Mathematics and Computers in Simulation 219 (2024) 28–49

Contents lists available at ScienceDirect

Mathematics and Computers in Simulation

journal homepage: www.elsevier.com/locate/matcom

Original articles

A promising exponentially-fitted two-derivative Runge–Kutta–Nyström method for solving $y'' = f(x, y)$: Application to Verhulst logistic growth model

K.C. Lee [a,*], R. Nazar [a], N. Senu [b,c], A. Ahmadian [d,e,f]

[a] Department of Mathematical Sciences, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia
[b] Institute for Mathematical Research, Universiti Putra Malaysia, 43400 UPM, Serdang, Malaysia
[c] Department of Mathematics and Statistics, Universiti Putra Malaysia, 43400 UPM, Serdang, Malaysia
[d] Decisions Lab, Mediterranea University of Reggio Calabria, Reggio Calabria, Italy
[e] Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon
[f] Faculty of Engineering and Natural Sciences, Istanbul Okan University, Istanbul, Turkey

ARTICLE INFO

Keywords:
Two-derivative Runge–Kutta–Nyström method
Second-order ordinary differential equations
Exponentially-fitted
Stability region
Numerical test

ABSTRACT

Explicit exponentially-fitted two-derivative Runge–Kutta–Nyström method with single f-function and multiple third derivatives is proposed for solving special type of second-order ordinary differential equations with exponential solutions. B-series and rooted tree theory for the proposed method are developed for the derivation of order conditions. Then, we build frequency-dependent coefficients for the proposed method by integrating the second-order initial value problem exactly with solution in the linear composition of set functions $e^{\lambda t}$ and $e^{-\lambda t}$ with $\lambda \in \mathbb{R}$. An exponentially-fitted two-derivative Runge–Kutta–Nyström method with three stages fifth order is derived. Linear stability and stability region of the proposed method are analyzed. The numerical tests show that the proposed method is more effective than other existing methods with similar algebraic order in the integration of special type of second-order ordinary differential equations with exponential solutions. Also, the proposed method is used to solve a famous application problem, Verhulst logistic growth model and the result shows the proposed method still works effectively for solving this model.

1. Introduction

This article focuses on the numerical solution of special class of second-order ordinary differential equations (ODEs) with exponential solution in the form of:

$$\begin{cases} y''(x) = f(x, y(x)), \\ y(x_0) = y_0, \quad y'(x_0) = y'_0, \quad x \in [x_0, x_{end}], \end{cases} \tag{1.1}$$

where $y : \mathbb{R} \to \mathbb{R}^N$, $f : \mathbb{R} \times \mathbb{R}^N \to \mathbb{R}^N$ is a continuous vector function.

Second-order ODEs are vastly used in modeling and forecasting the trend of scientific phenomena and application problems particularly in engineering and physics fields, such as electric circuits, damped oscillation and vibration, Pleiades constellation, classical mechanics and quantum mechanics [8,13,15,20]. The traditional approach for integrating high-order ODEs is by reducing

* Corresponding author.
E-mail address: kclee_1017@ukm.edu.my (K.C. Lee).

https://doi.org/10.1016/j.matcom.2023.12.018
Received 20 July 2023; Received in revised form 14 November 2023; Accepted 12 December 2023
Available online 15 December 2023

them into a system of first-order ODEs and solving them subsequently with numerical approaches, comprised of the Runge–Kutta method, block method, power series method, predictor–corrector method and others. Meanwhile, this approach can result in higher computational cost and truncational error, which leads to inaccuracy of numerical approximation. Thus, the direct method is more significantly commonly to be used to solve high-order ODEs directly [2,4,10].

Many studies have been conducted to develop effective techniques for integrating first or high-order ODEs with specific solutions or properties [3,19,21,24]. Some researchers focused on studying the behaviors and properties of ODEs and there are some interesting properties such as the growth of dependent variables undergoing exponential and periodic patterns. It causes a lot of researchers to start developing efficient Runge–Kutta methods or one-step methods with dependent coefficients through exponential fitting approaches. Berghe et al. [5] introduced s-stage exponentially-fitted Runge–Kutta methods, with the lower bound being related to explicit methods and the upper bound applicable to collocation methods. Error and step-size is controlled based on Richardson extrapolation and their proposed method is very efficient in solving first-order ODEs with exponential solutions. Later, Franco [12] extended their works by introducing an embedded pair of explicit Runge–Kutta methods with exponentially-fitted variants to solve first-order ODEs. The embedded Runge–Kutta method can control the step size dynamically based on error estimation. Similarly, Ixaru [18] focused on developing two and three-stage explicit Runge–Kutta methods and implemented an exponential fitting approach into them to solve first-order ODEs effectively. Enhancement has been made by extending the A-stability property and his methods are proven as efficient tools for solving stiff first-order ODEs.

Apart from explicit Runge–Kutta methods, diagonally implicit and fully implicit Runge–Kutta methods can be adapted with exponentially-fitting techniques. Ehigie et al. [11] proposed two-derivative diagonally implicit Runge–Kutta methods through the theorem involving bi-colored rooted trees and elementary weights. Exponential fitting conditions are adapted into proposed methods with fourth, fifth and sixth-order. Phase-lag and stability analysis of the methods are investigated to obtain the optimized scheme and numerical experiments exhibit the efficiency of proposed methods. Besides Runge–Kutta methods, block methods which are considered popular multistep methods can be fitted with exponential properties through the implementation of exponential integration in the polynomial time integration framework. A new class of parallel exponential polynomial block methods are constructed based on Legendre points for solving unpartitioned and partitioned first-order initial value problems [7].

Other than the integration of first-order ODEs, there is a lot of academics were extending previous work on developing and analyzing exponentially-fitted Runge–Kutta and Runge–Kutta–Nyström methods for solving a general class of second-order ODEs with exponential solutions with extremely high precision [1,23]. Mohamed et al. [22] developed exponentially-fitted two-derivative Runge–Kutta–Nyström method with two-stage fourth-order and three-stage fifth-order for solving general second-order ODEs. Then, Zhai and Chen [28] added symmetric and symplectic properties into the classical Runge–Kutta–Nyström method and implemented the exponentially-fitting technique into it. The new integrator integrates exactly second-order differential equations with solutions that are represented as linear combinations of the functions from the set $\{e^{\lambda t}, e^{-\lambda t} | \lambda \in \mathbb{C}\}$. Local truncation errors of the proposed method and numerical tests are examined to verify the efficiency of the method.

Based on past research, we have observed a gap in the development of efficient Runge–Kutta or Runge–Kutta–Nyström methods for solving a specific type of second-order ODEs, represented as $y'' = f(x, y)$. Not all second-order application problems are formulated in the general form, and some are simplified, involving only the variable x and y in the function evaluation, omitting the y' term. Additionally, we believe that there is room for improvement in the application of exponentially-fitting techniques to make them more suitable and effective for solving $y'' = f(x, y)$ with exponential solutions. To address this gap and build upon previous research on deriving Runge–Kutta or Runge–Kutta–Nyström methods and implementing exponentially-fitting techniques, we propose an efficient two-derivative Runge–Kutta–Nyström method with minimal function evaluations and implement exponentially-fitting technique into it, which is the main objective of this research, with the motivation to improve the accuracy and precision of existing methods for solving the specific class of ODEs with exponential solutions.

Here we highlight the major contributions and findings of this article as follows:

- Presented the derivation of two-derivative Runge–Kutta–Nyström method with minimal function evaluation for solving $y''(x) = f(x, y)$.
- Demonstrated the implementation of exponentially-fitting technique into the two-derivative Runge–Kutta–Nyström method.
- Presented the stability analysis of two-derivative Runge–Kutta–Nyström method.
- Verified the numerical efficiency of two-derivative Runge–Kutta–Nyström method with exponentially-fitting technique for solving $y''(x) = f(x, y)$ with exponential solutions.

In this study, the conventional two-derivative direct Runge–Kutta–Nyström scheme is changed by replacing the multiple increment function by single second derivative function, f-evaluation and multiple third derivative functions, g-evaluation remain in the formulation. Three-stage fifth-order two-derivative Runge–Kutta–Nyström (TDRKN) method with a minimal number of function evaluations is derived based on the rooted tree theory in Section 2. Then, the exponentially-fitting technique is implemented into the parameters of the proposed method for transforming them into frequency-dependent and customized to solve the exponential problem in Section 3. The second-order initial value problem has been integrated exactly with a solution in linear composition of set functions $e^{\lambda t}$ and $e^{-\lambda t}$ with $\lambda \in \mathbb{R}$. Linear stability and stability region of the proposed method are investigated to study the uniform bound for stability. Section 4 displays numerical tests of the proposed method compared to other existing methods with similar properties and order. Discussion of the numerical results is presented in Section 5 and this article ends in Section 6 with a conclusion.

Table 1
TDRKN methods in Butcher tableau.

c	A	$\hat{A}$
	b^T	b'^T

2. The formulation of TDRKN method

TDRKN method is derived with the inclusion of a third derivative, $y'''(x)$ in the formulation as shown below:

$$y'''(x) = g(x, y, y') = f_x(x, y, y') + f_y(x, y, y')y'. \tag{2.2}$$

s-stage TDRKN method consists of multiple evaluations in the third derivative and a single evaluation in the second derivative that is expressed as follows:

$$\begin{aligned} y_{n+1} &= y_n + hy'_n + \frac{h^2}{2} f(x_n, y_n) + h^3 \sum_{i=1}^{s} b_i g\left(x_n + c_i h, Y_i, Y'_i\right), \\ y'_{n+1} &= y'_n + hf(x_n, y_n) + h^2 \sum_{i=1}^{s} b'_i g\left(x_n + c_i h, Y_i, Y'_i\right), \end{aligned}$$

where

$$\begin{aligned} Y_i &= y_n + c_i hy'_n + \frac{(c_i h)^2}{2} f(x_n, y_n) + h^3 \sum_{j=1}^{s} A_{i,j} g\left(x_n + c_j h, Y_j, Y'_j\right), \\ Y'_i &= y'_n + c_i hf(x_n, y_n) + h^2 \sum_{j=1}^{s} \hat{A}_{i,j} g\left(x_n + c_j h, Y_j, Y'_j\right), \end{aligned} \tag{2.3}$$

where $c_i, b_i, b'_i, b''_i, A_{i,j} \hat{A}_{i,j} \in \mathbb{R}, i, j = 1, 2, \dots, s \in \mathbb{Z}^+$. Also, Eqs. (2.3) can be expressed in Butcher's tableau (see Table 1):

TDRKN methods are implicit methods if $A_{i,j}$ and $\hat{A}_{i,j}$ are not equal to 0 for $i \leq j$, and are explicit methods otherwise. TDRKN method, in contrast to the conventional two-derivative Runge–Kutta–Nyström approach, consist of one function evaluation of f and numerous function evaluations of g per step, and thus, having fewer total function evaluations than the existing two-derivative Runge–Kutta–Nyström method which consists of numerous f and g evaluations per step [8].

Here are some key assumptions that are aligned with the objectives of the study as follows:

- B-series and rooted tree theorem can be derived for two-derivative Runge–Kutta–Nyström with minimal function evaluation with the motivation of generating order conditions for y and y'.
- Exponentially-fitting technique can be implemented to two-derivative Runge–Kutta–Nyström with minimal function evaluation.

3. Construction of TDRKN methods

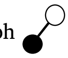
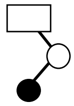
In this section, all coefficients of TDRKN methods, $c_i, b_i, b'_i, A_{i,j} \hat{A}_{i,j}$ will be found in this part. For obtaining a general formula for higher derivatives of the analytical solution of problem (2.3), we consider the expression of first to seventh derivatives of the analytical solution $y(x)$ at $x = x_0$.

$$\begin{aligned} y^{(1)} =& y', \quad y^{(2)} = f, \quad y^{(3)} = g, \quad y^{(4)} = g'_y y' + g'_{y'} f, \\ y^{(5)} =& g^{(2)}_{yy} y'^2 + 2g^{(2)}_{yy'} y' f + g^{(2)}_{y'y'} f^2 + g'_y f + g'_{y'} g, \\ y^{(6)} =& g^{(3)}_{yyy} y'^3 + 3g^{(3)}_{yyy'} y'^2 f + 3g^{(3)}_{yy'y'} y' f^2 + g^{(3)}_{y'y'y'} f^3 + 3g^{(2)}_{yy} y' f + 3g^{(2)}_{yy'} y' g + 3g^{(2)}_{y'y'} fg + g'_y g \\ &+ g'_{y'}(g'_y y' + g'_{y'} f) + 3g^{(2)}_{yy'} f^2, \\ y^{(7)} =& 12g^{(3)}_{yy'y'} y' fg + g^{(4)}_{yyyy} y'^4 + 3g^{(2)}_{yy} f^2 + 6g^{(3)}_{yy'y'} f^3 + g^{(4)}_{y'y'y'y'} f^4 + 3g^{(2)}_{y'y'} g^2 + g'_y(g'_y y' + g'_{y'} f) \\ &+ g'_{y'}\left(g^{(2)}_{yy} y'^2 + 2g^{(2)}_{yy'} y' f + g^{(2)}_{y'y'} f^2 + g'_y f + g'_{y'} g\right) + 6g^{(3)}_{y'y'y'} f^2 g + 4g^{(2)}_{y'y'}(g'_y y' + g'_{y'} f) \\ &+ 4g^{(4)}_{yyyy'} y'^3 f + 6g^{(3)}_{yyy} y'^2 f + 6g^{(4)}_{yyy'y'} y'^2 f^2 + 6g^{(3)}_{yyy'} y'^2 g + 12g^{(3)}_{yyy'} y' f^2 + 4g^{(2)}_{yy} y' g \\ &+ 4g^{(4)}_{yy'y'y'} y' f^3 + 4g^{(2)}_{yy'} y'(g'_y y' + g'_{y'} f) + 10g^{(2)}_{yy'} fg. \end{aligned} \tag{3.4}$$

3.1. Rooted trees and B-series theory

Fundamental of the set of rooted trees for TDRKN method are described and analyzed as follows:

Definition 3.1. The set R_T of rooted trees is recursively interpreted as

(i) The graph  indicated as τ_1, belongs to rooted tree; the graph  indicated as τ_2 and the graph  indicated as τ_3;

(ii) If $t_1, \ldots, t_\alpha, t_{\alpha+1}, \ldots, t_\beta \in R_T, t_{\alpha+1}, \ldots, t_\beta$ different from τ_1, then the root-connection graph with $t_1, \ldots, t_\alpha$ linking downwards to a new white rectangle vertex, combining the roots of $t_{\alpha+1}, \ldots, t_\beta$ into the white rectangle vertex, and subsequently linking downwards to a new white circle vertex and to a new black circle vertex, belongs to R_T. It is expressed as

$$t = \left[t_1, \ldots, t_\alpha, \langle t_{\alpha+1}, \ldots, t_\beta\rangle\right]_2, \tag{3.5}$$

in which the new black circle vertex is the roots of the R_T.

