Applied Numerical Mathematics 94 (2015) 75–87

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

An analysis of the Prothero–Robinson example for constructing new adaptive ESDIRK methods of order 3 and 4

Joachim Rang

Institute of Scientific Computing, TU Braunschweig, Hans-Sommer Str. 65, 38106 Braunschweig, Germany

ARTICLE INFO

Article history:
Received 6 August 2014
Received in revised form 4 March 2015
Accepted 13 March 2015
Available online 17 March 2015

Keywords:
ODEs
Order reduction
DIRK methods
Adaptivity
Differential-algebraic equations

ABSTRACT

Explicit singly-diagonally-implicit (ESDIRK) Runge–Kutta methods have usually order reduction if they are applied on stiff ODEs, such as the example of Prothero and Robinson. It can be observed that the numerical order of convergence decreases to the stage order, which is limited to two. In this paper we analyse the Prothero–Robinson example and derive new order conditions to avoid order reduction. New third and fourth order ESDIRK methods are created, which are applied to the Prothero–Robinson example and to an index-2 DAE. Numerical examples show that the new methods have better convergence properties than usual ESDIRK methods.

1. Introduction

It is possible with Runge–Kutta methods to solve stiff ODEs (see for example the problem of Prothero–Robinson [14]) and differential algebraic equations [7,19]. Explicit Runge–Kutta methods may not be a good choice, because to get a stable numerical solution a step size restriction should be accepted, i.e. the problem should be solved with very small time steps. Therefore, it might be better to use implicit Runge–Kutta methods. Yet, the convergence order may not be achieved [7,19], i.e. the so-called order reduction phenomenon can be observed. In [7] convergence results for implicit Runge–Kutta methods applied on the example of Prothero and Robinson [14] can be found where the so-called stage order plays an important role. Ostermann and Roche prove in [13] that implicit Runge–Kutta methods may have a fractional order of convergence for general linear ODEs.

Fully implicit Runge–Kutta methods may be ineffective for solving high dimensional ODEs, since they need a high computational effort, which can be reduced if diagonally implicit Runge–Kutta (DIRK) methods are used. We distinguish two classes of DIRK methods. First we have the singly-diagonally-implicit Runge–Kutta (SDIRK) methods, where all diagonal elements are non-zero and equal. In this case the stage order is limited to one. Therefore, in [2] Cameron introduced the so-called quasi-stage order. This concept is improved in paper [3], where the method SDIRK2 is derived. In [15] an analysis of the example of Prothero and Robinson applied on SDIRK methods can be found. An analysis of the local error and numerical examples shows that at least second order convergence can be achieved in the case of the stiff Prothero–Robinson example [16], since further order conditions are fulfilled.

Stage order 2 is possible if the first diagonal entry is equal to zero. In this case the methods are called *explicit singly-diagonally-implicit Runge–Kutta (ESDIRK) methods*. These methods are widely used in the solution of ODEs and PDEs [10,9]. The order reduction can be decreased if order conditions for index-2 DAEs [6,8] are satisfied, as is shown in [20,18]. But

E-mail address: j.rang@tu-bs.de.

http://dx.doi.org/10.1016/j.apnum.2015.03.003

when these methods are applied on medium stiff problems strong order reduction may be observed, although the numerical approximations are of good quality [16].

The main task of this paper is the generalisation of the concept in [15] to ESDIRK methods to develop further order conditions. With the help of these conditions we create more effective ESDIRK methods.

The paper is structured as follows: we first consider ESDIRK methods and apply them to the Prothero–Robinson example. In Section 2 we consider the local error of these methods in the stiff case. We derive further order conditions to improve the convergence order for stiff ODEs. Two third order and one fourth order ESDIRK methods are created in Section 3, and finally we present some numerical results and apply our new methods on several test examples. In the case of the Prothero–Robinson we show that we reach full order p. Since our new methods satisfy order conditions for DAEs of index 2 our second example is index-2 DAE. It is shown that our new methods are much more effective than the known ones.

2. ESDIRK methods

2.1. Application to ODEs

We start our considerations with an ODE of the form

$$\dot{\mathbf{u}} = \mathbf{F}(t, \mathbf{u}), \quad \mathbf{u}(0) = \mathbf{u}_0. \tag{1}$$

A Runge–Kutta method (RK method) with s internal stages [7,19] is a one-step-method for solving (1) of the form

$$\mathbf{k}_i = \mathbf{F}\left(t_m + c_i\tau_m, \mathbf{u}_m + \tau_m \sum_{j=1}^{s} a_{ij}\mathbf{k}_j\right), \quad i = 1, \ldots, s, \tag{2}$$

$$\mathbf{u}_{m+1} = \mathbf{u}_m + \tau_m \sum_{i=1}^{s} b_i \mathbf{k}_i. \tag{3}$$

The coefficients a_{ij}, b_i and c_i should be chosen in such a way that some order conditions are satisfied to obtain a sufficient consistency order. In this paper the coefficients of the RK method (2)–(3) satisfy $a_{ij} = 0$ for $i < j$, $i, j \in \{1, \ldots, s\}$, $a_{11} = 0$, and $a_{ii} \neq 0$ for $i \in \{2, \ldots, s\}$. RK methods satisfying these conditions are called *explicit singly-diagonally-implicit Runge–Kutta methods (ESDIRK methods)*. These methods are discussed in several papers and books, e.g. in [19,7]. Butcher introduces in [1] the so-called simplifying conditions, which are given by

$$B(p): \quad \sum_{i=1}^{s} b_i c_i^{k-1} = 1/k, \quad k = 1, \ldots, p,$$

$$C(q): \quad \sum_{j=1}^{s} a_{ij} c_j^{k-1} = c_i^k/k, \quad i = 1, \ldots, s, k = 1, \ldots, q.$$

Solving a stiff ODE with the help of an RK method, the convergence order may drop down from p to q if $p > q$, see [7], i.e. the method has order reduction. The minimum of p and q is often called *stage order of the Runge–Kutta-method*.

Runge–Kutta methods have the advantage that they allow an easy implementation of an adaptive time step control. Consider a Runge–Kutta method of order $p \geq 2$. An adaptive time step control employs a second Runge–Kutta method, which has the coefficients a_{ij}, $\hat{b}_i$ and c_i, $i, j = 1, \ldots, s$, and order $p - 1$. The solution of the second method at time t_{m+1} is given by

$$\hat{\mathbf{u}}_{m+1} = \mathbf{u}_m + \sum_{i=1}^{s} \hat{b}_i \mathbf{k}_i.$$

Now, the next time step τ_{m+1} is proposed to be

$$\tau_{m+1} = \rho \frac{\tau_m^2}{\tau_{m-1}} \left(\frac{TOL \cdot r_m}{r_{m+1}^2}\right)^{1/p}, \tag{4}$$

where $\rho \in (0, 1]$ is a safety factor, $TOL > 0$ is a given tolerance and

$$r_{m+1} := \left\| \mathbf{u}_{m+1} - \hat{\mathbf{u}}_{m+1} \right\|. \tag{5}$$

This step size selection rule is called PI-controller and goes back to Gustafsson et al. [5]. For details on the numerical error and the implementation of automatic step length control we refer to [7,11].

