Applied Mathematics and Computational Intelligence
Volume 15, No. 3, 2026 [1 − 21]

Numerical Solutions of Higher-Order Ordinary Differential Equations Using Runge–Kutta Methods with Extrapolation Techniques

Fatin Nor Athirah Muhamad[1], Azhar Ahmad[2], Nur Izzati Khairudin[3], Nurul Ainina Redwan[4*]

[1,4]Center of Mathematical Sciences, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Cawangan Terengganu Kampus Kuala Terengganu, 21080 Kuala Terengganu, Terengganu, Malaysia
[2]Department of Computer Science and Mathematics, Universiti Teknologi MARA Cawangan Pulau Pinang Kampus Permatang Pauh, 13500 Permatang Pauh, Penang, Malaysia
[3]Center of Mathematical Sciences, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Cawangan Perlis Kampus Arau, 02600 Arau, Perlis, Malaysia

* Corresponding author: ainina@uitm.edu.my

Received: 5 August 2025
Revised: 21 September 2025
Accepted: 26 September 2025

ABSTRACT

Higher-order ordinary differential equations (ODEs) play a central role in modeling complex dynamical systems, but their numerical solution remains challenging in terms of accuracy and computational efficiency. This study systematically investigates the performance of classical numerical methods—Euler's method, Heun's method, fourth-order Runge–Kutta (RK4), fifth-order Runge–Kutta (RK5), and the Runge–Kutta–Fehlberg method (RK45)—for solving fourth- and fifth-order homogeneous and non-homogeneous ODEs. To enhance precision, we integrate Richardson's and Aitken's extrapolation techniques with these methods and assess their impact on accuracy and efficiency. The novelty of this work lies in providing a fixed-step, systematic comparison of classical Runge–Kutta methods combined with extrapolation, an area that has not been sufficiently explored for higher-order ODEs. Approximate solutions are tabulated and visualized, with absolute errors and CPU times analyzed to evaluate trade-offs between accuracy and efficiency. The results show that RK5 with Aitken's extrapolation achieves the highest accuracy for homogeneous problems, while RK45 with Aitken's extrapolation is most effective for non-homogeneous cases. These findings highlight the practical value of extrapolation techniques in refining classical methods for advanced ODE problems.

Keywords: Aitken's Extrapolation, Euler's Method, Heun's Method, Richardson's Extrapolation, Runge-Kutta-Fehlberg, Runge-Kutta Fifth-Order, Runge-Kutta Fourth-Order

1 NOMENCLATURE

Abbreviations

EM	Euler's Method
HM	Heun's Method
RK	Runge–Kutta Method
RK4	Fourth-Order Runge–Kutta Method
RK5	Fifth-Order Runge–Kutta Method
RK45	Runge–Kutta–Fehlberg Method
AE	Aitken's Extrapolation
RE	Richardson's Extrapolation
ODE	Ordinary Differential Equation
CPU	Central Processing Unit

Variables and Parameters

h	Step size
t	Independent variable (time)
y	Exact solution of the IVP
y_n	Numerical solution at step n
y_{n+1}	Numerical solution at step $n+1$
$f(t,y)$	Right-hand side function of the ODE, $\frac{dy}{dt} = f(t,y)$
y_0	Initial value of the dependent variable
n	Step index (integer counter)

Error Notation

e_n	Local error at step n
E	Global error over the integration interval
$\lVert\cdot\rVert$	Norm (used for error measurement)

2 INTRODUCTION

Differential equations are highly useful in solving complex mathematical problems across various fields of engineering, science, and mathematics. Many real-world problems can be expressed in the form of differential equations, which can either be ordinary or partial differential equations. However, these problems are often so complex that finding the exact solution is challenging. The approximate solutions are typically found using one of two main approaches to address this. The first method simplifies differential equations into a form that can be solved exactly, and this solution is used to estimate the original problem. The second method uses approximation techniques, which provide more accurate results with smaller errors. Since many differential equations are too complex for exact solutions, numerical methods are often used, as most equations require numerical techniques under specific initial conditions to find solutions.

First-order differential equations involve only the first derivative, while higher-order equations include second or higher derivatives, describing systems with varying rates of change. These equations capture complex dynamics and are fundamental in understanding time-dependent behaviors. This research focuses on enhancing numerical methods for higher-order ordinary differential equations (ODEs), comparing the performance of Euler's Method (EM), Heun's Method (HM), fourth-order Runge–Kutta method (RK4), fifth-order Runge–Kutta method (RK5) and Runge–Kutta–Fehlberg method (RK45). Additionally, improving accuracy through Aitken's extrapolation technique (AE) and Richardson's extrapolation technique (RE) is emphasized. These approaches aim to refine the precision of numerical solutions and advance their application in solving real-world problems.

Numerical methods play a vital role in solving ODE which frequently arise in engineering, physics and applied sciences. However, the efficiency and accuracy of these methods can vary significantly depending on the type of equation, whether it is homogeneous or non-homogeneous equation. While methods such as EM, RK4, RK5 and RK45 along with accuracy improving techniques like RE and AE are widely used, its remains unclear which approach consistently provides the best balance between accuracy and computational efficiency across diverse problem types. This gap hinders the ability of researchers and practitioners to select the most suitable method for solving complex real-world problems that require high precision, especially for systems with high stability requirements. Addressing this challenge requires a comprehensive analysis and comparison of these numerical methods under varying conditions to identify the optimal solution for different classes of ODE.

2.1 Overview

Higher-order numerical methods are necessary for solving Initial Value Problems (IVP) involving ODEs particularly when accuracy and efficiency are critical. These methods address key challenges such as computational cost, complexity and error propagation thus providing robust solutions for a variety of scientific and engineering applications [1].

Earlier methods which are EM introduced by Leonhard Euler in 1768 [2], provide foundational insights but often yield less accurate results due to their reliance on local linearity for approximations [3]. The Modified Euler's or Heun's Method, introduced by Karl Heun in 1889, improves accuracy by incorporating additional evaluations of the solution curve [2]. While these foundational methods are simpler, they serve as the basis for the development of more sophisticated techniques.

The Runge-Kutta method builds upon these foundational approaches, offering a more generalized framework for solving IVPs involving ODEs. Initially introduced by Carl David Tolmé Runge in 1895 and later refined by Martin Wilhelm Kutta in 1901, these methods are designed to overcome the limitations of simpler techniques like EM and HM [4]. Moreover, the study comparison between RK4 and EM shows that RK4 is more accurate, consistent, stable, and convergent, making it the preferred method for solving IVP in ODE [5]. Other than that, a comparative analysis of RK4 and Butcher RK5 for solving first-order linear ODE systems, results revealed that Butcher RK5 exhibited greater accuracy and faster convergence rates than RK4 [6]. Further advancements include adaptive methods such as the Runge-Kutta-Fehlberg (RK45) method, introduced in 1969. By incorporating variable step size control, RK45 optimizes performance for problems with varying solution behavior, reducing computational overhead without compromising accuracy [7, 8]. One of the research projects concluded that the RK45 method is the most accurate and efficient choice for solving IVP in ODE, outperforming EM, HM and RK4, it is easily adaptable in MATLAB [9].