Definition 3.2. Order of integer-value function $\rho : R_T \to \mathbb{N}$ is expressed recursively as:

(i) $\rho(\tau_1) = 1,\ \rho(\tau_2) = 2,\ \rho(\tau_3) = 3,$

(ii) for $t = \left[t_1, \ldots, t_\alpha, \langle t_{\alpha+1}, \ldots, t_\beta\rangle\right]_2 \in R_T$,

$$\rho(t) = 3 + \sum_{i=1}^{\alpha} \rho(t_i) + \sum_{i=\alpha+1}^{\beta} (\rho(t_i) - 1). \tag{3.6}$$

For every $t \in R_T$, the order ρ is the number of tree vertices. The set of all rooted trees of order k is represented by the notation RT_k.

Definition 3.3. For every tree $t \in R_T$, the fundamental differential is a vector function $G(t) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ expressed as follows:

(i) $G(\tau_1)(y, y') = y',\ G(\tau_2)(y, y') = f(y),\ G(\tau_3)(y, y') = g(y, y'),$

(ii) for $t = \left[t_1, \ldots, t_\alpha, \langle t_{\alpha+1}, \ldots, t_\beta\rangle\right]_2 \in R_T$

$$G(t)(y, y') = \frac{\partial^\beta g}{\partial y^\alpha \partial y'^{\beta-\alpha}} \left(y, y'\right) \left[G(t_1)(y, y'), \ldots, G(t_\beta)(y, y')\right]. \tag{3.7}$$

Definition 3.4. An integer function, $\sigma : R_T \to \mathbb{N}$ is recursively described as follows:

(i) $\sigma(\tau_1) = \sigma(\tau_2) = \sigma(\tau_3) = 1,$

(ii) for $t = \left[t_1^{\mu_1}, \ldots, t_\alpha^{\mu_\alpha}, \langle t_{\alpha+1}^{\mu_{\alpha+1}}, \ldots, t_\beta^{\mu_\beta}\rangle\right] \in R_T$, with $t_1, \ldots, t_\alpha$ and $t_{\alpha+1}, \ldots, t_\beta$ distinct,

$$\sigma(t) = \prod_{i=1}^{n} \mu_i! \left(\sigma(t_i)^{\mu_i}\right), \tag{3.8}$$

in which μ_i is the multiplicity of t_i for $i = 1, \ldots, \beta$.

Theorem 3.1. *Given the analytical solution $y(x_0 + h)$ of Eq. (1.1) as a B-series, $B(e, y_0, y_0')$ with real function e prescribed on $R_T \cup \{\emptyset\}$, then*

$$e(\emptyset) = e(\tau_1) = 1,\ e(\tau_2) = \frac{1}{2},$$

and for $t = \left[t_1, \ldots, t_\alpha, \langle t_{\alpha+1}, \ldots, t_\beta\rangle\right]_2$,

$$e(t) = \frac{1}{\rho(t)(\rho(t) - 1)} \prod_{i=1}^{\alpha} e(t_i) \prod_{i=\alpha+1}^{\beta} \rho(t_i) e(t_i).$$

Proof. By assumption,

$$\begin{aligned} y(x_0 + h) =& e(\emptyset) y_0 + h e(\tau_1) y'(0) + h^2 e(\tau_2) f(y_0) + \sum_{t \in R_T \setminus \{\tau_1, \tau_2\}} \frac{h^{\rho(t)}}{\sigma(t)} e(t) G(t)(y_0, y_0'), \\ =& B(e, y_0, y_0'). \end{aligned} \tag{3.9}$$

The derivatives of $y(x_0 + h)$ are shown below

$$\begin{aligned} \left(y(x_0 + h)\right)' =& e(\tau_1) y'(0) + 2h e(\tau_2) f(y_0) + \sum_{t \in R_T \setminus \{\tau_1, \tau_2\}} \frac{\rho(t) h^{\rho(t)-1}}{\sigma(t)} e(t) G(t)(y_0, y_0'), \\ =& B\left(\frac{\rho}{h} e, y_0, y_0'\right). \\ \left(y(x_0 + h)\right)'' =& 2e(\tau_2) f(y_0) + \sum_{t \in R_T \setminus \{\tau_1, \tau_2\}} \frac{\rho(t)(\rho(t) - 1) h^{\rho(t)-2}}{\sigma(t)} e(t) G(t)(y_0, y_0') \\ =& B\left(\frac{\rho(\rho - 1)}{h^2} e, y_0, y_0'\right). \end{aligned} \tag{3.10}$$

Besides,

$$g\left(B\left(e,y_0,y_0'\right),B\left(\frac{\rho}{h}e,y_0,y_0'\right)\right)=\sum_{t\in R_T\setminus\{\tau_1,\tau_2\}}\frac{h^{\rho(t)-3}}{\sigma(t)}e''(t)G(t)(y_0,y_0'), \tag{3.11}$$

where $t=\left[t_1,\ldots,t_\alpha,\langle t_{\alpha+1},\ldots,t_\beta\rangle\right]_2\in R_T\setminus\{\tau_1,\tau_2\}$ and $e''(t)=\prod_{i=1}^{\alpha}e(t_i)\prod_{i=\alpha+1}^{\beta}\rho(t_i)e(t_i)$.

Combining (3.7) and (3.8) into the problem (1.1) and coefficients of the fundamental differential are compared on both sides, yield

$$e(\tau_2)=\frac{1}{2},$$

and for $t=\left[t_1,\ldots,t_\alpha,\langle t_{\alpha+1},\ldots,t_\beta\rangle\right]_2\in R_T\setminus\{\tau_1,\tau_2\}$,

$$e(t)=\frac{1}{\rho(t)(\rho(t)-1)}\prod_{i=1}^{\alpha}e(t_i)\prod_{i=\alpha+1}^{\beta}\rho(t_i)e(t_i). \quad \square$$

Through Taylor series expansion of $y(x_0+h)$ around $h=0$,we get that $e(\emptyset)=e(\tau_1)=1$.

For every $t\in R_T$, a density function and a positive integer-valued integer can be defined as $\gamma(t)=\frac{1}{e(t)}$ and $\xi(t)=\frac{\rho(t)!}{\sigma(t)\gamma(t)}$. Two propositions can be derived based on Theorem 3.1.

Proposition 3.1. *Density, $\gamma(t)$ for every tree $t\in R_T$, is defined as positive integer function on set R_T*

(i) $\gamma(\tau_1)=1,\ \gamma(\tau_2)=2,\ \gamma(\tau_3)=6$,
(ii) for $t=\left[t_1,\ldots,t_\alpha,\langle t_{\alpha+1},\ldots,t_\beta\rangle\right]_2\in R_T, \gamma(t)=\rho(t)(\rho(t)-1)\prod_{i=1}^{\alpha}\gamma(t_i)\prod_{i=\alpha+1}^{\beta}\frac{\gamma(t_i)}{\rho(t_i)}$.

Proposition 3.2. *Positive integer $\xi(t)$ for every tree $t\in R_T$ is defined as*

(i) $\xi(\tau_1)=\xi(\tau_2)=\xi(\tau_3)=1$,
(ii) for $t=\left[t_1^{\mu_1},\ldots,t_\alpha^{\mu_\alpha},\langle t_{\alpha+1}^{\mu_{\alpha+1}},\ldots,t_\beta^{\mu_\beta}\rangle\right]_2\in R_T$*, whereby $t_1,\ldots,t_\alpha$ distinct and $t_{\alpha+1},\ldots,t_\beta$ distinct,*

$$\xi(t)=(\rho(t)-2)!\prod_{i=1}^{\alpha}\frac{1}{\mu_i!}\left(\frac{\xi(t_i)}{\rho(t_i)!}\right)^{\mu_i}\prod_{i=\alpha+1}^{\beta}\frac{1}{\mu_i!}\left(\frac{\xi(t_i)}{(\rho(t_i)-1)!}\right)^{\mu_i}, \tag{3.12}$$

where μ_i is the multiplicity of $t_i, i=1,\ldots,\beta$.

For tree $\bar{t}=\left[t\setminus\tau_{4,1}\right]$ or $\left[t\setminus\tau_{4,2}\right]$ with $t\in R_T$ and $\rho(t)\geq 5$,

$$\kappa^{(3)}=\sum_{\bar{t}}\left(\prod_{i=1}^{\alpha}\bar{\kappa}(\bar{t}_i)\prod_{i=\alpha+1}^{\beta}\rho(\bar{t}_i)\kappa(\bar{t}_i)\right).$$

B-series of TDRKN method can be defined as

$$B(\kappa,y,y')=\kappa(\emptyset)y+\sum_{t\in R_T}\frac{h^{\rho(t)}}{\rho(t)!}\kappa(t)\gamma(t)\xi(t)G(t)(y,y'), \tag{3.13}$$

and $g\left(B(\bar{\kappa},y,y'),B\left(\frac{\rho}{h}\kappa,y,y'\right)\right)$ can be denoted as

$$g\left(B(\bar{\kappa},y,y'),B\left(\frac{\rho}{h}\kappa,y,y'\right)\right)=\sum_{t\in R_T\setminus\{\tau_1,\tau_2\}}\frac{h^{\rho(t)-3}}{\rho(t)!}\kappa^{(3)}(t)\gamma(t)\xi(t)G(t)(y,y'), \tag{3.14}$$

where $\bar{\kappa}:R_T\cup\{\emptyset\}\to\mathbb{R}$ and $\kappa:R_T\to\mathbb{R}$ be two mappings satisfying $\bar{\kappa}(\emptyset)=1$ and $\kappa(\tau_1)=1$.

3.2. Analytical solution and exact derivative on B-series

Theorem 3.2. *Exact solution $y(x_0+h)$ and the derivative $y'(x_0+h)$ of the problem (1.1) have the forms as follows*

$$\begin{aligned}
y(x_0+h)&=y_0+hy_0'+\frac{1}{2}h^2f_0+\sum_{t\in R_T}\frac{h^{\rho(t)}}{\rho(t)!}\xi(t)G(t)(y_0,y_0')=B\left(\frac{\xi\sigma}{\rho!},y_0,y_0'\right)=B\left(\frac{1}{\gamma},y_0,y_0'\right),\\
y'(x_0+h)&=y_0'+hf_0+\sum_{t\in R_T}\frac{h^{\rho(t)-1}}{(\rho(t)-1)!}\xi(t)G(t)(y_0,y_0')=B\left(\frac{\xi\sigma}{h(\rho-1)!},y_0,y_0'\right)=B\left(\frac{\rho}{h\gamma},y_0,y_0'\right).
\end{aligned} \tag{3.15}$$

Proof. The conclusion is based on Theorem 3.1 and Eqs. (3.13).

