An embedded 4(3) pair explicit two derivative Runge-Kutta-Nyström method for solving

\[ y''(x) = f(x, y, y') \]

T S Mohamed¹, N Senu²,³, N M A Nik Long²,³ and Z B Ibrahim²,³

¹Department of Mathematics, Faculty of Science, Misurata University, Misurata, Libya
²Institute for Mathematical Research, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
³Department of Mathematics, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

E-mail: t.salama06@gmail.com

Abstract. In this paper, we present a pair of two derivative Runge-Kutta-Nyström method for solution of second-order ordinary differential equations (ODEs) of the form \( y''(x) = f(x, y, y') \) denoted as ETDRKN method. The pair has two stages and algebraic orders four and three. The techniques used in the derivation of the method is that the higher order method is very precise and the lower order method give the best error estimate. The experimental results show the efficiency of the proposed method in comparison with the well-known embedded Runge-Kutta pairs in the scientific literature.

1. Introduction

We are dealing in this work with numerical integration of second-order ODEs of the form:

\[ y''(x) = f(x, y, y'), \quad (1) \]

with the given initial conditions \( y(x_0) = y_0, \quad y'(x_0) = y'_0, \quad x \geq x_0, \) where \( y, y' \in \mathbb{R}^N, \)

\( f : \mathbb{R} \times \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N \) is a continuous vector-valued function.

Equation (1) can be solved numerically by reducing it to system of first-order equations and then use embedded Runge-Kutta (RK) method such method can be seen in [1] and [2]. Or it can be solved directly using Runge-Kutta-Nyström (RKN) pairs as can be seen in [3], [4], [5] and [6].

The remainder of this paper is organized as follows: In section 2, we develop the new TDRKN method of fourth order. In section 3, we present the construction of the new embedded TDRKN pair of order 4(3). In section 4, we carry out the numerical result to show the efficiency of the new embedded TDRKN pair when compared with the well-known Runge-Kutta pairs from the scientific literature. Conclusions of the paper are given in section 5.
2. Developmet new TDRKN method of fourth order

In this part, we develop a new method of two derivative Runge-Kutta-Nyström methods of order 4. We suppose that the third derivative for (1) is known. An s-stage two derivative Runge-Kutta-Nyström (TDRKN) methods for (1) is defined by the formula (see Chen et al. [7]):

\[ y_{n+1} = y_n + hy'_n + h^2 \sum_{i=1}^{s} b_i k_i + h^3 \sum_{i=1}^{s} d_i k'_i \]

\[ y'_{n+1} = y'_n + h \sum_{i=1}^{s} b'_i k_i + h^2 \sum_{i=1}^{s} d'_i k'_i \]

\[ k_i = f(x_n + c_i h, y_n + c_i hy'_n + h^2 \sum_{j=1}^{s} a_{ij} k_j + h^3 \sum_{j=1}^{s} r_{ij} k'_j, y'_n + h \sum_{j=1}^{s} s_{ij} k_j + h^2 \sum_{j=1}^{s} t_{ij} k'_j) \]

\[ k'_i = g(x_n + c_i h, y_n + c_i hy'_n + h^2 \sum_{j=1}^{s} a_{ij} k_j + h^3 \sum_{j=1}^{s} r_{ij} k'_j, y'_n + h \sum_{j=1}^{s} s_{ij} k_j + h^2 \sum_{j=1}^{s} t_{ij} k'_j) \]

where \( i = 1, ..., s, y'''(x) = g(x, y, y') = f_x(x, y, y') + f_y(x, y, y')y' + f_{y'}(x, y, y')f(x, y, y') \), \( c_i, b_i, b'_i, d_i, a_{i,j}, r_{i,j}, s_{i,j}, t_{i,j} \) are real numbers.