The research evaluates numerical methods for solving second-order ODEs, comparing the accuracy and efficiency of EM, HM, RK3, RK4 and Butcher RK5 methods under varying step sizes, with results indicating that Butcher RK5 method is the most accurate, while EM is the fastest. These higher-order methods represent significant progress in addressing the limitations of earlier numerical approaches.

Complementary to these methods, extrapolation techniques like RE and AE refine numerical solutions by improving convergence rates. RE is particularly effective in computational fluid dynamics, where precision is essential [10], while AE accelerates convergence in iterative schemes, such as the Gauss-Seidel method, enhancing computational efficiency [11]. The study develops a stable and consistent higher-order fitted operator finite difference method with RE, achieving fourth-order convergence and more accurate numerical solutions for singularly perturbed parabolic convection-diffusion problems compared to standard finite difference and upwind schemes [12]. Then, the research about estimating implied volatility using the Newton-Raphson, secant, and accelerated secant methods with AE concludes that the secant method accelerated with AE outperforms the Newton-Raphson method in both accuracy and convergence rate [13]. Together, these numerical approaches demonstrate the evolution of methods designed to solve ODEs with higher accuracy, efficiency and adaptability.

Recent studies have explored advanced extrapolation strategies and high-order ODE solvers with an emphasis on adaptivity, symmetry and hybrid approaches. [14] introduced extrapolation semi-implicit multistep methods with adaptive step-size control using novel local error estimators, achieving improved accuracy–efficiency trade-offs. [15] extended extrapolation techniques to semilinear fractional differential equations, providing rigorous error bounds and convergence analysis. [16] proposed a semi-implicit solver with adjustable symmetry and variable step sizes, demonstrating the impact of symmetry on solver performance in chaotic dynamics. In a different application domain, [17] applied extrapolation with multiple ODE solutions to accelerate diffusion model sampling, showing the continued relevance of extrapolation beyond traditional scientific computing. Finally, [18] analyzed global Richardson extrapolation for linear multistep methods, establishing theoretical guarantees on order improvement and stability. These works highlight ongoing interest in adaptivity, symmetry, and error control, but none provide a systematic, fixed-step comparison of classical one-step methods combined with extrapolation for higher-order ODEs, which is the focus of the present study.

The higher-order ODE has become essential tools in addressing complex real-world problems in scientific and engineering domains. The study by [19] focused on solving second-order differential equations derived from Newton's second law to model car suspension dynamics, using higher-order numerical methods, HM and RK4. Next, [20] modeled beam deflection using a fourth-order ODE, which was then converted into a system of first-order ODEs, with numerical solutions obtained via the RK4 method and expressed in terms of elliptical functions.

The present work advances the field by systematically investigating the role of extrapolation in improving numerical methods for higher-order ODEs. The novelty of this study lies in extending Richardson's and Aitken's extrapolation techniques to classical one-step schemes such as EM, HM, RK4, RK5, and RK45 when applied to fourth- and fifth-order equations. While extrapolation has been widely employed for lower-order or specialized problems, its systematic integration with these well-established methods for higher-order ODEs has not been adequately addressed. Through a rigorous comparative analysis that includes error quantification and CPU time evaluation, this study demonstrates how extrapolation enhances both accuracy and computational efficiency. By filling this gap, the work provides a clearer framework for selecting optimal numerical strategies in solving higher-order ODEs under different conditions.

3 METHODOLOGY

3.1 Basic concept for higher-order linear differential equations

The general nth-order linear differential equation is:

$$a_n(x)y^n + a_{n-1}y^{n-1} + \cdots + a_1(x)y' + a_0(x)y = g(x), \tag{1}$$

where $a_0(x), \ldots, a_n(x)$ and $g(x)$ can be zero or non-zero function, constant or non-constant function, linear or non-linear function. Equation (1) is considered homogeneous if $g(x) = 0$; otherwise, if $g(x) \neq 0$, then it is non-homogeneous. Then, IVP for (1) is

$$y(x_0) = y_0, y'(x_0) = y_1, \ldots, y^n(x_0) = y_{n-1}. \tag{2}$$

3.2 Solving higher-order ODE

Consider the nth-order linear differential equation (1) with $n-1$ initial conditions. It can be solved by assuming a solution of the form

$$\begin{aligned}
y &= z_1,\\
\frac{dy}{dx} &= \frac{dz_1}{dx} = z_2,\\
\frac{d^2y}{dx^2} &= \frac{dz_2}{dx} = z_3\\
\frac{d^{n-1}y}{dx^{n-1}} &= \frac{dz_{n-1}}{dx} = z_n,\\
\frac{d^ny}{dx^n} &= \frac{dz_n}{dx} = \frac{1}{a_n}\big(- a_{n-1}\frac{d^{n-1}y}{dx^{n-1}} - \cdots - a_1\frac{dy}{dx} - a_0y + g(x)\big),\\
&= \frac{dz_n}{dx} = \frac{1}{a_n}\big(- a_{n-1}z_n - \cdots - a_1z_2 - a_0z_1 + g(x)\big).
\end{aligned} \tag{3}$$

Each of the nth first-order ODEs are accompanied by one initial condition. These equations are simultaneous in nature but can be solved using the methods typically applied to first-order ODEs.

3.3 Numerical Methods

3.3.1 Euler's Method

The simplest Euler scheme to advance a solution from one computational solution point (x_i, y_i) to another (x_{i+1}, y_{i+1}) is to consider a step size, h in x such that $x_{i+1} = x_{i+h}$, for all i and an approximate derivative in the EM by a simple forward difference approximation given by

$$\frac{\Delta y}{h} \approx f(x, y). \tag{4}$$

Considering the points i and $i + 1$, the expression becomes

$$\frac{y_{i+1} - y_i}{h} \approx f(x_i, y_i). \tag{5}$$

This gives the explicit solution for the new value of y_{i+1} form EM,

$$y_{i+1} \approx y_i + hf(x_i, y_i). \tag{6}$$

3.3.2 Heun's Method (HM)

HM is also known as the second-order RK method. The first-order ODE is given by

$$\frac{dy}{dx} = f(x, y(x)), \tag{7}$$

with the initial value

$$y(x_0) = y_0. \tag{8}$$

The general solution is

$$y_{i+1} = y_i + \frac{h}{2}(k_1 + K_2), \tag{9}$$

where

$$\begin{aligned} k_1 &= f(x_i, y_i), \\ k_2 &= f(x_i + h, y_i + hk_1). \end{aligned} \tag{10}$$