3.3. Numerical solution and numerical derivative on B-series

B-series for a TDRKN method is established regarding the numerical solution of y_1 and its numerical derivatives, y_1' of the problem (1.1). Y_i and Y_i' are expanded as B-series as $Y_i = B(\bar{\Psi}_i, y_0, y_0')$ and $Y_i' = B(\frac{\rho}{h}\hat{\Psi}_i, y_0, y_0')$ respectively. Hence, we can transform equations in (2.3) into

$$\begin{aligned} B(\bar{\Psi}_i, y_0, y_0') &= y_0 + c_i h y_0' + \frac{(c_i h)^2}{2} f_0 + h^3 \sum_{j=1}^{s} A_{i,j} g\left(B(\bar{\Psi}_j, y_0, y_0'), B\left(\frac{\rho}{h}\hat{\Psi}_j, y_0, y_0'\right)\right), \\ B\left(\frac{\rho}{h}\hat{\Psi}_i, y_0, y_0'\right) &= y_0' + c_i h f_0 + h^2 \sum_{j=1}^{s} \hat{A}_{i,j} g\left(B(\bar{\Psi}_j, y_0, y_0'), B\left(\frac{\rho}{h}\hat{\Psi}_j, y_0, y_0'\right)\right). \end{aligned} \tag{3.16}$$

Referring to (3.13) and (3.14), the previous two equations can be expressed as

$$\begin{aligned} &\bar{\Psi}_i(\emptyset) y + \sum_{t\in R_T} \frac{h^{\rho(t)}}{\rho(t)!} \bar{\Psi}_i(t)\gamma(t)\xi(t)G(t)(y_0, y_0') \\ &= y_0 + c_i h y_0' + \frac{(c_i h)^2}{2} f_0 + \sum_{j=1}^{s} \sum_{R_T\setminus\{\tau_1,\tau_2,\tau_3\}} \frac{h^{\rho(t)}}{\rho(t)!} A_{i,j}\Psi_j^{(3)}(t)\gamma(t)\xi(t)G(t)(y_0, y_0'), \\ &\sum_{t\in R_T} \frac{h^{\rho(t)-1}}{(\rho(t)-1)!} \hat{\Psi}_i(t)\gamma(t)\xi(t)G(t)(y_0, y_0') \\ &= y_0' + c_i h f_0 + \sum_{j=1}^{s} \sum_{R_T\setminus\{\tau_1,\tau_2,\tau_3\}} \frac{h^{\rho(t)-1}}{\rho(t)!} \hat{A}_{i,j}\Psi_j^{(2)}(t)\gamma(t)\xi(t)G(t)(y_0, y_0'). \end{aligned} \tag{3.17}$$

It follows

$$\begin{aligned} &\bar{\Psi}_i(\emptyset) = 1,\ \bar{\Psi}_i(\tau_1) = c_i,\ \bar{\Psi}_i(\tau_2) = \frac{(c_i)^2}{2}, \\ &\hat{\Psi}_i(\tau_1) = 1,\ \hat{\Psi}_i(\tau_2) = c_i, \end{aligned} \tag{3.18}$$

and

$$\begin{aligned} \bar{\Psi}_i(t) &= \sum_{j=1}^{s} A_{i,j}\Psi_j^{(3)}(t), \quad t \in R_T \setminus \{\tau_1, \tau_2, \tau_3\}, \\ \hat{\Psi}_i(t) &= \frac{1}{\rho(t)} \sum_{j=1}^{s} \hat{A}_{i,j}\Psi_j^{(3)}(t), \quad t \in R_T \setminus \{\tau_1, \tau_2, \tau_3\}. \end{aligned} \tag{3.19}$$

where $\Psi_j^{(3)}(t) = \sum_{\bar{t}} \left(\prod_{i=1}^{\alpha} \bar{\Psi}_j(\bar{t}) \prod_{i=\alpha+1}^{\beta} \rho(\bar{t}_i)\hat{\Psi}_j(\bar{t})\right)$, $\quad \bar{t} = [t \setminus \tau_{4,1}]_2$ or $[t \setminus \tau_{4,2}]_2$.

Hereby, we denote the vectors $\Phi(t) = (\Phi_1(t), \dots, \Phi_s(t))^T$. The rooted trees with order up to six and the values of related functions are listed in Table 2. Elementary weight for y_1, $\phi(t)$ can be listed as follows:

$$\phi(t) = \sum_{i=1}^{s} b_i \Phi_i(t) = b^T \Phi(t), \tag{3.20}$$

and elementary weight for y_1' is expressed as $\phi'(t)$

$$\phi'(t) = \sum_{i=1}^{s} b_i' \Phi_i(t) = b'^T \Phi(t). \tag{3.21}$$

Hence, TDRKN method (2.3) yielded the following B-series for the numerical solution, y_1 and numerical derivative, y_1' for problem (1.1),

$$\begin{aligned} y_1(x_0 + h) &= y_0 + h y_0' + \frac{h^2}{2} f_0 + \sum_{t\in R_T\setminus\{\tau_1,\tau_2\}} \frac{h^{\rho(t)}}{\rho(t)!} \phi(t)\gamma(t)\xi(t)G(t)(y_0, y_0'), \\ y_1'(x_0 + h) &= y_0' + h f_0 + \sum_{t\in R_T\setminus\{\tau_1,\tau_2\}} \frac{h^{\rho(t)-1}}{\rho(t)!} \phi'(t)\gamma(t)\xi(t)G(t)(y_0, y_0'). \end{aligned} \tag{3.22}$$

Rooted trees of TDRKN method up to order six for y and y' are developed based on the B-series in Eqs. (3.22) and shown in Table 2. The elementary differentials are represented geometrically by a set of rooted trees, the version presented by Hairer et al. [14] and Butcher [6].

Following are the order conditions for explicit TDRKN methods:

The order conditions for y:

Third order:

$$b^T e = \frac{1}{6}. \tag{3.23}$$

Table 2
Root trees for TDRKN method up to order six.

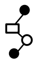
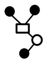
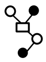
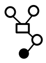
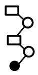
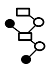
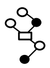
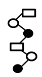
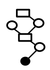
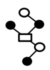
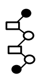
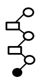
Order $\rho(t)$	Tree t	Graph	$\xi(t)$	Density $\gamma(t)$	$\phi(t)$	Elementary differential
1		τ_1	1	1		y'
2		τ_2	1	2		f
3		τ_3	1	6	e	g
4		$\tau_{4,1}$	1	24	c	$g_y y'$
		$\tau_{4,2}$	1	24	c	$g_{y'} f$
5		$\tau_{5,1}$	2	120	c^2	$g_{yy} y'^2$
		$\tau_{5,2}$	1	120	$\hat{A}e$	$g_{yy'} y' f$
		$\tau_{5,3}$	1	120	c^2	$g_{y'y'} f^2$
		$\tau_{5,4}$	1	120	c^2	$g_y f$
		$\tau_{5,5}$	1	120	$\hat{A}e$	$g_{y'} g$
6		$\tau_{6,1}$	3	360	c^3	$g_{yyy} y'^3$
		$\tau_{6,2}$	3	360	c^3	$g_{yyy'} y'^2 f$
		$\tau_{6,3}$	3	360	c^3	$g_{yy'y'} y' f^2$
		$\tau_{6,4}$	2	360	c^3	$g_{y'y'y'} f^3$
		$\tau_{6,5}$	2	720	c^3	$g_{yy'} y' g$
		$\tau_{6,6}$	1	720	c^3	$g_{yy'} f^2$
		$\tau_{6,7}$	1	720	Ae	$g_y g$
		$\tau_{6,8}$	1	720	$c\hat{A}$	$g_{y'y'} f g$
		$\tau_{6,9}$	1	720	c^3	$g_{yy} y' f$
		$\tau_{6,10}$	1	720	$\hat{A}c$	$g_{y'} g_y y'$
		$\tau_{6,11}$	1	720	c^3	$g_{y'} g_{y'} f$

Table 3
The TDRKN5 method.

c_1	0	0	0			0		
c_2	δ_2	$\hat{\delta}_2$	$A_{2,1}$	0		$\hat{A}_{2,1}$	0	
c_3	δ_3	$\hat{\delta}_3$	$A_{3,1}$	$A_{3,2}$	0	$\hat{A}_{3,1}$	$\hat{A}_{3,2}$	0
			b_1	b_2	b_3	b'_1	b'_2	b'_3

Fourth order:

$$b^T c = \frac{1}{24}. \tag{3.24}$$

Fifth order:

$$b^T c^2 = \frac{1}{60}. \tag{3.25}$$

Sixth order:

$$b^T c^3 = \frac{1}{120}, \qquad b^T \hat{A} e = \frac{1}{720}. \tag{3.26}$$

The order conditions for y':

Second order:

$$b'^T e = \frac{1}{2}. \tag{3.27}$$

Third order:

$$b'^T c = \frac{1}{6}. \tag{3.28}$$

Fourth order:

$$b'^T c^2 = \frac{1}{12}. \tag{3.29}$$

Fifth order:

$$b'^T c^3 = \frac{1}{20}, \qquad b'^T \hat{A} e = \frac{1}{120}. \tag{3.30}$$

Sixth order:

$$b'^T c^4 = \frac{1}{30}, \qquad b_i'^T \hat{A} c = \frac{1}{720}, \qquad b'^T A e = \frac{1}{720}, \qquad b'^T c \hat{A} e = \frac{1}{180}. \tag{3.31}$$

The simplifying assumption is used to generate the coefficients of TDRKN method as follows:

$$\sum_{i}^{s} \hat{A}_{i,j} = \frac{c_i^2}{2}, \qquad \sum_{i}^{s} A_{i,j} = \frac{c_i^3}{6}. \tag{3.32}$$

3.4. Three-stage fifth-order TDRKN method

The fifth-order TDRKN method is derived using algebraic order conditions up to order five, which are composed of Eqs. (3.23)–(3.25), (3.27)–(3.30), (3.32) and $b'^T c^4 = \frac{1}{30}$. Altogether it involves 13 equations and 14 variables and contains 1 free parameter, $A_{3,1}$ after solving those equations. The coefficients of the proposed method are presented in Butcher tableau and denoted by TDRKN5 as seen in Table 3:

where

$$\begin{aligned} &c_1 = 0, \quad c_2 = \frac{1}{2} + \frac{\sqrt{5}}{10}, \quad c_3 = \frac{1}{2} - \frac{\sqrt{5}}{10}, \quad b_1 = \frac{1}{24}, \quad b_2 = \frac{1}{16} - \frac{\sqrt{5}}{48}, \quad b_3 = \frac{1}{16} + \frac{\sqrt{5}}{48}, \\ &b'_1 = \frac{1}{12}, \quad b'_2 = \frac{5}{24} - \frac{\sqrt{5}}{24}, \quad b'_3 = \frac{5}{24} + \frac{\sqrt{5}}{24}, \quad \hat{A}_{2,1} = \frac{3}{20} + \frac{\sqrt{5}}{20}, \quad \hat{A}_{3,1} = 0, \quad \hat{A}_{3,2} = \frac{3}{20} - \frac{\sqrt{5}}{20}, \\ &A_{2,1} = \frac{1}{30} + \frac{\sqrt{5}}{75}, \quad A_{3,1} = -A_{3,2} + \frac{1}{30} - \frac{\sqrt{5}}{75}, \quad \delta_2 = 1, \quad \delta_3 = 1, \quad \hat{\delta}_2 = 1, \quad \hat{\delta}_3 = 1. \end{aligned} \tag{3.33}$$

For simplification purpose, we set $A_{3,1} = 0$, hence $A_{3,2} = \frac{1}{30} - \frac{\sqrt{5}}{75}$. Error norms for y and y' with order six of the proposed method are defined as follows:

$$\|\tau^{(6)}\|_2 = \sqrt{\sum_{i=1}^{N}\left(\tau_i^{(6)}\right)^2},$$

$$\|\tau'^{(6)}\|_2 = \sqrt{\sum_{i=1}^{M}\left(\tau_i'^{(6)}\right)^2},$$

$$\|\tau^{(6)}\|_g = \sqrt{\sum_{i=1}^{N+M}\left(\tau_i^{(6)}\right)^2 + \left(\tau_i'^{(6)}\right)^2},$$

where N and M are the total number of local truncation errors for y and y', $\tau^{(6)}$ and $\tau'^{(6)}$ are the local truncation error norms with order six for y and y' of proposed method respectively and $\|\tau^{(6)}\|_g$ is the global error norm with order six. For our method, $\|\tau^{(6)}\|_2 \approx 1.626 \times 10^{-3}$, $\|\tau'^{(6)}\|_2 \approx 9.932 \times 10^{-3}$ and $\|\tau^{(6)}\|_g \approx 1.006 \times 10^{-2}$.

Next, we analyze the zero stability and stability region of the proposed method. Generalizing the theorem proposed by [16] applied to Runge–Kutta type method for solving special second-order ODEs and the idea proposed by [17], we introduce the zero-stability for high-order method in the definition below.

Definition 3.5. The numerical method that is used for solving high-order ODEs with order p is zero stable with the numerical solutions remaining bounded as step size, $h \to 0$ if and only if the modulus of roots for the first characteristic polynomial, $\rho(\xi)$ satisfy the following conditions:

- $|\xi_j| \leq 1$ for $j = 1, 2, \ldots, r$,
- If ξ_j is a repeated root, then the multiplicity of the root of modulus 1 must be at most p.