2.2. Application to the example of Prothero–Robinson

In the following we consider the example of Prothero–Robinson, which is given by

$$\dot{u} = \lambda(u - \varphi(t)) + \dot{\varphi}(t), \quad u(0) = \varphi(0), \tag{6}$$

where $\lambda \ll 0$ and $\varphi(t)$ is a given function. The exact solution of Eq. (6) is given by $u(t) = \varphi(t)$. Next we apply the ESDIRK method (2)–(3) on the ODE (6). We obtain

$$k_i = \lambda \left(u_m + \tau \sum_{j=1}^{i} a_{ij} k_j - \varphi(t_m + c_i\tau) \right) + \dot{\varphi}(t_m + c_i\tau), \qquad i = 1, \ldots, s.$$

With the notations

$$\begin{aligned}
&\varphi_m^{(k)} := \varphi^{(k)}(t_m), \varphi_i^{(k)} := \varphi^{(k)}(t_m + c_i\tau), \quad i = 1, \ldots, s, k \geq 0, \\
&\tilde{\Phi}^{(k)} := (\varphi_2^{(k)}, \ldots, \varphi_s^{(k)})^\top, \quad \tilde{\mathbf{k}} := (k_2, \ldots, k_s)^\top, \quad \tilde{\mathbf{e}} := (1, \ldots, 1)^\top \in \mathbb{R}^{s-1}, \\
&\quad \tilde{\mathbf{c}} := (c_2, \ldots, c_s)^\top
\end{aligned}$$

it follows

$$\begin{aligned}
k_1 &= \lambda(u_m - \varphi_m) + \dot{\varphi}_m, \\
k_i &= \lambda \left(u_m + \tau \sum_{j=1}^{i} a_{ij} k_j - \varphi_i \right) + \dot{\varphi}_i \\
&= \lambda \left(u_m + \tau a_{i1} k_1 + \tau \sum_{j=2}^{i} a_{ij} k_j - \varphi_i \right) + \dot{\varphi}_i, \quad i = 2, \ldots, s.
\end{aligned}$$

Using the vector notation introduced above we obtain

$$\tilde{\mathbf{k}} = \lambda \left(u_m \tilde{\mathbf{e}} + \tau \mathbf{a}_1 k_1 + \tau \tilde{A} \tilde{\mathbf{k}} - \Phi \right) + \dot{\Phi},$$

where

$$\mathbf{a}_1 = (a_{21}, \ldots, a_{s1})^\top, \quad \tilde{A} = \begin{pmatrix} a_{21} & \ldots & a_{2s} \\ \vdots & & \vdots \\ a_{s1} & \ldots & a_{ss} \end{pmatrix}.$$

With $z = \lambda\tau$ it follows

$$\tilde{\mathbf{k}} = (I - z\tilde{A})^{-1} \left(\lambda \left(u_m \tilde{\mathbf{e}} + \tau \mathbf{a}_1 k_1 - \tilde{\Phi} \right) + \dot{\tilde{\Phi}} \right). \tag{7}$$

We insert Eq. (7) into (3) and get

$$\begin{aligned}
u_{m+1} &= u_m + \tau \sum_{i=1}^{s} b_i k_i = u_m + \tau b_1 k_1 + \tau \tilde{\mathbf{b}}^\top \tilde{\mathbf{k}} \\
&= u_m + \tau b_1 k_1 + \tau \tilde{\mathbf{b}}^\top (I - z\tilde{A})^{-1} \left[\lambda \left(u_m \tilde{\mathbf{e}} + \tau \mathbf{a}_1 k_1 - \tilde{\Phi} \right) + \dot{\tilde{\Phi}} \right] \\
&= u_m + \tau b_1 k_1 + z \tilde{\mathbf{b}}^\top (I - z\tilde{A})^{-1} \left(u_m \tilde{\mathbf{e}} + \tau \mathbf{a}_1 k_1 - \tilde{\Phi} \right) \\
&\quad + \tau \tilde{\mathbf{b}}^\top (I - z\tilde{A})^{-1} \dot{\tilde{\Phi}},
\end{aligned} \tag{8}$$

where $\tilde{\mathbf{b}} = (b_2, \ldots, b_s)^\top$.

2.3. The local error

The local error of an ESDIRK method applied to the Prothero–Robinson example is given by

$$\begin{aligned}\epsilon(t_{m+1}) &= u_{m+1} - \varphi(t_{m+1}) \\ &= u_m - \varphi_{m+1} + \tau b_1 k_1 + z\tilde{\mathbf{b}}^\top (I - z\tilde{A})^{-1}\left[u_m \tilde{\mathbf{e}} + \tau \mathbf{a}_1 k_1 - \tilde{\Phi}\right] \\ &\quad + \tau \tilde{\mathbf{b}}^\top (I - z\tilde{A})^{-1}\dot{\tilde{\Phi}}.\end{aligned}$$

In the case of the local error we assume that one time step is computed with the help of the exact solution at the previous time. Therefore, we replace u_m by the exact solution $\varphi(t_m)$. Next we write the local error in the form

$$\epsilon(t_{m+1}) = R(z)(u_m - \varphi(t_m)) + \delta_\tau(t_m), \tag{9}$$

where

$$\begin{aligned}\delta_\tau(t_{m+1}) &= \varphi_m - \varphi_{m+1} + \tau b_1 \dot{\varphi}_m + z\tilde{\mathbf{b}}^\top (I - z\tilde{A})^{-1}[\varphi_m \tilde{\mathbf{e}} + \tau \mathbf{a}_1 \dot{\varphi}_m - \tilde{\Phi}] \\ &\quad + \tau \tilde{\mathbf{b}}^\top (I - z\tilde{A})^{-1}\dot{\tilde{\Phi}}\end{aligned} \tag{10}$$

and $R(z)$ is the stability function of the ESDIRK method and given by

$$R(z) = 1 + zb_1 + z\tilde{\mathbf{b}}^\top (I - z\tilde{A})^{-1}(\tilde{\mathbf{e}} + z\mathbf{a}_1).$$

In the stiff case τ tends to 0 and z to infinity [7]. Therefore we expand the term $(I - z\tilde{A})^{-1}$ for large z as follows

$$(I - z\tilde{A})^{-1} = -(\tilde{A}z)^{-1} - (\tilde{A}z)^{-2} - \dots.$$

Since $(I - z\tilde{A})^{-1}$ is expanded in two variables τ and z we need the derivatives of $(I - z\tilde{A})^{-1}$. Let $\tilde{z} = 1/z$. Then

$$\left[\left(I - \frac{\tilde{A}}{\tilde{z}}\right)^{-1}\right]^{(k)} \to -k!\tilde{A}^{-k}, \qquad \text{for } \tilde{z} \to 0,$$

if $k \geq 1$. Then the Taylor expansion of $\delta_\tau(t_{m+1})$ reads as

$$\begin{aligned}\delta_\tau(t_{m+1}) &= -\sum_{k=1}^{p} \varphi_m^{(k)} \frac{\tau^k}{k!} + \tau b_1 \dot{\varphi}_m + \mathcal{O}(\tau^{p+1}) \\ &\quad - z\sum_{k=1}^{\infty} \tilde{\mathbf{b}}^\top \sum_{l=1}^{k} \binom{k}{l} l!\tilde{A}^{-l}\left[\varphi_m \tilde{\mathbf{e}}\delta_{k-l,0} + \mathbf{a}_1 \dot{\varphi}_m \delta_{k-l,1} - \tilde{\mathbf{c}}^{k-l}\varphi_m^{(k-l)}\right]\frac{\tau^{k-l}}{k!z^l} \\ &\quad - \tau\sum_{k=1}^{\infty} \tilde{\mathbf{b}}^\top \sum_{l=1}^{k} \binom{k}{l} l!\tilde{A}^{-l}\tilde{\mathbf{c}}^{k-l}\varphi_m^{(k-l+1)}\frac{\tau^{k-l}}{k!z^l}.\end{aligned}$$

