Derivation of continuous explicit two-step Runge–Kutta methods of order three

Z. Bartoszewski\textsuperscript{a,}\textsuperscript{*}, Z. Jackiewicz\textsuperscript{b,}\textsuperscript{1}

\textsuperscript{a}Faculty of Applied Physics and Mathematics, Gdańsk University of Technology, Narutowicza 11/12, 80-952 Gdańsk, Poland
\textsuperscript{b}Department of Mathematics, Arizona State University, Tempe, AZ 85287, USA

Received 6 July 2005

Abstract

We describe a construction of continuous extensions to a new representation of two-step Runge–Kutta methods for ordinary differential equations. This representation makes possible the accurate and reliable estimation of local discretization error, facilitates the efficient implementation of these methods in variable stepsize environment, and adapts readily to the numerical solution of a class of delay differential equations. A number of numerical tests carried out on the obtained methods of order 3 with quadratic interpolants show their efficiency and robust performance which allow them to compete with the state-of-the-art \texttt{dde23} code from Matlab.

© 2006 Elsevier B.V. All rights reserved.

MSC: 65L05; 65L06

Keywords: Two-step Runge–Kutta methods; Continuous interpolants; Ordinary and delay differential equations; Nordsieck representation

1. Introduction

Over the last three decades many papers devoted to continuous extensions to Runge–Kutta methods have been published. The authors of the papers deal mainly with deriving continuous extensions to one-step Runge–Kutta methods and investigate their stability properties, their convergence and test the effectiveness of obtained methods on different problems, very often modelling biological or physical phenomena. We refer the reader to [2,3,5], where a general theory of such methods is described and examples of their application to different problems are provided. The reader can also find there interesting historical and bibliographical notes and extensive literature.

As the two-step Runge–Kutta methods (TSRK in short) were introduced not so long ago (see [7,1] for historical notes related to TSRK methods) and have been more intensively investigated over the last decade there is only one paper [8] dealing with continuous TSRK methods, where the variable stepsize and their continuity are the intrinsic part of the process of their construction.

\textsuperscript{*} Corresponding author.

E-mail addresses: zbart@mif.pg.edu.pl (Z. Bartoszewski), jackiewi@math.la.asu.edu (Z. Jackiewicz).

\textsuperscript{1} The work of this author was supported by the National Science Foundation under Grant NSF DMS-0509597.
The recently constructed [1] explicit TSRK methods of order 3 for ODEs were tested on many numerically demanding problems and their potential was demonstrated by comparison of the corresponding codes based on TSRK methods with the state-of-the-art ode23 code from Matlab ODE suite and they have proved to be quite robust.

The main features of the new TSRK methods:

- they are based on approximations to the scaled derivatives of the solution up to the order \( p \) (Nordsieck vector);
- no extra evaluations of the right-hand side of the differential equation for the accurate computation of the past values after the stepsize change;
- an efficient and reliable estimation of the local discretization errors;
- they facilitate efficient implementation of TSRK formulas in a variable stepsize environment.

So, it is natural to make an attempt to derive continuous extensions to these methods (CTSRK methods in short) which preserve all the best features of the underlying discrete TSRK methods as well as their effectiveness and to demonstrate their potential by comparison of the corresponding codes based on their continuous extensions with the state-of-the-art code from Matlab. Later in the paper we derive continuous extensions to two TSRK methods of order 3 corresponding to two different error constants: \( E = \frac{1}{48} \) and \( E = \frac{1}{120} \). However, it is worth mentioning that, in contrast to above-mentioned paper [8], we derive a continuous extension to discrete TSRK methods primarily constructed for the constant stepsize which are efficiently implemented in a variable stepsize environment.

The organization of the paper is as follows. In Section 2 we give a short description of Nordsieck representation of TSRK methods and all the necessary data that makes the paper self-contained and makes it possible to implement the CTSRK methods without referring to paper [1]. In Section 3 we derive continuous extensions to two TSRK methods of order 3. In Section 4 we test the derived CTSRK methods on four numerical examples (two one-dimensional problems and two systems of delay differential equations (DDE)) and compare the corresponding codes based on their continuous extensions with the state-of-the-art dde23 code from Matlab by taking into account the number of steps, the number of function calls and the achieved accuracy. Finally, we end with some concluding remarks in Section 5.

