Runge-Kutta-Nyström-type parallel block predictor-corrector methods *

Nguyen Huu Cong, Karl Strehmel and Rüdiger Weiner

Abstract

This paper describes the construction of block predictor-corrector methods based on Runge-Kutta-Nyström correctors. Our approach is to apply the predictor-corrector method not only with stepsize h, but, in addition (and simultaneously) with stepsizes $a_i h, i = 1, \ldots, r$. In this way, at each step, a whole block of approximations to the exact solution at off-step points is computed. In the next step, these approximations are used to obtain a high-order predictor formula using Lagrange or Hermite interpolation. Since the block approximations at the off-step points can be computed in parallel, the sequential costs of these block predictor-corrector methods are comparable with those of a conventional predictor-corrector method. Furthermore, by using Runge-Kutta-Nyström corrector methods, the computation of the approximation at each off-step point is also highly parallel. By a number of widely-used test problems, a class of the resulting block predictor-corrector methods is compared with both sequential and parallel RKN methods available in the literature and is shown to demonstrate superior behaviour.

Key words: Runge-Kutta-Nyström methods, predictor-corrector methods, stability, parallelism.
AMS(MOS) subject classifications (1991): 65M12, 65M20
CR subject classifications: G.1.7

1 Introduction

Consider the numerical solution of nonstiff initial value problems (IVPs) for systems of special second-order, ordinary differential equations (ODEs)

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

where $\mathbf{y} : \mathbb{R} \to \mathbb{R}^d$, $\mathbf{f} : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$. Problems of the form (1.1) are encountered in e.g., celestial mechanics. A (simple) approach for solving this problem is to convert it into a system

*This work was supported by a three-month DAAD research grant

of first-order ODEs with double dimension and to apply e.g., a (parallel) Runge-Kutta-type method (RK-type method), ignoring the special form of (1.1) (the *indirect* approach). However, taking into account the fact that $\mathbf{f}$ does not depend on the first derivative, the use of a direct method tuned to the special form of (1.1) is usually more efficient (the *direct* approach). Such direct methods are generally known as Runge-Kutta-Nyström-type method (RKN-type method). Sequential explicit RKN methods up to order 10 can be found in [11, 12, 16, 19]. The performance of the tenth-order explicit RK method requiring 17 sequential $\mathbf{f}$-evaluations in [17] and the tenth-order explicit RKN method requiring only 11 sequential $\mathbf{f}$-evaluations in [19] is an example showing the advantage of the direct approach for sequential explicit RK and RKN methods. It is highly likely that in the class of parallel methods, the direct approach also leads to an improved efficiency.

In the literature, several classes of parallel explicit RKN-type methods have been investigated in [5, 4, 7, 8, 25]. A common challenge in these papers is to reduce, for a given order, the required number of sequential $\mathbf{f}$-evaluations per step, using parallel processors. In the present paper, we investigate a particular class of explicit RKN-type block predictor-corrector methods (PC methods) for use on parallel computers. Our approach consists of applying the PC method not only at step points, but also at off-step points (block points), so that, in each step, a whole block of approximations to the exact solutions is computed. This approach was firstly used in [10] for increasing reliability in explicit RK methods. It was also successfully applied in [21] for improving efficiency of RK-type PC methods. We shall use this approach to construct PC methods of RKN-type requiring few numbers of sequential $\mathbf{f}$-evaluations per step with acceptable stability properties. In this case as in [21], the block of approximations is used to obtain a highly accurate predictor formula in the next step by using Lagrange or Hermite interpolation. The precise location of the off-step points can be used for minimizing the interpolation errors and also for developing various cheap strategies for stepsize control. Since the approximations to the exact solutions at off-step points to be computed in each step can be obtained in parallel, the sequential costs of the resulting RKN-type block PC methods are equal to those of conventional PC methods. Furthermore, by using Runge-Kutta-Nyström corrector methods, the PC iteration method computing the approximation to the exact solution at each off-step point, itself is also highly parallel (cf. [5, 25]). The parallel RKN-type PC methods investigated in this paper may be considered as block versions of the parallel-iterated RKN methods (PIRKN methods) in [5, 25], and will therefore be termed *block PIRKN methods* (BPIRKN methods). Moreover, by using direct RKN correctors, we have obtained BPIRKN methods possessing both faster convergence and smaller truncation error resulting in better efficiency than by using indirect RKN correctors (cf. e.g., [5]).

In the next section, we shall formulate the block PIRKN methods. Furthermore, we consider order conditions for the predictor, convergence and stability boundaries, and the choice of block abscissas. In Section 3, we present numerical efficiency tests and comparisons with parallel and sequential explicit RKN methods available in the RKN literature.

In the following sections, for the sake of simplicity of notation, we assume that the IVP (1.1) is a scalar problem. However, all considerations below can be straightforwardly extended to a system of ODEs, and therefore, also to nonautonomous equations.

1.1 Direct and indirect RKN methods

A general s-stage RKN method for numerically solving the scalar problem (1.1) is defined by (see e.g., [22, 25], and also [1, p. 272])

$$\begin{aligned} \mathbf{U}_n &= u_n\mathbf{e} + hu'_n\mathbf{c} + h^2Af(\mathbf{U}_n), \\ u_{n+1} &= u_n + hu'_n + h^2\mathbf{b}^Tf(\mathbf{U}_n), \\ u'_{n+1} &= u'_n + h\mathbf{d}^Tf(\mathbf{U}_n), \end{aligned} \tag{1.2}$$

where $u_n \approx y(t_n)$, $u'_n \approx y'(t_n)$, h is the stepsize, $s \times s$ matrix A, s-dimensional vectors $\mathbf{b}, \mathbf{c}, \mathbf{d}$ are the method parameters matrix and vectors, $\mathbf{e}$ being the s-dimensional vector with unit entries (in the following, we will use the notation $\mathbf{e}$ for any vector with unit entries, and $\mathbf{e}_j$ for any jth unit vector, however, its dimension will always be clear from the context). Vector $\mathbf{U}_n$ denotes the stage vector representing numerical approximations to the exact solution vector $y(t_n\mathbf{e}+\mathbf{c}h)$ at the nth step. Furthermore, in (1.2), we use for any vector $\mathbf{v} = (v_1, \ldots, v_s)^T$ and any scalar function f, the notation $f(\mathbf{v}) := (f(v_1), \ldots, f(v_s))^T$. Similarly to a RK method, the RKN method (1.2) is also conveniently presented by the Butcher array (see e.g., [25], [1, p. 272])

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline & \mathbf{b}^T \\ & \mathbf{d}^T \end{array}$$

This RKN method will be referred to as the corrector method. We distinguish two types of RKN methods: *direct* and *indirect*. Indirect RKN methods are derived from RK methods for first-order ODEs. Writing (1.1) in first-order form, and applying a RK method with Butcher array

$$\begin{array}{c|c} \mathbf{c} & A_{RK} \\ \hline & \mathbf{b}^T_{RK} \end{array}$$

yield the indirect RKN method defined by

$$\begin{array}{c|c} \mathbf{c} & [A_{RK}]^2 \\ \hline & \mathbf{b}^T_{RK}A_{RK} \\ & \mathbf{b}^T_{RK} \end{array}$$

If the originating implicit RK method is of collocation type, then the resulting indirect implicit RKN will be termed *indirect collocation implicit RKN* method. Direct implicit RKN methods are directly constructed for second-order ODEs of the form (1.1). A first family of these direct implicit RKN methods is obtained by means of collocation technique (see [22]) and will be also called *direct collocation implicit RKN* method. In this paper, we will confine our considerations to collocation high-order implicit RKN methods that is the Gauss-Legendre and Radau IIA methods (briefly called direct or indirect Gauss-Legendre, Radau

IIA). This class contains methods of arbitrarily high order. Indirect collocation unconditionally stable methods can be found in [18]. Direct collocation conditionally stable methods were investigated in [22, 5]

1.2 Parallel explicit RKN methods

A first class of parallel explicit RKN methods called PIRKN is considered in [25]. These PIRKN methods are closely related to the block PIRKN methods to be considered in the next section. Using indirect collocation implicit RKN of the form (1.2) as corrector method, a general PIRKN method of [25] assumes the following form (see also [5])

$$\begin{aligned} \mathbf{U}_n^{(0)} &= y_n\mathbf{e} + hy_n'\mathbf{c}, \\ \mathbf{U}_n^{(j)} &= y_n\mathbf{e} + hy_n'\mathbf{c} + h^2 A f(\mathbf{U}_n^{(j-1)}), \qquad j = 1, \ldots, m, \\ y_{n+1} &= y_n + hy_n' + h^2\mathbf{b}^T f(\mathbf{U}_n^{(m)}), \\ y_{n+1}' &= y_n' + h\mathbf{d}^T f(\mathbf{U}_n^{(m)}), \end{aligned} \tag{1.3}$$

where, $y_n \approx y(t_n)$, $y_n' \approx y'(t_n)$. Let p be the order of the corrector (1.2), by setting $m = [(p-1)/2]$, $[\cdot]$ denoting the integer function, the PIRKN method (1.3) is indeed an explicit RKN method of order p with the Butcher array (cf. [5, 25])

$$\begin{array}{c|ccccc} \mathbf{0} & O & & & & \\ \mathbf{c} & A & O & & & \\ \mathbf{c} & O & A & O & & \\ \vdots & \vdots & & \ddots & \ddots & \\ \mathbf{c} & O & \ldots & O & A & O \\ \hline & \mathbf{0}^T & \ldots & \mathbf{0}^T & \mathbf{0}^T & \mathbf{b}^T \\ & \mathbf{0}^T & \ldots & \mathbf{0}^T & \mathbf{0}^T & \mathbf{d}^T \end{array} \tag{1.4}$$

where, O and $\mathbf{0}$ denote $s \times s$ matrix and s-dimensional vector with zero entries, respectively. From the Butcher array (1.4), it is clear that the PIRKN method (1.3) has the number of stages equal to $(m+1)\cdot s$. However, in each iteration, the evaluation of s components of vector $f(\mathbf{U}_n^{(j-1)})$ can be obtained in parallel, provided that we have an s-processor computer. Consequently, the number of sequential $\mathbf{f}$-evaluations equals $s^* = m + 1$. This class also contains the methods of arbitrarily high order and belongs to the set of efficient parallel methods for nonstiff problems of the form (1.1). For the detailed performance of these PIRKN methods, we refer to [25]. A further development of parallel RKN-type methods was considered in e.g., [4, 5, 7, 8]. Notice that for the PIRKN method, apart from parallelism across the methods and across the problems, it does not have any further parallelism. In Section 3 the PIRKN methods will be used in the numerical comparisons with the new block PIRKN methods.

