Science and Technology Indonesia
e-ISSN:2580-4391 *p*-ISSN:2580-4405
Vol. 10, No. 3, July 2025

Research Paper

An Improved Fifth-Order Runge-Kutta Method with Higher Accuracy and Efficiency for Solving Initial Value Problems

Ummu Habibah[1*], Fermin Franco Medrano[2], Adith Chandra Permana[1], Dita Ardiana[1], Trisilowati[1]

[1]*Department of Mathematics, Brawijaya University, East Java, 65145, Indonesia*
[2]*Department of Mathematics, Faculty of Science, Autonomous University of Baja California, 21100, Mexico*
***Corresponding author**: ummu_habibah@ub.ac.id

Abstract
Solving initial value problems (IVPs) in ordinary differential equations (ODEs) often requires numerical methods, with the fifth-order Runge-Kutta method being a widely used approach due to its balance between accuracy and computational efficiency. A novel and straightforward formula for the fifth-order Runge-Kutta method is proposed, aiming to simplify calculations while maintaining high accuracy and stability. The method is derived using an optimized Taylor series expansion, leading to a more efficient formulation. Numerical experiments are conducted to compare the proposed method with existing fifth-order Runge-Kutta methods. The results show that the proposed formula outperforms existing methods in terms of accuracy, stability, and computational efficiency. This new formula provides a practical alternative for solving IVPs in ODEs with improved performance.

Keywords
Fifth-Order Runge-Kutta Method, Taylor Series, Numerical Solution, Accuracy, Stability, Computational Efficiency

Received: 14 November 2024, Accepted: 4 May 2025
https://doi.org/10.26554/sti.2025.10.3.802-816

1. INTRODUCTION

Ordinary differential equations (ODEs) play a fundamental role in modeling a wide range of real-world phenomena across various scientific and engineering disciplines (Dennis, 2012). They are used to describe dynamic processes in physics, chemistry, biology, economics, and engineering, among others. For example, in physics, ODEs are utilized to model motion, wave propagation, and thermodynamic systems (Kalogiratou et al., 2014). In chemistry, reaction kinetics are often expressed in terms of ODEs to predict the concentration of reactants and products over time (Kopecz and Meister, 2018). Similarly, in epidemiology, ODEs are widely applied to simulate disease transmission and the effectiveness of interventions (Mechee et al., 2013).

Despite their broad applicability, solving ODEs analytically is often infeasible, especially for nonlinear or complex systems. Analytical methods are limited to specific cases, such as linear differential equations with constant coefficients, where closed-form solutions exist (Kalogiratou et al., 2014). However, many real-world systems involve nonlinear interactions, making it difficult or impossible to derive exact solutions. As a result, numerical methods have become essential tools for solving ODEs (Kennedy and Carpenter, 2019; Suryaningrat et al., 2020).

Several numerical techniques have been developed to approximate solutions of ODEs, including Euler's method, Taylor series methods, and multistep methods (Mathews and Fink, 2004). Among these, the Runge-Kutta (RK) methods are widely used due to their balance between accuracy and computational efficiency (Mechee et al., 2013). The RK methods provide higher accuracy than the basic Euler method while avoiding the need for higher-order derivatives required in Taylor series methods (Monovasilis et al., 2016).

The Runge-Kutta methods exist in various orders, ranging from first-order (RK1) to higher-order versions such as RK4 and RK5. The classical fourth-order Runge-Kutta (RK4) method is the most commonly used due to its favorable trade-off between accuracy and computational cost (Kennedy and Carpenter, 2019). Sukron et al. (2021) solve the Saint-Venant Equation (SVE) by transforming it into a semi-discrete equation by using a finite difference in space (x) such that the SVE becomes ordinary differential equations (ODEs). Furthermore, the semi-discrete form of SVE solved using fourth-order Runge-Kutta method. However, in many applications requiring higher precision, such as orbital mechanics, fluid dynamics, and chaotic systems, fifth-order Runge-Kutta (RK5) methods are preferred (Ram, 2020). Several RK5 methods have been proposed in the literature, including modified Patankar - Runge

- Kutta schemes (Kopecz and Meister, 2018), explicit and implicit RK formulations (Kennedy and Carpenter, 2019), and symplectic Runge-Kutta approaches (Monovasilis et al., 2016). These methods offer improved accuracy over RK4, but they come with additional computational complexity. Moreover, certain RK5 variants require multiple function evaluations per step, which increases computational cost (Agbeboh et al., 2020; Chandru et al., 2017; Dennis, 2012; Fardinah, 2017; Goeken and Johnson, 1999; Kalogiratou et al., 2014; Rabiei and Ismail, 2012)

In recent decades, the development of RK5 methods has continued to enhance computational efficiency and numerical stability. Recent studies have shown that the Runge-Kutta method can be optimized through symplectic approaches and implicit-explicit (IMEX) methods, which significantly improve stability in stiff differential systems (Bosede et al., 2012). Additionally, exponential and trigonometric-fitting approaches have been developed within the Runge-Kutta framework to enhance accuracy in oscillatory problems (Demba et al., 2016). Recent research has also focused on the application of Runge-Kutta methods in modern fields, such as the analysis of differential equation-based epidemiological models, where efficient numerical methods are essential for long-term simulations (Gumus et al., 2022; Hossain et al., 2017).

Recent advancements in Runge-Kutta methods have enhanced accuracy and efficiency in solving various differential equations. The numerical method of the line improves stability for singularly perturbed Burgers' equations (Aliyi and Muleta, 2021), while performance assessments highlight variations in RK methods for first-order problems (Audu et al., 2023). High-order numerical schemes optimize mine impact burial simulations (Donas et al., 2021), and modified RK methods enhance the solution of oscillatory problems (Ghazal and Hussain, 2021). Additionally, improved RK formulations offer better efficiency for third-order ODEs, such as thin film flow models (Hussain et al., 2015; Mechee et al., 2013). These developments underscore the need for a more optimized fifth-order RK method, as proposed in this study.

Despite the advantages of RK5 methods, several challenges remain, particularly in terms of computational efficiency, stability, and coefficient complexity (Agbeboh et al., 2020; Chandru et al., 2017; Dennis, 2012; Kalogiratou et al., 2014). Many existing RK5 formulas involve intricate coefficient calculations, which lead to increased computational overhead (Monovasilis et al., 2018). For instance, some methods employ complex implicit formulations that require solving nonlinear algebraic equations at each step, making them computationally expensive (Kennedy and Carpenter, 2019).

Additionally, the stability properties of RK5 methods vary significantly. Some formulations have limited stability regions, restricting their applicability to stiff ODEs (Monovasilis et al., 2016). Other methods, such as the exponentially and trigonometrically fitted RK approaches (Monovasilis et al., 2016), offer improved stability for oscillatory problems but require additional function evaluations per step. This trade-off between stability and computational cost remains a key challenge in the development of efficient RK5 methods (Kalogiratou et al., 2014).

Although numerous RK5 methods exist, many still exhibit limitations in terms of computational efficiency, stability, and coefficient selection. For instance, recent studies have demonstrated that a simpler RK5 method with fewer function evaluations can significantly reduce computational time without compromising accuracy (Hussain and Hasan, 2023). This approach is increasingly relevant in the context of modern engineering applications, including high-order dynamic system simulations and model-based design optimization (Hussain et al., 2015). Furthermore, the integration of Runge-Kutta methods with machine learning-based approaches has also begun to emerge as a potential solution to overcome traditional limitations in adaptive step-size selection (Kafle et al., 2021).

A significant limitation in many existing RK5 methods is the lack of simplicity in coefficient selection, which complicates both implementation and theoretical analysis (Monovasilis et al., 2018). The complexity of these formulas can hinder their adoption in practical applications, especially in fields requiring rapid numerical integration of large-scale ODE systems (Agbeboh et al., 2020). Furthermore, some existing RK5 methods show reduced efficiency when applied to systems with small step sizes, leading to unnecessary computational burden (Chandru et al., 2017).

Given the challenges in existing RK5 methods, this study proposes a novel and simplified fifth-order Runge-Kutta formula that improves computational efficiency while maintaining high accuracy and stability. The proposed method is developed using an optimized Taylor series expansion, leading to a more straightforward formulation with carefully selected coefficients that minimize computational cost. This formulation aims to strike a balance between simplicity, accuracy, and stability, making it an attractive alternative to existing RK5 approaches.

To validate the effectiveness of the proposed method, comprehensive numerical experiments are conducted, comparing its performance against existing RK5 methods (Agbeboh et al., 2020; Chandru et al., 2017). These experiments evaluate key aspects such as accuracy, execution time, and stability properties. The results demonstrate that the new RK5 formula achieves comparable or better accuracy than existing RK5 methods while reducing computational complexity by simplifying coefficient calculations. Additionally, it maintains numerical stability, making it suitable for a wide range of ODE problems. Furthermore, the proposed method performs efficiently in both single ODEs and systems of ODEs, highlighting its versatility in various applications.

