



Order Runge-Kutta with Extended Formulation for Solving Ordinary Differential Equations

Wahyu Suryaningrat^{1,*}, Rizky Ashgi², Sri Purwani³

^{1,2,3}*Department of Mathematics, Universitas Padjadjaran, Sumedang, Indonesia*

**Corresponding author email: wahyusuryaningrat@gmail.com*

Abstract

The mathematical model has been used to understand many phenomena and natural interactions. Since including many variables and parameters, the complex models are not easy to find analytical solutions. In this paper, we analyze one of the family of Runge–Kutta method with an expansion of evaluation function. We applied the proposed method to solve ordinary differential equations problems and compared it with other well-known Runge-Kutta methods. The computation cost and accuracy for each method have been analyzed.

Keywords: Accuracy, computation cost, extended Runge-Kutta, ODE.

1. Introduction

Differential equations are the mathematical tool that can be used to model problems in various fields of sciences such as economics, biology, physics, mathematics, robotic, encryption, circuit (Amelia, 2020; Anggriani 2019; Aldila 2020; Ndii, 2020; Vaidyanathan et al., 2017; Sambas et al., 2020; Sukono et al., 2020; Sambas et al., 2019; Mobayen et al., 2019;). A differential equation is an equation that contains a derivative of a function. Based on the number of independent variables, the differential equations are divided into two, namely ordinary differential equations and partial differential equations. The combination of several differential equations is called a system of differential equations (Atkinson, 2008).

Numerical methods are also capable of solving a large, non-linear, and very complex system of differential equations with known initial conditions. A search using numerical methods produces an approximate value of an analytical solution so that the solution contains an error value. Numerical methods can be used to solve differential equation problems with the aid of a computer as a calculation tool. One of the numerical methods used to obtain the exact value of differential equation problems is Runge Kutta.

Mathematical models are designed with varying difficulties, relying on the characteristics of the problem. Some models require high computational cost for simulation, especially for a complex model. Many numerical methods have been developed by researchers for solving nonlinear differential equations. Numerical methods for solving first-order IVPs often fall into one of two large categories: linear multistep methods, or Runge–Kutta methods (Griffiths, 2010). A further division can be realized by dividing methods into those that are explicit and those that are implicit.

Many researchers analyzed various Runge-Kutta methods and proposed some modifications. Goeken (2000) developed a class of Runge-Kutta method with higher derivatives approximations for the 3rd and 4th-order method. Wu (2003) proposed a new family of Runge-Kutta formula with reduced evaluations of function. Phohomsiri and Udawadia (2004) proposed the Accelerated Runge-Kutta using two functions evaluation per step. Gadisa (2017) have compared the convergence and stability of the Runge-Kutta method with the high order Taylor method. Other researchers have implemented Runge-Kutta 6th order method to solve the initial value problem and partial differential equations model (Al-Shimmmary, 2017; Sun 2017). The implicit form of the Runge-Kutta 6th order function and its application to ordinary differential equations problem has been investigated by Ghawadri (2019) and Huang (2019). In this study, we discuss a new family of Runge-Kutta method, which includes the derivation of its formulation and application in computer programming, as well as obtaining the results of solutions given by each Runge Kutta formulation. Apart from this, each method is then compared in terms of error rate, speed, and computational efficiency. We apply the proposed Runge-Kutta formulation for solving non-linear differential equations problem. We also compared the proposed method with other Runge-Kutta family.

2. Methodology

2.1. Explicit Runge-Kutta methods

The Runge–Kutta method can lead to a large family of methods that have close structure. The following are some of the basic properties used in constructing the formula for the Runge-Kutta Method.

Definition 1.

The generalization of the Runge-Kutta method is given by

$$y_{n+1} = y_n + h \sum_{i=1}^s b_i k_i, \quad (1)$$

where

$$k_i = f \left(y_n + c_i h, y_n + h \sum_{j=1}^{i-1} a_{ij} k_j \right), i = 1, 2, \dots, s,$$

with assumption that

$$c_i = \sum_{j=1}^s a_{ij}, \text{ and } \sum_{i=1}^s b_i = 1$$

To derived it explicitly, we needs to find the integer s , and the coefficients a_{ij} (for $1 \leq j < i \leq s$), b_i (for $i = 1, 2, \dots, s$) and c_i (for $i = 2, 3, \dots, s$).

