Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. *, No. *, *, pp. 1-20
DOI:10.22034/cmde.2026.67441.3211

Enhanced multistep two-derivative Runge-Kutta methods: application to coupled spring systems

Jin Yoong Lim[1,*], Khai Chien Lee[2,*], Nurul Huda Abdul Aziz[3,*], Zhuoyu Sun[1], and Norazak Senu[4,5]
[1]Institute of Actuarial Science and Data Analytics, UCSI University, 56000 Cheras, Kuala Lumpur, Malaysia.
[2]Department of Mathematical Sciences, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia.
[3]Institute of Engineering Mathematics, University of Malaysia Perli, 02600 Arau, Perlis, Malaysia.
[4]Institute for Mathematical Research, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia.
[5]Department of Mathematics and Statistics, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia.

Abstract

Explicit multistep two-derivative Runge–Kutta methods with single f–evaluation and multiple g–evaluations whilst incorporating the value of 2 step-points x_i and x_{i-1} is proposed for solving first-order ordinary differential equations with exponential solutions. The proposed methods are constructed based on the algebraic order conditions derived from Taylor series expansion. Then, multistep two-derivative Runge–Kutta (MTDRK) methods of two-stages fourth order and three-stages fifth order, are derived. The properties of absolute stability and stability region for both MTDRK methods are analyzed. The numerical tests show that the proposed method is more effective than other existing methods with similar algebraic order in solving first-order ordinary differential equations. Also, the proposed method is used to solve a coupled springs system, and the results show that the proposed methods work effectively for solving this model.

Keywords. Two-derivative Runge-Kutta methods, Multistep, First-order ODEs, Stability, Coupled spring systems, Numerical tests.
1991 Mathematics Subject Classification. 65L06, 65L20, 65G30.

1. Introduction

In this study, we consider the first-order ordinary differential equation (ODE) in the form:

$$\begin{cases} y'(x) = f(x, y(x)), \\ y(x_0) = y_0. \end{cases} \tag{1.1}$$

First-order ODEs are equations that involve a function and its first derivative. They appear frequently in mathematical modeling to describe a wide range of phenomena in various fields such as physics, engineering, and natural sciences. Several numerical techniques are available for attaining the approximate solution of first-order ODEs depending on the form and characteristics of the equation such as Taylor's method, Picard's method, Euler's method and Heun's (modified Euler) method. Sezer and Akyüz-Daşcıoğlu [33] attempted to use Taylor polynomials to approximate the solution of generalized pantograph equations with linear functional argument. The numerical performance of various forms of Euler's method in solving first-order ODE have been studied and compared [3]. On the other hand, Hyun's method has also been implemented in solving first-order ODEs and is proven to be more accurate than Euler's method [23]. A comparative analysis showing the strength of both Picard's and Taylor's method in numerically solving first-order ODEs based on real-life problems have been studied [12]. Recent studies have further focused on the development of efficient numerical algorithms for a broader class of differential equations. For example, block multistep methods have been proposed for solving neutral delay differential equations with derivative discontinuities [5], while

Received: 22 May 2025; Accepted: 01 September 2026.
∗ Corresponding author. Email: kclee_1017@ukm.edu.my, hudaaziz@unimap.edu.my.

Boole's quadrature rule has been employed to obtain numerical solutions of second-kind Volterra integro-differential equations [11]. In addition, numerical methods have also been applied in engineering applications, such as predicting stress distribution in orthopedic screw implants embedded in trabecular bone [38], highlighting the versatility of numerical techniques beyond conventional ODE problems. According to various studies, although the aforementioned methods have been generally proven to have high applicability in solving first-order ODEs, a particular method known as Runge-Kutta method, is proven to be a more powerful technique with higher numerical performance in finding the solution of first-order ODEs in comparison to the exact solution [15, 19, 39].

However, as the classical RK methods are considered as first derivative single-step methods, some authors are motivated to extend and modify the RK methods into two-step methods in order to enhance the numerical performance of the classical RK methods. Some authors have extended the method into a certain special case of general two-step RK methods such that the value of the previous point (x_{i-1}, y_{i-1}) is utilised to solve (x_{i+1}, y_{i+1}) (see [14, 27, 28, 35–37]). Their proposed methods incorporate higher-order derivatives of the solution in internal stages or external stages in RK methods. For example, Rabiei & Ismail [29] developed the fifth-order Improved RK method with lower number of function evaluations and has proven that this method has a lower computational cost compared to the classical fifth-order RK method (RK5). However, the derivatives are approximated using the difference quotient of past and current evaluations of the solution instead of directly evaluating the higher order derivatives assuming that they exceed the original function itself making the RK method 'derivative-free'. Their methods have proven to be more attractive than the classical RK methods in terms of computational efficiency with a reduced number of function evaluations of similar order. Despite that, some authors have found it worthwhile to include the evaluation of higher derivative terms instead of approximating them.

Furthermore, with the aim of enhancing classical RK methods, modifications have been introduced by incorporating a two-derivative term. Two-derivative Runge-Kutta (TDRK) methods serve as an extension of RK methods for problems of the form $y' = f(y)$, involving the second derivative $y'' = f'(x, y)f(x, y) = g(x, y)$. Explicit two-derivative Runge-Kutta methods capitalizing on the cost-effective computations of second derivatives were derived [7]. The method is constructed by involving only one evaluation of the first derivative function and multiple evaluations of the second derivative function per step. They also presented methods with stages up to five and of order up to seven, including some embedded pairs. Chan, Wang, and Tsai [8] later delved further into the approaches of both explicit and implicit TDRK methods for stiff ODE problems. They applied these TDRK methods to diverse partial differential equations. After applying simplifying conditions, Kalogiratou, Monovasilis and Simos [22] presented order conditions up to order six for the TDRK methods. Using a similar approach of simplifying conditions, a regular TDRK method of order five was derived [20, 21]. In the work of Monovasilis and Kalogiratou [26], they considered a special type of explicit TDRK methods where the function f is evaluated only once at each step. More recently, frequency-dependent explicit improved two-derivative Runge-Kutta-Nyström methods have been developed to efficiently solve oscillatory second-order initial value problems, particularly the two-body problem, by optimizing the frequency-dependent parameters of the numerical scheme [24]. Similarly, frequency-dependent coefficients have been proposed for explicit improved two-derivative Runge–Kutta-type methods to enhance the accuracy and stability of solving third-order initial value problems [25]. Then in 2023, by building on the work of Chan and Tsai, they also presented new 7th order methods with minimized dispersion and dissipation error.

In this research paper, traditional TDRK scheme is modified by removing the increment function $\sum_{i=1}^{s} b_i k_i$ in first derivative and replaced it with a single function $f(x, y)$ from the first-order numerical problem as well as inserting evaluation function with second derivative in the formulation.

The objective of this paper is to design multistep two-derivative Runge–Kutta type (MTDRK) methods which are two-step, with minimal number of functional evaluations, consisting of two-stage fourth-order and three-stage fifth-order. In Section 2, general formulation of MTDRK methods is proposed. In Section 3, construction of MTDRK method based on Taylor series expansion will be shown and the methods with two-stage fourth-order and three-stage fifth-order are derived. The linear stability analysis of MTDRK methods is discussed. In Section 4, capability of MTDRK methods is illustrated through the numerical test. Numerical results of MTDRK methods and other existing methods are exhibited in Section 5. This paper ends in Section 6 with the discussion and conclusion.

2. Formulation of MTDRK methods

For a general, non-autonomous first-order ODE, the second derivative term is computed as the total derivative of f along the solution:

$$g(x, y) = y'' = \frac{d}{dx} f(x, y) = f_x(x, y) + f_y(x, y) f(x, y). \tag{2.1}$$

Here f_x is the partial derivative with respect to x, while f_y is the Jacobian with respect to y. In the numerical experiments, these derivatives are evaluated analytically for each test problem.

However, describing our method as an autonomous case provides simplicity by carrying lesser notation around, benefitting the method to be applicable to a wider range of cases [4]. Hence, we rewrite the derivative for the first-order ODEs in Eq. (2.1) is defined as follows:

$$y'' = f'(x, y) f(x, y) = g(x, y), \quad g : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N.$$

The general form of an s-stage MTDRK method is defined as follows (Ökten Turaci and Öziş 2018):

$$\begin{aligned} y_{n+1} = y_n &+ h \sum_{i=1}^{s} B_i \left[f(x_n + c_i h, Y_i) - f(x_{n-1} + c_i h, Y_{-i}) \right] \\ &+ h^2 \sum_{i=1}^{s} b_i \left[g(x_n + c_i h, Y_i) - g(x_{n-1} + c_i h, Y_{-i}) \right], \end{aligned} \tag{2.2}$$

where

$$Y_i = y_n + h \sum_{j=1}^{i-1} A_{ij} f(x_n + c_i h, Y_j) + h^2 \sum_{j=1}^{s} a_{ij} g(x_n + c_i h, Y_j), \quad i = 1, 2, \ldots, s. \tag{2.3}$$

$$Y_{-i} = y_{n-1} + h \sum_{j=1}^{i-1} A_{ij} f(x_{n-1} + c_i h, Y_{-j}) + h^2 \sum_{j=1}^{s} a_{ij} g(x_{n-1} + c_i h, Y_{-j}), \quad i = 1, 2, \ldots, s. \tag{2.4}$$

