ELSEVIER

Journal of Computational and Applied Mathematics 101 (1999) 105–116

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Continuous variable stepsize explicit pseudo two-step RK methods[1]

Nguyen Huu Cong

Faculty of Mathematics, Mechanics and Informatics, Hanoi University of Sciences, 334 Nguyen Trai, Thanh Xuan, Hanoi, Vietnam

Abstract

The aim of this paper is to apply a class of constant stepsize explicit pseudo two-step Runge–Kutta methods of arbitrarily high order to nonstiff problems for systems of first-order differential equations with variable stepsize strategy. Embedded formulas are provided for giving a cheap error estimate used in stepsize control. Continuous approximation formulas are also considered for use in an eventual implementation of the methods with dense output. By a few widely used test problems, we compare the efficiency of two pseudo two-step Runge–Kutta methods of orders 5 and 8 with the codes DOPRI5, DOP853 and PIRK8. This comparison shows that in terms of f-evaluations on a parallel computer, these two pseudo two-step Runge–Kutta methods are a factor ranging from 3 to 8 cheaper than DOPRI5, DOP853 and PIRK8. Even in a sequential implementation mode, fifth-order new method beats DOPRI5 by a factor more than 1.5 with stringent error tolerances.

Keywords: Runge–Kutta methods; Two-step Runge–Kutta methods; Embedded and dense output formulas; Parallelism

1. Introduction

The arrival of parallel computers influences the development of methods for the numerical solution of a nonstiff initial value problem (IVP) for systems of first-order ordinary differential equations (ODEs)

$$y'(t)=f(t,y(t)), \quad y(t_0)=y_0, \quad y,f\in\mathbb{R}^d. \tag{1.1}$$

The most efficient numerical methods for solving this problem are the explicit Runge–Kutta methods (RK methods). In the literature, sequential explicit RK methods up to order 10 can be found in, e.g., [10–12]. In order to exploit the facilities of parallel computers, several classes of parallel explicit methods have been investigated in, e.g., [2, 4, 5, 7, 8, 13–15, 17–19]. A common challenge

[1] This work was partly supported by DAAD, N.R.P.F.S. and QG-96-02.

0377-0427/99/$ – see front matter
PII: S0377-0427(98)00199-X

in the latter mentioned works is to reduce, for a given order of accuracy, the required number of effective sequential f-evaluations per step, using parallel processors. In our previous work [6], we have considered a general class of explicit pseudo two-step RK methods (EPTRK methods) for solving problems of the form (1.1). A general s-stage (constant stepsize) EPTRK method based on an s-dimensional collocation vector $\boldsymbol{c}=(c_1,\ldots,c_s)^{\mathrm{T}}$ with distinct abscissas c_i has the form

$$\boldsymbol{Y}_n=\boldsymbol{e}\otimes y_n+h(A\otimes I)\boldsymbol{F}(t_{n-1}\boldsymbol{e}+h\boldsymbol{c},\boldsymbol{Y}_{n-1}), \tag{1.2a}$$

$$y_{n+1}=y_n+h(\boldsymbol{b}^{\mathrm{T}}\otimes I)\boldsymbol{F}(t_n\boldsymbol{e}+h\boldsymbol{c},\boldsymbol{Y}_n). \tag{1.2b}$$

This method has been specified by the tableau

$$\begin{array}{c|c|c} A & \boldsymbol{c} & O \\ \hline & y_{n+1} & \boldsymbol{b}^{\mathrm{T}} \end{array}$$

The (constant) $s\times s$ matrix A and s-dimensional vector $\boldsymbol{b}$ of the method parameters are given by (see [6, Section 2.1])

$$\begin{aligned} &A=PQ^{-1}, \quad P=(p_{ij})=\left(\frac{c_i^j}{j}\right), \quad Q=(q_{ij})=((c_i-1)^{j-1}),\\ &\boldsymbol{b}^{\mathrm{T}}=\boldsymbol{g}^{\mathrm{T}}R^{-1}, \quad \boldsymbol{g}=(g_i)=\left(\frac{1}{i}\right), \quad R=(r_{ij})=(c_i^{j-1}),\\ &i=1,\ldots,s, \quad j=1,\ldots,s. \end{aligned} \tag{1.3}$$

The method (1.2) is of order p and stage order q at least equal s for any collocation vector $\boldsymbol{c}$, it has the highest order $p=s+1$ if $\boldsymbol{c}$ satisfies the orthogonality relation (cf. [6, Theorem 2.2]). The number of f-evaluations per step equals s in a sequential implementation and equals 1 in a parallel implementation using s processors. This class of EPTRK methods implemented with constant stepsize was shown to be very efficient for the solution of problems with stringent accuracy demand (cf. [6, Section 3]).

