ACTA MATHEMATICA VIETNAMICA
Volume 17, Number 2, 1992

PARALLEL DIRECT COLLOCATION-BASED IMPLICIT RUNGE-KUTTA-NYSTRÖM METHODS WITH HIGH STABILITY

NGUYEN HUU CONG[1]

Abstract. In this paper, we study parallel iteration of predictor-corrector methods (parallel PC methods) with direct collocation-based implicit Runge-Kutta-Nyström (IRKN) correctors in solving stiff initial-value problems for special second-order ordinary differential equations (ODEs) on parallel computers. The parallel methods were constructed in order to achieve high stability. We will show that they are rather efficient for solving stiff initial-value problems.

1. Introduction

The subject of the paper is the numerical integration of the stiff initial-value problem for systems of special second-order, ordinary differential equations (ODEs) of dimension d:

$$\boldsymbol{y}''(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)), \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \boldsymbol{y}'(t_o) = \boldsymbol{y}'_0, \boldsymbol{y} : \mathbb{R} \to \mathbb{R}^d, \boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^d, t_0 \le t \le t_{end}. \tag{1.1}$$

One of the most efficient numerical methods for integrating problem (1.1) is implicit Runge-Kutta-Nyström method of the form:

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + h\boldsymbol{y}'_n + b_0 h^2 \boldsymbol{f}(t_n, \boldsymbol{y}_n) + h^2 \sum_{i=1}^{k} b_i \boldsymbol{f}(t_n + c_i h, Y_i), \tag{1.2a}$$

$$\boldsymbol{y}'_{n+1} = \boldsymbol{y}'_n + d_0 h \boldsymbol{f}(t_n, \boldsymbol{y}_n) + h \sum_{i=1}^{k} d_i \boldsymbol{f}(t_n, +c_i h, Y_i), \tag{1.2b}$$

$$Y_i = \boldsymbol{y}_n + c_i h \boldsymbol{y}'_n + a_i h^2 \boldsymbol{f}(t_n, \boldsymbol{y}_n) + h^2 \sum_{j=1}^{k} a_{ij} \boldsymbol{f}(t_n + c_j h, Y_j), \quad i = 1, \ldots, k, \tag{1.2c}$$

Received January 25, 1992.
[1]Partially supported by the National Basic Research Program.

or using the Butcher array notation (cf.[4]),

$$\begin{array}{c|cc} 0 & 0 & \mathbf{0}^T \\ \boldsymbol{c} & \boldsymbol{a} & A \\ \hline & b_0 & \boldsymbol{b}^T \\ & d_0 & \boldsymbol{d}^T \end{array} \tag{1.3}$$

where $\boldsymbol{b} = (b_i), \boldsymbol{c} = (c_i)$ and $\boldsymbol{d} = (d_i)$ are k-dimensional vectors, and $A = (a_{ij})$ is a k-by-k matrix. We always assume that the matrix A is nonsingular. If the vector $\boldsymbol{a}$ does not vanish, then (1.2) presents a $(s = k + 1)$-stage RKN method requiring k *implicit* stages and one explicit stage. If $\boldsymbol{a} = \mathbf{0}$, then (1.2) reduces to the general $(s = k)$-stage RKN method with s implicit stages. The collocation methods (1.2) have excellent A-stability but they are very expensive in solving stage-vector equations (1.2c). These methods belong to two collocation types: *direct* collocation and *indirect* collocation. For a full discussion of the IRKN methods, their order of accuracy p and stage order r, we refer to [3], [8].

In this paper, we shall concentrate on the parallel iteration of the single A-stable direct collocation-based IRKN correctors which are denoted by $Dr(3/4, 1), Dr(-1/5, 9/10, 1), Dr(-1/5, 9/10, 19/20, 1)$ and $Dr(-1/4, 0, 9/10, 19/20, 1)$. Here $(3/4, 1)^T, (-1/5, 9/10, 1)^T, (-1/5, 9/10, 19/20, 1)^T$ and $(-1/4, 0, 9/10, 19/20, 1)^T$ are the vectors of collocation points (more details about this direct collocations based methods can be found in [8]). The use of non A-stable RKN correctors will be investigated in another paper in order to exploit the performance in high step point order.

We consider the parallel integration methods based on parallel iteration of fully implicit Runge-Kutta-Nyström (RKN) methods of direct collocation type in preditor-corrector mode. Such parallel RKN methods were considered in [11]. The parallel methods constructed in [9] and [11] require much more number of iterations per step to become A-stable than convergence criterion requires. In this paper we construct parallel methods which are A-stable for the smallest number of iterations per step. The investigation in this paper is considered as an addition to the results obtained in [9] and [11]. The numerical experiments

and comparisons clearly show the efficiency of the parallel methods proposed in this paper.