Through these improvements, this study contributes to the advancement of numerical methods for solving initial value problems (IVPs) in ODEs. By offering a simpler yet effective RK5 formulation, this work provides a practical alternative that can be readily implemented in scientific computing applications.

2. EXPERIMENTAL SECTION

2.1 The Runge-Kutta Method

The Runge-Kutta (RK) method is a widely used numerical approach for solving initial value problems (IVPs) in ordinary differential equations (ODEs). In general, an IVP can be expressed as follows (Mathews and Fink, 2004)

$$\frac{dy}{dt} = f(t,y),\ t \geq 0, y(t_0) = y_0 \tag{1}$$

To approximate the solution of Equation 1, the RK method discretizes the time interval into subintervals of step size h, producing numerical estimates y_i at discrete points t_i. Unlike the simple Euler method, which uses only the first derivative information at each step, RK methods incorporate multiple evaluations of f(t,y) at different points within each step to enhance accuracy (Chapra and Canale, 2015). The RK method is derived from the Taylor series expansion, ensuring higher-order accuracy without requiring explicit computation of higher-order derivatives. If y_{i+1} is expanded as a Taylor series around t_i, we obtain the result as in Equation 2 (Dukkipati, 2010).

$$y_{i+1} = y_i + hf + \frac{h^2}{2}f' + \frac{h^3}{6}f'' + \frac{h^4}{24}f''' + \cdots \tag{2}$$

Since calculating higher-order derivatives directly is often impractical, the RK method approximates these terms using weighted averages of function evaluations at intermediate points. This leads to the general form of the explicit RK method, given by Mathews and Fink (2004). The calculation of the approximate value of y_{i+1} by using the Runge-Kutta method, which begins by determining the value of f',f'',f''', $\cdots$, as follows Mathews and Fink (2004).

$$f_i = \frac{dy}{dt}, \tag{3}$$

$$f_i' = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\frac{dy}{dt} = f_t + f_y f_i, \tag{4}$$

$$\begin{aligned} f_i'' &= \frac{\partial(f_t + f_y f_i)}{\partial t} + \frac{\partial(f_t + f_y f_i)}{\partial y}\frac{dy}{dt} \\ &= f_{tt} + f_t f_y + 2f_i f_{ty} + f_i f_t f_y + f_{yy} f_i^2, \end{aligned} \tag{5}$$

$$\begin{aligned} f_i''' &= \frac{\partial(f_{tt} + f_t f_y + 2f_i f_{ty} + f_i f_t f_y + f_{yy} f_i^2)}{\partial t} \\ &+ \frac{\partial(f_{tt} + f_t f_y + 2f_i f_{ty} + f_i f_t f_y + f_{yy} f_i^2)}{\partial y}\frac{dy}{dt} \\ &= f_{ttt} + f_{tt} f_y + 3f_t f_{ty} + 3f_i f_{tty} + f_t f_y^2 + 5f_i f_y f_{ty} \\ &+ 3f_i f_t f_{yy} + 4f_i^2 f_y f_{yy} + f_i f_y^3 + f_i^3 f_{yyy}. \end{aligned} \tag{6}$$

The Runge-Kutta method is a computational technique that is employed to solve initial value problems in both linear and nonlinear ordinary differential equations. The Runge-Kutta method, derived from the Taylor series expansion is used to solve differential equations. This technique is employed to get a greater level of precision and circumvent the need for larger derivatives by evaluating the function *f(x,y)*.

If a succession of precise solution approximations is known, with distinct sub-intervals (h), denoted as $y_{h1}, y_{h2}, y_{h3}, \cdots$, the order of accuracy (p) of these approximations can be examined using the method mentioned in reference (Mathews and Fink, 2004) which is written in Equation 7.

$$p = \log_2\left|\frac{y_{\text{exact}} - y_h}{y_{\text{exact}} - y_{h/2}}\right|, \tag{7}$$

when the exact solution is unknown, it can be investigated using Equation 8.

$$p = \log_2\left|\frac{y_h - y_{h/2}}{y_{h/2} - y_{h/4}}\right|, \tag{8}$$

$$y_{i+1} = y_i + a_1 k_1 + a_2 k_2 + \cdots + a_n k_n, \tag{9}$$

Furthermore, to validate the correctness of the order, the difference can be determined by computing the error between the numerical solution and the precise solution. For instance, let $\hat{a}$ denote the approximate value obtained from the numerical technique, and let a represent the b exact value obtained from the analytical method. In this case, the absolute error (ε) is defined as $\varepsilon = |a - \hat{a}|$ (Fardinah, 2017). The general form of the n Runge-Kutta method is written as follows Mathews and Fink (2004).

$$k_1 = hf(t_i, y_i), \tag{10}$$

$$k_2 = hf(t_i + p_1 h, y_i + q_{11} k_1), \tag{11}$$

$$k_3 = hf(t_i + p_2 h, y_i + q_{21} k_1 + q_{22} k_2), \tag{12}$$

$$\vdots$$

$$\begin{aligned} k_n &= hf\Big(t_i + p_{(n-1)}h,\ y_i + q_{(n-1,1)}k_1 + q_{(n-1,2)}k_2 + \cdots \\ &\quad + q_{(n-1,n-1)}k_{n-1}\Big). \end{aligned} \tag{13}$$

The values of a_i, p_i, and q_{ij} in Equations 9 to 13 need to be calculated in order to minimize the error for each step. The order of the Runge-Kutta method varies based on the value of n that is utilized. When $n = 1$, the method is referred to as the first order Runge-Kutta method, thus transforming Equation 9 to the form in Equation 14.

$$y_{i+1} = y_i + a_1 k_1, \tag{14}$$

where $a_1 = 1$ and k_1=hf(t_i, y_i). For $n = 2$, the method is referred to as the second-order Runge-Kutta method with the following form.

$$y_{i+1} = y_i + a_1 k_1 + a_2 k_2, \tag{15}$$

with a_1, a_2 are constant. When $a_1 = a_2 = \frac{1}{2}$, the method is called the Heun method.

$$k_1 = hf(t_i, y_i), \tag{16}$$

$$k_2 = hf(t_i + p_1 h, y_i + q_{11} k_1). \tag{17}$$

The values of a_1, a_2, p_1, and q_{11} are determined by setting Equation 15 equal to a second-order Taylor series. Let's assume that f_i represents the value of f at the point (t_i, y_i). In such case, f_t denotes the partial derivative of f(t_i, y_i) with respect to t, and f_y represents the partial derivative of f(t_i, y_i) with respect to y. In Equation 17, the variable k_2 is expanded using the Taylor series of two variables until only the first order is obtained.

$$k_2 = (f_i + p_1 h f_t + q_{11} h f_i f_y),$$

$$f(x + r, y + s) = f(x, y) + r\frac{\partial f}{\partial x + s}\frac{\partial f}{\partial y}. \tag{18}$$

k_1 does not need to be expanded because it is already in the form (t_i, y_i). Then the Equation 18 and Equation 16 are substituted into Equation 15 to get

$$y_(i + 1) = y_i + (a_1 + a_2) h f_i + a_2 h^2 (p_1 f_t + q_{11} h f_i f_y). \tag{19}$$

$y_(i + 1)$ is expanded into the Taylor series until the second-order term in obtained

$$y_(i + 1) = y_1 + h y_i' + 1/2 h^2 y_i''. \tag{20}$$

Because $y_i' = f(t_i, y_i) = f_i$ and $y_i'' = f'(t_i, y_i) = \frac{df(t_i,y_i)}{dt} = \frac{\partial f}{\partial t}\frac{dt}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt}$, equation 20 can be written as follows

$$y_(i + 1) = y_i + h f_i + \frac{1}{2} h^2 (f_t + f_y f_i) \tag{21}$$

A comparison between the Equation 19 and Equation 21 yields Equation 22

$$h f_i + 1/2 h^2 (f_t + f_y f_i) = (a_1 + a_2) h f_i + a_2 h^2 (p_1 f_t + q_1 1 h f_i f_y), \tag{22}$$

so that the following equations are obtained.

$$a_1 + a_2 = 1, a_2 p_1 = \frac{1}{2}, a_2 q_1 1 = \frac{1}{2}. \tag{23}$$

A system of Equation 23 has three equations and involves four unknown variables. Therefore, if a solution exists, it will not be unique. Obtaining a single solution is only possible by assigning a value to one of the variables. If, for instance, a_2 is equal to s, then the Equation 23 can be expressed in the following manner

$$a_1 = 1 - a_2 = 1 - s, p_1 = \frac{1}{2a_2} = \frac{1}{2s}, q_1 1 = \frac{1}{2a_2} = \frac{1}{2s}. \tag{24}$$

The value of s is not fixed, which results in multiple variants of second-order Runge-Kutta methods. Substituting s=$\frac{1}{2}$ into Equation 24 yields Equation 25

$$a_1 = \frac{1}{2}, a_2 = \frac{1}{2}, p_1 = q_1 1 = 1. \tag{25}$$

so that the general form of the second-order Runge-Kutta method can be written as follows.