3.3.3 Fourth-Order Runge–Kutta Method (RK4)

Consider the IVP (7) and (8), the general solution using RK4 is given by

$$y_{i+1} \approx y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \tag{11}$$

where

$$\begin{aligned} k_1 &= f(x_i, y_i), \\ k_2 &= f(x_i + \frac{h}{2}, y_i + \frac{1}{2}k_1), \\ k_3 &= f(x_i + \frac{h}{2}, y_i + \frac{1}{2}k_2), \\ k_4 &= f(x_i + h, y_i + k_3). \end{aligned} \tag{12}$$

3.3.4 Fifth-Order Runge–Kutta Method (RK5)

Consider the IVP (7) and (8). Then, the method using RK5 involves calculating y_{k+1} are as follows

$$y_{k+1} = y_k + \frac{h}{90}(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6), \tag{13}$$

where

$$\begin{aligned}
x_{k+1} &= x_k + h,\\
k_1 &= f(x_k, y_k),\\
k_2 &= f(x_k + \frac{h}{4}y_k, y_k + \frac{1}{4}k_1 h),\\
k_3 &= f(x_k + \frac{h}{4}, y_k + \frac{1}{8}k_1 h + \frac{1}{8}k_2 h),\\
k_4 &= f(x_k + \frac{h}{2}, y_k - \frac{1}{2}k_2 h + k_3 h),\\
k_5 &= f\big(x_k + \frac{3}{4}h, y_k + \frac{3}{16}k_1 h + \frac{9}{16}k_4 h\big),\\
k_6 &= f\big(x_k + h, y_k - \frac{3}{7}k_1 h + \frac{2}{7}k_2 h + \frac{12}{7}k_3 h - \frac{2}{7}k_4 h + \frac{8}{7}k_5 h\big).
\end{aligned} \tag{14}$$

3.3.5 Runge–Kutta–Fehlberg Method (RK45)

The Runge–Kutta–Fehlberd method (RK45) is described here. The general solution to the IVP (7) and (8) is given by

$$y_{i+1} \approx y_i + h\big(\frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4101}k_4 - \frac{1}{5}k_5\big), \tag{15}$$

$$z_{i+1} \approx y_i + h\big(\frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56420}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6\big), \tag{16}$$

where

$$\begin{aligned}
k_1 &= f(x_i, y_i),\\
k_2 &= f(x_i + \frac{h}{4}y_k, y_i + \frac{1}{4}k_1 h),\\
k_3 &= f(x_i + \frac{3}{8}h, y_i + \frac{3}{32}k_1 h + \frac{9}{32}k_2 h),\\
k_4 &= f(x_i + \frac{12}{13}h, y_i + \frac{1932}{2197}k_1 h - \frac{7200}{2197}k_2 h + \frac{7296}{2197}k_3 h),\\
k_5 &= f\big(x_i + \frac{12}{13}h, y_i + \frac{439}{216}k_1 h - 8hk_2 + + \frac{3680}{513}k_3 h - \frac{845}{4104}k_4 h\big),\\
k_6 &= f\big(x_i + \frac{1}{2}h, y_i - \frac{8}{27}k_1 h + 2k_2 h - \frac{3544}{2565}k_3 h + \frac{1859}{4104}k_4 h - \frac{11}{40}k_5 h\big).
\end{aligned} \tag{17}$$

In the error-control theory, an initial value of h at the ith steps is used to find the first values of (15) and (16). These are then employed to determine

$$\begin{aligned}\hat{q} &= (\frac{\varepsilon h}{2|y_{i+1} - z_{i+1}})^{1/4}, \\ &= 0.84(\frac{\epsilon}{R})^{1/4},\end{aligned} \tag{18}$$

where $\epsilon = 10^{-6}$ and the calculation is repeated. At the ith step, the value of $\hat{q}$ is applied as follows:

- If $R > \epsilon$, the current step size h is rejected and the computation is repeated with a reduced step size qh, and
- If $R \geq \epsilon$, the computed solution at the itth steps is accepted, but the step size adjusted to qh for the $(i+1)$st step.

The step-size adjustment procedure is therefore essential to maintain both stability and accuracy in the error-control scheme [21].

3.4 Extrapolation Techniques

3.4.1 Richardson's Extrapolation

Richardson's Extrapolation (RE) is a technique used to enhance accuracy. Consider the numerical value of a physical quantity ϕ obtained by using the computational step size δ, such as time or space steps, expressed as $\phi(\delta)$ (Fenton, 2010). If the computational scheme is of nth-order, and y_E represents the exact solution, it can be expressed as:

$$y_E = \phi(\delta) + a_n(\delta)^n + a_{n-1}\delta^{n-1} + \dots. \tag{19}$$

If the goal is to obtain numerical values δ_1 and δ_2, (19) needs to be written twice, as given by,

$$y_E = \phi(\delta_1) + a_n(\delta_1)^n + a_{n-1}\delta_1^{n-1} + \dots, \tag{20}$$

$$y_E = \phi(\delta_1) + a_n(\delta_2)^n + a_{n-1}\delta_2^{n-1} + \dots. \tag{21}$$

By neglecting the terms with $a_n, a_(n-1), \cdots$ in (20)-(21), and then multiplying (20) by q^n and subtracting (20) from (21), the resulting equation is,

$$y_E = \frac{\phi(\delta_2) - q^n\phi(\delta_1)}{1 - q^n}, \tag{22}$$

where $\delta_2/\delta_1)$ is the ratio of the δ. Feldman (2000) stated that for a simple EM with $n = 1$ and $q = 1/2$, equation (21) simplifies to,

$$y_E \approx 2\phi(\delta_2) - \phi(\delta_1). \tag{23}$$

This approach can be easily implemented, assigning twice as much weight to the solution with the halved time step. For the HM, when $n = 2$ and $q = 1/2$, equation (22) simplifies to,

$$y_E \approx \frac{1}{3}(4\phi(\delta_2) - \phi(\delta_1)). \tag{24}$$

If the RK4 method is applied, where $n = 4$ and $q = 1/2$ equation (22) simplifies to,

$$y_E \approx \frac{1}{15}(16\phi(\delta_2) - \phi(\delta_1)). \tag{25}$$

Then, RK5 and RK45 method, $n = 5$ and $q = 1/2$ equation (22) simplifies to,

$$y_E \approx \frac{1}{31}(32\phi(\delta_2) - \phi(\delta_1)). \tag{26}$$

3.4.2 Aitken's Extrapolation

The Aitken's Extrapolation (AE) technique can be employed without knowing the order n of the scheme (Saud, 2014). Since this introduces an additional unknown, a third computation step is required. Consequently, a third equation must be formulated alongside equations (20) and (21), such that,

$$y_E = \phi(\delta_3) + a_n\delta_3^n + a_{n-1}\delta_3^{n-1} + \cdots . \tag{27}$$

