Embedded Explicit Two-Step Runge-Kutta-Nyström for Solving Special Second-Order IVPs

L. M. Ariffin

Department of Mathematics and Statistics,
Faculty of Science, Technology and Human Development
Universiti Tun Hussein Onn Malaysia
86400 Batu Pahat, Johor, Malaysia

N. Senu, M. Suleiman and Z.A. Majid

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

Copyright © 2014 L. M. Ariffin, N. Senu, M. Suleiman and Z.A. Majid. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

A third-order three-stage explicit Two-step Runge-Kutta-Nyström (TSRKN) method embedded into fourth-order three-stage TSRKN method is developed to solve special second-order initial value problems (IVPs) directly. The stability of the method is investigated. Numerical results are obtained by solving a standard set of test problems, which then reduced to first-order system when solved using Runge-Kutta (RK) method and comparison are made with existing RK method with same order using variable step-size. The results clearly showed the advantage and the efficiency of the new method.

Mathematics Subject Classification: 65L05, 65L20

Keywords: Two-step Runge-Kutta-Nyström methods, Second Order ODEs, Absolute Stability
1 Introduction

Special second-order ordinary differential equations (ODE’s) is given by

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

(1)

with the initial conditions

\[ y(x_0) = y_0, \quad y'(x_0) = y'_0 \]

where \( y(x) f(x, y) \in \mathbb{R}^n \).

Equation (1) can be solved when using standard Runge-Kutta (RK) method provided that it has to be reduce to an equivalent first-order system of twice the dimension. However, Sharp and Fine [1], Dormand et al. [2] and El-Mikkawy and El-Desouky [3] shows that they manage to solve equation (1) directly and efficiently by using Runge-Kutta-Nyström (RKN) method. In general, RK and RKN codes involving the embedded pairs of order \( q(p) \) is efficient when the method of order \( q = p + 1 \) is used to obtain the numerical solutions of the problem where as the method of order \( p \) is used to obtain the local truncation error. Unlike two-step RK method used in Jackiewicz and Verner [4], RK method for the numerical solution of (1) requires many evaluations of the function \( f \) per step and hence is not as efficient as linear multistep methods, when the derivative evaluations are relatively expensive.

According to Senu et al. [5], when solving (1) numerically, the algebraic order of the method is essential where it is the main criterion to achieve high accuracy and to have lower stage of RKN method with maximal order in order to reduce the computational cost. Thus, in this paper we derive an embedded pair which is explicit and two-step in nature.

We consider the TSRKN method for the initial value problem (1) by Paternoster [6] which was derived as an indirect method from the two-step RK method presented by Jackiewicz et al. [7], as follows

\[
y_{i+2} = (1 - \theta) y_{i+1} + \theta y_i + h \sum_{j=1}^{m} v_j y'_i + h \sum_{j=1}^{m} w_j y'_{i+1} + h^2 \sum_{j=1}^{m} \bar{v}_j f \left( x_i + c_j h, Y^j_i \right) + \bar{w}_j f \left( x_{i+1} + c_j h, Y^j_{i+1} \right)
\]

(2)

\[
y'_{i+2} = (1 - \theta) y'_{i+1} + \theta y'_i + h \sum_{j=1}^{m} v_j f \left( x_i + c_j h, Y^j_i \right) + w_j f \left( x_{i+1} + c_j h, Y^j_{i+1} \right)
\]

(3)

where

\[
Y^j_{i+1} = y_{i+1} + h c_j y'_{i+1} + h^2 \sum_{s=1}^{m} a_{js} f \left( x_{i+1} + c_s h, Y^s_{i+1} \right), \quad j = 1, \ldots, m
\]
\begin{equation}
Y_j^i = y_i + hc_j y'_i + h^2 \sum_{s=1}^{m} a_{js} f \left( x_i + c_s h, Y_s^i, j = 1, \ldots m \right) \tag{4}
\end{equation}

where \( \theta, v_j, w_j, \bar{v}_j, \bar{w}_j, a_{js} \) for \( j, s = 1, \ldots m \) are the coefficients of the methods with \( m \) is the number of stages for the method.