$$y_(i + 1) = y_i + \frac{1}{2}(k_1 + k_2), \tag{26}$$

with

$$k_1 = hf(t_i, y_i), k_2 = hf(t_i + h, y_i + k_1).$$

The third-order Runge-Kutta technique with $n = 3$ can be derived in a similar manner as Equation 26, as shown in Equation 27 (Chapra and Canale, 2015).

$$y_(i + 1) = y_i + (\frac{1}{6}k_1 + \frac{4}{6}k_2 + \frac{1}{6}k_3), \tag{27}$$

with

$$k_1 = hf(t_i, y_i),$$
$$k_2 = hf(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1),$$
$$k_3 = hf(t_i + h, y_i - k_1 + 2k_2).$$

Moreover, the fourth-order Runge-Kutta technique with a value of n equal to 4 is widely favored due to its frequent application in differential equation solving and its accuracy of $O(h^4)$. The fourth-order Runge-Kutta method can be as shown in Equation 28 (Dukkipati, 2010)

$$y_(i+1) = y_i + (\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4), \tag{28}$$

with

$$k_1 = hf(t_i, y_i),$$
$$k_2 = hf(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1),$$
$$k_3 = hf(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2),$$
$$k_4 = hf(t_i + h, y_i + k_3).$$

The efficiency of RK methods depends on the number of function evaluations per step. While RK4 requires four function evaluations, RK5 typically requires five or six evaluations per step (Kalogiratou et al., 2014). Some modified RK5 methods attempt to reduce the number of function evaluations while maintaining high accuracy (Agbeboh et al., 2020; Dennis, 2012). Implicit RK methods, such as Backward Euler or Gauss-Legendre RK, provide better stability properties for stiff ODEs, but they require solving nonlinear equations at each step, increasing computational cost (Kopecz and Meister, 2018). RK methods are extensively used in various fields due to their robustness and ease of implementation. In physics and engineering, they are applied to model motion in classical mechanics, wave propagation, and electrical circuit analysis (Kalogiratou et al., 2014). In the aerospace industry, RK methods are used to simulate spacecraft trajectories and orbital mechanics (Chandru et al., 2017). In biology and medicine, they play a crucial role in modeling population dynamics, epidemiological models, and pharmacokinetics (Mechee et al., 2013). In climate science, they are employed to solve fluid dynamics and atmospheric models for weather prediction (Kennedy and Carpenter, 2019). Additionally, in economics, RK methods are utilized to forecast financial markets and model dynamic economic systems (Monovasilis et al., 2016).

The Runge-Kutta (RK) method offers several advantages, including higher accuracy than Euler methods without requiring explicit higher-order derivatives, simple implementation for both explicit and implicit variants, and wide applicability across various scientific disciplines. However, it also has some limitations, such as higher computational cost compared to multistep methods like Adams-Bashforth, limited stability in explicit methods, which necessitates careful step size selection for stiff ODEs (Kopecz and Meister, 2018), and more function evaluations per step compared to lower-order methods. To address these challenges, ongoing efforts to enhance RK methods focus on improving efficiency, stability, and error control, leading to various modifications such as embedded RK methods with adaptive step sizes (Chandru et al., 2017; Kalogiratou et al., 2014).

3. RESULTS AND DISCUSSION

This section outlines the derivation process of the fifth-order Runge-Kutta (RK5) formula, detailing the sequential steps involved in its formulation. The proposed method is then applied to solve initial value problems (IVPs) in both single ordinary differential equations (ODEs) and systems of ODEs. Furthermore, a comprehensive evaluation of its accuracy and stability is conducted to assess its effectiveness compared to existing RK5 methods.

3.1 Derivation of the Fifth Order Runge-Kutta Formula

The general form of the fifth-order Runge-Kutta method is

$$y_(i+1) = y_i + a_1k_1 + a_2k_2 + a_3k_3 + a_4k_4 + a_5k_5, \tag{29}$$

where a_1, a_2, a_3, a_4, and a_5 are constants.

$$k_1 = hf(t_i, y_i),$$
$$k_2 = hf(t_i + p_1h, y_i + q_{11}k_1),$$
$$k_3 = hf(t_i + p_2h, y_i + q_{21}k_1 + q_{22}k_2),$$
$$\vdots$$
$$k_4 = hf(t_i + p_3h, y_i + q_{31}k_1 + q_{32}k_2 + q_{33}k_3),$$
$$k_5 = hf(t_i + p_4h, y_i + q_{41}k_1 + q_{42}k_2 + q_{43}k_3 + q_{44}k_4).$$

Then, we expand k_2, k_3, k_4, k_5 using a two-variable of Taylor series as follows

$$k_1 = hf(t_i, y_i) = hf_i, \tag{30}$$

$$k_2 = hf(t_i + p_1h, y_i + q_{11}k_1) = h(f_i + p_1hf_t + q_{11}hf_if_y), \tag{31}$$

$$\begin{aligned} k_3 =& hf(t_i + p_2h, y_i + q_21k_1 + q_22k_2) \\ &= h(f_i + p_2hf_t + q_{21}hf_if_y + q_{22}hf_if_y + q_{22}h^2p_1f_tf_y \\ &+ q_{22}q_{11}h^2f_i(fy)^2), \end{aligned} \tag{32}$$

$$
\begin{aligned}
k_4 =& hf(t_i + p_3 h, y_i + q_{31}k_1 + q_{32}k_2 + q_{33}k_3) \\
&= h(f_i + p_3 h f_t + q_{31} h f_i f_y + q_{32} h f_i f_y + p_1 q_{32} h^2 f_t f_y \\
&+ q_{11} q_{32} h^2 f_i (f_y)^2 + q_{33} h f_i f_y + p_2 q_{33} h^2 f_t f_y \\
&+ q_{21} q_{33} h^2 f_i (f_y)^2 + q_{22} q_{33} h^2 f_i (f_y)^2 + p_1 q_{22} q_{33} h^3 \\
&\quad f_t (f_y)^2 + q_{11} q_{22} q_{33} h^3 f_i (f_y)^3),
\end{aligned} \tag{33}
$$

$$
\begin{aligned}
k_5 =& hf(t_i + p_4 h, y_i + q_{41}k_1 + q_{42}k_2 + q_{43}k_3 + q_{44}k_4), \\
& h(f_i + p_4 h f_t + q_{41} h f_i f_y + q_{42} h f_i f_y + p_1 q_{42} h^2 f_t f_y \\
&+ q_{11} q_{42} h^2 f_i (f_y)^2 + q_{43} h f_i f_y + p_2 q_{43} h^2 f_t f_y \\
&+ q_2 1 q_{43} h^2 f_i (f_y)^2 + q_{22} q_{43} h^2 f_i (f_y)^2 \\
&+ q_{22} q_{43} h^3 p_1 f_t (f_y)^2 + q_{22} q_{11} q_{43} h^3 f_i (f_y)^3 \\
&+ q_{44} h f_i f_y + p_3 q_{44} h^2 f_t f_y + q_{31} q_{44} h^2 f_i (f_y)^2 \\
&+ q_{32} q_{44} h^2 f_i (f_y)^2 + p_1 q_{32} q_{44} h^3 f_t (f_y)^2 \\
&+ q_{11} q_{32} q_{44} h^3 f_i (f_y)^3 \\
&+ q_{33} q_{44} h^2 f_i (f_y)^2 + p_2 q_{33} q_4 4 h^3 f_t (f_y)^2 \\
&+ q_{21} q_{33} q_{44} h^3 f_i (f_y)^3 + q_{22} q_{33} q_{44} h^3 f_i (f_y)^3 \\
&+ p_1 q_{22} q_{33} q_{44} h^4 f_t (f_y)^3 \\
&+ q_{11} q_{22} q_{33} q_{44} h^4 f_i (f_y)^4).
\end{aligned} \tag{34}
$$

The Equations (30 - 34) are substituted into Equation 29, yielding

$$
\begin{aligned}
y_(i+1) =& y_i + (a_1 + a_2 + a_3 + a_4 + a_5) h f_i + (a_2 p_1 + a_3 p_2 \\
&+ a_4 p_3 + a_5 p_4) h^2 f_t + (a_2 q_{11} + a_3 q_{21} + a_3 q_{22} \\
&+ a_4 q_{31} + a_4 q_{32} + a_4 q_{33} + a_5 q_{41} + a_5 q_{42} \\
&+ a_5 q_{43} + a_5 q_{44}) h^2 f_i f_y + (a_3 p_1 q_{22} \\
&+ a_4 p_1 q_{32} + a_4 p_2 q_{33} + a_5 p_1 q_{42} + a_5 p_2 q_{43} \\
&+ a_5 p_3 q_{44}) h^3 f_t f_y + (a_3 q_{11} q_{22} + a_4 q_{11} q_{32} \\
&+ a_4 q_{21} q_{33} + a_4 q_{22} q_{33} + a_5 q_{11} q_{42} + a_5 q_{21} q_{43} \\
&+ a_5 q_{22} q_{43} + a_5 q_{31} q_{44} + a_5 q_{32} q_4 \\
&+ a_5 q_{33} q_{44}) h^3 f_i (f_y)^2 + (a_4 p_1 q_{22} q_{33} \\
&+ a_5 p_1 q_{22} q_{43} + a_5 p_1 q_{32} q_{44} \\
&+ a_5 p_2 q_{33} q_{44}) h^4 f_t (f_y)^2 + (a_4 q_{11} q_{22} q_{33} \\
&+ a_5 q_{11} q_{22} q_{43} + a_5 q_{11} q_{32} q_{44} \\
&+ a_5 q_{21} q_{33} q_{44} \\
&+ a_5 q_{22} q_{33} q_{44}) h^4 f_i (f_y)^3 \\
&+ (a_5 p_1 q_{22} q_{33} q_{44}) h^5 f_t (f_y)^3 \\
&+ (a_5 q_{11} q_{22} q_{33} q_{44}) h^5 f_i (f_y)^4.
\end{aligned} \tag{35}
$$

