HNUE JOURNAL OF SCIENCE
Natural Science, 2024, Volume 69, Issue 2, pp. 3-16
This paper is available online at http://hnuejs.edu.vn/ns
DOI: 10.18173/2354-1059.2024-0015

EXPLICIT PSEUDO THREE-STEP RUNGE-KUTTA METHODS FOR NONSTIFF INITIAL VALUE PROBLEMS

Nguyen Thu Thuy
Faculty of Mathematics, Hanoi National University of Education, Hanoi city, Vietnam
Corresponding author: Nguyen Thu Thuy, e-mail: ntthuy@hnue.edu.vn

Received May 17, 2024. Revised June 16, 2024. Accepted June 23, 2024.

Abstract. This paper investigates a class of explicit pseudo three-step Runge-Kutta methods for arbitrarily high order nonstiff initial value problems for systems of first-order differential equations. By using collocation techniques and by suitably choosing collocation points we can obtain a stable s-stage explicit pseudo three-step Runge-Kutta method (EPThRK method) of order $p = 2s$ requiring only *one* effective sequential $\mathbf{f}$- evaluation per step on s-processor parallel computers. By a few widely-used test problems, we show the superiority of the new EPThRK methods proposed in this paper over red well-known parallel *PIRK* codes and efficient sequential *ODEX*, *DOPRI5* and *DOP853* codes available in the literature.

Keywords: Runge-Kutta methods, three-step methods, stability, parallelism.

1. Introduction

We consider numerical solutions of following nonstiff initial value problem (IVP) of first-order ordinary differential equations (ODEs)

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}(t)), \qquad \mathbf{y}(t_0) = \mathbf{y}_0, \qquad t_0 \leqslant t \leqslant T, \tag{1.1}$$

where $\mathbf{y}$, $\mathbf{f} \in \mathbb{R}^d$. The most efficient numerical methods for solving this problem are the explicit Runge-Kutta methods (RK methods). In the literature, the sequential explicit RK methods up to order 10 can be found in e.g., [1]-[4]. In order to exploit the facility of parallel computers, a number of parallel RK-type methods have been investigated in e.g., [5]-[27]. A common challenge in these mentioned papers is how to reduce, for a given order of accuracy, the required number of sequential $\mathbf{f}$-evaluations per step by using parallel processors. In the present paper, we investigate a particular class of numerical methods called explicit pseudo three-step RK methods (EPThRK methods) for the numerical solution of the problem (1.1). The three-step nature of the methods considered in this paper is similar to the two-step nature of the methods investigated in [10]. The approach that we apply here is to approximate the stage values at the present step by using the stage values from the preceding steps. By using collocation techniques, we can obtain an s-stage EPThRK method possessing step point order p^* and stage order q^* with $s \leqslant p^*, q^* \leqslant 2s$, requiring s $\mathbf{f}$-evaluations per step. However, each of these s $\mathbf{f}$-evaluations can be computed in parallel. Consequently, when an s-stage EPThRK method is implemented on an s-processor computer, only *one* sequential $\mathbf{f}$-evaluation per step is required. The approach used in this paper can be extended to the case of special second-order ODEs (cf. [28]).

In Section 2, we introduce the proposed EPThRK methods. We also study their order conditions and stability properties. In Section 3, we present numerical comparisons of EPThRK methods with the most efficient parallel and sequential numerical codes available in up-to-date literarture.

2. Explicit pseudo three-step Runge-Kutta methods

We suppose that an s-dimensional collocation vector $\mathbf{c}$ =$(c_1, \ldots, c_s)^T$ with distinct abscissas c_i is given. An s-stage EPThRK method for solving the problem (1.1) is defined by

$$\begin{aligned} \mathbf{Y}_{n,i} = \mathbf{y}_n + h\sum_{j=1}^{s} b_{ij}\mathbf{f}(t_{n-2} + hc_j, \mathbf{Y}_{n-2,j}) \\ + h\sum_{j=1}^{s} a_{ij}\mathbf{f}(t_{n-1} + hc_j, \mathbf{Y}_{n-1,j}), \quad i = 1, \ldots, s, \end{aligned} \tag{2.1a}$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\sum_{j=1}^{s} b_j\mathbf{f}(t_n + hc_j, \mathbf{Y}_{n,j}). \tag{2.1b}$$

In Eq. (2.1), $\mathbf{y}_{n+1} \approx \mathbf{y}(t_{n+1})$, $\mathbf{y}_n \approx \mathbf{y}(t_n)$, h is the stepsize, the $s \times s$ matrices $A = (a_{ij}), B = (b_{ij})$, and the s-dimensional vector $\mathbf{b} = (b_j)$ are the method parameters matrices and vectors. The vector components $\mathbf{Y}_{n,j}$, $\mathbf{Y}_{n-1,j}$ and $\mathbf{Y}_{n-2,j}$ denote the jth vector components of the stage vectors representing numerical approximations to the exact solutions $\mathbf{y}(t_n + hc_j)$, $\mathbf{y}(t_{n-1} + hc_j)$ and $\mathbf{y}(t_{n-2} + hc_j)$, respectively. The method parameters matrices A, B and vector $\mathbf{b}$ will be determined by order conditions (see Section 2.1). This method is similar to a RK method but it is not a RK method. It has no implicit relation and carries information from preceding nth, $(n-1)$th and $(n-2)$th steps, and we, therefore, call method (2.1) the *s-stage explicit pseudo three-step Runge-Kutta method (EPThRK method)* based on the collocation vector $\mathbf{c}$. For convenience, we specify this EPThRK method by the following tableau

$$\begin{array}{c|c|c|c} B & A & \mathbf{c} & O \\ \hline & & \mathbf{y}_{n+1} & \mathbf{b}^T \end{array}$$