Alternatively TSRKN (2) and (3) can be written as

\begin{equation}
y_{i+2} = (1 - \theta) y_{i+1} + \theta y_i + h \sum_{j=1}^{m} v_j y'_i + h \sum_{j=1}^{m} w_j y_{i+1}' + h^2 \sum_{j=1}^{m} \bar{v}_j k^j_i + \bar{w}_j k^j_{i+1} \tag{5}
\end{equation}

\begin{equation}
y'_{i+2} = (1 - \theta) y'_{i+1} + \theta y'_i + h \sum_{j=1}^{m} v_j k^j_i + w_j k^j_{i+1} \tag{6}
\end{equation}

where

\begin{align*}
k^j_i &= f \left( x_i + c_j h, y_i + hc_j y'_i + h^2 \sum_{s=1}^{m} a_{js} k^s_i \right), \quad j = 1, \ldots m \\
k^j_{i+1} &= f \left( x_{i+1} + c_j h, y_{i+1} + hc_j y'_{i+1} + h^2 \sum_{s=1}^{m} a_{js} k^s_{i+1} \right), \quad j = 1, \ldots m. \tag{7}
\end{align*}

Essentially, the methods derived should have the properties of zero stable and consistent. According to Paternoster [6], TSRKN is zero stable if \(-1 < \theta \leq 1\) and it is consistent if \( \sum_{j=1}^{m} (v_j + w_j) = 1 + \theta \). Fulfillment of these two properties implies that the method is convergent (see Watt [8] and Jackiewicz et al. [7]). Thus, we apply both properties into the derivation of our new method with \( \theta = 0 \) that will guarantee zero stability of our method as proposed by Jackiewicz and Verner [4].

An embedded \( q(p) \) pair of TSRKN methods is based on the method \((c, A, v, w, \bar{v}, \bar{w})\) of order \( q \) and the other TSRKN method \((c, A, \hat{v}, \hat{w}, \hat{\bar{v}}, \hat{\bar{w}})\) of order \( p(p < q) \) which is similar with the embedded pair for one-step RKN derived by Senu et al. [5]. It can also be presented by the following Butcher array:

where \( c = [c_1, c_2, \ldots, c_m]^T, A = [a_{ij}], \theta = 0, v = [v_1, v_2, \ldots, v_m]^T, w = [w_1, w_2, \ldots, w_m]^T, \bar{v} = [\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_m]^T, \bar{w} = [\bar{w}_1, \bar{w}_2, \ldots, \bar{w}_m]^T, \hat{v} = [\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_m]^T, \hat{w} = [\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_m]^T \). The main motivation for the embedded pair of TSRKN is to obtain cheap local error estimation which is to be used in a variable step-size algorithm.
Table 1: Embedded m-stage 2-step explicit RKN method

<table>
<thead>
<tr>
<th>c</th>
<th>A</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(c_2) 0</td>
<td></td>
</tr>
<tr>
<td>(v)</td>
<td>(a_{2,1}) 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(w)</td>
<td>(\vdots) (\vdots) (\vdots) (\vdots)</td>
<td>(\theta)</td>
<td>(v_1) (v_2) (\cdots) (v_m)</td>
</tr>
<tr>
<td>(\bar{v})</td>
<td>(w_1) (w_2) (\cdots) (w_m)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\bar{w})</td>
<td>(\hat{v}_1) (\hat{v}_2) (\cdots) (\hat{v}_m)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\hat{w})</td>
<td>(\hat{w}_1) (\hat{w}_2) (\cdots) (\hat{w}_m)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \hat{v}_j = \hat{y}_j + h \left( -\lambda^2 Y^j_{i+1} \right) \]

2 Stability of the Method

In this section the linear stability of the TSRKN method will be discussed. By substituting

\[ x_{i+1} = x_i + h, \quad y_{i+1} = y_i + hy_i' + \frac{h^2}{2} y_i'' \]

and apply the test equation \( y'' = f(x, y) = -\lambda^2 y \) into TSRKN method (2) and (3), we obtain

\[ y_{i+2} = y_i + hy_i' + \frac{h^2}{2} (-\lambda^2 y_i) + h \sum_{j=1}^{m} v_j y_i' + h \sum_{j=1}^{m} w_j \left( y_i' + h (-\lambda^2 y_i) \right) + h^2 \sum_{j=1}^{m} \bar{v}_j \left( -\lambda^2 Y^j_{i+1} \right) + \bar{w}_j \left( -\lambda^2 \bar{Y}^j_{i+1} \right) \]  