To define Eqs. (2.2)–(2.4) with a minimal number of evaluation functions, the equations are modified by transforming multiple f-evaluations into a single f-evaluation that is dependent on x and y only. Therefore, the new definitions of the equations are as follows:

$$\begin{aligned} y_{n+1} = y_n &+ \frac{3}{2} h f(x_n, y_n) - \frac{1}{2} h f(x_{n-1}, y_{n-1}) + h^2 \left[b_1 g(x_n + c_1 h, Y_1) - b_1 g(x_{n-1} + c_1 h, Y_{-1}) \right] \\ &+ h^2 \sum_{i=2}^{s} b_i \left(g(x_n + c_i h, Y_i) - g(x_{n-1} + c_i h, Y_{-i}) \right), \end{aligned} \tag{2.5}$$

$$Y_i = y_n + c_i h f(x_n, y_n) + h^2 \sum_{i=1}^{s} a_{ij} g(x_n + c_i h, Y_j), \quad i = 1, 2, \ldots, s, \tag{2.6}$$

$$Y_{-i} = y_{n-1} + c_i h f(x_{n-1}, y_{n-1}) + h^2 \sum_{i=1}^{s} a_{ij} g(x_{n-1} + c_i h, Y_{-j}), \quad i = 1, 2, \ldots, s. \tag{2.7}$$

An alternative expression for Eq. (2.5) can be written as follows:

$$y_{n+1} = y_n + \frac{3}{2} h f(x_n, y_n) - \frac{1}{2} f(x_{n-1}, y_{n-1}) + h^2 \left[b_1 k_1 - b_1 k_{-1} + \sum_{i=2}^{s} b_i (k_i - k_{-i}) \right], \tag{2.8}$$

TABLE 1. Butcher tableau for MTDRK methods.

$$\begin{array}{c|c} \mathrm{c} & \mathrm{a} \\ \hline b_{-1} & b^T \end{array}$$

TABLE 2. Butcher tableau for explicit form of MTDRK methods.

$$\begin{array}{c|ccccc} 0 & 0 & & & \\ c_2 & a_{21} & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & \cdots & 0 \\ \hline b_{-1} & b_1 & b_2 & \cdots & b_s \end{array}$$

where

$$k_i = g\left(x_n + c_i h,\, y_n + c_i h f(x_n, y_n) + h^2 \sum_{j=1}^{s} a_{ij} k_j\right), \tag{2.9}$$

$$k_{-i} = g\left(x_{n-1} + c_i h,\, y_{n-1} + c_i h f(x_{n-1}, y_{n-1}) + h^2 \sum_{j=1}^{s} a_{ij} k_{-j}\right). \tag{2.10}$$

In this study, we focus exclusively on the explicit form of MTDRK methods. These methods will also include an additional condition, $a_{ij} = 0$ for all $i \leq j$. Therefore, the coefficients of our MTDRK method can be presented in the following Butcher tableau:

The MTDRK methods will involve a single f-evaluation and multiple g-evaluations per step. The motivation behind proposing these methods is to reduce the number of function evaluations within each computational step and improve the accuracy of existing methods.

To determine the coefficients of the explicit MTDRK method given in Eqs. (2.5)–(2.7), component Y_j and Y_{-j} will be substituted into the g-evaluation functions in Eqs. (2.6) and (2.7) respectively in order to expand them to derive Y_i and Y_{-i}. Taylor series expansion will then be applied to the expanded g-evaluations in Eq. (2.5) to obtain the complete form of y_{n+1}. The order conditions for our explicit MTDRK methods can then be obtained by comparing the coefficients of different powers of h with the Taylor series expansion of y_{n+1}.

The order conditions of y for explicit MTDRK methods are shown as below:

Order 2:

$$b_{-1} = b_1. \tag{2.11}$$

Order 3:

$$b_{-1} + \sum_{i=2}^{s} b_i = \frac{5}{12}. \tag{2.12}$$

Order 4:

$$\sum_{i=2}^{s} b_i c_i = \frac{1}{6}. \tag{2.13}$$

Order 5:

$$\sum_{i=2}^{s} b_i c_i^2 = \frac{31}{360}, \tag{2.14}$$

$$\sum_{i=2}^{s}\sum_{j=1}^{i-1} b_i a_{ij} = \frac{31}{720}. \tag{2.15}$$

Order 6:

$$\sum_{i=2}^{s} b_i c_i^3 = \frac{1}{20}, \tag{2.16}$$

$$\sum_{i=2}^{s}\sum_{j=2}^{i-1} b_i c_i a_{ij} = \frac{1}{40}, \tag{2.17}$$

$$\sum_{i=3}^{s}\sum_{j=2}^{i-1} b_i a_{ij} c_j = \frac{1}{120}. \tag{2.18}$$

There are free unknown parameters as the number of equations is not equal to the number of unknown parameters. Hence, the following simplifying assumption, also known as the Nystrom row assumption, is applied in practice to solve all free unknown parameters:

$$\sum_{j=1}^{s-1} a_{ij} = \frac{1}{2}c_i^2, \quad i = 2, \dots, s. \tag{2.19}$$

This simplifying assumption was first proposed by Hairer et al. [16] and has been proven to be highly convenient and utilised in the derivation of various explicit Runge-Kutta methods [7, 20–22, 26].

3. Construction of MTDRK Methods

3.1. Two-stage Fourth-order MTDRK Method.

To derive the fourth-order MTDRK method, we apply the algebraic order conditions up to order four in the Eqs. (2.11)–(2.13). Hence, we obtain a system of equations consisting of 3 nonlinear equations with 5 unknown variables. By solving the system simultaneously with the simplifying assumption:

$$a_{21} = \frac{(c_2)^2}{2}, \tag{3.1}$$

we attain a solution with one free parameter c_2 as follows:

$$b_{-1} = \frac{-2+5c_2}{12c_2}, \quad b_1 = \frac{-2+5c_2}{12c_2}, \quad b_2 = \frac{1}{6c_2}. \tag{3.2}$$

Minimization of the error equation of fifth-order conditions are utilised to select the parameter that generate minimum value of truncation error norm for y_n. The above solution is substituted into $\|\tau^{(5)}\|_2$ as the following:

$$\|\tau^{(5)}\|_2 = \sqrt{\sum_{i=1}^{n}(\tau^{(5)})^2} = \sqrt{\left(b_2c_2 - \frac{31}{360}\right)^2 + \left(b_2a_{21} - \frac{31}{720}\right)^2}, \tag{3.3}$$

where n is the total number of conditions with order five for y and $\tau^{(5)}$ is the error norm with order five for y_n.

The error equation achieves a minimum value at $c_2 = \frac{31}{60}$. By substituting into the solution in Eq. (3.1), all the coefficients of the new method are obtained and presented in the Butcher tableau, and denoted as MTDRK4 as shown in Table 3:

With $c_2 = \frac{31}{60}$, the global truncation error of the fifth-order condition is

$$\|\tau^{(5)}\|_2 = 0.00144117. \tag{3.4}$$

Table 3. Butcher tableau for MTDRK4.

$$
\begin{array}{c|cc}
0 & 0 & \\
\frac{31}{60} & \frac{961}{7200} & 0 \\
\hline
\frac{35}{372} & \frac{35}{372} & \frac{10}{31}
\end{array}
$$

3.2. **Three-stage Fifth-order MTDRK Method.** To derive the fifth-order MTDRK method, we apply the algebraic order conditions up to order five in the Eqs. (2.11)–(2.15). Hence, we obtain a system of equations consisting of 5 nonlinear equations with 9 unknown variables. By solving the system simultaneously, we attain a solution with three free parameters a_{31}, a_{32}, and c_3 as follows:

$$
\begin{aligned}
a_{21} &= \frac{(720a_{31}b_3 + 720a_{32}b_3 - 31)(360b_3c_3^2 - 31)}{259200b_3^2c_3^2 - 86400b_3c_3 + 7200}, \\
b_{-1} &= \frac{1800b_3c_3^2 - 1440b_3c_3 + 372b_3 - 35}{4320b_3c_3^2 - 372}, \\
b_1 &= \frac{1800b_3c_3^2 - 1440b_3c_3 + 372b_3 - 35}{4320b_3c_3^2 - 372}, \\
b_2 &= \frac{-10(36b_3^2c_3^2 - 12b_3c_3 + 1)}{360b_3c_3^2 - 31}, \\
c_2 &= \frac{360b_3c_3^2 - 31}{360b_3c_3 - 60},
\end{aligned}
\tag{3.5}
$$

Minimization of the error equation of sixth-order conditions is utilised to select the parameters that generate the minimum value of truncation error norm for y_n. The above solution is substituted into $\|\tau^{(6)}\|_2$ as the following:

$$
\begin{aligned}
\|\tau^{(6)}\|_2 &= \sqrt{\sum_{i=1}^{n}(\tau^{(6)})^2} \\
&= \sqrt{\left(b_2c_2^3 + b_3c_3^3 - \frac{1}{20}\right)^2 + \left(b_2c_2a_{21} + b_3c_3(a_{31}+a_{32}) - \frac{1}{40}\right)^2 + \left(b_3a_{32}c_2 - \frac{1}{120}\right)^2}.
\end{aligned}
\tag{3.6}
$$

where n is the total number of conditions with order six for y and $\tau^{(6)}$ is the error norm with order six for y_n.