In order to start the method (2.1), an appropriate starting procedure is needed to generate a sufficiently accurate starting vector components $\mathbf{Y}_{0,j}, \mathbf{Y}_{1,j}, j = 1, \ldots, s$ and value $\mathbf{y}_2$ from y_0. This can be done, for example, by using an appropriate BPIRK-type method (cf. e.g., [19]). For the EPThRK method (2.1), at each step, we need to compute $3s$ $\mathbf{f}$-evaluations of $\mathbf{f}(t_{n-2} + hc_j, \mathbf{Y}_{n-2,j})$, $\mathbf{f}(t_{n-1} + hc_j, \mathbf{Y}_{n-1,j})$, $\mathbf{f}(t_n + hc_j, \mathbf{Y}_{n,j})$, $j = 1, \ldots, s$. However, $2s$ $\mathbf{f}$-evaluations of $\mathbf{f}(t_{n-2} + hc_j, \mathbf{Y}_{n-2,j})$, $\mathbf{f}(t_{n-1} + hc_j, \mathbf{Y}_{n-1,j})$, $j = 1, \ldots, s$ are already available from the two preceding steps so that only s $\mathbf{f}$-evaluations of $\mathbf{f}(t_n + hc_j, \mathbf{Y}_{n,j})$, $j = 1, \ldots, s$ are required. These s $\mathbf{f}$-evaluations can be evaluated in parallel on s processors. Consequently, the s-stage EPThRK method (2.1) implemented on an s-processor computer, requires just *one* effective sequential $\mathbf{f}$-evaluation per step.

The order and stage order of EPThRK method (2.1) can be studied in the same way as the order and stage order of RK methods. Thus suppose that $\mathbf{y}_n = \mathbf{y}(t_n)$ and $\mathbf{Y}_{n-2,j} = \mathbf{y}(t_{n-2} + hc_j)$, $\mathbf{Y}_{n-1,j} = \mathbf{y}(t_{n-1} + hc_j)$, $j = 1, \ldots, s$. Then we have the following orders definition

Definition 2.1. *The EPThRK method* (2.1) *is said to be of order* p^* *if*

$$\mathbf{y}(t_{n+1}) - \mathbf{y}_{n+1} = O(h^{p^*+1})$$

and stage order $q^* = min\{p^*, q\}$ *if in addition,*

$$\mathbf{y}(t_n + hc_i) - \mathbf{Y}_{n,i} = O(h^{q+1}), \quad i = 1, \ldots, s.$$

Notice that the local stage order of the method (2.1) equals $q + 1$. Now in the next section, we shall consider the order conditions for EPThRK methods.

2.1. Order conditions

In this section we consider order conditions for the EPTRK methods. For the (fixed) stepsize h, the qth-order conditions for $(2.1a)$ and the pth-order conditions for $(2.1b)$ are derived by replacing $\mathbf{Y}_{n-2,j}$, $\mathbf{Y}_{n-1,j}$, $\mathbf{y}_n$, $\mathbf{Y}_{n,j}$ and $\mathbf{y}_{n+1}$ by the exact solution values $\mathbf{y}(t_{n-2} + hc_j) = \mathbf{y}(t_n + h(c_j - 2))$, $\mathbf{y}(t_{n-1} + hc_j) = \mathbf{y}(t_n + h(c_j - 1))$, $\mathbf{y}(t_n)$, $\mathbf{y}(t_n + hc_j)$ and $\mathbf{y}(t_{n+1}) = \mathbf{y}(t_n + h)$, respectively. On substitution of these exact solution values into (2.1), we are led to the relations

$$\begin{aligned}\mathbf{y}(t_n + hc_i) - \mathbf{y}(t_n) - h\sum_{j=1}^{s} b_{ij}\mathbf{y}'(t_n + h(c_j - 2)) \\ - h\sum_{j=1}^{s} a_{ij}\mathbf{y}'(t_n + h(c_j - 1)) = O(h^{q+1}), \quad i = 1, \ldots, s\end{aligned} \tag{2.2a}$$

$$\mathbf{y}(t_n + h) - \mathbf{y}(t_n) - h\sum_{j=1}^{s} b_j\mathbf{y}'(t_n + hc_j) = O(h^{p+1}). \tag{2.2b}$$

By using the Taylor expansions in the neighbourhood of t_n, we can expand the left-hand sides of (2.1) in powers of h and obtain

$$\sum_{l=1}^{q} C_i^{(l)}\Big(h\frac{d}{dt}\Big)^l \mathbf{y}(t_n) + C_i^{(q+1)}\Big(h\frac{d}{dt}\Big)^{q+1}\mathbf{y}(t_i^*) = O(h^{q+1}), \tag{2.3a}$$

$$\sum_{l=1}^{p} D^{(l)}\Big(h\frac{d}{dt}\Big)^l \mathbf{y}(t_n) + D^{(p+1)}\Big(h\frac{d}{dt}\Big)^{p+1}\mathbf{y}(t^*) = O(h^{p+1}), \tag{2.3b}$$

where, t_i^* and t^* are suitably chosen points in the interval $[t_{n-1}, t_{n+1}]$ and

$$C_i^{(l)} = \frac{(c_i)^l}{l} - \sum_{j=1}^{s} b_{ij}(c_j-2)^{l-1} - \sum_{j=1}^{s} a_{ij}(c_j-1)^{l-1}, \quad i = 1,\dots,s, \tag{2.3c}$$

$$D^{(l)} = \frac{1}{l} - \sum_{j=1}^{s} b_j(c_j)^{l-1}, \tag{2.3d}$$

For the order and stage order of EPThRK methods, we have the following theorem:

Theorem 2.1. *If the function f is Lipschitz continuous, and if*

$$\frac{(c_i)^l}{l} = \sum_{j=1}^{s} b_{ij}(c_j-2)^{l-1} + \sum_{j=1}^{s} a_{ij}(c_j-1)^{l-1}, \quad i = 1,\dots,s, \quad l = 1,\dots,q, \tag{2.4a}$$

$$\frac{1}{l} = \sum_{j=1}^{s} b_j(c_j)^{l-1}, \quad l = 1,\dots,p, \tag{2.4b}$$

then the EPThRK method (2.1) has order $p^* = min\{p, q+1\}$ and stage order $q^* = min\{p, q\}$ for any collocation vector $\mathbf{c} = (c_1,\dots,c_s)^T$ with distinct abscissas c_i.