(8)

\[ y_{i+2}' = y_i' + h (-\lambda^2 y_i) + h \sum_{j=1}^{m} v_j \left( -\lambda^2 Y^j_{i+1} \right) + w_j \left( -\lambda^2 \bar{Y}^j_{i+1} \right) \]  

(9)

where

\[ Y^j_{i+1} = y_i + hy_i' + \frac{h^2}{2} (-\lambda^2 y_i) + h c_j \left( y_i' + h (-\lambda^2 y_i) \right) + h^2 \sum_{s=1}^{m} a_{js} (-\lambda^2 Y^s_{i+1}) \]  

(10)

\[ Y^j_i = y_i + hy_i' c_j + h^2 \sum_{s=1}^{m} a_{js} (-\lambda^2 Y^s_i) . \]  

(11)
Multiply equation (9) by \( h \) gives
\[
hy'_{i+2} = hy'_i + h^2 \left( -\lambda^2 y_i \right) + h^2 \sum_{j=1}^{m} v_j \left( -\lambda^2 Y_{ij} \right) + w_j \left( -\lambda^2 Y_{i+1,j} \right). \tag{12}
\]
The application of the test equation to equation (10) and (11) yields the recursion \( Y_i \) and \( Y_{i+1} \) respectively.

\[
Y_i = N^{-1} (y_i e + hy'_i c) \quad \text{and} \quad Y_{i+1} = N^{-1} \left( y_i (e + e\frac{h}{2} + cH) + hy'_i (e + c) \right).
\]

Elimination of the auxiliary vectors \( Y_i \) and \( Y_{i+1} \) in equations (8) and (9) yields
\[
y_{i+2} = y_i \left( 1 + \frac{H}{2} + H w e + H v N^{-1} e + \bar{w} N^{-1} e + \frac{H}{2} \bar{w} N^{-1} c + H \bar{w} N^{-1} c \right) +
hy'_i \left( 1 + v e + w e + H v N^{-1} c + \bar{w} N^{-1} e + \bar{w} N^{-1} c \right) \tag{13}
\]
\[
hy'_{i+2} = y_{i+1} \left( H + H v N^{-1} e + H w N^{-1} e + \frac{H^2}{2} \bar{w} N^{-1} e + H^2 \bar{w} N^{-1} c \right) +
hy'_i \left( 1 + H v N^{-1} c + H w N^{-1} e + H \bar{w} N^{-1} c \right) \tag{14}
\]
with \( v = (v_1, v_2, \ldots, v_m) \), \( w = (w_1, w_2, \ldots, w_m) \), \( \bar{v} = (\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_m) \) and \( \bar{w} = (\bar{w}_1, \bar{w}_2, \ldots, \bar{w}_m) \).

The resulting recursion is
\[
Z_{i+2} = D(H) Z_i \quad \text{with} \quad Z_{i+2} = (y_{i+2}, hy'_{i+2})^\top \quad \text{and} \quad Z_i = (y_i, hy'_i)^\top
\]
\[
D(H) = \begin{pmatrix} A(H) & B(H) \\ A'(H) & B'(H) \end{pmatrix}
\]
with
\[
A(H) = 1 + \frac{H}{2} + H w e + H v N^{-1} e + \bar{w} N^{-1} e + \frac{H}{2} \bar{w} N^{-1} c + H \bar{w} N^{-1} c
\]
\[
B(H) = 1 + ve + we + H v N^{-1} c + \bar{w} N^{-1} e + \bar{w} N^{-1} c
\]
\[
A'(H) = H + H v N^{-1} e + H w N^{-1} e + \frac{H^2}{2} \bar{w} N^{-1} e + H^2 \bar{w} N^{-1} c
\]
\[
B'(H) = 1 + H v N^{-1} c + H w N^{-1} e + H \bar{w} N^{-1} c
\]
where \( N = I - HA, H = - (\lambda h)^2 \), \( c = (c_1, \ldots, c_s)^\top \), \( e = (1, \ldots, 1)^\top \) with \( D(H) \) is the stability matrix for the TSRKN methods above. The characteristic equation of \( D \) can be written as
\[
\xi^2 - \text{trace}(D)\xi + \text{det}(D) = 0 \tag{15}
\]
which is the stability polynomial of the TSRKN method.