In the second row the term $k = l$ vanishes. In the last term we sum from $l = 0$ to $k - 1$ and get

$$\begin{aligned}\delta_\tau(t_{m+1}) &= -\sum_{k=1}^{p} \varphi_m^{(k)} \frac{\tau^k}{k!} + \tau b_1 \dot{\varphi}_m \\ &\quad - \sum_{k=2}^{\infty} \tilde{\mathbf{b}}^\top \sum_{l=1}^{k-1} \tilde{A}^{-l}\left[\mathbf{a}_1 \dot{\varphi}_m \delta_{k-l,1} - \tilde{\mathbf{c}}^{k-l}\varphi_m^{(k-l)}\right]\frac{\tau^{k-l}}{(k-l)!z^{l-1}} \\ &\quad - \sum_{k=1}^{\infty} \tilde{\mathbf{b}}^\top \sum_{l=0}^{k-1} \tilde{A}^{-l}\tilde{\mathbf{c}}^{k-l-1}\varphi_m^{(k-l)}\frac{\tau^{k-l}}{(k-l-1)!z^{l+1}} + \mathcal{O}(\tau^{p+1}).\end{aligned}$$

Then we split the second sum and obtain

$$\begin{aligned}\delta_\tau(t_{m+1}) &= -\sum_{k=1}^{p} \varphi_m^{(k)} \frac{\tau^k}{k!} + \mathcal{O}(\tau^{p+1}) + \tau b_1 \dot{\varphi}_m \\ &\quad - \sum_{k=2}^{\infty} \tilde{\mathbf{b}}^\top \tilde{A}^{-1}\left[\mathbf{a}_1 \dot{\varphi}_m \delta_{k-1,1} - \tilde{\mathbf{c}}^{k-1}\varphi_m^{(k-1)}\right]\frac{\tau^{k-1}}{(k-1)!}\end{aligned}$$

$$- \sum_{k=3}^{\infty} \tilde{\mathbf{b}}^\top \sum_{l=2}^{k-1} \tilde{A}^{-l} \left[\mathbf{a}_1 \dot{\varphi}_m \delta_{k-l,1} - \tilde{\mathbf{c}}^{k-l} \varphi_m^{(k-l)} \right] \frac{\tau^{k-l}}{(k-l)! z^{l-1}}$$
$$- \sum_{k=1}^{\infty} \tilde{\mathbf{b}}^\top \sum_{l=0}^{k-1} \tilde{A}^{-l-1} \tilde{\mathbf{c}}^{k-l-1} \varphi_m^{(k-l)} \frac{\tau^{k-l}}{(k-l-1)! z^{l+1}}.$$

Next we sum in the second row from $k = 1$ and combine this sum to the sum in the first row. Moreover, we sum in the third and fourth row from $k = 2$. We obtain

$$\begin{aligned}
\delta_\tau(t_{m+1}) = {} & \tau \left[b_1 + \tilde{\mathbf{b}}^\top \tilde{A}^{-1} (\tilde{\mathbf{c}} - \mathbf{a}_1) - 1 \right] \dot{\varphi}_m \\
& + \sum_{k=2}^{p} \left[\tilde{\mathbf{b}}^\top \tilde{A}^{-1} \tilde{\mathbf{c}}^k - 1 \right] \varphi_m^{(k)} \frac{\tau^k}{k!} + \mathcal{O}(\tau^{p+1}) \\
& - \sum_{k=2}^{\infty} \tilde{\mathbf{b}}^\top \sum_{l=1}^{k-1} \tilde{A}^{-l-1} \left[\mathbf{a}_1 \dot{\varphi}_m \delta_{k-l,1} - \tilde{\mathbf{c}}^{k-l} \varphi_m^{(k-l)} \right] \frac{\tau^{k-l}}{(k-l)! z^{l}} \\
& - \sum_{k=2}^{\infty} \tilde{\mathbf{b}}^\top \sum_{l=1}^{k-1} \tilde{A}^{-l} \tilde{\mathbf{c}}^{k-l-1} \varphi_m^{(k-l)} \frac{\tau^{k-l}}{(k-l-1)! z^{l}}.
\end{aligned}$$

The last two sums can be combined, but we separate the case $l = k - 1$. Then we have

$$\begin{aligned}
\delta_\tau(t_{m+1}) = {} & \tau \left[b_1 + \tilde{\mathbf{b}}^\top \tilde{A}^{-1} (\tilde{\mathbf{c}} - \mathbf{a}_1) - 1 \right] \dot{\varphi}_m \\
& + \sum_{k=2}^{p} \left[\tilde{\mathbf{b}}^\top \tilde{A}^{-1} \tilde{\mathbf{c}}^k - 1 \right] \varphi_m^{(k)} \frac{\tau^k}{k!} + \mathcal{O}(\tau^{p+1}) \\
& + \sum_{k=2}^{\infty} \tilde{\mathbf{b}}^\top \left[\tilde{A}^{-k} (\tilde{\mathbf{c}} - \mathbf{a}_1) - \tilde{A}^{-k+1} \tilde{\mathbf{e}} \right] \dot{\varphi}_m \frac{\tau}{z^{k-1}} \\
& - \sum_{k=2}^{\infty} \tilde{\mathbf{b}}^\top \sum_{l=1}^{k-2} \tilde{A}^{-l} \left[\tilde{A}^{-1} \tilde{\mathbf{c}}^{k-l} - (k-l) \tilde{\mathbf{c}}^{k-l-1} \right] \varphi_m^{(k-l)} \frac{\tau^{k-l}}{(k-l)! z^{l}}.
\end{aligned}$$

2.4. Order conditions

From the representation of the local error we get new order conditions, which are given

$$b_1 + \tilde{\mathbf{b}}^\top \tilde{A}^{-1} (\tilde{\mathbf{c}} - \mathbf{a}_1) = 1, \tag{11}$$
$$\tilde{\mathbf{b}}^\top \tilde{A}^{-1} \tilde{\mathbf{c}}^k = 1, \quad k = 2, \ldots, p, \tag{12}$$
$$\tilde{\mathbf{b}}^\top \left[\tilde{A}^{-k} (\tilde{\mathbf{c}} - \mathbf{a}_1) - \tilde{A}^{-k+1} \tilde{\mathbf{e}} \right] = 0, \quad k = 2, \ldots, p, \tag{13}$$
$$\tilde{\mathbf{b}}^\top \tilde{A}^{-l} \left[\tilde{A}^{-1} \tilde{\mathbf{c}}^{k-l} - (k-l) \tilde{\mathbf{c}}^{k-l-1} \right] = 0 \tag{14}$$

for $k = 2, \ldots, \infty$ and $l = \max\{1, k-p\}, \ldots, k-2$.

Theorem 1.

- *If the ESDIRK method (2)–(3) is stiffly accurate the conditions (11) and (12) are automatically satisfied.*
- *Let the ESDIRK method (2)–(3) be consistent and the simplifying condition $C(1)$ valid. Then conditions (11) and (13) are fulfilled.*
- *If the ESDIRK method (2)–(3) satisfies the simplifying conditions $B(q)$ and $C(q)$, Eq. (12) is automatically valid $k = q$.*

Proof.

- First we consider condition (11) and write it component-wise as

$$b_1 + \sum_{i,j=2}^{s} \tilde{b}_i \omega_{ij} (c_j - a_{j1}) = 1,$$

where the entries of $\tilde{A}^{-1}$ are denoted by ω_{ij}. Since $\tilde{b}_i = \tilde{a}_{si}$ holds for a stiffly accurate ESDIRK method we have

$$\sum_{i=2}^{s} \tilde{b}_i \omega_{ij} = \sum_{i=2}^{s} \tilde{a}_{si} \omega_{ij} = \delta_{sj}$$

and

$$b_1 + \sum_{i,j=2}^{s} \tilde{b}_i \omega_{ij} (c_j - a_{j1}) = b_1 + \sum_{j=2}^{s} \delta_{sj} (c_j - a_{j1}) = b_1 + c_s - a_{s1} = 1.$$

Eq. (12) can be proven in an analogous way.

