Pergamon

Computers Math. Applic. Vol. 35, No. 9, pp. 45–57, 1998
Copyright©1998 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0898-1221/98 $19.00 + 0.00
PII: S0898-1221(98)00056-X

RKN-Type Parallel Block PC Methods with Lagrange-Type Predictors

NGUYEN HUU CONG
Faculty of Mathematics, Mechanics and Informatics, Hanoi University of Sciences
90 Nguyen Trai, Dong Da, Hanoi, Vietnam

(Received August 1997; accepted October 1997)

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 at step points, but also at off-step points (block points), so that in 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 interpolation. By suitable choice of the abscissas of the off-step points, a much more accurately predicted value is obtained than by predictor formulas based on last step values. Since the block of 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. Application of the resulting block predictor-corrector methods to a few widely-used test problems reveals that the sequential costs are reduced by a factor ranging from 4 to 50 when compared with the best sequential methods from the literature.

Keywords—Runge-Kutta-Nyström methods, Predictor-corrector methods, Stability, Parallelism.

1. INTRODUCTION

Consider numerical solution of nonstiff Initial Value Problems (IVPs) for the systems of special second order, Ordinary Differential Equations (ODEs)

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

where $\mathbf{y}, \mathbf{f} \in \mathbb{R}^d$. Problems of the form (1.1) are encountered in, e.g., celestial mechanics. The most efficient numerical methods for solving this problem is the explicit Runge-Kutta-Nyström methods (RKN methods) directly designed for problem like (1.1). In the literature, sequential explicit RKN methods up to order 10 can be found in [1–4]. In order to exploit the facilities of parallel computers, several class of parallel Predictor-Corrector methods (PC methods) based on RKN-type correctors have been investigated in [5–9]. A common challenge in these papers is to reduce, for a given order of accuracy, the required number of sequential f-evaluations per step, using parallel processors. In the present paper, we investigate a particular class of RKN-type parallel block PC methods considered in [10]. Following the approach used in that paper, we apply

This work was partly supported by DAAD, N.R.P.F.S., and QG-96-02.
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 K. Strehmel, R. Weiner, and all the staff members of the institute for the hospitality and for their interest in my research.

Typeset by $\mathcal{A}_\mathcal{M}\mathcal{S}$-TEX

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 first used in [11] for obtaining reliable defect control in explicit RK methods. It was also successfully applied in [12] for increasing efficiency of RK-type parallel PC methods. In [10], the block of approximations is used to obtain a highly accurate predictor formula by using Hermite-type interpolation. In this paper, we also use this block of approximations to obtain a high-order predictor formula by using Lagrange-type interpolation. The RKN-type parallel PC methods based on this approach require few numbers of sequential f-evaluations per step with acceptable stability properties. The precise location of the off-step points can be used for minimizing the Lagrange interpolation errors and also for obtaining various cheap error estimates 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 parallel 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. [6,9]). Consequently, the RKN-type parallel PC methods considered in this paper use a large number of processors. They can be considered as block versions of the Parallel-Iterated RKN methods (PIRKN methods) considered in [6,9] using block Lagrange-type predictors, and will therefore be termed *Block PIRKN methods with Lagrange-type predictor* (BPIRKN-L methods). Moreover, if the PC iteration process continues until convergence, then using direct RKN correctors, leads to BPIRKN-L methods possessing both faster convergence and smaller truncation error resulting in better efficiency than by using indirect RKN correctors (cf., e.g., [6]).

In the next section, we shall formulate and investigate the block PIRKN-L methods. Furthermore, in Section 3, we present numerical comparisons of BPIRKN-L methods 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.

2. BPIRKN-L METHODS

Our starting point is an s-stage collocation-based implicit RKN methods (see, e.g., [13; 14, p. 272]) given by

$$\begin{aligned} \mathbf{U}_n &= u_n\mathbf{e} + hu'_n\mathbf{c} + h^2Af(\mathbf{U}_n), \\ u_{n+1} &= u_n + hu'_n + h^2\mathbf{b}^\top f(\mathbf{U}_n), \\ u'_{n+1} &= u'_n + h\mathbf{d}^\top f(\mathbf{U}_n), \end{aligned} \tag{2.1}$$

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 j^{th} 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 n^{th} step. Furthermore, in (2.1), we use for any vector $\mathbf{v} = (v_1, \dots, v_s)^\top$ and any scalar function f, the notation $f(\mathbf{v}) := (f(v_1), \dots, f(v_s))^\top$. This RKN method will be referred to as the corrector method. We distinguish two types of collocation implicit RKN methods: *direct* and *indirect* (cf. [13]). Indirect collocation implicit RKN methods are derived from collocation implicit RK methods for first-order ODEs. Direct collocation implicit RKN methods are directly constructed for second-order ODEs of the form in (1.1). A first family of these direct implicit RKN methods is obtained by means of collocation technique considered in [13]. In this paper, we will confine our considerations to high-order collocation implicit RKN methods that is the Gauss-Legendre and Radau IIA methods. This class contains methods of arbitrarily high order.

Indirect collocation implicit RKN methods can be found in [15]. Direct collocation implicit RKN methods were investigated in [6,13].

Applying the RKN method (2.1) at step t_n with r distinct stepsizes $a_i h$, where $i = 1, \dots, r$ and $a_1 = 1$, we obtain in this way, a block of correctors

$$\begin{aligned}
\mathbf{U}_{n,i} &= u_n \mathbf{e} + a_i h u'_n \mathbf{c} + a_i^2 h^2 A f(\mathbf{U}_{n,i}), \\
u_{n+1,i} &= u_n + a_i h u'_n + a_i^2 h^2 \mathbf{b}^{\mathrm{T}} f(\mathbf{U}_{n,i}), \\
u'_{n+1,i} &= u'_n + a_i h \mathbf{d}^{\mathrm{T}} f(\mathbf{U}_{n,i}), \qquad i = 1, \dots, r.
\end{aligned} \tag{2.2}$$