**Definition 2.1 (Absolute stability interval)** An interval \((-H_a, 0)\) is called the interval of absolute stability of the method (2) and (3) if for all \( H \in (-H_a, 0) \), \( \xi_{1,2} < 1 \).
3 Construction of the Method

The ETSRKN4(3) parameters must satisfy the following algebraic conditions as given in Ariffin et al. [9]. These order conditions was obtained in a more elementary way using the Taylor series expansion as proposed by Williamson [10]. Different approach was done by Jackiewicz et al. [7] where they used the theory of Hairer and Wanner [11] in order to derive the TSRK order conditions up to fourth-order. Here are the order conditions for TSRKN methods:

Order conditions for $y$

order 1 :
\[ \sum v_i + w_i = 1 \]  
(16)

order 2 :
\[ \sum w_i + \tilde{v}_i + \tilde{w}_i = \frac{3}{2} \]  
(17)

order 3 :
\[ \sum \bar{w}_i + c_i \bar{v}_i = \frac{4}{3}, \quad \sum \bar{w}_i + c_i (\bar{v}_i + \bar{w}_i) = \frac{4}{3} \]  
(18)

order 4 :
\[ \frac{1}{2} \sum \bar{w}_i + c_i^2 \bar{v}_i = \frac{2}{3}, \quad \frac{1}{2} \sum \bar{w}_i + 2a_{ij} \bar{v}_i = \frac{2}{3}, \]  
(19)
\[ \frac{1}{2} \sum \bar{w}_i + c_i^2 (\bar{v}_i + \bar{w}_i) + 2c_i \bar{w}_i = \frac{2}{3}, \quad \sum \bar{w}_i + c_i^2 \bar{v}_i + c_i \bar{w}_i = \frac{4}{3}. \]  
(20)

Order conditions for $y'$

order 1 :
\[ \sum v_i + w_i = 1 \]  
(21)

order 2 :
\[ \sum w_i + c_i (v_i + w_i) = 2, \quad \sum w_i + c_i v_i = 2 \]  
(22)

order 3 :
\[ \frac{1}{2} \sum w_i + c_i^2 v_i = \frac{4}{3}, \quad \frac{1}{2} \sum w_i + 2a_{ij} v_i = \frac{4}{3}, \]  
(23)
\[ \frac{1}{2} \sum w_i + c_i^2 (v_i + w_i) + 2c_i w_i = \frac{4}{3}, \quad \sum w_i + c_i^2 v_i + c_i w_i = \frac{8}{3} \]  
(24)

order 4 :
\[ \frac{1}{2} \sum w_i + 2a_{ij} c_i v_i = 2, \quad \frac{1}{6} \sum w_i + c_i^3 v_i = \frac{2}{3}, \]  
(25)
\[ \sum a_{ij} c_i w_i = 0, \quad \frac{1}{6} \sum w_i + c_i^3 (v_i + w_i) + 3c_i w_i + 3c_i^2 w_i = \frac{2}{3}, \]  
(26)
\[ \frac{1}{2} \sum w_i + c_i w_i + c_i^2 w_i + c_i^3 v_i = 2, \quad \frac{1}{2} \sum w_i + c_i w_i + 2a_{ij}c_i v_i = 2, \quad (27) \]

\[ \frac{1}{2} \sum w_i + c_i w_i + c_i^2 v_i = 2, \quad \sum a_{ij}c_j v_i = \frac{2}{3} \quad (28) \]

Paternoster [12] gives the following definition:

**Definition 3.1** An m-stage TSRKN method is said to satisfy the following simplifying conditions if its parameters satisfy

\[ \sum_{j=1}^{m} a_{ij}c_j^{k-2} = \frac{c_i^k}{k(k-1)}, \quad i = 1, \ldots, m, \quad k = 1, \ldots, q. \quad (29) \]

Equation (29) allow the reduction of order conditions in the theory of TSRKN methods.