In the present work, we equip the EPTRK methods with an ability of being able to change the stepsize. Since the EPTRK methods are of a two-step nature, we consider the method with (variable) parameters which are functions of stepsizes (see Section 2). For a practical error estimation used in a stepsize selection, an approach for constructing embedded formulas is discussed in Section 3. Section 4 is devoted to a continuous extension of EPTRK methods where a general explicit expression of dense output formulas is given. Notice that for EPTRK methods, embedded and dense output formulas are provided without additional f-evaluations. Finally, in Section 5, we present numerical results of the currently available codes DOPRI5, DOP853, PIRK8 and two comparable order EPTRK methods by applying them to the three widely used test examples, viz. two-body problem, Fehlberg problem, and Jacobian elliptic functions problem (cf., e.g., [12, p. 240; 2, 14, 16]) for a performance comparison of various methods.

2. Variable stepsize EPTRK methods

It is well known that an efficient integration method must be able to change stepsizes. Because EPTRK methods are of a two-step nature, there is an additional difficulty in using these methods

with variable stepsize mode. There exist in principle two approaches for overcoming this difficulty (cf., e.g., [12, p. 397; 3, p. 44]):

- interpolating past stage values,
- deriving methods with variable parameters.

The first approach using polynomial interpolation to reproduce the starting stage values for the new step involves with computational cost which increases as the dimension of the problem increases, while for the second approach, the computational cost is independent of the dimension of the problem. For this reason, the variable parameter approach is more feasible and robust. Thus, we consider the EPTRK method

$$\boldsymbol{Y}_n = \boldsymbol{e} \otimes y_n + h_n(A_n \otimes I)\boldsymbol{F}(t_{n-1}\boldsymbol{e} + h_{n-1}\boldsymbol{c}, \boldsymbol{Y}_{n-1}), \tag{2.1a}$$

$$y_{n+1} = y_n + h_n(\boldsymbol{b}^{\mathrm{T}} \otimes I)\boldsymbol{F}(t_n\boldsymbol{e} + h_n\boldsymbol{c}, \boldsymbol{Y}_n), \tag{2.1b}$$

with variable stepsize $h_n = t_{n+1} - t_n$ and variable parameter matrix A_n. The order and stage order of a variable stepsize EPTRK method is defined in the same way as in the case of constant stepsize EPTRK methods (cf. [6, Definition 2.1]). The matrix A_n in the method (2.1) can be determined by order conditions as a matrix function of the stepsize ratios. The s-order conditions can be derived by replacing $\boldsymbol{Y}_{n-1}$, y_n and $\boldsymbol{Y}_n$ in (2.1a) by the exact solution values $\boldsymbol{y}(t_{n-1}\boldsymbol{e} + h_{n-1}\boldsymbol{c})$, $\boldsymbol{y}(t_n)$ and $\boldsymbol{y}(t_n\boldsymbol{e} + h_n\boldsymbol{c})$, respectively, that is

$$\boldsymbol{y}(t_n\boldsymbol{e} + h_n\boldsymbol{c}) - \boldsymbol{e} \otimes \boldsymbol{y}(t_n) - h_n(A_n \otimes I)\boldsymbol{y}'(t_{n-1}\boldsymbol{e} + h_{n-1}\boldsymbol{c}) = \mathrm{O}(h_n^{s+1}). \tag{2.2}$$

Let us suppose that the stepsize ratio h_n/h_{n-1} is bounded from above (i.e., $h_n/h_{n-1} \leqslant \Omega$), then along the same lines of [6, Section 2.1], using Taylor expansions, we can expand the left-hand side of (2.2) in powers of h_n and obtain the order conditions for determining A_n given by

$$\boldsymbol{C}^{(j)} = \frac{1}{j!}\left[\left(\frac{h_n}{h_{n-1}}\right)^{j-1}\frac{\boldsymbol{c}^j}{j} - A_n(\boldsymbol{c}-\boldsymbol{e})^{j-1}\right] = \boldsymbol{0}, \quad j = 1,\ldots,s. \tag{2.3a}$$

Condition (2.3a) can be written in the form (cf. (1.3))

$$A_nQ - P\,\mathrm{diag}\left\{1, \frac{h_n}{h_{n-1}}, \ldots, \left(\frac{h_n}{h_{n-1}}\right)^{s-1}\right\} = \mathrm{O}, \tag{2.3b}$$

which yields the explicit expression of A_n defined as

$$A_n = P\,\mathrm{diag}\left\{1, \frac{h_n}{h_{n-1}}, \ldots, \left(\frac{h_n}{h_{n-1}}\right)^{s-1}\right\} Q^{-1}. \tag{2.3c}$$