The error equation achieves a minimum value at $a_{31} = \frac{112933}{137197}$, $a_{32} = \frac{-18011}{23675}$, and $c_3 = \frac{-9026}{25553}$. By substituting them into the solution in Eq. (3.6), all the coefficients of the new method are obtained and presented in the Butcher tableau, and denoted as MTDRK5 as shown in Table 4:

Table 4. Butcher Tableau for MTDRK5.

$$
\begin{array}{c|ccc}
0 & 0 & & \\
c_2 & a_{21} & 0 & \\
c_3 & a_{31} & a_{32} & 0 \\
\hline
b_{-1} & b_1 & b_2 & b_3
\end{array}
$$

where

$$
\begin{aligned}
&a_{21} = \frac{7366385373560511819684980848 31}{478873713071265091532254764720 0}, \quad a_{31} = \frac{112933}{137197}, \quad a_{32} = \frac{-18011}{23675}, \\
&b_{-1} = \frac{3428952315066565}{23088890368349628}, \quad b_1 = \frac{3428952315066565}{23088890368349628}, \quad b_2 = \frac{511910346497447788 90}{177805620652463122759}, \\
&b_3 = \frac{-1825}{92411}, \quad c_2 = \frac{1924074197362469}{3468886570583940}, \quad c_3 = \frac{-9026}{25553}.
\end{aligned}
$$

With the solved coefficients, it is observed that the global truncation error of the sixth-order condition is

$$\|\tau^{(6)}\|_2 = 6.80064 \times 10^{-12}. \tag{3.7}$$

4. Stability analysis

In this section, the stability analysis of the MTDRK4 and MTDRK5 methods will be carried out, and their stability regions will be plotted to illustrate the efficient region for step sizes. The stability region is a crucial factor in analysing the numerical performance of newly proposed methods. For the Dahlquist test equation, the region of absolute stability is the set of values of the scaled variable $v = \lambda h$ for which the numerical solution remains bounded as the number of steps increases.

4.1. **Absolute stability [9].** The classical stability regions are defined given the behaviour of numerical methods on the standard Dahlquist test problem [10], where

$$y' = \lambda y, \tag{4.1}$$

where λ is a complex constant. A method is absolutely stable for a given value of $v = \lambda h$ when the roots of its stability polynomial is uniformly bounded inside the unit circle. The numerical method is said to be stable if there exists $h > 0$ for each differential equation such that changing the initial values by a fixed amount produces a bounded change in the numerical solution for all h [1].

4.2. **Region of absolute stability.** A one-step method with scalar stability function $M(v)$ has region of absolute stability

$$S = \{v \in \mathbb{C} \mid |M(v)| < 1\}, \tag{4.2}$$

while, for the two-step MTDRK recurrence below, the corresponding condition is imposed on the eigenvalues of the stability matrix.

4.3. **Derivation of Stability Region.** In this section, we study the linear stability of the MTDRK methods using the Dahlquist test problem [2]:

$$y' = \lambda y, \quad y'' = \lambda^2 y, \quad \lambda \in \mathbb{C}. \tag{4.3}$$

By applying the test problem (4.3) to Eqs. (2.5)–(2.7), along with the substitution $v = \lambda h$, we derived the following:

$$\begin{aligned} &Y = y_n + ch\lambda y_n + h^2 a(\lambda^2 Y).\\ &(I - v^2 a)Y = (1 + cv)y_n.\\ &\Rightarrow Y = (I - v^2 a)^{-1}(1 + cv)y_n, \end{aligned} \tag{4.4}$$

$$\begin{aligned} &\bar{Y} = y_{n-1} + ch\lambda y_{n-1} + h^2 a(\lambda^2 \bar{Y}).\\ &(I - v^2 a)\bar{Y} = (1 + cv)y_{n-1}.\\ &\Rightarrow \bar{Y} = (I - v^2 a)^{-1}(1 + cv)y_{n-1}, \end{aligned} \tag{4.5}$$

where $\bar{Y}$ is the matrix for the values of Y_{-j} for $j = 1, 2, ..., i-1$, and

$$\begin{aligned} y_{n+1} =& y_n + \frac{3}{2}h\lambda y_n - \frac{1}{2}h\lambda y_{n-1} + h^2 b^T(\lambda^2 Y - \lambda^2 \bar{Y})\\ =& y_n + \frac{3}{2}vy_n + v^2 b^T((I - v^2 a)^{-1}(1 + cv)y_n) - \frac{1}{2}vy_{n-1} - v^2 b^T((I - v^2 a)^{-1}(1 + cv)y_{n-1})\\ =& y_n + \frac{3}{2}vy_n + v^2 b^T(I - v^2 a)^{-1}ey_n + v^2 b^T(I - v^2 a)^{-1}cvy_n - \frac{1}{2}vy_{n-1}\\ &- v^2 b^T(I - v^2 a)^{-1}ey_{n-1} + v^2 b^T(I - v^2 a)^{-1}cvy_{n-1}. \end{aligned} \tag{4.6}$$

After arrangement, we yield

$$\begin{aligned} y_{n+1} = &\left[(1 + v^2 b^T (I - v^2 a)^{-1} e) + \left(\frac{3}{2} v + v^3 b^T (I - v^2 a)^{-1} c\right)\right] y_n \\ &- \left[v^2 b^T (I - v^2 a)^{-1} e + \left(\frac{1}{2} v + v^3 b^T (I - v^2 a)^{-1} c\right)\right] y_{n-1}. \end{aligned} \tag{4.7}$$

However, based on the order condition in Eq. (2.11), we know that $b_{-1} = b_1$, which implies that $\hat{b}^T = b^T$. Thus, we redefine Eq. (4.7) as follows:

$$\begin{aligned} y_{n+1} = &\left[(1 + v^2 b^T (I - v^2 a)^{-1} e) + \left(\frac{3}{2} v + v^3 b^T (I - v^2 a)^{-1} c\right)\right] y_n \\ &- \left[v^2 b^T (I - v^2 a)^{-1} e + \left(\frac{1}{2} v + v^3 b^T (I - v^2 a)^{-1} c\right)\right] y_{n-1}. \end{aligned} \tag{4.8}$$

Hence, we obtained the difference equation into a form of matrix as follows:

$$y_{n+1} = M(v) \begin{bmatrix} y_n \\ y_{n-1} \end{bmatrix}, \quad v = \lambda h, \tag{4.9}$$

where

$$M(v) = \begin{bmatrix} M_1 & M_2 \end{bmatrix}, \tag{4.10}$$

with

$$M_1 = (1 + v^2 b^T (I - v^2 a)^{-1} e) + \left(\frac{3}{2} v + v^3 b^T (I - v^2 a)^{-1} c\right), \tag{4.11}$$

$$M_2 = v^2 b^T (I - v^2 a)^{-1} e + \left(\frac{1}{2} v + v^3 b^T (I - v^2 a)^{-1} c\right). \tag{4.12}$$

$M(v)$ is also known as the stability matrix. The stability region of the MTDRK method is defined as follows:

$$E_R = \{v : |r_i(M(v))| < 1, i = 1, 2\}. \tag{4.13}$$

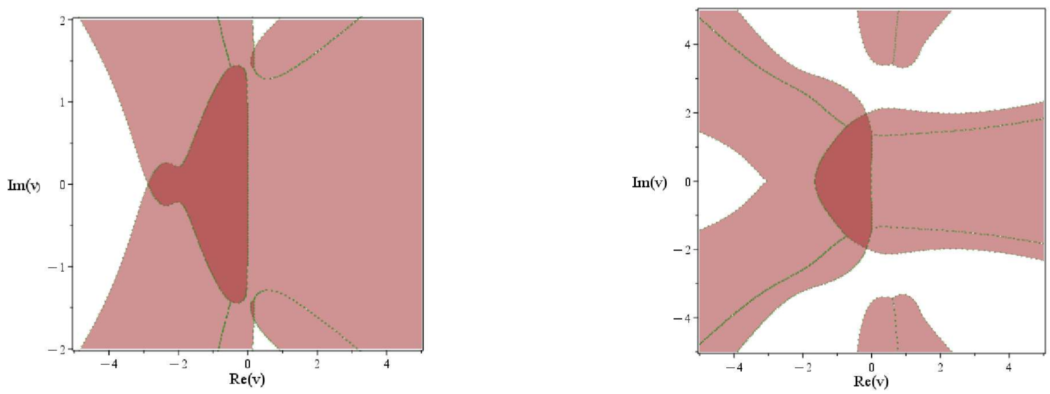
r_i are eigenvalues of $M(v)$. By substituting y_{n+1} with ξ^2 and y_n with ξ, we obtained the stability polynomial for MTDRK4 as follows:

$$\rho(\xi, v) = \xi^2 - \left(1 + \frac{3}{2} v + \frac{5}{12} v^2 + \frac{1}{6} v^3 + \frac{31}{720} v^4\right) \xi + \frac{1}{2} v + \frac{5}{12} v^2 + \frac{1}{6} v^3 + \frac{31}{720} v^4. \tag{4.14}$$

Using the same procedure, the stability polynomial for MTDRK5 is given by:

$$\begin{aligned} \rho(\xi, v) = \xi^2 - &\left(\begin{array}{l} 1 + \frac{3}{2} v + \frac{5}{12} v^2 + \frac{1}{6} v^3 + \frac{31}{720} v^4 \\ + \frac{816057589327343976 97}{979269107096445670758 0} v^5 \\ + \frac{3124305028488219878082990927193720 3}{1351863441161728522511959265879130400} v^6 \end{array}\right) \xi \\ &+ \frac{1}{2} v + \frac{5}{12} v^2 + \frac{1}{6} v^3 + \frac{31}{720} v^4 + \frac{81605758932734397697}{97926910709644567075 80} v^5 \\ &+ \frac{31243050284882198780829909271937203}{13518638441161728522511959265879130400} v^6. \end{aligned} \tag{4.15}$$

Stability region of the methods is the set of values of such that all the roots of stability polynomial are within the unit circle. The stability regions for MTDRK4 and MTDRK5 are shown in Figures 1a and 1b below. The stability



(A) MTDRK4. (B) MTDRK5.

