Indian Journal of Science and Technology, *Vol 9(28), DOI: 10.17485/ijst/2016/v9i28/97748, July 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Two Derivative Runge-Kutta Method with FSAL Property for the Solution of First Order Initial Value Problems

Nur Amirah Ahmad* and Norazak Senu

Department of Mathematics and Institute for Mathematical Research, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

Abstract

A new Two Derivative Runge-Kutta method (TDRK) based on First Same as Last (FSAL) technique for the numerical solution of first order Initial Value Problems (IVPs) is derived. We present a fourth order three stages TDRK method designed using the FSAL property. The stability of the new method is analyzed. The numerical experiments are carried out to show the efficiency of our methods in comparison with other existing Runge-Kutta methods (RK).

Keywords: Explicit Methods; FSAL Technique; IVPs; TDRK Methods

1. Introduction

We consider the numerical solution of the IVPs for first order Ordinary Differential Equations (ODEs) in the form of

$$y' = f(x, y), \qquad y(x_0) = y_0. \tag{2}$$

In the last few years, many researches have proposed several TDRK methods as well as implementing FSAL technique. Chan and Tsai[1] in their paper presented a theoretical formulation used for the derivation of TDRK methods. They constructed a special class of explicit methods of order up to seven that involve one f -evaluation and a minimum number of g -evaluations. In[2] developed a new Trigonometrically Fitted TDRK method of algebraic order five, analyze the linear stability and phase properties of the new method. Numerical results are reported to indicate the efficiency and competence of the new method compared with few highly efficient methods.

Other from that, in[3] derived three practical exponentially fitted TDRK (EFTDRK) methods where numerical results show that the new EFTDRK methods are more accurate and more efficient than their prototype TDRK methods or some well-known RK methods of the same order and the traditional exponentially fitted RK method in the literature. Recently, in[4] introduced a new class of implicit TDRK collocation methods designed for the numerical solution of systems of equations and showed how they have been implemented in an efficient parallel computing environment.

In the earliest year, in[5] derived a family of embedded RK formulae RK5(4) by implementing FSAL technique. An extended region of absolute stability and a 'small' principal truncation terms in the fifth order formulae are presented in their paper. In[6] derived explicit Exponentially Fitted RKN methods (EFRKN) with two and three stages and third and fourth algebraic orders as well as a 4(3) embedded pair based on the FSAL technique for the numerical integration of second order IVPs with oscillatory solutions.

By using FSAL property[7] presented extended Runge-Kutta Nyström methods (RKN) for numerical integration of perturbed oscillators where the derivation of family of fixed step-size ERKN methods and the embedded pairs of ERKN methods are presented. The next method proposed by[8] is higher order RK (pair) method which are pairs of

**Author for correspondence*

embedded RK methods of fourth and fifth order as well as a new RK method of order five are specially adapted to the numerical integration of IVPs with oscillatory solutions.

In[9] proposed a new way for constructing efficient embedded modified RK methods for the numerical solution of the Schrödinger equation. The embedded pairs are constructed based on the FSAL technique which has fourth and fifth algebraic order. The new pair is more efficient compared to other well-known comparable embedded RK pairs based on the applications of the new pair to several problems related to the Schrödinger equation.

Hence, in this paper, we construct a fourth order three stages TDRK method by implementing the FSAL property. In Section 2, an overview of TDRK method is given. The new FSAL TDRK method is constructed and the stability of the new method is analyzed in Section 3. Meanwhile in Section 4, the numerical results are discussed. The discussion and conclusion of the new method are reported in Section 5 and 6 respectively.