The following lemma can easily be deduced from (2.3c)

Lemma 2.1. *For the variable stepsize EPTRK method* (2.1), *the variable parameter matrix* A_n *is uniformly bounded whenever the stepsize ratio* h_n/h_{n-1} *is bounded from above.*

For $h_n/h_{n-1} \leqslant \Omega$, the principal error vector $\boldsymbol{C}^{(s+1)}$ is also uniformly bounded. Consequently, similar to the order considerations for a general variable stepsize multistep method (cf., e.g., [12, p. 401]),

relations (2.3) imply that locally

$$\boldsymbol{Y}(t_n\boldsymbol{e}+h_n\boldsymbol{c})-\boldsymbol{Y}_n=\mathrm{O}(h_n^{s+1}).$$

Along the lines of the proof of Theorems 2.1 and 2.2 in [6], we have that if the function f is Lipschitz continuous and if the condition of Lemma 2.1 is satisfied then at t_{n+1}

$$y(t_{n+1})-y_{n+1}=\mathrm{O}(h_n^{p+1})+\mathrm{O}(h_n^{s+2}),$$

where p is the order of the associated constant stepsize EPTRK method. Hence, the order and stage order of the variable stepsize EPTRK method defined by (2.1) and (2.3c) is identical with those of the associated constant stepsize EPTRK method (see [6, Theorem 2.2], also Section 1). Thus we have

Theorem 2.2. *An s-stage variable stepsize EPTRK method defined by* (2.1) *with parameters vector* $\boldsymbol{b}$ *as defined in* (1.3) *and matrix* A_n *defined by* (2.3c) *is of order* $p=s$ *and of stage order* $q=s$ *for any collocation vector* $\boldsymbol{c}$ *with distinct abscissas* c_i *if* h_n/h_{n-1} *is bounded from above. It has order* $p=s+1$ *if in addition the orthogonality relation*

$$P_j(1)=0, \qquad P_j(x):=\int_0^x \xi^{j-1}\prod_{i=1}^{s}(\xi-c_i)\,\mathrm{d}\xi,$$

is satisfied for $j\geqslant 1$.

Remark 2.3. The condition $h_n/h_{n-1}\leqslant\Omega$ is a reasonable assumption for a numerical code.

Remark 2.4. Since zero-stability property of EPTRK methods is independent of the method parameters (see [6, Section 2.2]), the variable stepsize EPTRK methods are always stable.

3. Embedded EPTRK methods

With the aim to have a cheap error estimate for stepsize control in an implementation of EPTRK methods, parallelly with the pth-order method (2.1), we consider a second $\widehat{p}$th-order EPTRK method based on collocation vector $\widetilde{\boldsymbol{c}}=(\widetilde{c}_1,\ldots,\widetilde{c}_{\widetilde{s}})^{\mathrm{T}}$ of the form

$$\begin{aligned}\widetilde{\boldsymbol{Y}}_n &= \widetilde{\boldsymbol{e}}\otimes\widetilde{y}_n+h_n(\widetilde{A}\otimes I)\boldsymbol{F}(t_{n-1}\widetilde{\boldsymbol{e}}+h_{n-1}\widetilde{\boldsymbol{c}},\widetilde{\boldsymbol{Y}}_{n-1}),\\ \widetilde{y}_{n+1} &= \widetilde{y}_n+h_n(\widetilde{\boldsymbol{b}}^{\mathrm{T}}\otimes I)\boldsymbol{F}(t_n\widetilde{\boldsymbol{e}}+h_n\widetilde{\boldsymbol{c}},\widetilde{\boldsymbol{Y}}_n),\end{aligned} \tag{3.1}$$

where, $p>\widehat{p}$, the vector $\widetilde{\boldsymbol{c}}$ is a subvector of the vector $\boldsymbol{c}$, i.e., $\{\widetilde{c}_1,\ldots,\widetilde{c}_{\widetilde{s}}\}\subset\{c_1,\ldots,c_s\}$. By introducing a new parameter vector $\widehat{\boldsymbol{b}}=(\widehat{b}_1,\ldots,\widehat{b}_s)^{\mathrm{T}}$ which is defined according to

$$\begin{aligned}&\text{if} \quad c_i=\widetilde{c}_j, \quad \text{then } \widehat{b}_i=\widetilde{b}_j, \quad j=1,\ldots,\widetilde{s},\\ &\qquad\qquad \text{else} \quad \widehat{b}_i=0, \quad i=1,\ldots,s,\end{aligned} \tag{3.2}$$

we obtain an embedded formula without additional f-evaluations given by

$$\widehat{y}_{n+1}=y_n+h_n(\widehat{\boldsymbol{b}}^{\mathrm{T}}\otimes I)\boldsymbol{F}(t_n\boldsymbol{e}+h_n\boldsymbol{c},\boldsymbol{Y}_n). \tag{3.3}$$