Here, we consider \( s = 2 \). The simplifying condition for two stage TDRKN methods are: \( s_{21} = c_2, a_{21} = \frac{c_2^2}{2} \). From Chen et al.([7]), the order conditions for two stage fourth order TDRKN can be given as follows:

First order:

\[ e_1 + e_2 = 1. \]

Second order:

\[ b_1 + b_2 = \frac{1}{2}, g_1 + g_2 + e_2c_2 = \frac{1}{2}. \]

Third order:

\[ d_1 + d_2 + b_2c_2 = \frac{1}{6}, e_2c_2^2 + 2g_2c_2 = \frac{1}{3}, e_2t_{2,1} + g_2c_2 = \frac{1}{6} \]

Fourth order:

\[ b_2c_2^2 + 2d_2c_2 = \frac{1}{12}, b_2t_{21} + d_2c_2 = \frac{1}{24}, e_2c_2^2 + 3g_2c_2^2 = \frac{1}{4}, \]

\[ e_2c_2t_{21} + g_2c_2^2 = \frac{1}{8}, g_2c_2^2 = \frac{1}{12}, e_2r_{21} + g_2a_{21} = \frac{1}{24}, g_2t_{21} = \frac{1}{24} \]

By solving the system nonlinear in (3) - (7) we obtain a solution with two free parameters \( b_2 \) and \( r_{21} \).

The choices \( b_2 = \frac{1}{10}, r_{21} = \frac{41}{1000} \) leads to the method with coefficients: \( c_2 = \frac{1}{2}, b_1 = \frac{2}{5}, d_1 = d_2 = \frac{7}{120}, e_1 = 1, e_2 = 0, g_1 = \frac{1}{6}, g_2 = \frac{1}{3}, s_{21} = \frac{1}{2}, a_{21} = \frac{1}{8}, t_{21} = \frac{1}{8} \).

3. Derivation of 4(3) pair

An embedded \( p(q) \) pair of TDRKN methods based on the method \( (c, A, R, S, T, b, d, e, g) \) of order \( p \) and another TDRKN method \( (c, A, R, S, T, b', d', e', g') \) of order \( q \) \( (q < p) \) and can be written in Butcher’s tableau of coefficients as follows:
The derivation is based on formula derived in section 2 for the higher order formula. According to the values $A, R, S, T$ and $c$, we solve a lower order method in which the family of solutions of $b', d', e'$ and $g'$ is obtained. The following techniques are used for developing the efficient pair.

(i) The following quantities of $\| \tau_g^{(p+1)} \|_2$ and $\| \tau_g^{(q+1)} \|_2$ should be as small as possible for higher and lower order TDRKN formulas respectively (see Dormand et al. [8] and Hussain et al.[9]) where

$$\| \tau_g^{(p+1)} \|_2 = \sqrt{\sum_{i=1}^{n_1} \left( \tau_i^{(p+1)} \right)^2 + \sum_{i=1}^{n_2} \left( \tau_i'^{(p+1)} \right)^2}$$

and

$$\| \tau_g^{(q+1)} \|_2 = \sqrt{\sum_{i=1}^{n_1} \left( \tau_i^{(q+1)} \right)^2 + \sum_{i=1}^{n_2} \left( \tau_i'^{(q+1)} \right)^2}.$$  

where $\tau^{(p+1)}$ and $\tau'^{(p+1)}$ are the local truncation error norms for $y$ and $y'$ respectively.

(ii) A local error estimation at the point $x_{n+1}$ is determined by the formula:

$$LTE = \max\{ \| \delta_{n+1} \|_\infty, \| \delta_{n+1}' \|_\infty \}$$

where

$$\delta_{n+1} = \tilde{y}_{n+1} - y_{n+1}, \quad \delta_{n+1}' = \tilde{y}'_{n+1} - y'_{{n+1}}$$

where $y_{n+1}$, $y'_{n+1}$, $\tilde{y}_{n+1}$, and $\tilde{y}'_{n+1}$ are solutions using the higher order and the lower order formula respectively. These local error estimations, LTE can be used to control the step size $h$ by the standard formula as given in [10].