Let us suppose that at $(n-1)^{\text{th}}$ step, a block of predictions $\mathbf{U}^{(0)}_{n-1,i}$, $i = 1, \dots, r$, 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 $y(t_{n-1} + a_i h)$, $i = 1, \dots, r$, defined by

$$\begin{aligned}
\mathbf{U}^{(j)}_{n-1,i} &= y_{n-1} \mathbf{e} + a_i h y'_{n-1} \mathbf{c} + a_i^2 h^2 A f\left(\mathbf{U}^{(j-1)}_{n-1,i}\right), \qquad j = 1, \dots, m_b, \\
y_{n,i} &= y_{n-1} + a_i h y'_{n-1} + a_i^2 h^2 \mathbf{b}^{\mathrm{T}} f\left(\mathbf{U}^{(m_b)}_{n-1,i}\right), \\
y'_{n,i} &= y'_{n-1} + a_i h \mathbf{d}^{\mathrm{T}} f\left(\mathbf{U}^{(m_b)}_{n-1,i}\right), \qquad i = 1, \dots, r.
\end{aligned}$$

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

$$\begin{aligned}
\mathbf{Y}_n &:= (y_{n,1}, \dots, y_{n,r})^{\mathrm{T}}, \qquad y_{n,1} = y_n, \\
\mathbf{Y}'_n &:= \left(y'_{n,1}, \dots, y'_{n,r}\right)^{\mathrm{T}}, \qquad y'_{n,1} = y'_n,
\end{aligned} \tag{2.3}$$

we can construct the following Lagrange-type predictor formulas:

$$\mathbf{U}^{(0)}_{n,i} = V_i \mathbf{Y}_n, \qquad i = 1, \dots, r, \tag{2.4}$$

where predictor matrices V_i, $i = 1, \dots, r$, will be determined by order conditions (see Section 2.1). Apart from (2.4), we can construct predictors of other types like, e.g., Hermite and Adam types (cf. [10,12]). Regarding (2.2) as block corrector methods and (2.4) as block predictor methods for the stage vectors, we leave the class of one-step methods and arrive at a block PC method in P(CE)^{m}E mode

$$\mathbf{U}^{(0)}_{n,i} = V_i \mathbf{Y}_n, \tag{2.5a}$$

$$\begin{aligned}
\mathbf{U}^{(j)}_{n,i} &= \mathbf{e}\mathbf{e}_1^{\mathrm{T}} \mathbf{Y}_n + a_i h \mathbf{c}\mathbf{e}_1^{\mathrm{T}} \mathbf{Y}'_n + a_i^2 h^2 A f\left(\mathbf{U}^{(j-1)}_{n,i}\right), \qquad j = 1, \dots, m, \\
y_{n+1,i} &= \mathbf{e}_1^{\mathrm{T}} \mathbf{Y}_n + a_i h \mathbf{e}_1^{\mathrm{T}} \mathbf{Y}'_n + a_i^2 h^2 \mathbf{b}^{\mathrm{T}} f\left(\mathbf{U}^{(m)}_{n,i}\right), \\
y'_{n+1,i} &= \mathbf{e}_1^{\mathrm{T}} \mathbf{Y}'_n + a_i h \mathbf{d}^{\mathrm{T}} f\left(\mathbf{U}^{(m)}_{n,i}\right), \qquad i = 1, \dots, r.
\end{aligned} \tag{2.5b}$$

It can be seen that the block PC method (2.5) consists of a block of PIRKN-type corrections using a block of Lagrange-type predictions at the off-step points (block points) (cf. [6,9]). Therefore, we shall call method (2.5) the *r-dimensional block PIRKN method with Lagrange-type predictors (BPIRKN-L method).* Given the vectors $\mathbf{Y}_n$ and $\mathbf{Y}'_n$, the r values $y_{n,i}$ can be computed in parallel and, on a second level, the components of the i^{th} stage vector iterate $\mathbf{U}^{(j)}_{n,i}$ can also be evaluated in parallel (cf. [6,9]). Hence, the r-dimensional BPIRKN-L methods (2.5) based on s-stage RKN correctors can be implemented on a computer possessing $r \cdot s$ parallel processors. The number of sequential f-evaluations per step of length h in each processor equals $s^* = m + 1$.

2.1. Order Conditions for the Predictor

In this section, we consider order conditions for the predictors. For the fixed stepsize h, the q^{th}-order conditions for (2.5a) are derived by replacing $\mathbf{U}_{n,i}^{(0)}$ and $\mathbf{Y}_n$ by the exact solution values $y(t_n\mathbf{e} + a_i h\mathbf{c})$ and $y(t_{n-1}\mathbf{e} + h\mathbf{a}) = y(t_n\mathbf{e} + h(\mathbf{a} - \mathbf{e}))$, respectively, with $\mathbf{a} = (a_1, \dots, a_r)^{\mathrm{T}}$. On substitution of these exact values into (2.5a) and by requiring that the residue is of order $q + 1$ in h, we are led to

$$y\left(t_n\mathbf{e} + a_i h\mathbf{c}\right) - V_i y\left(t_n\mathbf{e} + h\left(\mathbf{a} - \mathbf{e}\right)\right) = O\left(h^{q+1}\right), \qquad i = 1, \dots, r. \tag{2.6}$$

Using Taylor expansions, we can expand the left-hand side of (2.6) in powers of h and obtain

$$\begin{gathered}\left[\exp\left(ha_i\mathbf{c}\frac{d}{dt}\right) - V_i \exp\left(h(\mathbf{a} - \mathbf{e})\frac{d}{dt}\right)\right] y(t_n) \\ = \sum_{j=0}^{q} \mathbf{C}_i^{(j)} \left(h\frac{d}{dt}\right)^j y(t_n) + \mathbf{C}_i^{(q+1)} \left(h\frac{d}{dt}\right)^{q+1} y\left(t_i^*\right) = O\left(h^{q+1}\right), \\ i = 1, \dots, r,\end{gathered} \tag{2.7}$$

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

$$\begin{gathered}\mathbf{C}_i^{(j)} = \frac{1}{j!}\left[(a_i\mathbf{c})^j - V_i\left(\mathbf{a} - \mathbf{e}\right)^j\right], \\ j = 0, \dots, q, \qquad i = 1, \dots, r.\end{gathered} \tag{2.8a}$$

The vectors $\mathbf{C}_i^{(j)}$, $i = 1, \dots, r$ represent the error vectors of the block predictors (2.5a). From (2.7), we obtain the order conditions

$$\mathbf{C}_i^{(j)} = \mathbf{0}, \qquad j = 0, 1, \dots, q, \qquad i = 1, \dots, r. \tag{2.8b}$$

The vectors $\mathbf{C}_i^{(q+1)}$, $i = 1, \dots, r$, are the *principal* error vectors of the block predictors. The conditions (2.8), imply that

$$\mathbf{U}_{n,i} - \mathbf{U}_{n,i}^{(0)} = O\left(h^{q+1}\right), \qquad i = 1, \dots, r. \tag{2.9}$$

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

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

Furthermore, for the local truncation error of the BPIRKN-L method (2.5), we may write

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

where p is the order of the generating RKN corrector (2.1). Thus, as in [10], we have the similar theorem.