Figure 1. Stability region for MTDRK methods.

region for v is the intersection region of stability region for v_1 (corresponding to y) and v_2 (corresponding to y').

4.4. **Discussion.** We denote S, which is the overlapping region as our absolute stability region. According to Figures 1(A) and 1(B), as the bounded regions satisfies Eq. (4.2), hence the plotted regions indicate absolute stability of the MTDRK4 and MTDRK5 methods over the bounded subsets of the v-plane. By comparing these two stability regions for MTDRK4 and MTDRK5 methods, it is evident that MTDRK4 method has a slightly bigger area than the region of MTDRK5 method. Specifically, the stability region for MTDRK4 method spans the intervals $[-2.75, -2.9]$ and $[-2.1, 0]$ on the real part of v, denoted as $\mathrm{Re}(v)$ whereas the stability region of MTDRK5 falls within $[-1.85, 0]$ on the $\mathrm{Re}(v)$. Since the stability regions of the methods are relatively small, this means that a relatively large step size may cause the error of approximations to grow uncontrollably large for some scenarios. Hence, this suggests that a relatively smaller step size may be required for the utilisation of these methods for numerical approximations.

On the other hand, the stability region of numerical methods with higher order is generally expected to be larger than that of methods with lower order, which is reflected differently by our methods as the MTDRK4 method has a bigger stability region than MTDRK5. Despite this, this does not pose as a big issue as it has been presented that the evolution of the size of the region of stability does not depend on the order of the method [31]. Some numerical methods with higher order region do have smaller stability region, and vice versa for some methods.

5. Numerical Experiment

In this section, we test the efficiency of the proposed methods, MTDRK4 and MTDRK5, on selected numerical problems for comparison purpose. The numerical results of the following methods are utilised for comparison. The experiments were conducted using Maple 2024.1 and Predator laptop with the following specifications: 13th Gen Intel(R) Core(TM) i7-13700HX CPU @2.10 GHz.

- MTDRK4: Explicit multistep two-derivative Runge-Kutta method with two-stage fourth-order,
- MTDRK5: Explicit multistep two-derivative Runge-Kutta method with three-stage fifth-order,
- RK4: Classical Runge-Kutta fourth-order method as given in Butcher [6],
- IRK4: Fourth-order multistep Runge-Kutta method proposed by Rabiei, Ismail & Suleiman [18],
- IRK5: Fifth-order multistep Runge-Kutta method proposed by Rabiei & Ismail [29],
- TDRK4: Explicit three-stage fourth-order two-derivative Runge-Kutta method proposed by Amirah & Senu [32]
- TDRK5: Explicit four-stage fifth-order two-derivative Runge-Kutta method proposed by Senu, Ahmad & Bachok [32].

TABLE 5. Maximum Error (log10) and Computational Time for Problem 5.1.

Method	$h = 0.025$		$h = 0.0125$		$h = 0.00625$		$h = 0.003125$		$h = 0.0015625$	
	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time
MTDRK4	-9.438388265	0.004	-10.92812959	0.009	-12.42578662	0.020	-13.92727217	0.037	-15.43059105	0.073
MTDRK5	-10.66553664	0.006	-12.47068702	0.015	-14.27648436	0.030	-16.08249729	0.059	-17.88860678	0.121
RK4	-9.040627691	0.004	-10.24689786	0.009	-11.45200589	0.014	-12.65663807	0.028	-13.86101108	0.053
IRK4	-9.469096652	0.006	-10.66579911	0.012	-11.86780515	0.048	-13.07126559	0.070	-14.27515526	0.149
IRK5	-6.11030654	0.006	-6.393832772	0.039	-6.68633513	0.054	-6.983155572	0.075	-7.282094802	0.160
TDRK4	-8.135719457	0.006	-9.339479733	0.010	-10.54336943	0.023	-11.7473731	0.040	-12.95143706	0.099
TDRK5	-10.79025524	0.007	-12.29293288	0.017	-13.79614798	0.034	-15.300594	0.066	-16.80539796	0.142

All the methods taken for comparison are of similar order to our proposed methods, which are fourth and fifth order. The RK4 method is considered as it's the classical RK method. Since our methods are two-stepped, hence the IRK4 and IRK5 methods are used. However, these methods only consist of the first-derivative term whereas our methods are two-derivative. Hence, the TDRK4 and TDRK5 methods are applied but they are only single-step. Hence, if our methods of lower order can perform similarly or better than these methods, then the favourableness of our methods can be highlighted. The evaluation technique will be based on the logarithmic value of the maximum global truncation error obtained from the approximated results of RK4 methods with very small values of h. The methods with the lower maximum global truncation error will be deemed as the better methods in solving first-order ODEs.

It is important to note that since the proposed methods are two-step methods, an additional starting value, y_1 is required other than the initial conditions. In this work, the missing starting value was generated using two approaches: (1) using the exact solution of the test problem, or, (2) using a one-step method of order at least equal to the order of the proposed methods if no exact solution is available. Since the MTDRK methods are of order 4 and 5, the Runge-Kutta method of order 4 will be used.

Below are the numerical problems to be tested with the numerical methods mentioned above.

Problem 5.1. We consider a non-linear first-order ODE [35]:

$$y' = \frac{-xy}{1+x^2}, \quad y(0) = 1, \quad x \in [0, 5]. \tag{5.1}$$

Exact solution: $y(x) = \frac{1}{\sqrt{1+x^2}}$.

Overall, it can be seen that the IRK5 method somehow performs the worst as it generally produces large truncation error despite the reduction of time steps. From Figure 2 and Table 5, it is without a doubt that our proposed MTDRK4 and MTDRK5 methods are clearly favourable as they produce approximations with better accuracy compared to the other family of Runge-Kutta methods. Despite the proposed methods having a higher complexity of single function evaluation as they incorporate a higher derivative and the information of an extra step-point, they have shorter computational time, which shows their enhanced efficiency compared to the existing methods.

Problem 5.2. We consider an oscillatory problem [17]:

$$y' = y\cos(x), \quad y(0) = 1, \quad x \in [0, 5]. \tag{5.2}$$

Exact solution: $y(x) = e^{\sin(x)}$.

Similarly in Problem 5.2, it is obvious that our proposed methods have the most optimal performance compared to the other methods with similar order. Furthermore, our MTDRK method of order 4 has similar performance to TDRK5 methods in terms of accuracy but of higher order and shorter computational time. Hence this further justifies that our methods have a better numerical performance.

Problem 5.3. We consider a problem which its stiffness depends on the value of λ [7]:

$$y' = \lambda(y - \sin(x)) + \cos(x), \quad y(0) = 0, \quad x \in [0, 5]. \tag{5.3}$$

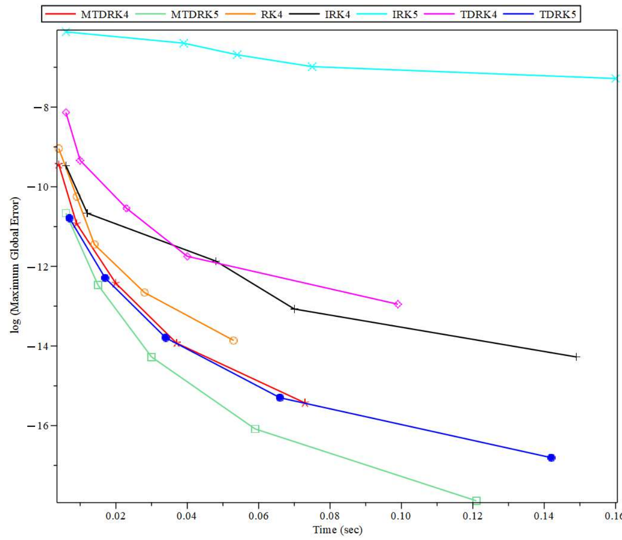


FIGURE 2. Time curves for all methods for Problem 5.1 with $h = 0.0015625(2^i)$, $i = 0, 1, 2, 3, 4$.

TABLE 6. Maximum Error (log10) and Computational Time for Problem 5.2.