The left-hand side of Equation (29) is expanded using a Taylor series as follows

$$
\begin{aligned}
y_(i+1) = y_i + hf + \frac{h^2}{2} f' + \frac{h^3}{6} f'' + \frac{h^4}{24} f''' \\
+ \frac{h^5}{120} f^{(4)},
\end{aligned} \tag{36}
$$

with f, f', f'', and f''' are defined in the Equations (3 - 6) respectively, and $f^{(4)}$ is defined as follows

$$
\begin{aligned}
f^{(4)} = f^{(4)}(t_i, y_i) =& \frac{\partial}{\partial t}(f_t tt + f_t t f_y \\
&+ 3 f_t f_t y + 3 f_i f_t t y + f_t f_y^2 + 5 f_i f_y f_t y \\
&+ 3 f_i f_t f_y y + 3 f_i^2 f_t y y + 4 f_i^2 f_y f_y y \\
&+ f_i f_y^3 + f_i^3 f_y y y) + \frac{\partial}{\partial y}(f_t tt + f_t t f_y \\
&+ 3 f_t f_t y + 3 f_i f_t t y + f_t f_y^2 \\
&+ 5 f_i f_y f_t y + 3 f_i f_t f_y y + 3 f_i^2 f_t y y \\
&+ 4 f_i^2 f_y f_y y + f_i f_y^3 + f_i^3 f_{yyy}) \frac{d}{dt}, \\
=& f_{tttt} + f_{ttt} f_y + 4 f_t t f_{ty} + 6 f_t f_{tty} \\
&+ 4 f_i f_t t t y + f_t t (f_y)^2 \\
&+ 7 f_t f_t y f_y + 8 f_i (f_t y)^2 + 9 f_i f_y f_t t y \\
&+ 3 (f_t)^2 f_y y + 3 f_i f_t t f_y y \\
&+ 9 f_i f_t f_t y y + 6 (f_i)^2 f_{ttyy} \\
&+ f_t t f_t y f_i + 3 f_t f_t t y f_i \\
&+ 13 f_i f_t f_y f_y y + 12 (f_i)^2 f_t y f_y y \\
&+ 15 (f_i)^2 f_y f_t y y + f_t (f_y)^3 \\
&+ 9 f_i f_t y (f_y)^2 + 6 (f_i)^2 f_t f_{yyy} \\
&+ 4 (f_i)^3 f_{tyyy} + 11 (f_i)^2 (f_y)^2 f_y y \\
&+ 4 (f_i)^3 (f_{yy})^2 + 7 (f_i)^3 f_y f_{yyy} \\
&+ f_i (f_y)^4 + (f_i)^4 f_{yyyy}.
\end{aligned} \tag{37}
$$

The Equations (3 - 6) and Equation 37 are substituted into the Equation 36, we obtain

$$
\begin{aligned}
y_(i+1) =& y_i + h f_i + \frac{h^2}{2}(f_t + f_y f_i) \\
&+ \frac{h^3}{6}(f_{tt} + 2 f_{ty} f_i + f_y f_t + f_{yy}(f_i)^2 \\
&+ (f_y)^2 f_i) + \frac{h^4}{24}(f_{ttt} + f_t t f_y + 3 f_t f_t y \\
&+ 3 f_i f_{tty}) + \frac{h^4}{24}(f_t (f_y)^2 + 5 f_i f_y f_t y \\
&+ 3 f_i f_t f_{yy} + 3 (f_i)^2 f_{tyy} + 4 (f_i)^2 f_y f_{yy} + f_i (f_y)^3 \\
&+ (f_i)^3 f_{yyy}) + \frac{h^5}{120}(f_{tttt} + f_{ttt} f_y)
\end{aligned}
$$

$$+\frac{h^5}{120}(4f_{tt}f_{ty}+6f_tf_{tty}+4f_if_{ttty}$$
$$+f_{tt}(f_y)^2+7f_tf_ty f_y+8f_i(f_ty)^2$$
$$+9f_if_yf_tty+3(f_t)^2f_yy+3f_if_ttf_yy$$
$$+\frac{h^5}{120}(9f_if_tf_{tyy}+6(f_i)^2f_{ttyy}$$
$$+f_{tt}f_tyf_i+3f_tf_{tty}f_i$$
$$+13f_if_tf_yf_yy+12(f_i)^2f_tyf_{yy}$$
$$+15(f_i)^2f_yf_{tyy})+\frac{h^5}{120}(f_t(f_y)^3$$
$$+9f_if_ty(f_y)^2+6(f_i)^2f_tf_{yyy}$$
$$+4(f_i)^3f_tyyy+11(f_i)^2(f_y)^2f_yy$$
$$+4(f_i)^3(f_yy)^2)+V(7(f_i)^3f_yf_yyy$$
$$+f_i(f_y)^4+(f_i)^4f_yyyy). \tag{38}$$

Furthermore, based on the comparison between the coefficients of the Equation 35 and Equation 38, a system of equations is obtained as follows

$$a_1+a_2+a_3+a_4+a_5=1, \tag{39}$$

$$a_2p_1+a_3p_2+a_4p_3+a_5p_4=\frac{1}{2}, \tag{40}$$

$$a_2q_{11}+a_3q_{21}+a_3q_{22}+a_4q_{31}+a_4q_{32}$$
$$+a_4q_{33}+a_5q_{41}+a_5q_{42}+a_5q_{43}+a_5q_{44}=\frac{1}{2}, \tag{41}$$

$$a_3p_1q_{22}+a_4p_1q_{32}+a_4p_2q_{33}+a_5p_1q_{42}$$
$$+a_5p_2q_{43}+a_5p_3q_{44}=\frac{1}{6}, \tag{42}$$

$$a_3q_{11}q_{22}+a_4q_{11}q_{32}+a_4q_{21}q_{33}+a_4q_{22}q_{3}3$$
$$+a_5q_11q_{42}+a_5q_{21}q_{43}+a_5q_{22}q_{43}$$
$$+a_5q_{31}q_{44}+a_5q_{32}q_{44}+a_5q_{33}q_{44}=\frac{1}{6}, \tag{43}$$

$$a_4p_1q_{22}q_{3}3+a_5p_1q_2 2q_4 3$$
$$+a_5p_1q_3 2q_4 4+a_5p_2q_3 3q_4 4=\frac{1}{24}, \tag{44}$$

$$a_4q_{11}q_{22}q_{33}+a_5q_{11}q_{22}q_{43}$$
$$+a_5q_{11}q_{32}q_{44}+a_5q_{21}q_{33}q_{44}$$
$$+a_5q_{22}q_{33}q_{44}=\frac{1}{24}, \tag{45}$$

$$a_5p_1q_{22}q_{33}q_{44}=\frac{1}{120}, \tag{46}$$

$$a_5q_{11}q_{22}q_{33}q_{44}=\frac{1}{120}. \tag{47}$$

The system of Equations (39 - 47) consists of nine equations with 19 unknown parameter values. In order to derive the formula, some requirements must be provided as follows in Equation 48

$$a_1=\frac{1}{6}, a_2=\frac{1}{3}, a_3=\frac{1}{3}, a_4=\frac{1}{10},$$
$$a_5=\frac{1}{15}, p_1=\frac{1}{2}, p_2=\frac{1}{2}, p_3=1, p_4=1,$$
$$q_{11}=\frac{1}{2}, q_{21}=0, q_{22}=\frac{1}{2}, q_{31}=0, q_{32}=0,$$
$$q_{33}=1, q_{41}=\frac{1}{2}, q_{42}=0, q_{43}=0, q_{44}=\frac{1}{2}. \tag{48}$$

Substituting the determined parameter values into Equation (29) yields a simplified and more efficient fifth-order Runge-Kutta formula.

$$y_(i+1)=y_i+\frac{1}{6}k_1+\frac{1}{3}k_2+\frac{1}{3}k_3+\frac{1}{10}k_4+\frac{1}{15}k_5, \tag{49}$$

with

$$k_1=hf(t_i,y_i),$$
$$k_2=hf(t_i+\frac{1}{2}h, y_i+\frac{1}{2}k_1),$$
$$k_3=hf(t_i+\frac{1}{2}h, y_i+\frac{1}{2}k_2),$$
$$k_4=hf(t_i+h, y_i+k_3),$$
$$k_5=hf(t_i+h, y_i+\frac{1}{2}k_1+\frac{1}{2}k_4).$$