THEOREM 2.1. *If the conditions (2.8) are satisfied and if the generating RKN corrector (2.1) has step point order p, then the BPIRKN-L method (2.5) has the iteration error order $p_{\text{iter}} = 2m+q+1$ and step point order $p^* = \min\{p, p_{\text{iter}}\}$.*

In order to express V_i, $i = 1, \dots, r$ explicitly in terms of vectors $\mathbf{a}$ and $\mathbf{c}$, we suppose that $q = r - 1$ and define the matrices

$$P_i := \left(\mathbf{e}, a_i\mathbf{c}, a_i^2\mathbf{c}^2, a_i^3\mathbf{c}^3, a_i^4\mathbf{c}^4, \dots, a_i^{r-1}\mathbf{c}^{r-1}\right),$$
$$Q := \left(\mathbf{e}, (\mathbf{a}-\mathbf{e}), (\mathbf{a}-\mathbf{e})^2, (\mathbf{a}-\mathbf{e})^3, \dots, (\mathbf{a}-\mathbf{e})^{r-1}\right),$$
$$i = 1, \dots, r.$$

Then the order conditions (2.8) can be presented in the form

$$P_i - V_i Q = O, \qquad i = 1, \dots, r. \tag{2.10}$$

Since the components a_i of vector $\mathbf{a}$ are assumed to be distinct implying that Q is nonsingular, and from (2.10), we may write

$$V_i = P_i Q^{-1}, \qquad i = 1, \dots, r. \tag{2.11}$$

In view of Theorem 2.1 and the simply explicit expressions of the predictor matrices V_i in (2.11), we have the following theorem.

THEOREM 2.2. *If $q = r-1$, and the predictor matrices V_i, $i = 1, \dots, r$ satisfy the relations (2.11), then for the BPIRKN-L methods (2.5), $p_{\text{iter}} = r + 2m$, $p^* = \min\{p, p_{\text{iter}}\}$, and $s^* = m + 1$.*

In the application of BPIRKN-L methods, we have two classes of high-order corrector methods: Gauss-Legendre correctors and Radau IIA correctors. The use of high-order corrector methods is motivated by reducing the number of processors needed for implementation of BPIRKN-L 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 BPIRKN-L methods using Gauss-Legendre correctors. In the near future, we intend to investigate a class of more suitable correctors methods in order to reduce round-off effect (cf. Section 2.3).

2.2. Convergence Boundaries

In actual implementation of BPIRKN-L methods, the number of iterations m is determined by some iteration strategy, rather than by order conditions using minimal number of iterations to attain 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., [6,10], 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 $\frac{\partial \mathbf{f}}{\partial \mathbf{y}}$. For this equation, we obtain the iteration error equation

$$\mathbf{U}_{n,i}^{(j)} - \mathbf{U}_{n,i} = a_i^2 zA\left[\mathbf{U}_{n,i}^{(j-1)} - \mathbf{U}_{n,i}\right], \qquad z := h^2\lambda, \quad j = 1, \dots, m. \tag{2.12}$$

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 $a_i^2 zA$, $i = 1, \dots, r$. Requiring that $\rho(a_i^2 zA) < 1$, leads us to the convergence condition

$$a_i^2|z| < \frac{1}{\rho(A)} \quad \text{or} \quad a_i^2 h^2 < \frac{1}{\rho\left(\dfrac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)\rho(A)}. \tag{2.13}$$

We shall call $1/\rho(A)$ the *convergence boundary*. In actual computation, the integration stepsize h should be substantially smaller than allowed by condition (2.13). 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

$$a_i^2|z| \le \gamma(\alpha) \quad \text{or} \quad a_i^2 h^2 \le \frac{\gamma(\alpha)}{\rho\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)}, \qquad \gamma(\alpha) = \frac{\alpha}{\rho(A)}, \tag{2.14}$$

where $\gamma(\alpha)$ presents the boundary of convergence with damping factor α of the method. Specification of the convergence boundaries $\gamma(\alpha)$ of the BPIRKN-L methods based on direct and indirect Gauss-Legendre and Radau IIA correctors can be found in, e.g., [6,10]. Notice that for a given stepsize h, the maximal damping factor is defined by

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

The convergence boundaries of the BPIRKN-L methods given in, e.g., [6,10] show that the direct Gauss-Legendre and Radau IIA correctors give rise to faster convergence than indirect ones of the same order. Therefore, if the PC iteration is continued until convergence, then the BPIRKN-L methods based on direct collocation RKN correctors are prefered (cf. [6,10]). These corrector methods are not A-stable (see [13]), but their stability regions are sufficiently large for nonstiff problems (cf. [6]). The similar parallel PC methods based on PC pairs Hermite-Radau IIA have been investigated in [10]. In this paper, we shall consider BPIRKN-L methods based on Gauss-Legendre correctors (PC pairs Gauss-Legendre-Lagrange).