Proof. Suppose that $y_n = y(t_n)$ and $\mathbf{Y}_{n-2,i} = \mathbf{y}(t_{n-2} + hc_i)$, $\mathbf{Y}_{n-1,i} = \mathbf{y}(t_{n-1} + hc_i)$, $i = 1,\dots,s$. The Eq. (2.3) and (2.4) ensure that the Eq. (2.2) are satisfied, then we have the following local order relations

$$\mathbf{y}(t_n + hc_i) - \mathbf{Y}_{n,i} = O(h^{q+1}), \quad i = 1,\dots,s. \tag{2.5}$$

Furthermore

$$\begin{aligned}\mathbf{y}(t_{n+1}) - \mathbf{y}_{n+1} &= \mathbf{y}(t_{n+1}) - \mathbf{y}(t_n) - h\sum_{j=1}^{s} b_j\mathbf{y}'(t_n + hc_j)\\ &+ h\sum_{j=1}^{s} b_j[\mathbf{f}(t_n + hc_j, \mathbf{y}(t_n + hc_j)) - \mathbf{f}(t_n + hc_j, \mathbf{Y}_{n,j})] = O(h^{p+1}) + O(h^{q+2})\end{aligned} \tag{2.6}$$

Definition (2.1) and Eq. (2.5), (2.6) give us $p^* = min\{p, q+1\}$, $q^* = min\{p, q\}$ and Theorem (2.1) is proved. □

In order to express the parameter vector and matrices $\mathbf{b} = (b_i)$, $B = (b_{ij})$, $A = (a_{ij})$ explicitly in terms of the collocation vector $\mathbf{c}$, we define the following matrices and vector

$$\begin{aligned}
P &:= \Big(\mathbf{c}, \frac{\mathbf{c}^2}{2}, \frac{\mathbf{c}^3}{3}, \ldots, \frac{\mathbf{c}^s}{s}\Big), & P^* &:= \Big(\frac{\mathbf{c}^{s+1}}{s+1}, \frac{\mathbf{c}^{s+2}}{s+2}, \ldots, \frac{\mathbf{c}^{2s}}{2s}\Big),\\
Q &:= \Big(\mathbf{e}, (\mathbf{c}-\mathbf{e}), \ldots, (\mathbf{c}-\mathbf{e})^{s-1}\Big), & Q^* &:= \Big((\mathbf{c}-\mathbf{e})^{s}, \ldots, (\mathbf{c}-\mathbf{e})^{2s-1}\Big),\\
V &:= \Big(\mathbf{e}, (\mathbf{c}-2\mathbf{e}), \ldots, (\mathbf{c}-2\mathbf{e})^{s-1}\Big), & V^* &:= \Big((\mathbf{c}-2\mathbf{e})^{s}, \ldots, (\mathbf{c}-2\mathbf{e})^{2s-1}\Big),\\
R &:= \Big(\mathbf{e}, \mathbf{c}, \mathbf{c}^2, \mathbf{c}^3, \ldots, \mathbf{c}^{s-1}\Big), & \mathbf{g} &:= \Big(1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots, \frac{1}{s}\Big)^T.
\end{aligned}$$

Then the order conditions (2.4) in Theorem (2.1) for $q = 2s, p = s$ can be presented in the form

$$AQ + BV = P, \tag{2.7a}$$

$$AQ^* + BV^* = P^*, \tag{2.7b}$$

$$\mathbf{b}^T R - \mathbf{g}^T = 0. \tag{2.7c}$$

Since the abscissas c_i of the vector $\mathbf{c}$ are assumed to be distinct, the matrices V and R are nonsingular, assuming that the matrix $QV^{-1}V^* - Q^*$ is also nonsingular, from (2.7), we can write

$$A = (PV^{-1}V^* - P^*)(QV^{-1}V^* - Q^*)^{-1}, \quad B = (P - AQ)V^{-1}, \quad \mathbf{b}^T = \mathbf{g}^T R^{-1}. \tag{2.8}$$

In view of Theorem 2.1, it follows from (2.8) that

$$\begin{aligned}
&\mathbf{y}(t_{n+1}) - \mathbf{y}_{n+1} = O(h^{\min\{p,2s+1\}})\\
&\mathbf{y}(t_n + hc_i) - \mathbf{Y}_{n,i} = O(h^{\min\{p,2s\}}), \quad i = 1, \ldots, s.
\end{aligned}$$

Notice that the vector $\mathbf{b}$ defined in (2.8) by the condition $(2.7c)$ is the vector of weights of the collocation implicit RK method based on collocation vector $\mathbf{c}$ so that p at least equals s. With a special choice of the vector $\mathbf{c}$, it is possible to increase the order p beyond s (superconvergence) by satisfying the orthogonality relation (see [3, p. 207]. The following theorem is a direct consequence of Theorem (2.1), the explicit expressions of the parameters of EPThRK methods and the application of the orthogonality relation.