Moreover, the truncation error of the fifth-order Runge-Kutta method in Equation 49, as given in Equation 53, can be determined by comparing the values of $y_{(i+1)}$ obtained using the fifth-order Runge-Kutta method with those derived from the Taylor series expansion. The truncation error for this method is computed as follows:

Table 1. Comparison of Numerical Solution of $y' = t + y;\ y(0) = 0,\ 0 \le t \le 1$

i	t_i	Exact Solution	RK5 (Agbeboh et al., 2020)	RK5 (Chandru et al., 2017)	Proposed RK5
1	0.00	0.0000000000	0.0000000000	0.0000000000	0.0000000000
2	0.10	0.0051709181	0.0051709182	0.0050199558	0.0051709167
3	0.20	0.0214027582	0.0214027585	0.0210212324	0.0214027550
4	0.30	0.0498588076	0.0498588081	0.0496114270	0.0498588024
5	0.40	0.0918246976	0.0918246985	0.0915305987	0.0918246900
6	0.50	0.1487212707	0.1487212718	0.1483774167	0.1487212602
7	0.60	0.2221188004	0.2221188019	0.2217208354	0.2221187865
8	0.70	0.3137527075	0.3137527094	0.3132952336	0.3137526895
9	0.80	0.4255409285	0.4255409310	0.4250175874	0.4255409058
10	0.90	0.5596031112	0.5596031142	0.5590065808	0.5596030829
11	1.00	0.7182818285	0.7182818322	0.7176037733	0.7182817938

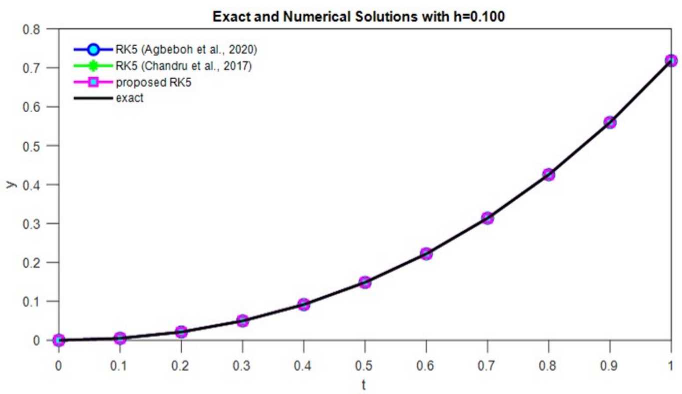

Figure 1. Numerical Solution of $y' = t + y; y(0) = 0$, with $0 \le t \le 1$

Table 2. Comparison of Time of Running Program (1)

RK5 (Agbeboh et al., 2020)	RK5 (Chandru et al., 2017)	Proposed RK5
0.00001 s	0.00007 s	0.06

$$
\begin{aligned}
error = {} & \frac{h^3}{6}(f_{tt} + 2f_{ty}f_i + f_{yy}(f_i)^2) + \frac{h^4}{24}(f_{ttt} + f_{tt}f_y \\
& + 3f_tf_{ty} + 5f_if_yf_{ty} + 3f_if_tf_{yy} \\
& + 3(f_i)^2f_{tyy} + 4(f_i)^2f_yf_{yy} \\
& + (f_i)^3f_{yyy}) + \frac{h^5}{120}(f_{tttt} + f_{ttt}f_y \\
& + 4f_tt f_{ty} + 6f_tf_{tty} + 4f_if_{ttty} + f_{tt}(f_y)^2 \\
& + 7f_tf_{ty}f_y + 8f_i(f_ty)^2 + 9f_if_yf_{tty} + 3(f_t)^2f_{yy} \\
& + 3f_if_{tt}f_{yy} + 9f_if_tf_{tyy} + 6(f_i)^2f_{ttyy} \\
& + f_{tt}f_{ty}f_i + 3f_tf_{tty}f_i \\
& + 13f_if_tf_yf_yy + 12(f_i)^2f_ty f_{yy} + 15(f_i)^2f_yf_{tyy} \\
& + 9f_if_ty(f_y)^2 + 6(f_i)^2f_tf_{yyy} \\
& + 4(f_i)^3f_{tyyy} + 11(f_i)^2(f_y)^2f_{yy} \\
& + 4(f_i)^3(f_yy)^2 + 7(f_i)^3f_yf_{yyy} \\
& + (f_i)^4f_{yyyy}) + O(h^6). \qquad (50)
\end{aligned}
$$

The truncation error in Equation 50 confirms that the fifth-order Runge-Kutta method described in Equation (53) exhibits a local truncation error of $O(h^6)$ and a global truncation error of $O(h^5)$, ensuring high accuracy and numerical stability.

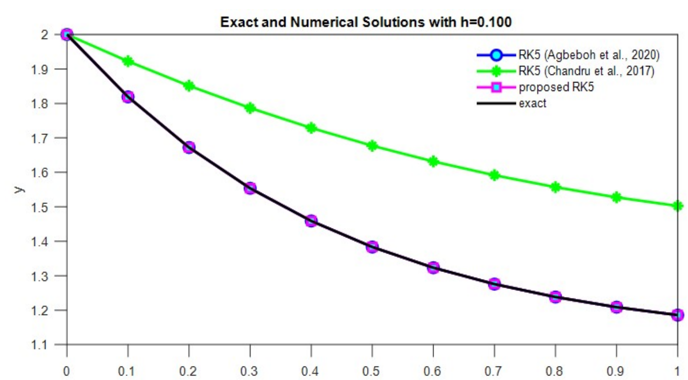

Figure 2. Numerical Solution of $y' = 2cos(t)(1 - y)$ $; y(0) = 2, 0 \le t \le 1$

3.2 Numerical Solution of Ordinary Differential Equations

In this section, the newly developed formula is applied alongside the formulas presented in references Agbeboh et al. (2020)

Table 3. Comparison of Numerical Solutions $y' = 2\cos(t)(1 - y);\ y(0) = 2,\ \ 0 \le t \le 1$

i	t_i	Exact solution	RK5 (Agbeboh et al., 2020)	RK5 (Chandru et al., 2017)	Proposed RK5
1	0.00	2.0000000000	2.0000000000	2.0000000000	2.0000000000
2	0.10	1.8190035724	1.8189971191	1.9222195189	1.8190009824
3	0.20	1.6721063705	1.6720974729	1.8511293493	1.6721030295
4	0.30	1.5537508552	1.5537418359	1.7867083394	1.5537478135
5	0.40	1.4589395924	1.4589315990	1.7288103600	1.4589373540
6	0.50	1.3833330531	1.3833264779	1.6772000835	1.3833317735
7	0.60	1.3232643250	1.3232591203	1.6315843283	1.3232639581
8	0.70	1.2757018159	1.2756977126	1.5916385951	1.2757022164
9	0.80	1.2381839078	1.2381805540	1.5570288537	1.2381848963
10	0.90	1.2087425052	1.2087395456	1.5274289387	1.2087439062
11	1.00	1.1858264752	1.1858235899	1.5025340887	1.1858281370

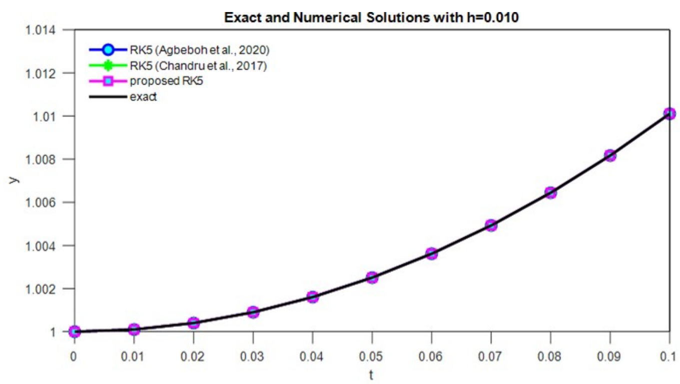

Figure 3. Comparison of Running Program Time (52)

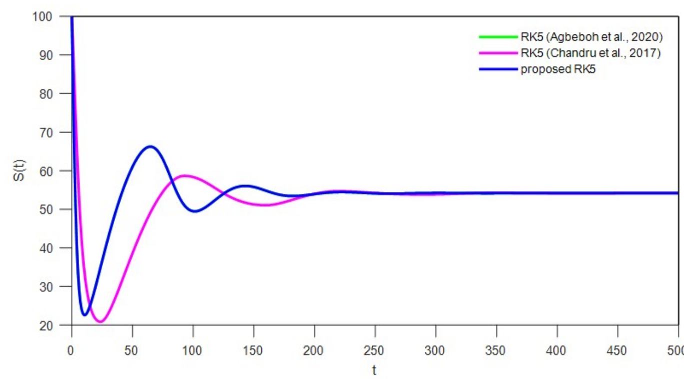

Figure 4. Numerical Solution of S in System 52

Table 4. Comparison of Time of Running Program (2)

RK5 (Agbeboh et al., 2020)	RK5 (Chandru et al., 2017)	Proposed RK5
0.000011 s	0.00009 s	0.00007

and Chandru et al. (2017) to solve various initial value problems in ordinary differential equations. The numerical solutions obtained from each method are then evaluated against the exact solution to assess their accuracy. Furthermore, the precision of the proposed method is analyzed by examining its order of accuracy. Several test cases are considered to compare the performance of the new formula with the established Runge-Kutta methods from references Agbeboh et al. (2020) and Chandru et al. (2017) . These comparisons provide insights into the efficiency, stability, and overall effectiveness of the proposed approach.