2.3. The Choice of Block Abscissas a_i

The accuracy of Lagrange interpolation formulas is improved if the interpolation abscissas are more narrowly spaced. However, this will increase the magnitude of the entries of the predictor matrices V_i, causing serious round-off errors. There are several ways to reduce this round-off effect as were discussed in [12]. Also in [11], where Hermite interpolation formulas were used for increasing reliability in explicit RK methods, it was found that on a 15 digits precision computer, the Hermite 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 ε of the block vectors $\mathbf{Y}_n$ and $\mathbf{Y}'_n$ within a single step (the similar analysis was given in [10,12]). We shall study this for the model test equation $y''(t) = \lambda y(t)$. For that purpose, we first express $y_{n+1,i}$ and $hy'_{n+1,i}$ in terms of $\mathbf{Y}_n$ and $h\mathbf{Y}'_n$. Since

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

applying (2.5),(2.12) to the model test equation for a given number m, we obtain

$$\begin{aligned}
y_{n+1,i} &= \mathbf{e}_1^{\mathsf{T}}\mathbf{Y}_n + a_i h \mathbf{e}_1^{\mathsf{T}}\mathbf{Y}'_n + a_i^2 z \mathbf{b}^{\mathsf{T}}\left[\mathbf{U}_{n,i}^{(m)} - \mathbf{U}_{n,i}\right] + a_i^2 z\mathbf{b}^{\mathsf{T}}\mathbf{U}_{n,i} \\
&= \mathbf{e}_1^{\mathsf{T}}\mathbf{Y}_n + a_i h \mathbf{e}_1^{\mathsf{T}}\mathbf{Y}'_n + a_i^2 z \mathbf{b}^{\mathsf{T}}\left[I - a_i^2 zA\right]^{-1}\left[\mathbf{e}\mathbf{e}_1^{\mathsf{T}}\mathbf{Y}_n + \mathbf{c}\mathbf{e}_1^{\mathsf{T}} a_i h\mathbf{Y}'_n\right] \\
&\quad + a_i^2 z\mathbf{b}^{\mathsf{T}}[a_i^2 zA]^m\left[\left[V_i - \left[I - a_i^2 zA\right]^{-1}\mathbf{e}\mathbf{e}_1^{\mathsf{T}}\right]\mathbf{Y}_n - \left[I - a_i^2 zA\right]^{-1}\mathbf{c}\mathbf{e}_1^{\mathsf{T}} a_i h\mathbf{Y}'_n\right] \qquad (2.15\text{a}) \\
&= \left[\mathbf{e}_1^{\mathsf{T}} + a_i^2 z\mathbf{b}^{\mathsf{T}}\left[I - a_i^2 zA\right]^{-1}\mathbf{e}\mathbf{e}_1^{\mathsf{T}} + a_i^2 z\mathbf{b}^{\mathsf{T}}\left[a_i^2 zA\right]^m\left[V_i - \left[I - a_i^2 zA\right]^{-1}\mathbf{e}\mathbf{e}_1^{\mathsf{T}}\right]\right]\mathbf{Y}_n \\
&\quad + \left[a_i\mathbf{e}_1^{\mathsf{T}} + a_i^2 z\mathbf{b}^{\mathsf{T}}\left[I - a_i^2 zA\right]^{-1} a_i\mathbf{c}\mathbf{e}_1^{\mathsf{T}} - a_i^2 z\mathbf{b}^{\mathsf{T}}\left[a_i^2 zA\right]^m\left[I - a_i^2 zA\right]^{-1} a_i\mathbf{c}\mathbf{e}_1^{\mathsf{T}}\right] h\mathbf{Y}'_n,
\end{aligned}$$

$$\begin{aligned}
hy'_{n+1,i} &= \mathbf{e}_1^\top h\mathbf{Y}'_n + +a_i z\mathbf{d}^\top \left[\mathbf{U}_{n,i}^{(m)} - \mathbf{U}_{n,i}\right] + a_i z\mathbf{d}^\top \mathbf{U}_{n,i} \\
&= h\mathbf{e}_1^\top \mathbf{Y}'_n + a_i z\mathbf{d}^\top \left[I - a_i^2 zA\right]^{-1} \left[\mathbf{e}\mathbf{e}_1^\top \mathbf{Y}_n + \mathbf{c}\mathbf{e}_1^\top a_i h\mathbf{Y}'_n\right] \\
&\quad + a_i z\mathbf{d}^\top \left[a_i^2 zA\right]^m \left[\left[V_i - \left[I - a_i^2 zA\right]^{-1} \mathbf{e}\mathbf{e}_1^\top\right] \mathbf{Y}_n - \left[I - a_i^2 zA\right]^{-1} \mathbf{c}\mathbf{e}_1^\top a_i h\mathbf{Y}'_n\right] \qquad (2.15\text{b}) \\
&= \left[a_i z\mathbf{d}^\top \left[I - a_i^2 zA\right]^{-1} \mathbf{e}\mathbf{e}_1^\top + a_i z\mathbf{d}^\top \left[a_i^2 zA\right]^m \left[V_i - \left[I - a_i^2 zA\right]^{-1} \mathbf{e}\mathbf{e}_1^\top\right]\right] \mathbf{Y}_n \\
&\quad + \left[\mathbf{e}_1^\top + a_i z\mathbf{d}^\top \left[I - a_i^2 zA\right]^{-1} a_i \mathbf{c}\mathbf{e}_1^\top - a_i z\mathbf{d}^\top \left[a_i^2 zA\right]^m \left[I - a_i^2 zA\right]^{-1} a_i \mathbf{c}\mathbf{e}_1^\top\right] h\mathbf{Y}'_n.
\end{aligned}$$

Let us now replace $\mathbf{Y}_n$ by $\mathbf{Y}_n^* = \mathbf{Y}_n + \varepsilon$ and $\mathbf{Y}'_n$ by $\mathbf{Y'}_n^* = \mathbf{Y}'_n + \varepsilon$. Then from (2.15), 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

$$\begin{aligned}
y_{n+1,i}^* = y_{n+1,i} &+ \left[\mathbf{e}_1^\top + a_i^2 z\mathbf{b}^\top \left[I - a_i^2 zA\right]^{-1} \mathbf{e}\mathbf{e}_1^\top\right. \\
&+ \left. a_i^2 z\mathbf{b}^\top \left[a_i^2 zA\right]^m \left[V_i - \left[I - a_i^2 zA\right]^{-1} \mathbf{e}\mathbf{e}_1^\top\right]\right] \varepsilon \\
&+ \left[a_i \mathbf{e}_1^\top + a_i^2 z\mathbf{b}^\top \left[I - a_i^2 zA\right]^{-1} a_i \mathbf{c}\mathbf{e}_1^\top\right. \\
&- \left. a_i^2 z\mathbf{b}^\top \left[a_i^2 zA\right]^m \left[I - a_i^2 zA\right]^{-1} a_i \mathbf{c}\mathbf{e}_1^\top\right] h\varepsilon,
\end{aligned} \qquad (2.16\text{a})$$

