), \xi_n \in (x_n, x_{n+1}) \quad (2)$$

$$R_n = \frac{h^2}{2!} y''(\xi_n), \xi_n \in (x_n, x_{n+1})$$

When  $h$  is very small, the error term can be omitted.

The following is approximate formula of differential equation with precise solution.

$$y(x_{n+1}) \approx y(x_n) + hy'(x_n), y'(x_n) = f(x_n, y_n) \quad (3)$$

$$y(x_{n+1}) \approx y(x_n) + hf(x_n, y_n) \quad (4)$$

Where  $y_i$  is the approximate value of  $y(x_i)$ ,  $(i = 1, 2, \dots, n)$ , then  $y_{n+1} = y_n + hf(x_n, y_n)$ .  
Euler formula is:

$$y_{n+1} = y_n + hf(x_n, y_n), x \in [a, b], y(a) = y_0 \quad (5)$$

Euler's method calculated  $y_{n+1}$  by  $y_n$ . It is a single step. The local truncation error is  $O(h^2)$  and truncation error is  $O(h)$ . It is the first order convergence method.

## 2.2 Improved Euler Method.

In order to get a more accurate method, it uses trapezoidal integral formula instead of integral type  $\int_{x_n}^{x_{n+1}} f(t, y(t)) dt$  then

$$\int_{x_n}^{x_{n+1}} f(t, y(t)) dt \approx \frac{h}{2} \{f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))\} \quad (6)$$

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(t, y(t)) dt$$

$$y(x_{n+1}) \approx y(x_n) + \frac{h}{2} \{f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))\} \quad (7)$$

Let  $y_n$  to  $y(x_n)$  and  $y_{n+1}$  to  $y(x_{n+1})$  Where

$$y_{n+1} = y_n + \frac{h}{2} \{f(x_n, y_n) + f(x_{n+1}, y_{n+1})\} \quad (8)$$

Trapezoid formula is a second order convergence method, but it is an implicit algorithm, so it is improved. It gets a predictor  $y_{n+1} = y_n + hf(x_n, y_n)$  by Euler method, then it gets an approximate

value by trapezoidal correction formula approximation.  $y_{n+1} = y_n + \frac{h}{2} \{f(x_n, y_n) + f(x_{n+1}, y_{n+1})\}$

Predictor  $\tilde{y}_{n+1} = y_n + hf(x_n, y_n)$

Corrector  $y_{n+1} = y_n + \frac{h}{2} \{f(x_n, y_n) + f(x_{n+1}, \tilde{y}_{n+1})\}$

Improved Euler formula is:

$$\begin{cases} \tilde{y}_{n+1} = y_n + hf(x_n, y_n), \\ y_{n+1} = y_n + \frac{h}{2} \{f(x_n, y_n) + f(x_{n+1}, \tilde{y}_{n+1})\} \end{cases} \quad (9)$$

Local truncation error is  $O(h^3)$ . The improved Euler formula is a second order convergence method.

## 2.3 Classic Runge-Kutta Method.

Runge-Kutta method reduces data requirements to reach the same precision without higher

derivative calculation.

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(t, y(t)) dt$$

According to the integral mean value theory:

$$\int_{x_n}^{x_{n+1}} f(t, y(t)) dt = f(x_n + \theta h, y(x_n + \theta h)), 0 \leq \theta \leq 1 \quad (10)$$

The mean value  $f(x_n + \theta h, y(x_n + \theta h))$  can be expressed by a linear combination of function values

$$y_{n+1} = y_n + h \sum_{i=1}^s c_i f(x_i, y_i). \quad (11)$$

We get  $S$  order Runge-Kutta scheme. According to the different requirements of precision, we calculate the undetermined coefficients. So we can get different Runge-Kutta formula. Classic fourth-order Runge Kutta formula is widely used in the engineering. It is a kind of absolute stability of the algorithm. [8][9]The local truncation error is  $O(h^5)$ .

Classic fourth-order Runge-Kutta formula is:

$$\left\{ \begin{array}{l} y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \\ k_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2) \\ k_4 = f(x_n + h, y_n + hk_3) \end{array} \right. \quad (12)$$

### 3. Implementation by Computer

$$\left\{ \begin{array}{l} \frac{dy}{dx} = y - \frac{2x}{y}, x \in [0, 1.4], h = 0.2 \\ y(0) = 1 \end{array} \right.$$

The exact solution of equation is  $y = \sqrt{2x+1}$  It performs in MATAB command window:

```
dyfun=inline('y-2x/y','x','y');
[x,y]=ouler(dyfun,xspan,y0,h);
[x,y]=gaijinouler(dyfun,x0,y0,h,n);
[x,y]=Rungekutta(dyfun,x0,y0,h,n);
```

Table 1 Euler's method, the improved Euler method and Runge-Kutta method numerical solution