Theorem 3.1. *If the function f is Lipschitz continuous, then the numerical approximations at t_{n+1} defined by* (2.1b) *and by locally satisfy the order relation*

$$y_{n+1} - \hat{y}_{n+1} = O(h_n^{\hat{p}+1}). \tag{3.4}$$

Proof. As the EPTRK method (3.1) has order $\hat{p}$ less than order p of the EPTRK method (1.2), we may write

$$\begin{aligned} y_{n+1} - \hat{y}_{n+1} &= (y_{n+1} - \tilde{y}_{n+1}) + (\tilde{y}_{n+1} - \hat{y}_{n+1}) \\ &= O(h_n^{\hat{p}+1}) + (\tilde{y}_{n+1} - \hat{y}_{n+1}). \end{aligned} \tag{3.5a}$$

Since the function f is Lipschitz continuous, from the definition of the vector $\hat{\boldsymbol{b}}$ in (3.2) we have

$$\tilde{y}_{n+1} - \hat{y}_{n+1} = (\tilde{y}_n - y_n) + O(h_n^{\tilde{s}+2}). \tag{3.5b}$$

In view of the relations (3.5), Theorem 3.1 is proved. □

Thus, for a practical error estimation used in a stepsize selection we have the embedded EPTRK method given by (2.1a), (2.1b) and (3.3) which can be specified by the tableau

A_n	c	O
	y_{n+1}	$\boldsymbol{b}^{\mathrm{T}}$
	$\hat{y}_{n+1}$	$\hat{\boldsymbol{b}}^{\mathrm{T}}$

The local error estimate is then defined by (3.4). By this approach of constructing embedded EPTRK methods, there exist several embedded formulas for an EPTRK method.

4. Continuous EPTRK methods

A numerical method is inefficient, if the number of output points becomes very large (cf. [12, p. 188]). In the literature almost efficient embedded RK pairs have been provided with a dense output formula. In this section we also consider such a dense output formula for EPTRK methods. Since the EPTRK methods are of collocation nature, a continuous extension is very natural and straightforward. Thus, we consider a continuous extension of EPTRK method (2.1) defined by

$$\boldsymbol{Y}_n = \boldsymbol{e} \otimes y_n + h_n(A \otimes I)\boldsymbol{F}(t_{n-1}\boldsymbol{e} + h_{n-1}\boldsymbol{c}, \boldsymbol{Y}_{n-1}), \tag{4.1a}$$

$$y_{n+\xi} = y_n + h_n(\boldsymbol{b}^{\mathrm{T}}(\xi) \otimes I)\boldsymbol{F}(t_n\boldsymbol{e} + h_n\boldsymbol{c}, \boldsymbol{Y}_n), \tag{4.1b}$$

where $0 \leqslant \xi \leqslant 1$, $y_{n+\xi} \approx y(t_{n+\xi})$, with $t_{n+\xi} = t_n + \xi h_n$. Furthermore, $\boldsymbol{b}(\xi)$ satisfies the continuity conditions $\boldsymbol{b}(0) = \boldsymbol{0}$ and $\boldsymbol{b}(1) = \boldsymbol{b}$. The vector $\boldsymbol{b}(\xi)$ is a vector function of ξ and can be determined by order conditions. Along the same lines of Section 2 (see also [6, Section 2.1]), by using Taylor

expansions we obtain the s-order conditions for determining $\boldsymbol{b}^{\mathrm{T}}(\xi)$ in (4.1b)

$$D^{(j)} = \frac{1}{j!}\left[\frac{\xi^j}{j} - \boldsymbol{b}^{\mathrm{T}}(\xi)\boldsymbol{c}^{j-1}\right] = 0, \quad j = 1, \ldots, s. \tag{4.2a}$$

The order conditions (4.2a) can be seen to be of the form (cf. (1.3))

$$\boldsymbol{b}^{\mathrm{T}}(\xi)R - \boldsymbol{g}^{\mathrm{T}} \operatorname{diag}\{\xi, \xi^2, \ldots, \xi^s\} = \boldsymbol{0}. \tag{4.2b}$$