Method	$h = 0.025$		$h = 0.0125$		$h = 0.00625$		$h = 0.003125$		$h = 0.0015625$	
	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time
MTDRK4	-9.216232393	0.034	-10.72299329	0.087	-12.22898819	0.116	-13.73457033	0.270	-15.23994142	0.552
MTDRK5	-10.69366768	0.045	-12.49707592	0.090	-14.30185715	0.210	-16.10736281	0.410	-17.91320626	0.807
RK4	-8.358044234	0.021	-9.56382277	0.059	-10.76880598	0.102	-11.97336763	0.169	-13.1777109	0.301
IRK4	-7.871414332	0.030	-9.073863209	0.060	-10.27719232	0.138	-11.48093266	0.315	-12.68486641	0.552
IRK5	-5.382816821	0.035	-5.687502938	0.072	-5.99017597	0.141	-6.291969316	0.350	-6.593368233	0.588
TDRK4	-8.000314311	0.063	-9.2068852	0.091	-10.41223814	0.125	-11.61695286	0.280	-12.82137891	0.584
TDRK5	-10.109131	0.078	-11.61029941	0.095	-13.11345383	0.215	-14.61760628	0.422	-16.12225906	0.869

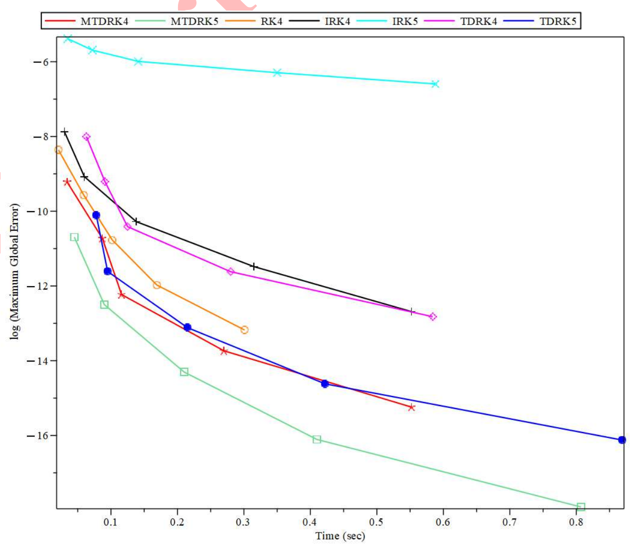


FIGURE 3. Time curves for all methods for Problem 5.2 with $h = 0.0015625(2^i)$, $i = 0, 1, 2, 3, 4$.

TABLE 7. Maximum Error (log10) and Computational Time for Problem 5.3 ($\lambda = -1$).

Method	$h = 0.025$		$h = 0.0125$		$h = 0.00625$		$h = 0.003125$		$h = 0.0015625$	
	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time
MTDRK4	-10.31896067	0.007	-11.82410804	0.013	-13.32925546	0.024	-14.8344	0.065	-16.3396	0.269
MTDRK5	-12.60383388	0.012	-14.4101214	0.023	-16.21634324	0.082	-18.0225	0.270	-19.8287	0.616
RK4	-8.563640138	0.057	-9.770413172	0.086	-10.97586297	0.143	-12.1806	0.290	-13.3851	0.540
IRK4	-8.332359201	0.060	-9.535728437	0.107	-10.73947066	0.252	-11.9434	0.508	-13.1474	0.972
IRK5	-5.770625347	0.058	-6.047469891	0.113	-6.336850502	0.262	-6.6322	0.499	-6.9304	0.990
TDRK4	-8.802678472	0.009	-10.00945544	0.017	-11.21489424	0.039	-12.4197	0.251	-13.6241	0.550
TDRK5	-12.0729673	0.014	-13.6239614	0.037	-15.1304751	0.210	-16.6363	0.576	-18.1417941	0.970

TABLE 8. Maximum Error (log10) and Computational Time for Problem 5.3 ($\lambda = -200$).

Method	$h = 0.00625$		$h = 0.003125$		$h = 0.0015625$		$h = 0.00078125$		$h = 0.000390625$	
	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time
MTDRK4	-10.84945675	0.096	-12.35460791	0.225	-13.85975821	0.439	-15.3649	0.867	-16.8701	1.685
MTDRK5	-11.89499212	0.156	-14.00212305	0.300	-16.10901127	0.613	-18.2149	1.187	-20.3171	2.340
RK4	-5.940236656	0.149	-7.271474356	0.283	-8.538806866	0.549	-9.7756	1.011	-10.9963	1.980
IRK4	-8.29247896	0.257	-9.542981538	0.494	-10.72388491	1.024	-11.9168	1.958	-13.1155	3.881
IRK5	-8.243013402	0.257	-8.783918672	0.497	-9.835236354	1.053	-9.8326	1.975	-9.8755	4.030
TDRK4	-8.338182466	0.129	-9.748596597	0.256	-11.01592899	0.467	-12.2527	0.889	-13.4734	1.769
TDRK5	-9.90060541	0.174	-11.8220813	0.308	-13.6908383	0.655	-15.5296	1.268	-17.3523	2.631

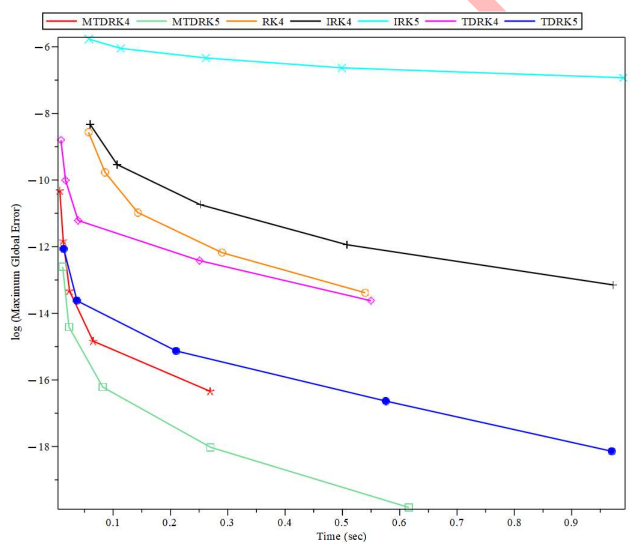


FIGURE 4. Time curves for Problem 5.3 ($\lambda = -1$) with $h = 0.0015625(2^i)$, $i = 0, 1, 2, 3, 4$.

Exact solution: $y(x) = \sin(x)$.

For Problem 5.3, the ODEs are non-stiff when $\lambda = -1$ and the equations are mildly stiff when $\lambda = -200$ respectively. In Figures 4 and 5, it can clearly be seen the TDRK and MTDRK methods are not only efficient in solving non-stiff ODEs, but also show great significance in solving mildly stiff equations.

Problem 5.4. We consider a system of first-order ODEs [34]:

$$\begin{aligned} &y_1' = -2y_1(x) + y_2(x) + 2\sin(x), y_2' = y_1(x) - 2y_2(x) + 2(\cos(x) - \sin(x)) \\ &y_1(0) = 2, \quad y_2(0) = 3, \quad x \in [0, 5]. \end{aligned} \tag{5.4}$$

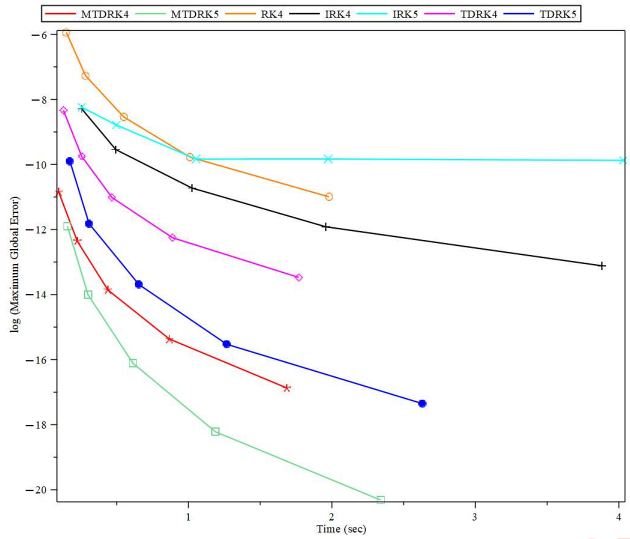


FIGURE 5. Time curves for all methods for Problem 5.3 ($\lambda = -200$) with $h = \frac{1}{160(2^i)}$, $i = 0, 1, 2, 3, 4$.

TABLE 9. Maximum Error (log10) and Computational Time for Problem 5.4.

Method	$h = 0.025$		$h = 0.0125$		$h = 0.00625$		$h = 0.003125$		$h = 0.0015625$	
	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time
MTDRK4	-9.7095	0.034	-11.2146	0.072	-12.7197	0.122	-14.2249	0.328	-15.7300	0.682
MTDRK5	-12.1291	0.061	-13.9347	0.125	-15.7402	0.276	-17.5459	0.537	-19.3519	1.096
RK4	-7.5828	0.089	-8.7947	0.110	-10.0028	0.183	-11.2088	0.347	-12.4139	0.657
IRK4	-7.7479	0.092	-8.9493	0.181	-10.1520	0.304	-11.3554	0.635	-12.5592	1.213
IRK5	-5.8233	0.141	-6.0903	0.204	-6.3749	0.342	-6.6678	0.646	-6.9648	1.277
TDRK4	-8.0948	0.069	-9.3060	0.106	-10.5137	0.168	-11.7196	0.383	-12.9246	0.773
TDRK5	-11.7908	0.093	-13.3025	0.165	-14.8034	0.298	-16.3047	0.603	-17.8075	1.270

Exact solution: $y_1(x) = 2e^{-x} + \sin(x)$, $y_2(x) = 2e^{-x} + \cos(x)$.

Problem 5.5. We consider the Kaps Problem given by Hull et al. [17], which the system's stiffness depends on the value of λ:

$$y' = \begin{bmatrix} -y_1 \cdot (1 + y_1) + y_2 \\ \lambda(y_1^2 - y_2) - 2y_2 \end{bmatrix}, \quad y(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad x \in [0, 5]. \tag{5.5}$$

Exact solution:

$$y(x) = \begin{bmatrix} e^{-x} \\ e^{-2x} \end{bmatrix}.$$

For Problem 5.5, the ODEs are non-stiff when $\lambda = 1$ and are mildly stiff when $\lambda = 100$. Similarly to Problem 5.3, in Figures 7 and 8, it can clearly be seen the TDRK and MTDRK methods show greater significance in solving equations that are mildly stiff. This further justifies that TDRK methods are very viable in solving mildly stiff ODEs, as mentioned in the literature [7].