For the fourth-order method, solving simultaneously the equations (16)–(28) together with simplifying assumption (29). It involved 20 equations with 17 parameters and by setting \( c_2, c_3 \) and \( \bar{v}_3 \) as free parameters, the following solution of three-parameter family is obtained:

\[ a_{21} = \frac{1}{2}c_2^2, \quad a_{31} = \frac{2c_2}{c_2-2}, \quad (28) \]

\[ a_{32} = \frac{c_3(c_2-2)-3}{c_2}, \quad v_1 = \frac{2}{3}c_3, \quad (29) \]

\[ v_2 = \frac{2(c_2-2)}{3c_3(c_2-2)}, \quad (29) \]

\[ w_1 = \frac{2(-4c_3+3c_2+6-4c_2)}{3(-1+c_2)(-1+c_3)}, \quad (29) \]

\[ \bar{v}_1 = \frac{-[6\bar{v}_3c_3c_2-6\bar{v}_3c_3^2+12\bar{v}_3c_3c_2-6\bar{v}_3c_3^2]c_2-6\bar{v}_3c_3c_2+12\bar{v}_3c_3c_2-15c_3c_2-6\bar{v}_3c_3+23c_2+6\bar{v}_3c_3-6\bar{v}_3c_2-15c_2]}{[6c_2(-1+c_2)\bar{v}_3c_3(1+c_2)]}, \quad (29) \]

\[ \bar{v}_2 = \frac{-4+4c_2-3\bar{v}_3c_3c_2+3\bar{v}_3c_3^2}{3(-1+c_2)}, \bar{w}_1 = 0, \bar{w}_2 = 0, \bar{w}_3 = 0. \quad (29) \]

According to Dormand [2], the strategy to choose the free parameter is by minimizing the truncation error coefficients which are defined by

\[ \| \tau^{(p+1)} \|_2 = \sqrt{\sum_{j=1}^{n_{p+1}} \left( \tau_j^{(p+1)} \right)^2} \quad \text{and} \quad \| \tau^{(p+1)} \|_2 = \sqrt{\sum_{j=1}^{n_{p+1}} \left( \tau_j^{(p+1)} \right)^2} \quad \text{for} \ y_n \quad \text{and} \ y_n'. \quad (29) \]

Thus, we use this approach to obtain the value of all free parameters obtained from the above solution.
Letting \( c_2 = \frac{5}{6} \) gives \( ||\tau'(5)||_2 \) in term of \( c_3 \). By using the minimize command in MAPLE, \( ||\tau'(5)||_2 \) has a minimum value zero at \( c_3 = \frac{3950}{2473} \). Since TSRKN is a two-step method, we may consider for the case \( c_3 \in [0, 2] \). Obtaining the value of \( c_3 \) gives \( ||\tau'(5)||_2 \) in terms of \( \bar{v}_3 \). Again, by using the minimize command \( ||\tau'(5)||_2 \) has a minimum value zero at \( \bar{v}_3 = \frac{145797760102019788 \times 10^{-2}}{37292647323125} \). The above obtained value of the free parameters gives \( ||\tau'(5)||_2 = 2.925926110901890175 \times 10^{-2} \) and \( ||\tau'(5)||_2 = 4.3167507760102019788 \times 10^{-2} \). The stability interval of the fourth-order formula is approximately \((-0.1805, 0)\).

For the third-order method, using the same values for \( a_{ij} \) and \( c_i \) obtained in the fourth-order method, solving the first four equations for \( y_n \) and the first seven equations for \( y'_n \) simultaneously, the following solution of four-parameter is obtained:

\[
\bar{v}_1 = -\frac{1}{5} + \frac{3348359}{6115729} \bar{v}_3, \quad \bar{v}_2 = \frac{-24 + 42005880}{6115729} \bar{v}_3,
\]
\[
\hat{\bar{v}}_1 = 6 - \frac{44773250}{6115729} \bar{v}_3, \quad \bar{w}_2 = 0, \quad \bar{w}_3 = 0,
\]
\[
\hat{\bar{w}}_2 = \frac{4740}{2267} \hat{\bar{w}}_1 + \frac{144570}{2267} \bar{v}_3 + \frac{23700}{2267} \hat{\bar{v}}_1 - \frac{23700}{2267} \bar{v}_3 - \frac{468075000}{6115729} \bar{v}_3,
\]
\[
\hat{\bar{w}}_3 = \frac{2473}{2267} \hat{\bar{w}}_1 - \frac{150853}{4534} \hat{\bar{w}}_1 + \frac{12365}{2267} \hat{\bar{v}}_1 + 5 \bar{v}_3 + \frac{98750}{2473} \bar{v}_3.
\]