$$h_{n+1} = 0.9 h_n \left( \frac{Tol}{LTE} \right)^{\frac{1}{p+1}}$$

where 0.9 is a safety factor and represents the local error estimation at each step and $Tol$ is the accuracy required which is the maximum allowable local error. If $LTE \leq Tol$, then the step is accepted and we accepted the procedure of performing local extrapolation (or higher order mode) meaning that the more accurate approximation will be used to advance the integration and $h$ will be updated using formula (10). If $LTE > Tol$ then the step is failed and the step-size, $h$ could be halved. Now consider the TDRKN method in section 2 and based on $A$ and $c$ values, we can derive second method with fourth and third order. By solving the order conditions in section 2 up to order three, we obtain the following solution: $b'_1 = \frac{1}{6} + 2d'_1 + 2d'_2$, $b'_2 = \frac{1}{3} - 2d'_1 - 2d'_2$, $e'_1 = -\frac{1}{3} + 4g'_2$, $e'_2 = \frac{4}{3} - 4g'_2$, $g'_1 = -\frac{1}{3} + g'_2$. Let $d'_1 = \frac{7}{120}$ and $d'_2 = \frac{7}{120}$. By using Minimize command in Maple Software, we get $g'_2 = \frac{1}{10}$ and the minimum local truncation error is $165 \times 10^{-2}$. We denote this pair as ETDRKN4(3) method and it can be written in Butcher tableau as follows:

| \hline
| $c$ & $A$ & $R$ & $S$ & $T$ |
| \hline
| $b^T$ & $d^T$ & $e^T$ & $g^T$ |
| \hline
| $b'^T$ & $d'^T$ & $e'^T$ & $g'^T$ |
| \hline
Table 2. Butcher tableau for ETDRKN4(3) method

| 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|
| 1/2 | 1/2 | 1/8 | 1/1000 | 0 | 1/2 |
| 1/8 | 1/8 | 1/10 | 7/120 | 7/120 | 1 | 0 |
| 1/8 | 1/10 | 7/120 | 7/120 | 53/3 | −50/3 | 13/3 | 9/2 |

4. Problems tested and numerical results
The numerical results that are given in the tables below for solving problems (1)-(3). In addition, the following abbreviations will be used in Tables (1)-(3)

- **TOL:** Tolerance.
- **METHOD:** Method employed.
- **STEP:** Number of steps.
- **F.N:** Number of function evaluations.
- **FSTEP:** Failure steps.
- **MAXE:** Maximum error of the computed solution.
- **TIME:** Execution time taken in microseconds.
- **ETDRKN4(3):** The new 4(3) pair of embedded TDRKN derived in this paper.
- **ERK4(3)Z:** Runge-Kutta 4(3) pair given in [11].
- **ERK4(3)D:** Runge-Kutta 4(3) pair derived by [8].

**Problem1:**
\[
y'' = -49 y - 14 y' + x, \quad y(0) = \frac{2}{343}, \quad y'(0) = \frac{3}{49}, \quad x_{end} = 10
\]
whose analytic solution is \( y(x) = -\frac{2}{343} + \frac{1}{39}x + \frac{4}{343}e^{-7x} \).

**Problem2:**
\[
y'' = \frac{1}{40} (10 - y) y', \quad y(0) = 1, \quad y'(0) = \frac{19}{80}, \quad x_{end} = 20.
\]
whose analytic solution is \( y(x) = \frac{20}{1 + 19e^{-\frac{x}{4}}} \).

**Problem3:**
\[
y''_1 = \frac{1}{4} e^{3x} y'^2_2, \quad y_1(0) = 1, \quad y'_1(0) = -1
\]
\[
y''_2 = 4y'^2_1, \quad y_2(0) = 1, \quad y'_2(0) = -2, \quad x_{end} = 10
\]
whose analytic solution is \( y_1(x) = e^{-x}, \quad y_2(x) = e^{-2x} \).
### Table 3. Comparison results between ETDRKN4(3), ERK4(3)Z and ERK4(3)D methods for Problem 1