$x_i$	$y(x_i)$	$y_i Euler$	$y_i Im proveuler$	$y_i Rungekutta$
0	1.00000	1.00000	1.000000	1.000000
0.2	1.183216	1.200000	1.186667	1.183229
0.4	1.341641	1.373333	1.348312	1.341667
0.6	1.483240	1.531495	1.493704	1.483281
0.8	1.612452	1.681085	1.627861	1.612514
1.0	1.732051	1.826948	1.754205	1.732142

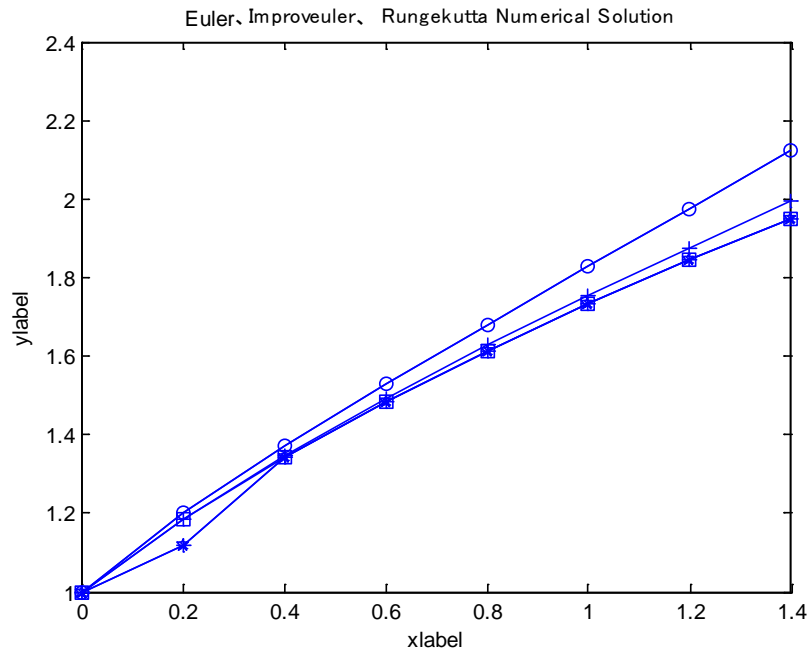


Fig.1 Curve of numerical algorithms method

#### 4. Conclusion

From the comparison we can draw conclusion that improved Euler method's numerical precision is higher than Euler, and the fourth-order Runge-Kutta method is obviously better than Euler method and the improved Euler method. From the computing workload, classic fourth-order Runge-Kutta method of calculation is twice as large as improved Euler method, and four times as great as Euler method. From the standpoint error perspective, Euler method has a big error, improved Euler method's error is reduced to a certain extent. Classical fourth-order Runge-Kutt sharply reduces its error and it has high precision. So in the actual calculation, according to different engineering function, we should choose the appropriate algorithm. We not only consider method's simplicity, and but also reduce the amount of calculation. At the same time, we must insure the error in specified range. Generally in the study of ordinary differential equation initial value problem, the classical fourth-order Runge-Kutta method has high precision and is conducive to computer programming implementation. It also has good stability. It can be considered as the preferred the implementation of the algorithm.

#### Acknowledgement

In this paper, the research was sponsored by Fund Project of Education Science in Jilin Province (Project No. GH13366).

#### References

- [1] Zhu Jianxin, Li Youfa .Numerical methods the third edition [M]. Beijing: Higher Education Press. 2012(7)67-68.
- [2] Guan Ye. The numerical analysis [M]. Beijing: Higher Education Press. 1997.5:122-223.
- [3] Li Rong-hua, Liu Bo. The numerical method for solving the differential equation the third edition [M]. Beijing: Higher Education Press.2009. (3)188-189.
- [4] Zhang De-feng. Numerical analysis and application [M]. Beijing: National Defence Industry press, 2007(1)102-103.
- [5] Practice Teaching Reform and Thinking about the "Numerical Methods for Differential

Equations" Journal of Lanzhou University of Arts and Science (Natural Sciences) 2014.(6)96-97.

[6] ZHANG Shu-juan. The Arithmetic and Solution of Initial Value Problems of First-order Ordinary Differential Equation [J]. Journal of Jingmen Technical. 2007(3)52-53.

[7] WEI Ming-qiang. The Comparison of Four Methods on the First Order Ordinary Differential Equation Numerical Solution.[J] Journal of Communication University of China Science and Technology 2016,23(2).154-156.

[8] ZHANG Qiu-sheng. Solution of Initial Value in Ordinary Differential Equation via Single-step Method [J]. Bulletin of Science and Technology 2012, 28(2)232-235.

[9] Huo Xiao-cheng. Study on the numerical method of ordinary differential equationsjournal of Juamjusi [J]. Education Institute, 2011(5)40-48.