Using the minimize error approach; we obtain the value for all parameters. By minimizing \( ||\tau'(4)||_2 \), we obtain the minimum value zero at \( \hat{\bar{v}}_3 = 1.0672840388671126404 \). However, we choose \( \hat{\bar{v}}_3 = \frac{103}{100} \) and it gives \( ||\tau'(4)||_2 = 3.6354254266897047443 \times 10^{-2} \). Letting \( \hat{\bar{v}}_1 = \frac{3}{2} \) and \( \bar{w}_1 = -\frac{38221192}{25000000} \), we obtain the value of \( \hat{\bar{v}}_3 \) with \( ||\tau'(4)||_2 = 1.8965491105519564054 \times 10^{-2} \). The coefficients in Table 2 are generated using MAPLE where the significant digits is set to 20 by the command Digits.

## 4 Problems Tested

Below are some of the problems tested:

**Problem 1 (Homogeneous)**

\[
\frac{d^2 y(x)}{dx^2} = -64 y(x), \quad y(0) = 1, \quad y'(0) = -2.
\]

Exact solution: \( y(x) = -\frac{1}{4} \sin(8x) + \cos(8x) \).

Source: van der Houwen and Sommeijer [13]
Table 2: Coefficients for ETSRKN4(3) method represented by Butcher array

<table>
<thead>
<tr>
<th>0</th>
<th>5/6</th>
<th>25/72</th>
</tr>
</thead>
<tbody>
<tr>
<td>3950/2473</td>
<td>795590450/15124197817</td>
<td>11336586900/15124197817</td>
</tr>
<tr>
<td>-23236/29025</td>
<td>143424/56675</td>
<td>15124197817/14170733625</td>
</tr>
<tr>
<td>-1148/633</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>174326794511/98701316250</td>
<td>1201521888/447449125</td>
<td>14579726695588/37293647323125</td>
</tr>
<tr>
<td>-7640452/4997535</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-467195557/611572900</td>
<td>69552786/30578645</td>
<td>103/100</td>
</tr>
<tr>
<td>-18844147/12291458</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3/7</td>
<td>4422254760/1652229617</td>
<td>2601113613/6505683564</td>
</tr>
<tr>
<td>38221103/25000000</td>
<td>341705759/23736878661</td>
<td>100899825/1343328072</td>
</tr>
</tbody>
</table>
The first order system: The new variable are $y_1 = y$ and $y_2 = y'$.

$$y_1' = y_2, \quad y_2' = -64y_1, \quad y_1(0) = 1, \quad y_2(0) = -2, \quad 0 \leq x \leq 50.$$ 

Exact solutions are $y_1(x) = -1/4 \sin(8x) + \cos(8x), y_2(x) = -2 \cos(8x) - 8 \sin(8x)$.

**Problem 2** (Inhomogeneous)

$$\frac{d^2y(x)}{dx^2} = -v^2y(x) + (v^2 - 1) \sin(x), \quad y(0) = 1, \quad y'(0) = v + 1, \quad x \geq 0,$$

where $v \gg 1$, $0 \leq x \leq 50$.

Exact solution is $y(x) = \cos(vx) + \sin(vx) + \sin(x)$. Numerical result is for the case $v = 10$.

Source: van der Houwen and Sommeijer [13]

The first order system:

$$y_1' = y_2, \quad y_2' = -100y_1 + 99 \sin(x), \quad y_1(0) = 1, \quad y_2(0) = 11, \quad 0 \leq x \leq 50.$$ 

Exact solutions are $y_1(x) = \cos(10x) + \sin(10x) + \sin(x), y_2(x) = -10 \sin(10x) + 10 \cos(10x) + \cos(x)$.

**Problem 3**

$$\frac{d^2y_1(x)}{dx^2} + y_1(x) = 0.001 \cos(x), \quad y_1(0) = 1, \quad y_1'(0) = 0 \quad 0 \leq x \leq 1000$$

$$\frac{d^2y_2(x)}{dx^2} + y_2(x) = 0.001 \sin(x), \quad y_2(0) = 0, \quad y_2'(0) = 0.9995, \quad 0 \leq x \leq 1000$$

Exact solutions are $y_1(x) = \cos(x) + 0.0005x \sin(x), y_2(x) = \sin(x) - 0.0005x \cos(x)$.