$$\begin{aligned}
y'^*_{n+1,i} = y'_{n+1,i} &+ \frac{1}{h} \left[a_i z\mathbf{d}^\top \left[I - a_i^2 zA\right]^{-1} \mathbf{e}\mathbf{e}_1^\top\right. \\
&+ \left. a_i z\mathbf{d}^\top \left[a_i^2 zA\right]^m \left[V_i - \left[I - a_i^2 zA\right]^{-1} \mathbf{e}\mathbf{e}_1^\top\right]\right] \varepsilon \\
&+ \left[\mathbf{e}_1^\top + a_i z\mathbf{d}^\top \left[I - a_i^2 zA\right]^{-1} a_i \mathbf{c}\mathbf{e}_1^\top\right. \\
&- \left. a_i z\mathbf{d}^\top \left[a_i^2 zA\right]^m \left[I - a_i^2 zA\right]^{-1} a_i \mathbf{c}\mathbf{e}_1^\top\right] \varepsilon.
\end{aligned} \qquad (2.16\text{b})$$

These relations show that the first component of the perturbation ε 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 [12], 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 (2.8) is minimized.

In our case of Lagrange-type predictors, where $q = r - 1$ (cf. Theorem 2.2), we have to minimize the magnitude of $\|\mathbf{C}_1^{(r)}\|_\infty$. Although we may use (2.8a) for minimizing $\|\mathbf{C}_1^{(r)}\|_\infty$, it is more convenient to start with usual expression of Lagrange interpolation formulas. For r-times continuously differentiable function $y(t)$, the r-point Lagrange interpolation formula can be written as (see, e.g., [16, p. 878; 17])

$$\begin{aligned}
y(t_n + \tau h) &= \sum_{i=1}^{r} l_i(\tau) y(t_n + a_i h) + C^{(r)}(\tau) \left(h\frac{d}{dt}\right)^r y\left(t_\tau^*\right), \\
l_i(\tau) &= \prod_{j=1, j\neq i}^{r} \frac{(\tau - a_j)}{(a_i - a_j)}, \qquad C^{(r)}(\tau) = \frac{1}{r!} \prod_{j=1}^{r} (\tau - a_j),
\end{aligned} \qquad (2.17)$$

where t_τ^* is a suitably chosen point in the interval $[t_n, t_n + \tau h]$. Hence, we have the following alternative form of (2.7):

$$\begin{gathered}
y(t_n + a_i c_k h) = \sum_{j=1}^{r} l_j(1 + a_i c_k) y\left(t_n + (a_j - 1)h\right) + C^{(r)}(1 + a_i c_k) \left(h\frac{d}{dt}\right)^r y\left(t_{ik}^*\right), \\
k = 1, \dots, s, \qquad i = 1, \dots, r,
\end{gathered} \qquad (2.18)$$

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 Lagrange-type predictor formulas defined by (2.8a) are given by $\mathbf{C}_i^{(r)} =$

$C^{(r)}(\mathbf{e}+a_i\mathbf{c})$, $i=1,\dots,r$. Recalling that for $i=1$, we are led to minimize the magnitude of the values

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

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

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

By this choice, the principal error vector $\mathbf{C}_1^{(r)}$ vanishes (i.e., $\|\mathbf{C}_1^{(r)}\|_\infty=\|C^{(r)}(\mathbf{e}+\mathbf{c})\|_\infty=0$), 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. (2.16)). If $r>s+1$, then we have additional abscissas for improving the predictor formulas. It is tempting to use these additional abscissas for reducing the magnitude of the norm of other principal error vectors. Therefore, it seems recommendable to choose the remaining abscissas outside the interval $[1,1+c_s]$ with *averaged* spacing of the abscissas $a_1,\dots,a_{s+1}$ equal to $1/(s+1)$ (cf. [12]). This leads us to define the remaining abscissas according to:

$$\begin{aligned}&\text{if } c_s\neq 1, &&\text{then } a_i=\frac{s+i}{s+1}, && i=s+2,\dots,r,\\ & &&\text{else } a_i=\frac{s+i-1}{s}, && i=s+2,\dots,r.\end{aligned} \tag{2.20b}$$

For Gauss-Legendre correctors of order $p=2s$, the choice (2.20) results in an averaged spacing of $2/(p+2)$. Recalling that on a 15-digit precision machine, the minimal spacing of 0.2 is acceptable in the case of Hermite interpolation (cf. [11]), we expect that for order up to 10, this averaged spacing $2/(p+2)$ should be acceptable for more stable Lagrange interpolation formulas. Finally, we remark that the abscissas defined by (2.20) enable us to develop various cheap strategies for stepsize control. For example, if $r\geq s+2$, then the difference $y_{n-1,s+2}-y_{n,1}$ can be used for obtaining an error estimate.

2.4. Stability Boundaries

The linear stability of the BPIRKN-L methods (2.5) is investigated by again using the model test equation $y''(t)=\lambda y(t)$, where λ is assumed to be negative. From (2.15), 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

$$M_m^0(z)=\begin{pmatrix}\mathbf{e}_1^\top+a_1^2z\mathbf{b}^\top\left[I-a_1^2zA\right]^{-1}\mathbf{e}\mathbf{e}_1^\top+a_1^2z\mathbf{b}^\top\left[a_1^2zA\right]^m\left[V_1-\left[I-a_1^2zA\right]^{-1}\mathbf{e}\mathbf{e}_1^\top\right] & \mathbf{0}^\top\\ \vdots & \vdots\\ \mathbf{e}_1^\top+a_r^2z\mathbf{b}^\top\left[I-a_r^2zA\right]^{-1}\mathbf{e}\mathbf{e}_1^\top+a_r^2z\mathbf{b}^\top\left[a_r^2zA\right]^m\left[V_r-\left[I-a_r^2zA\right]^{-1}\mathbf{e}\mathbf{e}_1^\top\right] & \mathbf{0}^\top\\ a_1z\mathbf{d}^\top\left[I-a_1^2zA\right]^{-1}\mathbf{e}\mathbf{e}_1^\top+a_1z\mathbf{d}^\top\left[a_1^2zA\right]^m\left[V_1-\left[I-a_1^2zA\right]^{-1}\mathbf{e}\mathbf{e}_1^\top\right] & \mathbf{0}^\top\\ \vdots & \vdots\\ a_rz\mathbf{d}^\top\left[I-a_r^2zA\right]^{-1}\mathbf{e}\mathbf{e}_1^\top+a_rz\mathbf{d}^\top\left[a_r^2zA\right]^m\left[V_r-\left[I-a_r^2zA\right]^{-1}\mathbf{e}\mathbf{e}_1^\top\right] & \mathbf{0}^\top\end{pmatrix}, \tag{2.21b}$$