2 Block PIRKN methods

Applying the RKN method (1.2) at t_n with r distinct stepsizes a_ih, where $i = 1, \ldots, r$ and $a_1 = 1$, we have

$$\begin{aligned} \mathbf{U}_{n,i} &= u_n\mathbf{e} + a_ihu_n'\mathbf{c} + a_i^2h^2Af(\mathbf{U}_{n,i}), \\ u_{n+1,i} &= u_n + a_ihu_n' + a_i^2h^2\mathbf{b}^Tf(\mathbf{U}_{n,i}), \\ u_{n+1,i}' &= u_n' + a_ih\mathbf{d}^Tf(\mathbf{U}_{n,i}), \qquad i = 1, \ldots, r. \end{aligned} \tag{2.1}$$

Let us suppose that at $(n-1)$th step, a block of predictions $\mathbf{U}^{(0)}_{n-1,i}, i = 1, \ldots, r$, and the approximations $y_{n-1} \approx y(t_{n-1})$, $y'_{n-1} \approx y'(t_{n-1})$ are given. We shall compute r approximations $y_{n,i}$ to the exact solutions $y(t_{n-1} + a_ih), i = 1, \ldots, r$, defined by

$$\begin{aligned} \mathbf{U}^{(j)}_{n-1,i} &= y_{n-1}\mathbf{e} + a_ihy'_{n-1}\mathbf{c} + a_i^2h^2Af(\mathbf{U}^{(j-1)}_{n-1,i}), \quad j = 1, \ldots, m_b, \\ y_{n,i} &= y_{n-1} + a_ihy'_{n-1} + a_i^2h^2\mathbf{b}^Tf(\mathbf{U}^{(m_b)}_{n-1,i}), \\ y'_{n,i} &= y'_{n-1} + a_ih\mathbf{d}^Tf(\mathbf{U}^{(m_b)}_{n-1,i}), \quad i = 1, \ldots, r. \end{aligned}$$

In the next step, these r approximations are used to create high-order predictors. By denoting

$$\begin{aligned} \mathbf{Y}_n &:= (y_{n,1}, \ldots, y_{n,r})^T, \qquad y_{n,1} = y_n, \\ \mathbf{Y}'_n &:= (y'_{n,1}, \ldots, y'_{n,r})^T, \qquad y'_{n,1} = y'_n, \end{aligned} \tag{2.2}$$

we can construct the following predictor formulas

$$\mathbf{U}^{(0)}_{n,i} = V_i\mathbf{Y}_n, \tag{2.3a}$$

$$\mathbf{U}^{(0)}_{n,i} = V_i\mathbf{Y}_n + hW_i\mathbf{Y}'_n \tag{2.3b}$$

$$\mathbf{U}^{(0)}_{n,i} = V_i\mathbf{Y}_n + hW_i\mathbf{Y}'_n + h^2\Lambda_if(\mathbf{Y}_n), \qquad i = 1, \ldots, r. \tag{2.3c}$$

where V_i, W_i and Λ_i are $s \times r$ extrapolation matrices which will be determined by order conditions (see Section 2.1). The predictors (2.3a), (2.3b) and (2.3c) are referred to as Lagrange, Hermite-I and Hermite-II, respectively. Apart from (2.3), we can construct predictors of other types like e.g., Adam type (cf. [21]). Regarding (2.1) as block corrector methods and (2.3) as block predictor methods for the stage vectors, we leave the class of one-step methods and arrive at a block PC method

$$\mathbf{U}^{(0)}_{n,i} = V_i\mathbf{Y}_n + h\theta^2W_i\mathbf{Y}'_n + h^2[\theta^2(1-\theta)/2]\Lambda_if(\mathbf{Y}_n) \tag{2.4a}$$

$$\begin{aligned} \mathbf{U}^{(j)}_{n,i} &= \mathbf{e}\mathbf{e}_1^T\mathbf{Y}_n + a_ih\mathbf{c}\mathbf{e}_1^T\mathbf{Y}'_n + a_i^2h^2Af(\mathbf{U}^{(j-1)}_{n,i}), \quad j = 1, \ldots, m, \\ y_{n+1,i} &= \mathbf{e}_1^T\mathbf{Y}_n + a_ih\mathbf{e}_1^T\mathbf{Y}'_n + a_i^2h^2\mathbf{b}^Tf(\mathbf{U}^{(m)}_{n,i}), \\ y'_{n+1,i} &= \mathbf{e}_1^T\mathbf{Y}'_n + a_ih\mathbf{d}^Tf(\mathbf{U}^{(m)}_{n,i}), \quad i = 1, \ldots, r, \qquad \theta \in \{0, 1, -1\}. \end{aligned} \tag{2.4b}$$

Notice that for a general presentation, in (2.4), the three different predictor formulas (2.3) have been combined in a common one (2.4a), where, $\theta = 0$, 1 and -1 respectively indicate

Lagrange, Hermite-I and Hermite-II predictor. With $\theta = -1$ the block PC method (2.4) is in $PE(CE)^m E$ mode and, with $\theta = 0$ or 1, the $PE(CE)^m E$ mode is reduced to $P(CE)^m E$ mode.
It can be seen that the block PC method (2.4) consists of a block of PIRKN-type corrections using a block of predictions at the off-step points (block points) (cf. Section 1.2). Therefore, we call the method (2.4) the *r-dimensional block PIRKN method (BPIRKN method)* (cf. [21]). In the case of Hermite-I predictor (2.3b), for $r = 1$, the BPIRKN method (2.4) really reduces to a PIRKN method of the form (1.3) studied in [5, 25].
Once the vectors $\mathbf{Y}_n$ and $\mathbf{Y}'_n$ are given, the r values $y_{n,i}$ can be computed in parallel and, on a second level, the components of the ith stage vector iterate $\mathbf{U}_{n,i}^{(j)}$ can also be evaluated in parallel (cf. also Section 1.2). Hence, the r-dimensional BPIRKN methods (2.4) based on s-stage RKN correctors can be implemented on a computer possessing $r \cdot s$ parallel processors. The number of sequential $\mathbf{f}$-evaluations per step of length h in each processor equals $s^* = m + \theta^2(1-\theta)/2 + 1$, where $\theta \in \{0, 1, -1\}$.

2.1 Order conditions for the predictor

In this section we consider order conditions for the predictors. For fixed stepsize h, the qth-order conditions for (2.4a) are derived by replacing $\mathbf{U}_{n,i}^{(0)}$ and $\mathbf{Y}_n$ by the exact solution values $y(t_n\mathbf{e} + a_i h\mathbf{c}) = y(t_{n-1}\mathbf{e} + h(a_i\mathbf{c} + \mathbf{e}))$ and $y(t_{n-1}\mathbf{e} + h\mathbf{a})$, respectively, with $\mathbf{a} = (a_1, \ldots, a_r)^T$. On substitution of these exact values into (2.4a) and by requiring that the residue is of order $q+1$ in h, we are led to

$$
\begin{aligned}
&y(t_{n-1}\mathbf{e} + h(a_i\mathbf{c} + \mathbf{e})) - V_i y(t_{n-1}\mathbf{e} + h\mathbf{a}) \\
&\qquad - \theta^2 h W_i y'(t_{n-1}\mathbf{e} + h\mathbf{a}) - [\theta^2(1-\theta)/2] h^2 \Lambda_i y''(t_{n-1}\mathbf{e} + h\mathbf{a}) = O(h^{q+1}) \\
&\qquad i = 1, \ldots, r.
\end{aligned} \tag{2.5}
$$

Using Taylor expansions, we can expand the left-hand side of (2.5) in powers of h and obtain

$$
\begin{aligned}
&\left[exp\left(h(a_i\mathbf{c}+\mathbf{e})\frac{d}{dt}\right) - \left(V_i + \theta^2 h W_i \frac{d}{dt} + [\theta^2(1-\theta)/2] h^2 \Lambda_i \frac{d^2}{dt^2}\right) exp\left(h\mathbf{a}\frac{d}{dt}\right)\right] y(t_{n-1}) \\
&\qquad = \sum_{j=0}^{q} \mathbf{C}_i^{(j)} \left(h\frac{d}{dt}\right)^j y(t_{n-1}) + \mathbf{C}_i^{(q+1)} \left(h\frac{d}{dt}\right)^{q+1} y(t_i^*) = O(h^{q+1}), \\
&\qquad i = 1, \ldots, r,
\end{aligned} \tag{2.6}
$$

where, t_i^* is a suitably chosen point in the interval $[t_{n-1}, t_{n-1} + (1 + a_i)h]$, and

$$
\begin{aligned}
\mathbf{C}_i^{(j)} &= \frac{1}{j!}\left[(a_i\mathbf{c} + \mathbf{e})^j - V_i\mathbf{a}^j - j\theta^2 W_i \mathbf{a}^{j-1} - j(j-1)[\theta^2(1-\theta)/2]\Lambda_i \mathbf{a}^{j-2}\right], \\
j &= 0, \ldots, q, \qquad i = 1, \ldots, r.
\end{aligned} \tag{2.7a}
$$