2. Two Derivative Runge-Kutta Methods

In this paper, we consider the following scalar ODE as given in Equation (1) with $f: \Re^N \to \Re^N$. In this study, we assume that the second derivative is also known where

$$y'' = g(y) := f'(y)f(y), \qquad g: \Re^N \to \Re^N. \quad (2)$$

When a general s -stage TDRK method is applied to the ODE Equation (1) and (2), the following equations are obtained

$$Y_i = y_n + h\sum_{j=1}^{s} a_{ij} f(y_j) + h^2 \sum_{j=1}^{s} \hat{a}_{ij} g(Y_j), \quad (3)$$

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i f(y_i) + h^2 \sum_{i=1}^{s} \hat{b}_i g(Y_i). \quad (4)$$

Where, $i = 1, \ldots, s.$

An explicit TDRK method can be presented using the Butcher Tableau with the coefficient in Equation (3) and (4) as given in Table 1.

Explicit methods which have minimal number of function evaluations can be constructed by considering the methods in the form

Table 1. Butcher tableau for explicit TDRK method

$$\begin{array}{c|cccccc|cccccc}
0 & 0 & & & & & & 0 & & & & & \\
c_2 & a_{2,1} & 0 & & & & & \hat{a}_{2,1} & 0 & & & & \\
c_3 & a_{3,1} & a_{3,2} & 0 & & & & \hat{a}_{3,1} & \hat{a}_{3,2} & 0 & & & \\
\vdots & \vdots & \vdots & a_{4,3} & \ddots & & & \vdots & \vdots & \hat{a}_{4,3} & \ddots & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & \ddots & & \vdots & \vdots & \vdots & \ddots & \ddots & \\
c_s & a_{s,1} & a_{s,2} & \cdots & \cdots & a_{s-1,s} & 0 & \hat{a}_{s,1} & \hat{a}_{s,2} & \cdots & \cdots & \hat{a}_{s-1,s} & 0 \\
\hline
 & b_1 & b_2 & \cdots & \cdots & b_{s-1} & b_s & \hat{b}_1 & \hat{b}_2 & \cdots & \cdots & \hat{b}_{s-1} & \hat{b}_s
\end{array} \quad (5)$$

$$Y_i = y_n + hc_i f(x_n, y_n) + h^2 \sum_{j=1}^{i-1} \hat{a}_{ij} g(x_n + c_j h, Y_j), \quad (6)$$

$$y_{n+1} = y_n + hf(x_n, y_n) + h^2 \sum_{i=1}^{s} \hat{b}_i g(x_n \quad (7)$$

Where, $i = 2, \ldots, s.$

This method is called Special Explicit TDRK methods. The special part of this method is that it involves only one evaluation of f per step which makes it different from traditional explicit RK methods.

Table 2. Butcher tableau for special explicit TDRK method

$$\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & \hat{a}_{2,1} & 0 & & & & \\
c_3 & \hat{a}_{3,1} & \hat{a}_{3,2} & 0 & & & \\
\vdots & \vdots & \vdots & \hat{a}_{4,3} & \ddots & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \\
c_s & \hat{a}_{s,1} & \hat{a}_{s,2} & \cdots & \cdots & \hat{a}_{s-1,s} & 0 \\
\hline
 & \hat{b}_1 & \hat{b}_2 & \cdots & \cdots & \hat{b}_{s-1} & \hat{b}_s
\end{array} \quad (8)$$

According to[1], the order conditions for special explicit TDRK methods up to four are listed as below:

- Order 2:

$$\sum_{i=1}^{s} \hat{b}_i = \frac{1}{2}. \quad (9)$$

- Order 3:

$$\sum_{i=2}^{s} \hat{b}_i c_i = \frac{1}{6}. \quad (10)$$

- Order 4:

$$\sum_{i=2}^{s} \hat{b}_i c_i^2 = \frac{1}{12}. \quad (11)$$

The following simplifying assumption is used in practice

$$\sum_{i=1}^{s} \hat{a}_{ij} = \frac{1}{2} c_i^2, \tag{12}$$

3. A Fourth order Two Derivative Runge-Kutta Method with FSAL Property

A particularly interesting special class of explicit RK methods is that for which the coefficients have a special structure known as First Same as Last (FSAL) where

$$\hat{b}_i = \hat{a}_{s,i}, \qquad i = 1, \dots, s-1 \text{ and } \hat{b}_s = 0. \tag{13}$$

The advantage of FSAL methods is that the function value k_s at the end of one integration step is the same as the first function value k_i at the next integration step.