$$M_m^1(z)=\begin{pmatrix}\mathbf{0}^\top & a_1\mathbf{e}_1^\top+a_1^2z\mathbf{b}^\top\left[I-a_1^2zA\right]^{-1}a_1\mathbf{c}\mathbf{e}_1^\top-a_1^2z\mathbf{b}^\top\left[a_1^2zA\right]^m\left[I-a_1^2zA\right]^{-1}a_1\mathbf{c}\mathbf{e}_1^\top\\ \vdots & \vdots\\ \mathbf{0}^\top & a_r\mathbf{e}_1^\top+a_r^2z\mathbf{b}^\top\left[I-a_r^2zA\right]^{-1}a_r\mathbf{c}\mathbf{e}_1^\top-a_r^2z\mathbf{b}^\top\left[a_r^2zA\right]^m\left[I-a_r^2zA\right]^{-1}a_r\mathbf{c}\mathbf{e}_1^\top\\ \mathbf{0}^\top & \mathbf{e}_1^\top+a_1z\mathbf{d}^\top\left[I-a_1^2zA\right]^{-1}a_1\mathbf{c}\mathbf{e}_1^\top-a_1z\mathbf{d}^\top\left[a_1^2zA\right]^m\left[I-a_1^2zA\right]^{-1}a_1\mathbf{c}\mathbf{e}_1^\top\\ \vdots & \vdots\\ \mathbf{0}^\top & \mathbf{e}_1^\top+a_rz\mathbf{d}^\top\left[I-a_r^2zA\right]^{-1}a_r\mathbf{c}\mathbf{e}_1^\top-a_rz\mathbf{d}^\top\left[a_r^2zA\right]^m\left[I-a_r^2zA\right]^{-1}a_r\mathbf{c}\mathbf{e}_1^\top\end{pmatrix}. \tag{2.21c}$$

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

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

It is evident from (2.21) that if z satisfies the convergence conditions (2.13), then the stability function of the BPIRKN-L method $\rho(M_m(z))$ converges to the stability function of the RKN corrector method as $m \to \infty$ (cf., e.g., [13; 14, 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 $(-\beta_{\text{corr}}, 0)$ of the RKN corrector and its convergence region defined by (2.13).

Confining the considerations to the BPIRKN-L methods with blocksize $r = p = 2s$, we numerically calculated the values of $\beta(m)$ for various resulting BPIRKN-L methods as listed in Table 1. The boundaries $\beta(m)$ for BPIRKN-L methods based on direct and indirect Gauss-Legendre correctors are denoted by $\beta(m)_{\text{direct}}$ and $\beta(m)_{\text{indirect}}$, respectively. From this table we observe that the stability boundaries show a rather irregular behaviour. For $m = 0$ or $m = 1$, the BPIRKN-L methods based on direct and indirect Gauss-Legender correctors have a comparable stability boundaries. From Table 1, we can select a whole set of BPIRKN-L methods of order p up to 10 requiring 1 or 2 **f**-evaluations per step with acceptable stability intervals for nonstiff problems (cf. Theorem 2.2).

Table 1. Stability boundaries $\beta(m)$ for various p^{th}-order BPIRKN-L methods based on direct and indirect correctors.

BPIRKN Methods	$p = 4$	$p = 6$	$p = 8$	$p = 10$
$\beta(0)_{\text{direct}}$	0.714	0.022	0.507	0.018
$\beta(1)_{\text{direct}}$	0.015	1.055	0.232	1.324
$\beta(2)_{\text{direct}}$	0.104	0.150	1.311	5.073
$\beta(3)_{\text{direct}}$	0.691	1.730	0.871	1.502
$\beta(4)_{\text{direct}}$	0.437	0.808	3.537	4.249
$\beta(5)_{\text{direct}}$	0.667	1.410	2.213	3.489
$\beta(6)_{\text{direct}}$	1.042	4.354	4.614	4.736
$\beta(0)_{\text{indirect}}$	0.714	0.022	0.5012	0.019
$\beta(1)_{\text{indirect}}$	2.229	1.055	0.232	1.324
$\beta(2)_{\text{indirect}}$	0.021	1.636	3.688	5.073
$\beta(3)_{\text{indirect}}$	0.117	0.304	0.461	3.052
$\beta(4)_{\text{indirect}}$	2.281	0.683	2.319	1.723
$\beta(5)_{\text{indirect}}$	0.328	2.689	1.535	2.910
$\beta(6)_{\text{indirect}}$	0.552	1.513	3.377	3.396

3. NUMERICAL EXPERIMENTS

In this section, we report numerical results obtained by the BPIRKN-L methods (2.5). As it was mentioned in the previous sections, we confine our considerations to the r-dimensional BPIRKN methods based on Gauss-Legendre correctors of orders up to 10 with $r = p$ and block points defined by (2.20). We shall compare the BPIRKN-L methods with parallel and sequential explicit RKN methods from the literature. In the first step, we always use the trivial predictor formulas given by

$$\mathbf{U}^{(0)}_{n,i} = y_n\mathbf{e} + a_i h y'_n \mathbf{c}, \qquad i = 1, \dots, r.$$