The vectors $\mathbf{C}_i^{(j)}$, $i = 1, \ldots, r$ represent the error vectors of the block predictors (2.4a). From (2.6), we obtain the order conditions

$$
\mathbf{C}_i^{(j)} = \mathbf{0}, \qquad j = 0, 1, \ldots, q, \qquad i = 1, \ldots, r. \tag{2.7b}
$$

The vectors $\mathbf{C}_i^{(q+1)}$, $i = 1, \ldots, r$, are the *principal* error vectors of the block predictors. The conditions (2.7), imply that

$$\mathbf{U}_{n,i} - \mathbf{U}_{n,i}^{(0)} = O(h^{q+1}), \qquad i = 1, \ldots, r. \tag{2.8}$$

Since each iteration raises the order of the iteration error by 2, the following order relations are obtained

$$\begin{aligned}
\mathbf{U}_{n,i} - \mathbf{U}_{n,i}^{(m)} &= O(h^{2m+q+1}), \\
u_{n+1,i} - y_{n+1,i} &= a_i^2 h^2 \mathbf{b}^T \big[f(\mathbf{U}_{n,i}) - f(\mathbf{U}_{n,i}^{(m)})\big] = O(h^{2m+q+3}), \\
u'_{n+1,i} - y'_{n+1,i} &= a_i h \mathbf{d}^T \big[f(\mathbf{U}_{n,i}) - f(\mathbf{U}_{n,i}^{(m)})\big] = O(h^{2m+q+2}), \\
i &= 1, \ldots, r.
\end{aligned}$$

Furthermore, for the local truncation error of the BPIRKN method (2.4), we may write

$$\begin{aligned}
y(t_{n+1}) - y_{n+1} &= \big[y(t_{n+1}) - u_{n+1}\big] + \big[u_{n+1} - y_{n+1}\big] = O(h^{p+1}) + O(h^{2m+q+3}), \\
y'(t_{n+1}) - y'_{n+1} &= \big[y'(t_{n+1}) - u'_{n+1}\big] + \big[u'_{n+1} - y'_{n+1}\big] = O(h^{p+1}) + O(h^{2m+q+2}),
\end{aligned}$$

where, p is the order of the generating RKN corrector (1.2). Thus, we have the theorem

Theorem 2.1 *If the conditions (2.7) are satisfied and if the generating RKN corrector (1.2) has order p, then the BPIRKN method (2.4) has the order $p^* = min\{p, p_{iter}\}$, where $p_{iter} = 2m + q + 1$.*

In order to express V_i, W_i and Λ_i explicitly in terms of vectors $\mathbf{a}$ and $\mathbf{c}$, we suppose that $q = [1 + \theta^2 + \theta^2(1-\theta)/2]r - 1$, $\theta \in \{0, 1, -1\}$, and for $i = 1, \ldots, r$, define the matrices

$$\begin{aligned}
P_i &:= \Big(\mathbf{e}, (a_i\mathbf{c} + \mathbf{e}), (a_i\mathbf{c} + \mathbf{e})^2, \ldots, (a_i\mathbf{c} + \mathbf{e})^{r-1}\Big), \\
Q &:= \Big(\mathbf{e}, \mathbf{a}, \mathbf{a}^2, \mathbf{a}^3, \ldots, \mathbf{a}^{r-1}\Big), \\
R &:= \Big(\mathbf{0}, \mathbf{e}, 2\mathbf{a}, 3\mathbf{a}^2, \ldots, (r-1)\mathbf{a}^{r-2}\Big), \\
S &:= \Big(\mathbf{0}, \mathbf{0}, 2\mathbf{e}, 6\mathbf{a}, 12\mathbf{a}^2, \ldots, (r-1)(r-2)\mathbf{a}^{r-3}\Big),
\end{aligned}$$

$$\begin{aligned}
P_i^* &:= \Big((a_i\mathbf{c} + \mathbf{e})^r, (a_i\mathbf{c} + \mathbf{e})^{r+1}, \ldots, (a_i\mathbf{c} + \mathbf{e})^{2r-1}\Big), \\
Q^* &:= \Big(\mathbf{a}^r, \mathbf{a}^{r+1}, \mathbf{a}^{r+2}, \ldots, \mathbf{a}^{2r-1}\Big), \\
R^* &:= \Big(r\mathbf{a}^{r-1}, (r+1)\mathbf{a}^r, \ldots, (2r-1)\mathbf{a}^{2r-2}\Big), \\
S^* &:= \Big(r(r-1)\mathbf{a}^{r-2}, (r+1)r\mathbf{a}^{r-1}, \ldots, (2r-1)(2r-2)\mathbf{a}^{2r-3}\Big),
\end{aligned}$$

$$
\begin{aligned}
P_i^{**} &:= \Big((a_i\mathbf{c}+\mathbf{e})^{2r}, (a_i\mathbf{c}+\mathbf{e})^{2r+1}, \ldots, (a_i\mathbf{c}+\mathbf{e})^{q}\Big),\\
Q^{**} &:= \Big(\mathbf{a}^{2r}, \mathbf{a}^{2r+1}, \mathbf{a}^{2r+2}, \ldots, \mathbf{a}^{q}\Big),\\
R^{**} &:= \Big(2r\mathbf{a}^{2r-1}, (2r+1)\mathbf{a}^{2r}, \ldots, q\mathbf{a}^{q-1}\Big),\\
S^{**} &:= \Big(2r(2r-1)\mathbf{a}^{2r-2}, (2r+1)2r\mathbf{a}^{2r-1}, \ldots, q(q-1)\mathbf{a}^{q-2}\Big),
\end{aligned}
$$

where, for $\theta = 0$, the matrices $P_i^*, Q^*, R^*, S^*, P_i^{**}, Q^{**}, R^{**}, S^{**}$ are assumed to be zero, and for $\theta = 1$, only $P_i^{**}, Q^{**}, R^{**}, S^{**}$ are assumed to be zero matrices. Then the order conditions (2.7) can be presented in the form

$$
\begin{aligned}
&P_i - V_iQ - \theta^2 W_iR - [\theta^2(1-\theta)/2]\Lambda_i S = O,\\
&P_i^* - V_iQ^* - \theta^2 W_iR^* - [\theta^2(1-\theta)/2]\Lambda_i S^* = O,\\
&P_i^{**} - V_iQ^{**} - \theta^2 W_iR^{**} - [\theta^2(1-\theta)/2]\Lambda_i S^{**} = O.
\end{aligned}
\tag{2.9}
$$

Since the components a_i are assumed to be distinct implying that Q is nonsingular, and from (2.9), for $i = 1, \ldots, r$, we may write

$$
\begin{aligned}
V_i &= \big[P_i - \theta^2 W_iR - [\theta^2(1-\theta)/2]\Lambda_i S\big]Q^{-1}\\
\theta^2 W_i &= \big[[\theta^2(1-\theta)/2]\Lambda_i(S^* - SQ^{-1}Q^*) + (P_iQ^{-1}Q^* - P_i^*)\big]\\
&\quad\times \big[RQ^{-1}Q^* - R^*\big]^{-1}\\
[\theta^2(1-\theta)/2]\Lambda_i &= \big[(P_i^{**} - P_iQ^{-1}Q^{**}) + (P_iQ^{-1}Q^* - P_i^*)(RQ^{-1}Q^* - R^*)^{-1}(RQ^{-1}Q^{**} - R^{**})\big]\\
&\quad\times \big[(SQ^{-1}Q^* - S^*)(RQ^{-1}Q^* - R^*)^{-1}(RQ^{-1}Q^{**} - R^{**}) + (S^{**} - SQ^{-1}Q^{**})\big]^{-1},
\end{aligned}
\tag{2.10}
$$

where the matrices $(SQ^{-1}Q^* - S^*)(RQ^{-1}Q^* - R^*)^{-1}(RQ^{-1}Q^{**} - R^{**}) + (S^{**} - SQ^{-1}Q^{**})$ and $RQ^{-1}Q^* - R^*$ are assumed to be nonsingular. In view of Theorem 2.1 and the explicit expressions of the predictor matrices V_i, W_i and Λ_i, in (2.10), we have the following theorem:

Theorem 2.2 *If $q = [1 + \theta^2 + \theta^2(1-\theta)/2]r - 1$, and the predictor matrices V_i, W_i, Λ_i, $i = 1, \ldots, r$ satisfy the relations (2.10), then for the BPIRKN methods (2.4), $p_{iter} = [1 + \theta^2 + \theta^2(1-\theta)/2]r + 2m$, $p^* = min\{p, p_{iter}\}$ and $s^* = m + \theta^2(1-\theta)/2 + 1$, $\theta \in \{0, 1, -1\}$.*

In the application of BPIRKN methods, we have some natural combinations of the predictors (2.4a) with Gauss-Legendre and Radau IIA correctors. Using Lagrange and Hermite-I predictors has the advantage of possessing no additional $\mathbf{f}$-evaluations in the predictor. An important disadvantage of using Lagrange predictor is that, for a given order q of predictor formulas, its block dimension is two times larger than that of using Hermite-I, yielding a double number of processors needed for the implementation of BPIRKN methods. However, recent developments indicate that the number of processors is no longer an important issue. In this paper, we concentrate our considerations on the Hermite-I predictors. In the near future, we intend to compare BPIRKN methods employing Lagrange and Hermite-II predictors.