We implement the FSAL technique into the TDRK methods where the order conditions in Equation (9)–(11) along with the simplifying assumption in Equation (12) need to be satisfied in order for a method to be a TDRK method. In particular, we consider in this paper a three stage explicit TDRK method given by the following Butcher tableau which has FSAL property.

$$\begin{array}{c|ccc} 0 & 0 & & \\ c_2 & \hat{a}_2 & 0 & \\ 1 & \hat{a}_3 & \hat{a}_3 & 0 \\ \hline & \hat{a}_3 & \hat{a}_3 & 0 \end{array} \tag{14}$$

3.1 Construction of the New Method

Evaluating the simplifying assumption in Equation (12) we will get

$$\hat{a}_{21} = \frac{c_2^2}{2}, \qquad \hat{a}_{31} = \frac{1}{2} - \hat{a}_{32}. \tag{15}$$

According to the order conditions in Equation (9)–(11) we have

$$\frac{1}{2} - \frac{1}{2} = 0, \tag{16}$$

$$\hat{a}_{32} c_2 - \frac{1}{6} = 0, \tag{17}$$

$$\hat{a}_{32} c_2^2 - \frac{1}{12} = 0. \tag{18}$$

Solving Equation (16)–(18) we obtain

$$\hat{a}_{32} = \frac{1}{3}, \qquad c_2 = \frac{1}{2}.$$

The new method can be written in the following Butcher tableau:

$$\begin{array}{c|ccc} 0 & 0 & & \\ \frac{1}{2} & \frac{1}{8} & 0 & \\ 1 & \frac{1}{6} & \frac{1}{3} & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & 0 \end{array} \tag{19}$$

This new method is called as FSALTDRK3(4) method.

3.2 Stability of the New Method

In this sub-section, we will investigate the linear stability of the new method. The stability function of TDRK method is given as follows:

$$R(z) = 1 + z b^T (I - zA - z^2 \hat{A})^{-1} e + z^2 \hat{b}^T (I - zA - z^2 \hat{A})^{-1} e. \tag{20}$$

Meanwhile for special explicit TDRK method, we consider the following test equation

$$y' = \lambda y, \qquad y'' = \lambda^2 y \text{ where } \lambda > 0. \tag{21}$$

Applying Equation (21) to the special explicit TDRK method which is Equation (6) and (7) produces the difference equation

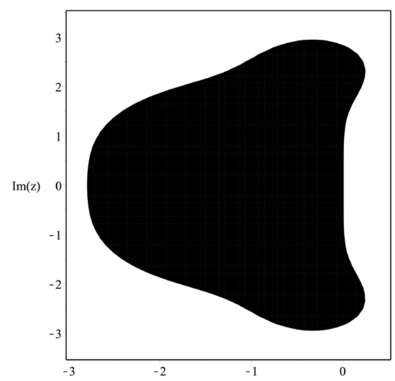

Figure 1. Stability region of method FSALTDRK3(4).

$$y_{n+1} = H(z) y_n, \qquad z = \lambda h, \tag{22}$$

where

$$H(z) = (1 + z^2 \hat{b}(I - z^2 \hat{A})^{-1} e) + (z + z^3 \hat{b}(I - z^2 \hat{A})^{-1} c) \tag{23}$$

$\hat{A}$ is the coefficient of the new method with

$$I=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},\qquad e=[1\quad 1\quad 1]^T,\qquad c=\begin{bmatrix}0\\ \frac{1}{2}\\ 1\end{bmatrix},\qquad \hat{b}=\begin{bmatrix}\frac{1}{6}&\frac{1}{3}&0\end{bmatrix}. \tag{24}$$

The stability function of the FSALTDRK3(4) method is

$$H(z)=1+z+\frac{1}{2}z^2+\frac{1}{6}z^3+\frac{1}{24}z^4 \tag{25}$$

The stability region of the FSALTDRK3(4) method is plotted in Figure 1.