2. A short description of Nordsieck representation of TSRK methods

We define our approximating formulae by reference to the initial-value problem for the system of ODEs

\[
y'(x) = f(y(x)), \quad x \in [x_0, X], \quad y(x_0) = y_0, \tag{1}
\]

where \( f : \mathbb{R}^n \rightarrow \mathbb{R}^n \). The extension to non-autonomous problems \( y'(x) = f(x, y(x)) \) is standard, and the extension to simple DDE

\[
y'(x) = f(x, y(x), y(x - \tau)), \quad x \in [x_0, X] \text{ where } y(x) = \phi(x), \quad x \in [x_0 - \tau, x_0],
\]

where \( \tau > 0 \) follows the lines discussed in the literature, notably [3, p. 9]. (For small \( \tau \) and for more complex DDE, not considered here, the adaptation require further discussion.)

To solve the initial-value problem for the system of ODEs (1) on the non-uniform grid given by

\[x_0 < x_1 < x_2 < \cdots < x_N, \quad x_N = X,\]

we consider the class of explicit TSRK methods defined by

\[
\begin{align*}
Y_i^{[n]} &= u_i \tilde{y}_{n-1} + (1 - u_i) y_n + h_n \sum_{j=1}^{s} (a_{ij} f(\tilde{Y}_j^{[n-1]}) + b_{ij} f(Y_j^{[n]})), \\
y_{n+1} &= y_n + h_n \sum_{j=1}^{s} (v_j f(\tilde{Y}_j^{[n-1]}) + w_j f(Y_j^{[n]})),
\end{align*}
\tag{2}
\]

\( n = 1, 2, \ldots, N - 1 \). Here: \( h_n = x_{n+1} - x_n, \ y_n \approx y(x_n), \ \tilde{y}_{n-1} \approx y(x_n - h_n), \ \tilde{Y}_j^{[n-1]} \approx y(x_n + (c_j - 1)h_n), \ Y_i^{[n]} \approx y(x_n + c_i h_n), \ i = 1, 2, \ldots, s, \ c = [c_1, \ldots, c_s]^T \) —a given vector and \( u = [u_1, \ldots, u_s]^T, \ A = [a_{ij}], \ B = [b_{ij}], \ v = [v_1, \ldots, v_s]^T, \ w = [w_1, \ldots, w_s]^T \) —coefficients of the method.
An alternative vector form of (2) is
\[
Y^{[n]} = \begin{cases}
    (u \otimes I_m)\tilde{y}_{n-1} + ((e - u) \otimes I_m)y_n \\
    + h_n ((A \otimes I_m) f(\tilde{Y}^{[n-1]}) + (B \otimes I_m) f(Y^{[n]})),
\end{cases}
\]
\[y_{n+1} = y_n + h_n ((u^T \otimes I_m) f(\tilde{Y}^{[n-1]}) + (w^T \otimes I_m) f(Y^{[n]})),
\]
for \(n = 1, 2, \ldots, N - 1\), where \(e = [1, \ldots, 1]^T \in \mathbb{R}^s\), \(I_m\) is the identity matrix of dimension \(m\), and
\[
Y^{[n]} = \begin{bmatrix} Y^{[n]}_1 \\ \vdots \\ Y^{[n]}_s \end{bmatrix}, \quad f(Y^{[n]}) := \begin{bmatrix} f(Y^{[n]}_1) \\ \vdots \\ f(Y^{[n]}_s) \end{bmatrix}.
\]

We also need:

- a starting procedure to advance from \(x_0\) to \(x_1\) with an initial step \(h_0\);
- approximations \(\tilde{y}_n \approx y(x_1 - h_1)\) and \(\tilde{Y}^{[0]}_i \approx y(x_1 - (c_i - 1)h_1)\) corresponding to a new stepsize \(h_1\);
- a method of computation of approximations:
\[
\tilde{y}_n \approx y(x_{n+1} - h_{n+1}), \quad h_{n+1} f(\tilde{Y}^{[n]}_i) \approx h_{n+1} y'(x_{n+1} + (c_i - 1)h_{n+1})
\]
corresponding to the stepsize \(h_{n+1}\) after the step from \(x_n\) to \(x_{n+1}\) is accepted.