Theorem 2.2. *Suppose that* $\mathbf{c} = (c_1, \ldots, c_s)^T$ *is a collocation vector with distinct abscissas* c_i *and the matrix* $QV^{-1}V^* - Q^*$ *is nonsingular, then an* s*-stage EPThRK method defined by* (2.1) *is of order* $p^* \geqslant s$ *and of stage order* $q^* \geqslant s$ *if the method parameter matrices* A, B *and vector* $\mathbf{b}$ *satisfy the relations in* (2.8). *It has order* $p^* = s + r$ *and stage order* $q^* = s + r$ *if, in addition, the following orthogonality relation is satisfied*

$$P_j(1) = 0, \qquad P_j(x) := \int_0^x \xi^{j-1} \prod_{i=1}^{s} (\xi - c_i) d\xi, \qquad j = 1, \ldots, r. \tag{2.9}$$

We know that if $\mathbf{c}$ is a Gauss-Legendre collocation vector (the vector abscissas are the Gauss-Legendre collocation points), the orthogonality relation (2.9) is verified for $j = 1, \ldots, s$. Thus, we have the following corollary.

Corollary 2.1. *If* $\mathbf{c}$ *is Gauss-Legendre collocation vector, then the* s*-stage EPThRK method defined by* (2.1) *with parameter matrices and vector satisfying relations in* (2.8) *has the order* $p^* = 2s$ *and stage order* $q^* = 2s$.

Thus, the attainable order and stage order for an s-stage EPThRK method is $2s$.

According to the analysis of the local errors in this section, the starting vectors $\mathbf{Y}_{0,j}, \mathbf{Y}_{1,j}, j = 1, \ldots, s$ and value $\mathbf{y}_1$ should verify the order relations

$$\begin{aligned}&\mathbf{y}(t_0 + hc_j) - \mathbf{Y}_{0,j} = O(h^{p^*+1}), \quad \mathbf{y}(t_1 + hc_j) - \mathbf{Y}_{1,j} = O(h^{p^*+1}),\\ &\mathbf{y}(t_2) - \mathbf{y}_2 = O(h^{p^*+1}), \quad j = 1, \ldots, s,\end{aligned} \tag{2.10}$$

where p^* is the order of the underlying EPThRK method.

2.2. Stability properties

To study stability properties, we apply the method (2.1) to the model test equation $y'(t) = \lambda y(t)$, where λ runs through the eigenvalue set of the Jacobian matrix $\partial\mathbf{f}/\partial\mathbf{y}$ which are supposed to be confined within in the left complex plane. For the model test equation, the EPThRK method (2.1) assumes the form

$$\begin{aligned}\mathbf{Y}_n =& y_n\mathbf{e} + zB\mathbf{Y}_{n-2} + zA\mathbf{Y}_{n-1}\\ =& zB\mathbf{Y}_{n-2} + zA\mathbf{Y}_{n-1} + y_n\mathbf{e}, \qquad (2.11a)\\ y_{n+1} =& y_n + z\mathbf{b}^T\mathbf{Y}_n\\ =& y_n + z\mathbf{b}^T(y_n\mathbf{e} + zB\mathbf{Y}_{n-2} + zA\mathbf{Y}_{n-1})\\ =& (1 + z\mathbf{b}^T\mathbf{e})y_n + z^2\mathbf{b}^TB\mathbf{Y}_{n-2} + z^2\mathbf{b}^TA\mathbf{Y}_{n-1}\\ =& z^2\mathbf{b}^TB\mathbf{Y}_{n-2} + z^2\mathbf{b}^TA\mathbf{Y}_{n-1} + (1 + z)y_n. \qquad (2.11b)\end{aligned}$$

From (2.11), we are led to the recursion

$$\begin{pmatrix}\mathbf{Y}_{n-1}\\ \mathbf{Y}_n\\ y_{n+1}\end{pmatrix} = M(z)\begin{pmatrix}\mathbf{Y}_{n-2}\\ \mathbf{Y}_{n-1}\\ y_n\end{pmatrix}, \tag{2.12a}$$

where $M(z)$ is the $(2s + 1) \times (2s + 1)$ matrix defined by

$$M(z) = \begin{pmatrix} O & I & \mathbf{0}\\ zB & zA & \mathbf{e}\\ z^2\mathbf{b}^TB & z^2\mathbf{b}^TA & 1 + z\end{pmatrix}. \tag{2.12b}$$

Here, I is $s \times s$ identity matrix and O is zero $s \times s$ matrix. The $(2s+1) \times (2s+1)$ matrix $M(z)$ will be called the *amplification matrix*, and its spectral radius $\rho(M(z))$ the *stability function*. The stability region denoted by $\mathbb{S}_{stab}$ of EPThRK methods is given by

$$\mathbb{S}_{stab} := \{z : \rho(M(z)) \leqslant 1\}.$$

Since EPThRK methods of the form (2.1) are of three-step nature, we have to confirm their property of zero-stability by the following evident theorem.

Theorem 2.3. *EPThRK methods based on any collocation vector* $\mathbf{c} = (c_1, \ldots, c_s)^T$ *with distinct abscissas* c_i *are always zero-stable.*

The associated real and imaginary stability boundaries denoted by β_{re} and β_{im}, respectively, can be defined in the usual way. The construction of EPThRK methods possessing large stability regions and also their stability plots will be considered in the forthcoming paper. The stability pairs (β_{re}, β_{im}) of some specified EPThRK methods used in the numerical experiments are reported in Section 3.

3. Numerical comparisons

This section shows the efficiency of the EPThRK methods by numerical comparisons of these EPThRK methods with the most efficient existing computational ODE codes. The numerical comparisons can be done by applying various methods and codes to the numerical solution of widely used test problems taken from the literature.