Meanwhile, the stability interval of this new method is $(-2.78,0)$.

4. Problems Tested and Numerical Results

In this section, we compare the performance of the proposed method FSALTDRK3(4) with existing RK methods by considering the following problems. All problems below are tested using C code for solving differential equations.

Problem 1 (In[10])

$$y_1'=y_2,\qquad y_2'=-\frac{y_1}{\left(\sqrt{y_1^2+y_3^2}\right)^3},$$

$$y_3'=y_4,\qquad y_4'=-\frac{y_3}{\left(\sqrt{y_1^2+y_3^2}\right)^3},$$

$$y_1(0)=1,\qquad y_2(0)=0,\qquad y_3(0)=0,\qquad y_4(0)=1.$$

The exact solution is

$$y_1(x)=\cos(x),\qquad y_2(x)=-\sin(x),$$

$$y_3(x)=\sin(x),\qquad y_4(x)=\cos(x).$$

Problem 2 (In[11])

$$y_1'=y_2,\qquad y_2'=-16y_1+116e^{-10x},$$

$$y_3'=y_4,\qquad y_4'=-16y_3+116e^{-10x},$$

$$y_1(0)=1.1,\qquad y_2(0)=-10,\qquad y_3(0)=1,\qquad y_4(0)=-9.6.$$

The exact solution is

$$y_1(x)=0.1\cos(4x)+e^{-10x},$$

$$y_2(x)=-0.4\sin(4x)-10e^{-10x},$$

$$y_3(x)=0.1\sin(4x)+e^{-10x},$$

$$y_4(x)=0.4\cos(4x)-10e^{-10x}.$$

Problem 3 (In[11])

$$y_1'=y_2,\qquad y_2'=-y_3+\sin(\pi x),$$

$$y_3'=y_4,\qquad y_4'=-y_1+1+\pi\cos(\pi x),$$

$$y_1(0)=0,\qquad y_2(0)=-1,\qquad y_3(0)=1,\qquad y_4(0)=1+\pi.$$

The exact solution is

$y_1(x)=1-e^x,$	$y_2(x)=-e^x,$
$y_3(x)=e^x+\sin(\pi x),$	$y_4(x)=e^x+\pi\cos(\pi x).$

Problem 4 (In[12])

$$y'=-y\tan(x)-\frac{1}{\cos(x)},\qquad y(0)=1.$$

The exact solution is $y(x)=\cos(x)-\sin(x)$.

Problem 5 (In[13])

$$y'=-2xy,\qquad y(0)=1.$$

The exact solution is $y(x)=e^{-x^2}$.

The following notations are used in Figures (2-6):

- **FSALTDRK3(4):** New TDRK method with FSAL property of fourth order three stages derived in this paper.
- **RKM5(4):** Existing fourth order five stages Merson's method given in[14].
- **RKE6(4):** Existing fourth order six stages England's method given in[14].
- **RKB5(4):** Existing fourth order five stages RK method derived by[15].
- **RK4:** Existing fourth order four stages classical RK method given in[14].
- **RK4(3/8 rule):** Existing fourth order four stages classical 3/8 rule RK method given in[14].

The numerical results and the graphic performance of these methods are presented in the following Tables (3-7) and Figures (2-6):

Table 3. Numerical results for problem 1

h	Methods	T=10	Function Evaluation
0.1	FSALTDRK3(4)	1.020036 (-5)	303
	RKM5(4)	1.334286 (-5)	505
	RKE6(4)	4.016703 (-5)	606
	RKB5(4)	5.633442 (-5)	505
	RK4	4.135314 (-5)	404
	RK4(3/8 rule)	1.546268 (-4)	404
0.05	FSALTDRK3(4)	5.659144 (-7)	600