The order conditions for the method (2) are given by the following theorem (cp [1, Theorem 1]):

**Theorem 1.** The method (2) has order \(p\) and stage order \(q = p\) if and only if
\[
\begin{cases}
    A(c - e)^j - 1 + Bc^j - 1 = \frac{c^j - (-1)^j u}{j}, \\
    v^T(c - e)^j - 1 + w^T c^j - 1 = \frac{1}{j}, \quad j = 1, 2, \ldots, p,
\end{cases}
\]
where \(c^j\) denotes componentwise operation. Moreover, the error constant \(E\) is given by
\[
E = \frac{1}{(p + 1)!} - \frac{v^T(c - e)^p + w^T c^p}{p!}.
\]

The formula for the local discretization error \(\text{le}(x_{n+1})\) of the method (2) of order \(p\) and stage order \(q = p\) at \(x_{n+1}\) is given by the formula
\[
\text{le}(x_{n+1}) := y(x_{n+1}) - y_{n+1} = Eh_n^{p+1} y^{(p+1)}(x_{n+1}) + O(h_n^{p+2}).
\]

**2.1. Starting procedure**

To advance from \(x_0\) to \(x_1\) with an initial stepsize \(h_0\) we need an explicit continuous Runge–Kutta (CRK) method of uniform order \(p\), for example, given by
\[
\begin{cases}
    Y_i = y_0 + h_0 \sum_{j=1}^{i-1} a_{ij}^s f(Y_j), \quad i = 1, 2, \ldots, s^*, \\
    y_{n+1}(x_0 + \theta h_0) = y_0 + h_0 \sum_{j=1}^{s^*} b_j^s(\theta) f(Y_j),
\end{cases}
\]
for \(\theta \in [0, 1]\), where, \(Y_i\) are approximations to \(y(x_0 + c_i^* h_0)\) and \(c^* = [c_1^*, \ldots, c_{s^*}^*]^T\) is a given abscissa vector, \(a_{ij}^*\) are coefficients and \(b_j^s(\theta)\) are continuous weights. Such methods were investigated by Zennaro [16,17], Owren [11], and Owren and Zennaro [12,13].
Following [4] we define the initial stepsize $h_0$ by the formula  
$$h_0 = \min(|X - x_0|, \text{Tol}^{1/(p+1)} \| f(x_0, y_0) \|^{-1}),$$
where $p$ is the order of the method and Tol is a given accuracy tolerance. This formula is based on a somewhat arbitrary assumption that the local discretization error is proportional to $(h_0 \| f(x_0, y_0) \|)^{p+1}$ and has many defects which are discussed in [14]. In all our examples it led to rejected steps at the beginning of integration. The refinement of this procedure is discussed in [5].

Assuming that $h_1 \leq h_0$ we can compute $\tilde{y}_0$ and $Y_i^{[0]}$ which are needed to start the TSRK method (2) from the formulas
\begin{align}
\tilde{y}_0 &= y_h(x_0 + (1 - r_1)h_0), \\
Y_i^{[0]} &= y_h(x_0 + (1 + (c_i - 1)r_1)h_0), \quad i = 1, 2, \ldots, s,
\end{align}
where $r_1 = h_1/h_0$, and $y_h$ is defined by (5).

### 2.2. Computation of approximations to the Nordsieck vector $z(x_{n+1}, h_n)$ and $h_n^{p+1}y^{(p+1)}(x_{n+1})$

To implement the method (2) in a variable stepsize environment we need an approximation $h_{n+1}f(\tilde{Y}^{[n]})$ to $h_{n+1}y'(x_{n+1} + (c - e)h_n)$ and $\tilde{y}_n$ to $y(x_{n+1} - h_{n+1})$ after the step from $x_n$ to $x_{n+1}$ is completed. They are expressed in terms of approximation $\tilde{z}(x_{n+1}, h_n)$ to the Nordsieck vector

$$z(x_{n+1}, h_n) = [y(x_{n+1}), h_n y'(x_{n+1}), \ldots, h_n^p y^{(p)}(x_{n+1})]^T.$$

The vector $\tilde{z}(x_{n+1}, h_n)$ has the form
\begin{equation}
\tilde{z}(x_{n+1}, h_n) = (\alpha \otimes I_m)y_n + (\beta \otimes I_m)y_{n+1} + h_n (\Gamma \otimes I_m)f(Y^{[n]}),
\end{equation}
where $\alpha = [x_{1i}]_{i=1,1,\ldots,p}$, $\beta = [\beta_1]_{i=1,1,\ldots,p}$, $\Gamma = [\gamma_{ij}]_{i=1,1,\ldots,p}$.