Matrix finite difference equation of TDRKN5 methods can be written as

$$IY_{n+1} = AY_n + h^2(B \cdot F_n) + h^3(C \cdot G_n), \tag{3.34}$$

in which I is the 2×2 identity matrix, $Y_{n+1} = [y_{n+1}, hy'_{n+1}]^T$, $Y_n = [y_n, hy'_n]^T$, $F_n = [f_n, f_n]^T$, $G_n = [g_n, g_n]^T$, A, B and C are matrices 2×2. In Eq. (3.34), matrix A can be defined as

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Then,

$$I\xi - A = \begin{pmatrix} \xi - 1 & 1 \\ 0 & \xi - 1 \end{pmatrix}.$$

Hence, the first characteristic polynomial can be defined as

$$p(\xi) = \det[I\xi - A] = (\xi - 1)^2. \tag{3.35}$$

Thus, according to Definition 3.5, TDRKN5 method is zero stable with the roots of polynomial, $\xi_i = 1, i = 1, 2$ since the root of the first characteristic polynomial is 1 (repeated root) and the multiplicity of the root of the modulus 1 is two.

Then, the linear stability of TDRKN5 method is investigated. We use the test problem as follows:

$$y'' = \lambda^2 y. \tag{3.36}$$

Hence, we can obtain

$$g(Y, Y') = \lambda^2 Y'. \tag{3.37}$$

By applying TDRKN5 method to the test problem, we get

$$Y' = y'_n + ch(\lambda^2 y_n) + h^2 \hat{A}\left(\lambda^2 Y'\right). \tag{3.38}$$

Let us denote $v = \lambda h$ and multiply both sides with h, we yield

$$hY' = hy'_n + cv^2 y_n + v^2 A\left(hY'\right). \tag{3.39}$$

After rearrangement,

$$hY' = N.cv^2 y_n + N.e(hy'_n), \tag{3.40}$$

where $N = \left(I - v^2 \hat{A}\right)^{-1}$.

Similarly, we substitute (3.36) and (3.37) into the first two equations in (2.3),

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2}(\lambda^2 y_n) + h^3 b^T(\lambda^2 Y'),$$

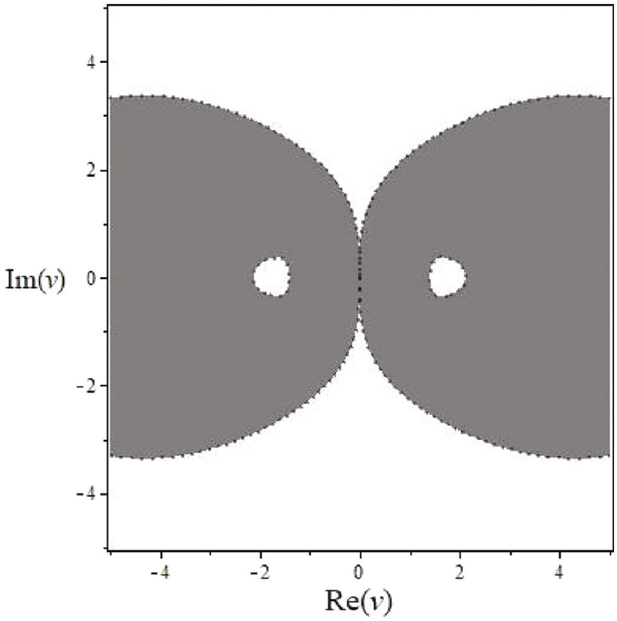


Fig. 1. The stability region of TDRKN5 method.

$$y'_{n+1} = y'_n + h(\lambda^2 y_n) + h^2 b'^T(\lambda^2 Y'). \tag{3.41}$$

Then, we substitute $v = \lambda h$ and (3.40) into (3.41) and carry out some simplification, we get the following matrix,

$$\begin{pmatrix} y_{n+1} \\ hy'_{n+1} \end{pmatrix} = \begin{pmatrix} 1 + \frac{v^2}{2} + v^4 b^T N.c & 1 + v^2 b^T N.e \\ v^2 + v^4 b'^T N.c & 1 + v^2 b'^T N.e \end{pmatrix} \begin{pmatrix} y_n \\ hy'_n \end{pmatrix}, \tag{3.42}$$

where $e = [1, 1, 1]^T$. After simplification,

$$\begin{pmatrix} y_{n+1} \\ hy'_{n+1} \end{pmatrix} = P(v) \begin{pmatrix} y_n \\ hy'_n \end{pmatrix}, \tag{3.43}$$

where $P(v) = \begin{pmatrix} 1 + \frac{v^2}{2} + v^4 b^T N.c & 1 + v^2 b^T N.e \\ v^2 + v^4 b'^T N.c & 1 + v^2 b'^T N.e \end{pmatrix}$.

The stability region in the complex plane of TDRKN5 method can be defined as

$$S_R = \{v : |\lambda_i(P(v))| < 1, i = 1, 2\}, \tag{3.44}$$

where λ_i are eigenvalues of $P(v)$. The stability region of TDRKN5 method is shown in Fig. 1.

3.5. Exponentially-fitted two-derivative Runge–Kutta-Nyström method

The motivation of the exponentially-fitting technique is to modify the existing method into a method with frequency-dependent parameters that works perfectly on solving differential equations with exponential solutions. We focus on the class of equations in the following form:

$$y'' = f(x, y), \quad y(0) = \alpha, \quad y'(0) = \beta, \tag{3.45}$$

where $\alpha, \beta \in \mathbb{R}$ and the exact solution is exponential form.

To develop TDRKN5 method with exponentially-fitting technique, we integrate the exponential function, $e^{\lambda x}$ and $e^{-\lambda x}$ at every stage and $v = \lambda h$, $\lambda \in \mathbb{R}$, we obtain

$$e^{\pm c_i v} = 1 \pm c_i v + \frac{c_i^2}{2} v^2 \pm v^3 \sum_{j=1}^{s} A_{i,j} e^{\pm c_j v}, \tag{3.46}$$

$$e^{\pm c_i v} = 1 \pm c_i v + v^2 \sum_{j=1}^{s} \hat{A}_{i,j} e^{\pm c_j v}. \tag{3.47}$$

We insert the fitting terms, δ_i and $\hat{\delta}_i$ into the equations above as follows:

$$e^{\pm c_i \upsilon} = \delta_i \pm c_i \upsilon + \frac{c_i^2}{2}\upsilon^2 \pm \upsilon^3 \sum_{j=1}^{s} A_{i,j} e^{\pm i c_j \upsilon}, \tag{3.48}$$

$$e^{\pm c_i \upsilon} = 1 \pm \hat{\delta}_i c_i \upsilon + \upsilon^2 \sum_{j=1}^{s} \hat{A}_{i,j} e^{\pm c_j \upsilon}. \tag{3.49}$$

The equations corresponding to y and y' are:

$$e^{\pm \upsilon} = 1 \pm \upsilon + \frac{1}{2}\upsilon^2 \pm \upsilon^3 \sum_{i=1}^{s} b_i e^{\pm c_i \upsilon}, \tag{3.50}$$

$$e^{\pm \upsilon} = 1 \pm \upsilon + \upsilon^2 \sum_{i=1}^{s} b_i' e^{\pm c_i \upsilon}. \tag{3.51}$$

The relation $\cosh(\upsilon) = \frac{e^{\upsilon}+e^{-\upsilon}}{2}$ and $\sinh(\upsilon) = \frac{e^{\upsilon}-e^{-\upsilon}}{2}$ are substituted in the preceeding equations. In this case, we get hyperbolic functions of υ as below:

$$\begin{aligned}
\cosh(\upsilon) &= 1 + \frac{1}{2}\upsilon^2 + \upsilon^3 \sum_{i=1}^{s} b_i \left[\sinh(\upsilon c_i)\right], \\
\sinh(\upsilon) &= \upsilon + \upsilon^3 \sum_{i=1}^{s} b_i \left[\cosh(\upsilon c_i)\right], \\
\cosh(\upsilon) &= 1 + \upsilon^2 \sum_{i=1}^{s} b_i' \left[\cosh(\upsilon c_i)\right], \\
\sinh(\upsilon) &= \upsilon + \upsilon^2 \sum_{i=1}^{s} b_i' \left[\sinh(\upsilon c_i)\right].
\end{aligned} \tag{3.52}$$

The coefficients $A_{i,i-1}$ and $\hat{A}_{i,i-1}$ with fitting property can be determined through the formula below

$$\hat{A}_{i,i-1} = \frac{\cosh(c_i \upsilon) - 1 - \upsilon^2 \sum_{j=1}^{i-2} \hat{A}_{ij} \cosh(c_j \upsilon)}{\upsilon^2 \cosh(c_{i-1}\upsilon)}, \tag{3.53}$$

$$A_{i,i-1} = \frac{\sinh(c_i \upsilon) - c_i \upsilon - \upsilon^3 \sum_{j=1}^{i-2} A_{ij} \cosh(c_j \upsilon)}{\upsilon^3 \cosh(c_{i-1}\upsilon)}. \tag{3.54}$$

Then, we determine δ_i and $\hat{\delta}_i$ based on the coefficients $A_{i,i-1}$ and $\hat{A}_{i,i-1}$ through

$$\hat{\delta}_i = \frac{\sinh(c_i \upsilon) - \upsilon^2 \sum_{j=1}^{i-1} \hat{A}_{i,j} \sinh(c_j \upsilon)}{c_i \upsilon}, \tag{3.55}$$

$$\delta_i = \cosh(c_i \upsilon) - \frac{1}{2}(c_i \upsilon)^2 - \upsilon^3 \left(\sum_{j=1}^{i-1} A_{i,j} \sinh(c_j \upsilon) \right). \tag{3.56}$$

Modify Eqs. (3.27)–(3.30), coefficients for $a_{i,j}, \hat{a}_{i,j}, \gamma_i$ and $\hat{\gamma}_i$ can be determined as follows:

$$\begin{aligned}
\hat{A}_{2,1} &= \frac{\cosh(c_2 \upsilon) - 1}{\upsilon^2}, \\
\hat{A}_{3,2} &= \frac{\cosh(c_2 \upsilon) - 1 - \upsilon^2 \hat{A}_{3,1}}{\upsilon^2 \cosh(c_2 \upsilon)}, \\
A_{2,1} &= \frac{\sinh(c_2 \upsilon) - c_2 \upsilon}{\upsilon^3}, \\
A_{3,2} &= \frac{\sinh(c_3 \upsilon) - c_3 \upsilon - \upsilon^3 A_{3,1}}{\upsilon^3 \cosh(c_2 \upsilon)}, \\
\hat{\gamma}_2 &= \frac{\sinh(c_2 \upsilon)}{c_2 \upsilon}, \\
\hat{\gamma}_3 &= \frac{\sinh(c_3 \upsilon) + \upsilon^2 \left(\hat{A}_{3,2} \sinh(c_2 \upsilon)\right)}{c_3 \upsilon}, \\
\gamma_2 &= \cosh(c_2 \upsilon) - \frac{1}{2}(c_2 \upsilon)^2, \\
\gamma_3 &= \cosh(c_3 \upsilon) - \frac{1}{2}(c_3 \upsilon)^2 - \upsilon^3 \left(A_{3,2} \sinh(c_2 \upsilon)\right).
\end{aligned} \tag{3.57}$$

By substituting parameters in (3.33) into Eqs. (3.57) and applying Taylor series expansion, we yield all frequency-dependent coefficients for $A_{i,j}, \hat{A}_{i,j}, \delta_i$ and $\hat{\delta}_i$ of proposed methods,

$$\begin{aligned}
\hat{A}_{2,1} =& \frac{3}{20}+\frac{\sqrt{5}}{20}+\left(\frac{7}{1200}+\frac{\sqrt{5}}{400}\right)v^2+\left(\frac{1}{10000}+\frac{\sqrt{5}}{22500}\right)v^4+\left(\frac{47}{50400000}+\frac{\sqrt{5}}{2400000}\right)v^6 \\
&+\left(\frac{41}{7560000000}+\frac{11\sqrt{5}}{4536000000}\right)v^8+\mathcal{O}(v^{10}), \\
\hat{A}_{3,2} =& \frac{3}{20}-\frac{\sqrt{5}}{20}+\left(-\frac{1}{240}-\frac{\sqrt{5}}{400}\right)v^2+\left(\frac{11}{10000}+\frac{41\sqrt{5}}{90000}\right)v^4+\left(-\frac{2281}{10080000}-\frac{241\sqrt{5}}{2400000}\right)v^6 \\
&+\left(\frac{361961}{7560000000}+\frac{97009\sqrt{5}}{4536000000}\right)v^8+\mathcal{O}(v^{10}), \\
A_{2,1} =& \frac{1}{30}+\frac{\sqrt{5}}{75}+\left(\frac{1}{1200}+\frac{11\sqrt{5}}{30000}\right)v^2+\left(\frac{13}{1260000}+\frac{29\sqrt{5}}{6300000}\right)v^4+\left(\frac{17}{226800000}+\frac{19\sqrt{5}}{567000000}\right)v^6 \\
&+\left(\frac{89}{249480000000}+\frac{199\sqrt{5}}{1247400000000}\right)v^8+\mathcal{O}(v^{10}), \\
A_{3,2} =& \left(\frac{1}{1200}-\frac{11\sqrt{5}}{30000}\right)v^2+\left(-\frac{29}{1260000}+\frac{11\sqrt{5}}{1260000}\right)v^4+\left(\frac{121}{113400000}-\frac{31\sqrt{5}}{226800000}\right)v^6 \\
&+\left(\frac{911}{8910000000}-\frac{16693\sqrt{5}}{623700000000}\right)v^8+\mathcal{O}(v^{10}), \\
\hat{\delta}_2 =& 1+\left(\frac{5+2\sqrt{5}}{15\left(5+\sqrt{5}\right)}\right)v^2+\left(\frac{25+11\sqrt{5}}{3000\left(5+\sqrt{5}\right)}\right)v^4+\left(\frac{65+29\sqrt{5}}{630000\left(5+\sqrt{5}\right)}\right)v^6+\left(\frac{85+38\sqrt{5}}{113400000\left(5+\sqrt{5}\right)}\right)v^8+\mathcal{O}(v^{10}), \\
\hat{\delta}_3 =& 1+\left(\frac{5+\sqrt{5}}{30\left(-5+\sqrt{5}\right)}\right)v^2-\left(\frac{75+29\sqrt{5}}{3000\left(-5+\sqrt{5}\right)}\right)v^4+\left(\frac{1560+683\sqrt{5}}{315000\left(-5+\sqrt{5}\right)}\right)v^6-\left(\frac{235985+105209\sqrt{5}}{226800000\left(-5+\sqrt{5}\right)}\right)v^8+\mathcal{O}(v^{10}) \\
\delta_2 =& 1+\left(\frac{7}{1200}+\frac{\sqrt{5}}{400}\right)v^4+\left(\frac{1}{10000}+\frac{\sqrt{5}}{22500}\right)v^6+\left(\frac{47}{50400000}+\frac{\sqrt{5}}{2400000}\right)v^8+\mathcal{O}(v^{10}), \\
\delta_3 =& 1+\left(\frac{7}{1200}-\frac{\sqrt{5}}{400}\right)v^4+\left(-\frac{1}{7500}+\frac{\sqrt{5}}{18000}\right)v^6+\left(\frac{239}{50400000}-\frac{23\sqrt{5}}{16800000}\right)v^8+\mathcal{O}(v^{10}).
\end{aligned} \tag{3.58}$$