2.2 Convergence boundaries

In actual implementation of BPIRKN methods, the number of iterations m is determined by some iteration strategy, rather than by order conditions using minimal number of iterations to reach to order of the corrector. Therefore, it is of interest to know how the integration step effects the rate of convergence. The stepsize should be such that a reasonable convergence speed is achieved.
As in e.g., [5], we shall determine the rate of convergence by using the model test equation $y''(t) = \lambda y(t)$, where λ runs through the spectrum of the Jacobian matrix $\partial \mathbf{f}/\partial \mathbf{y}$. For this equation, we obtain the iteration error equation

$$\mathbf{U}_{n,i}^{(j)} - \mathbf{U}_{n,i} = a_i^2 zA\big[\mathbf{U}_{n,i}^{(j-1)} - \mathbf{U}_{n,i}\big], \quad z := h^2\lambda, \quad j = 1, \ldots, m. \tag{2.11}$$

Hence, with respect to the model test equation, the convergence factor is determined by the spectral radius $\rho(a_i^2 zA)$ of the iteration matrix $a_i^2 zA$, $i = 1, \ldots, r$. Requiring that $\rho(a_i^2 zA) < 1$, leads us to the convergence condition

$$a_i^2|z| < \frac{1}{\rho(A)} \quad \text{or} \quad a_i^2 h^2 < \frac{1}{\rho(\partial \mathbf{f}/\partial \mathbf{y})\rho(A)}. \tag{2.12}$$

We shall call $1/\rho(A)$ the *convergence boundary.* In actual computation, the integration

Table 2.1: Convergence boundaries $\gamma(\alpha)$ for various BPIRKN methods based on p-order correctors

Correctors	$p = 3$	$p = 4$	$p = 5$	$p = 6$	$p = 7$	$p = 8$	$p = 9$	$p = 10$
Indirect Gauss-Legendre		12.04α		21.73α		37.03α		52.63α
Direct Gauss-Legendre		20.83α		44.48α		55.54α		76.92α
Indirect Radau IIA	5.89α		13.15α		25.64α		40.0α	
Direct Radau IIA	10.41α		20.40α		37.03α		55.55α	

stepsize h should be substantial smaller than allowed by condition (2.12). By requiring that $\rho(a_i^2 zA)$ is less than a given damping factor α, ($\alpha \ll 1$), we are led to the condition

$$a_i^2|z| \le \gamma(\alpha) \quad \text{or} \quad a_i^2 h^2 \le \frac{\gamma(\alpha)}{\rho(\partial \mathbf{f}/\partial \mathbf{y})}, \quad \gamma(\alpha) = \frac{\alpha}{\rho(A)}, \tag{2.13}$$

where, $\gamma(\alpha)$ presents the boundary of convergence with damping factor α of the method. Table 2.1 lists the convergence boundaries $\gamma(\alpha)$ of the BPIRKN methods based on indirect and direct collocation Gauss-Legendre and Radau IIA RKN correctors of orders up to 10 (cf. e.g., [5, 22] and also Section 1.1). Notice that for a given stepsize h, the maximal damping factor is defined by

$$\alpha = \frac{a_i^2 h^2 \rho(\partial \mathbf{f}/\partial \mathbf{y})}{\gamma(1)},$$

so we can conclude that the direct collocation RKN correctors reported in Table 2.1 give rise to faster convergence. This conclusion leads us to restrict our considerations to the BPIRKN methods based on direct collocation RKN correctors (cf. [5]). These direct RKN methods are not A-stable (see [22]), but their stability regions are sufficiently large for nonstiff problems.

2.3 The choice of block abscissas a_i

The accuracy of Hermite-I interpolation formulas is improved if the interpolation abscissas are more narrowly spaced. However, this will increase the magnitude of the entries of the matrices V_i and W_i, causing serious round-off errors. There are several ways to reduce this round-off effect as were discussed in [21, Section 2.1] for Lagrange interpolation formulas. Also in [10] where Hermite interpolation formulas were used for deriving reliable error estimates for defect control, it was found that on a 15 digits precision machine, the interpolation abscissas should be separated by 0.2 in order to suppress rounding errors.
In order to derive a further criteria for the choice of suitable values of the abscissas a_i, we need to get insight into the propagation of a perturbation $\boldsymbol{\varepsilon}$ of the block vectors $\mathbf{Y}_n$ and $\mathbf{Y}'_n$ within a single step (the similar analysis was given in [21]). We shall study this for the model test equation $y''(t) = \lambda y(t)$. First we shall express $y_{n+1,i}$ and $hy'_{n+1,i}$ in terms of $\mathbf{Y}_n$ and $h\mathbf{Y}'_n$. Since

$$\begin{aligned}
\mathbf{U}_{n,i} &= [I - a_i^2 zA]^{-1}[\mathbf{e}\mathbf{e}_1^T\mathbf{Y}_n + \mathbf{c}\mathbf{e}_1^T a_i h\mathbf{Y}'_n] \\
\mathbf{U}_{n,i}^{(0)} - \mathbf{U}_{n,i} &= \left[V_i - [I - a_i^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T\right]\mathbf{Y}_n + \left[W_i - [I - a_i^2 zA]^{-1}\mathbf{c}\mathbf{e}_1^T a_i\right]h\mathbf{Y}'_n,
\end{aligned}$$

applying (2.4), (2.11) to the model equation for a given number m, we obtain

$$\begin{aligned}
y_{n+1,i} &= \mathbf{e}_1^T\mathbf{Y}_n + a_i h\mathbf{e}_1^T\mathbf{Y}'_n + a_i^2 z\mathbf{b}^T[\mathbf{U}_{n,i}^{(m)} - \mathbf{U}_{n,i}] + a_i^2 z\mathbf{b}^T\mathbf{U}_{n,i} \\
&= \mathbf{e}_1^T\mathbf{Y}_n + a_i h\mathbf{e}_1^T\mathbf{Y}'_n + a_i^2 z\mathbf{b}^T[I - a_i^2 zA]^{-1}[\mathbf{e}\mathbf{e}_1^T\mathbf{Y}_n + \mathbf{c}\mathbf{e}_1^T a_i h\mathbf{Y}'_n] \\
&+ a_i^2 z\mathbf{b}^T[a_i^2 zA]^m\left[\left[V_i - [I - a_i^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T\right]\mathbf{Y}_n + \left[W_i - [I - a_i^2 zA]^{-1}\mathbf{c}\mathbf{e}_1^T a_i\right]h\mathbf{Y}'_n\right] \\
&= \left[\mathbf{e}_1^T + a_i^2 z\mathbf{b}^T[I - a_i^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T + a_i^2 z\mathbf{b}^T[a_i^2 zA]^m\left[V_i - [I - a_i^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T\right]\right]\mathbf{Y}_n \\
&+ \left[a_i\mathbf{e}_1^T + a_i^2 z\mathbf{b}^T[I - a_i^2 zA]^{-1}a_i\mathbf{c}\mathbf{e}_1^T + a_i^2 z\mathbf{b}^T[a_i^2 zA]^m\left[W_i - [I - a_i^2 zA]^{-1}a_i\mathbf{c}\mathbf{e}_1^T\right]\right]h\mathbf{Y}'_n
\end{aligned} \tag{2.14a}$$

$$\begin{aligned}
hy'_{n+1,i} &= \mathbf{e}_1^T h\mathbf{Y}'_n + +a_i z\mathbf{d}^T[\mathbf{U}_{n,i}^{(m)} - \mathbf{U}_{n,i}] + a_i z\mathbf{d}^T\mathbf{U}_{n,i} \\
&= h\mathbf{e}_1^T\mathbf{Y}'_n + a_i z\mathbf{d}^T[I - a_i^2 zA]^{-1}[\mathbf{e}\mathbf{e}_1^T\mathbf{Y}_n + \mathbf{c}\mathbf{e}_1^T a_i h\mathbf{Y}'_n] \\
&+ a_i z\mathbf{d}^T[a_i^2 zA]^m\left[\left[V_i - [I - a_i^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T\right]\mathbf{Y}_n + \left[W_i - [I - a_i^2 zA]^{-1}\mathbf{c}\mathbf{e}_1^T a_i\right]h\mathbf{Y}'_n\right] \\
&= \left[a_i z\mathbf{d}^T[I - a_i^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T + a_i z\mathbf{d}^T[a_i^2 zA]^m\left[V_i - [I - a_i^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T\right]\right]\mathbf{Y}_n \\
&+ \left[\mathbf{e}_1^T + a_i z\mathbf{d}^T[I - a_i^2 zA]^{-1}a_i\mathbf{c}\mathbf{e}_1^T + a_i z\mathbf{d}^T[a_i^2 zA]^m\left[W_i - [I - a_i^2 zA]^{-1}a_i\mathbf{c}\mathbf{e}_1^T\right]\right]h\mathbf{Y}'_n.
\end{aligned} \tag{2.14b}$$

Let us now replace $\mathbf{Y}_n$ by $\mathbf{Y}^*_n = \mathbf{Y}_n + \boldsymbol{\varepsilon}$ and $\mathbf{Y}'_n$ by $\mathbf{Y}'^*_n = \mathbf{Y}'_n + \boldsymbol{\varepsilon}$. Then from (2.14), the perturbed values $y^*_{n+1,i}$ and $y'^*_{n+1,i}$ of $y_{n+1,i}$ and $y'_{n+1,i}$, respectively are given by

$$
\begin{aligned}
y^*_{n+1,i} = y_{n+1,i} &+ \Big[\mathbf{e}_1^T + a_i^2 z \mathbf{b}^T [I - a_i^2 zA]^{-1} \mathbf{e}\mathbf{e}_1^T + a_i^2 z\mathbf{b}^T [a_i^2 zA]^m \left[V_i - [I - a_i^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T\right]\Big]\varepsilon \\
&+ \Big[a_i\mathbf{e}_1^T + a_i^2 z \mathbf{b}^T [I - a_i^2 zA]^{-1} a_i\mathbf{c}\mathbf{e}_1^T + a_i^2 z\mathbf{b}^T [a_i^2 zA]^m \left[W_i - [I - a_i^2 zA]^{-1}a_i\mathbf{c}\mathbf{e}_1^T\right]\Big]h\varepsilon
\end{aligned} \tag{2.15a}
$$

$$
\begin{aligned}
y'^*_{n+1,i} = y'_{n+1,i} &+ \frac{1}{h}\Big[a_i z\mathbf{d}^T [I - a_i^2 zA]^{-1} \mathbf{e}\mathbf{e}_1^T + a_i z\mathbf{d}^T [a_i^2 zA]^m \left[V_i - [I - a_i^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T\right]\Big]\varepsilon \\
&+ \Big[\mathbf{e}_1^T + a_i z \mathbf{d}^T [I - a_i^2 zA]^{-1} a_i\mathbf{c}\mathbf{e}_1^T + a_i z\mathbf{d}^T [a_i^2 zA]^m \left[W_i - [I - a_i^2 zA]^{-1}a_i\mathbf{c}\mathbf{e}_1^T\right]\Big]\varepsilon.
\end{aligned} \tag{2.15b}
$$