	RKM5(4)	6.941171 (-7)	1000
	RKE6(4)	2.456736 (-6)	1200
	RKB5(4)	3.015911 (-6)	1000
	RK4	2.001408 (-6)	800
	RK4(3/8 rule)	8.018841 (-6)	800
0.025	FSALTDRK3(4)	3.311131 (-8)	1200
	RKM5(4)	3.897247 (-8)	2000
	RKE6(4)	1.515634 (-7)	2400
	RKB5(4)	1.723513 (-7)	2000
	RK4	1.067869 (-7)	1600
	RK4(3/8 rule)	4.481370 (-7)	1600
0.0125	FSALTDRK3(4)	1.998886 (-9)	2403
	RKM5(4)	2.297481 (-9)	4005
	RKE6(4)	9.208075 (-9)	4806
	RKB5(4)	1.026252 (-8)	4005
	RK4	6.101190 (-9)	3204
	RK4(3/8 rule)	2.632901 (-8)	3204
0.00625	FSALTDRK3(4)	1.225219 (-10)	4803
	RKM5(4)	1.395175 (-10)	8005
	RKE6(4)	5.855847 (-10)	9606
	RKB5(4)	6.257205 (-10)	8005
	RK4	3.636633 (-10)	6404
	RK4(3/8 rule)	1.593087 (-9)	6404

Table 4. Numerical results for problem 2

h	Methods	T=10	Function Evaluation
0.1	FSALTDRK3(4)	3.575835 (-3)	303
	RKM5(4)	8.586984 (-3)	505
	RKE6(4)	4.395414 (-3)	606
	RKB5(4)	3.731913 (-3)	505
	RK4	4.395414 (-3)	404
	RK4(3/8 rule)	7.375759 (-3)	404
0.05	FSALTDRK3(4)	1.693585 (-4)	600
	RKM5(4)	5.279925 (-4)	1000
	RKE6(4)	1.920408 (-4)	1200
	RKB5(4)	2.188156 (-4)	1000
	RK4	1.920408 (-4)	800
	RK4(3/8 rule)	4.240103 (-4)	800
0.025	FSALTDRK3(4)	9.950386 (-6)	1200
	RKM5(4)	3.272659 (-5)	2000
	RKE6(4)	1.033100 (-5)	2400
	RKB5(4)	1.361399 (-5)	2000
	RK4	1.033100 (-5)	1600
	RK4(3/8 rule)	2.556161 (-5)	1600
0.0125	FSALTDRK3(4)	6.131551 (-7)	2403
	RKM5(4)	2.032682 (-6)	4005
	RKE6(4)	5.934668 (-7)	4806
	RKB5(4)	8.517467 (-7)	4005
	RK4	5.934668 (-7)	3204
	RK4(3/8 rule)	1.570303 (-6)	3204
0.00625	FSALTDRK3(4)	3.794398 (-8)	4803
	RKM5(4)	1.265868 (-7)	8005
	RKE6(4)	3.829597 (-8)	9606
	RKB5(4)	5.324702 (-8)	8005
	RK4	3.829596 (-8)	6404
	RK4(3/8 rule)	9.734090 (-8)	6404

Table 5. Numerical results for problem 3

h	Methods	T=10	Function Evaluation
0.01	FSALTDRK3(4)	2.183865 (-9)	3003
	RKM5(4)	1.550359 (-1)	5005
	RKE6(4)	1.675961 (-1)	6006
	RKB5(4)	2.091357 (-2)	5005
	RK4	1.675961 (-1)	4004
	RK4(3/8 rule)	1.369084 (-1)	4004
0.005	FSALTDRK3(4)	6.373158 (-11)	6000
	RKM5(4)	9.198903 (-2)	10000
	RKE6(4)	8.135433 (-1)	12000
	RKB5(4)	5.065821 (-2)	10000
	RK4	8.135433 (-1)	8000
	RK4(3/8 rule)	4.476555 (-2)	8000
0.0025	FSALTDRK3(4)	8.056888 (-12)	12000
	RKM5(4)	3.131521 (-1)	20000
	RKE6(4)	1.681169 (-1)	24000
	RKB5(4)	1.232556 (-1)	20000
	RK4	1.681169 (-1)	16000
	RK4(3/8 rule)	1.881734 (-1)	16000
0.00125	FSALTDRK3(4)	2.583933 (-12)	24000
	RKM5(4)	4.383508 (-3)	40000
	RKE6(4)	3.220834 (-1)	48000
	RKB5(4)	2.696577 (-1)	40000
	RK4	3.220834 (-1)	32000
	RK4(3/8 rule)	2.432933 (-1)	32000
0.000625	FSALTDRK3(4)	3.696599 (-12)	48003