Subsequently, the coefficients above are used to generate parameters b_1, b_2, b_1' and b_2' through Taylor series expansion.

$$\begin{aligned}
b_2 =& \frac{1}{16}-\frac{\sqrt{5}}{48}+\left(-\frac{1}{201600}+\frac{\sqrt{5}}{504000}\right)v^4+\left(\frac{1}{181440000}+\frac{\sqrt{5}}{181440000}\right)v^6+\left(-\frac{17}{39916800000}+\frac{\sqrt{5}}{66528000000}\right)v^8+\mathcal{O}(v^{10}), \\
b_3 =& \frac{1}{16}+\frac{\sqrt{5}}{48}+\left(-\frac{1}{201600}+\frac{\sqrt{5}}{504000}\right)v^4+\left(\frac{1}{181440000}+\frac{\sqrt{5}}{181440000}\right)v^6+\left(-\frac{17}{39916800000}+\frac{\sqrt{5}}{66528000000}\right)v^8+\mathcal{O}(v^{10}), \\
b_2' =& \frac{5}{24}-\frac{\sqrt{5}}{24}+\left(\frac{\sqrt{5}}{252000}\right)v^4+\left(-\frac{1}{6048000}+\frac{\sqrt{5}}{90720000}\right)v^6+\left(\frac{1}{362880000}+\frac{\sqrt{5}}{33264000000}\right)v^8+\mathcal{O}(v^{10}), \\
b_3' =& \frac{5}{24}+\frac{\sqrt{5}}{24}-\left(\frac{\sqrt{5}}{252000}\right)v^4+\left(-\frac{1}{6048000}-\frac{\sqrt{5}}{90720000}\right)v^6+\left(\frac{1}{362880000}-\frac{\sqrt{5}}{33264000000}\right)v^8+\mathcal{O}(v^{10}).
\end{aligned} \tag{3.59}$$

4. Problem testing and numerical result

Fifth-order exponentially-fitted two-derivative Runge–Kutta–Nyström (EFTDRKN5) method is applied to solve second-order ODEs in the form $y''(x) = f(x, y(x))$ and application problem with exponential solution in this section. The numerical efficiency of the proposed method in the literature is shown by comparing multiple existing Runge–Kutta–Nyström methods with classical type and fitted techniques. The following methods have been chosen to be numerically compared.

- **EFTDRKN5** - Three-stage fifth-order explicit two-derivative Runge–Kutta–Nyström method with exponentially-fitting technique, proposed in this paper.
- **EFRKN5D** - Four-stage fifth-order explicit Runge–Kutta–Nyström method with exponentially-fitting technique, proposed by Demba et al. [9]

- **EFRKN5T** - Three-stage fifth-order modified explicit two-derivative Runge–Kutta–Nyström method with exponentially-fitting technique, proposed by Mohamed et al. [22]
- **PFRKN5** - Four-stage fifth-order explicit Runge–Kutta–Nyström method with phase fitting technique, proposed by Salih et al. [25]
- **TDRKN5C** - Three stage fifth-order explicit two-derivative Runge–Kutta–Nyström method, proposed by Chen et al. [8]
- **RKM5** - Explicit two-derivative Runge–Kutta–Nyström method with three stage fifth-order, proposed by Mohamed et al. [26]

Problem 1 (*Homogeneous Problem*).

$$y'' = 4y,$$

$$y(0) = 0, y'(0) = 1, \quad x \in [0, b],$$

with analytical solution, $y(x) = \frac{1}{4}e^{2x} - \frac{1}{4}e^{-2x}$.

Problem 2 (*Inhomogeneous Problem*).

$$y'' = 5y + \cosh(x),$$

$$y(0) = -\frac{1}{4}, y'(0) = 0, \quad x \in [0, b],$$

with analytical solution $y(x) = -\frac{1}{8}e^{x} - \frac{1}{8}e^{-x}$.

Problem 3 (*Inhomogeneous Problem*).

$$y'' = y + x - 1,$$

$$y(0) = 2, y'(0) = -2, \quad x \in [0, b],$$

with analytical solution $y(x) = 1 - x + e^{-x}$.

Problem 4 (*Homogeneous System*).

$$y_1'' = 8y_3, \quad y_2'' = 8y_1, \quad y_3'' = y_2,$$

$$y_1(0) = 2, y_1'(0) = 4, y_2(0) = 4, y_2'(0) = 8, y_3(0) = 1, y_3'(0) = 2, \quad x \in [0, b],$$

with analytical solution, $y_1(x) = 2e^{2x}, y_2(x) = 4e^{2x}$ and $y_3(x) = e^{2x}$.

Problem 5 (*Inhomogeneous System*).

$$y_1'' = -y_2 + e^{x}, \quad y_2'' = -y_1 + e^{x},$$

$$y_1(0) = 0, y_1'(0) = 2, y_2(0) = 1, y_2'(0) = -1, \quad x \in [0, b],$$

with analytical solution, $y_1(x) = e^{x} - e^{-x}, y_2(x) = e^{-x}$.

Problem 6 (*Prothero–Robinson Problem*).

$$y'' = \lambda^2 y - \left(y - e^{-\lambda x}\right)^3,$$

$$y(0) = 1, y'(0) = -\lambda, \quad x \in [0, b],$$

with analytical solution $y(x) = e^{-\lambda x}$. We take $\lambda = 2$ in this test problem.

Problem 7 (*Application Problem - Verhulst Logistic Growth Model*). Verhulst logistic growth model is a mathematical model used to describe population growth that starts with exponential growth and then slows down as it approaches a maximum value or carrying capacity. This model incorporates the concept of limiting factors, which are factors that restrict population growth as it approaches the carrying capacity. These factors can include limited resources, competition, predation, disease, and other factors that influence population dynamics. It originated by the first-order ODEs as follows:

$$y'(x) = ry(x)\left(1 - \frac{y(x)}{K}\right), \tag{4.60}$$

where K represents carrying capacity and r represents growth rate [27].

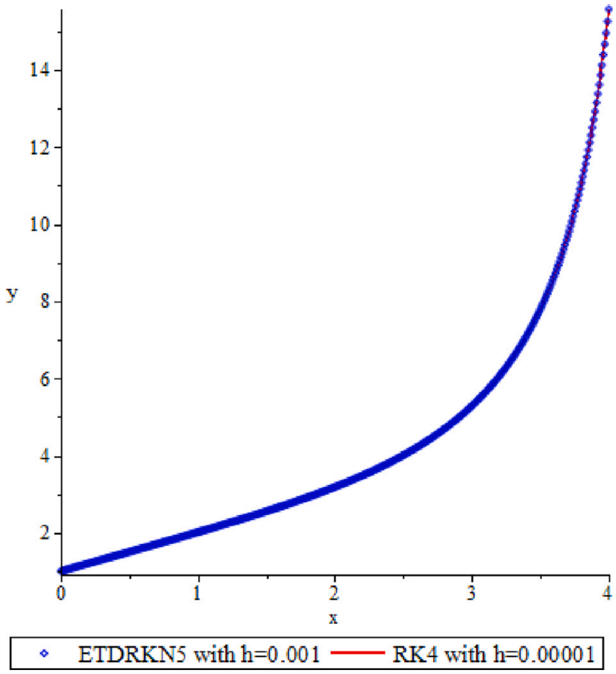


Fig. 2. Numerical simulation for logistic growth model with $y(0) = 1$ and $y'(0) = 1, r = \sqrt{0.02}$ and $K = 1$.

Eq. (4.60) can be modified to second-order ODEs as below:

$$y''(x) = r^2\left(y(x) - \frac{3}{K}y^2(x) + \frac{2}{K^2}y^3(x)\right),$$
$$y(0) = y_0, \quad y'(0) = y_0', \quad x \in [0, b], \tag{4.61}$$

where y_0, y_0' and b are real numbers, K represents carrying capacity and r represents growth rate.

In (4.61), not all equations consist of analytical solutions as they are highly dependent on the initial conditions. In this study, we take $y(0) = 1$ and $y'(0) = 1, r = \sqrt{0.02}$ and $K = 1$. Since there is no analytical solution for these initial conditions, we use classical Runge–Kutta method with order 4 as measurement tool to obtain numerical approximation with step size $h = 10^{-5}$ and compare it with all comparative methods.

The figure below shows the numerical simulation logistic growth model of the classical Runge–Kutta method with order 4 with step-size, $h = 10^{-5}$ and ETDRKN5 method with $h = 10^{-3}$ by taking the initial conditions we use in this study.

The numerical data are presented in Tables 4–9 with different step-size, h in particular endpoints, b. The tables contain the maximum global error, MAXERR; the number of function evaluations involved, FE and the time of computation in seconds. The maximum global error is written as $a(-b)$, instead of $a \cdot 10^{-b}$, where $a, b \in \mathbb{R}$ (see Fig. 2).

Figs. 3–8 demonstrate the numerical performance of proposed method and other selected methods in terms of maximum global truncation error against computational time. Meanwhile, Fig. 9 shows the numerical comparison for all selected methods in terms of maximum global truncation error against number of function evaluations. The model of computer used in computing the numerical results is Lenovo ideapad 330 Intel Core i5-8050U (1.8 GHz).

5. Discussion

In a comparison of the number of function evaluations required by all selected methods, EFTDRKN5 method is one of the best among all methods by requiring the least amount of function evaluation for all endpoints due to the least number of stages per step based on the results obtained in Tables 4–10 and Figs. 3–9. In computational time, the time consumption for the proposed method is considered moderate. This is because of the complexity of single function evaluation, which consists of the second derivative of function, g-evaluation which is more complex compared to classical f-evaluation. This causes the EFTDRKN5 method requiring more time to finalize the computational steps compared to some RKN methods without second derivative such as EFRKN5D, PFRKN5 and RKM5 methods. In the comparison of accuracy, EFTDRKN5 method outperforms other methods by generating the least maximum global error for solving all kinds of second-order ODEs with exponential solutions, comprising linear homogeneous and nonhomogeneous equations and systems. Also, our proposed method is effective in solving the nonlinear Prothero–Robinson problem, which is considered a stiff equation. This is largely because of the exponential-fitting techniques implemented in a huge

Table 4
Comparison between EFTDRKN5 method with existing methods for Problem 1.