These relations show that the first component of the perturbation $\boldsymbol{\varepsilon}$ is amplified by a factor $O(1)$ for both $\mathbf{Y}_n$ and $\mathbf{Y}'_n$, whereas all other components are amplified by a factor of $O(h^{2m+2})$ and $O(h^{2m+1})$ for $\mathbf{Y}_n$ and $\mathbf{Y}'_n$, respectively. Refering to the approach used in [21], leads us to the choice of the values a_i such that the maximum norm of the principal error vector $\mathbf{C}_1^{(q+1)}$ in (2.7) is minimized.

In our case of Hermite-I predictors where $q = 2r - 1$ (cf. Theorem 2.2), we have to minimize the magnitude of $\|\mathbf{C}_1^{(2r)}\|_\infty$. Although we may use (2.7a) for minimizing $\|\mathbf{C}_1^{(2r)}\|_\infty$, it is more convenient to start with usual expression of Hermite interpolation formulas. For $2r$-times continuously differentiable function $y(t)$, the r-point Hermite interpolation formula (for our Hermite-I case) can be written in the form (see e.g., [20, p. 261], [26, p. 52], and cf. (2.6))

$$
\begin{aligned}
y(t_{n-1} + \tau h) = &\sum_{i=1}^{r} v_i(\tau) y(t_{n-1} + a_i h) \\
&+ h\sum_{i=1}^{r} w_i(\tau) y'(t_{n-1} + a_i h) + C^{(2r)}(\tau)\left(h\frac{d}{dt}\right)^{2r} y(t^*_\tau),
\end{aligned} \tag{2.16}
$$

where, $v_i(\tau)$, $w_i(\tau)$ and $C^{(2r)}(\tau)$ are the scalar polynomials defined by

$$
\begin{aligned}
&v_i(\tau) = l_i^2(\tau)\big(1 - 2l_i'(a_i)(\tau - a_i)\big), \qquad w_i(\tau) = l_i^2(\tau)(\tau - a_i) \\
&l_i(\tau) = \prod_{j=1, j\neq i}^{r} \frac{(\tau - a_j)}{(a_i - a_j)}, \qquad C^{(2r)}(\tau) = \frac{1}{(2r)!}\prod_{j=1}^{r}(\tau - a_j)^2
\end{aligned} \tag{2.17}
$$

and t^*_τ is a suitably chosen point in the interval $[t_{n-1}, t_{n-1} + \tau h]$. Hence, we have the following alternative form of (2.6) (with $\theta = 1$)

$$y(t_n + a_i c_k h) = y(t_{n-1} + (1 + a_i c_k)h) = \sum_{j=1}^{r} v_j(1 + a_i c_k) y(t_{n-1} + a_j h) \tag{2.18}$$

$$+ h \sum_{j=1}^{r} w_j(1 + a_i c_k) y'(t_{n-1} + a_j h) + C^{(2r)}(1 + a_i c_k) \Big(h \frac{d}{dt} \Big)^{2r} y(t^*_{ik}),$$

$$k = 1, \ldots, s, \qquad i = 1, \ldots, r,$$

where, t^*_{ik} is also a suitably chosen point in the interval $[t_{n-1}, t_{n-1} + (1 + a_i)h]$. The principal error vectors of the Hermite-I predictor formulas defined by (2.7a) are given by $\mathbf{C}_i^{(2r)} = C^{(2r)}(\mathbf{e} + a_i \mathbf{c})$, $i = 1, \ldots, r$. Recalling that for $i = 1$ we are led to minimize the magnitude of the values

$$C^{(2r)}(1 + c_k) = \frac{1}{(2r)!} \prod_{j=1}^{r} (1 + c_k - a_j)^2, \qquad k = 1, \ldots, s. \tag{2.19}$$

Confining our considerations to the block dimensions $r = s + 1$, we set

$$a_1 = 1, \qquad a_i = 1 + c_{i-1}, \qquad i = 2, \ldots, s + 1, \tag{2.20a}$$

resulting in predictors of (local) order $2s + 1$. By this choice, the principal error vector $\mathbf{C}_1^{(2r)}$ vanishes (i.e., $\|\mathbf{C}_1^{(2r)}\|_\infty = \|C^{(2r)}(\mathbf{e} + \mathbf{c})\|_\infty = 0$), so that all inaccuracies introduced by the predictor formula are damped by a factor of $O(h^{2m+2})$ for $\mathbf{Y}_n$ and by a factor of $O(h^{2m+1})$ for $\mathbf{Y}'_n$ (cf. (2.15)). However, for high-order methods, the choice (2.20a) can violate the limitation on the minimal spacing allowed by 0.2 found in [10] with respect to the abscissas a_1 and a_2. Moreover, the large block dimensions often cause a drop in stability. Therefore limiting the block dimensions to $r = s$, we propose the following second choice of abscissas a_i

$$a_1 = 1, \qquad a_i = 1 + c_i, \qquad i = 2, \ldots, s, \tag{2.20b}$$

resulting in predictors of (local) order $2s - 1$. By the choice (2.20b), the maximum norm

Table 2.2: Values of $\|\mathbf{C}_1^{(2r)}\|_\infty$ for various pth-order BPIRKN methods based on abscissas defined by (2.20b)

Methods	$p = 3$	$p = 4$	$p = 5$	$p = 6$	$p = 7$	$p = 8$	$p = 9$	$p = 10$
$\|\mathbf{C}_1^{(2r)}\|_\infty$	$2.5E-2$	$7.4E-3$	$6.9E-4$	$1.9E-4$	$1.4E-5$	$3.6E-6$	$2.1E-7$	$5.5E-8$

of the principal error vector $\mathbf{C}_1^{(2r)}$ is evaluted by

$$\|\mathbf{C}_1^{(2r)}\|_\infty = |C^{(2r)}(1 + c_1)| = \frac{1}{(2r)!} \prod_{i=1}^{r} (1 + c_1 - a_i)^2$$

is significantly small (see Table 2.2) and the limitation on minimal spacing is satisfied. We also hope that the stability of the resulting class of BPIRKN methods will be improved (see Section 2.4). Finally, we remark that BPIRKN methods defined by the choice of abscissas given by (2.20) attain order of the correctors for $m \geq 0$ (cf. Theorem 2.2).

2.4 Stability boundaries

The linear stability of the BPIRKN methods (2.4) is investigated by again using the model test equation $y''(t) = \lambda y(t)$, where λ is assumed to be negative. From (2.14) we are led to the recursion

$$\begin{pmatrix} \mathbf{Y}_{n+1} \\ h\mathbf{Y}'_{n+1} \end{pmatrix} = M_m(z) \begin{pmatrix} \mathbf{Y}_n \\ h\mathbf{Y}'_n \end{pmatrix}, \qquad M_m(z) = M_m^0(z) + M_m^1(z), \tag{2.21a}$$

where $M_m^0(z)$ and $M_m^1(z)$ are the $2r \times 2r$ matrices defined by

$$M_m^0(z) = \begin{pmatrix} \mathbf{e}_1^T + a_1^2 z\mathbf{b}^T[I - a_1^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T + a_1^2 z\mathbf{b}^T[a_1^2 zA]^m\left[V_1 - [I - a_1^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T\right] & \mathbf{0}^T \\ \vdots & \vdots \\ \mathbf{e}_1^T + a_r^2 z\mathbf{b}^T[I - a_r^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T + a_r^2 z\mathbf{b}^T[a_r^2 zA]^m\left[V_r - [I - a_r^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T\right] & \mathbf{0}^T \\ a_1 z\mathbf{d}^T[I - a_1^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T + a_1 z\mathbf{d}^T[a_1^2 zA]^m\left[V_1 - [I - a_1^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T\right] & \mathbf{0}^T \\ \vdots & \vdots \\ a_r z\mathbf{d}^T[I - a_r^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T + a_r z\mathbf{d}^T[a_r^2 zA]^m\left[V_r - [I - a_r^2 zA]^{-1}\mathbf{e}\mathbf{e}_1^T\right] & \mathbf{0}^T \end{pmatrix}, \tag{2.21b}$$

$$M_m^1(z) = \begin{pmatrix} \mathbf{0}^T & a_1\mathbf{e}_1^T + a_1^2 z\mathbf{b}^T[I - a_1^2 zA]^{-1}a_1\mathbf{c}\mathbf{e}_1^T + a_1^2 z\mathbf{b}^T[a_1^2 zA]^m\left[W_1 - [I - a_1^2 zA]^{-1}a_1\mathbf{c}\mathbf{e}_1^T\right] \\ \vdots & \vdots \\ \mathbf{0}^T & a_r\mathbf{e}_1^T + a_r^2 z\mathbf{b}^T[I - a_r^2 zA]^{-1}a_r\mathbf{c}\mathbf{e}_1^T + a_r^2 z\mathbf{b}^T[a_r^2 zA]^m\left[W_r - [I - a_r^2 zA]^{-1}a_r\mathbf{c}\mathbf{e}_1^T\right] \\ \mathbf{0}^T & \mathbf{e}_1^T + a_1 z\mathbf{d}^T[I - a_1^2 zA]^{-1}a_1\mathbf{c}\mathbf{e}_1^T + a_1 z\mathbf{d}^T[a_1^2 zA]^m\left[W_1 - [I - a_1^2 zA]^{-1}a_1\mathbf{c}\mathbf{e}_1^T\right] \\ \vdots & \vdots \\ \mathbf{0}^T & \mathbf{e}_1^T + a_r z\mathbf{d}^T[I - a_r^2 zA]^{-1}a_r\mathbf{c}\mathbf{e}_1^T + a_r z\mathbf{d}^T[a_r^2 zA]^m\left[W_r - [I - a_r^2 zA]^{-1}a_r\mathbf{c}\mathbf{e}_1^T\right] \end{pmatrix}. \tag{2.21c}$$