- The simplifying condition $C(1)$ can be written as

$$\mathbf{a}_1 + \tilde{A}\tilde{\mathbf{e}} = \tilde{\mathbf{c}}. \tag{15}$$

Next we insert Eq. (15) into condition (11) and obtain

$$b_1 + \tilde{\mathbf{b}}^\top \tilde{A}^{-1} (\tilde{\mathbf{c}} - \mathbf{a}_1) = b_1 + \tilde{\mathbf{b}}^\top \tilde{A}^{-1} (\mathbf{a}_1 + \tilde{A}\tilde{\mathbf{e}} - \mathbf{a}_1) = b_1 + \tilde{\mathbf{b}}^\top \mathbf{e} = 1.$$

Under these assumptions condition (13) is fulfilled, too, because

$$\tilde{\mathbf{b}}^\top \left[\tilde{A}^{-k} (\tilde{\mathbf{c}} - \mathbf{a}_1) - \tilde{A}^{-k+1} \tilde{\mathbf{e}} \right] = \tilde{\mathbf{b}}^\top \left[\tilde{A}^{-k} (\mathbf{a}_1 + \tilde{A}\tilde{\mathbf{e}} - \mathbf{a}_1) - \tilde{A}^{-k+1} \tilde{\mathbf{e}} \right] = 0.$$

- The simplifying condition $C(q)$ can be written as $q\tilde{A}\tilde{\mathbf{c}}^{q-1} = \tilde{\mathbf{c}}^q$. It follows with the simplifying condition $B(q)$

$$\tilde{\mathbf{b}}^\top \tilde{A}^{-1} \tilde{\mathbf{c}}^q = q\tilde{\mathbf{b}}^\top \tilde{A}^{-1} \tilde{A}\tilde{\mathbf{c}}^{q-1} = q\tilde{\mathbf{b}}^\top \tilde{\mathbf{c}}^{q-1} = 1. \qquad \square$$

With the help of this theorem we can define B_{PR}-consistency.

Definition 1. A consistent ESDIRK method is called *B_{PR}-consistent of order $\bar{q}$* if the simplifying condition $C(1)$, condition (12) for $k = 2, \ldots, \bar{q}$ and (14) for $k = 2, \ldots, \infty$ and $l = \max\{1, k - \bar{q}\}, \ldots, k - 2$ are satisfied.

The simplifying condition $C(q)$ is directly related to B_{PR}-consistency of order q.

Theorem 2. *If the ESDIRK method (2)–(3) satisfies the simplifying condition $C(2)$, condition (14) is satisfied for $l = k - 2$, $k = 3, 4, \ldots$.*

Proof. The simplifying condition $C(2)$ is given by $A\mathbf{c} = \mathbf{c}^2/2$ or component-wise by

$$\sum_{j=1}^{s} a_{ij} c_j = \frac{c_i^2}{2}.$$

Since $c_1 = 0$ we can sum from 2 to s and get $\tilde{A}\tilde{\mathbf{c}} = \tilde{\mathbf{c}}^2/2$. For $l = k - 2$ condition (14) is given by

$$\tilde{\mathbf{b}}^\top \tilde{A}^{2-k} \left[\tilde{A}^{-1} \tilde{\mathbf{c}}^2 - 2\tilde{\mathbf{c}} \right] = 0.$$

Inserting $C(2)$ yields

$$\tilde{\mathbf{b}}^\top \tilde{A}^{2-k} \left[2\tilde{A}^{-1} \tilde{A}\tilde{\mathbf{c}} - 2\tilde{\mathbf{c}} \right] = 0$$

and everything is proven. $\square$

2.5. The global error

The global error can be obtained if the representation of the local error, i.e. Eq. (9) is applied several times. We have

$$\begin{aligned}
\epsilon(t_{m+1}) &= R(z)(u_m - \varphi(t_m)) + \delta_\tau(t_m) \\
&= R(z)^2 (u_{m-1} - \varphi(t_{m-1})) + R(z)\delta_\tau(t_{m-1}) + \delta_\tau(t_m) = \cdots = \\
&= R(z)^{m+1} (u_0 - \varphi(t_0)) + \sum_{j=0}^{m} R(z)^{m-j} \delta_\tau(t_j).
\end{aligned}$$

The first term vanishes, since the initial condition is valid. Next we define convergence in the stiff case as follows.

Definition 2. An ESDIRK method is called B_{PR}-convergent of order q if the global error satisfies

$$\epsilon(t_m) \leq C_1 \tau^q + C_2 \frac{\tau^{q+1}}{|z|},$$

where C_1 and C_2 are non-negative constants which are independent of the step size τ and the stiffness λ.

Next we show that B_{PR}-consistency and A-stability imply B_{PR}-convergence or, to be more precise, that B_{PR}-consistency of order q and A-stability imply B_{PR}-convergence of order $q-1$. For many ESDIRK methods it can be shown that they converge with order q, too, if certain assumptions are fulfilled, as the following theorem tells us.

Theorem 3. *Consider an A-stable ESDIRK method with $R(\infty) \leq 1$. Assume that the local error can be written in form (9) and that the ESDIRK method is B_{PR}-consistent of order $\bar{q}+1$.*

- *Then the ESDIRK method is B_{PR}-convergent of order $\bar{q}$.*
- *If $R(\infty) < 0$ then for constant step sizes τ the ESDIRK method is B_{PR}-convergent of order $\bar{q}+1$.*
- *If $R(\infty) = 0$ the ESDIRK method is B_{PR}-convergent of order $\bar{q}+1$.*

Proof. For the proof we refer to [7] or [16]. □

3. New ESDIRK methods

In this section we develop new stiffly accurate ESDIRK methods which satisfy the new order conditions (12) and (14). Moreover, adaptivity with an embedded method should be possible. Since Theorem 1 is valid for the embedded methods, too, all our embedded methods satisfy conditions (11) and (13) automatically.

3.1. The ESDIRK53PR method

First we create a stiffly accurate ESDIRK method of order 3 with 5 internal stages. The method should satisfy the simplifying conditions $B(1)$, $B(2)$, $B(3)$, $C(1)$ and $C(2)$. Also, condition (14) should be valid for $k=4$ and $l=1$, $k=5$ and $l=2$. Moreover, the method and its embedded method should be L-stable. If we use all the simple conditions such as $C(1)$ we have 13 degrees of freedom and 10 equations. The free coefficients are $c_2 = 5/9$, $c_4 = 9/10$ and $\hat{b}_4 = 1/2$. The remaining coefficients can be computed with the help of a computer algebra tool. The coefficients can be found in Table A.3 in Appendix A.

3.2. The ESDIRK63PR method

Next we want to improve the ESDIRK53PR method in such a way that the embedded method is stiffly accurate, too. We need 6 internal stages, and therefore we have more coefficients than in the previous case. We want to fulfil condition (14) for $k=4$ and $l=1$, $k=5$ and $l=2$, $k=6$ and $l=3$, $k=5$ and $l=1$. Then we have 11 equations and 13 variables. The variable c_2 is set to 5/6 and c_4 to 3/10. Again, the coefficients are determined with the help of a computer algebra tool and are given in Table A.4 in Appendix A.