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
\vdots	\vdots		\ddots		
c_s	a_{s1}	a_{s2}	\dots	$a_{s,s-1}$	
	b_1	b_2		b_{s-1}	b_s

Using Taylor series expansion, it can be derived that the Runge–Kutta formulation is consistent if and only if

$$\sum_{j=1}^s b_j = 1. \quad (2)$$

2.2. Extended Runge–Kutta Method

Previous research has been done to improve the accuracy of the Runge-Kutta method. Among others are improving the number of terms in Taylor series expansion. One of the approaches is done by Goeken (2008), with adding a term to the evaluation function using higher derivatives and presented new third and fourth-order numerical methods. Technically, the term f' is added in f . This leads to a new family of Runge–Kutta formulation as given in definition 2.

Definition 2.

The generalization of the Extended Runge-Kutta method is stated by

$$y_{n+1} = y_n + h \sum_{i=1}^s b_i k_i + h^2 \sum_{i=1}^s c_i l_i, \quad (3)$$

where

$$k_i = f \left(y_n + h \sum_{j=1}^{i-1} a_{is} k_j \right), l_i = f' \left(y_n + h \sum_{j=1}^{i-1} a_{is} l_j \right), i = 1, 2, \dots, s.$$

As in Runge-Kutta, equation (3) represents the main function of Runge-Kutta while k_i and l_i are evaluation function. In this paper, we use the sixth order Extended Runge-Kutta and Butcher table to determine the coefficient value of the formulation.

3.1. Formulation of Novel Runge-Kutta Methods

$$y_{n+1} = y_n + h \sum_{i=1}^6 b_i k_i + h^2 \sum_{i=1}^6 c_i l_i, \quad (4)$$
$$k_i = f(y_n + h \sum_{i=1}^{i-1} a_{iS} k_j), l_i = f'(y_n + h \sum_{i=1}^{i-1} a_{iS} l_j), i = 1, 2, \dots, 6.$$

Table 1. Table Butcher for Extended 6th Order Runge-Kutta

Ci									
0									
9	9								
50	50								
1	29	25							
6	324	324							
1	1		3						
4	16	0	16						
53	79 129		261 237	19 663					
		0	-						
100	250 000		250 000	15 625					
3	1 336 883	0	-	25 476	194 159	8225			
5	4 909 125		30 875	185 250	78 546				
4	2 459 386	0	-	19 504	2 377 474	6 157 250	902		
5	14 727 375		30 875	13 615 875	5 773 131	735			
1	2699	0	-	252	1 393 253	236 875	135	15	
	7410		1235	3993 990	72 618	49	22		
1	11	0	0	256	0	125	125	5	
	144			693		504	528	72	
b	11	0	0	256	0	125	125	5	0
	144			693		504	528	72	

3.2 Numerical Simulation

In this chapter, we will show the application of the extended 6th order Runge-Kutta method for solving examples of dynamic models. We use a simple logistic growth model and Lotka-Volterra equation system on a simple interaction of two populations (prey predatory model). Using the formulation that has been derived, we compute the solution of both models by using the Maple 15 program. The parameters used are hypothetical, for simulation purposes.

3.2.1 Application for solving Logistic Growth Model

Our first example concerns the simple scalar ODE problem

$$\frac{dN}{dt} = rN \left(1 - \frac{N}{K}\right) \quad (5)$$

where $N_0, r, K > 0$ are known constants. This is a common model for population dynamics in ecology where N represents the number of individuals, r the initial growth rate, K is the maximum number of individuals that is allowed by the environment (the so-called *carrying capacity* of the environment).

Table 2. Parameter's description and value for Model 1

Parameter	Description	Value
t	Time	[0, 25]
r	Growth rate	0.5
K	Carrying Capacity	25
$N(0)$	Initial value	10

Using the separable differential equations method with the parameter value given in Table 2 we can simply get the exact solution for Model 1 as follows

$$N(t) = \frac{50}{2 + 3 \exp\left(-\frac{1}{2}\right)} \quad (6)$$

The graph of the solution (6) is displayed in Figure 1. Along with applying the derived extended 6th order Runge-Kutta Method we compared the result of the solution with those from other well-known Runge-Kutta family, that is Runge-Kutta Fehlberg and Classical 6th order Runge-Kutta. In Figure 1 it is shown that the solution of each method approaches well to the analytical solution. We use 25 numerical points for solving Model 5.