The $2r \times 2r$ matrix $M_m(z)$ defined by (2.21) which determines the stability of the BPIRKN methods, will be called the *amplification matrix*, its spectral radius $\rho\big(M_m(z)\big)$ the *stability function*. For a given number m, the stability intervals of the BPIRKN methods are defined by

$$\big(-\beta(m), 0\big) := \big\{z : \rho\big(M_m(z)\big) < 1, z \leq 0\big\}.$$

Table 2.3: Stability boundaries $\beta(m)$ for various pth-order BPIRKN methods based on abscissas defined by (2.20a)

BPIRKN methods	$p=3$	$p=4$	$p=5$	$p=6$	$p=7$	$p=8$	$p=9$	$p=10$
$\beta(0)(s^*=1)$	0.067	0.122	0.006	0.014	0.001	0.007	0.000	0.001
$\beta(1)(s^*=2)$	0.285	0.015	0.226	1.103	0.393	0.234	0.298	0.241
$\beta(2)(s^*=3)$	0.847	0.104	0.443	0.150	0.319	1.294	0.289	2.055
$\beta(3)(s^*=4)$	0.935	0.678	0.915	1.816	0.615	0.871	0.452	1.343
$\beta(4)(s^*=5)$	1.096	0.437	1.547	0.809	1.104	2.015	0.806	1.617
$\beta(5)(s^*=6)$	1.996	0.668	1.686	1.413	1.792	2.180	1.331	2.272
$\beta(6)(s^*=7)$	1.400	1.042	1.877	4.279	3.465	3.969	2.035	3.164

It is evident from (2.21) that if z satisfies the convergence conditions (2.12), then the stability function of the BPIRKN method $\rho\big(M_m(z)\big)$ converges to the stability function of the RKN corrector method as $m \to \infty$ (cf. e.g., [22], also [1, p. 273]). Hence, the *asymptotic* stability interval for $m \to \infty$, $(-\beta(\infty), 0)$, is the intersection on the negative $z-$axis of the stability interval $(-\beta_{corr}, 0)$ of the RKN corrector and its convergence region defined by (2.12). We numerically calculated the values of $\beta(m)$ for various resulting BPIRKN methods. Table 2.3 and Table 2.4 list these stability boundaries $\beta(m)$ for two classes of

Table 2.4: Stability boundaries $\beta(m)$ for various p-order BPIRKN methods based on abscissas defined by (2.20b)

BPIRKN methods	$p=3$	$p=4$	$p=5$	$p=6$	$p=7$	$p=8$	$p=9$	$p=10$
$\beta(0)(s^*=1)$	0.488	0.840	0.006	0.020	0.016	0.188	0.003	0.054
$\beta(1)(s^*=2)$	1.065	0.013	0.609	1.058	0.188	0.213	0.312	0.424
$\beta(2)(s^*=3)$	1.642	0.738	0.877	0.142	0.585	0.932	0.523	3.210
$\beta(3)(s^*=4)$	1.837	0.670	1.554	1.360	0.969	2.216	0.702	1.408
$\beta(4)(s^*=5)$	1.969	0.355	2.446	0.750	1.570	1.140	1.140	2.265
$\beta(5)(s^*=6)$	2.008	0.785	2.359	1.367	2.370	3.818	1.764	2.954
$\beta(6)(s^*=7)$	2.131	1.027	2.502	5.868	4.413	5.031	2.567	3.985

so-called BPIRKN-A and BPIRKN-B methods defined by the relations (2.20a) and (2.20b), respectively. From these tables we observe that the stability boundaries show a rather irregular behaviour, especially for the BPIRKN methods based on direct Gauss-Legendre correctors. Moreover, the results reported in these two tables also show the better stability behaviour of the class of BPIRKN-B methods defined by the choice of abscissas given by

(2.20b). The stability regions of this class are also sufficiently large for nonstiff problems. From Table 2.4, we can select a whole set of BPIRKN-B methods of order p up to 10 (except for $p = 4$) using only 1 correction with acceptable stability regions for nonstiff problems (cf. Theorem 2.2). In the next section we shall compare the efficiency of BPIRKN-A methods with that of BPIRKN-B methods by numerical tests. In all numerical experiments, we shall confine our considerations to the BPIRKN methods using Radau IIA correctors.

3 Numerical experiments

In this section we report numerical results obtained by the BPIRKN methods (2.4). As it was mentioned in the previous sections, we confine our considerations to the BPIRKN methods based on Hermite-I predictor and direct Radau IIA corrector pairs with block points defined by (2.20) (cf. Sections 2.2 and Section 2.3). First, we test the efficiency of two classes of BPIRKN-A and BPIRKN-B methods (cf. Section 2.4). Next, we compare the best resulting class of BPIRKN methods with parallel and sequential explicit RKN methods from the literature. In order to see the efficiency (partly) defined by convergence behaviour of the various PC methods, we applied a dynamical strategy for determining the number of iterations in the successive steps as it was applied in [5, 7, 8]. So that we can reproduce the best numerical results in those papers of the PC methods to be compared with our BPIRKN methods. The discussion on the error propagation in Section 2.3 suggests the following stopping criterion (cf. [5, 7, 8])

$$\|\mathbf{U}_{n,1}^{(m)} - \mathbf{U}_{n,1}^{(m-1)}\|_\infty \leq TOL = Ch^{p-1}, \tag{3.1}$$

where p is the order of the corrector method, and C is a parameter depending on the method and on the problem. To have a comparison with the numerical results for the PIRKN methods reported in [7, 8], we used the same constant C as in [7, 8]. Notice that by this criterion, the iteration error for $y_{n+1,1} = y_{n+1}$ and $y'_{n+1,1} = y'_{n+1}$ is of the same order in h as the underlying corrector. In the first step, we always use the trivial predictors $\mathbf{U}_{n,i}^{(0)} = y_n\mathbf{e} + a_i h y'_n \mathbf{c}$, $i = 1, \ldots, r$. The absolute error obtained at the end point of the integration interval is presented in the form 10^{-NCD} (NCD may be interpreted as the number of correct decimal digits). The computational efforts are measured by N_{seq} denoting the total number of sequential $\mathbf{f}$-evaluations required over the whole integration interval. Furthermore, in the tables of results, N_{steps} denotes the total number of integration steps. All the computations were carried out on a computer with 28 digits precision. An actual implementation on a parallel machine is a subject of further research.

3.1 Efficiency of BPIRKN-A and BPIRKN-B methods

Section 2.4 compared the stability properties of the two classes of BPIRKN-A and BPIRKN-B methods. This section is devoted to mutually numerical comparison of efficiency of these methods. For that purpose, we apply the BPIRKN-A and BPIRKN-B methods to a selection of two test problems found in the literature.

The first problem is the well-known *nonlinear Fehlberg problem* (cf. e.g., [11, 12, 14, 15])

$$\frac{d^2\mathbf{y}(t)}{dt^2} = \begin{pmatrix} -4t^2 & -\frac{2}{\sqrt{y_1^2(t)+y_2^2(t)}} \\ \frac{2}{\sqrt{y_1^2(t)+y_2^2(t)}} & -4t^2 \end{pmatrix} \mathbf{y}(t), \tag{3.2}$$

$$\mathbf{y}(0) = (0,1)^T, \qquad \mathbf{y}'(0) = (-2\sqrt{\pi/2}, 0)^T \qquad \sqrt{\pi/2} \le t \le 10,$$

with highly oscillating exact solution given by $\mathbf{y}(t) = (cos(t^2), sin(t^2))^T$.
The second problem is the *scalar problem* taken from [14]

$$\frac{d^2y(t)}{dt^2} = -25y(t) + 100cos(5t), \quad y(0) = 1, \quad y'(0) = 5, \quad 0 \le t \le 10. \tag{3.3}$$

The exact solution $y(t) = cos(5t) + sin(5t) + 10tsin(5t)$ possesses rapidly oscillating components which are appearing with small and large variable amplitudes.