3.3. The ESDIRK74PR method

Next we want to find a 4th order ESDIRK method which satisfies the new order conditions. Therefore, we need 7 internal stages. In this case the simplifying condition $B(4)$ should be fulfilled, too. Moreover, condition (14) should be valid for $k=4$ and $l=1$, $k=5$ and $l=2$, $k=6$ and $l=3$, $k=5$ and $l=1$, $k=6$ and $l=2$. The free variables are chosen in the following way: $c_2 = 1/3$, $c_3 = 1/6$, $c_4 = 2/3$, $c_5 = 3/4$, $c_6 = 6/7$, $\hat{b}_2 = 1/10$, $\hat{b}_4 = 0$, and $a_{65} = 1/10$. Table A.5 in Appendix A presents the coefficients of the method.

3.4. Comparison of methods

In this section we compare different ESDIRK methods. It is interesting to know which order conditions are satisfied by the different methods. In the literature many ESDIRK methods can be found. The following list of ESDIRK methods is of course not complete.

First we mention ESDIRK methods of order 3 and 4, which can be found in the paper of Kennedy and Carpenter [9]. These methods have 4 and 6 internal stages and are denoted in this paper by ESDIRK3 and ESDIRK4. In [10] several ESDIRK methods of order 3, 4 and 5 were created. In newer papers ESDIRK methods are developed which satisfy order conditions for DAEs of index 2 or higher. One example is in the paper of Williams et al. [20] with the method ESDIRK32 which has 4

Table 1
Order conditions satisfied by the selected ESDIRK methods.

Name	s	$k=4$ $l=1$	5 2	6 3	5 1	6 2	Reference
ESDIRK32	4	x	–	–	–	–	[20]
ESDIRK3	4	–	–	–	–	–	[9]
ESDIRK32a	4	–	–	–	–	–	[10]
ESDIRK53PR	6	x	x	–	–	–	Section 3.1
ESDIRK63PR	6	x	x	x	x	–	Section 3.2
ESDIRK4	6	–	–	–	–	–	[9]
ESDIRK43a	5	–	–	–	–	–	[10]
ESDIRK43b	5	–	–	–	–	–	[10]
Skvortsov4–3	6	x	x	–	–	–	[18]
Skvortsov4–4	6	x	x	–	–	–	[18]
Skvortsov4–5	8	x	–	–	x	–	[18]
ESDIRK74PR	7	x	x	x	x	x	Section 3.3

internal stages and order 3. In [18] Skvortsov considers methods of order 4, which can be used for the solution of index-2 and index-3 DAEs.

Order conditions for index-2 DAEs were derived in the book of Hairer, Lubich and Roche [6], simplified by Higueras [8], and given by [20]

$$\tilde{\mathbf{b}}^\top \tilde{A}^{-1}\tilde{\mathbf{c}}^k = 1, \qquad k \in \{1, 2, 3\}, \tag{16}$$

$$\tilde{\mathbf{b}}^\top \tilde{A}^{-2}\tilde{\mathbf{c}}^k = k, \qquad k \in \{1, 2, 3\}. \tag{17}$$

Theorem 4. *Let a stiffly accurate ESDIRK method be given. If this method satisfies condition (14) for $l = 1$ and $k \in \{1, 2, 3\}$ the index-2 conditions (16) and (17) are automatically fulfilled.*

Proof. As we have shown before, condition (16) is fulfilled for all k if the ESDIRK method is stiffly accurate. Condition (14) reads in the case $l = 1$ and $k \in \mathbb{N}$ as

$$\tilde{\mathbf{b}}^\top \tilde{A}^{-2}\tilde{\mathbf{c}}^{k-1} = \tilde{\mathbf{b}}^\top \tilde{A}^{-1}(k-1)\tilde{\mathbf{c}}^{k-2}, \qquad k = 2, 3, \ldots.$$

Since our ESDIRK method is stiffly accurate, the right-hand side of the last equation is equal to $k - 1$. Next we make an index shift from $k - 1$ to k and get

$$\tilde{\mathbf{b}}^\top \tilde{A}^{-2}\tilde{\mathbf{c}}^k = k, \qquad k = 1, 2, \ldots.$$

This is the index-2 condition (17). □

In Table 1 it is shown which order conditions are satisfied by the considered ESDIRK methods. Condition (14) with $k - l = 2$ is not considered, since it is satisfied by the simplifying condition $C(2)$, which is fulfilled by all methods. The second column with $k = 4$ and $l = 1$ is a part of index-2 condition (17). It is satisfied by the index-2 methods, i.e. ESDIRK32 and the methods from Skvortsov. For the other columns one has to distinguish between third and fourth conditions. Most of the other conditions are only satisfied by our new methods, and only a few by the methods from Skvortsov. In the next section we show that it is important that all conditions are satisfied. Then it is possible to guarantee full order 3 or 4 in the case of the stiff Prothero–Robinson example. Otherwise the order of convergence can be rather poor for certain problems.

4. Numerical examples

In this section we apply our new ESDIRK methods to several test examples. In Table 2 we summarise the properties of the ESDIRK methods which are used for the computations of the numerical examples.

We first consider the example of Prothero and Robinson [14] to show that the new methods have full order for the stiff Prothero–Robinson example. For medium stiff problems the numerical results of the new methods are better, but the numerical order of convergence is rather poor. Then we consider a DAE of index 2. Finally, we see that our new methods can solve DAEs of index 2 effectively. Note that the methods from Skvortsov [18] are not equipped with an embedded method. Therefore, we use these methods only in cases where we solve our problems with equidistant time steps.

Table 2
Properties of the selected ESDIRK methods.

Name	s	p	q	$\|R(\infty)\|$	$\|\hat{R}(\infty)\|$	Reference
ESDIRK32	4	2	2	0.33	1	[20]
ESDIRK3	4	3	2	0	0.07	[9]
ESDIRK32a	4	3	2	0	0.96	[10]
ESDIRK53PR	5	3	2	0	0	Section 3.1
ESDIRK63PR	6	3	2	0	0	Section 3.2
ESDIRK4	6	4	2	0	0.07	[9]
ESDIRK43a	5	3	2	0	0.55	[10]
ESDIRK43b	5	3	2	0.72	0	[10]
Skvortsov4–3	6	4	2	0	–	[18]
Skvortsov4–4	6	4	2	$L(89.95°)$	–	[18]
Skvortsov4–5	8	5	2	$L(87.7°)$	–	[18]
ESDIRK74PR	6	4	2	0	0	Section 3.3

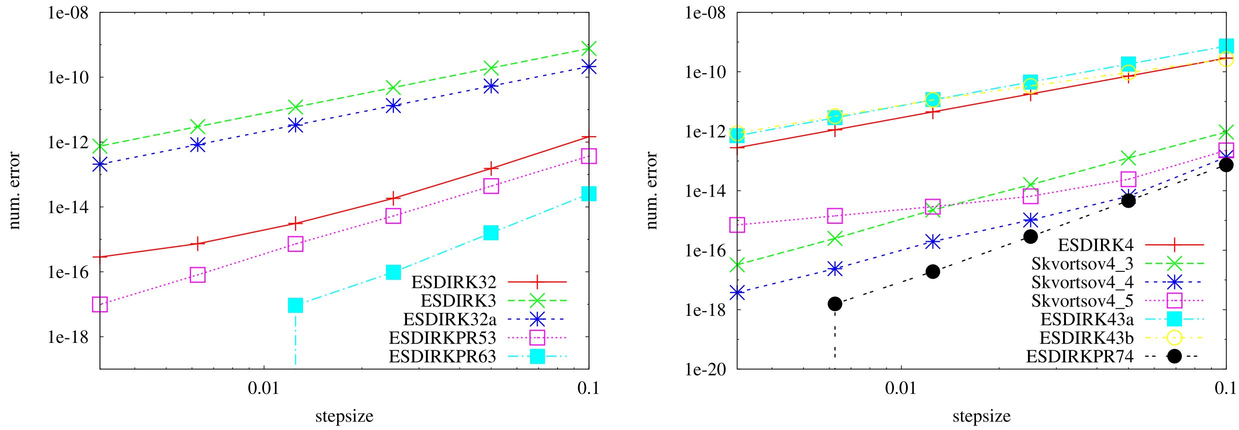

Fig. 1. τ versus error for (18) with $\lambda = -10^6$: third order methods (left) and fourth order methods (right).