The step ratios are selected to remain consistent, $\delta_1/\delta_2 = \delta_2/\delta_3$, which can typically be arranged. Assume $\{\phi_1\}_{n=0}^{\infty}$ is a linearly convergent sequence with the limit y_E, for $n = 1, 2, 3$, the following expression is obtained,

$$\frac{|\phi_2 - y_E|}{|\phi_1 - y_E|} \approx \frac{|\phi_3 - y_E|}{|\phi_2 - y_E|}. \tag{28}$$

Rearranging equation (27) to isolate y_E as the subject results in

$$y_E = \frac{\phi_1\phi_3 - \phi_2^2}{\phi_1 + \phi_3 - 2\phi_2}. \tag{29}$$

The calculation can become poorly conditioned when both the numerator and denominator of the fractional part approach zero (Saud, 2014). To address this, the solution (29) is rewritten as a correction to the most refined estimate, ϕ_3 as follows

$$y_E = \phi_3 - \frac{(\phi_3 - \phi_2)^2}{\phi_1 + \phi_3 - 2\phi_2}. \tag{30}$$

The convergence behavior of these numerical schemes, along with the approach for estimating local and global errors, is summarized in the following subsection.

3.5 Convergence and Error Estimation

The convergence of a numerical method requires both consistency and stability. Each base method employed in this study has a known local truncation error of order $O(h^{p+1})$, where p denotes the method's order. As the step size $h \to 0$, the local error tends to zero, ensuring consistency. Stability is preserved for bounded exact solutions when a sufficiently small step size is used, and is naturally incorporated in the fixed-step approach adopted here, with the exception of RK45 which is adaptive.

The order of accuracy, or rate of convergence, is method dependent: EM is first-order, HM second-order, RK4 fourth-order, RK5 fifth-order, and RK45 an embedded fourth- and fifth-order scheme. Extrapolation techniques (Richardson and Aitken) further enhance convergence by cancelling the leading error terms, thereby increasing the effective accuracy of the base methods.

Error estimation in this work is carried out by computing both local error at each step,

$$e_n = |y(x_n) - y_n|, \tag{31}$$

and global error across the integration interval,

$$E = \max_n |y(x_n) - y_n|. \tag{32}$$

Convergence is confirmed when the global error decreases at the expected theoretical rate as the step size is refined.

4 IMPLEMENTATION

Higher-order ODEs can be solved using the same numerical methods applied to first-order ODEs. The general form of a fourth-order differential equation is

$$a_4 y^{(4)} + a_3 y^{'''} + a_2 y^{''} + a_1 y^{'} + a_0 y = g(x), \tag{33}$$

with initial conditions at $x = x_0$ given by

$$y = b_0, \quad y' = b_1, \quad y^{''} = b_2 \quad \text{and} \quad y^{'''} = b_3. \tag{34}$$

Equation (33) is reduced to a system of four first-order ODEs as follows:

$$\begin{aligned} y' &= z, \\ y'' &= z' = w, \\ y''' &= z'' = r, \\ y^{(4)} &= z''' = \frac{1}{a_4}(-a_3 y''' - a_2 y'' - a_1 y' - a_0 y), \\ &= \frac{1}{a_4}(-a_3 r - a_2 w - a_1 z - a_0 y). \end{aligned} \tag{35}$$

4.1 Solving Fourth-Order Ordinary Differential Equations

Consider a fourth-order ODE

$$y^{(4)} - 3y'' + 2y = 0, \tag{36}$$

subject to the initial conditions at $x = 0$ given by

$$y = 1, \quad y' = 0, \quad y^{''} = 2 \quad \text{and} \quad y^{'''} = -1. \tag{37}$$

The equation (37) will be solved analytically and numerically using the EM method and a combination of EM with RE and AE.

4.1.1 Analytical Solution

The analytical solution is obtained by writing (36) as the characteristic polynomial. Let

$$y = e^{mx}, \quad y' = me^{mx}, \quad y^{''} = m^2e^{mx} \quad y^{'''} = m^3e^{mx}\text{and} \quad y^{(4)} = m^4e^{mx}. \tag{38}$$

Substituting (38) into (36) yields $(m^2 - 2)(m^2 - 1) = 0$. Therefore, the roots are

$$m_1 = 1, \quad m_2 = -1, \quad m_3 = \sqrt{2}, \quad \text{and} \quad m_4 = -\sqrt{2}. \tag{39}$$

where the roots are not repeated. Hence, the general solution is given by

$$y = C_1e^x + C_2e^{-x} + C_3e^{\sqrt{2}x} + C_4e^{-\sqrt{2}x}, \tag{40}$$

where C_1, C_2, C_3 and C_4 are arbitrary constants. Differentiating (40) yields

$$\begin{aligned} y' &= C_1e^x - C_2e^{-x} + \sqrt{2}C_3e^{\sqrt{2}x} - \sqrt{2}C_4e^{-\sqrt{2}x}, \\ y'' &= C_1e^x + C_2e^{-x} + 2C_3e^{\sqrt{2}} + 2C_4e^{-\sqrt{2}x}, \\ y''' &= C_1e^x - C_2e^{-x} + 2\sqrt{2}C_3e^{\sqrt{2}} - 2\sqrt{2}C_4e^{-\sqrt{2}x}. \end{aligned} \tag{41}$$

Based on the initial condition (37), the values for arbitrary constants are

$$C_1 = C_2 = \frac{1}{2}, \quad C_3 = \frac{2^{0.5}}{4} - \frac{1}{2} \quad \text{and} \quad C_4 = \frac{2^{0.5}(2^{0.5} + 1)}{4}, \tag{42}$$

and the particular solution is

$$y = \frac{e^x - e^{-x}}{2} - e^{(2^{0.5x})}\left(\frac{2^{0.5}}{4} - \frac{1}{2}\right) + \frac{2^{0.5}e^{(-2^{0.5x})}(2^{0.5} + 1)}{4}. \tag{43}$$

To find $y(0.75)$, substituting $x = 0.75$ into (43) and obtain

$$y(0.75) = 1.591334. \tag{44}$$

4.1.2 Numerical Method

In solving the fourth-order ODEs (36) subject to the initial conditions (37) using the EM method, (36) is reduced into four simultaneous first-order differential equations. Then, substituting (35) into (36) to give

$$\begin{aligned} y' &= f_1(x, y, z, w, r) = z, \\ y'' &= f_2(x, y, z, w, r) = w, \\ y''' &= f_1(x, y, z, w, r) = r, \\ y^{(4)} &= f_4(x, y, z, w, r) = 3w - 2y, \end{aligned} \tag{45}$$

and the initial value (37) can be written as

$$y(0) = y_0 = 1, \quad z(0) = z_0 = 0, \quad w(0) = w_0 = 2, \quad \text{and} \quad r(0) = r_0 = -1. \tag{46}$$

Table 1 : Approximate solution using EM for $h = 0.25$

Iteration, x_i	y_{i+1}	z_{i+1}	w_{i+1}	r_{i+1}
0	1.000000	0.000000	2.000000	-1.000000
0.25	1.000000	0.500000	1.750000	0.000000
0.50	1.125000	0.937500	1.750000	0.812500
0.75	1.359375	1.375000	1.953125	1.562500

The result obtains for step size $h = 0.25$ was summarized in Table 1.