From (4.2b) the explicit expression of the vector function $\boldsymbol{b}(\xi)$ then comes out

$$\boldsymbol{b}^{\mathrm{T}}(\xi) = \boldsymbol{g}^{\mathrm{T}} \operatorname{diag}\{\xi, \xi^2, \ldots, \xi^s\} R^{-1}. \tag{4.2c}$$

The following theorem holds:

Theorem 4.1. *The EPTRK method defined by* (4.1) *and* (4.2c) *gives rise to a continuous EPTRK method of order s, i.e., for all* ξ: $0 \leqslant \xi \leqslant 1$ *we have*

$$y(t_n + \xi h_n) - y_{n+\xi} = O(h_n^{s+1}).$$

We end this section by giving an example of a continuous variable stepsize embedded EPTRK pair $p(\hat{p}) = 4(2)$ with dense output formula of order 3 given by the following tableau:

$$\begin{array}{ccc|c|ccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\gamma(2\gamma+3)}{24} & -\frac{\gamma(\gamma+3)}{6} & \frac{2\gamma^2+9\gamma+12}{24} & \frac{1}{2} & 0 & 0 & 0 \\
\frac{\gamma(4\gamma+3)}{6} & -\frac{2\gamma(2\gamma+3)}{3} & \frac{4\gamma^2+9\gamma+6}{6} & 1 & 0 & 0 & 0 \\
\hline
 & & & y_{n+1} & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\
 & & & \hat{y}_{n+1} & 0 & 1 & 0 \\
 & & & y_{n+\xi} & \frac{\xi(4\xi^2-9\xi+6)}{6} & \frac{2\xi^2(3-2\xi)}{3} & \frac{\xi^2(4\xi-3)}{6}
\end{array} \tag{4.3}$$

where in Tableau (4.3), γ denotes the stepsize ratio h_n/h_{n-1}.

5. Numerical experiments

In this section we shall report the numerical results obtained by two new (parallel) EPTRK methods of orders 5 and 8, two sequential codes DOPRI5 and DOP853 and a parallel code PIRK8. The codes DOPRI5 and DOP853 are embedded explicit RK methods due to Dormand and Prince and coded by Hairer and Waner (cf. [12]). They are based on a pair 5(4) and a "triple" 8(5)(3), respectively. DOP853 is the new version of DOPRI8 with a "stretched" error estimator (see [12, p. 254]). PIRK8 is eighth-order parallel code taken from [14]. These three codes are currently recognized as being the most efficient sequential and parallel integrators for first-order ODE nonstiff problems.

The fifth-order EPTRK method is based on the collocation vector

$$c^5 = (0.089, 0.409, 0.788, 1.000, 1.409)^T, \tag{5.1a}$$

with fourth-order embedded formula based on

$$\tilde{c}^4 = (0.409, 0.788, 1.000, 1.409)^T. \tag{5.1b}$$

The eighth-order EPTRK method is based on the collocation vector

$$c^8 = (0.057, 0.277, 0.584, 0.860, 1.000, 1.277, 1.584, 1.860)^T, \tag{5.2a}$$

with two embedded formulas of orders 6 and 4 based on

$$\begin{aligned} \tilde{c}^6 &= (0.584, 0.860, 1.000, 1.277, 1.584, 1.860)^T, \\ \tilde{c}^4 &= (0.057, 0.277, 0.584, 0.860)^T. \end{aligned} \tag{5.2b}$$

Notice that the choice of the collocation vectors in (5.1a) and (5.2a) minimizes the principal error terms for some stage approximated values (cf. [6, Theorem 2.4]) and gives slightly larger stability boundaries. No special effort has been made to optimize the parameters of the above methods. An optimal choice of the method parameters was beyond the scope of this work.

The real and imaginary stability boundary pairs $(\beta_{\mathrm{im}}, \beta_{\mathrm{re}})$ of the methods defined by (5.1a) and (5.2a) are numerically calculated and equal to (0.414, 0.415) and (0.388, 0.388), respectively. In terms of considering stability of a method, it is the scaled stability region and not the stability region that is significant (cf., e.g., [3, p. 198]). The stability region of an EPTRK method is at the same time the scaled stability region. With these stability boundary pairs, the associated EPTRK methods are expected to be efficient for solving problem (1.1) especially with a stringent accuracy demand.