Table 3.5: Values of NCD/N_{seq} for problem (3.2) obtained by pth-order BPIRKN-A and BPIRKN-B methods

Methods	p	$N_{steps} = 100$	$N_{steps} = 200$	$N_{steps} = 400$	$N_{steps} = 800$	C
BPIRKN-A	3	(*)	1.1 / 439	1.9 / 800	2.8 / 1600	10^2
BPIRKN-B	3	(*)	1.2 / 400	1.9 / 800	2.8 / 1600	10^2
BPIRKN-A	5	2.0 / 464	3.4 / 479	4.8 / 801	6.4 / 1601	10^3
BPIRKN-B	5	1.9 / 317	3.0 / 401	4.8 / 801	6.4 / 1601	10^3
BPIRKN-A	7	4.2 / 403	6.0 / 507	8.0 / 802	10.2 / 1602	10^3
BPIRKN-B	7	4.1 / 354	5.9 / 440	8.0 / 833	10.2 / 1602	10^3
BPIRKN-A	9	6.0 / 482	8.7 / 579	11.4 / 912	14.2 / 1604	10^3
BPIRKN-B	9	6.0 / 435	8.6 / 494	11.4 / 879	14.2 / 1604	10^3

Applying the iteration strategy using the stopping criterion (3.1), we determined the values of the NCD and N_{seq} as listed in Table 3.5 and Table 3.6, where, (*) indicates that the iteration process is not convergent. From these tables we observe that in the low accuracy range (with large stepsizes) the BPIRKN-B methods are shown to be more efficient than BPIRKN-A methods. This is (partly) due to the fact that the BPIRKN-A methods have smaller stability regions, they may need a larger number of corrections to become stable. Therefore, in the following, we shall confine our considerations to the BPIRKN-B methods.

Table 3.6: Values of NCD/N_{seq} for problem (3.3) obtained by pth-order BPIRKN-A and BPIRKN-B methods

Methods	p	$N_{steps}=25$	$N_{steps}=50$	$N_{steps}=100$	$N_{steps}=200$	C
BPIRKN-A	3	(*)	-0.5 / 204	0.3 / 219	1.2 / 400	10^2
BPIRKN-B	3	(*)	-0.5 / 117	0.2 / 201	1.1 / 400	10^2
BPIRKN-A	5	0.3 / 497	1.5 / 202	3.0 / 217	4.5 / 401	10^2
BPIRKN-B	5	0.2 / 358	1.6 / 150	3.1 / 201	4.6 / 401	10^2
BPIRKN-A	7	1.0 / 256	4.3 / 231	5.5 / 202	7.8 / 402	10^2
BPIRKN-B	7	2.0 / 229	3.7 / 196	5.2 / 202	7.7 / 402	10^2
BPIRKN-A	9	2.5 / 249	6.1 / 258	8.8 / 236	10.4 / 403	10^2
BPIRKN-B	9	3.0 / 206	5.3 / 227	8.3 / 204	10.8 / 403	10^2

3.2 Comparison with parallel methods

In this section, we report numerical results obtained by the best parallel explicit RKN methods available in the literature, that is the (indirect) PIRKN methods proposed in [25] (cf. also Section 1.2) and the BPIRKN-B methods specified above. We selected a test set of four problems taken from the literature.

3.2.1 Linear nonautonomous problem

As a first numerical test, we apply the various pth-order PC methods to the linear nonautonomous problem (cf. [5])

$$\begin{gathered}\frac{d^2\mathbf{y}(t)}{dt^2} = \begin{pmatrix} -2\alpha(t)+1 & -\alpha(t)+1 \\ 2(\alpha(t)-1) & \alpha(t)-2 \end{pmatrix} \mathbf{y}(t), \\ \alpha(t) = max\{2cos^2(t), sin^2(t)\}, \qquad 0 \le t \le 20, \\ \mathbf{y}(0) = (0,0)^T, \qquad \mathbf{y}'(0) = (-1,2)^T, \end{gathered} \tag{3.4}$$

with exact solution $\mathbf{y}(t) = (-sin(t), 2sin(t))^T$. The results listed in Table 3.7 clearly show that the BPIRKN-B methods are more efficient than the PIRKN methods of the same order. The high-order BPIRKN-B methods offer a gain of a factor more than 2. The BPIRKN methods used nearly 1 correction per step so that the number of $\mathbf{f}$-evaluations per step for all BPIRKN-B methods (almost) equals 2.

Table 3.7: Values of NCD/N_{seq} for problem (3.4) obtained by various pth-order parallel PC methods

Methods	p	$N_{steps}=100$	$N_{steps}=200$	$N_{steps}=400$	$N_{steps}=800$	$N_{steps}=1600$	C
PIRKN	3	2.4 / 293	3.3 / 587	4.2 / 1174	5.1 / 2347	6.0 / 4695	10^{-1}
BPIRKN-B	3	2.9 / 200	3.8 / 400	4.7 / 800	5.6 / 1600	6.5 / 3200	10^{-1}
PIRKN	5	5.8 / 392	7.3 / 784	8.8 / 1568	10.3 / 3137	11.8 / 6274	10^{-2}
BPIRKN-B	5	6.4 / 201	7.9 / 401	9.4 / 801	10.9 / 1601	13.0 / 3201	10^{-2}
PIRKN	7	9.5 / 476	11.6 / 952	13.7 / 1903	15.8 / 3809	17.9 / 7618	10^{-3}
BPIRKN-B	7	10.3 / 202	12.3 / 402	14.4 / 802	16.5 / 1602	18.6 / 3202	10^{-3}
PIRKN	9	13.9 / 500	16.2 / 1000	18.9 / 2000	21.5 / 4000	24.4 / 8000	10^{-4}
BPIRKN-B	9	14.2 / 203	16.8 / 403	19.5 / 803	22.2 / 1603	24.6 / 3203	10^{-4}

3.2.2 Nonlinear Fehlberg problem

For the second numerical test, we apply the PC methods to the nonlinear Fehlberg problem (3.2) previously used in Section 3.1. The results are reported in Table 3.8. These numerical

Table 3.8: Values of NCD/N_{seq} for problem (3.2) obtained by various pth-order parallel PC methods

Methods	p	$N_{steps}=200$	$N_{steps}=400$	$N_{steps}=800$	$N_{steps}=1600$	$N_{steps}=3200$	C
PIRKN	3	0.6 / 450	1.4 / 913	2.3 / 1829	3.2 / 3663	4.2 / 7322	10^{2}
BPIRKN-B	3	1.2 / 400	1.9 / 800	2.8 / 1600	3.7 / 3200	4.6 / 6400	10^{2}
PIRKN	5	2.8 / 683	4.3 / 1363	5.8 / 2730	7.3 / 5459	8.8 / 10919	10^{3}
BPIRKN-B	5	3.0 / 401	4.8 / 801	6.4 / 1601	7.9 / 3201	9.4 / 6401	10^{3}
PIRKN	7	5.3 / 917	7.4 / 1836	9.5 / 3672	11.6 / 7345	13.7 / 14690	10^{3}
BPIRKN-B	7	5.9 / 440	8.0 / 833	10.2 / 1602	12.4 / 3202	14.5 / 6402	10^{3}
PIRKN	9	8.0 / 1148	10.7 / 2255	13.4 / 4502	16.1 / 9003	18.9 / 18007	10^{3}
BPIRKN-B	9	8.6 / 494	11.4 / 879	14.2 / 1604	16.9 / 3204	19.6 / 6403	10^{3}

results show that the BPIRKN-B methods are by far superior to the PIRKN methods of the same order. The average number of **f**-evaluations per step is again, about two for all

BPIRKN-B methods.

3.2.3 Newton's equation of motion problem

The third numerical example is the two-body gravitational problem for Newton's equation of motion (see [24, p. 245]).

$$\frac{d^2y_1(t)}{dt^2} = -\frac{y_1(t)}{\left(\sqrt{y_1^2(t)+y_2^2(t)}\right)^3}, \quad \frac{d^2y_2(t)}{d^2t} = -\frac{y_2(t)}{\left(\sqrt{y_1^2(t)+y_2^2(t)}\right)^3}, \quad 0 \le t \le 20,$$

$$y_1(0) = 1-\varepsilon, \quad y_2(0) = 0, \quad y_1'(0) = 0, \quad y_2'(0) = \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}. \tag{3.5}$$

This problem can also found in [15] or from the test set of problems in [23]. The solution components are $y_1(t) = cos(u(t)) - \varepsilon$, $y_2(t) = \sqrt{(1+\varepsilon)(1-\varepsilon)}\ sin(u(t))$, where u(t) is the solution of Keppler's equation $t = u(t) - \varepsilon\ sin(u(t))$ and ε denotes the eccentricity of the orbit. In this example, we set $\varepsilon = 0.3$. The results for this problem are given in Table 3.9 and give rise to roughly the same conclusions as formulated in the two previous examples.

Table 3.9: Values of NCD/N_{seq} for problem (3.5) obtained by various pth-order parallel PC methods

Methods	p	$N_{steps}=100$	$N_{steps}=200$	$N_{steps}=400$	$N_{steps}=800$	$N_{steps}=1600$	C
PIRKN	3	0.5 / 200	1.3 / 400	2.2 / 800	3.1 / 1600	4.0 / 3200	10^1
BPIRKN-B	3	0.9 / 200	1.8 / 400	2.7 / 800	3.6 / 1600	4.5 / 3200	10^1
PIRKN	5	3.0 / 340	4.5 / 680	6.0 / 1361	7.5 / 2720	9.0 / 5439	10^{-1}
BPIRKN-B	5	3.7 / 202	5.1 / 402	6.6 / 802	8.1 / 1602	9.7 / 3202	10^{-1}
PIRKN	7	5.6 / 436	7.7 / 873	9.8 / 1745	12.0 / 3492	14.1 / 6983	10^{-2}
BPIRKN-B	7	6.3 / 204	8.4 / 403	10.5 / 803	12.6 / 1603	14.9 / 3203	10^{-2}
PIRKN	9	8.2 / 500	10.9 / 1000	13.6 / 2001	16.3 / 4006	19.1 / 8013	10^{-2}
BPIRKN-B	9	9.4 / 203	11.7 / 403	14.4 / 803	17.1 / 1604	19.9 / 3204	10^{-2}

3.2.4 Scalar problem

The last example is the scalar problem (3.3) which has been used in Section 3.1. The results reported in Table 3.10 again, show the superiority of the BPIRKN-B methods over the PIRKN methods. The number of **f**-evaluations per step equal to 2 is retained as in the three preceding examples.