3.1. Choosing EPThRK methods

As we know from Section 2.1, an s-stage EPThRK method attains the highest order $p^* = 2s$ if the method is based on Gauss-Legendre collocation vector $\mathbf{c}$ (see Corollary 2.1). Therefore, we consider EPThRK methods based on Gauss-Legender collocation vectors. We restrict our consideration to the 2-stage and 3-stage EPThRK methods, that is the EPThRK methods based on

$$\mathbf{c} = \Big(\frac{3-\sqrt{3}}{6}, \frac{3+\sqrt{3}}{6}\Big)^T \quad \text{and} \quad \mathbf{c} = \Big(\frac{5-\sqrt{15}}{10}, \frac{1}{2}, \frac{5+\sqrt{15}}{10}\Big)^T, \tag{3.1}$$

respectively. By the choice of the collocation vector $\mathbf{c}$, the resulting EPThRK methods denoted by EPThRK4 and EPThRK6, have the order and stage order equal to $4, 6$, respectively (see Corollary 2.1). This choice of methods allows us to compare the existing numerical codes with the EPThRK methods of comparable orders. The stability pairs (β_{re}, β_{im}) as defined in Section 2.2 of these chosen EPThRK methods are computed to be equal to $(0.301, 0.297)$ and $(0.257, 0.249)$, respectively. These stability pairs are not large but good enough for nonstiff problems.

In the application of the EPThRK methods to the numerical integration, in the first step, the starting stage vectors $\mathbf{Y}_{0,i}, \mathbf{Y}_{1,i}, i = 1, \ldots, s$ and step value $\mathbf{y}_2$ of an EPThRK method will be generated from $\mathbf{y}_0$ by the associated BPIRK method proposed in [19]. This BPIRK method is based on the same collocation vector $\mathbf{c}$ as the underlying EPThRK method. The numerical approximations at appropriate block points will be used for the starting vectors $\mathbf{Y}_{0,i}, \mathbf{Y}_{1,i}, i = 1, \ldots, s$ and value $\mathbf{y}_2$. The vectors $\mathbf{Y}_{0,i}, \mathbf{Y}_{1,i}, i = 1, \ldots, s$ and value $\mathbf{y}_2$ generated in this way will satisfy the accuracy requirement defined by (2.10) in previous section.

The absolute error obtained at the end point of the integration interval is presented in the form 10^{-NCD} (NCD indicates the accuracy and may be interpreted as the average number of correct decimal digits). The computational costs are measured by the values of $NFUN$ denoting the total number of sequential $\mathbf{f}$-evaluations required over the total number of integration steps. Notice that $NFUN$ of EPThRK4 and EPThRK6 methods include the number of $\mathbf{f}$-evaluations in the two first steps with the use of BPIRK methods.

In the numerical comparisons, a method is considered more efficient if for a given $NFUN$, it can give higher NCD or equivalently, for a given NCD, it requires fewer $NFUN$.

The numerical comparisons of various methods in this section are based on the numerical solution of small widely used test problems taken from the literature show a potential superiority of the EPThRK methods over existing codes. This superiority is significant improvement in a parallel machine if the test problems are large enough and/or the $\mathbf{f}$-evaluations are expensive (cf., e.g., [7]).

All the computations were carried out on a 14-digit precision computer.

3.2. Test problems

For the numerical comparisons, we apply various methods and codes to the numerical integration of the three well-known test problems are taken from the ODE literature:

JACB - The Jacobi elliptic functions sn, cn, dn problem for the equation of motion of a rigid body without external forces on the integration interval $[0, 20]$ (cf., e.g., [4, p. 240], also [29])

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

The exact solution is given by the Jacobi elliptic functions $y_1(t) = sn(t;k)$, $y_2(t) = cn(t;k)$, $y_3(t) = dn(t;k)$ (see [30]).

FEHL - The often-used Fehlberg problem on the integration interval $[0, 5]$(cf., e.g., [3, 20])

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

with the exact solution $y_1(t) = \exp\bigl(\sin(t^2)\bigr)$, $y_2(t) = \exp\bigl(\cos(t^2)\bigr)$.

TWOB - The two body problem in the integration interval $[0, 20]$ with eccentricity $\varepsilon = \frac{3}{10}$ (cf., e.g., [20, 29])

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

3.3. Comparison with parallel codes

We compare the fourth-order EPThRK4 and sixth-order EPThRK6 methods proposed in this paper with parallel sixth-order *PIRK6* and eighth-order *PIRK8* codes proposed in [20]. The *PIRK* codes are recognized as one of the most reliable and efficient parallel codes available in the literature. The codes *PIRK6*, *PIRK8* are implemented with $TOL = 10^{-2}, 10^{-4}, \ldots, 10^{-12}$ and the methods EPThRK4, EPThRK6 are implemented with fixed stepsize $h = 1/20^j, j = 1, \ldots, 8$ (for *JACB*, *TWOB*) and $h = 1/5^j, j = 1, \ldots, 8$ (for *FEHL*). The values of NCD are plotted as a function of the values of $NFUN$ in Figures 1-3.