For these EPTRK methods we apply an implementation strategy using local extrapolation and a starting procedure based on corrections until convergence of an appropriate s-stage collocation RK corrector. The EPTRK pair 5(4) defined by (5.1) is implemented with the same strategy as in DOPRI5. The EPTRK "triple" 8(6)(4) defined by (5.2) is implemented with two embedded formulas of orders 6 and 4 giving a "stretched" error estimator of local order 9 following the approach used in DOP853. That is, if err_6 and err_4 are two error estimates given by the embedded formulas defined by (5.2b) of orders 6 and 4, respectively, then we consider

$$\mathrm{err} = \mathrm{err}_6 \frac{\mathrm{err}_6}{\mathrm{err}_4 + 0.01\,\mathrm{err}_6} = O(h^9)$$

as the error estimator. It behaves like the local error of the method. These two new EPTRK methods will be denoted by EPTRK54 and EPTRK864. The new stepsize is chosen in the same way as in DOPRI5 and DOP853 with Atol = Rtol, facmax = 2 and facmin = 0.5 (cf. [12, p. 167]).

Furthermore, in the tables of numerical results, NSfcn and NPfcn denote the number of f-evaluations in sequential and parallel implementation modes, NCD is the number of correct decimal digits, Nstep and Nrejct are the total number of integration steps and of rejected ones, respectively. All the computations were carried out on a 14-digit precision computer. An actual implementation on a parallel machine is a subject of our later work [9].

Table 1
Numerical results for Problem (5.3)

Methods	Tol	NSFcn	NPFcn	NCD	Nstep	Nrejct
DOPRI5	10^{-5}	188	188	2.5	31	7
	10^{-7}	356	356	4.4	59	7
	10^{-9}	758	758	6.5	126	0
	10^{-11}	1880	1880	8.7	313	0
	10^{-13}	4706	4706	10.8	784	0
EPTRK54	10^{-5}	375	75	2.7	49	9
	10^{-7}	550	110	6.6	99	0
	10^{-9}	1305	261	9.2	251	0
	10^{-11}	3205	641	11.8	632	0
DOP853	10^{-5}	179	179	4.5	15	3
	10^{-7}	307	307	5.6	26	7
	10^{-9}	495	495	7.0	42	11
	10^{-11}	780	780	8.9	66	14
	10^{-13}	1125	1125	10.7	94	5
EPTRK864	10^{-5}	480	60	2.6	45	4
	10^{-7}	632	79	5.8	67	0
	10^{-9}	984	123	8.9	112	0
	10^{-11}	1232	154	10.2	145	0

5.1. Two body problem

As a first test example, we integrate the two-body problem on the interval $[0, 2\pi]$, with eccentricity $\varepsilon = \frac{6}{10}$ given by (cf., e.g., [14, 16])

$$\begin{aligned}
y_1'(t) &= y_3(t), & y_1(0) &= 1-\varepsilon,\\
y_2'(t) &= y_4(t), & y_2(0) &= 0,\\
y_3'(t) &= \frac{-y_4(t)}{[y_1^2(t)+y_2^2(t)]^{3/2}}, & y_3(0) &= 0,\\
y_4'(t) &= \frac{-y_2(t)}{[y_1^2(t)+y_2^2(t)]^{3/2}}, & y_4(0) &= \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}.
\end{aligned} \tag{5.3}$$

The numerical results for this problem are listed in Table 1. We see from Table 1 that in parallel implementation mode, EPTRK54 offers a speed-up factor ranging from 3 to 8 when compared with DOPRI5 while EPTRK864 is a factor ranging from 3 to 6 faster than DOP853 (depending on the accuracy required). Even in sequential implementation mode, the method EPTRK54 beats DOPRI5 by a factor more than 1.5 with stringent error tolerances.

Table 2
Numerical results for Problem (5.4)

Methods	Tol	NSFcn	NPFcn	NCD	Nstep	Nrejct
DOPRI5	10^{-5}	452	452	3.2	75	8
	10^{-7}	974	974	5.3	162	4
	10^{-9}	2360	2360	7.4	393	2
	10^{-11}	5876	5876	9.4	979	0
	10^{-13}	14750	14750	11.4	2458	1
EPTRK54	10^{-5}	650	130	3.7	123	10
	10^{-7}	1490	298	6.4	292	10
	10^{-9}	3595	719	9.2	714	8
	10^{-11}	8925	1785	11.8	1780	8
DOP853	10^{-5}	552	552	4.5	47	14
	10^{-7}	825	825	6.2	70	17
	10^{-9}	1265	1265	8.0	107	21
	10^{-11}	1950	1950	10.2	164	20
	10^{-13}	3123	3123	12.2	261	11
EPTRK864	10^{-5}	1120	140	5.0	133	26
	10^{-7}	1608	201	7.7	195	21
	10^{-9}	2504	313	10.0	308	20
	10^{-11}	3096	387	10.8	382	13