The absolute error obtained at the end point of the integration interval is presented in the form $10^{-\text{NCD}}$ (NCD may be interpreted as the number of correct decimal digits). The computational efforts are measured by the values of N_{seq} denoting the total number of sequential **f**-evaluations required over the whole integration interval. For an easy comparison of the various methods, the (fixed) stepsize is chosen such that the number of sequential f-evaluations of each method (approximately) equals a prescribed number N_{seq}. To be more precise, let s^* denote the number of sequential f-evaluations per step, N_{steps} denote the total number of integration steps for the integration interval $[t_0, T]$, then $N_{\text{seq}} = N_{\text{steps}} \cdot s^*$ which leads us to

$$N_{\text{steps}} = \left[\frac{N_{\text{seq}}}{s^*} + \frac{1}{2}\right], \qquad h = \frac{T - t_0}{N_{\text{steps}}},$$

where $[\cdot]$ denotes the integer part function (the effect of the integer part operation causes that the actual number of sequential f-evaluations may be slightly different from the prescribed number N_{seq}). For p^{th}-order PIRKN methods, $s^* = [p/2]$ in all steps (cf. [9]). For any p^{th}-order BPIRKN-L methods, we used $s^* = [p/2]$ in first step and $s^* = 1$ after first step so that the stepsize h will be evaluated accordingly. Since the PC iterations will not be continued until convergence, we confine our considerations on the BPIRKN-L methods based on indirect Gauss-Legendre correctors. All the computations were carried out on a 28-digit precision computer. An actual implementation on a parallel machine is a subject of further study.

3.1. 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 [9] and the BPIRKN-L methods specified above. We selected a test set of three problems taken from the literature.

3.1.1. Linear nonautonomous problem

As a first numerical test, we apply the various p^{th}-order PC methods to the linear nonautonomous problem (cf., e.g., [6–8])

$$\begin{aligned} \frac{d^2\mathbf{y}(t)}{dt^2} &= \begin{pmatrix} -2\alpha(t)+1 & -\alpha(t)+1 \\ 2(\alpha(t)-1) & \alpha(t)-2 \end{pmatrix} \mathbf{y}(t), \\ \alpha(t) &= \max\left\{2\cos^2(t), \sin^2(t)\right\}, \qquad 0 \le t \le 20, \\ \mathbf{y}(0) &= (0,0)^{\mathrm{T}}, \qquad \mathbf{y}'(0) = (-1,2)^{\mathrm{T}}, \end{aligned} \tag{3.1}$$

with exact solution $\mathbf{y}(t) = (-\sin(t), 2\sin(t))^{\mathrm{T}}$. The numerical results listed in Table 2 clearly show that the BPIRKN-L methods are more efficient than the PIRKN methods of the same order. The high-order BPIRKN-L methods offer a gain of a factor more than 4.

Table 2. NCD-values for several values of N_{seq} for problem (3.1) obtained by various p^{th}-order parallel PC methods.

Methods	p	$N_{\text{seq}} = 100$	$N_{\text{seq}} = 200$	$N_{\text{seq}} = 400$	$N_{\text{seq}} = 800$	$N_{\text{seq}} = 1600$
PIRKN	4	2.4	3.7	4.9	6.1	7.3
BPIRKN-L	4	4.5	6.1	8.4	8.9	9.9
PIRKN	6	3.9	5.8	7.7	9.7	11.5
BPIRKN-L	6	7.7	9.8	12.0	14.4	16.9
PIRKN	8	6.4	8.5	11.3	14.1	16.1
BPIRKN-L	8	11.4	14.1	16.8	19.6	21.7
PIRKN	10	7.2	9.9	13.2	16.4	19.6
BPIRKN-L	10	15.0	18.4	20.2		

3.1.2. Nonlinear Fehlberg problem

For the second numerical test, we apply the various p^{th}-order PC methods to the well-known *nonlinear Fehlberg problem* (cf., e.g., [1,2,18,19])

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

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

with highly oscillating exact solution given by $\mathbf{y}(t) = (\cos(t^2), \sin(t^2))^{\mathsf{T}}$. The results are reported in Table 3. These numerical results show that the BPIRKN-L methods are again superior to the PIRKN methods of the same order by a speed-up factor more than 4.

Table 3. NCD-values for several values of N_{seq} for problem (3.2) obtained by various p^{th}-order parallel PC methods.

Methods	p	$N_{\text{seq}} = 300$	$N_{\text{seq}} = 600$	$N_{\text{seq}} = 1200$	$N_{\text{seq}} = 2400$	$N_{\text{seq}} = 4800$
PIRKN	4	0.3	1.6	2.8	4.0	5.2
BPIRKN-L	4	2.1	3.6	5.0	6.4	7.7
PIRKN	6	1.1	3.1	5.0	6.9	8.8
BPIRKN-L	6	4.8	6.8	8.9	11.0	13.0
PIRKN	8	1.5	4.1	6.8	9.4	12.0
BPIRKN-L	8	7.8	10.4	13.0	15.7	18.4
PIRKN	10	1.8	5.2	8.5	11.9	15.1
BPIRKN-L	10	10.4	14.0	17.3	19.4	

3.1.3. Newton's equation of motion problem

The third numerical example is the two-body gravitational problem for Newton's equation of motion (see [20, p. 245]).

$$\frac{d^2y_1(t)}{dt^2} = -\frac{y_1(t)}{\left(\sqrt{y_1^2(t)+y_2^2(t)}\right)^3}, \qquad \frac{d^2y_2(t)}{d^2t} = -\frac{y_2(t)}{\left(\sqrt{y_1^2(t)+y_2^2(t)}\right)^3}, \qquad 0 \le t \le 20, \tag{3.3}$$

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

This problem can also found in [19] or from the test set of problems in [21]. 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 4 and give rise to roughly the same conclusions as formulated in the two previous examples.

Table 4. NCD-values for several values of N_{seq} for problem (3.3) obtained by various p^{th}-order parallel PC methods.

Methods	p	$N_{\text{seq}} = 100$	$N_{\text{seq}} = 200$	$N_{\text{seq}} = 400$	$N_{\text{seq}} = 800$	$N_{\text{seq}} = 1600$
PIRKN	4	0.3	2.0	3.0	4.2	5.4
BPIRKN-L	4	1.6	3.1	4.5	6.0	7.5
PIRKN	6	2.1	3.3	4.9	6.7	8.6
BPIRKN-L	6	3.8	5.8	7.9	10.0	12.1
PIRKN	8	2.7	4.9	7.0	9.4	11.8
BPIRKN-L	8	6.6	9.8	11.9	14.4	17.0
PIRKN	10	3.3	5.5	8.4	11.4	14.5
BPIRKN-L	10	7.7	11.2	14.7	18.1	19.9

3.2. Comparison with Sequential Methods

In Section 3.1, a class of BPIRKN-L methods was compared with PIRKN methods (the most efficient parallel explicit RKN methods). In this section, we shall compare these BPIRKN-L methods with the sequential explicit RKN methods currently available.