Moreover, in Tables 10 and 11, and Figures 5 and 8, we may realise that the beginning step size used is $h = 0.00625$ and $h = 0.0125$ respectively instead of 0.025 like the rest. This is because during our numerical experiment, we observed that the maximum error of the approximations for MTDRK methods was very large, hence suggesting us to gradually reduce the step size used for the respective experiments in order for our proposed methods to be applicable. This outcome supports our discussion and results on the study of stability region for the MTDRK methods, where

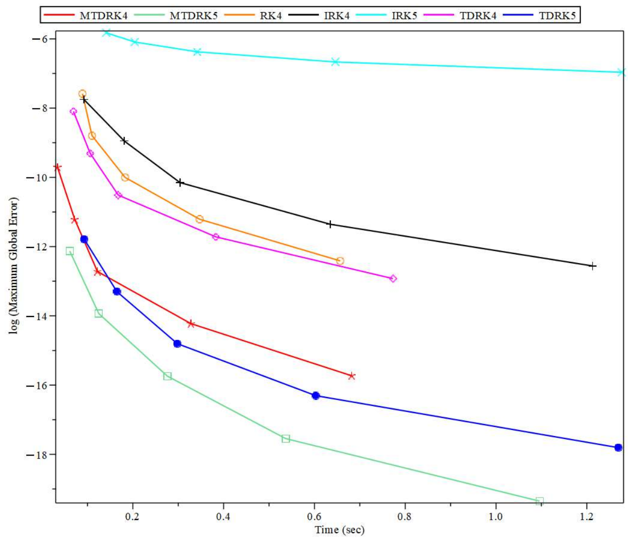


FIGURE 6. Time curves for all methods for Problem 5.4 with $h = 0.0015625(2^i)$, $i = 0, 1, 2, 3, 4$.

TABLE 10. Maximum Error (log10) and Computational Time for Problem 5.5 ($\lambda = 1$).

Method	$h = 0.025$		$h = 0.0125$		$h = 0.00625$		$h = 0.003125$		$h = 0.0015625$	
	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time
MTDRK4	-8.8136	0.013	-10.3176	0.026	-11.8221	0.051	-13.3270	0.107	-14.8320	0.211
MTDRK5	-10.8912	0.020	-12.7121	0.042	-14.5258	0.089	-16.3357	0.199	-18.1438	0.395
RK4	-7.2763	0.033	-8.4948	0.056	-9.7061	0.079	-10.9138	0.103	-12.1197	0.183
IRK4	-7.2550	0.031	-8.4525	0.044	-9.6533	0.100	-10.8558	0.178	-12.0591	0.313
IRK5	-5.8809	0.052	-6.1246	0.071	-6.3996	0.123	-6.6882	0.186	-6.9832	0.339
TDRK4	-7.5168	0.019	-8.7345	0.033	-9.9454	0.062	-11.1529	0.148	-12.3588	0.257
TDRK5	-11.1464	0.028	-12.5184	0.052	-13.9706	0.103	-15.4519	0.236	-16.9457	0.471

TABLE 11. Maximum Error (log10) and Computational Time for Problem 5.5 ($\lambda = 100$).

Method	$h = 0.0125$		$h = 0.00625$		$h = 0.003125$		$h = 0.0015625$		$h = 0.00078125$	
	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time
MTDRK4	-8.4459	0.029	-10.0923	0.058	-11.5976	0.119	-13.1027	0.225	-14.6079	0.484
MTDRK5	-9.4139	0.046	-11.6298	0.092	-13.6777	0.201	-15.6860	0.404	-17.6448	0.847
RK4	-5.0307	0.030	-6.3691	0.047	-7.6402	0.111	-8.8789	0.183	-10.1006	0.352
IRK4	-6.7678	0.049	-8.0582	0.098	-9.2373	0.148	-10.4295	0.282	-11.6277	0.565
IRK5	-6.3578	0.066	-6.6426	0.094	-6.9326	0.181	-7.2295	0.340	-7.5285	0.618
TDRK4	-6.7469	0.061	-8.0850	0.092	-9.3560	0.152	-10.5945	0.289	-11.8162	0.571
TDRK5	-8.2053	0.099	-10.1431	0.136	-12.0346	0.274	-13.9159	0.488	-15.8349	1.124

the error of approximations may grow uncontrollably large when the step-size is too large for certain scenarios due to the relatively small stability regions of our methods.

Problem 5.6. Consider an application which is a nonlinear homogeneous problem, the motion of two springs with respective weights attached to them [13]. In our study, we consider the nonlinear system of the coupled spring model

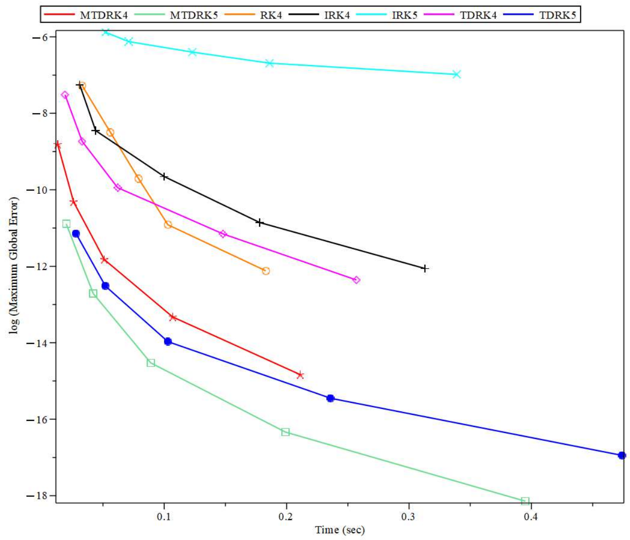


FIGURE 7. Time curves for Problem 5.5 ($\lambda = 1$) with $h = 0.0015625(2^i)$, $i = 0, 1, 2, 3, 4$.

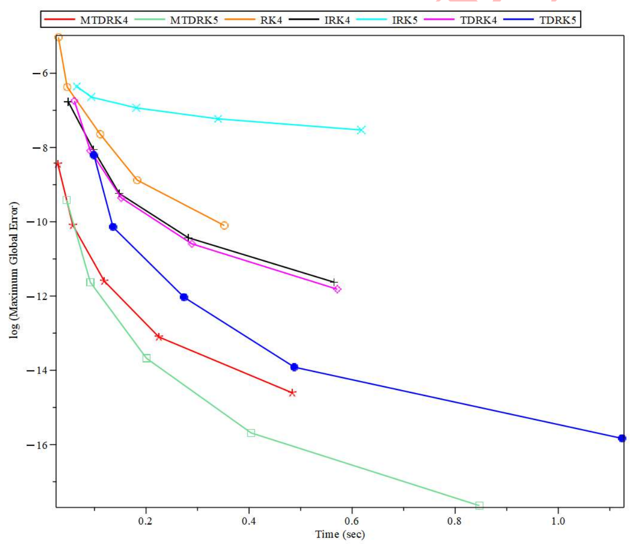


FIGURE 8. Time curves for all methods for Problem 5.5 ($\lambda = 100$) with $h = \frac{1}{80(2^i)}$, $i = 0, 1, 2, 3, 4$.

as follows:

$$\begin{aligned}
x_1' &= u, \\
x_2' &= v, \\
u' &= \frac{1}{m_1}\left(-\delta_1 u - k_1 x_1 + \mu_1 x_1^3 - k_2(x_1 - x_2) + \mu_2(x_1 - x_2)^3\right), \\
v' &= \frac{1}{m_2}\left(-\delta_2 v - k_2(x_2 - x_1) + \mu_2(x_2 - x_1)^3\right).
\end{aligned} \tag{5.6}$$

Table 12. Maximum Error (log10) and Computational Time for Problem 5.6.

Method	$h = 0.025$		$h = 0.0125$		$h = 0.00625$		$h = 0.003125$		$h = 0.0015625$	
	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time	Max Error	Time
MTDRK4	-8.964353717	0.039	-10.46501861	0.081	-11.96792797	0.165	-13.47191207	0.355	-14.97500617	0.733
MTDRK5	-9.828080772	0.068	-11.33608553	0.174	-12.84627309	0.339	-14.34912638	0.674	-15.874165	1.379
RK4	-7.612270426	0.024	-8.81306911	0.063	-10.01557465	0.100	-11.21890241	0.182	-12.42263497	0.404
IRK4	-7.30960804	0.039	-8.511498471	0.076	-9.714474448	0.171	-10.91801521	0.362	-12.12184703	0.738
IRK5	-5.061938327	0.068	-5.378866627	0.108	-5.68770183	0.218	-5.992693773	0.443	-6.295661631	0.879
TDRK4	-7.600475193	0.073	-8.800912417	0.107	-10.00323919	0.233	-11.20647488	0.461	-12.41016211	0.957
TDRK5	-11.07222194	0.095	-12.59398216	0.207	-14.1074599	0.432	-15.60610229	0.855	-16.85860853	1.744

where m, k, x, δ, μ, each represents mass, spring constant, displacement, damping coefficient, and nonlinear coefficient respectively.