5.2. *Fehlberg problem*

For the second test example, we consider the often-used test problem of Fehlberg on the interval [0, 5] (cf., e.g., [5, 14, 16])

$$\begin{aligned} y_1'(t) &= 2ty_1(t)\log(\max\{y_2(t), 10^{-3}\}) \qquad y_1(0) = 1, \\ y_2'(t) &= -2ty_2(t)\log(\max\{y_1(t), 10^{-3}\}) \quad y_2(0) = e, \end{aligned} \tag{5.4}$$

with the exact solution $y_1(t) = \exp(\sin(t^2))$, $y_2(t) = \exp(\cos(t^2))$. The results reported in Table 2 show a similar efficiency of EPTRK54 and EPTRK876 in parallel and sequential implementation modes as for the two-body problem when they are compared with DOPRI5 and DOP853.

5.3. *Jacobian elliptic functions problem*

The final test example is the Jacobian elliptic functions sn, cn, dn problem for the equation of motion of a rigid body without external forces on a long integration interval [0, 60] (cf., e.g., [12, Problem JACB, p. 240], also [1, 16])

$$\begin{aligned} y_1'(t) &= y_2(t)y_3(t), & y_1(0) &= 0, \\ y_2'(t) &= -y_1(t)y_3(t), & y_2(0) &= 1, \\ y_3'(t) &= -0.51y_1(t)y_2(t), & y_3(0) &= 1. \end{aligned} \tag{5.5}$$

Table 3
Numerical experiment results for Problem (5.5)

Methods	Tol	NSFcn	NPFcn	NCD	Nstep	Nrejct
DOPRI5	10^{-5}	968	968	4.0	161	30
	10^{-7}	2024	2024	5.2	337	23
	10^{-9}	4682	4682	6.8	780	0
	10^{-11}	11 768	11 768	8.7	1961	0
	10^{-13}	29 564	29 564	10.7	4927	0
EPTRK54	10^{-5}	1255	251	3.6	241	0
	10^{-7}	3050	610	6.7	601	0
	10^{-9}	7580	1516	9.3	1508	0
	10^{-11}	18 970	3794	11.8	3787	0
DOP853	10^{-5}	1066	1066	3.6	91	28
	10^{-7}	1458	1458	5.4	123	20
	10^{-9}	2339	2339	7.4	196	15
	10^{-11}	3830	3830	9.6	319	0
	10^{-13}	6818	6818	11.7	568	0
EPTRK864	10^{-5}	2104	263	4.4	253	28
	10^{-7}	3248	406	7.5	397	28
	10^{-9}	5160	645	9.6	1637	29
	10^{-11}	6512	814	10.4	806	24

Table 4
Numerical experiment results obtained by PIRK8 for various problems

Problems	Tol	NSFcn	NPFcn	NCD	Nstep	Nrejct
(5.3) (Section 5.1)	10^{-5}	652	163	4.8	21	5
	10^{-7}	1024	256	6.4	33	8
	10^{-9}	1592	398	8.2	51	10
	10^{-11}	2320	580	10.0	73	4
(5.4) (Section 5.2)	10^{-5}	1544	386	5.3	50	14
	10^{-7}	2336	584	6.7	75	16
	10^{-9}	3820	955	8.6	1228	21
	10^{-11}	6048	1512	10.5	191	16
(5.5) (Section 5.3)	10^{-5}	3192	798	5.1	103	26
	10^{-7}	5368	1342	6.7	173	42
	10^{-9}	8956	2239	8.5	288	65
	10^{-11}	13 564	3391	10.2	430	49

The numerical results are given in Table 3 which give rise to roughly the same conclusions as formulated in the two previous examples.

5.4. Comparison with a parallel PIRK code

Testing new numerical methods for nonstiff problems (1.1), most people compare them with the PIRK methods initially introduced by van der Houwen and Sommeijer [14] (see, e.g., [2, 5, 6, 8]). That is motivated by not only the orginality but also the high performance of the PIRK methods. In this section we also report on the numerical results obtained by PIRK8 code (a code based on the PIRK method of order 8) for the three above problems. Because a starting stepsize in the implementation of PIRK8 not much influences efficiency of the code for these problems, we set in all examples the starting stepsize $h_0 = \sqrt{\text{Tol}}$. A comparison of the results obtained by PIRK8 listed in Table 4 and of the ones obtained by EPTRK54 and EPTRK864 listed in Tables 1–3 shows that for the considered Tol-values, PIRK8 is less efficient than EPTRK54 and EPTRK864 in both sequential and parallel implementation modes.