4.1. Example of Prothero–Robinson

First we consider the well-known example from Prothero and Robinson, which is given by

$$\dot{u} = \lambda(u - \varphi(t)) + \dot{\varphi}(t), \quad u(0) = \varphi(0), \quad \lambda < 0. \tag{18}$$

The exact solution is given by $u(t) = \varphi(t)$ and the function $\varphi(t)$ by

$$\varphi(t) = \sin\left(\frac{\pi}{4} + t\right).$$

The ODE is solved (18) with equidistant step sizes $\tau = \frac{1}{10 \cdot 2^k}$, $k = 0, \dots, 5$ in the time interval $(0, 1/10]$. In Fig. 1 we present the numerical results for $\lambda = -10^6$. In the left figure we compare the third order methods. It can be observed that the ESDIRK32 method converges with a higher order than usual third order methods, which do not satisfy the index-2 conditions. The new methods ESDIRK53PR and ESDIRK63PR are the best methods in this case. All other methods converge with order 2.

The fourth order methods are compared in the right part of Fig. 1. Methods which are not designed for index-2 DAE converge with order 2 in the stiff case. An improvement can be observed if the methods are designed for index-2. The highest numerical order of convergence can be observed for our new method ESDIRK74PR. We get a different impression of the methods if we solve the example of Prothero and Robinson for $\lambda = -10^3$ (see Fig. 2). Let us first consider the 3rd order method, i.e. the left part of Fig. 2. ESDIRK3 and ESDIRK32a converge with order 2 for larger τ. For smaller step sizes the order of convergence increases. If the step size τ is large the ESDIRK32 method is much better than the ESDIRK3 and ESDIRK32a methods. For smaller step sizes the numerical results of these three methods are more or less the same. The ESDIRK53PR and ESDIRK63PR methods show a poor convergence behaviour for larger τ. In these cases the error terms τ^3/z^l dominate. For the ESDIRK53PR method we have the error term $\mathcal{O}(z^3/z^3)$, whereas for the ESDIRK63PR the error term is given by $\mathcal{O}(z^3/z^4)$. But these errors terms are small and therefore the numerical errors of the ESDIRK53PR and ESDIRK63PR methods are smaller than all the other numerical errors.

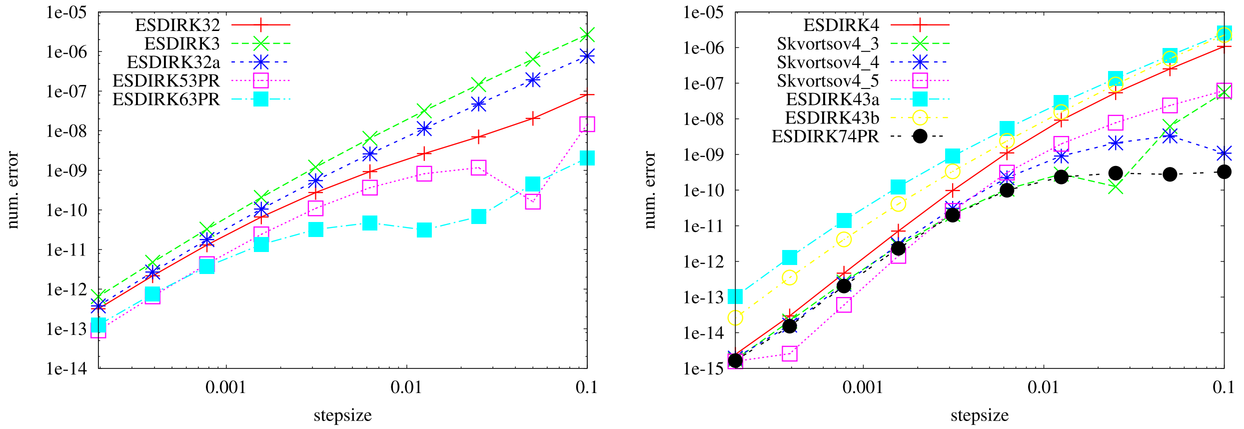

Fig. 2. τ versus error for (18) with $\lambda = -10^3$: third order methods (left) and fourth order methods (right).

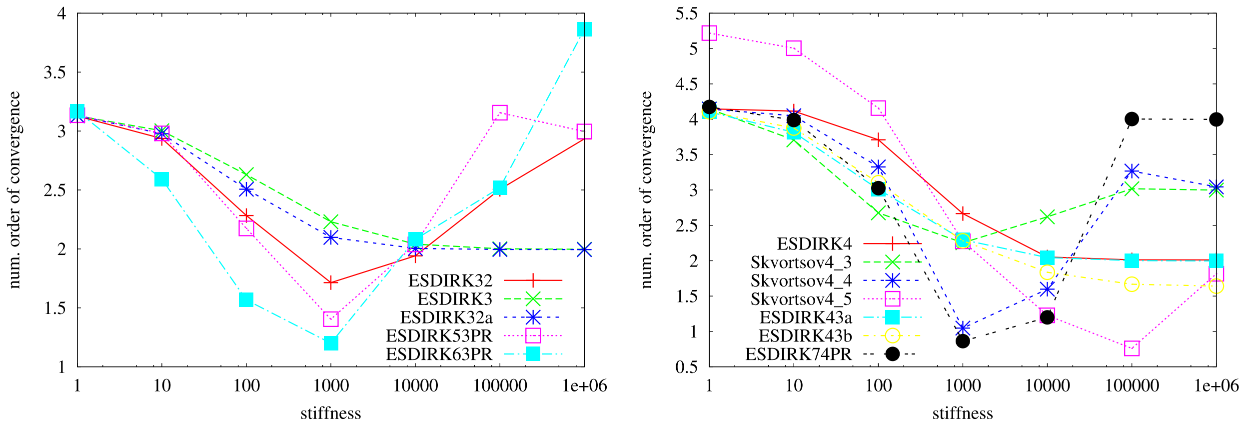

Fig. 3. Stiffness versus numerical order of convergence: third order methods (left) and fourth order methods (right).

In the case of the 4th order methods a similar observation can be made. Again, the ESDIRK4, ESDIRK43a and ESDIRK43b methods have the largest numerical error and order 2 for large step sizes. All the other methods have a poor convergence behaviour for larger step sizes. For example, the Skvortsov4–4 and the ESDIRK74PR method do not convergence at all for $\tau \in [1/100, 1/10]$, since the errors terms are given $\mathcal{O}(z^3/z^3)$ and $\mathcal{O}(z^3/z^4) + \mathcal{O}(z^4/z^3)$. But as in the case of the 3rd order methods the methods from Skvortsov and from this paper compute better results than the ESDIRK4, ESDIRK43a and ESDIRK43b methods. In Fig. 3 the stiffness $|\lambda|$ is plotted against the numerical order of convergence. We solve the Prothero–Robinson problem in the time interval $(0, 2]$, and as step size we use $\tau = 0.1 \cdot 2^{-l}$, where $l = 0, \ldots, 5$. From the discrete l_2-error the numerical order of convergence is computed. The medium numerical order of convergence is plotted in Fig. 3. It can be observed that the ESDIRK methods which do not satisfy any further order condition converge with order p in the non-stiff case, and with order 2 in the stiff case. Since none of the considered methods is B_{PR}-consistent of order 3 the methods from Skvortsov and from this paper show an interesting convergence behaviour. For medium stiff problems the convergence order decreases to 1, since the numerical error is dominated by factors of the form τ^4/z^l.