h	Methods	$b = 5$			$b = 10$		
		MAXERR	FE	Time (s)	MAXERR	FE	Time (s)
0.1	**EFTDRKN5**	4.613895(−14)	150	0.004	4.347613(−9)	300	0.011
	EFRKN5D	2.732603(−5)	200	0.003	6.018529(−1)	400	0.006
	EFRKN5T	7.137355(−5)	300	0.006	2.971495(+0)	600	0.012
	PFRKN5	1.193372(+0)	200	0.003	5.261285(+3)	400	0.005
	TDRKN5C	2.412671(−4)	300	0.006	1.145283(+1)	600	0.012
	RKM5	3.754358(−3)	150	0.004	1.660037(+2)	300	0.009
0.05	**EFTDRKN5**	1.453867(−18)	300	0.009	6.766962(−14)	600	0.020
	EFRKN5D	4.507752(−7)	400	0.006	9.928964(−3)	800	0.012
	EFRKN5T	1.229300(−6)	600	0.013	5.144427(−2)	1200	0.025
	PFRKN5	2.639118(−2)	400	0.006	1.162675(+3)	800	0.010
	TDRKN5C	4.097050(−6)	600	0.013	1.934314(−1)	1200	0.025
	RKM5	1.238840(−4)	300	0.008	5.467085(+0)	600	0.018
0.025	**EFTDRKN5**	2.240954(−23)	600	0.019	1.042569(−18)	1200	0.041
	EFRKN5D	7.223044(−9)	800	0.013	1.590981(−4)	1600	0.024
	EFRKN5T	2.016753(−8)	1200	0.027	8.460508(−4)	2400	0.051
	PFRKN5	4.033400(−3)	800	0.013	1.776841(+2)	1600	0.020
	TDRKN5C	6.672585(−8)	1200	0.027	3.141977(−3)	2400	0.050
	RKM5	3.977018(−6)	600	0.017	1.753500(−1)	1200	0.036
0.0125	**EFTDRKN5**	3.436549(−29)	1200	0.040	1.598412(−23)	2400	0.080
	EFRKN5D	1.142350(−10)	1600	0.026	2.516194(−6)	3200	0.049
	EFRKN5T	3.228988(−10)	2400	0.055	1.356215(−5)	4800	0.104
	PFRKN5	5.506476(−4)	1600	0.026	2.425766(+1)	3200	0.041
	TDRKN5C	1.064378(−9)	2400	0.054	5.005416(−5)	4800	0.102
	RKM5	1.259572(−7)	1200	0.034	5.551138(−3)	2400	0.070
0.00625	**EFTDRKN5**	5.256804(−33)	2400	0.080	2.444744(−28)	4800	0.160
	EFRKN5D	1.795543(−12)	3200	0.054	3.954946(−8)	6400	0.102
	EFRKN5T	5.107222(−12)	4800	0.110	2.146360(−7)	9600	0.210
	PFRKN5	7.175353(−5)	3200	0.055	3.160953(+0)	6400	0.081
	TDRKN5C	1.680359(−11)	4800	0.110	7.897070(−7)	9600	0.208
	RKM5	3.962532(−9)	2400	0.069	1.745980(−4)	4800	0.140

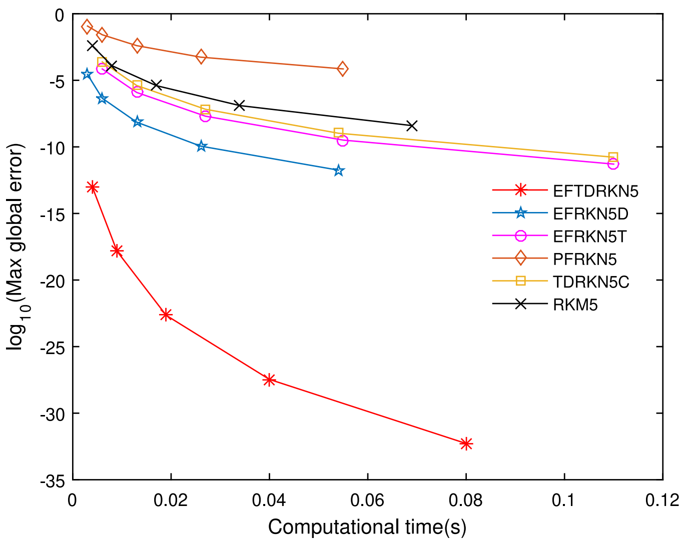


Fig. 3. Numerical curves of selected methods for Problem 1 with $b = 5$ and $h = \frac{0.1}{2^i}, i = 0, 1, \dots, 4$.

number of parameters in RKN method with two-derivative terms. In handling the selected second-order application problem, the logistic growth model, EFTDRKN5 method generates relatively low global truncation errors with different step-size, considered the

Table 5
Comparison between EFTDRKN5 method with existing methods for Problem 2.

h	Methods	$b = 5$			$b = 10$		
		MAXERR	FE	Time (s)	MAXERR	FE	Time (s)
0.1	**EFTDRKN5**	6.447814(−19)	150	0.017	4.633263(−14)	300	0.033
	EFRKN5D	1.119847(−4)	200	0.010	8.041715(+0)	400	0.021
	EFRKN5T	3.984426(−7)	300	0.020	2.862442(−2)	600	0.041
	PFRKN5	3.820059(−3)	200	0.010	2.748504(+2)	400	0.022
	TDRKN5C	7.499811(−6)	300	0.019	5.389274(−1)	600	0.041
	RKM5	6.602033(−5)	150	0.014	4.741567(+0)	300	0.028
0.05	**EFTDRKN5**	1.003519(−23)	300	0.034	7.210913(−19)	600	0.062
	EFRKN5D	3.637948(−6)	400	0.021	2.612511(−1)	800	0.044
	EFRKN5T	6.982695(−9)	600	0.038	5.016977(−4)	1200	0.081
	PFRKN5	6.684797(−4)	400	0.021	4.807806(+1)	800	0.045
	TDRKN5C	1.231044(−7)	600	0.040	8.845898(−3)	1200	0.082
	RKM5	2.112082(−6)	300	0.030	1.516907(−1)	600	0.054
0.025	**EFTDRKN5**	1.545171(−28)	600	0.066	1.110294(−23)	1200	0.115
	EFRKN5D	1.158397(−7)	800	0.043	8.318877(−3)	1600	0.088
	EFRKN5T	1.154695(−10)	1200	0.073	8.296729(−6)	2400	0.163
	PFRKN5	9.587669(−5)	800	0.044	6.894752(+0)	1600	0.089
	TDRKN5C	1.970884(−9)	1200	0.080	1.416198(−4)	2400	0.165
	RKM5	6.675812(−8)	600	0.060	4.794622(−3)	1200	0.106
0.0125	**EFTDRKN5**	2.367925(−33)	1200	0.104	1.701483(−28)	2400	0.226
	EFRKN5D	3.653543(−9)	1600	0.086	2.623760(−4)	3200	0.171
	EFRKN5T	1.855779(−12)	2400	0.123	1.333446(−7)	4800	0.318
	PFRKN5	1.276584(−5)	1600	0.088	9.179804(−1)	3200	0.172
	TDRKN5C	3.116965(−11)	2400	0.130	2.239713(−6)	4800	0.316
	RKM5	2.097924(−9)	1200	0.100	1.506750(−4)	2400	0.201
0.00625	**EFTDRKN5**	3.616680(−38)	2400	0.204	2.598781(−33)	4800	0.420
	EFRKN5D	1.146965(−10)	3200	0.166	8.236858(−6)	6400	0.312
	EFRKN5T	2.940679(−14)	4800	0.250	2.113008(−9)	9600	0.589
	PFRKN5	1.644909(−6)	3200	0.168	1.182813(−1)	6400	0.314
	TDRKN5C	4.899699(−13)	4800	0.250	3.520696(−8)	9600	0.586
	RKM5	6.574289(−11)	2400	0.197	4.721726(−6)	4800	0.389

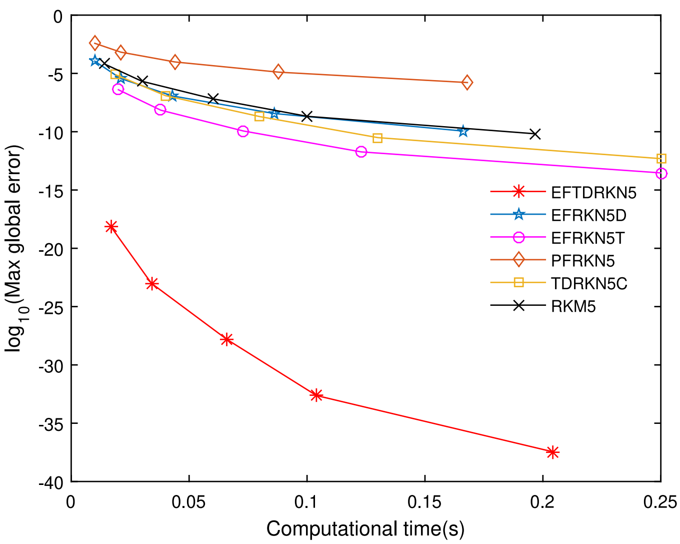


Fig. 4. Numerical curves of selected methods for Problem 2 with $b = 5$ and $h = 0.01 - 0.002i, i = 0, 1, \dots, 4$.

second-best method after RKM5 method. The number of function evaluations involved is the lowest as well in approximating the outcomes of this model. This is because the numerical simulation of this logistic growth model has the property of exponential growth based on the approximation obtained by RK4 method with extremely low step-size, thus the proposed method produces good numerical approximation.

Table 6
Comparison between EFTDRKN5 method with existing methods for Problem 3.

h	Methods	$b = 10$			$b = 20$		
		MAXERR	FE	Time (s)	MAXERR	FE	Time (s)
0.1	**EFTDRKN5**	3.256790(−19)	300	0.007	7.173558(−15)	600	0.018
	EFRKN5D	1.803104(−6)	400	0.005	3.971602(−2)	800	0.011
	EFRKN5T	2.442516(−7)	600	0.010	5.379999(−3)	1200	0.025
	PFRKN5	9.937543(−5)	400	0.005	2.188909(+0)	800	0.011
	TDRKN5C	1.175343(−6)	600	0.009	2.588865(−2)	1200	0.024
	RKM5	7.912984(−7)	300	0.006	1.742953(−2)	600	0.015
0.05	**EFTDRKN5**	5.002155(−24)	600	0.015	1.101798(−19)	1200	0.036
	EFRKN5D	2.889218(−8)	800	0.011	6.363926(−4)	1600	0.021
	EFRKN5T	3.834624(−9)	1200	0.020	8.446320(−5)	2400	0.049
	PFRKN5	3.195732(−6)	800	0.010	7.039175(−2)	1600	0.022
	TDRKN5C	1.838769(−8)	1200	0.020	4.050159(−4)	2400	0.048
	RKM5	3.844161(−8)	600	0.013	8.467336(−4)	1200	0.029
0.025	**EFTDRKN5**	7.654660(−29)	1200	0.033	1.686051(−24)	2400	0.070
	EFRKN5D	4.569401(−10)	1600	0.023	1.006478(−5)	3200	0.042
	EFRKN5T	6.004274(−11)	2400	0.043	1.322529(−6)	4800	0.096
	PFRKN5	1.012763(−7)	1600	0.022	2.230875(−3)	3200	0.043
	TDRKN5C	2.873975(−10)	2400	0.044	6.330352(−6)	4800	0.094
	RKM5	1.415534(−9)	1200	0.029	3.117923(−5)	2400	0.057
0.0125	**EFTDRKN5**	1.169568(−33)	2400	0.071	2.576144(−29)	4800	0.139
	EFRKN5D	7.182170(−12)	3200	0.049	1.581978(−7)	6400	0.083
	EFRKN5T	9.390973(−13)	4800	0.090	2.068499(−8)	9600	0.190
	PFRKN5	3.186511(−9)	3200	0.048	7.020168(−5)	6400	0.085
	TDRKN5C	4.490937(−12)	4800	0.092	9.891947(−8)	9600	0.186
	RKM5	4.758066(−11)	2400	0.062	1.048035(−6)	4800	0.112
0.00625	**EFTDRKN5**	1.669285(−38)	4800	0.145	3.676855(−34)	9600	0.275
	EFRKN5D	1.125513(−13)	6400	0.102	2.479108(−9)	12 800	0.165
	EFRKN5T	1.468042(−14)	9600	0.195	3.233577(−10)	19 200	0.378
	PFRKN5	9.986543(−11)	6400	0.101	2.201411(−6)	12 800	0.168
	TDRKN5C	7.017226(−14)	9600	0.198	1.545647(−9)	19 600	0.372
	RKM5	1.539141(−12)	4800	0.130	3.390187(−8)	9600	0.222

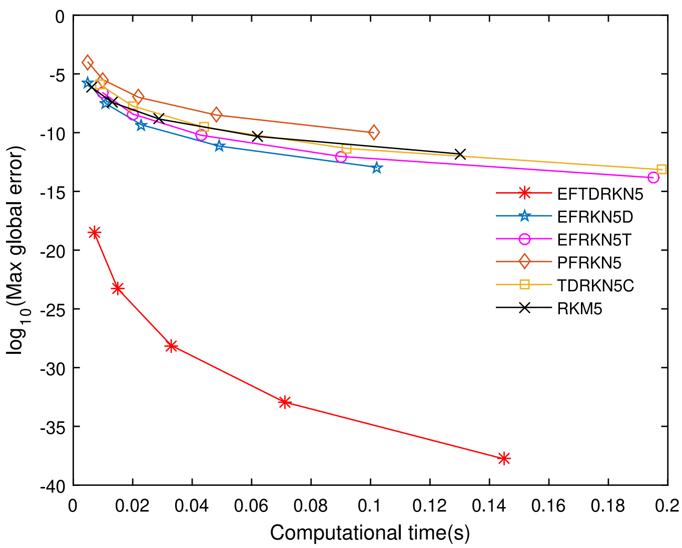


Fig. 5. Numerical curves of selected methods for Problem 3 with $b = 10$ and $h = 0.01 - 0.002i, i = 0, 1, \dots, 4$.