Source: Stiefel and Bettis [14].

The first order system:

$$y_1' = y_2, \quad y_2' = -y_1 + 0.001 \cos(x), \quad y_1(0) = 1, \quad y_2(0) = 0,$$

$$y_3' = y_4, \quad y_4' = -y_3 + 0.001 \sin(x), \quad y_3(0) = 0, \quad y_4(0) = 0.9995, \quad 0 \leq x \leq 1000.$$

Exact solutions are

$$y_1(x) = \cos(x) + 0.0005x \sin(x), y_2(x) = -\sin(x) + 0.0005x \cos(x) + 0.0005 \sin(x),$$

$$y_3(x) = \sin(x) - 0.0005x \cos(x), y_4(x) = \cos(x) + 0.0005x \sin(x) - 0.0005 \cos(x).$$

**Problem 4**
\[ y_1'' = \frac{-y_1}{(\sqrt{y_1^2 + y_2^2})^3}, \quad y_1(0) = 1, \quad y_1'(0) = 0, \]
\[ y_2'' = \frac{-y_2}{(\sqrt{y_1^2 + y_2^2})^3}, \quad y_2(0) = 0, \quad y_2'(0) = 1, \quad 0 \leq x \leq 15\pi. \]

Exact solutions are \( y_1(x) = \cos(x), y_2(x) = \sin(x). \)

Source: Dormand et al. [2].

The first order system:
\[ y_1' = y_2, \quad y_2' = -\frac{y_1}{(\sqrt{y_1^2 + y_2^2})^3}, \quad y_1(0) = 1, \quad y_2(0) = 0, \]
\[ y_3' = y_4, \quad y_4' = -\frac{y_3}{(\sqrt{y_1^2 + y_2^2})^3}, \quad y_1(0) = 0, \quad y_2(0) = 1, \quad 0 \leq x \leq 15\pi. \]

Exact solutions are \( y_1(x) = \cos(x), y_2(x) = -\sin(x), y_3(x) = \sin(x), y_4(x) = \cos(x). \)

**Problem 5**

\[ \frac{d^2 y(x)}{dx^2} = -y(x) + x, \quad y(0) = 1, \quad y'(0) = 2, \quad 0 \leq x \leq 500. \]

Exact solution \( y(x) = \sin(x) + \cos(x) + x. \)

Source: Allen and Wing [15].

The first order system:
\[ y_1' = y_2, \quad y_2' = -y_1 + x, \quad y_1(0) = 1, \quad y_2(0) = 2, \quad 0 \leq x \leq 500. \]

Exact solutions are \( y_1(x) = \sin(x) + \cos(x) + x, \quad y_2(x) = \cos(x) - \sin(x) + 1. \)

**Problem 6** (Inhomogeneous System)

\[ \frac{d^2 y_1(x)}{dx^2} = -v^2 y_1(x) + v^2 f(x) + f''(x), \quad y_1(0) = a + f(0), \quad y_1'(0) = f'(0), \]
\[ \frac{d^2 y_2(x)}{dx^2} = -v^2 y_2(x) + v^2 f(x) + f''(x), \quad y_2(0) = f(0), \quad y_2'(0) = va + f'(0), \quad 0 \leq x \leq 100. \]

Exact solutions are \( y_1(x) = a \cos(vx) + f(x), \quad y_2(x) = a \sin(vx) + f(x), \quad f(x) \) is chosen to be \( e^{-10x} \) and parameters \( v \) and \( a \) are 4 and 0.1 respectively.
Source: Lambert and Watson [16].