If some additional conditions are fulfilled then (cp [1, Theorem 3]) $\tilde{z}$ satisfies
\begin{equation}
\tilde{z}(x_{n+1}, h_n) = z(x_{n+1}, h_n) + O(h_n^{p+2}),
\end{equation}
if and only if $\alpha$, $\beta$, and $\Gamma$ satisfy the system of equations
\begin{align}
&z + \beta e_1 + \Gamma C = I_{p+1}, \\
&\left(\frac{(-1)^{p+1} \alpha}{(p+1)!} - E \beta + \frac{\Gamma(c - e)^p}{p!}\right) = 0.
\end{align}

Here, $e_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^{p+1}$, $\tau = [1, -1, \frac{1}{2}, \ldots, \frac{(-1)^p}{p!}]^T$, $I_{p+1}$ is the identity matrix of dimension $p + 1$ and
\begin{equation}
C = \begin{bmatrix}
0 & e - c - e & \cdots & \frac{(c - e)^{p-1}}{(p-1)!}
\end{bmatrix}.
\end{equation}

If some additional conditions are fulfilled then (cp [1, Theorem 3]) we also have the estimate $h_n^{p+1}y^{(p+1)}(x_{n+1})$ in the form
\begin{equation}
h_n^{p+1}y^{(p+1)}(x_{n+1}) = z_{p+1}y_n + \beta_{p+1}y_{n+1} + (\gamma_{p+1} \otimes I_m)h_n f(Y^{[n]}) + O(h_n^{p+2}),
\end{equation}
if and only if $z_{p+1}$, $\beta_{p+1} \in \mathbb{R}$ and $\gamma_{p+1} = [\gamma_{p+1,1,1}, \ldots, \gamma_{p+1,s,1}] \in \mathbb{R}^s$ satisfy the system of equations
\begin{align}
&\beta_{p+1}T + \beta_{p+1}e_1 + \gamma_{p+1}C = 0, \\
&\left(\frac{(-1)^{p+1} \beta_{p+1}}{(p+1)!} - E \beta_{p+1} + \frac{(c - e)^p}{p!}\right) = 1.
\end{align}

It means that the local discretization error of the method can be estimated by the formula
\begin{equation}
est(x_{n+1}) = E \delta(x_{n+1}, h_n)
\end{equation}
Table 1
Coefficients of TSRK formulas

<table>
<thead>
<tr>
<th>E</th>
<th>( \frac{1}{38} )</th>
<th>( \frac{1}{120} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_1 )</td>
<td>-1.3530148605919576</td>
<td>0.0736695598414358</td>
</tr>
<tr>
<td>( u_2 )</td>
<td>-0.1283921311571470</td>
<td>-0.0204487284571901</td>
</tr>
<tr>
<td>( u_3 )</td>
<td>-0.5656852809667421</td>
<td>0.5449670091349459</td>
</tr>
<tr>
<td>( b_{21} )</td>
<td>1.5112481814304699</td>
<td>0.9854340015176510</td>
</tr>
<tr>
<td>( b_{31} )</td>
<td>0.9905463465163611</td>
<td>1.76608305214015</td>
</tr>
<tr>
<td>( b_{32} )</td>
<td>0.8822201613023282</td>
<td>0.46701736875624</td>
</tr>
<tr>
<td>( w_1 )</td>
<td>0.6949205878983691</td>
<td>1.4898375016445287</td>
</tr>
</tbody>
</table>

where \( E \) is the error constant given by (4) and \( \delta(x_{n+1}, h_n) \) is defined by

\[
\delta(x_{n+1}, h_n) = \alpha_{p+1} \gamma_n + \beta_{p+1} \gamma_{n+1} + (\gamma_{p+1} \otimes I_m) h_n f(Y^{[n]}_1) .
\] (12)

2.3. Computation of \( \bar{y}_n \) and \( h_{n+1} f(\bar{Y}^{[n]}) \)

Put \( r_{n+1} = h_{n+1}/h_n, D(r_n) = \text{diag}(1, r_n, \ldots, r_n^p) \). Then (see [1]) we have

\[
\bar{y}_n = (r^T D(r_{n+1}) \otimes I_m) \tilde{z}(x_{n+1}, h_n) + g \left( \frac{(-1)^{p+1}}{(p+1)!} r_{n+1}^{p+1} - E(1 - r_{n+1})^{p+1} \right) \delta(x_{n+1}, h_n) .
\] (13)

Then for \( h_{n+1} f(\bar{Y}^{[n]}) \) we have the formula

\[
h_{n+1} f(\bar{Y}^{[n]}) = (C D(r_{n+1}) \otimes I_m) \tilde{z}(x_{n+1}, h_n) + g \left( \frac{(c - e)^p}{p!} \otimes I_m g \right) r_{n+1}^{p+1} \delta(x_{n+1}, h_n) .
\] (14)

Observe that (13) and (14) can be computed without any extra evaluations of the right-hand side of Eq. (1).