6. Conclusion

In this study, an explicit fifth-order two-derivative Runge–Kutta–Nyström method with minimal function evaluation and exponentially-fitting technique has been developed to solve $y''(x) = f(x, y(x))$ with exponential solutions. B-series and rooted tree theory specifically for TDRKN method are derived with the motivation to generate the order conditions for the proposed method. A three-stage fifth-order classical TDRKN method, denoted as TDRKN5 is constructed based on the algebraic order conditions up

Table 7
Comparison between EFTDRKN5 method with existing methods for Problem 4.

h	Methods	$b = 5$			$b = 10$		
		MAXERR	FE	Time (s)	MAXERR	FE	Time (s)
0.05	**EFTDRKN5**	6.141145(−18)	900	0.022	2.778520(−13)	1800	0.039
	EFRKN5D	1.438933(−11)	1200	0.017	6.523979(−7)	2400	0.031
	EFRKN5T	4.672950(−6)	1800	0.029	2.003971(−1)	3600	0.046
	PFRKN5	1.054653(−1)	1200	0.018	4.648513(+3)	2400	0.032
	TDRKN5C	7.025417(−5)	1800	0.029	3.198457(+0)	3600	0.046
	RKM5	1.978979(−3)	900	0.020	8.740364(+1)	1800	0.036
0.025	**EFTDRKN5**	9.464031(−23)	1800	0.045	4.280457(−18)	3600	0.076
	EFRKN5D	7.042297(−15)	2400	0.034	3.188359(−10)	4800	0.062
	EFRKN5T	7.683551(−8)	3600	0.059	3.299740(−3)	7200	0.092
	PFRKN5	1.613040(−2)	2400	0.036	7.106662(+2)	4800	0.065
	TDRKN5C	1.141164(−6)	3600	0.058	5.189170(−2)	7200	0.091
	RKM5	6.347853(−5)	1800	0.039	2.802213(+0)	3600	0.070
0.0125	**EFTDRKN5**	1.451166(−27)	3600	0.088	6.562253(−23)	7200	0.142
	EFRKN5D	3.442872(−18)	4800	0.069	1.557659(−13)	9600	0.124
	EFRKN5T	1.231552(−9)	7200	0.115	5.292606(−5)	14 400	0.183
	PFRKN5	2.202489(−3)	4800	0.072	9.702839(+1)	9600	0.129
	TDRKN5C	1.817964(−8)	7200	0.114	8.261880(−4)	14 400	0.181
	RKM5	2.009653(−6)	3600	0.078	8.869349(−2)	7200	0.133
0.00625	**EFTDRKN5**	2.219678(−32)	7200	0.170	1.003659(−27)	14 400	0.280
	EFRKN5D	1.682162(−21)	9600	0.140	7.608008(−17)	19 200	0.245
	EFRKN5T	1.948979(−11)	14 400	0.230	8.378591(−7)	28 800	0.364
	PFRKN5	2.870109(−4)	9600	0.144	1.264374(+1)	19 200	0.256
	TDRKN5C	2.868211(−10)	14 400	0.226	1.303099(−5)	28 800	0.360
	RKM5	6.321020(−8)	7200	0.161	2.789375(−3)	14 400	0.264
0.003125	**EFTDRKN5**	3.391068(−37)	14 400	0.338	1.533248(−32)	28 800	0.551
	EFRKN5D	8.216344(−25)	19 200	0.285	3.715420(−20)	38 400	0.487
	EFRKN5T	3.064743(−13)	28 800	0.457	1.317743(−8)	57 600	0.716
	PFRKN5	3.660959(−5)	19 200	0.293	1.612761(+0)	38 400	0.507
	TDRKN5C	4.503326(−12)	28 800	0.448	2.045673(−7)	57 600	0.710
	RKM5	1.981727(−9)	14 400	0.314	8.744567(−5)	28 800	0.515

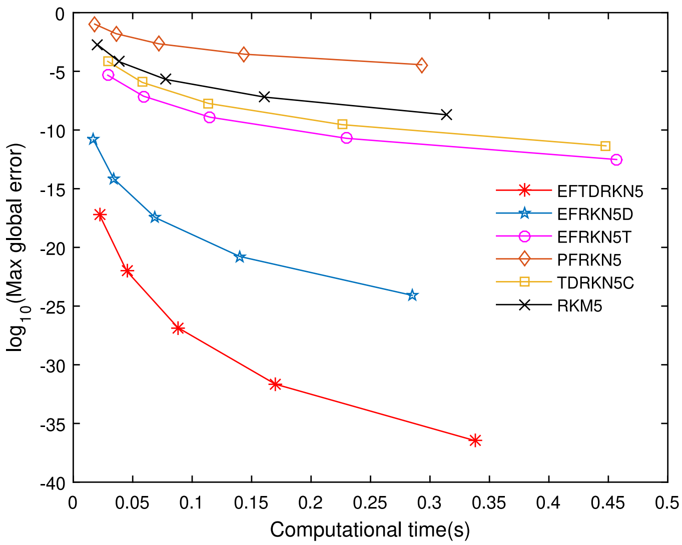


Fig. 6. Numerical curves of selected methods for Problem 4 with $b = 5$ and $h = 0.025 - 0.005i, i = 0, 1, \ldots, 4$.

to order five. Then, the zero stability and stability region of the proposed method are analyzed to assess the stability performance of TDRKN5 method. The proposed method is proven to have zero stability and the region of absolute stability, based on the test problem, is plotted.

The exponentially-fitting technique is implemented into TDRKN5 method. Exponential functions, $e^{-\lambda x}$ and $e^{\lambda x}$ are integrated. Coefficients such as $b_i, A_{i,j}, \hat{A}_{i,j}, \delta_i$ and $\hat{\delta}_i$ are adopted with the product of fitting frequency, λ and step size, h. When the fitting frequency approaches zero, EFTDRKN5 method will tend to reduce to classical TDRKN5 method of a similar algebraic order. The

Table 8
Comparison between EFTDRKN5 method with existing methods for Problem 5.

h	Methods	$b = 10$			$b = 20$		
		MAXERR	FE	Time (s)	MAXERR	FE	Time (s)
0.1	**EFTDRKN5**	3.069208(−18)	600	0.034	1.388955(−13)	1200	0.052
	EFRKN5D	4.692655(−5)	800	0.030	1.033619(+0)	1600	0.045
	EFRKN5T	2.827944(−6)	1200	0.042	1.110238(−1)	2400	0.068
	PFRKN5	5.262416(−2)	800	0.024	2.321863(+3)	1600	0.045
	TDRKN5C	8.073879(−6)	1200	0.042	3.842141(−1)	2400	0.067
	RKM5	2.228469(−4)	600	0.032	1.038527(+1)	1200	0.057
0.05	**EFTDRKN5**	4.730533(−23)	1200	0.067	2.139895(−18)	2400	0.104
	EFRKN5D	1.466702(−6)	1600	0.052	3.230609(−2)	3200	0.090
	EFRKN5T	4.611276(−8)	2400	0.080	1.819363(−3)	4800	0.135
	PFRKN5	8.049962(−3)	1600	0.049	3.549970(+2)	3200	0.090
	TDRKN5C	1.318666(−7)	2400	0.080	6.249030(−3)	4800	0.133
	RKM5	7.164989(−6)	1200	0.063	3.333208(−1)	2400	0.114
0.025	**EFTDRKN5**	7.254547(−28)	2400	0.130	3.280833(−23)	4800	0.207
	EFRKN5D	4.582614(−8)	3200	0.111	1.009383(−3)	6400	0.176
	EFRKN5T	7.360738(−10)	4800	0.142	2.911275(−5)	9600	0.269
	PFRKN5	1.100025(−3)	3200	0.101	4.848727(+1)	6400	0.177
	TDRKN5C	2.106394(−9)	4800	0.140	9.961559(−5)	9600	0.262
	RKM5	2.270928(−7)	2400	0.120	1.055557(−2)	4800	0.223
0.0125	**EFTDRKN5**	1.109738(−32)	4800	0.229	5.018056(−28)	9600	0.412
	EFRKN5D	1.431834(−9)	6400	0.203	3.153812(−5)	12 800	0.352
	EFRKN5T	1.162480(−11)	9600	0.245	4.603369(−7)	19 200	0.511
	PFRKN5	1.434210(−4)	6400	0.196	6.320003(+0)	12 800	0.353
	TDRKN5C	3.327699(−11)	9600	0.243	1.572139(−6)	19 200	0.508
	RKM5	7.146806(−9)	4800	0.215	3.320543(−4)	9600	0.438
0.00625	**EFTDRKN5**	1.699394(−37)	9600	0.404	7.682824(−33)	19 200	0.802
	EFRKN5D	4.474036(−11)	12 800	0.394	9.854680(−7)	25 600	0.702
	EFRKN5T	1.826116(−13)	19 200	0.484	7.235728(−9)	38 400	1.019
	PFRKN5	1.829927(−5)	12 800	0.385	8.062582(−1)	25 600	0.697
	TDRKN5C	5.228225(−13)	19 200	0.478	2.468776(−8)	38 400	1.003
	RKM5	2.241242(−10)	9600	0.423	1.041109(−5)	19 200	0.884

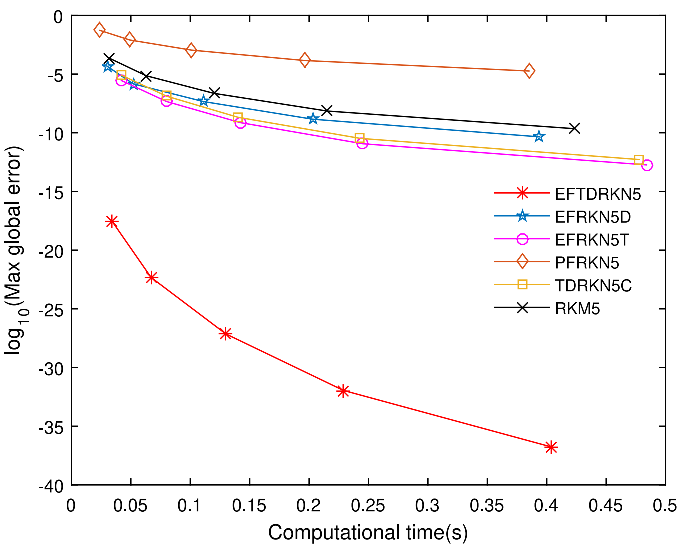


Fig. 7. Numerical curves of selected methods for Problem 5 with $b = 10$ and $h = \frac{0.2}{2^i}, i = 0, 1, \ldots, 4$.

proposed method with exponentially-fitting technique simulates exactly some standard exponential function and contributes to excellent accuracy and efficiency in solving second-order ODEs with exponential solutions.

Numerical experiments are carried out in terms of maximum global error versus time of computation for a proposed method with fitting technique and other existing methods. Six different second-order initial value problems with exponential solutions are employed in the numerical test to evaluate the numerical performance of all approaches. EFTDRKN5 method is compared to the existing fifth-order exponential-fitted and phase-fitted Runge–Kutta–Nyström methods, including EFRKN5D, EFRKN5T, PFRKN5,

Table 9
Comparison between EFTDRKN5 method with existing methods for Problem 6.

h	Methods	$b = 5$			$b = 10$		
		MAXERR	FE	Time (s)	MAXERR	FE	Time (s)
0.02	**EFTDRKN5**	1.409648(−25)	750	0.074	3.104955(−21)	1500	0.129
	EFRKN5D	7.610983(−9)	1000	0.063	1.676431(−4)	2000	0.099
	EFRKN5T	1.006095(−9)	1500	0.080	2.216072(−5)	3000	0.151
	PFRKN5	1.053097(−6)	1000	0.061	2.319612(−2)	2000	0.100
	TDRKN5C	4.820947(−9)	1000	0.080	1.061884(−4)	3000	0.147
	RKM5	1.349528(−8)	750	0.066	2.972535(−4)	1500	0.113
0.01	**EFTDRKN5**	2.155764(−30)	1500	0.131	4.748386(−26)	3000	0.254
	EFRKN5D	1.200699(−10)	2000	0.106	2.644715(−6)	4000	0.194
	EFRKN5T	1.574616(−11)	3000	0.153	3.468323(−7)	6000	0.294
	PFRKN5	3.327795(−8)	2000	0.101	7.330548(−4)	4000	0.196
	TDRKN5C	7.534236(−11)	2000	0.152	1.659526(−6)	6000	0.289
	RKM5	4.778754(−10)	1500	0.126	1.052591(−5)	3000	0.225
0.005	**EFTDRKN5**	3.292891(−35)	3000	0.236	7.253072(−31)	6000	0.520
	EFRKN5D	1.884979(−12)	4000	0.201	4.151943(−8)	8000	0.396
	EFRKN5T	2.462262(−13)	6000	0.304	5.423492(−9)	12 000	0.595
	PFRKN5	1.045489(−9)	4000	0.194	2.303560(−5)	8000	0.399
	TDRKN5C	1.177283(−12)	4000	0.302	2.593139(−9)	12 000	0.586
	RKM5	1.581039(−11)	3000	0.256	3.482472(−7)	6000	0.456
0.0025	**EFTDRKN5**	4.415846(−40)	6000	0.465	1.107288(−35)	12 000	1.023
	EFRKN5D	2.952191(−14)	8000	0.380	6.502632(−10)	16 000	0.788
	EFRKN5T	3.848747(−15)	12 000	0.590	8.477429(−11)	24 000	1.196
	PFRKN5	3.273223(−11)	8000	0.370	7.218572(−7)	16 000	0.790
	TDRKN5C	1.839528(−14)	8000	0.586	4.051830(−10)	24 000	1.168
	RKM5	5.077691(−13)	4800	0.538	1.118437(−8)	12 000	0.902
0.00125	**EFTDRKN5**	7.672505(−45)	9600	0.920	1.689981(−40)	24 000	1.982
	EFRKN5D	4.618183(−16)	12 800	0.752	1.017222(−11)	32 000	1.562
	EFRKN5T	6.014794(−17)	19 200	1.258	1.324846(−12)	48 000	2.405
	PFRKN5	4.023086(−12)	12 800	0.730	2.258921(−8)	32 000	1.568
	TDRKN5C	2.874271(−16)	19 200	1.248	6.331004(−12)	48 000	2.389
	RKM5	1.608171(−14)	9600	1.076	3.542236(−10)	24 000	1.793