(1) $y' = t + y; y(0) = 0, 0 \le t \le 1$, and the exact solution is $y(t) = -t - 1 + e^t$.

The exact and numerical solutions for each formula are presented in Table 1. As illustrated in Figure 1, the proposed formula exhibits strong agreement with the existing formula presented in Agbeboh et al. (2020), as both methods converge closely to the exact solution, as shown in Table 1. In contrast, the numerical solution obtained using the Runge-Kutta method in Chandru et al. (2017) shows noticeable deviation from the exact solution. The execution times for each method are summarized in Table 2, where all three formulas demonstrate computational efficiency with execution times on the order of 10^{-5} seconds. Among them, the formula in Agbeboh et al. (2020) achieves the fastest execution time compared to the other methods.

(2) $y' = 2cos(t)(1 - y); y(0) = 2, 0 \le t \le 1$ with exact solution $y(t) = 1 + e^{(-2sin(t))}$.

According to Figure 2, the newly derived Equation (53) and the method presented in Agbeboh et al. (2020) closely match the exact solution. However, upon zooming in on the graph, the proposed formula demonstrates higher precision than existing methods.

In contrast, the numerical solution obtained from the formula in Chandru et al. (2017) shows a significant deviation from the exact solution. The execution times for all three formulas are within the range of 10^{-5} seconds, with the method in Agbeboh et al. (2020) demonstrating the fastest execution

Table 5. Comparison of Numerical Solutions of $y' = 2ty + 2t^3y^2;\ y(0) = 1,\ 0 \le t \le 0.1$

i	t_i	Exact solution	RK5 (Agbeboh et al., 2020)	RK5 (Chandru et al., 2017)	Proposed RK5
1	0.00	1.0000000000	1.0000000000	1.0000000000	1.0000000000
2	0.01	1.0001000100	1.0001000092	1.0000970812	1.0001000100
3	0.02	1.0004001601	1.0004001584	1.0003966569	1.0004001601
4	0.03	1.0009008107	1.0009008082	1.0008969360	1.0009008107
5	0.04	1.0016025641	1.0016025608	1.0015983909	1.0016025641
6	0.05	1.0025062657	1.0025062615	1.0025018193	1.0025062657
7	0.06	1.0036130068	1.0036130018	1.0036082871	1.0036130069
8	0.07	1.0049241282	1.0049241224	1.0049191179	1.0049241283
9	0.08	1.0064412238	1.0064412171	1.0064358930	1.0064412239
10	0.09	1.0081661458	1.0081661382	1.0081604537	1.0081661459
11	0.10	1.0101010101	1.0101010017	1.0100949062	1.0101010102

Table 6. Comparison of Time of Running Program (3)

RK5 (Agbeboh et al., 2020)	RK5 (Chandru et al., 2017)	Proposed RK5
0.000037 s	0.000018 s	0.000016

Table 7. Comparison of Numerical Solution of System (52)

Variable	Endemic Equilibrium	RK5 (Agbeboh et al., 2020)	RK5 (Chandru et al., 2017)	Proposed RK5
S_{tn}	54.1666666667	54.1661226174	54.1702413778	54.1661215113
I_{tn}	3.4423076923	3.4424488131	3.4439229237	3.4424498982

time among them (Table 3 and Table 4).

(3) $y' = 2ty + 2t^3y^2; y(0) = 1, 0 \le t \le 0.1$, and exact solution $y(t) = \frac{1}{1-t^2)}$.

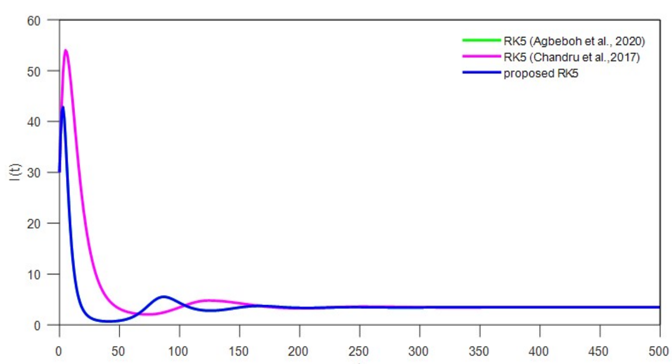

Figure 5. Numerical Solution of I in System (52)

The numerical solution for case 3 is illustrated in Figure 3, showing that the proposed formula closely approximates the exact solution. Furthermore, the proposed formula achieves the shortest execution time among all methods, as presented in Tables 5 and 6, with execution times falling within the range of 10^{-5} seconds.

Table 8. Comparison of Time of Running Program (52)

RK5 (Agbeboh et al., 2020)	RK5 (Chandru et al., 2017)	Proposed RK5
0.336251 s	0.284864 s	0.359252 s

3.3 Numerical Solution for a System or Ordinary Differential Equations

In this section, we apply the newly developed formula along with the formulas presented in Agbeboh et al. (2020) and Chandru et al. (2017) to solve an initial value problem in a system of ordinary differential equations. Subsequently, we evaluate the convergence of all solutions toward the equilibrium point. A related approach involving the SVIR model for COVID-19 cases in Indonesia has been demonstrated in Parhusip et al. (2022). Consider the following a slightly modified system of differential equations based on Gumus et al. (2022):

$$\begin{aligned} \frac{dS}{dt} &= -pS - \frac{\alpha}{N}IS + \beta(N - S), \\ \frac{dI}{dt} &= \frac{\alpha}{N}IS + (-\beta - \gamma)I, \\ \frac{dR}{dt} &= -\beta R + \gamma I + pS, \end{aligned} \tag{51}$$

Table 9. Numerical Solution of $y' = t + y;\ y(0) = 0,\ 0 \le t \le 1$ with $h_1 = 0.1$ and $h_2 = 0.05$

i	t_i	y_{exact}	$y_{h=0.1}$	$y_{h=0.05}$
1	0.00	0.0000000000	0.0000000000	0.0000000000
2	0.05	0.0012710964		0.0012710964
3	0.10	0.0051709181	0.0051709167	0.0051709181
4	0.15	0.0118342427		0.0118342427
5	0.20	0.0214027582	0.0214027550	0.0214027581
6	0.25	0.0340254167		0.0340254166
7	0.30	0.0498588076	0.0498588024	0.0498588074
8	0.35	0.0690675436		0.0690675484
9	0.40	0.0918246976	0.0918246900	0.0918246974
10	0.45	0.1183121855		0.1183121852
11	0.50	0.1487212607	0.1487212602	0.1487212603
12	0.55	0.1832530179		0.1832530175
13	0.60	0.2221188004	0.2221187365	0.2221187999
14	0.65	0.2655408290		0.2655408285
15	0.70	0.3137525075	0.3137526895	0.3137525069
16	0.75	0.3670000166		0.3670000160
17	0.80	0.4255409285	0.4255409058	0.4255409278
18	0.85	0.4896468511		0.4896468511
19	0.90	0.5596031112	0.5596030829	0.5596031101
20	0.95	0.6357096593		0.6357096583
21	1.00	0.7182818285	0.7182817938	0.7182818273

Table 10. Order of Accuracy of Numerical Solutions of $y' = t + y; y(0) = 0, 0 \le t \le 1$

i	1	2	3	4	5	6	7	8	9	10	11
t_i	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Accuracy Order			4.95450	4.70043	5.24792	5.12928	4.79701	4.90689	5.01919	4.96962	4.97936

with $S(0) > 0$, $I(0) > 0$, and $R(0) > 0$. Since the system in Equation 51 does not explicitly depend on the variable R, it can be omitted. As a result, the system 51 can be reformulated as:

$$\begin{aligned} \frac{dS}{dt} &= -pS - \frac{\alpha}{N}IS + \beta(N - S), \\ \frac{dS}{dt} &= \frac{\alpha}{N}IS + (-\beta - \gamma)I, \end{aligned} \tag{52}$$

with $S(0) > 0 and I(0) > 0$. Based on the system in Equation 52, the disease-free equilibrium point $P_0 = (\frac{\beta N}{\beta + p}, 0)$ and the endemic equilibrium point $P_1 = \frac{(\beta+\gamma)N}{\alpha}, \frac{N(\alpha\beta-(\beta+\gamma)(p+\beta)}{\alpha(\beta+\gamma)}$ are obtained.