2.4. TSRK methods for \( p = q = s = 3, c = [0, \frac{1}{2}, 1]^T, E = \frac{1}{38} \) and \( E = \frac{1}{120} \)

The coefficients of two TSRK methods derived by a method of maximization of stability region are given in Table 1. The remaining coefficients of the methods can be found from Eqs. (3) and (4). The components of the vectors \( \alpha, \beta, \gamma_4 \), the entries of the matrix \( \Gamma \) and the numbers \( z_4 \) and \( \beta_4 \) can be found from Eqs. (9) and (11).

3. Continuous extensions to the TSRK methods of order 3

We will construct the continuous extension to TSRK method (2) in the form

\[
\begin{align*}
Y_i^{[n]} &= u_i \bar{y}_{n-1} + (1 - u_i) y_n + h_n \sum_{j=1}^s (a_{ij} f(\bar{Y}_{j}^{[n-1]}) + b_{ij} f(Y_j^{[n]})) , \\
y_n (x_n + \theta h_n) &= y_n + h_n \sum_{j=1}^s (v_j(\theta) f(\bar{Y}_{j}^{[n-1]}) + w_j(\theta) f(Y_j^{[n]})) , \quad \theta \in [0, 1],
\end{align*}
\] (15)

where \( n = 0, 1, \ldots, N - 1 \).
Observe that such a method is zero stable (cp [8, Theorem 1]). For our continuous extensions of TSRK methods (with constant step) let

\[
\begin{aligned}
C_\mu = \frac{c^\mu - (-1)^\mu u}{\mu!} - \frac{A(c - e)^{\mu-1}}{(\mu - 1)!} - \frac{Be^{\mu-1}}{(\mu - 1)!}, & \quad \mu = 1, 2, \ldots, \\
\hat{C}_\mu(\theta) = \frac{\theta^\mu}{\mu!} - \frac{v(\theta)^T(c - e)^{\mu-1}}{(\mu - 1)!} - \frac{w(\theta)^T e^{\mu-1}}{(\mu - 1)!}, & \quad \mu = 1, 2, \ldots, \\
E(\theta) = \frac{\theta^{p+1}}{(p + 1)!} - \frac{v(\theta)^T(c - e)^p}{p!} - \frac{w(\theta)^T e^p}{p!}.
\end{aligned}
\]

(16)

Observe that the equations \(C_\mu(\theta) = C_\mu = 0, \quad \mu = 0, \ldots, p \) and \(\hat{C}_\mu(1) = 0, \mu = 0, \ldots, p\) are equivalent to (3) and that for \(\theta = 1\) (17) is equivalent to (4).

These equations and the theorem below are valid for continuous TSRK methods of an arbitrary order \(p\). Following exactly the method that was used in [8] to prove Theorem 3 we can prove the following result.

**Theorem 2.** Assume that the continuous TSRK method (15) is zero stable, the errors of the initial approximations \(y_0\) and \(y_1\) are of order \(O(h^p)\) and that it has order of consistency \(p - 1\) and stage order of consistency \(p\), i.e.,

\[
\hat{C}_\mu(\theta) = 0, \quad C_\mu = 0,
\]

(18)

for \(\mu = 1, 2, \ldots, p - 1, \theta \in [0, 1]\) and that

\[
\hat{C}_\mu(1) = 0.
\]

(19)

Then the method has order of convergence \(p\) and stage order of convergence \(p\).

So, in order to construct a continuous extension to the given TSRK method our method defined by Eqs. (15) will satisfy conditions (18) and (19) that guarantee its convergence with uniform order \(p\). The constructed TSRK methods of order 3 satisfy the second condition in (18) and condition (19) by construction. To secure the continuity of the solution the following conditions must be satisfied:

\[
\begin{align*}
v(0) &= w(0) = (0, \ldots, 0)^T, \\
v(1) &= v, \quad w(1) = w.
\end{align*}
\]

(20)

These conditions also preserve the good stability properties of the underlying discrete method.