2. Parallel RKN methods

We shall construct itegration methods by diagonal-implicit PC iteration of fully implicit RKN methods. Thus assuming that in (1.2) the matrix $A = (a_{ij})$ is a full matrix, we have to find the solution of the stage-vector equation and the stage vector $Y = (Y_i)$. Our aim is to construct solution methods that run fast on parallel computers. In the case where all eigenvalues of the Jacobian matrix are close to the origin, the stage vector equation (1.2c) can be solved by fixed point iteration which is well-suited for implementation on parallel computers. For first-order ODEs this has been discussed in [13], [10], and [5]. If there are also largely negative eigenvalues, then fixed point iteration would dictate rather small stepsizes in order to get convergence. We will consider a more powerful class of parallel iteration processes which lead to the same degree of implicitness as in SDIRKN methods. These processes are similar to the *stiff iteration* method applied in [7] and parallel PC iteration applied in [9] and [11].

Thus, let $Y_i^{(\mu)}$ denote the μ-th iterate to Y_i, and define

$$X_i := Y_i - x_i, X_i^{(\mu)} = Y_i^{(\mu)} - x_i, \text{ where } x_i := y_n + c_i h y_n' + a_i h^2 \boldsymbol{f}(t_n, y_n). \quad (2.1)$$

We shall compute iterates $X_i^{(\mu)}$, rather than $Y_i^{(\mu)}$, because the quantities $X_i^{(\mu)}$ are of smaller magnitude and are therefore less sensitive to rounding errors. In terms of X_i and x_i, The stage-vector equation (1.2c) reads

$$X_i = h^2 \sum_{j=1}^{k} a_{ij} \boldsymbol{f}(t_n + c_j h, X_j + x_j), \quad i = 1, \ldots, k. \quad (1.2cc)$$

We define the parallel RKN methods as follows

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + h\boldsymbol{y}_n' + b_0 h^2 \boldsymbol{f}(t_n, \boldsymbol{y}_n) + h^2 \sum_{i=1}^{k} b_i \boldsymbol{f}(t_n + c_i h, X_i^{(\mu)} + x_i), \quad (2.2a)$$

$$\boldsymbol{y}_{n+1}' = \boldsymbol{y}_n' + d_0 h \boldsymbol{f}(t_n, y_n) + h \sum_{i=1}^{k} d_i \boldsymbol{f}(t_n + c_i h, X_i^{(\mu)} + x_i) \quad (2.2b)$$

$$X_i^{(\mu)} - h^2\delta_i \boldsymbol{f}(t_n + c_i h, X_i^{(\mu)} + x_i) = X_i^{(\mu-1)} - \delta_i h^2 \boldsymbol{f}(t_n + c_i h, X_i^{(\mu-1)} + x_i)$$
$$-\omega_\mu [X_i^{(\mu-1)} - h^2 \sum_{j=1}^{k} \boldsymbol{f}(t_n + c_j h, X_j^{(\mu-1)} + x_i)], \qquad (2.2c)$$

where $i = 1, \ldots, k$; $\mu = 1, 2, \ldots, m$; ω_μ are the relaxation parameters, and δ_i are the iteration parameters which are assumed to be positive.

Since the k systems which are to be solved in each iteration step of (2.2c) can be solved in parallel and each has the dimension equal to that of the system (1.1), the iteration process (2.2) is on a k-processor computers of the same coputational complexity as a m-stage SDIRKN method on an one-processor computer.

For the sake of simplicitty of notation and presentation, we assume that (1.1) is a scalar initial-value problem. However, all considerations can be trivially extended to the systems of equations. For scalar ODE $y" = f(t, y)$, we suppose that the solution of (1.1) uniquely exists and equal to $y(t)$. In some neighbourhood of this exact solution, $y(t)$ can be well represented by a solution of the so called variational equation:

$$y" = f(t, y(t)) + \partial f/\partial y(t, y(t))[y - y(t)] \qquad (2.3)$$

Assuming that $\partial f/\partial y$ can be locally "frozen", then (2.3) takes the form $y" = \lambda y + g(t, y(t))$. Since $g(t, y(t))$ does not depend on y, we arrive at the conclusion that it is enough to perform our investigations only for the scalar test equation

$$y" = \lambda y. \qquad (**)$$

2.1 Predictor

For starting the iteration (2.2c), we need a predictor to compute the initial approximations $X_i^{(0)}$. For more details about the predictors we refer to [9]. As in [11] we follow two more attrative predictors:

Explicit predictor I with $X_i^{(0)} = -a_i h^2 \boldsymbol{f}(t_n, y_n)$

Implicit predictor II with $X_i^{(0)} = -a_i h^2 \boldsymbol{f}(t_n, y_n) + h^2 \delta_i \boldsymbol{f}(t_n + c_i h, X_i^{(0)} + x_i)$.