The values of the endemic equilibrium point S,I and the final-time values of S,I for each method are presented in Table 7. According to Figure 4 and 5, the proposed RK5 formula and the existing formula in Agbeboh et al. (2020) demonstrate a higher degree of convergence toward the endemic equilibrium points S and I. Table 8 presents the execution time for each method, highlighting the computational efficiency of the proposed approach.

3.4 Accuracy Level Analysis

3.4.1 Order of Accuracy

The local truncation error and global truncation error of the proposed fifth-order Runge-Kutta method were determined to be of order $O(h^6)$ and $O(h^5)$, respectively. To validate this accuracy, we conducted a numerical test on case 1, approximating the solution using two different sub-interval widths. The chosen step sizes were $h_1 = h = 0.1$ and $h_2 = \frac{1}{2}h = 0.05$, allowing for a comparative analysis of error reduction as the step size decreases. case 1 represents a differential equation with a known exact solution, enabling the computation of the truncation error using the accuracy order formula in Equation 7. The numerical solutions at each step are evaluated in Table 9, which is then used to determine the accuracy order recorded in Table 10. The estimated order of accuracy for the revised fifth-order Runge-Kutta Equation 53 ranges from 4.70043-5.24792, confirming that the method achieves an accuracy of $O(h^5)$.

Table 11. Comparison of Numerical Error for Each Case (1)

i	t_i	Error RK5 (Agbeboh et al., 2020)	Error RK5 (Chandru et al., 2017)	Error of Proposed RK5
1	0.00	0.0000000000	0.000000000	0.000000000
2	0.10	0.0000000002	0.0001509623	0.000000014
3	0.20	0.0000000003	0.0002015258	0.000000031
4	0.30	0.0000000006	0.0002473806	0.000000052
5	0.40	0.0000000008	0.0002940990	0.000000076
6	0.50	0.0000000011	0.0003438540	0.000000105
7	0.60	0.0000000015	0.0003979650	0.000000139
8	0.70	0.0000000020	0.0004574739	0.000000180
9	0.80	0.0000000025	0.0005233411	0.000000227
10	0.90	0.0000000031	0.0005965304	0.000000282
11	1.00	0.0000000038	0.0006780552	0.000000347

3.4.2 Error Comparison

In this section, we analyze the differences between the numerical and exact solutions for each variant of the fifth-order Runge-Kutta method. Additionally, we assess the computational efficiency of each method in generating accurate results. (1) $y' = t + y; y(0) = 0, 0 \leq t \leq 1$ with exact solution $y(t) = -t + 2e^{t-1} - 1$.

Table 11 shows that the maximum error for Equation (53), solved using the fifth-order Runge-Kutta method, is 3.47×10^{-5}, which is significantly smaller than the estimated accuracy of $h^5 = 10^{-10}$. Furthermore, the proposed formula, as compared to that of Agbeboh et al. (2020) consistently yields errors of lower magnitude.

(2) $y' = 2cos9t(1 - y); y(0) = 2, 0 \leq t \leq 1$ with exact solution $y(t) = 1 + e^{-2sint}$.

Table 12 shows that the maximum error produced by the proposed Runge-Kutta method is 3.3409×10^{-6}, which is significantly smaller than the estimated order of accuracy, $h^5 = 10^{-5}$. Moreover, the error obtained using the proposed method is consistently lower than that of the existing Runge-Kutta methods, demonstrating its superior accuracy.

(3) $y' = 2ty + 2t^3y^2; y(0) = 1, 0 \leq t \leq 0.1$ with exact solution $y(t) = \frac{1}{(1-t^2)}$.

Table 13 shows that the proposed fifth-order Runge-Kutta method achieves a maximum error of 10^{-10}, which aligns with the expected order of accuracy. Additionally, this method exhibits lower error compared to other existing fifth-order Runge-Kutta methods, further validating its enhanced precision and numerical robustness.

Based on the presented test cases, the implementation of the fifth-order Runge-Kutta method for Equation (53), as well as the methods proposed by Agbeboh et al. (2020) and Chandru et al. (2017), consistently demonstrates that the new formula in Equation (53) yields lower errors compared to existing methods, while maintaining an accuracy order of $O(h^5)$. These results indicate that the proposed fifth-order Runge-Kutta method is superior to other methods of the same order in terms of both accuracy and efficiency. Furthermore, the novel and simplified formulation of this approach exhibits excellent performance and significantly reduces execution time, establishing it as a highly effective and efficient alternative for solving initial value problems in ordinary differential equations.

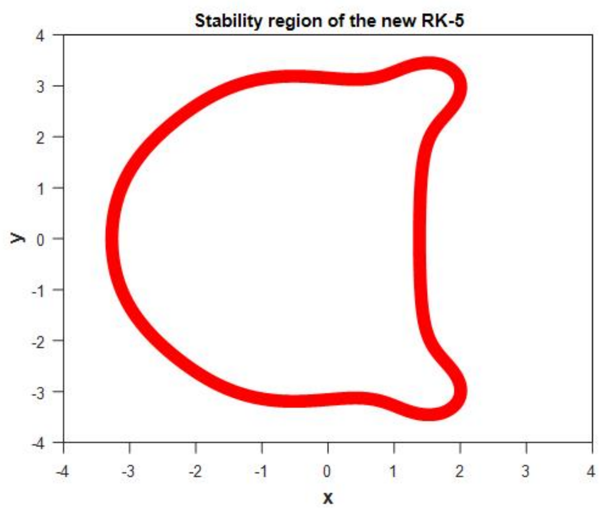

Figure 6. Stability Region of the New-Fifth Runge-Kutta Formula

3.4.3 Stability Analysis

Based on reference Chandru et al. (2017), the proposed method is applied to the standard linear test equation $f(t, y) = y' = \lambda y$ to analyze its stability properties. By introducing the scaling factor $\hat{h} = h\lambda$ determine the stability region of the new fifth-order Runge-Kutta method. Consequently, we obtain

$$y_(i+1) = y_i(1 + h + \frac{1}{2}\hat{h}^2 + \frac{1}{6}\hat{h}^3 + \frac{1}{24}\hat{h}^4 + \frac{1}{120}\hat{h}^5 + \frac{1}{120}\hat{h}^6) = R(\hat{h})y_i, \quad (53)$$

Table 12. Comparison of Numerical Error for Each Case (2)

i	t_i	Error RK5 (Agbeboh et al., 2020)	Error RK5 (Chandru et al., 2017)	Error of Proposed RK5
1	0.00	0.0000000000	0.0000000000	0.0000000000
2	0.10	0.0000064533	0.1032159465	0.0000025900
3	0.20	0.0000088976	0.1790229788	0.0000033409
4	0.30	0.0000090193	0.2329574842	0.0000030417
5	0.40	0.0000079934	0.2698707677	0.0000022383
6	0.50	0.0000065752	0.2938670303	0.0000012796
7	0.60	0.0000052047	0.3083200033	0.0000003669
8	0.70	0.0000041033	0.3159367792	0.0000004005
9	0.80	0.0000033539	0.3188449459	0.0000009885
10	0.90	0.0000029596	0.3186864336	0.0000014010
11	1.00	0.0000028853	0.3167076135	0.0000016618

Table 13. Comparison of Numerical Error for Each Case (3)

i	t_i	Error RK5 (Agbeboh et al., 2020)	Error RK5 (Chandru et al., 2017)	Error of Proposed RK5
1	0.00	0.0000000000	0.0000000000	0.0000000000
2	0.01	0.0000000008	0.0000029288	0.0000000000
3	0.02	0.0000000017	0.0000035031	0.0000000000
4	0.03	0.0000000025	0.0000038747	0.0000000000
5	0.04	0.0000000033	0.0000041732	0.0000000000
6	0.05	0.0000000042	0.0000044463	0.0000000000
7	0.06	0.0000000050	0.0000047198	0.0000000000
8	0.07	0.0000000059	0.0000050103	0.0000000000
9	0.08	0.0000000067	0.0000053308	0.0000000001
10	0.09	0.0000000076	0.0000056921	0.0000000001
11	0.10	0.0000000084	0.0000061039	0.0000000001

with

$$
\begin{aligned}
k_1 &= \hat{h}y_i,\\
k_2 &= \hat{h}\left(y_i + \frac{1}{2}k_1\right) = \hat{h}y_i + \frac{1}{2}\hat{h}^2 y_i,\\
k_3 &= \hat{h}\left(y_i + \frac{1}{2}k_2\right) = \hat{h}y_i + \frac{1}{2}\hat{h}^2 y_i + \frac{1}{4}\hat{h}^3 y_i,\\
k_4 &= \hat{h}(y_i + k_3) = \hat{h}y_i + \hat{h}^2 y_i + \frac{1}{2}\hat{h}^3 y_i + \frac{1}{4}\hat{h}^4 y_i,\\
k_5 &= \hat{h}\left(y_i + \frac{1}{2}k_1 + \frac{1}{2}k_4\right) = \hat{h}y_i + \hat{h}^2 y_i + \frac{1}{2}\hat{h}^3 y_i\\
&\quad + \frac{1}{4}\hat{h}^4 y_i + \frac{1}{8}\hat{h}^5 y_i.
\end{aligned}
$$

The absolute stability region is defined by the condition $|R(z)| < 1$, where $R(z)$ denotes the stability function, as illustrated in Figure 6. The stability characteristics of the method are further analyzed through the characteristic polynomial, given by $P(z) = z - R(z)$, where $z = \hat{h}$.