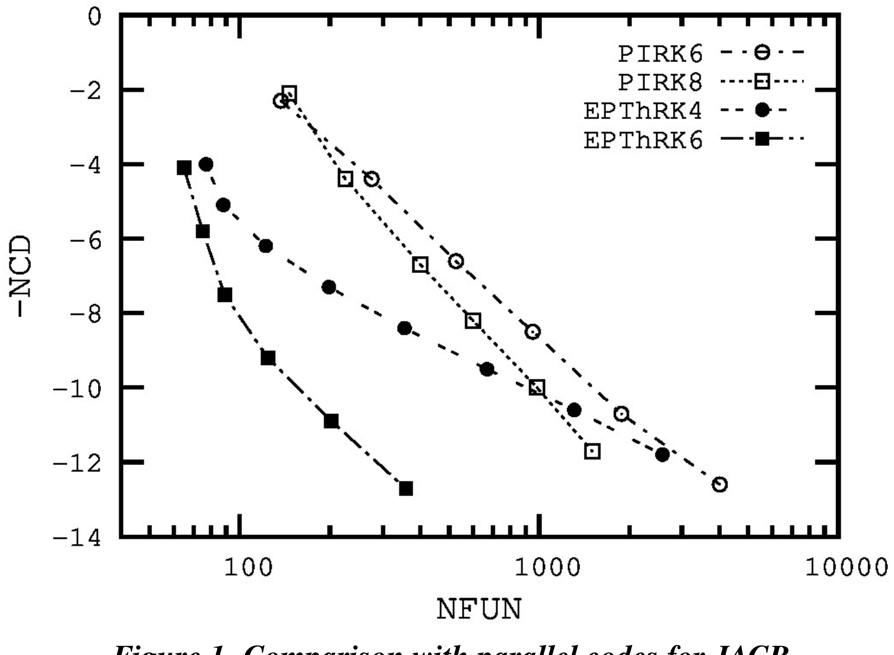

Figure 1. Comparison with parallel codes for JACB

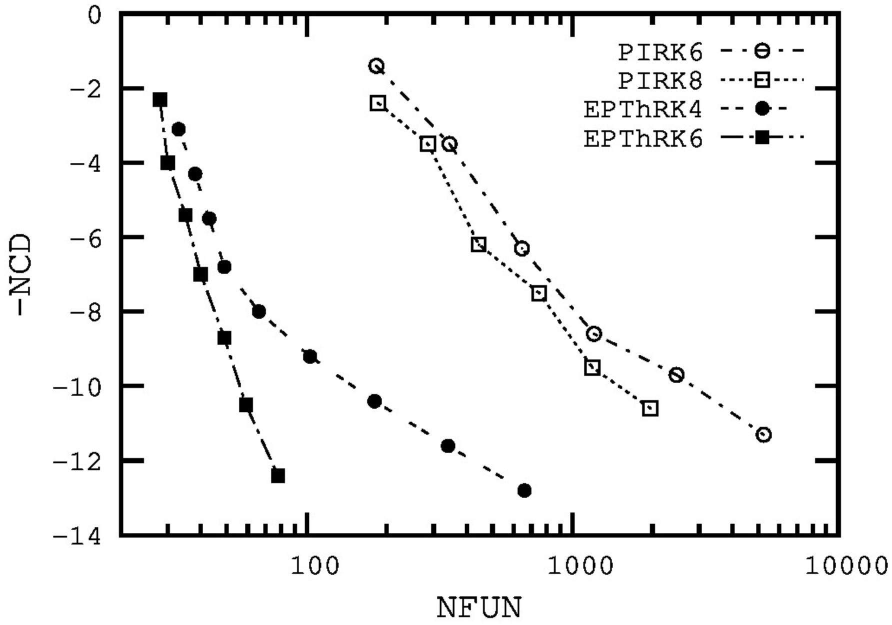

Figure 2. Comparison with parallel codes for FEHL

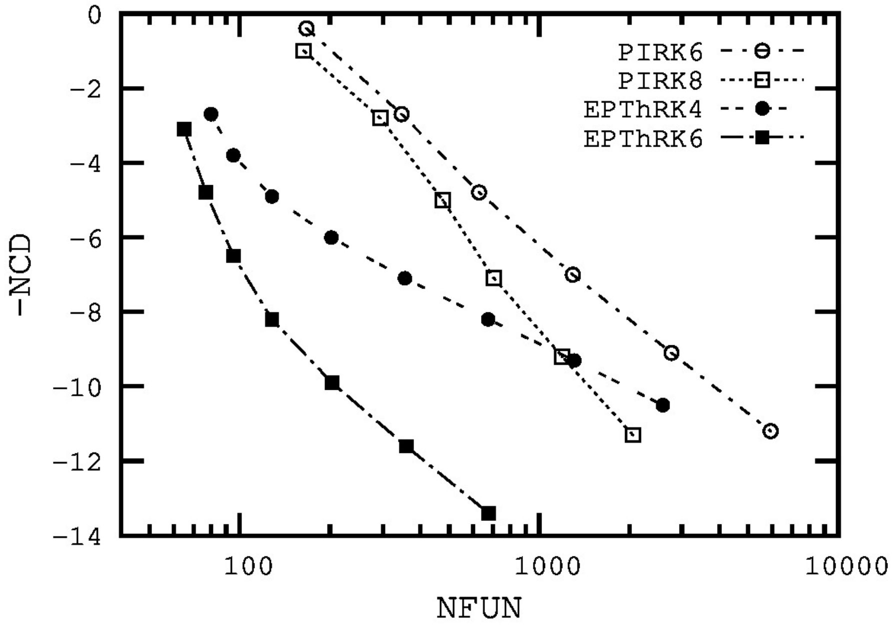

Figure 3. Comparison with parallel codes for TWOB

For the problem *JACB*, the $NCDs$ and $NFUNs$ presented in Figure 1 clearly show that the EPThRK6 method is by far superior to *PIRK6* and *PIRK8* codes. In the low and middle accuracy range, EPThRK4 method is also more efficient than *PIRK6* and *PIRK8* codes

For the problem *FEHL*, the $NCDs$ and $NFUNs$ presented in Figure 2 show that EPThRK4 and EPThRK6 methods are much superior to *PIRK6* and *PIRK8* codes.

For the problem *TWOB*, the $NCDs$ and $NFUNs$ are presented in Figure 3 giving us nearly the same conclusions as formulated in the case of *JACB*.