Given $m_1 = m_2 = 1$, $k_1 = 0.4$, $k_2 = 1$, $\delta_1 = 0.1$, $\delta_2 = 0.2$, $\mu_1 = \frac{1}{6}$, $\mu_2 = 0.1$, the problem is integrated on the interval $[0, 5]$ with the initial conditions $(x_1(0), x_2(0), u(0), v(0)) = (0.7, 0.1, 0, 0)$,and the problem is used to model the motion of two springs with weights attached and hung in series from the ceiling where damping and forced motion due to external forces are considered. This equation has no analytical solution. Hence, Runge–Kutta of order 4 method is used to obtain numerical approximation as the reference solution for this problem. The computation of solution using step sizes of $h = 0.0001$ and $h = 0.00001$ is repeated and the numerical difference of x_n, y_n, u_n and v_n between the two step sizes are compared respectively. Therefore, we have:

$$\begin{aligned} &\epsilon_x = -6.3447722547752 \times 10^{-18}; \epsilon_y = 5.488055106791 \times 10^{-18}; \\ &\epsilon_u = -3.761020468923 \times 10^{-18}; \epsilon_v = 2.437447085636 \times 10^{-18}. \end{aligned} \tag{5.7}$$

Since the difference betwwen them are so small and could be considered neglegible, the numerical approximation with $h = 0.0001$ is used as the reference solution.

The model is constructed based on the problem of two springs and two weights attached in series, hanging from the ceiling. By assuming that the restoring forces behave according to Hooke's Law, the model is linear where each weight is described by a coupled, second-order, linear differential equation. However, the restoring force tends to be more physically meaningful when factors such as damping which maybe encountered in the beginning courses, and slight nonlinearity which is most certainly true for large vibrations, are incorporated to the model. Therefore, in this paper, we utilised the modified model which had considered damping and nonlinearity, making it more meaningful and relatable to real-life.

In order to display the characteristics of the damping and nonlinearity coefficients, a few plots are illustrated as Figures 9–11. The simulated plots utilized the same parameters as Problem 5.6 except for the damping and nonlinearity coefficients.

In Figure 9, it can be observed that without damping and nonlinearity, the oscillations of x_1 and x_2 are somehow periodic, indicating that the motion of spring is consistent throughout the time which is dissimilar to real-life motions. However, in Figures 10–11, we can see that the inclusion of δ and μ displays a plot describing that the motion of spring will eventually die-off with time, which is more appropriate and realistic. Furthermore, in Figure 11, it can be seen that the higher values of μ tend to cause more fluctuations and inconsistencies in the oscillations of x_1 and x_2.

For Problem 5.6, we can see that our methods outperform most of the methods of similar order, except for TDRK5. However, this does not contradict our results as it is not surprising for the TDRK5 method to have better performance as it has four stages, which is one more than our MTDRK5 method. On the other hand, this further justifies our previous statements for Problem 5.1–5.5 as our methods with lesser funtion evaluation generally have better numerical performance than methods of similar order but with higher stage.

Overall, as our proposed methods are able to perform similarly to their respective compared methods of same or higher orders but with lesser function evaluations and lower stages, it is sufficient to say that they are numerically more proficient in solving initial value problems in the form of $y' = f(x, y)$. Moreover, they are proven to show an

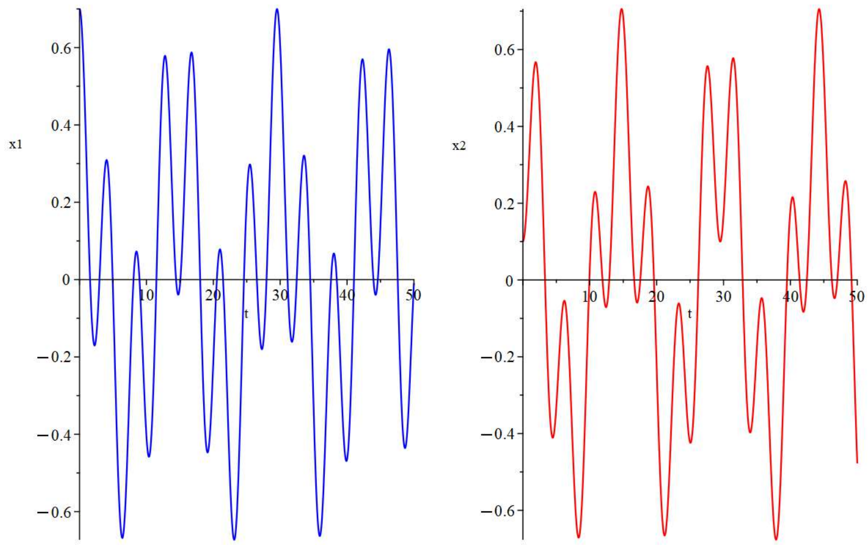


FIGURE 9. Plots of x_1 and x_2 without damping and nonlinearity.

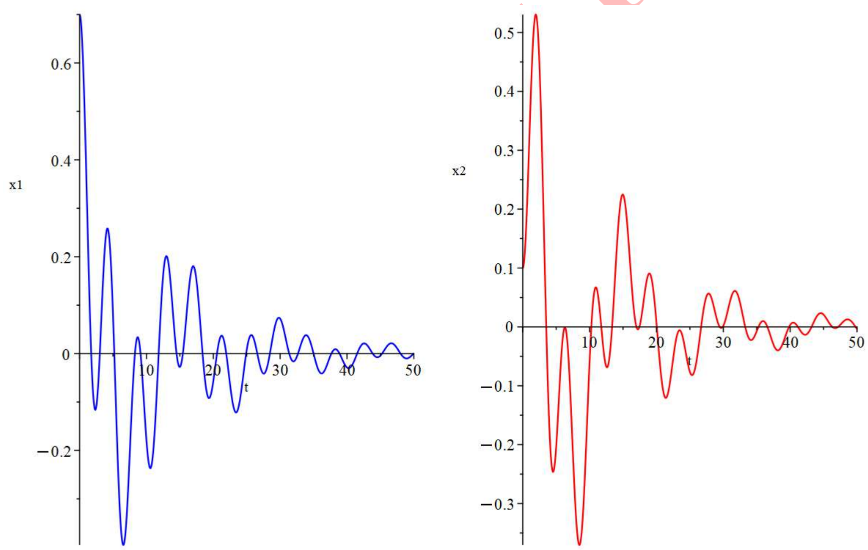


FIGURE 10. Plots of x_1 and x_2 with $\delta_1 = 0.1$, $\delta_2 = 0.2$, $\mu_1 = \frac{1}{6}$, $\mu_2 = 0.1$.

improvement in efficiency as they take lesser computational time than the existing TDRK methods whilst retaining a higher accuracy. At the same time, they are highly effective in solving non-stiff and mildly stiff ODEs.

However, as they are proficient in solving first-order ODEs, they have their drawbacks, which they have a bounded stability region, which may limit their effectiveness in solving stiff problems, which usually requires the property of A-stability, where the roots of the stability polynomial (4.3) covers the entire left-half plane of the stability region. This is an important criterion in solving stiff ODEs [9] which our proposed methods do not have.

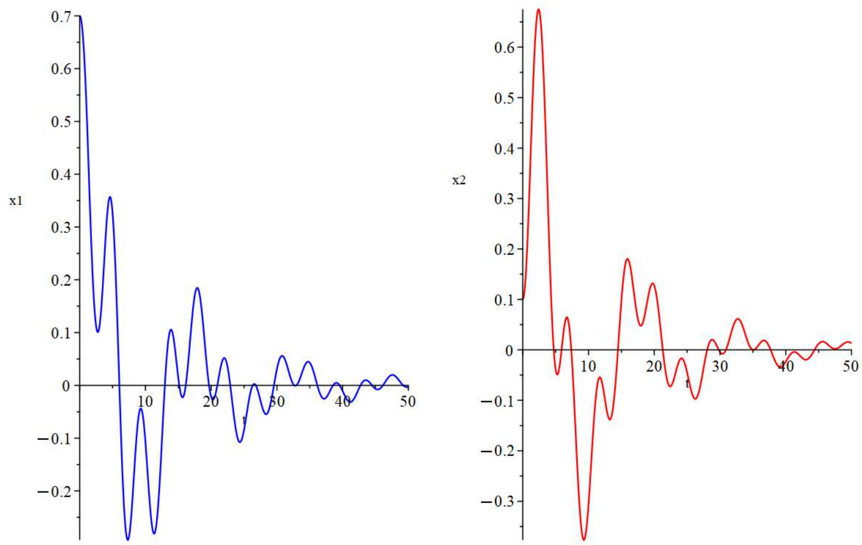


FIGURE 11. Plots of x_1 and x_2 with $\delta_1 = 0.1$, $\delta_2 = 0.2$, $\mu_1 = 1$, $\mu_2 = 0.5$.

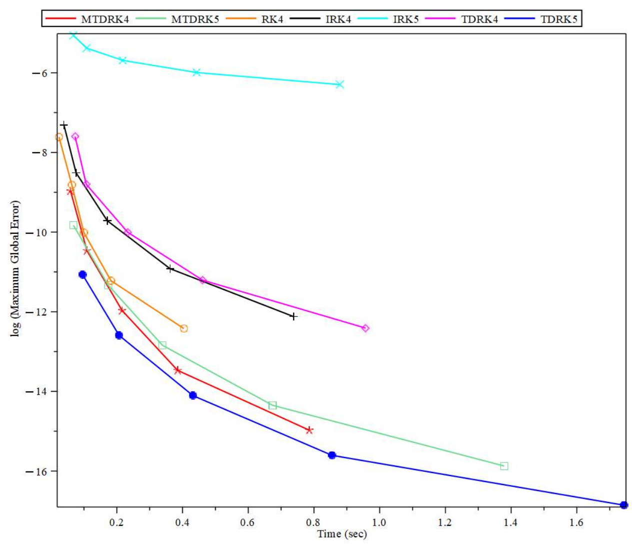


FIGURE 12. Time curves for all methods for Problem 5.6 with $h = 0.0015625(2^i)$, $i = 0, 1, 2, 3, 4$.