4.2. An index-2 DAE

Next we consider the differential-algebraic equation (see [12], [4, Example 10] or [7, page 461]).

$$\begin{cases} \dot{u}_1 - u_3\dot{u}_2 + u_2\dot{u}_3 = 0 \\ u_2 = \epsilon \sin(\omega t) \\ u_3 = \epsilon \cos(\omega t) \\ u_1(0) = 0 \end{cases} . \tag{19}$$

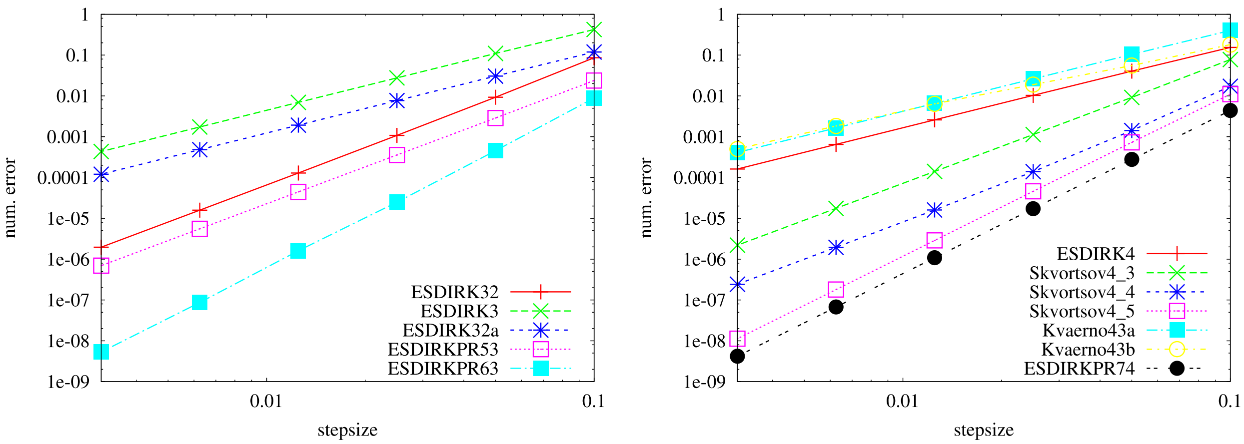

Fig. 4. The solution of the index-2 DAE with constant step sizes: third order methods (left) and fourth order methods (right).

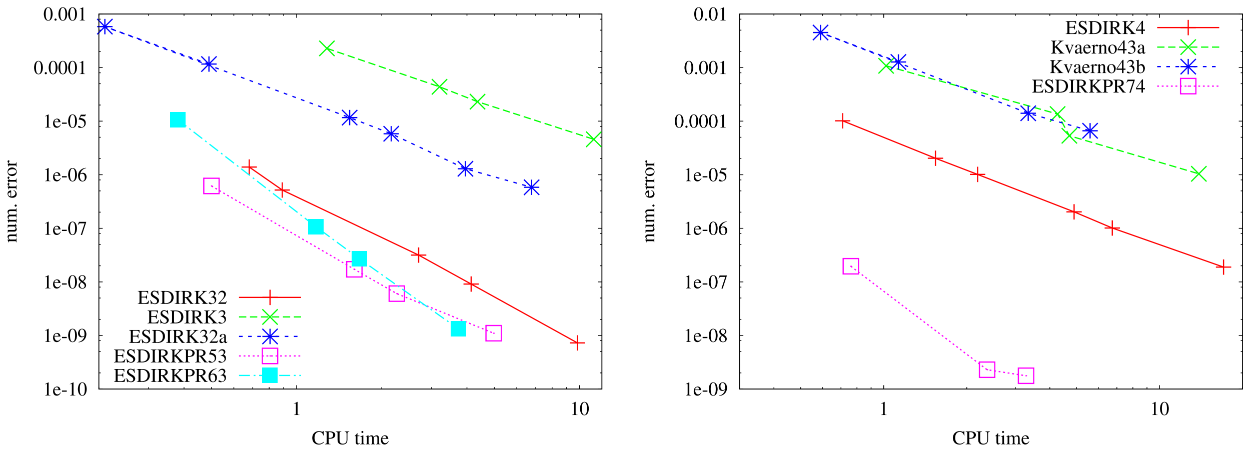

Fig. 5. The solution of the index-2 DAE with adaptive timestep control: third order methods (left) and fourth order methods (right).

This problem has differentiation index 1, but perturbation index 2 [7, page 461]. Numerical results applied on this problem should be designed in such a way that order conditions for index-2 DAEs are satisfied (see [17]). The DAE (19) can be solved numerically by introducing new variables $z_i := \dot{u}_i$. With this setting it is possible to rewrite problem (19) in the form

$$\dot{\mathbf{u}} = \mathbf{z}, \qquad 0 = \mathbf{F}(t, \mathbf{u}, \mathbf{z}).$$

For our numerical experiments we chose $\epsilon = 1$ and $\omega = 25$. First we solve this problem in the time interval $[0, 1/10]$ with equidistant time steps $\tau = 0.1 \cdot (1/2)^k$, where $k = 0, \ldots, 5$ and present the numerical results in Fig. 4. We get similar results as in the case of the Prothero–Robinson example. Methods such as ESDIRK32a, -43a and so on, which are not designed for index-2 DAEs, give the most inaccurate results. For the third order methods our new methods ESDIRK53PR and ESDIRK63PR are better than the ESDIRK32 method. In the case of the fourth order methods our new scheme ESDIRK74PR gives similar results as the Skvortsov4–5 method, but this method has no embedded method.

Next we solve the index-2 DAE (19) with an adaptive time step control time interval $[0, 50]$, where we set $\omega = 10$. The numerical results are shown in Fig. 5. In the left part we have the third order methods, which behave similarly to the previous simulation with constant step sizes. Again, we have the situation that the index-2 methods perform better than the others. Moreover, the new methods ESDIRK53PR and ESDIRK63PR are more effective than ESDIRK32. The same situation can be observed for the fourth order methods. Here the new ESDIRK74PR method is the most effective one.

5. Summary and outlook

In this paper we have considered ESDIRK methods and analysed the example of Prothero and Robinson. We showed that further order conditions must be satisfied if we want to have full order convergence. Therefore, we created new third and fourth order ESDIRK methods, which in our test examples perform much better than the usual ESDIRK methods.

Appendix A. Coefficients of methods

Table A.3
Set of coefficients for the ESDIRK53PR method.

$a_{21} =$ 2.777777777777778e−01	$a_{22} =$ 2.777777777777778e−01
$a_{31} =$ 3.456552483519272e−01	$a_{32} =$ 1.681740315717733e−01
$a_{33} =$ 2.777777777777778e−01	$a_{41} =$ 3.965643047257401e−01
$a_{42} =$ 1.001154404932533e−01	$a_{43} =$ 1.255424770032288e−01
$a_{44} =$ 2.777777777777778e−01	$a_{51} =$ 2.481479828780141e−01
$a_{52} =$ 2.139473588935955e−01	$a_{53} =$ 1.206274239267400e+00
$a_{54} =$ −9.461473588167871e−01	$a_{55} =$ 2.777777777777778e−01
$b_1 =$ 2.481479828780141e−01	$\hat{b}_1 =$ 4.445537532713554e−01
$b_2 =$ 2.139473588935955e−01	$\hat{b}_2 =$ −1.065203443758999e−01
$b_3 =$ 1.206274239267400e+00	$\hat{b}_3 =$ 2.533129069755295e−01
$b_4 =$ −9.461473588167871e−01	$\hat{b}_4 =$ 5.000000000000000e−01
$b_5 =$ 2.777777777777778e−01	$\hat{b}_5 =$ −9.134631587098500e−02

Table A.4
Set of coefficients for the ESDIRK63PR method.