3.4. Comparison with sequential codes

In Section 3.3, the fourth-order EPThRK4 and sixth-order EPThRK6 methods were compared with the sixth-order *PIRK6* and eighth-order *PIRK8* codes. In this section, we shall compare these EPThRK4 and EPThRK6 methods with some of the best sequential nonstiff codes currently available, that is the three codes *ODEX*, *DOPRI5* and *DOP853*. The code *ODEX* is an extrapolation-algorithm based on explicit midpoint rule. It was coded by Hairer and Wanner (see [3, Section II.9]). The codes *DOPRI5* and *DOP853* are embedded in explicit RK methods due to Dormand and Prince. They are based on the pair 5(4) and the "triple" 8(5)(3), respectively. The *DOPRI5* and *DOP853* codes also were coded by Hairer and Wanner (see [4]). *DOP853* is the new version of *DOPRI8* with a "stretched" error estimator (see [4, p. 254]). These three codes belong to the most efficient currently existing sequential codes for nonstiff first-order ODE problems.

We applied the methods EPThRK4, EPThRK6 and the codes *ODEX*, *DOPRI5* and *DOP853* with $ATOL = RTOL = 10^{-2}, 10^{-4}, \ldots, 10^{-12}$ to the above three test problems. The obtained NCD and $NFUN$ values are plotted in Figures 4-6.

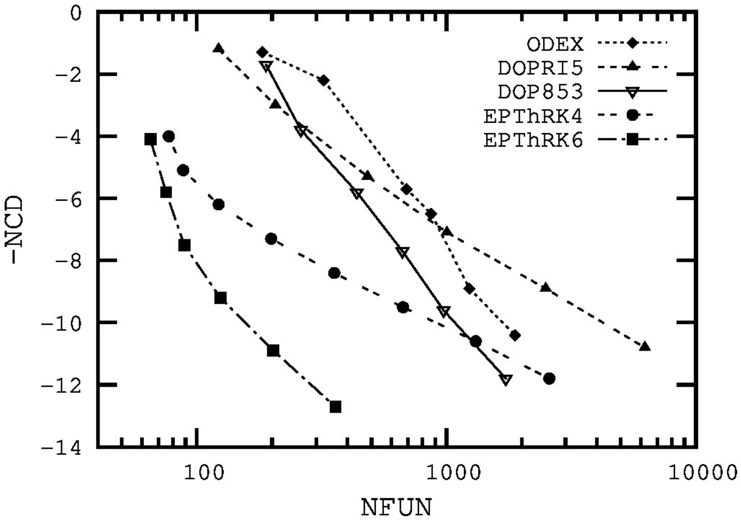

Figure 4. Comparison with sequential codes for JACB

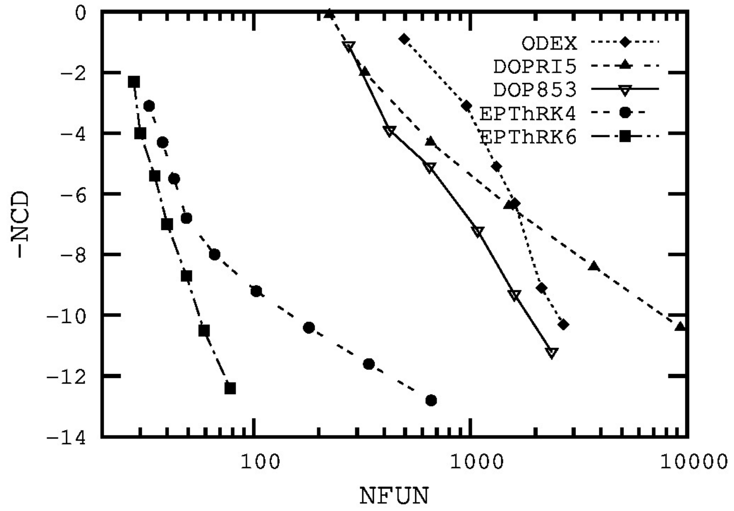

Figure 5. Comparison with sequential codes for FEHL

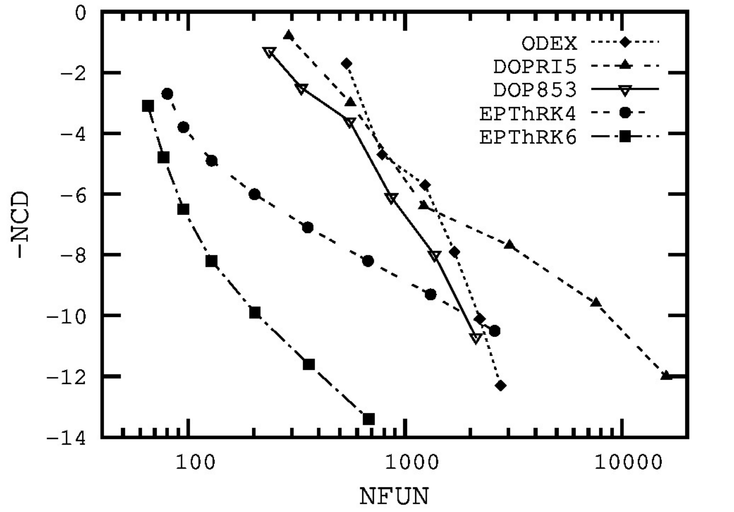

Figure 6. Comparison with sequential codes for TWOB

Looking at Figure 4, Figure 5 and Figure 6, we see that the EPThRK4 and EPThRK6 methods are shown to be more efficient than *ODEX*, *DOPRI5* and *DOP853* codes as in the case of comparisons with parallel codes (see Section 3.3).

4. Conclusions

In this paper, we considered explicit pseudo three-step RK methods (EPThRK methods) which are suitable for use on parallel computers. The resulting EPThRK methods are very competitive numerical integrators in terms of computational cost. After two first steps, the methods require only *one* sequential $\mathbf{f}-evaluation$ per step. We have compared the new EPThRK methods with the most efficient parallel and sequential numerical codes available in recent literarure by applying them to the numerical solution of well-known test problems. In spite of the fact that the results of parallel and sequential codes are obtained by using a stepsize strategy, whereas the EPThRK4 and EPThRK6 methods are applied with fixed stepsizes, it is the EPThRK4 and EPThRK6 methods that are the most efficient.