From the Table 1, the approximate solution at, $x = 0.75$ is $y(0.75)_{EM} \approx 1.359375$. From (44) the exact solution is $y(0.75)_{exact} \approx 1.591334$. Then calculate the relative error based on the formula for relative error

$$\begin{aligned} \text{relative error} &= \left|\frac{\text{exact-approximate}}{\text{exact}}\right|, \\ &= \left|\frac{1.591334 - 1.359375}{1.591334}\right|, \\ &= 0.145764. \end{aligned} \tag{47}$$

The relative error with EM is considered bad because it is a little bit high. This can be improved by combining EM with RE, where RE requires an approximate solution using two different step sizes. Assume the ratio between the two-step sizes is $q = \delta_2/\delta_1 = 1/2$ with $n = 1$.

Next, consider another step size, $h = \delta_2 = 0.125$. The process to compute $y(0.75)$ is like solving a problem with $h = 0.25$, but it requires 6 iterations to calculate $y(0.75)$. The results for $h = 0.125$ are presented in Table 2. From Table 2, the approximate solution for, $h = 0.125$ is, $y(0.750)_{EM} \approx 1.443817$. Let $\phi(\delta_1)$ represent the approximate solution for $h = 0.25$ and $\phi(\delta_2)$ for $h = 0.125$. Therefore,

$$\phi(\delta_1) = 1.359375, \tag{48}$$

$$\phi(\delta_2) = 1.443817. \tag{49}$$

Substituting values (48) and (49) into equation (23) gives the following results:

$$y(0.75)_{EM+RE} \approx 1.528259. \tag{50}$$

The relative error for the combination of the method with the exact solution is

$$\text{relative error} = \left|\frac{1.591334 - 1.528259}{1.591334}\right| = 0.039637. \tag{51}$$

The result (51) demonstrate that integrating the EM with RE effectively reduces the relative error compare to EM alone. In enhancing further accuracy, the EM can be combined with AE, which requires three different step sizes.

Table 2 : Approximate solution using EM for $h = 0.125$

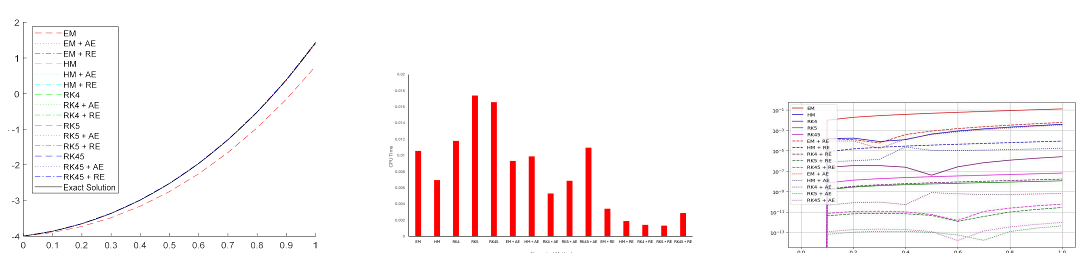
Iteration, x_i	y_{i+1}	z_{i+1}	w_{i+1}	r_{i+1}
0	1.000000	0.000000	2.000000	-1.000000
0.125	1.000000	0.250000	1.875000	-0.500000
0.250	1.031250	0.484375	1.812500	-0.046875
0.375	1.091797	0.710938	1.806641	0.375000
0.500	1.180664	0.936768	1.853516	0.779541
0.625	1.297760	1.168457	1.950958	1.179443
0.750	1.443817	1.412327	2.098389	1.586613

5 RESULT AND DISCUSSION

We consider homogeneous and non-homogeneous equations of the fourth-order ODE. A step size, $h = \delta_1 = 0.1$ was chosen to solve the problem. For the combination with RE, half of the initial step size, $h = \delta_2 = 0.05$, was used. An additional step size, $h = \delta_3 = 0.025$, was applied for the combination with AE. Therefore, all the numerical results have been tested using a computer with an Intel(R) Celeron(R) N4020 CPU @ 1.10GHz and 8GB RAM.

5.1 Homogeneous Equation

The exact and approximate solutions of the homogeneous equation (36) with the initial condition (37) for each method are tabulated in Table 3, 4 and 5, while the errors for each method are in Table 6, 7 and 8. The highest error is observed in EM with the error of 1.27E-01, as clearly illustrated in the approximation graph in Figure 6. As illustrated in Figure 7 and 8, incorporating RE and AE across all methods significantly enhances accuracy, demonstrating the effectiveness of extrapolation techniques in error reduction, especially the AE. Notably, the combination of RK5 with AE achieves the lowest error, highlighting its superior accuracy in solving the problem. Additionally, as shown in Figure 1(c), RK5 has the highest CPU time, followed by RK45, while RK5 + AE and RK45 + AE are considered to exhibit low CPU times. Therefore, RK5 + AE is considered the best method for this equation, balancing accuracy and efficiency.



(a) (b) (c)

Figure 1 : (a) Approximate, (b) error and (c) CPU time graph for solving homogeneous equation (32) with the initial conditions (33)

.