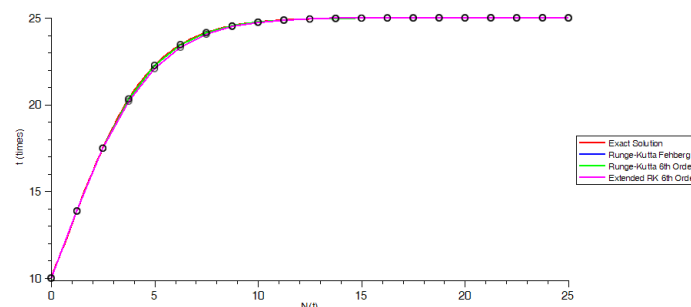


Figure 1. Comparison of each method for solving Model 5

The error of Runge-Kutta Fehlberg (RKF), 6th order Runge-Kutta (RK 6), and Extended 6th order Runge-Kutta (XRK6) for some time steps in numerical point is shown in Table 3. We use Root Mean Square Error (RMSE) to summarize the performance result for a method, which follows

$$RMSE = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{n}} \quad (7)$$

Table 4 shows the RMSE for each method when used to solve Model 1. XRK6 method has a smaller error than RKF and RK 6. However, in terms of computing cost, the XRK6 requires the largest amount of memory and the longest time to complete a simulation.

Table 3. Computational result for Model 1

t	Analytical Solution	Error RKF	Error RK 6	Error Extended RK6
0	10			
5	22.26742671	0.00815513	0.00230066	0.00686553
10	24.74985518	0.00460688	0.00243338	0.00167942
15	24.97927653	0.00077001	0.00035643	0.00003863
20	24.99829762	0.00009494	0.00009024	0.00003296
25	24.99986025	0.00001035	0.00001238	0.00000532

Table 4. Comparison of computational cost and results of Runge-Kutta Fehlberg, and Proposed Runge-Kutta for solving Model 1

Method	RMSE	Memory Used (kilobytes)	Time Cost (millisecond)
Runge-Kutta Fehlberg	$4.020308338 \times 10^{-3}$	49.36	1000
Runge-Kutta 6th order	$3.200666812 \times 10^{-3}$	66.26	3000
Extended RK 6	$1.983032538 \times 10^{-3}$	116.49	13000

3.2.2. Application for solving Lotka-Volterra Model

One of the simpler population models that describe prey interactions is the Lotka-Volterra, named after research by Alfred J. Lotka (1910) and Vito Vorterra (1926). Model consists of a pair of non-linear differential equations, which follows