We will look for the functions \(v(\theta)\) and \(w(\theta)\) for which their components are quadratic functions, i.e. of the form

\[
\begin{align*}
v(\theta) &= \begin{bmatrix} x_{01} + x_1 \theta + x_2 \theta^2 \\ x_{02} + x_3 \theta + x_4 \theta^2 \\ x_{03} + x_5 \theta + x_6 \theta^2 \end{bmatrix}, \\
w(\theta) &= \begin{bmatrix} x_{04} + x_7 \theta + x_8 \theta^2 \\ x_{05} + x_9 \theta + x_{10} \theta^2 \\ x_{06} + x_{11} \theta + x_{12} \theta^2 \end{bmatrix}.
\end{align*}
\]

(21)

The first condition of (20) results in \(x_{01} = x_{02} = x_{03} = x_{04} = x_{05} = x_{06} = 0\).

The second condition of (20) and the first condition of (18) result in a linear system of 10 equations from which one equation can be eliminated. Solving this system for \(x_2, x_4, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}\) we obtain the following
For $p = 3$ we will choose the two parameters $x_1$ and $x_3$ in such a way that the graphs of the error function (17) and the function $\hat{C}_3(\theta)$ deviate from the $x$-axis as little as possible for as long time as possible remembering that $E(1) = E$, and that the error constant $E$ was treated as a parameter of the method during the process of construction of the discrete method. For

$$a = -x_1 - \left(\frac{1}{3}\right)x_3 + \left(\frac{1}{3}\right)x_9 + x_{11},$$

$$b = -x_2 - \left(\frac{1}{3}\right)x_4 + \left(\frac{1}{3}\right)x_{10} + x_{12},$$

$$e = x_1 + \left(\frac{1}{3}\right)x_3 + \left(\frac{1}{3}\right)x_9 + x_{11},$$

$$f = x_2 + \left(\frac{1}{3}\right)x_4 + \left(\frac{1}{3}\right)x_{10} + x_{12},$$

(23)
Table 2
Example 1: cost statistics and global error

<table>
<thead>
<tr>
<th>Tol</th>
<th>ctsrk3</th>
<th>dde23</th>
<th>ctsrk3</th>
<th>dde23</th>
<th>ctsrk3</th>
<th>dde23</th>
<th>ctsrk3</th>
<th>dde23</th>
</tr>
</thead>
<tbody>
<tr>
<td>E (=\frac{1}{48})</td>
<td>ns</td>
<td>ns</td>
<td>nr</td>
<td>nr</td>
<td>nfc</td>
<td>nfc</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>(10^{-6})</td>
<td>2888</td>
<td>2365</td>
<td>109</td>
<td>17</td>
<td>6013</td>
<td>7147</td>
<td>1.6e - 07</td>
<td>2.8e - 06</td>
</tr>
<tr>
<td>(10^{-8})</td>
<td>4565</td>
<td>7051</td>
<td>3</td>
<td>22</td>
<td>9155</td>
<td>21220</td>
<td>2.7e - 10</td>
<td>8.8e - 08</td>
</tr>
<tr>
<td>(10^{-10})</td>
<td>14517</td>
<td>21825</td>
<td>0</td>
<td>43</td>
<td>29053</td>
<td>65605</td>
<td>1.3e - 12</td>
<td>5.2e - 09</td>
</tr>
<tr>
<td>(10^{-12})</td>
<td>56606</td>
<td>72028</td>
<td>89</td>
<td>1835</td>
<td>113419</td>
<td>221590</td>
<td>1.5e - 14</td>
<td>2.7e - 10</td>
</tr>
<tr>
<td>E (=\frac{1}{120})</td>
<td>ns</td>
<td>ns</td>
<td>nr</td>
<td>nr</td>
<td>nfc</td>
<td>nfc</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>(10^{-6})</td>
<td>1855</td>
<td>2365</td>
<td>5</td>
<td>17</td>
<td>3739</td>
<td>7147</td>
<td>5.8e - 07</td>
<td>2.8e - 06</td>
</tr>
<tr>
<td>(10^{-8})</td>
<td>2691</td>
<td>7051</td>
<td>17</td>
<td>22</td>
<td>5435</td>
<td>21220</td>
<td>6.5e - 10</td>
<td>8.8e - 08</td>
</tr>
<tr>
<td>(10^{-10})</td>
<td>14767</td>
<td>21825</td>
<td>0</td>
<td>43</td>
<td>29553</td>
<td>65605</td>
<td>1.4e - 12</td>
<td>5.2e - 09</td>
</tr>
<tr>
<td>(10^{-12})</td>
<td>69801</td>
<td>72028</td>
<td>7</td>
<td>1835</td>
<td>139645</td>
<td>221590</td>
<td>1.3e - 14</td>
<td>2.7e - 10</td>
</tr>
</tbody>
</table>