REFERENCES

[1] Curtis AR, (1975). High-order explicit Runge-Kutta formulae, their uses and limitations. *Journal of the Institute of Mathematics and its Applications*, 16, 35-55.

[2] Hairer E, (1978). A Runge-Kutta method of order 10. *Journal of the Institute of Mathematics and its Applications*, 21, 47-59.

[3] Hairer E, Nørsett SP and Wanner G, (1987). *Solving Ordinary Differential Equations I. Nonstiff Problems*, (1st edition), Springer-Verlag, Berlin.

[4] Hairer E, Nørsett SP & Wanner G, (1993). *Solving Ordinary Differential Equations I. Nonstiff Problems*, (2nd edition), Springer-Verlag, Berlin.

[5] Burrage K, (1993). Efficient block predictor-corrector methods with a small number of corrections. *The Journal of Computational and Applied Mathematics*, 45, 139-150.

[6] Burrage K, (1993). Parallel methods for initial value problems. *Applied Numerical Mathematics*, 11, 5-25.

[7] Burrage K, (1995). *Parallel and Sequential Methods for Ordinary Differential Equations*. Clarendon Press, Oxford.

[8] Burrage K & Suhartanto H, (1997). Parallel iterated methods based on multistep Runge-Kutta methods of Radau type. *Advances in Computational Mathematics*, 7, 37-57.

[9] Nguyen HC, (1994). Parallel iteration of symmetric Runge-Kutta methods for nonstiff initial-value problems. *Journal of Computational and Applied Mathematics*, 51, 117-125.

[10] Nguyen HC, (1999). Explicit pseudo two-step Runge-Kutta methods for parallel computers. *International Journal of Computer Mathematics*, 73, 77-91.

[11] Nguyen HC, (1999). Continuous variable stepsize explicit pseudo two-step RK methods. *Journal of Computational and Applied Mathematics*, 101, 105-116.

[12] Nguyen HC & Mitsui T, (1997). A class of explicit parallel two-step Runge-Kutta methods. *Japan Journal of Industrial and Applied Mathematics*, 14, 303-313.

[13] Nguyen HC & Mitsui T, (2003). Parallel PC iteration of pseudo two-step RK methods for nonstiff IVPs. *Japan Journal of Industrial and Applied Mathematics*, 20, 51-64.

[14] Nguyen HC, Podhaisky H & Weiner R, (1998). Numerical experiments with some explicit pseudo two-step RK methods on a shared memory computer. *Computational and Applied Mathematics*, 36, 107-116.

[15] Nguyen HC & Hoang TV, (1995). An improvement for explicit parallel Runge-Kutta methods. *Vietnam Journal of Mathematics*, 23, 241-252.

[16] Nguyen HC & Le NX, (2003). Parallel-iterated RK-type PC methods with continuous output formulas. *International Journal of Computer Mathematics*, 80, 1027-1037.

[17] Nguyen HC $ Le NX, (2008). Improved parallel-iterated pseudo two-step RK methods for nonstiff IVPs. *Applied Numerical Mathematics*, 58, 160-170.

[18] Nguyen HC & Le NX, (2008). Twostep-by-twostep PIRK-type PC methods with continuous output formulas. *Journal of Computational and Applied Mathematics*, 221, 165-173.

[19] Van der Houwen PJ & Nguyen HC, (1993). Parallel block predictor-corrector methods of Runge-Kutta type. *Applied Numerical Mathematics*, 13, 109-123.

[20] Van der Houwen PJ & Sommeijer BP, (1990). Parallel iteration of high-order Runge-Kutta methods with stepsize control. *Journal of Computational and Applied Mathematics*, 29, 111-127.

[21] Van der Houwen PJ & Sommeijer BP (1992). Block Runge-Kutta methods on parallel computers. *The Journal of Applied Mathematics and Mechanics*, 68, 3-10.

[22] Kulikov GYu & Weiner R, (2010). Variable-stepsize interpolating explicit parallel peer methods with inherent global error control. *SIAM Journal on Scientific Computing*, 32, 1695-1723.

[23] Kulikov GYu & Weiner R, (2010). Doubly quasi-consistent parallel explicit peer methods with built-in global error estimation. *Journal of Computational and Applied Mathematics*, 233, 2351-2364.

[24] Nørsett SP & Simonsen HH, (1989). Aspects of parallel Runge-Kutta methods, in Numerical Methods for Ordinary Differential Equations, *Proceedings L'Aquilla 1987, Lecture Notes in Mathematics*, 1386, (Edited by A. Bellen, C.W. Gear and E. Russo), Springer-Verlag, Berlin.

[25] Schmitt BA, Weiner R & Jebens S, (2009). Parameter optimization for explicit parallel peer two-step methods, *Applied Numerical Mathematics*, 59, 769-782.

[26] Schmitt BA & Weiner R, (2010). Parallel start for explicit parallel two-step peer methods. *Numerical Algorithms*, 53, 363-381.

[27] Weiner R, Kulikov GYu & Podhaisky H, (2012). Variable-stepsize doubly quasi-consistent parallel explicit peer methods with global error control. *Applied Numerical Mathematics*, 62, 1591-1603.

[28] Nguyen HC, Explicit pseudo three-step Runge-Kutta-Nyström methods, in preparation.

[29] Hull TE, Enright WH, Fellen BM & Sedgwick AE, (1972). Comparing numerical methods for ordinary differential equations. *SIAM Journal on Numerical Analysis*, 9, 603-637.

[30] Curtis AR,(1964). *Tables of Jacobian Elliptic Functions Whose Arguments are Rational Fractions of the Quater Period*, H.M.S.O., London.