4. CONCLUSIONS

This study presents a novel and simplified fifth-order Runge-Kutta (RK5) formula for solving initial value problems (IVPs) in ordinary differential equations (ODEs), derived through optimized coefficient selection based on a Taylor series expansion. The proposed method ensures high accuracy while reducing computational complexity. Numerical experiments demonstrate that the proposed formula outperforms existing RK5 methods in terms of accuracy, execution time, and stability. Error analysis confirms that it maintains fifth-order accuracy while producing lower numerical errors. Stability analysis, evaluated via the stability function, reveals a comparable or improved stability region relative to existing RK5 methods. The proposed method performs especially well in complex differential equations, benefiting from its streamlined coefficient structure. Its application to systems of ODEs, such as epidemiological models, further highlights its effectiveness in handling coupled equations with variable parameters. While the method shows clear advantages, further evaluation is warranted for highly stiff problems. Incorporating adaptive step-size control may enhance its computational efficiency. Overall, the proposed RK5 formula provides a more efficient and accurate alternative for solving IVPs in ODEs, with promising extensions to adaptive Runge-Kutta methods, high-dimensional dynamical systems, and machine learning-integrated numerical solvers.

5. ACKNOWLEDGEMENT

The authors would like to express their gratitude to Prof. Dr. Marjono, MPhil. for valuable comments on the research, and Pusat Publikasi Ilmiah dan Ketahanan Jurnal (PPIKJ) of Brawijaya University.

REFERENCES

Agbeboh, G. U., L. O. Adoghe, and M. E. Ehiemua (2020). On the Derivation of a Sixth-Stage-Fifth-Order Runge-Kutta Method for Solving Initial Value Problems in Ordinary Differential Equations. *American Journal of Sciences and Engineering Research*, **3**(5); 29–41

Aliyi, K. and H. Muleta (2021). Numerical Method of the Line for Solving One Dimensional Initial-Boundary Singularly Perturbed Burger Equation. *Indian Journal of Advanced Mathematics*, **1**(2); 4–14

Audu, K. J., A. R. Taiwo, and A. A. Soliu (2023). Assessment of Numerical Performance of Some Runge-Kutta Methods and New Iteration Method on First Order Differential Problems. *Dutse Journal of Pure and Applied Sciences*, **9**(4a); 58–70

Bosede, O. R., F. S. Emmanuel, and O. J. Temitayo (2012). On Some Numerical Methods for Solving Initial Value Problems in Ordinary Differential Equations. *IOSR Journal of Mathematics (IOSRJM)*, **1**(3); 25–31

Chandru, M., R. Ponalagusamy, and P. J. A. Alphonse (2017). A New Fifth-Order Weighted Runge-Kutta Algorithm Based on Heronian Mean for Initial Value Problems in Ordinary Differential Equations. *Journal of Applied Mathematics & Informatics*, **35**(1–2); 191–204

Chapra, S. C. and R. P. Canale (2015). *Numerical Methods for Engineers*. McGraw-Hill Education

Demba, M. A., N. Senu, and F. Ismail (2016). A 5(4) Embedded Pair of Explicit Trigonometrically-Fitted Runge-Kutta-Nyström Methods for the Numerical Solution of Oscillatory Initial Value Problems. *Mathematical and Computational Applications*, **21**(4); 1–14

Dennis, G. Z. (2012). *A First Course in Differential Equations with Modeling Applications*. Cengage Learning, Loyola Marymount University, 9th edition

Donas, A., I. Famelis, P. C. Chu, and G. Galanis (2021). Optimization of the Navy's Three-Dimensional Mine Impact Burial Prediction Simulation Model, Impact35, Using High-Order Numerical Methods. *Journal of Defense Modeling and Simulation*, **20**(2); 197–212

Dukkipati, R. V. (2010). *Numerical Methods*. New Age International Limited Publisher

Fardinah (2017). Solusi Persamaan Diferensial Biasa dengan Metode Runge-Kutta Orde Lima. *Jurnal Matematika Dan Statistika Serta Applikasinya*, **5**(1); 30–36 (in Indonesia)

Ghazal, Z. K. and K. A. Hussain (2021). Solving Oscillating Problems Using Modifying Runge-Kutta Methods. *Ibn AL-Haitham Journal For Pure and Applied Sciences*, **34**(4); 58–67

Goeken, D. and O. Johnson (1999). Fifth-Order Runge-Kutta with Higher Order Derivative Approximations. In *Electronic Journal of Differential Equations*, volume 02. University of Central Oklahoma

Gumus, O. A., Q. Cui, G. M. Selvam, and A. Vianny (2022). Global Stability and Bifurcation Analysis of a Discrete Time SIR Epidemic Model. *Miskolc Mathematical Notes*, **23**(1); 193–210

Hossain, M. B., M. J. Hosain, M. M. Miah, and M. S. Alam (2017). A Comparative Study on Fourth Order and Butcher's Fifth Order Runge-Kutta Methods with Third Order Initial Value Problem (IVP). *Applied and Computational Mathematics*, **6**(6); 243–253

Hussain, K. A. and W. J. Hasan (2023). Improved Runge-Kutta Method for Oscillatory Problem Solution Using Trigonometric Fitting Approach. *Ibn AL-Haitham Journal For Pure and Applied Sciences*, **36**(1); 345–354

Hussain, K. A., F. Ismail, N. Senu, F. Rabiei, and R. Ibrahim (2015). Integration for Special Third-Order Ordinary Differential Equations Using Improved Runge-Kutta Direct Method. *Malaysian Journal of Science*, **34**(2); 172–179

Kafle, J., B. K. Thakur, and I. B. Bhandari (2021). Application of Numerical Methods for the Analysis of Damped Parallel RLC Circuit. *Journal of Institute of Science and Technology*, **26**(1); 28–34

Kalogiratou, Z., T. Monovasilis, G. Psihoyios, and T. E. Simos (2014). Runge-Kutta Type Methods with Special Properties for the Numerical Integration of Ordinary Differential Equations. *Physics Reports*, **536**(3); 75–146

Kennedy, C. A. and M. H. Carpenter (2019). Higher-Order Additive Runge–Kutta Schemes for Ordinary Differential Equations. *Applied Numerical Mathematics*, **136**; 183–205

Kopecz, S. and A. Meister (2018). On Order Conditions for Modified Patankar–Runge–Kutta Schemes. *Applied Numerical Mathematics*, **123**; 159–179

Mathews, J. H. and K. D. Fink (2004). *Numerical Methods Using MATLAB*. Prentice-Hall, Inc., 3rd edition

Mechee, M., N. Senu, F. Ismail, B. Nikouravan, and Z. Siri (2013). A Three-Stage Fifth-Order Runge-Kutta Method for Directly Solving Special Third-Order Differential Equation with Application to Thin Film Flow Problem. *Mathematical Problems in Engineering*, **2013**; 1–9

Monovasilis, T., Z. Kalogiratou, and T. E. Simos (2016). Construction of Exponentially Fitted Symplectic Runge–Kutta–Nyström Methods from Partitioned Runge–Kutta Methods. *Mediterranean Journal of Mathematics*, **13**(4); 2271–2285

Monovasilis, T., Z. Kalogiratou, and T. E. Simos (2018). Trigonometrical Fitting Conditions for Two Derivative Runge-Kutta Methods. *Numerical Algorithms*, **79**(3); 787–800

Parhusip, H. A., S. Trihandaru, B. Aryo, A. Wicaksono, D. Indrajaya, Y. Sardjono, and O. P. Vyas (2022). Susceptible

Vaccine Infected Removed (SVIR) Model for COVID-19 Cases in Indonesia. *Science and Technology Indonesia*, **7**(3); 400–408

Rabiei, F. and F. Ismail (2012). Fifth-Order Improved Runge-Kutta Method With Reduced Number of Function Evaluations. *Australian Journal of Basic and Applied Sciences*, **6**(3); 97–105

Ram, M. (2020). *Recent Advances in Mathematics for Engineering*. CRC Press, 1st edition

Sukron, M., U. Habibah, and N. Hidayat (2021). Numerical Solution of Saint-Venant Equation Using Runge-Kutta Fourth-Order Method. *Journal of Physics: Conference Series*, **1872**(1); 012036

Suryaningrat, W., R. Ashgi, and S. Purwani (2020). Order Runge-Kutta with Extended Formulation for Solving Ordinary Differential Equations. *International Journal of Global Operations Research*, **1**(4); 160–167