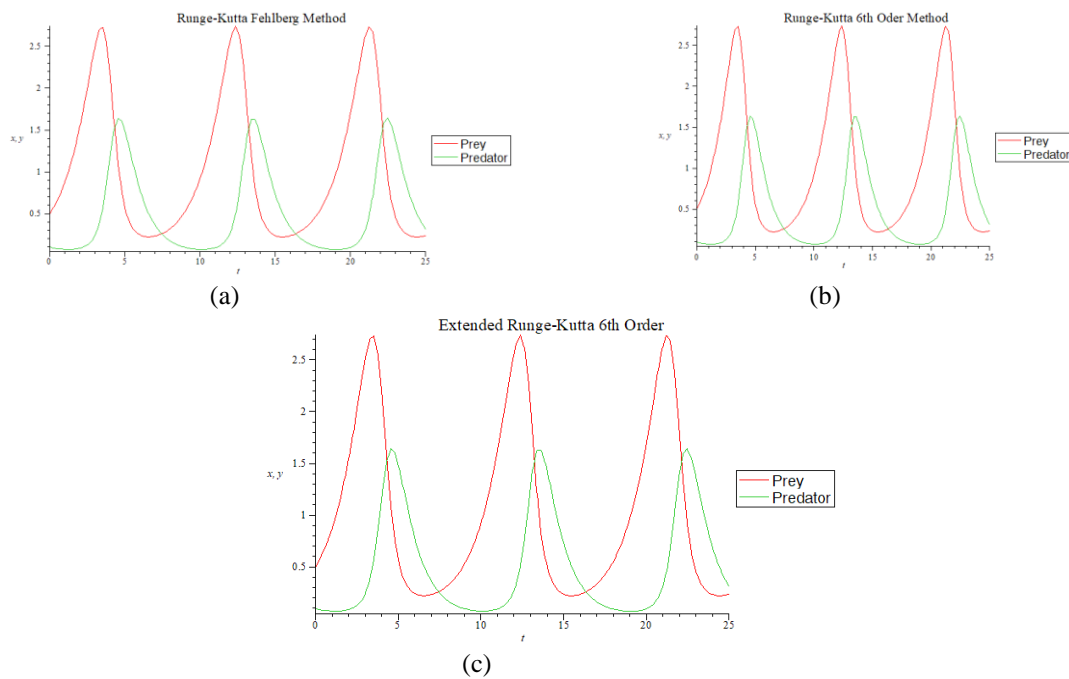
$$\begin{aligned}\frac{dx}{dt} &= (\alpha - \beta y)x \\ \frac{dy}{dt} &= (\delta x - \gamma)y.\end{aligned}\tag{8}$$

Parameter α described the natural growth rate of species x (prey). The number of prey is diminished by predation βyx . We used hypothetical parameters for simulation shown in Table 5.

Table 5. Parameter's description and value for Model 1

Parameter	Description	Value
t	Time	[0, 25]
α	Growth rate of prey	2/3
β	Predation rate of prey	4/3
δ	Predation rate of predator	1
γ	Natural death rate of predator	1
$x(t), y(t)$	Initial values	0.5, 0.1

With the same procedure as in the previous simulation, we compare the solutions of the Lotka-Volterra model given by each method. Figure 3 shows the dynamic populations of prey and predator produced by each method. The peak points and oscillations of each population can also be captured by method.

**Figure 3.** Plot dynamic population solution for Model 2**Table 6.** Computational result for Model 2

t	Solution of $x(t)$ by XRK6	$ XRK6 - RKF $	$ XRK6 - RK6 $
0	0.5	0	0
5	0.559641	1.04×10^{-6}	3.30×10^{-7}
10	0.918807	1.45×10^{-6}	2.37×10^{-8}
15	0.236079	1.42×10^{-6}	6.22×10^{-7}
20	1.699455	8.54×10^{-7}	-2.14×10^{-7}
25	0.239242	1.36×10^{-6}	4.05×10^{-7}

Table 7. Comparison of computational results of Runge-Kutta Fehlberg, and Extended for solving Model 2

Method	Memory Used (kilobytes)	Time Cost (milliseconds)
Runge-Kutta Fehlberg	70.02	2000
Runge-Kutta 6th order	75.53	1000
Extended RK6	310.18	5000

Table 6 shows the solution produced by Extended able to approach the solution produced by the other two methods. Agreed with the previous simulation, the computation cost required for this method is the most for solving the Model as shown in Table 7.

4. Conclusion

The family Runge-Kutta method for solving differential equation system has been analyzed. We derived basic properties of Runge-Kutta and proposed the new family of Runge-Kutta with sixth-order and extended formulation. We use the proposed method to solving the simple dynamic models. By comparing the dynamical population of the model and the computational cost which is obtained by each method, the proposed Runge-Kutta method quite close to the analytical solution. Further the results show the proposed method can be an alternative way for solving ODE problems.