The first order system:

\[
\begin{align*}
y'_1 &= y_2, y'_2 = -16y_1 + 16e^{-10x} + 100e^{-10x}, y_1(0) = 1.1, y_2(0) = -10, \\
y'_3 &= y_4, y'_4 = -16y_3 + 16e^{-10x} + 100e^{-10x}, y_3(0) = 1, y_2(0) = -9.6.
\end{align*}
\]

Exact solutions are \( y_1(x) = 0.1 \cos(4x) + e^{-10x}, y_2(x) = -0.4 \sin(4x) - 10e^{-10x}, y_3(x) = 0.1 \sin(4x) + e^{-10x}, y_4(x) = 0.4 \cos(4x) - 10e^{-10x}. \)

Problem 7 (Nonlinear System)

\[
\begin{align*}
\frac{d^2 y_1(x)}{dx^2} &= -4x^2 y_1 - \frac{y_2}{\sqrt{y_1^2 + y_2^2}}, y_1(t_0) = 0, y'_1(x_0) = -\sqrt{\frac{\pi}{2}} \\
\frac{d^2 y_2(x)}{dx^2} &= -4x^2 y_2 + \frac{y_1}{\sqrt{y_1^2 + y_2^2}}, y_2(x_0) = 1, y'_2(x_0) = 0, \sqrt{\frac{\pi}{2}} \leq x \leq 50
\end{align*}
\]

Exact solutions are \( y_1(x) = \cos(x^2), y_2(t) = \sin(x^2). \)

Source: Sharp et al. [17].

The first order system:

\[
\begin{align*}
y'_1 &= y_2, y'_2 = -4x^2 y_1 - \frac{2y_3}{\sqrt{y_1^2 + y_2^3}}, y_1(0) = 0, y_2(0) = -\sqrt{2\pi}, \\
y'_3 &= y_4, y'_2 = -4x^2 y_2 + \frac{2y_1}{\sqrt{y_1^2 + y_2^3}}, y_1(0) = 1, y_2(0) = 0.
\end{align*}
\]

Exact solutions are \( y_1(x) = \cos(x^2), y_2(x) = -2x \sin(x^2), y_3(x) = \sin(x^2), y_4(x) = 2x \cos(x^2). \)

5 Implementation and Numerical Results

The set of test problems in section 4 is solved using the new method and the results are compared with the numerical results when the same set of test problems are reduced to first order system twice the dimension and solve using method by Fehlberg [18] and Butcher [19]. For the new method and the existing methods, the next step size is determined by

\[
h_{\text{new}} = 0.4 \times \left( \frac{TOL}{2 \times LTE} \right)^{p+1} \times h_{\text{old}}
\]

where TOL is the chosen tolerance, \( h_{\text{old}} \) is the current step size, \( p \) is the order of the method.
The numerical results are given in Figs. 1-7 and the notations used as follows:

MAXE – maximum global error (max \( \| y_n - y(x_n) \| \)), that is the computed solution minus the true solution.

ETSRKN 4(3) pair derived in this paper.
ERK 4(3) pair by Butcher [19].
ERK 4(3) pair by Fehlberg [18].

Figure 1: The efficiency curves of the ETSRKN4(3) method and its comparisons for Problem 1 with \( x_{\text{end}} = 50 \)
Figure 2: The efficiency curves of the ETSRKN4(3) method and its comparisons for Problem 2 with $x_{end} = 50$

Figure 3: The efficiency curves of the ETSRKN4(3) method and its comparisons for Problem 3 with $x_{end} = 1000$
Figure 4: The efficiency curves of the ETSRKN4(3) method and its comparisons for Problem 4 with $x_{end} = 15\pi$

Figure 5: The efficiency curves of the ETSRKN4(3) method and its comparisons for Problem 5 with $x_{end} = 500$
Figure 6: The efficiency curves of the ETSRKN4(3) method and its comparisons for Problem 6 with $x_{end} = 100$

Figure 7: The efficiency curves of the ETSRKN4(3) method and its comparisons for Problem 7 with $x_{end} = 50$
6 Discussion and Conclusion

From Figures 1-7, we observed that ETSRK4N(3) is more efficient when compared to ERK4(3)B and ERK4(3)F in terms of computational time. This is due to the fact that when second-order problems is solve using method ERK4(3)B and ERK4(3)F, it has to be reduce to first-order system that is twice its dimension. In terms of global error, ETSRK4N(3) produced smaller error compared to ERK4(3)B and ERK4(3)F for all problems except for problem 4 where ETSRK4N(3) global error is comparable with ERK4(3)B and ERK4(3)F. Thus we can conclude here that ETSRK4N(3) is more efficient than the existing technique where all the special second-order problems are solved directly whereby for other techniques, the problems need to reduce to first-order system of ODEs. Hence less time is needed to solve the same set of problems.

References


Received: June 5, 2014