Table 3 : Approximate solution for each method

x_i	**Exact**	**EM**	**HM**	**RK4**	**RK5**	**RK45**
0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
0.1	1.009850	1.000000	1.010000	1.009850	1.009850	1.009850
0.2	1.038926	1.020000	1.039100	1.038926	1.038926	1.038926
0.3	1.086797	1.059000	1.086878	1.086797	1.086797	1.086797
0.4	1.153387	1.116400	1.153266	1.153388	1.153387	1.153387
0.5	1.238966	1.191970	1.238532	1.238966	1.238966	1.238966
0.6	1.344142	1.285828	1.343282	1.344142	1.344142	1.344142
0.7	1.469863	1.398425	1.468458	1.469862	1.469863	1.469862
0.8	1.617425	1.530539	1.615345	1.617423	1.617425	1.617425
0.9	1.788486	1.683268	1.785584	1.788484	1.788486	1.788486
1.0	1.985086	1.858043	1.981195	1.985083	1.985086	1.985086

Table 4 : Approximate solution for combination with Richardson's method

x	**EM + RE**	**HM + RE**	**RK4 + RE**	**RK5 + RE**	**RK45 + RE**
0	1.000000	1.000000	1.000000	1.000000	1.000000
0.1	1.010000	1.009842	1.009850	1.009850	1.009850
0.2	1.039050	1.038911	1.038926	1.038926	1.038926
0.3	1.086739	1.086775	1.086797	1.086797	1.086797
0.4	1.153002	1.153359	1.153387	1.153387	1.153387
0.5	1.238109	1.238931	1.238966	1.238966	1.238966
0.6	1.342660	1.344098	1.344142	1.344142	1.344142
0.7	1.467588	1.469810	1.469863	1.469863	1.469863
0.8	1.614164	1.617362	1.617425	1.617425	1.617425
0.9	1.784012	1.788411	1.788486	1.788486	1.788486
1.0	1.979128	1.984996	1.985086	1.985086	1.985086

Table 5 : Approximate solution for combination with Aitken's method

x	**EM + AE**	**HM + AE**	**RK4 + AE**	**RK5 + AE**	**RK45 + AE**
0	1.000000	1.000000	1.000000	1.000000	1.000000
0.1	1.009762	1.009851	1.009850	1.009850	1.009850
0.2	1.038834	1.038927	1.038926	1.038926	1.038926
0.3	1.086780	1.086796	1.086797	1.086797	1.086797
0.4	1.153527	1.153411	1.153387	1.153387	1.153387
0.5	1.239350	1.238977	1.238966	1.238966	1.238966
0.6	1.344865	1.344152	1.344142	1.344142	1.344142
0.7	1.471034	1.469874	1.469863	1.469863	1.469863
0.8	1.619169	1.617438	1.617425	1.617425	1.617425
0.9	1.790950	1.788501	1.788486	1.788486	1.788486
1.0	1.988443	1.985104	1.985086	1.985086	1.985086

Table 6 : Absolute error for each method

x	**EM**	**HM**	**RK4**	**RK5**	**RK45**
0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.1	9.85E-03	1.50E-04	2.39E-07	1.52E-09	7.13E-09
0.2	1.89E-02	1.74E-04	3.52E-07	2.77E-09	1.32E-08
0.3	2.78E-02	8.14E-05	3.51E-07	3.83E-09	1.86E-08
0.4	3.70E-02	1.22E-04	2.46E-07	4.76E-09	2.36E-08
0.5	4.70E-02	4.35E-04	4.05E-08	5.65E-09	2.87E-08
0.6	5.83E-02	8.60E-04	2.66E-07	6.54E-09	3.41E-08
0.7	7.14E-02	1.40E-03	6.76E-07	7.50E-09	4.02E-08
0.8	8.69E-02	2.08E-03	1.20E-06	8.60E-09	4.72E-08
0.9	1.05E-01	2.90E-03	1.85E-06	9.88E-09	5.56E-08
1.0	1.27E-01	3.89E-03	2.63E-06	1.14E-08	6.56E-08

Table 7 : Absolute error for combination with Richardson's method

x	**EM + RE**	**HM + RE**	**RK4 + RE**	**RK5 + RE**	**RK45 + RE**
0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.1	1.50E-04	8.09E-06	1.80E-09	4.24E-12	7.28E-12
0.2	1.24E-04	1.53E-05	3.33E-09	6.65E-12	1.12E-11
0.3	5.81E-05	2.20E-05	4.70E-09	7.42E-12	1.21E-11
0.4	3.86E-04	2.87E-05	5.99E-09	6.72E-12	1.02E-11
0.5	8.58E-04	3.56E-05	7.30E-09	4.64E-12	5.66E-12
0.6	1.48E-03	4.33E-05	8.70E-09	1.20E-12	1.47E-12
0.7	2.27E-03	5.21E-05	1.03E-08	3.62E-12	1.13E-11
0.8	3.26E-03	6.25E-05	1.21E-08	9.90E-12	2.40E-11
0.9	4.47E-03	7.48E-05	1.43E-08	1.78E-11	3.99E-11
1.0	5.96E-03	8.96E-05	1.69E-08	2.74E-11	5.94E-11

Table 8 : Absolute error for combination with Aitken's method

x	**EM + AE**	**HM + AE**	**RK4 + AE**	**RK5 + AE**	**RK45 + AE**
0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.1	8.77E-05	8.40E-07	4.72E-11	6.44E-14	1.18E-13
0.2	9.24E-05	1.05E-06	8.03E-11	1.06E-13	1.86E-13
0.3	1.70E-05	1.41E-06	9.38E-11	1.26E-13	2.08E-13
0.4	1.40E-04	2.36E-05	5.33E-11	1.25E-13	1.87E-13
0.5	3.84E-04	1.08E-05	7.84E-10	1.02E-13	1.24E-13
0.6	7.24E-04	1.06E-05	6.15E-10	5.62E-14	1.58E-14
0.7	1.17E-03	1.15E-05	5.12E-10	1.69E-14	1.39E-13
0.8	1.74E-03	1.31E-05	5.31E-10	1.21E-13	3.46E-13
0.9	2.46E-03	1.51E-05	5.87E-10	2.60E-13	6.10E-13
1.0	3.36E-03	1.77E-05	6.67E-10	4.41E-13	9.39E-13

5.2 Non-Homogeneous Equation

Consider a non-homogeneous equation,

$$y^{(4)} + 2y''' - 2y' - y = 12xe^{2x} + 24e^{-2x} - 12\cos 2x, \tag{52}$$

with the initial condition,

$$y = -4, \quad y' = 1, \quad y'' = 7, \quad \text{and} \quad y''' = 5. \tag{53}$$

Table 7 shows the exact and approximate solutions for EM, HM, RK4, RK5 and RK45 in solving (52) subject to the initial condition (53). Table 8 shows the results of combining these five methods with AE and RE. The absolute errors for each method are tabulated in Tables 9 and 10.

Figure 2(a) shows that the EM method has a significant difference from the exact solution. In Table 10, RK45 + AE has the lowest error at 5.13E-12, while RK5 + AE is slightly higher at 7.01E-12, showing only a small difference in accuracy. This difference can be seen in Figure 2(b). However, Figure 2(c) shows that RK45 + AE has a higher CPU time compared to other methods. Meanwhile, RK5 + RE is considered good accuracy at 6.28E-10 and has the lowest CPU time, making it much more efficient than RK45 + AE. Still, RK45 + AE is chosen as the best method because of its higher accuracy.