Table 3.10: Values of NCD/N_{seq} for problem (3.3) obtained by various pth-order parallel PC methods

Methods	p	$N_{steps} = 100$	$N_{steps} = 200$	$N_{steps} = 400$	$N_{steps} = 800$	$N_{steps} = 1600$	C
PIRKN	3	0.0 / 280	0.8 / 559	1.6 / 1120	2.5 / 2238	3.4 / 4480	10^2
BPIRKN-B	3	0.2 / 201	1.1 / 400	2.0 / 800	2.9 / 1600	3.8 / 3200	10^2
PIRKN	5	2.5 / 390	3.9 / 780	5.4 / 1561	6.9 / 3119	8.4 / 6235	10^2
BPIRKN-B	5	3.1 / 201	4.6 / 401	6.0 / 801	7.5 / 1601	9.0 / 3201	10^2
PIRKN	7	5.3 / 489	7.4 / 975	9.4 / 1948	11.5 / 3905	13.6 / 7807	10^2
BPIRKN-B	7	5.2 / 202	7.7 / 402	10.0 / 802	12.1 / 1602	14.2 / 3202	10^2
PIRKN	9	8.3 / 572	11.0 / 1151	13.7 / 2306	16.4 / 4609	19.1 / 9210	10^2
BPIRKN-B	9	8.3 / 204	10.8 /403	14.3 /803	17.0 / 1603	19.7 / 3203	10^2

3.3 Comparison with sequential methods

In Section 3.2, the most efficient class of BPIRKN-B methods was compared with PIRKN methods (the most efficient parallel explicit RKN methods). In this section, we shall compare these BPIRKN-B methods with the sequential explicit RKN methods currently available.

Table 3.11: Comparison with sequential methods for problem (3.3)

Methods	N_{steps}	NCD	N_{seq}
11(12) pair (from [14])	243	15.7	4861
DOPRIN	49	2.6	393
	191	7.2	1529
	580	11.3	4641
	2060	15.2	16481
	7604	19.1	60833
$PIRKN_9$ (in this paper)	100	8.3	204
	200	10.8	403
	400	14.3	803
	800	17.0	1603
	1600	19.7	3203

We restricted our numerical experiments to the comparison of our 9-th order BPIRKN-B method ($BPIRKN_9$ method) with a few well-known sequential codes for nonlinear Fehlberg problem (3.2) and for scalar problem (3.3). We selected some embedded RKN pairs presented in the form $p(p+1)$ or $(p+1)p$ constructed in [11, 12, 14, 15] and the code DOPRIN taken from [20]. We reproduced the best results obtained by these sequential methods given in the literature (cf., e.g., [25]) and added the results obtained by $BPIRKN_9$ method. In spite of the fact that the results of the sequential methods are obtained using a stepsize strategy, whereas $BPIRKN_9$ method is applied with fixed stepsizes, it is the $BPIRKN_9$ method that performs most efficiently (see Table 3.11 and Table 3.12). When compared to the code DOPRIN from [20], the $BPIRKN_9$ offers a speed-up factor ranging from 7 to 20 (depending on the accuracy required).

Table 3.12: Comparison with sequential methods for problem (3.2)

Methods	N_{steps}	NCD	N_{seq}
11(12)-pair (from [14])	876	20.3	17521
11(10)-pair (from [15])	919	20.7	15614
9(10)-pair (from [12])	628	15.1	8793
	3235	21.4	45291
8(9)-pair (from [11])	1452	13.5	15973
DOPRIN (from [20])	79	3.8	633
	353	8.3	2825
	1208	12.3	9665
	4466	16.3	35729
	16667	20.3	133337
$PIRKN_9$ (in this paper)	200	8.6	494
	400	11.4	879
	800	14.2	1604
	1600	16.9	3204
	3200	19.6	6403

3.4 Remarks

From the numerical results obtained in this section, we remark that in spite of the dynamical strategy for stopping iteration process (3.1), the BPIRKN methods used only one correction per step. Some additional corrections as viewed in the tables of numerical results are due to the low order predictions in the first step. This remark suggests us to implement the BPIRKN methods in $P(CE)^m E$ mode, where $m = [(p-1)/2]$ for the first step and $m = 1$ otherwise. Applying this mode of implementation to the above test problems led to the same performance of BPIRKN-B methods. In the forthcoming paper, we shall apply this mode of implementation for BPIRKN methods with stepsize control.

4 Concluding remarks

This paper described an algorithm to obtain Runge-Kutta-Nyström-type parallel block PC methods (BPIRKN methods) requiring (almost) 2 sequential $\mathbf{f}$-evaluations per step for any order of accuracy. The structure of BPIRKN methods also enables us to obtain various cheap error estimates for stepsize control. The sequential costs of a resulting class of these BPIRKN methods (BPIRKN-B methods) implemented with fixed stepsize strategy are already considerably less than those of the best parallel and sequential methods available in the literature. These conclusions encourage us to pursue the study of BPIRKN methods. In particular, we will concentrate on performance analysis of other predictor methods and on stepsize control that exploit the special structure of BPIRKN methods.

References

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

[2] J.C. Butcher, *Implicit Runge-Kutta processes*, Math. Comp. **18**, (1964), 50-64.

[3] J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations, Runge-Kutta and General Linear Methods*, (Wiley, New York, 1987).

[4] N.H. Cong, *An improvement for parallel-iterated Runge-Kutta-Nyström methods*, Acta Math. Viet. **18** (1993), 295-308.

[5] N.H. Cong, *Note on the performance of direct and indirect Runge-Kutta-Nyström methods*, J. Comput. Appl. Math. **45** (1993), 347-355.

[6] N.H. Cong, *Direct collocation-based two-step Runge-Kutta-Nyström methods*, SEA Bull. Math. **19** (1995), 49-58.

[7] N.H. Cong, *Explicit symmetric Runge-Kutta-Nyström methods for parallel computers*, Comput. Math. Appl. **31** (1996), 111-122.

[8] N.H. Cong, *Explicit parallel two-step Runge-Kutta-Nyström methods*, Comput. Math. Appl. **32** (1996), 119-130.

[9] N.H. Cong, *Parallel Runge-Kutta-Nyström-type PC methods with stepsize control*, in preparation.

[10] W.H. Enright and D.J. Highman, *Parallel defect control*, BIT **31** (1991), 647-663.

[11] E. Fehlberg, *Klassische Runge-Kutta-Nyström-Formeln mit Schrittweitenkontrolle für Differentialgleichungen $x'' = f(t,x)$*, Computing **10** (1972), 305-315.

[12] E. Fehlberg, *Eine Runge-Kutta-Nyström-Formel 9-ter Ordnung mit Schrittweitenkontrolle für Differentialgleichungen $x'' = f(t,x)$*, Z. Angew. Math. Mech. **61** (1981), 477-485.

[13] E. Fehlberg, S. Filippi und J. Gräf, *Ein Runge-Kutta-Nyström-Formelpaar der Ordnung 10(11) für Differentialgleichungen* $y'' = f(t,y)$, Z. Angew. Math. Mech. **66** (1986), 265-270.

[14] S. Filippi und J. Gräf, *Ein Runge-Kutta-Nyström-Formelpaar der Ordnung 11(12) für Differentialgleichungen der Form* $y'' = f(t,y)$, Computing **34** (1985), 271-282.

[15] S. Filippi and J. Gräf, *New Runge-Kutta-Nyström formula-pairs of order 8(7), 9(8), 10(9) and 11(10) for differential equations of the form* $y'' = f(t,y)$, J. Comput. Appl. Math. **14** (1986), 361-370.

[16] E. Hairer, *Methodes de Nyström pour l'équations differentielle* $y''(t) = f(t,y)$, Numer. Math. **27** (1977), 283-300.

[17] E. Hairer, *A Runge-Kutta method of order 10*, J. Inst. Math. Appl. **21** (1978), 47-59.

[18] E. Hairer, *Unconditionally stable methods for second order differential equations*, Numer. Math. **32** (1979), 373-379.

[19] E. Hairer, *A one-step method of order 10 for* $y''(t) = f(t,y)$, IMA J. Numer. Anal. **2** (1982), 83-94.

[20] E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations, I. Nonstiff Problems*, (second revised edition, Springer-Verlag, Berlin, 1993).

[21] P.J. van der Houwen and N.H. Cong, *Parallel block predictor-corrector methods of Runge-Kutta type*, Appl. Numer. Math. **13** (1993), 109-123.

[22] P.J. van der Houwen, B.P. Sommeijer and N.H. Cong, *Stability of collocation-based Runge-Kutta-Nyström methods*, BIT **31** (1991), 469-481.

[23] T.E. Hull, W.H. Enright, B.M. Fellen and A.E. Sedgwick, *Comparing numerical methods for ordinary differential equations*, SIAM J. Numer. Anal. **9** (1972), 603-637.

[24] L.F. Shampine and M.K. Gordon, *Computer Solution of Ordinary Differential Equations, The Initial Value Problems*, (W.H. Freeman and Company, San Francisco, 1975).

[25] B.P. Sommeijer, *Explicit, high-order Runge-Kutta-Nyström methods for parallel computers*, Appl. Numer. Math. **13** (1993), 221-240.

[26] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, (Springer-Verlag, New York, 1983).

Nguyen Huu Cong
Faculty of Mathematics
Mechanics and Informatics
Hanoi University of Sciences
90 Nguyen Trai
Dong Da, Hanoi
Vietnam

Karl Strehmel, Rüdiger Weiner
FB Mathematik und Informatik
Martin-Luther-Universität
Halle-Wittenberg
Postfach
D-06099 Halle/Saale
Germany
e-mail: rw@mail.mathematik.uni-halle.de