6. Concluding remarks

In this paper we have developed a class of continuous variable stepsize embedded explicit pseudo two-step RK methods requiring only *one* effective sequential f-evaluations per step for any order of accuracy. Implemented with a variable stepsize strategy using embedding techniques, two explicit pseudo two-step RK methods derived from this class are shown to be superior to the currently most efficient sequential and parallel codes DOPRI5, DOP853 and PIRK8. In a very stringent accuracy range, these methods are expected to have an efficiency equal if not superior to sequential codes even in a sequential implementation. These conclusions encourage us to pursue the study of explicit pseudo two-step RK methods. In particular, we will concentrate on the optimal choice of the method parameters, numerical experiments with high-order explicit pseudo two-step RK methods and also on an implementation of these methods on parallel computers.

Acknowledgements

A part of this work was done while I was a guest at the Institute of Numerical Mathematics, Halle University, Germany. I would like to thank Prof. Dr. K. Strehmel, Prof. Dr. R. Weiner for their kind regards and interest in my research works.

References

[1] M. Abramowitz, I.A. Stegun, Handbook of Mathematical Functions, National Bureau of Standards Applied Mathematics Series, vol. 55, Dover, New York, 1970.
[2] K. Burrage, Efficient block predictor-corrector methods with a small number of corrections, J. Comput. Appl. Math. 45 (1993) 139–150.
[3] K. Burrage, Parallel and Sequential Methods for Ordinary Differential Equations, Clarendon Press, Oxford, 1995.
[4] M.T. Chu, H. Hamilton, Parallel solution of ODE's by multi-block methods, SIAM J. Sci. Statist. Comput. 8 (1987) 137–157.
[5] N.H. Cong, Parallel iteration of symmetric Runge–Kutta methods for nonstiff initial-value problems, J. Comput. Appl. Math. 51 (1994) 117–125.

[6] N.H. Cong, Explicit pseudo two-step Runge–Kutta methods for parallel computers, submitted for publication.
[7] N.H. Cong, T. Mitsui, Collocation-based two-step Runge–Kutta methods, Japan J. Ind. Appl. Math. 13 (1996) 171–183.
[8] N.H. Cong, T. Mitsui, A class of explicit parallel two-step Runge–Kutta methods, Japan J. Ind. Appl. Math. 14 (1997) 303–313.
[9] N.H. Cong, H. Podhaisky, R. Weiner, Numerical experiments with some explicit pseudo two-step RK methods on a shared memory parallel computer, Math. Applic. 36 (2) (1998) 107–116.
[10] A.R. Curtis, High-order explicit Runge–Kutta formulae, their uses, and limitations, J. Inst. Math. Appl. 16 (1975) 35–55.
[11] E. Hairer, A Runge–Kutta method of order 10, J. Inst. Math. Appl. 21 (1978) 47–59.
[12] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations, I. Nonstiff Problems, Springer, Berlin, 1993.
[13] P.J. van der Houwen, N.H. Cong, Parallel block predictor-corrector methods of Runge–Kutta type, Appl. Numer. Math. 13 (1993) 109–123.
[14] P.J. van der Houwen, B.P. Sommeijer, Parallel iteration of high-order Runge–Kutta methods with stepsize control, J. Comput. Appl. Math. 29 (1990) 111–127.
[15] P.J. van der Houwen, B.P. Sommeijer, Block Runge–Kutta methods on parallel computers, Z. Angew. Math. Mech. 68 (1992) 3–127.
[16] T.E. Hull, W.H. Enright, B.M. Fellen, A.E. Sedgwick, Comparing numerical methods for ordinary differential equations, SIAM J. Numer. Anal. 9 (1972) 603–637.
[17] K.R. Jackson, S.P. Nørsett, Parallel Runge–Kutta methods, Manuscript 1988.
[18] I. Lie, Some aspects of parallel Runge–Kutta methods, Report No. 3/87, Division Numerical Mathematics, University of Trondheim, Norway, 1987.
[19] S.P. Nørsett, H.H. Simonsen, Aspects of parallel Runge–Kutta methods, in: A. Bellen, C.W. Gear, E. Russo (Eds.), Numerical Methods for Ordinary Differential Equations, Proceedings L'Aquilla 1987, Lecture Notes in Mathematics, vol. 1386, Springer, Berlin 1989.