	RKM5(4)	2.314545 (-2)	80005
	RKE6(4)	6.878039 (-3)	96006
	RKB5(4)	2.125186 (-2)	80005
	RK4	6.878039 (-3)	64004
	RK4(3/8 rule)	7.583173 (-3)	64004

Table 6. Numerical results for problem 4

h	Methods	T=10	Function Evaluation
0.1	FSALTDRK3(4)	8.570419 (-2)	303
	RKM5(4)	1.428479 (-1)	505
	RKE6(4)	2.105403 (-1)	606
	RKB5(4)	1.319206 (-1)	505
	RK4	2.105403 (-1)	404
	RK4(3/8 rule)	2.218055 (-1)	404
0.05	FSALTDRK3(4)	5.167567 (-3)	600
	RKM5(4)	8.158682 (-3)	1000
	RKE6(4)	1.226338 (-2)	1200
	RKB5(4)	7.669778 (-3)	1000
	RK4	1.226338 (-2)	800
	RK4(3/8 rule)	1.300372 (-2)	800
0.025	FSALTDRK3(4)	3.361561 (-4)	1200
	RKM5(4)	5.130850 (-4)	2000
	RKE6(4)	7.815214 (-4)	2400
	RKB5(4)	4.879689 (-4)	2000
	RK4	7.815214 (-4)	1600
	RK4(3/8 rule)	8.311731 (-4)	1600
0.0125	FSALTDRK3(4)	2.174382 (-5)	2403
	RKM5(4)	3.258069 (-5)	4005
	RKE6(4)	5.000124 (-5)	4806
	RKB5(4)	3.118758 (-5)	4005
	RK4	5.000124 (-5)	3204
	RK4(3/8 rule)	5.325290 (-5)	3204
0.00625	FSALTDRK3(4)	1.355205 (-6)	4803
	RKM5(4)	2.020031 (-6)	8005
	RKE6(4)	3.112604 (-6)	9606
	RKB5(4)	1.938293 (-6)	8005
	RK4	3.112553 (-6)	6404
	RK4(3/8 rule)	3.318062 (-6)	6404

5. Discussion

The efficiency of the method developed is presented in Figures (2-6) by plotting the graph of the decimal logarithm of the maximum global error against the logarithm number of function evaluations. Observing from the graph plotted in Figures (2-5) we can see that FSALTDRK3(4) have the smallest maximum global error and number of function evaluations per step compared to other existing RK methods. Meanwhile according to Table 7 and Figure 6, FSALTDRK3(4) is significantly more efficient in term of the number of function evaluations per step than other existing RK methods although its maximum global error is slightly bigger than RKM5(4).

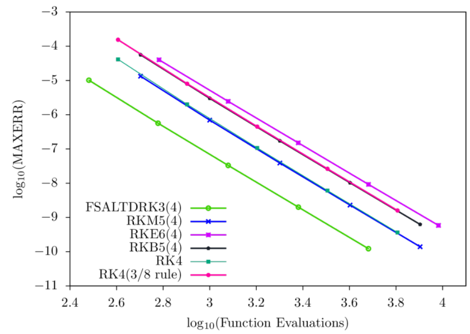

Figure 2. Efficiency graph for problem 1 with $T_{end} = 10$ and $h = i(0.00625), i = 1,2,4,8,16.$