| TOL  | METHOD       | STEP | F.N | FSTEP | MAXE            | TIME(s) |
|------|--------------|------|-----|-------|-----------------|---------|
| 10^-2| ETDRKN4(3)   | 54   | 218 | 1     | 1.577695(5)     | 0.044   |
|      | ERK4(3)Z     | 29   | 488 | 14    | 1.936580(4)     | 0.045   |
|      | ERK4(3)D     | 32   | 326 | 5     | 6.590269(4)     | 0.048   |
| 10^-4| ETDRKN4(3)   | 73   | 294 | 1     | 1.594003(7)     | 0.059   |
|      | ERK4(3)Z     | 37   | 584 | 14    | 2.181070(6)     | 0.061   |
|      | ERK4(3)D     | 30   | 410 | 5     | 4.201192(6)     | 0.065   |
| 10^-6| ETDRKN4(3)   | 135  | 542 | 1     | 1.592098(9)     | 0.065   |
|      | ERK4(3)Z     | 61   | 752 | 2     | 2.812832(8)     | 0.088   |
|      | ERK4(3)D     | 51   | 662 | 5     | 5.435729(8)     | 0.078   |
| 10^-8| ETDRKN4(3)   | 337  | 1350| 1     | 1.590922(11)    | 0.085   |
|      | ERK4(3)Z     | 149  | 1978| 19    | 2.075103(10)    | 0.111   |
|      | ERK4(3)D     | 117  | 1454| 5     | 3.767143(10)    | 0.099   |
| 10^-10| ETDRKN4(3)  | 976  | 3906| 1     | 1.594419(13)    | 0.097   |
|      | ERK4(3)Z    | 422  | 5074| 3     | 2.908340(12)    | 0.116   |
|      | ERK4(3)D    | 321  | 3902| 5     | 5.540873(12)    | 0.117   |

### Table 4. Comparison results between ETDRKN4(3), ERK4(3)Z and ERK4(3)D methods for Problem 2

| TOL  | METHOD       | STEP | F.N | FSTEP | MAXE            | TIME(s) |
|------|--------------|------|-----|-------|-----------------|---------|
| 10^-2| ETDRKN4(3)   | 13   | 58  | 3     | 3.425294(4)     | 0.047   |
|      | ERK4(3)Z     | 12   | 164 | 2     | 7.271001(3)     | 0.048   |
|      | ERK4(3)D     | 8    | 116 | 2     | 7.207305(4)     | 0.052   |
| 10^-4| ETDRKN4(3)   | 37   | 152 | 2     | 5.550431(6)     | 0.050   |
|      | ERK4(3)Z     | 33   | 406 | 1     | 3.894414(5)     | 0.067   |
|      | ERK4(3)D     | 22   | 294 | 3     | 1.957709(5)     | 0.067   |
| 10^-6| ETDRKN4(3)   | 114  | 462 | 3     | 2.632109(8)     | 0.056   |
|      | ERK4(3)Z     | 100  | 1210| 1     | 4.565549(7)     | 0.066   |
|      | ERK4(3)D     | 65   | 800 | 2     | 2.016162(7)     | 0.069   |
| 10^-8| ETDRKN4(3)   | 356  | 1426| 1     | 1.037584(9)     | 0.064   |
|      | ERK4(3)Z     | 315  | 3790| 1     | 4.807418(9)     | 0.082   |
|      | ERK4(3)D     | 202  | 2444| 2     | 2.411650(9)     | 0.092   |
| 10^-10| ETDRKN4(3) | 1124 | 4500| 2     | 9.791279(12)    | 0.090   |
|      | ERK4(3)Z    | 992  | 11914| 1     | 4.899547(11)    | 0.113   |
|      | ERK4(3)D    | 633  | 7606| 1     | 2.951595(11)    | 0.135   |
Problem 1: Efficiency

![Graph of Problem 1 Efficiency](image1)

Figure 1. The efficiency curve for problem 1

Problem 2: Efficiency

![Graph of Problem 2 Efficiency](image2)

Figure 2. The efficiency curve for problem 2

Problem 3: Efficiency

![Graph of Problem 3 Efficiency](image3)