$a_{21} =$ 4.166666666666667e−01	$a_{22} =$ 4.166666666666667e−01
$a_{31} =$ 3.640473915723038e−01	$a_{32} =$ −4.189886135331312e−02
$a_{33} =$ 4.166666666666667e−01	$a_{41} =$ −2.894969214392781e+00
$a_{42} =$ −2.256341718064659e+01	$a_{43} =$ 2.534171972837271e+01
$a_{44} =$ 4.166666666666667e−01	$a_{51} =$ 2.309551022782098e−01
$a_{52} =$ −1.849667242832423e+00	$a_{53} =$ 2.197073089164931e+00
$a_{54} =$ 4.972384722615363e−03	$a_{55} =$ 4.166666666666667e−01
$a_{61} =$ 3.054968378466108e−01	$a_{62} =$ 4.057983152922798e+00
$a_{63} =$ −2.202162095667910e+00	$a_{64} =$ 1.333484429273537e−01
$a_{65} =$ −1.711333004695519e+00	$a_{66} =$ 4.166666666666667e−01
$b_1 =$ 3.054968378466108e−01	$\hat{b}_1 =$ 2.309551022782098e−01
$b_2 =$ 4.057983152922798e+00	$\hat{b}_2 =$ −1.849667242832423e+00
$b_3 =$ −2.202162095667910e+00	$\hat{b}_3 =$ 2.197073089164931e+00
$b_4 =$ 1.333484429273537e−01	$\hat{b}_4 =$ 4.972384722615363e−03
$b_5 =$ −1.711333004695519e+00	$\hat{b}_5 =$ 4.166666666666667e−01
$b_6 =$ 4.166666666666667e−01	$\hat{b}_6 =$ 0.000000000000000e+00

Table A.5
Set of coefficients for the ESDIRK74PR method.

$a_{21} =$ 1.666666666666667e−01	$a_{22} =$ 1.666666666666667e−01
$a_{31} =$ 4.166666666666666e−02	$a_{32} =$ −4.166666666666666e−02
$a_{33} =$ 1.666666666666667e−01	$a_{41} =$ −1.500000000000000e+00
$a_{42} =$ −1.333333333333333e+00	$a_{43} =$ 3.333333333333333e+00
$a_{44} =$ 1.666666666666667e−01	$a_{51} =$ −1.580729166666667e+00
$a_{52} =$ −1.349609375000000e+00	$a_{53} =$ 3.472656250000000e+00
$a_{54} =$ 4.101562500000000e−02	$a_{55} =$ 1.666666666666667e−01
$a_{61} =$ −2.005366150605651e+00	$a_{62} =$ −1.768688648609954e+00
$a_{63} =$ 4.341269295345690e+00	$a_{64} =$ 2.326169434610579e−02
$a_{65} =$ 1.000000000000000e−01	$a_{66} =$ 1.666666666666667e−01
$a_{71} =$ 1.684854267805816e−01	$a_{72} =$ 7.501080898831836e−01
$a_{73} =$ −2.255843889686931e−01	$a_{74} =$ −9.134421504267402e−01
$a_{75} =$ 1.618140253772232e+00	$a_{76} =$ −5.643738977072310e−01
$a_{77} =$ 1.666666666666667e−01	
$b_1 =$ 1.684854267805816e−01	$\hat{b}_1 =$ −3.930182461751728e−01
$b_2 =$ 7.501080898831836e−01	$\hat{b}_2 =$ 1.000000000000000e−01
$b_3 =$ −2.255843889686931e−01	$\hat{b}_3 =$ 9.916346405575472e−01
$b_4 =$ −9.134421504267402e−01	$\hat{b}_4 =$ 0.000000000000000e+00
$b_5 =$ 1.618140253772232e+00	$\hat{b}_5 =$ −2.511232158528943e−01
$b_6 =$ −5.643738977072310e−01	$\hat{b}_6 =$ 4.393912810497486e−01
$b_7 =$ 1.666666666666667e−01	$\hat{b}_7 =$ 1.131155404207712e−01

References

[1] J.W. Butcher, On Runge–Kutta processes of high order, J. Aust. Math. Soc. 4 (1964) 179–194.
[2] F. Cameron, A class of low order DIRK methods for a class of DAEs, Appl. Numer. Math. 31 (1) (1999) 1–16.
[3] F. Cameron, M. Palmroth, R. Piché, Quasi stage order conditions for SDIRK methods, Appl. Numer. Math. 42 (1–3) (2002) 61–75.
[4] S. Campbell, W. Gear, The index of general nonlinear DAEs, Numer. Math. 72 (2) (1995) 173–196.
[5] K. Gustafsson, M. Lundh, G. Söderlind, A PI stepsize control for the numerical solution of ordinary differential equations, BIT Numer. Math. 28 (2) (1988) 270–287.
[6] E. Hairer, C. Lubich, M. Roche, The Numerical Solution of Differential-Algebraic Systems by Runge–Kutta Methods, Springer-Verlag, Berlin, 1989.
[7] E. Hairer, G. Wanner, Solving Ordinary Differential Equations. II: Stiff and Differential-Algebraic Problems, Springer Series in Computational Mathematics, vol. 14, Springer-Verlag, Berlin, 1996.
[8] I. Higueras, On simplifying assumptions of Runge–Kutta methods for index 2 differential algebraic problems, Computing 54 (2) (1995) 185–190.
[9] C.A. Kennedy, M.H. Carpenter, Additive Runge–Kutta schemes for convection–diffusion-reaction equations, Appl. Numer. Math. 44 (2003) 139–181.
[10] A. Kværnø, Singly diagonally implicit Runge–Kutta methods with an explicit first stage, BIT Numer. Math. 44 (3) (2004) 489–502.
[11] J. Lang, Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems, Lecture Notes in Computational Science and Engineering, vol. 16, Springer-Verlag, Berlin, 2001.
[12] C. Lubich, Linearly implicit extrapolation methods for differential-algebraic systems, Numer. Math. 55 (1989) 197–211.
[13] A. Ostermann, M. Roche, Runge–Kutta methods for partial differential equations and fractional orders of convergence, Math. Comput. 59 (200) (1992) 403–420.
[14] A. Prothero, A. Robinson, On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations, Math. Comput. 28 (1974) 145–162.
[15] J. Rang, An analysis of the Prothero–Robinson example for constructing new DIRK and ROW methods, J. Comput. Appl. Math. 262 (2014) 105–114.
[16] J. Rang, The Prothero and Robinson example: convergence studies for Runge–Kutta and Rosenbrock–Wanner methods, Informatik-Bericht 2014-08, TU Braunschweig, Braunschweig, 2014, http://www.digibib.tu-bs.de/?docid=00057194.
[17] J. Rang, L. Angermann, Remarks on the differentiation index and on the perturbation index of non-linear DAEs, Mathematik-Bericht 2005/3, Institut für Mathematik, TU Clausthal (2005).
[18] L. Skvortsov, Diagonally implicit Runge–Kutta methods for differential algebraic equations of indices two and three, Comput. Math. Math. Phys. 50 (6) (2010) 993–1005.
[19] K. Strehmel, R. Weiner, Linear-implizite Runge–Kutta-Methoden und Ihre Anwendung, Teubner-Texte zur Mathematik, vol. 127, Teubner, Stuttgart, 1992.
[20] R. Williams, K. Burrage, I. Cameron, M. Kerr, A four-stage index 2 diagonally implicit Runge–Kutta method, Appl. Numer. Math. 40 (3) (2002) 415–432.