The best choice of \(a\) and \(e\) is:

\[ a = -\left(\frac{1}{16}\right) \text{ for } E = \frac{1}{48} \text{ and } a = -0.1235 \text{ for } E = \frac{1}{120}, \text{ and } e = -\left(\frac{1}{6}\right). \]

For each method we put \(x_5 = 1\). The graphs of the error function for \(E = \frac{1}{48}\) and \(E = \frac{1}{120}\) are presented in Figs. 1 and 2, respectively, and the graph of \(\hat{C}_3\) is presented in Fig. 3.
Table 3
Example 2: cost statistics and global error

<table>
<thead>
<tr>
<th></th>
<th>ctsrk3</th>
<th>dde23</th>
<th>ctsrk3</th>
<th>dde23</th>
<th>ctsrk3</th>
<th>dde23</th>
<th>ctsrk3</th>
<th>dde23</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tol</td>
<td>ns</td>
<td>ns</td>
<td>nr</td>
<td>nr</td>
<td>nfc</td>
<td>nfc</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>$E = 1 \times 10^{-6}$</td>
<td>1276</td>
<td>1050</td>
<td>8</td>
<td>19</td>
<td>2587</td>
<td>3208</td>
<td>7.8e-03</td>
<td>2.9e-00</td>
</tr>
<tr>
<td>$E = 1 \times 10^{-8}$</td>
<td>5710</td>
<td>4985</td>
<td>19</td>
<td>23</td>
<td>11477</td>
<td>15025</td>
<td>2.8e-04</td>
<td>5.5e-02</td>
</tr>
<tr>
<td>$E = 1 \times 10^{-10}$</td>
<td>26677</td>
<td>23123</td>
<td>31</td>
<td>20</td>
<td>53435</td>
<td>69430</td>
<td>1.6e-05</td>
<td>7.2e-04</td>
</tr>
<tr>
<td>$E = 1 \times 10^{-12}$</td>
<td>123476</td>
<td>107304</td>
<td>58</td>
<td>22</td>
<td>247087</td>
<td>321979</td>
<td>1.8e-07</td>
<td>1.1e-05</td>
</tr>
</tbody>
</table>

Table 4
Example 3: cost statistics and global error

<table>
<thead>
<tr>
<th></th>
<th>ctsrk3</th>
<th>dde23</th>
<th>ctsrk3</th>
<th>dde23</th>
<th>ctsrk3</th>
<th>dde23</th>
<th>ctsrk3</th>
<th>dde23</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tol</td>
<td>ns</td>
<td>ns</td>
<td>nr</td>
<td>nr</td>
<td>nfc</td>
<td>nfc</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>$E = 1 \times 10^{-6}$</td>
<td>9384</td>
<td>8567</td>
<td>12</td>
<td>6</td>
<td>18811</td>
<td>25720</td>
<td>4.4e-05</td>
<td>1.7e-05</td>
</tr>
<tr>
<td>$E = 1 \times 10^{-8}$</td>
<td>44371</td>
<td>39967</td>
<td>17</td>
<td>2</td>
<td>88795</td>
<td>119908</td>
<td>6.3e-07</td>
<td>1.4e-07</td>
</tr>
<tr>
<td>$E = 1 \times 10^{-10}$</td>
<td>206954</td>
<td>185709</td>
<td>18</td>
<td>0</td>
<td>413963</td>
<td>557128</td>
<td>6.9e-09</td>
<td>1.2e-09</td>
</tr>
</tbody>
</table>