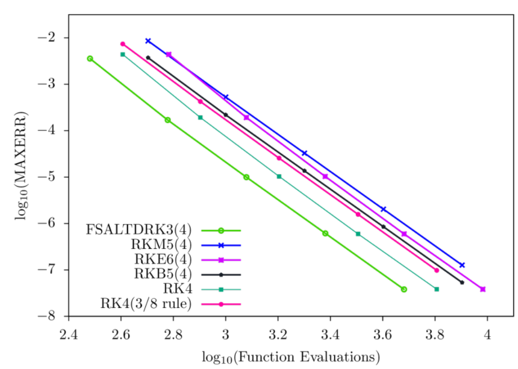

Figure 3. Efficiency graph for problem 2 with $T_{end} = 10$ and $h = i(0.00625), i = 1,2,4,8,16.$

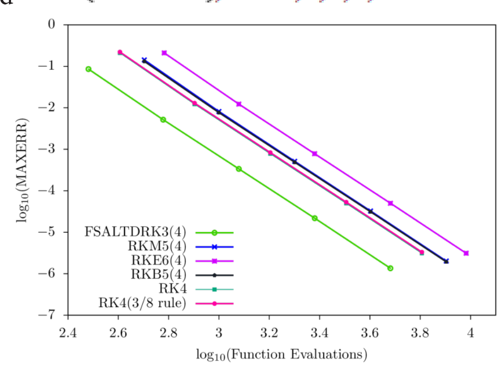

Figure 4. Efficiency graph for problem 3 with $T_{end} = 10$ and $h = i(0.000625), i = 1,2,4,8,16.$

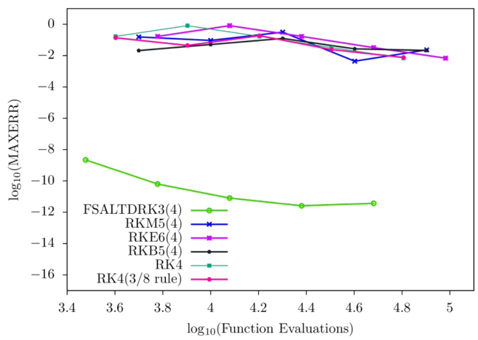

Figure 5. Efficiency graph for problem 4 with $T_{end} = 10$ and $h = i(0.00625), i = 1,2,4,8,16$.

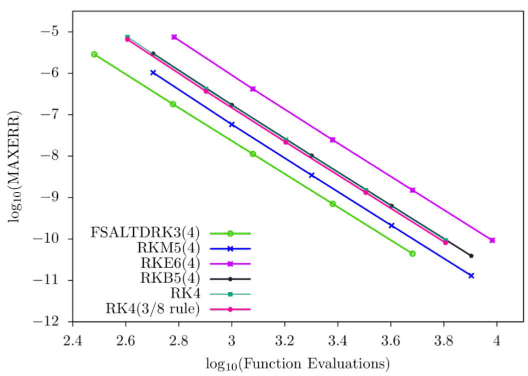

Figure 6. Efficiency graph for problem 5 with $T_{end} = 10$ and $h = i(0.00625), i = 1,2,4,8,16$.

Table 7. Numerical results for problem 5

h	Methods	T=10	Function Evaluation
0.1	FSALTDRK3(4)	2.870389 (-6)	303
	RKM5(4)	1.033965 (-6)	505
	RKE6(4)	7.470167 (-6)	606
	RKB5(4)	2.976730 (-6)	505
	RK4	7.470167 (-6)	404
	RK4(3/8 rule)	6.552390 (-6)	404
0.05	FSALTDRK3(4)	1.793384 (-7)	600
	RKM5(4)	5.814523 (-8)	1000
	RKE6(4)	4.175982 (-7)	1200
	RKB5(4)	1.718551 (-7)	1000
	RK4	4.175982 (-7)	800
	RK4(3/8 rule)	3.660042 (-7)	800
0.025	FSALTDRK3(4)	1.125747 (-8)	1200
	RKM5(4)	3.455697 (-9)	2000
	RKE6(4)	2.467759 (-8)	2400
	RKB5(4)	1.029720 (-8)	2000
	RK4	2.467759 (-8)	1600
	RK4(3/8 rule)	2.159810 (-8)	1600
0.0125	FSALTDRK3(4)	7.043062 (-10)	2403
	RKM5(4)	2.105660 (-10)	4005
	RKE6(4)	1.499930 (-9)	4806
	RKB5(4)	6.303309 (-10)	4005
	RK4	1.499930 (-9)	3204
	RK4(3/8 rule)	1.311178 (-9)	3204
0.00625	FSALTDRK3(4)	4.405021 (-11)	4803
	RKM5(4)	1.299799 (-11)	8005
	RKE6(4)	9.244484 (-11)	9606
	RKB5(4)	3.898320 (-11)	8005
	RK4	9.244479 (-11)	6404
	RK4(3/8 rule)	8.075920 (-11)	6404