We restricted the numerical experiments to the comparison of our 10th order BPIRKN-L method (BPIRKN$_{10}$ method) with a few well-known sequential codes for nonlinear Fehlberg problem (3.2). We selected some embedded RKN pairs presented in the form $p(p+1)$ or $(p+1)p$ constructed in [1,2,18,19] and the code DOPRIN taken from [22]. We reproduced the best results obtained by these sequential methods given in the literature (cf., e.g., [9,19]) and added the results obtained by BPIRKN$_{10}$ method. In spite of the fact that the results of the sequential methods are obtained using a stepsize strategy, whereas BPIRKN$_{10}$ method is applied with fixed stepsizes, it is the BPIRKN$_{10}$ method that performs most efficiently (see Table 5). When compared to the code DOPRIN from [22], the BPIRKN$_{10}$ offers a speed-up factor ranging from 7 to 50 (depending on the accuracy required).

Table 5. Comparison with sequential methods for problem (3.2).

Methods	N_{steps}	NCD	N_{seq}
11(12)-pair (from [15])	876	20.3	17521
11(10)-pair (from [16])	919	20.7	15614
9(10)-pair (from [13])	628	15.1	8793
	3235	21.4	45291
8(9)-pair (from [12])	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$_{10}$ (in this paper)	200	8.6	494
	296	10.4	300
	596	14.0	600
	1196	17.3	1200
	2396	20.4	2400

4. CONCLUDING REMARKS

This paper described an algorithm to obtain Runge-Kutta-Nyström-type parallel block PC methods with Lagrange-type predictors (BPIRKN-L methods) requiring one or two sequential f-evaluations per step for any order of accuracy. The structure of BPIRKN-L methods also enables us to obtain various cheap error estimates for stepsize control. The sequential costs of a resulting class of BPIRKN-L 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-L methods. In particular, we will concentrate on performance analysis of predictor methods of other types like Hermite and Adams types, and on stepsize control that exploits the special structure of BPIRKN-L methods.

REFERENCES

1. E. Fehlberg, Klassische Runge-Kutta-Nyström formeln mit Schrittweiten-Kontrolle für differentialgleichungen $x'' = f(t,x)$, *Computing* **10**, 305–315, (1972).
2. E. Fehlberg, Eine Runge-Kutta-Nyström formel 9-ter Ordnung mit Schrittweitenkontrolle für differentialgleichungen $x'' = f(t,x)$, *Z. Angew. Math. Mech.* **61**, 477–485, (1981).
3. E. Hairer, Methodes de Nyström pour l'équations differentielle $y''(t) = f(t,y)$, *Numer. Math.* **27**, 283–300, (1977).

4. E. Hairer, A one-step method of order 10 for $y''(t) = f(t,y)$, *IMA J. Numer. Anal.* **2**, 83–94, (1982).
5. N.H. Cong, An improvement for parallel-iterated Runge-Kutta-Nyström methods, *Acta Math. Viet.* **18**, 295–308, (1993).
6. N.H. Cong, Note on the performance of direct and indirect Runge-Kutta-Nyström methods, *J. Comput. Appl. Math.* **45**, 347–355, (1993).
7. N.H. Cong, Explicit symmetric Runge-Kutta-Nyström methods for parallel computers, *Computers Math. Applic.* **31** (2), 111–122, (1996).
8. N.H. Cong, Explicit parallel two-step Runge-Kutta-Nyström methods, *Computers Math. Applic.* **32** (3), 119–130, (1996).
9. B.P. Sommeijer, Explicit, high-order Runge-Kutta-Nyström methods for parallel computers, *Appl. Numer. Math.* **13**, 221–240, (1993).
10. N.H. Cong, K. Strehmel and R. Weiner, Runge-Kutta-Nyström-type parallel block predictor-corrector methods, Report No. 29, Computer Science and Scientific Computing, Halle University, Germany, (1997).
11. W.H. Enright and D.J. Highman, Parallel defect control, *BIT* **31**, 647–663, (1991).
12. P.J. van der Houwen and N.H. Cong, Parallel block predictor-corrector methods of Runge-Kutta type, *Appl. Numer. Math.* **13**, 109–123, (1993).
13. P.J. van der Houwen, B.P. Sommeijer and N.H. Cong, Stability of collocation-based Runge-Kutta-Nyström methods, *BIT* **31**, 469–481, (1991).
14. K. Burrage, *Parallel and Sequential Methods for Ordinary Differential Equations*, Clarendon Press, Oxford, (1995).
15. E. Hairer, Unconditionally stable methods for second order differential equations, *Numer. Math.* **32**, 373–379, (1979).
16. M. Abramowitz and I.A. Stegun, Handbook of mathematical functions, In *Applied Mathematics Series 55*, National Bureau of Standards, Dover, New York, (1970).
17. J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, (1983).
18. S. Filippi und J. Gräf, Ein Runge-Kutta-Nyström formelpaar der Ordnung 11(12) für differentialgleichungen der form $y'' = f(t,y)$, *Computing* **34**, 271–282, (1985).
19. 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**, 361–370, (1986).
20. L.F. Shampine and M.K. Gordon, Computer solution of ordinary differential equations, In *The Initial Value Problems*, W.H. Freeman, San Francisco, CA, (1975).
21. 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**, 603–637, (1972).
22. E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations, I. Nonstiff Problems*, Second revised edition, Springer-Verlag, Berlin, (1993).
23. J.C. Butcher, The numerical snalysis of ordinary differential equations, In *Runge-Kutta and General Linear Methods*, Wiley, New York, (1987).
24. N.H. Cong, Direct collocation-based two-step Runge-Kutta-Nyström methods, *SEA Bull. Math.* **19**, 49–58, (1995).
25. N.H. Cong, Parallel Runge-Kutta-Nyström-type PC methods with stepsize control, (in preparation).
26. E. Fehlberg, S. Filippi and J. Gräf, Eine Runge-Kutta-Nyström formelpaar der Ordnung 10(11) für differentialgleichungen $y'' = f(t,y)$, *Z. Angew. Math. Mech.* **66**, 265–270, (1986).