Table 5
Example 4: cost statistics and global error

<table>
<thead>
<tr>
<th></th>
<th>ctsrk3</th>
<th>dde23</th>
<th>ctsrk3</th>
<th>dde23</th>
<th>ctsrk3</th>
<th>dde23</th>
<th>ctsrk3</th>
<th>dde23</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tol</td>
<td>ns</td>
<td>ns</td>
<td>nr</td>
<td>nr</td>
<td>nfc</td>
<td>nfc</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>$E = 1 \times 10^{-6}$</td>
<td>949</td>
<td>742</td>
<td>22</td>
<td>10</td>
<td>1961</td>
<td>2257</td>
<td>9.1e-08</td>
<td>1.2e-06</td>
</tr>
<tr>
<td>$E = 1 \times 10^{-8}$</td>
<td>2632</td>
<td>2465</td>
<td>15</td>
<td>5</td>
<td>5313</td>
<td>7411</td>
<td>2.9e-09</td>
<td>5.3e-08</td>
</tr>
<tr>
<td>$E = 1 \times 10^{-10}$</td>
<td>11054</td>
<td>10235</td>
<td>19</td>
<td>6</td>
<td>22165</td>
<td>30724</td>
<td>5.8e-11</td>
<td>2.7e-10</td>
</tr>
<tr>
<td>$E = 1 \times 10^{-12}$</td>
<td>51647</td>
<td>46481</td>
<td>22</td>
<td>140</td>
<td>103357</td>
<td>139864</td>
<td>7.8e-12</td>
<td>7.3e-12</td>
</tr>
</tbody>
</table>

4. Numerical examples

To illustrate the potential of the new formulas we compared the variable stepsize implementation of these methods with state-of-the-art dde23 code from Matlab (see [15]) using two scalar and two two-dimensional examples. The cost and accuracy statistics are given in Tables 2–5 for the corresponding examples and also illustrated in Figs. 4–7.
correspondingly. We use the following notation: ‘$E$’—the error constant, ‘Tol’—the absolute and relative tolerance, ‘ns’—the number of steps, ‘nr’—the number of rejected steps and ‘nfc’—the number of function calls.

**Example 1** (Ito et al. [6]).

\[
y'(t) = Ay(t) + y(t - (3\pi/2)) - A \sin(t), \quad t \in (0, 20],
\]

\[
y(t) = \exp(pt) + \sin t, \quad t \leq 0;
\]

where $A = p - \exp(-3p\pi/2); \ p = -1$. Exact solution: $y(t) = \exp(pt) + \sin t$. 

Fig. 4. Example 1: precision versus number of function calls and number of steps.

Fig. 5. Example 2: precision versus number of function calls and number of steps.
Fig. 6. Example 3: precision versus number of function calls and number of steps.

Fig. 7. Example 4: precision versus number of function calls and number of steps.

**Example 2 (Jones [9]).**

$$y'(t) = -\lambda y(t - 1)(1 + y(t)), \quad t \in (0, 20],$$

$$y(t) = t, \quad t \leq 0;$$

where $\lambda = 3$. Exact solution: not available.
Example 3 (Oberle and Pesch [10]).

\[
y_1'(t) = y_2(t), \quad t \in (0, 10],
\]

\[
y_2'(t) = 0.5(y_2(t - 0.1))^3 - 10y_2(t) - 25y_2(t - 0.1) - 100y_1(t),
\]

\[
y_1(t) = 0.5, \quad y_2(t) = 2\pi \cos(20\pi t), \quad t \leq 0;
\]

Exact solution: not available.

Example 4 (Oberle and Pesch [10]).

\[
y_1'(t) = \frac{1.1}{1 + \sqrt{10(y_1(t) - 20))^{1.25}} - \frac{10y_1(t)}{1 + 40y_2(t)},
\]

\[
y_2'(t) = \frac{100y_1(t)}{1 + 40y_2(t)} - 2.43y_2(t), \quad t \in (0, 100],
\]

\[
y_1(t) = 0.352556756666666667, \quad y_2(t) = 0.343571163666666667, \quad t \leq 0;
\]

Exact solution: not available.

5. Conclusions

We constructed two continuous extensions to the third order TSRK methods and tested their efficiency on a number of examples taken from the literature. We also compared their performance with the performance of dde23 Matlab code. It is well known (see, for example [3]) that multistep methods are more sensitive to the smoothness of the solution. We can observe this fact by comparing the performance of our methods for Example 1 with smooth solution and Example 2 with a solution suffering from \((n + 1)\text{st}\) order discontinuity at \(t = n\). A similar observation can be made in the case of two-dimensional examples by comparing the performance of our methods for Example 3 with smoother solution than the solution of Example 4. These intrinsic differences between one-step and multistep methods result in severe restrictions in the implementation of multistep methods. Although in our implementation of continuous extensions to TSRK methods of third order discontinuities of the solution to DDEs were not handled at any way, the deterioration of their performance in comparison with one-step methods on which dde23 is based is not very drastic, which indicates that they may be used as an alternative methods of solution of many problems that require dense output.

Acknowledgment

The authors wish to express their gratitude to anonymous referees for their useful comments and to Guest Editors for editorial advice.

References