6. Conclusion

In this paper, we have developed a new special explicit TDRK method with FSAL property. Based on the numerical results obtained, we can conclude that the new FSALTDRK3(4) method is more promising compared to other well-known existing explicit RK methods of the same order in term of accuracy and the number of function evaluations per step.

7. Acknowledgement

We are grateful and thankful to the Institute of Mathematical Research (INSPEM) and the Department of Mathematics, Universiti Putra Malaysia for the endless support and assistance during the research work.

8. References

1. Chan RP, Tsai AY. On explicit two-derivative Runge-Kutta methods. Numerical Algorithms. 2010 Mar 1; 53(2-3):171–94.
2. Zhang Y, Che H, Fang Y, You X. A new trigonometrically fitted two-derivative Runge-Kutta method for the numerical solution of the Schrödinger equation and related problems. Journal of Applied Mathematics. 2013 Dec 3; 2013.
3. Chen Z, Li J, Zhang R, You X. Exponentially fitted two-derivative Runge-Kutta methods for simulation of oscillatory genetic regulatory systems. Computational and Mathematical Methods in Medicine. 2015 Oct 13; 2015.

4. Yakubu DG, Kwami AM. Implicit two-derivative Runge–Kutta collocation methods for systems of initial value problems. Journal of the Nigerian Mathematical Society. 2015 Aug 31; 34(2):128–42.
5. Dormand JR, Prince PJ. A family of embedded Runge-Kutta formulae. Journal of Computational and Applied Mathematics. 1980 Mar 1; 6(1):19–26.
6. Franco JM. Exponentially fitted explicit Runge–Kutta–Nyström methods. Journal of Computational and Applied Mathematics. 2004 May 1; 167(1):1–9.
7. Fang Y, Li Q, Wu X. Extended RKN methods with FSAL property for oscillatory systems. Computer Physics Communications. 2010 Sep 30; 181(9):1538–48.
8. Fang Y, Song Y, Wu X. New embedded pairs of explicit Runge–Kutta methods with FSAL properties adapted to the numerical integration of oscillatory problems. Physics Letters A. 2008 Oct 27; 372(44):6551–9.
9. Van de Vyver H. An embedded phase-fitted modified Runge–Kutta method for the numerical integration of the radial Schrödinger equation. Physics Letters A. 2006 Apr 3; 352(4):278–85.
10. Simos TE, Vigo-Aguiar J. Exponentially fitted symplectic integrator. Physical Review E. 2003 Jan 8; 67(1):016701.
11. Senu N. Runge-Kutta-Nystrom methods for solving oscillatory problems [Doctoral dissertation]. Universiti Putra Malaysia.
12. Jawias C, Izzati N, Ismail F, Suleiman M, Jaafar A. Fourth order four-stage diagonally implicit Runge-Kutta method for linear ordinary differential equations. Malaysian Journal of Mathematical Sciences. 2010; 4(1):95–105.
13. Yaakub AR, Evans DJ. A fourth order Runge–Kutta RK (4, 4) method with error control. International Journal of Computer Mathematics. 1999 Jan 1; 71(3):383–411.
14. Lambert JD. Numerical methods for ordinary differential systems: The initial value problem. John Wiley and Sons, Inc; 2000.
15. Butcher JC. Numerical methods for ordinary differential equations. John Wiley and Sons, Inc; 2008.