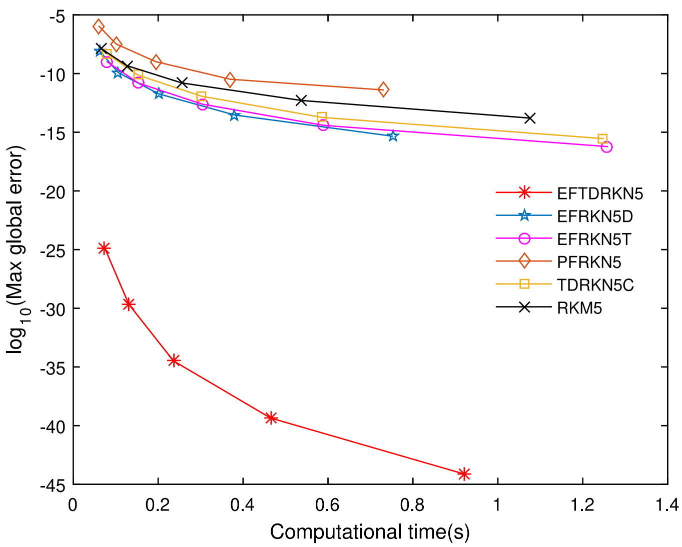


Fig. 8. Numerical curves of selected methods for Problem 6 with $b = 5$ and $h = \frac{0.1}{2^i}, i = 0, 1, \ldots, 4$.

TDRKN5C and RKM5 methods. Numerical performance is assessed using the maximum global error, computational time and number of function evaluations. The numerical results are displaced in Tables 4–10 and Figs. 3–9.

In a nutshell, our proposed method is very effective in solving all kinds of special class of second-order ODEs with exponential solutions and surpasses all selected methods in numerical efficiency by generating the least maximum global error in a similar computational time for solving Problems 1–6. In coping with the logistic growth model, the proposed method is still working and able to produce relatively low global truncation error compared to others. However, there is a limitation of our proposed method,

Table 10
Comparison between EFTDRKN5 method with existing methods for logistic model problem.

h	Methods	$b = 2$		
		MAXERR	FE	Time (s)
0.02	EFTDRKN5	3.600902(−13)	300	0.016
	EFRKN5D	6.652165(−13)	400	0.011
	EFRKN5T	8.786405(−13)	600	0.025
	PFRKN5	7.561200(−10)	300	0.010
	TDRKN5C	8.839655(−13)	600	0.025
	RKM5	3.308139(−13)	300	0.014
0.01	EFTDRKN5	1.091609(−14)	600	0.032
	EFRKN5D	2.061077(−14)	800	0.022
	EFRKN5T	2.780900(−14)	1200	0.050
	PFRKN5	9.958537(−11)	600	0.021
	TDRKN5C	2.797904(−14)	1200	0.050
	RKM5	1.045092(−14)	600	0.027
0.005	EFTDRKN5	3.357652(−16)	1200	0.065
	EFRKN5D	6.409315(−16)	1600	0.045
	EFRKN5T	8.745954(−16)	2400	0.100
	PFRKN5	1.276645(−11)	1200	0.043
	TDRKN5C	8.799665(−16)	2400	0.099
	RKM5	3.283438(−16)	1200	0.054
0.0025	EFTDRKN5	1.040811(−17)	2400	0.129
	EFRKN5D	1.963341(−17)	3200	0.091
	EFRKN5T	2.741863(−17)	4800	0.195
	PFRKN5	1.615742(−12)	2400	0.087
	TDRKN5C	2.758739(−17)	4800	0.194
	RKM5	1.028799(−17)	2400	0.108
0.00125	EFTDRKN5	3.239608(−19)	4800	0.255
	EFRKN5D	6.241578(−19)	6400	0.180
	EFRKN5T	8.582391(−19)	9600	0.385
	PFRKN5	2.032151(−13)	4800	0.178
	TDRKN5C	8.635270(−19)	9600	0.384
	RKM5	3.218899(−19)	4800	0.215

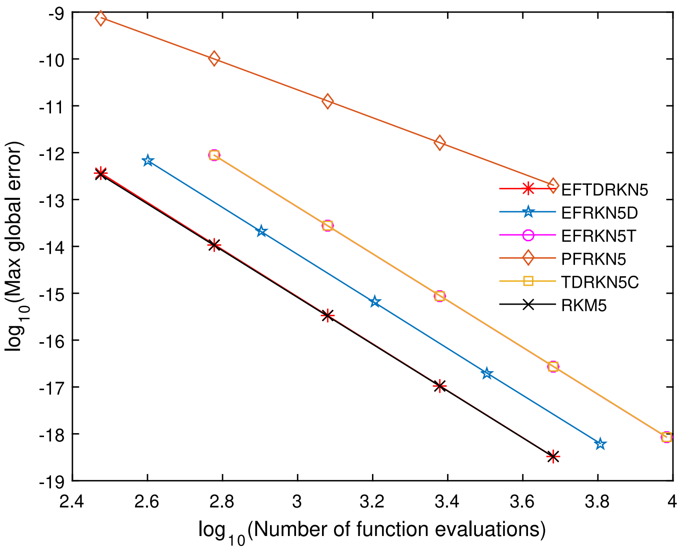


Fig. 9. Numerical curves of selected methods for Problem 7 with $b = 2$ and $h = \frac{0.02}{2^i}, i = 0, 1, \ldots, 4$.

which is not able to generate good approximation in dealing with second-order ODEs with non-exponential solutions. However, the purpose of developing an exponentially-fitting technique for solving differential equations is not solely limited to its effectiveness on problems with exponential solutions. While it may have limitations in terms of applicability to a broader range of problems, there can still be value in developing and utilizing such a technique. Insights gained from an exponentially-fitting technique can inspire or contribute to the development of more versatile or generalized methods that can handle a wider range of problem types, apart from problems with exponential solutions. It can serve as a building block for future advancements which are effective in all types of problems, regardless of consisting exponential solutions or not. Also, since parameters are frequency-dependent, if we use the proposed method with exponentially-fitting technique to solve differential equations with most probably non-exponential solutions, we can set $v = 0$, then the proposed method will turn into a classical method without exponential-fitting property, which is more suitable and useful to solve the non-exponential differential problem by generating low global truncation error.

For future research, another type of fitting technique can be implemented to classical Runge–Kutta–Nyström method with minimal function evaluation, such as phase-fitting and trigonometrically-fitting techniques for solving different types of second-order ODEs. Also, the proposed method can be further analyzed in terms of stability and error bound to determine the stability region and numerical limitation. Based on the analysis, a suitable technique or formula modification can be implemented to solve a wider range of differential equations effectively.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was supported by the Grant Schemes (Ref. No. GGPM-2023-029 and Ref. No. ST-2022-015) awarded by Universiti Kebangsaan Malaysia. The authors declare that there is no conflict of interest related to the publication of this paper.

References

[1] R. A'Ambrosio, P. Beatrice, G. Santomauro, Revised exponentially fitted Runge–Kutta–Nyström methods, Appl. Math. Lett. 30 (2014) 56–60.
[2] R. Abdelrahim, Z. Omar, Direct solution of second-order ordinary differential equation using a single-step hybrid block method of order five, Math. Comput. Appl. 21 (2016) 12.
[3] C.E. Abhulimen, L. Ukpebor, A family of exponentially fitted multiderivative method for stiff differential equations, J. Adv. Math. 13 (2017) 7155–7162.
[4] A.I. Asnor, S.A.M. Yatim, Z.A. Ibrahim, Solving directly higher order ordinary differential equations by using variable order block backward differentiation formulae, Symmetry 11 (2019) 1289.
[5] G.V. Berghe, H.D. Meyer, M.V. Daele, T.V. Hecke, Exponentially fitted Runge–Kutta methods, J. Comput. Appl. Math. 125 (2000) 107–115.
[6] J.C. Butcher, Numerical methods for ordinary differential methods in the 20th century, J. Comput. Appl. Math. 125 (2000) 1–29.
[7] T. Buvoli, Exponential polynomial block methods, J. Sci. Comput. 43 (2021) http://dx.doi.org/10.1137/20M132134.
[8] Z. Chen, Z. Qiu, J. Li, X. You, Two-derivative Runge–Kutta–Nyström methods for second-order ordinary differential equations, Numer. Algorithms 70 (2015) 897–927.
[9] M.A. Demba, P. Kumam, W. Watthayu, P. Phairatchatniyom, Embedded exponentially-fitted explicit Runge–Kutta–Nyström methods for solving periodic problems, Computation 8 (2020) 32.
[10] M.K. Duromola, A.L. Momoh, Hybrid numerical method with block extension for direct solution of third order ordinary differential equations, Amer. J. Comput. Math. 9 (2019) 68–80.
[11] J.O. Ehigie, V.T. Luan, A.O. Solomon, X. You, Exponentially fitted two-derivative DIRK methods for oscillatory differential equations, Appl. Math. Comput. 418 (2022) 126770.
[12] J.M. Franco, An embedded pair of exponentially fitted explicit Runge–Kutta methods, J. Comput. Appl. Math. 149 (2002) 407–414.
[13] J.M. Franco, L. Randez, Eighth-order explicit two-step hybrid methods with symmetric nodes and weights for solving orbital and oscillatory IVPs, Int. J. Modern Phys. C 29 (2018) http://dx.doi.org/10.1142/S012918311850002X.
[14] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I Nonstiff Problems, Springer, Berlin, 1993.
[15] A. Hasan, M.A. Halim, M.A. Meia, Application of linear differential equation in an analysis transient and steady response for second order RLC closed series circuit, Circuits Syst. Signal Process. 5 (2019) 1–8.
[16] P. Henrici, Discrete Variable Methods in Ordinary Differential Equations, John Wiley & Sons, New York, 1962.
[17] F. Ismail, K. Hussain, N. Senu, A sixth-order rkfd method with four-stage for directly solving special fourth-order ODEs, Sains Malaysiana, 45 (11) 1747–1754.
[18] L.G. Ixaru, Runge–Kutta methods with equation dependent coefficients, Comput. Phys. Commun. 183 (2012) 63–69.
[19] L.G. Ixaru, Numerical computation of the coefficients in exponential fitting, Numer. Algorithms 87 (2020) 1097–1106.
[20] V.V. Kovalnogov, R.V. Fedorov, T.V. Karpukhina, T.E. Simos, C. Tsitouras, Runge–Kutta pairs of orders 5(4) trained to best address Keplerian type orbits, 9 (2021) 2400.
[21] K.C. Lee, N. Senu, A. Ahmadian, S.N.I. Ibrahim, High-order exponentially fitted and trigonometrically fitted explicit two-derivative Runge–Kutta-type methods for solving third-order oscillatory problems, Math. Sci. 16 (2021) 281–297.
[22] T.S. Mohamed, N. Senu, Z.B. Ibrahim, N.M.A. Nik Long, Exponentially fitted and trigonometrically fitted two-derivative Runge–Kutta–Nyström methods for solving $y(x) = f(x, y, y')$, Math. Probl. Eng. (2018) 7689854, 19pages.
[23] T. Monovasilis, Z. Kalogiratou, T.E. Simos, Construction of exponentially fitted symplectic Runge–Kutta–Nyström methods from partitioned Runge–Kutta methods, Appl. Math. Inf. Sci. 9 (2015) 1923–1930.
[24] F.F. Ngwane, S. Jator, A trigonometrically fitted block method for solving oscillatory second-order initial value problems and Hamiltonian systems, Int. J. Differ. Equ. (2017) 1–14.
[25] M.M. Salih, F. Ismail, N. Senu, Fifth order Runge–Kutta–Nyström methods for solving linear second order oscillatory problems, Far East J. Appl. Mathe. 95 (2016) 141–156.
[26] M.M. Salih, F. Ismail, N. Senu, Efficient two-derivative Runge–Kutta–Nyström methods for solving general second order ordinary differential equations $y(x) = f(x, y, y')$, Discrete Dyn. Nat. Soc. (2018) 2393015.
[27] A. Tsoularis, J. Wallace, Analysis of logistic growth models, Math. Biosci. 179 (2002) 21–55.
[28] W. Zhai, B. Chen, A fourth order implicit symmetric and symplectic exponentially fitted Runge–Kutta–Nyström method for solving oscillatory problems, Amer. Inst. Math. Sci. 9 (2019) 71–84.