Table 9 : Approximate solution for each method

x_i	**Exact**	**EM**	**HM**	**RK4**	**RK5**	**RK45**
0	-4.000000	-4.000000	-4.000000	-4.000000	-4.000000	-4.000000
0.1	-3.864168	-3.900000	-3.865000	-3.864167	-3.864168	-3.864168
0.2	-3.653368	-3.730000	-3.655000	-3.653367	-3.653368	-3.653368
0.3	-3.362692	-3.485000	-3.365081	-3.362693	-3.362692	-3.362692
0.4	-2.987183	-3.160000	-2.990358	-2.987186	-2.987183	-2.987183
0.5	-2.521554	-2.750115	-2.525684	-2.521560	-2.521554	-2.521554
0.6	-1.959751	-2.250457	-1.965197	-1.959761	-1.959752	-1.959751
0.7	-1.294371	-1.655833	-1.301739	-1.294384	-1.294371	-1.294371
0.8	-0.515922	-0.960274	-0.526121	-0.515940	-0.515923	-0.515922
0.9	0.388076	-0.156414	0.373776	0.388052	0.388075	0.388076
1.0	1.434201	0.765282	1.414089	1.434170	1.434200	1.434201

6 CONCLUSION

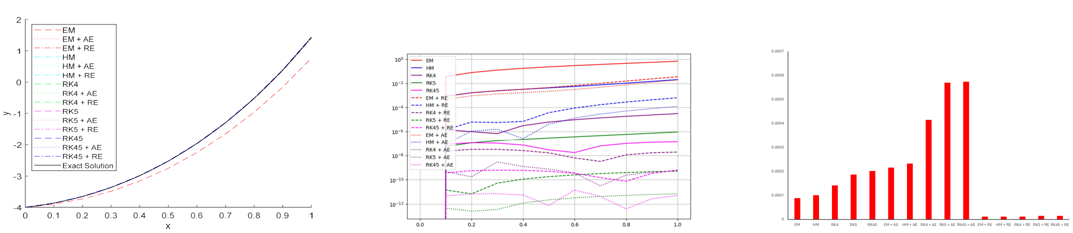
In conclusion, RK5 + AE and RK45 + AE demonstrate superior accuracy by achieving the lowest errors across various problems. While these methods require higher CPU time, accuracy takes priority in selecting the most reliable approach. Although RK5 + RE and RK45 + RE exhibit good efficiency, their accuracy is not as high as RK5 + AE and RK45 + AE. Among all methods, RK5 + AE stands out as the most robust for solving homogeneous equations, while RK45 + AE proves to be the best choice for non-homogeneous equations due to its lowest error. Both methods

Table 10 : Approximate solution of combination with Richardson's method

x_i	**EM + RE**	**HM + RE**	**RK4 + RE**	**RK5 + RE**	**RK45 + RE**	**RK45**
0	-4.000000	-4.000000	-4.000000	-4.000000	-4.000000	-4.000000
0.1	-3.865000	-3.864167	-3.864168	-3.864168	-3.864168	-3.864168
0.2	-3.655000	-3.653362	-3.653368	-3.653368	-3.653368	-3.653368
0.3	-3.365053	-3.362687	-3.362692	-3.362692	-3.362692	-3.362692
0.4	-2.990364	-2.987190	-2.987183	-2.987183	-2.987183	-2.987183
0.5	-2.525950	-2.521591	-2.521554	-2.521554	-2.521554	-2.521554
0.6	-1.966178	-1.959840	-1.959751	-1.959751	-1.959751	-1.959751
0.7	-1.304163	-1.294537	-1.294371	-1.294371	-1.294371	-1.294371
0.8	-0.531037	-0.516199	-0.515922	-0.515922	-0.515922	-0.515922
0.9	0.364941	0.387649	0.388076	0.388076	0.388076	0.388076
1.0	1.399458	1.433577	1.434201	1.434201	1.434201	1.434201

Table 11 : Approximate solution of combination with Aitken's method

x_i	**EM + AE**	**HM + AE**	**RK4 + AE**	**RK5 + AE**	**RK45 + AE**
0	-4.000000	-4.000000	-4.000000	-4.000000	-4.000000
0.1	-3.863704	-3.864168	-3.864168	-3.864168	-3.864168
0.2	-3.652450	-3.653369	-3.653368	-3.653368	-3.653368
0.3	-3.361399	-3.362694	-3.362692	-3.362692	-3.362692
0.4	-2.985526	-2.987183	-2.987183	-2.987183	-2.987183
0.5	-2.519387	-2.521550	-2.521554	-2.521554	-2.521554
0.6	-1.956691	-1.959738	-1.959751	-1.959751	-1.959751
0.7	-1.289713	-1.294343	-1.294371	-1.294371	-1.294371
0.8	-0.508527	-0.515872	-0.515922	-0.515922	-0.515922
0.9	0.399934	0.388156	0.388076	0.388076	0.388076
1.0	1.453020	1.434321	1.434201	1.434201	1.434201



(a) (b) (c)

Figure 2 : (a) Approximate, (b) error and (c) CPU time graph for solving non-homogeneous equation (48) with the initial conditions (49)

.

provide enhanced accuracy and stability, making them the preferred choices for solving higher-order ODEs. Overall, it can be concluded that AE is effective in improving accuracy.

Table 12 : Absolute error for each method

x_i	**EM**	**HM**	**RK4**	**RK5**	**RK45**
0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.1	3.58E-02	8.32E-04	1.41E-06	5.39E-08	8.19E-08
0.2	7.66E-02	1.63E-03	9.76E-07	1.07E-07	1.09E-07
0.3	1.22E-01	2.39E-03	6.65E-07	1.61E-07	1.01E-07
0.4	1.73E-01	3.18E-03	3.09E-06	2.19E-07	7.04E-08
0.5	2.29E-01	4.13E-03	6.07E-06	2.85E-07	2.84E-08
0.6	2.91E-01	5.45E-03	9.51E-06	3.61E-07	1.69E-08
0.7	3.61E-01	7.37E-03	1.35E-05	4.54E-07	5.97E-08
0.8	4.44E-01	1.02E-02	1.81E-05	5.71E-07	9.53E-08
0.9	5.44E-01	1.43E-02	2.37E-05	7.21E-07	1.20E-07
1.0	6.69E-01	2.01E-02	3.07E-05	9.13E-07	1.32E-07