Figure 3. The efficiency curve for all methods Problem 3
### Table 5. Comparison results between ETDRKN4(3), ERK4(3)Z and ERK4(3)D methods for Problem 3

| TOL  | METHOD   | STEP | F.N | FSTEP | MAXE         | TIME(s) |
|------|----------|------|-----|-------|--------------|---------|
| $10^{-2}$ | ETDRKN4(3) | 7    | 30  | 1     | 2.973165(-4) | 0.044   |
|       | ERK4(3)Z | 4    | 48  | 0     | 3.975717(-3) | 0.046   |
|       | ERK4(3)D | 4    | 48  | 0     | 4.154701(-4) | 0.047   |
| $10^{-4}$ | ETDRKN4(3) | 20   | 82  | 1     | 3.619274(-6) | 0.056   |
|       | ERK4(3)Z | 10   | 120 | 0     | 7.892022(-5) | 0.060   |
|       | ERK4(3)D | 10   | 120 | 0     | 4.779309(-6) | 0.058   |
| $10^{-6}$ | ETDRKN4(3) | 59   | 238 | 1     | 3.475449(-8) | 0.070   |
|       | ERK4(3)Z | 35   | 420 | 0     | 5.356079(-7) | 0.078   |
|       | ERK4(3)D | 30   | 360 | 0     | 5.048267(-8) | 0.071   |
| $10^{-8}$ | ETDRKN4(3) | 186  | 746 | 1     | 3.669262(-10)| 0.073   |
|       | ERK4(3)Z | 111  | 1332| 0     | 4.934041(-9) | 0.073   |
|       | ERK4(3)D | 91   | 1092| 0     | 5.141118(-10)| 0.075   |
| $10^{-10}$ | ETDRKN4(3) | 584  | 2338| 1     | 3.664236(-12)| 0.091   |
|       | ERK4(3)Z | 353  | 4236| 0     | 4.801121(-11)| 0.111   |
|       | ERK4(3)D | 284  | 3408| 0     | 5.175554(-12)| 0.101   |

### 5. Discussion and Conclusion

From the numerical results, we noticed that for all the problems ETDRKN4(3) gives better results in terms of function evaluations and total time taken to solve the problems compared to ERK4(3)Z and ERK4(3)D. In terms of absolute error method ETDRKN4(3) produced the smallest error compared to ERK4(3)Z and ERK4(3)D.

### References

[1] Dormand J R and Prince P J 1980 A family of embedded Runge-Kutta formulae *Journal of Computational and Applied Mathematics* **6**(1) 19–26

[2] Verner J H 1978 Explicit Runge-Kutta methods with estimates of the local truncation error *SIAM Journal on Numerical Analysis* **15**(4) 772–790

[3] Fine J M 1987 Low order practical Runge-Kutta-Nyström methods *Computing* **38**(4) 281–297

[4] Fine J 1985 Low order Runge-Kutta-Nyström methods with interpolants *Department of Computer Science Technical report No. 183/85, University of Toronto*

[5] Ismail F, Al-Khasawneh R A and Suleiman M 2007 Embedded singly diagonally implicit Runge-Kutta-Nyström general method (3, 4) in (4, 5) for solving second order ivps *International Journal of Applied Mathematics* **37**:2

[6] Fehlberg E 1974 Classical seventh-sixth- and fifth-order Runge-Kutta-Nyström formulas with stepsize control for general second-order differential equations *NASA Technical Report R-432*

[7] Chen Z, Qiu Z, Li J and You X 2015 Two-derivative Runge-Kutta-Nyström methods for second-order ordinary differential equations *Numerical Algorithms* **70**(4) 897–927

[8] Dormand J R 1996 *Numerical methods for differential equations: a computational approach* (CRC Press)

[9] Hussain K, Ismail F and Senu N 2015 Two embedded pairs of Runge-Kutta type methods for direct solution of special fourth-order ordinary differential equations *Mathematical Problems in Engineering* Article ID 196595

[10] Senu N 2010 *Runge-Kutta-Nyström methods for solving oscillatory problems* (Ph.D. thesis, Department of Mathematics, Faculty of Science, 43400 UPM Serdang, Malaysia)

[11] Hairer E, Norsett S P and Wanner G 2008 *Solving ordinary differential equations I. Nonstiff problems* Berlin: Springer-Verlag