References

- Al-Shimmary, A. F. A. (2017). Solving Initial Value Problem Using Runge-Kutta 6th Order Method. *ARP Journal of Engineering and Applied Sciences*, 12, 3953-3961.
- Aldila, D., Ndi, M. Z., & Samiadji, B. M. (2020). Optimal control on COVID-19 eradication program in Indonesia under the effect of community awareness. *Math. Biosci. Eng*, 17, 6355-6389.
- Amelia, R., Anggriani, N., Istifadah, N., & Supriatna, A. K. (2020, September). Dynamic analysis of mathematical model of the spread of yellow virus in red chili plants through insect vectors with logistical functions. In *AIP Conference Proceedings* (Vol. 2264, No. 1, p. 040006). AIP Publishing LLC.
- Anggriani, N., Tasman, H., Ndi, M. Z., Supriatna, A. K., Soewono, E., & Siregar, E. (2019). The effect of reinfection with the same serotype on dengue transmission dynamics. *Applied Mathematics and Computation*, 349, 62-80.
- Atkinson, K. E. (2008). An introduction to numerical analysis. New York: John Wiley & sons.
- Ndi, M. Z., & Adi, Y. A. (2020). Modelling the transmission dynamics of COVID-19 under limited resources. *Commun. Math. Biol. Neurosci.*, 2020, 1-24.
- Gadisa, G., & Garoma, H. (2017). Comparison of higher order Taylor's method and Runge-Kutta methods for solving first order ordinary differential equations. *Journal of Computer and Mathematical Sciences*, 8(1), 12-23.
- Ghawadri, N., Senu, N., Adel Fawzi, F., Ismail, F., & Ibrahim, Z. B. (2019). Diagonally implicit Runge-Kutta type method for directly solving special fourth-order ordinary differential equations with Ill-Posed problem of a beam on elastic foundation. *Algorithms*, 12(1), art. id. 10.
- Goeken, D., & Johnson, O. (2000). Runge-Kutta with higher order derivative approximations. *Applied numerical mathematics*, 34(2-3), 207-218.
- Griffiths, D. F., & Higham, D. J. (2010). Numerical methods for ordinary differential equations: initial value problems. Berlin: Springer Science & Business Media.
- Huang, D. Z., Persson, P. O., & Zahr, M. J. (2019). High-order, linearly stable, partitioned solvers for general multiphysics problems based on implicit-explicit Runge-Kutta schemes. *Computer Methods in Applied Mechanics and Engineering*, 346, 674-706.
- Mobayen, S., Vaidyanathan, S., Sambas, A., Kacar, S., & Çavuşoğlu, Ü. (2019). A novel chaotic system with boomerang-shaped equilibrium, its circuit implementation and application to sound encryption. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 43(1), 1-12.
- Phohomsiri, P., & Udawadia, F. E. (2004). Acceleration of Runge-Kutta integration schemes. *Discrete Dynamics in Nature and Society*, 2004(2), 307-314.
- Rabiei, F., & Ismail, F. (2012). Fifth-order Improved Runge-Kutta method for solving ordinary differential equation. *Australian Journal of Basic and Applied Sciences*, 6(3), 97-105.
- Sambas, A., Vaidyanathan, S., Zhang, S., Zeng, Y., Mohamed, M. A., & Mamat, M. (2019). A new double-wing chaotic system with coexisting attractors and line equilibrium: bifurcation analysis and electronic circuit simulation. *IEEE Access*, 7, 115454-115462.
- Sambas, A., Vaidyanathan, S., Tlelo-Cuautle, E., Abd-El-Atty, B., Abd El-Latif, A. A., Guillén-Fernández, O., & Gundara, G. (2020). A 3-D multi-stable system with a peanut-shaped equilibrium curve: Circuit design, FPGA realization, and an application to image encryption. *IEEE Access*, 8, 137116-137132.
- Sukono, Sambas, A., He, S., Liu, H., Vaidyanathan, S., Hidayat, Y., & Saputra, J. (2020). Dynamical analysis and adaptive fuzzy control for the fractional-order financial risk chaotic system. *Advances in Difference Equations*, 2020(1), 1-12.
- Sun, Z., & Shu, C. W. (2017). Stability of the fourth order Runge-Kutta method for time-dependent partial differential equations. *Annals of Mathematical Sciences and Applications*, 2(2), 255-284.
- Vaidyanathan, S., Sambas, A., Mamat, M., & Sanjaya, M. (2017). A new three-dimensional chaotic system with a hidden attractor, circuit design and application in wireless mobile robot. *Archives of Control Sciences*, 27(4), 541-554.

- Wu, X. (2003). A class of Runge–Kutta formulae of order three and four with reduced evaluations of function. *Applied Mathematics and Computation*, 146(2-3), 417-432.