Table 13 : Absolute error for combination with Richardson's

x_i	**EM + RE**	**HM + RE**	**RK4 + RE**	**RK5 + RE**	**RK45 + RE**
0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.1	8.32E-04	1.41E-06	2.23E-08	1.44E-11	3.57E-10
0.2	1.63E-03	6.13E-06	3.10E-08	6.85E-12	5.47E-10
0.3	2.36E-03	5.56E-06	3.07E-08	5.18E-11	6.11E-10
0.4	3.18E-03	6.87E-06	2.48E-08	1.11E-10	5.83E-10
0.5	4.40E-03	3.65E-05	1.59E-08	1.78E-10	4.87E-10
0.6	6.43E-03	8.81E-05	6.13E-09	2.48E-10	3.38E-10
0.7	9.79E-03	1.67E-04	3.17E-09	3.17E-10	1.49E-10
0.8	1.51E-02	2.77E-04	1.09E-08	3.84E-10	7.46E-11
0.9	2.31E-02	4.27E-04	1.62E-08	4.51E-10	3.33E-10
1.0	3.47E-02	6.24E-04	1.85E-08	5.19E-10	6.28E-10

Table 14 : Absolute error for combination with Aitken's method

x_i	**EM + AE**	**HM + AE**	**RK4 + AE**	**RK5 + AE**	**RK45 + AE**
0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.1	4.64E-04	9.23E-08	4.62E-10	4.61E-13	4.63E-12
0.2	9.18E-04	1.09E-06	1.73E-10	2.62E-13	6.81E-12
0.3	1.29E-03	1.54E-06	2.87E-09	3.61E-13	7.08E-12
0.4	1.66E-03	2.55E-07	1.21E-09	1.22E-12	5.63E-12
0.5	2.17E-03	4.07E-06	7.40E-10	2.20E-12	7.54E-13
0.6	3.06E-03	1.29E-05	3.60E-10	3.22E-12	1.41E-11
0.7	4.66E-03	2.78E-05	3.15E-11	4.23E-12	4.24E-12
0.8	7.40E-03	4.99E-05	2.34E-10	5.20E-12	4.03E-13
0.9	1.19E-02	8.02E-05	4.18E-10	6.12E-12	2.72E-12
1.0	1.88E-02	1.20E-04	5.01E-10	7.01E-12	5.13E-12

ACKNOWLEDGEMENT

The authors would like to express gratitude to Universiti Teknologi MARA Cawangan Terengganu Kampus Kuala Terengganu for the support given in the advancement of the research at the university.

REFERENCES

[1] J. Kuboye, O. Elusakin, and O. Quadri, "Numerical algorithms for direct solution of fourth order ordinary differential equations," *Journal of the Nigerian Society of Physical Sciences*, pp. 218–227, 2020.

[2] C. D. Pratiwi and S. Mungkasi, "Euler's and Heun's numerical solutions to a mathematical model of the spread of COVID-19," in *AIP Conference Proceedings*, vol. 2353, no. 1. AIP Publishing LLC, 2021, p. 030110.

[3] M. A. Akanbi, "Third order Euler method for numerical solution of ordinary differential equations," *ARPN Journal of Engineering and Applied Sciences*, vol. 5, pp. 42–49, 2010.

[4] J. Butcher, "Runge and his legacy," in *AIP Conference Proceedings*, vol. 1863, no. 1. AIP Publishing LLC, 2017, p. 320003.

[5] A. A. Okeke, P. Tumba, O. F. Anorue, and A. Dauda, "Analysis and comparative study of numerical solutions of initial value problems (IVP) in ordinary differential equations (ODE) with Euler and Runge-Kutta methods," *American Journal of Engineering Research (AJER)*, vol. 8, no. 8, pp. 6–15, 2019.

[6] K. J. Audu, T. A. Abubakar, Y. Y. Amuda, and J. E. Nkereuwem, "Comparative numerical evaluation of some Runge-Kutta methods for solving first order systems of odes," *Journal of Engineering and Basic Sciences*, vol. 3, pp. 20–28, 2025.

[7] E. Fehlberg, *Low-Order Classical Runge-Kutta Formulas with Stepsize Control and their*

Application to Some Heat Transfer Problems. National aeronautics and Space Administration, 1969, vol. 315.

[8] A. Abdulsalam, N. Senu, Z. A. Majid, and N. M. A. N. Long, "Development of high-order adaptive multi-step Runge-Kutta-Nyström method for solving special second-order odes," *Mathematics and Computers in Simulation*, vol. 216, pp. 104–125, 2024.

[9] S. Banu, I. Raju, and U. H. M. Zaman, "A study on numerical solution of initial value problem by using Euler's method, Runge-Kutta 2nd order, Runge-Kutta 4th order, and Runge-Kutta-Fehlberg method with MATHLAB," *International Journal of Scientific & Engineering Research*, vol. 12, no. 3, pp. 61–73, 2021.

[10] T. S. Phillips and C. J. Roy, "Richardson extrapolation-based discretization uncertainty estimation for computational fluid dynamics," *Journal of Fluids Engineering*, vol. 136, no. 12, p. 121401, 2014.

[11] A. T. Tiruneh, "Higher order Aitken extrapolation with application to converging and diverging Gauss-Seidel iterations," *arXiv preprint arXiv:1310.4288*, 2013.

[12] T. Bullo, G. Duressa, and G. Degla, "Higher order fitted operator finite difference method for two-parameter parabolic convection-diffusion problems," *International Journal of Engineering and Applied Sciences*, vol. 11, no. 4, pp. 455–467, 2019.

[13] S. Purwani, A. Ridwan, R. Hidayana, and S. Sukono, "Secant method with Aitken extrapolation outperform Newton-Raphson method in estimating stock implied volatility." *IAENG International Journal of Computer Science*, vol. 50, no. 2, 2023.

[14] D. N. Butusov, "Adaptive stepsize control for extrapolation semi-implicit multistep methods," *Mathematics*, vol. 9, no. 9, p. 950, 2021.

[15] H. Yang, L. Zhang, and Y. Zhou, "High-order schemes based on extrapolation for semilinear fractional differential equations," *Calcolo*, vol. 60, no. 4, p. 53, 2023.

[16] V. Rybin, V. Khryashchev, and E. Rybin, "Variable-step semi-implicit solver with adjustable symmetry and its application for chaos-based communication," *Mathematics*, vol. 13, no. 8, p. 1229, 2025.

[17] J. Choi, H. Kang, B. Han, and J. Lee, "Enhanced diffusion sampling via extrapolation with multiple ode solutions (rx-dpm)," *arXiv preprint*, 2025.

[18] I. Fekete and L. Lóczi, "Linear multistep methods and global richardson extrapolation," *arXiv preprint*, 2022.

[19] A. Materdey, "Higher-order numerical solutions of the quarter car suspension model," in *Proceedings of the 4th World Congress on Mechanical, Chemical, and Material Engineering (MCM'18), Madrid, Spain, Article*, no. 135, 2018.

[20] C. Bulte, "The differential equation of the deflection curve," *International Journal of*

Mathematical Education in Science and Technology, vol. 23, no. 1, pp. 51–63, 1992.

[21] A. M. B. Richard L. Burden, J. Douglas Faires, *Numerical Analysis*. Cengage Learning, MA, 2014, vol. 10.