6. CONCLUSION

A group of multistep explicit two-derivative Runge–Kutta (MTDRK) methods that involve one f-evaluation and multiple g-evaluations, whilst incorporating the value of 2 step-points x_i and x_{i-1} for solving initial value problems in the form of $y' = f(x, y)$ were presented. These newly proposed methods are extensions from direct methods that are two-step and two-derivative for solving $y' = f(x, y)$.

In this study, two-stages of order four and three-stages of order five, denoted as MTDRK4 and MTDRK5 methods, are presented based on the algebraic order conditions derived from Taylor series expansion to solve first order ODEs directly. The properties of stability for both MTDRK methods are analysed.

From the numerical results and analysis, we observe that both MTDRK4 and MTDRK5 methods have higher efficiency than the other methods. This is because the MTDRK methods incorporate an extra derivative of f-evaluation and the information of another step-point in order to solve first-order ODEs. Moreover, as MTDRK methods are a class of multi-step methods, it can clearly be seen to be competitive or perform even better for certain types of mildly stiff problems. However, there are some drawbacks of the MTDRK methods that still exist. As the complexity of function evaluation is high, it eventually leads to a high computational time in solving particular problems. Furthermore, the stability region for the MTDRK methods is relatively small, which indicates that the truncation error has a tendency to be uncontrollably high if the step size is too large.

In light of this research, a number of intriguing topics can be explored. Firstly, implicit MTDRK methods can be investigated and constructed as it can reach a higher order than explicit methods. Also, fitting techniques such as exponentially-fitting and trigonometrically-fitting techniques can be adapted into our proposed methods to make it more efficient for solving first-order ODEs with oscillatory solutions. Furthermore, our MTDRK methods may be considered in solving differential equations of various types such as partial differential equations (PDE) and delay differential equations (DDE). This may be significant as the applicability of the MTDRK methods to solve other types of differential equations allows them to be utilised in different fields of study such as the heat equation in enginnering which consists of partial differential equations.

References

[1] N. Y. Abdul-Hassan and Z. J. Kadum and A. H. Ali, *An efficient third-order scheme based on Runge–Kutta and Taylor series expansion for solving initial value problems*, Algorithms, *123*(3) (2024), 123.

[2] N. A. Ahmad and N. Senu, *Two derivative runge-kutta method with fsal property for the solution of first order initial value problems*, Indian J. Sci. Technol., *9*(28) (2016), 1-8.

[3] M. A. Akanbi, *Propagation of errors in euler methods*, Arch. Appl. Sci. Res., *2*(2) (2010), 1457-469.

[4] U. M. Ascher and Linda R. Petzold, *Computer methods for ordinary differential equations and differential algebraic equations*, Cambridge University Press, Cambridge, 1998.

[5] N. H. A. Aziz, L. Mohamed Faris, and M. A. Zanariah, *Numerical approaches of block multistep method for propagation of derivatives discontinuities in neutral delay differential equations*, Alex. Eng. J., *75* (2023), 577-588.

[6] J. C. Butcher, *Numerical methods for ordinary differential equations*, John Wiley & Sons, Ltd, 2nd ed., 2008.

[7] R. P. K. Chan and A. Y. J. Tsai, *On explicit two-derivative runge-kutta methods*, Numer. Algorithms, *53* (2010), 171-194.

[8] R. P. K. Chan, S. Wang, and A. Y. J. Tsai, *Two-derivative runge-kutta methods for differential equations*, AIP Conf. Proc., *1479* (2012), 262-266.

[9] R. M. Corless, C. Y. Kaya, and R. H. C. Moir, *Optimal residuals and the dahlquist test problem*, Numer. Algorithms, *81* (2019), 1253-1274.

[10] G. G. Dahlquist, *A special stability problem for linear multistep methods*, BIT, *3* (1963), 27-43.

[11] M. A. Darus, N. H. A. Aziz, F. Deraman, A. Salina, M. S. Anuar, and H. L. Zakaria *Numerical approximation of volterra integro-differential equations of the second kind using Boole's quadrature rule method*, Math. Stat., *11*(3) (2023), 566-573.

[12] K. Elnour, *Comparative studies between picard's and taylor's methods of numerical solutions of first ordinary order differential equations arising from real-life problems*, J. Appl. Math. Phys., *12* (2024), 877-896.

[13] T. H. Fay and S. D. Graham, *Coupled spring equations*, Int. J. Math. Educ. Sci. Technol., *34* (2003), 65-79.

[14] D. Goeken and O. Johnson, *Runge-kutta with higher order derivative approximations*, Appl. Numer. Math., *34* (2000), 207-218.

[15] S. Gurvinder, *Comparison between various methods for solving first order differential equations*, Int. J. Electron. Eng., *11* (2019), 291-298.

[16] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer, 1st ed., 1993.

[17] T. E. Hull, W. H. Enright, B. M. Fellen, and A. E. Sedgwick, *Comparing numerical methods for ordinary differential equations*, J. Numer. Anal., *9* (1982), 603-637.

[18] F. Ismail, F. Rabiei, and M. Suleiman, *Improved runge-kutta methods for solving ordinary differential equations*, Aust. J. Basic Appl. Sci., *42* (2013), 1679-1687.

[19] D. A. Jemal, *Comparison of numerical methods for system of first order ordinary differential equations*, Pure Appl. Math. J., *9* (2020), 32-36.

[20] Z. Kalogiratou, T. Monovasilis, and T.E. Simos, *Construction of two derivative runge kutta methods of order five*, AIP Conf. Proc., *1863* (2017), 560092.

[21] Z. Kalogiratou, T. Monovasilis, and T.E. Simos, *New fifth-order two-derivative runge kutta methods with constant and frequency-dependent coefficients*, Math. Methods Appl. Sci., *42*(6) (2019), 1955-1966.

[22] Z. Kalogiratou, T. Monovasilis, and T.E. Simos, *Two derivative Runge-Kutta methods of order six*, AIP Conf. Proc., *2116* (2019), 450106.

[23] D. I. Lanlege, U. M. Garba, and A. Aluebho, *Using modified euler method (mem) for the solution of some first order differential equations with initial value problems (ivps)*, Pac. J. Sci. Technol., *16* (2015), 63-81.

[24] K. C. Lee, M. A. Alias N. Senu, and A. Ahmadian, *On efficient frequency-dependent parameters of explicit two-derivative improved runge-kutta-nyström method with application to two-body problem*, Alex. Eng. J., *72* (2023), 605-620.

[25] K. C. Lee, N. Senu, Ali, A. Ahmadian, and S. N. I. Ibrahim, *Efficient frequency-dependent coefficients of explicit improved two-derivative runge-kutta type Methods for solving third-order ivps.*, Pertanika J. Sci. Technol., *31*(2) (2023), 843-873.

[26] T. Monovasilis and Z. Kalogiratou, *High order two-derivative runge-kutta methods with optimized dispersion and dissipation error*, Mathematics, *9* (2021), 232-242.

[27] P. Phohomsiri and F. E. Udwadia, *Acceleration of runge-kutta integration schemes*, Discrete Dyn. Nat. Soc., *2004*(2) (2004), 307-314.

[28] F. Rabiei and F. Ismail, *Third-order improved runge-kutta method for solving ordinary differential equation*, Int. J. Appl. Phys. Math., *1* (2011), 191-194.

[29] F. Rabiei and F. Ismail, *Fifth-order improved runge-kutta method with reduced number of function evaluations*, Aust. J. Basic Appl. Sci., *6* (2012), 97-105.

[30] H. Seka and K. R. Assui, *Order of the runge-kutta method and evolution of the stability region*, Ural Math. J., *5* (2019), 64-71.

[31] H. Seka and K. R. Assui, *Order of the runge-kutta method and evolution of the stability region*, Ural Math. J., *5* (2019), 64-71.

[32] N. Senu, N. A. Ahmad, and N. Bachok, *Fifth order two derivative runge-kutta method with reduced function evaluations for solving ivps.*, Int. J. Math. Comput. Methods, *1* (2016), 421-428.

[33] M. Sezer and A. Akyüz-Daşcıog˘lu, *A taylor method for numerical solution of generalized pantograph equations with linear functional argument*, J. Comput. Appl. Math., *200* (2007), 217-225.

[34] L. F. Shampine, *Numerical solution of ordinary differential equations*, Chapman & Hall, London, 1994.

[35] F. E. Udwadia and A. Farahani, *Accelerated runge-kutta methods*, Discrete Dyn. Nat. Soc., *2008*(1) (2008), 790619.

[36] X. Wu, *A class of runge-kutta formulae of order three and four with reduced evaluations of function*, Appl. Math. Comput., *146*, (2003), 417-432.

[37] X. Wu and J. Xia *Extended runge-kutta-like formulae*, Appl. Numer. Math., *56*, (2006), 1584-1605.

[38] N. A. M. Zain, R. Daud, N. H. A. Aziz, K. R. Ahmad, A. A. Ismail, and B. Izzawati, *Stress analysis prediction on screw orthopedic implant in trabecular bone*, Mater. Today Proc., *16* (2019), 1838-1845.

[39] L. Zhang and T. Guan, *Comparison of several numerical algorithms for solving ordinary differential equation initial value problem*, Atlant. Press, *78* (2018), 454-458.