In view of stability, an important property of the predictors is the degree of amplification of stiff components(here, stiff components are understood to be

eigenvector components corresponding to large negative eigenvalues of $\partial f/\partial y$). From Theorem 2.1 of [11] we know that both of the predictor I and predictor II are of order one, i.e.

$$\epsilon^{(0)} = X_i - X_i^{(0)} = O(h^2).$$

2.2 Convergence

It is evident that if $X_i^{(\mu)}$ defined by (2.2c) converges as μ tends to ∞, then $X_i^{(\mu)}$ converges to the i-th component of the solution of the stage-vector solution of (1.2cc), i.e., to X_i. Concerning the convergence of the iteration process (2.2c), we have the following theorem

THEOREM 2.1. *If* $\| I - \omega_\mu(I - zD)^{-1}(I - zA) \| \leq C_0 < 1$ *for any* $z \leq 0$, *where* D *is the diagonal matrix with diagonal entries* δ_i *then* $X_i^{(\mu)}$ *defined by (2.2c) converges to* X_i *as* μ *tends* ∞.

PROOF: Applying our parallel method to the scalar test equation (**) we have

$$[X - X^{(\mu)}] = [I - \omega_\mu(I - zD)^{-1}(I - zA)][X - X^{(\mu-1)}]$$

$$\text{with } X = (X_i), X^{(\mu)} = (X_i^{(\mu)}), z = \lambda h^2. \tag{2.5}$$

It follows from (2.5) that

$$\| X - X^{\mu} \| \leq \| I - \omega_\mu(I - zD)^{-1}(I - zA) \| \| X - X^{\mu-1} \| . \tag{2.6}$$

By the assumption of the theorem we get the recusion

$$\| X - X^{(\mu)} \| \leq C_0 \| X - X^{(\mu-1)} \| \text{ with } C_0 < 1. \tag{2.7}$$

The recusion (2.7) leads to Theorem 2.1. Notice that condition $\| I - \omega_\mu(I - zD)^{-1}(I - zA) \| < 1$ is considered as a minimal requirement for a method like (2.2). In the folowing section the norm $\| I - \omega_\mu(I - zD)^{-1}(I - zA) \|$ is replaced by its spectral radius.

2.3 Stability

In [9] and [11] we have studied the stability of parallel RKN methods of the form (2.2) by applying the method to the scalar test equation (**) and we have derived the stability function for these methods.

If we denote $\boldsymbol{v}_{n+1} = \begin{pmatrix} y_{n+1} \\ hy'_{n+1} \end{pmatrix}$ then we get $\boldsymbol{v}_{n+1} = [M(z) - E_m(z)]\boldsymbol{v}_n$, where

$$M(z) := \begin{pmatrix} 1 + zb_0 + z\boldsymbol{b}^T(I - zA)^{-1}(\boldsymbol{c} + z\boldsymbol{a}) & 1 + z\boldsymbol{b}^T(I - zA)^{-1}\boldsymbol{c} \\ zd_0 + z\boldsymbol{d}^T(I - zA)^{-1}(\boldsymbol{c} + z\boldsymbol{a}) & 1 + z\boldsymbol{d}^T(I - zA)^{-1}\boldsymbol{c} \end{pmatrix} \tag{2.8}$$

is the stability matrix of IRKN methods, e is k-dimensional vector with unit entries.

$$E_m(z) := \begin{pmatrix} \boldsymbol{p}^T_{P_m(H(z))K_1(z)} & \boldsymbol{p}^T_{P_m(H(z))K_2(z)} \\ \boldsymbol{d}^T A^{-1} P_m(H(z))K_1(z) & \boldsymbol{d}^T A^{-1} P_m(H(z))K_2(z) \end{pmatrix} \tag{2.9}$$

is the iteration error matrix with $\boldsymbol{p}^T = b^T A^{-1}$ for nonstiffly accurate corrector, $\boldsymbol{p}^T = \boldsymbol{e}_k^T$ for stiffly accurate correctors, $P_m(x) := \prod_{\mu-1}^{m}(1 - \omega_\mu x)$ and $H(z) := [I - zD]^{-1}[I - zA]$. Futhermore, in (2.9) $K_1(z) = [I - zA]^{-1} zA[\boldsymbol{e} + z\boldsymbol{a}]$, $K_2(z) = [I - zA]^{-1} zA\boldsymbol{c}$ for the predictor I $K_1(z) = [I - zA]^{-1} zA[\boldsymbol{e} + z\boldsymbol{a}] - [I - zD]^{-1}[zD(e + za) - za]$, $K_2(z) = [I - zA]^{-1} zA - (I - zD)^{-1} zD)]\boldsymbol{c}$ for the predictor II, and

$$R_m(z) := \rho([M(z) - E_m(z)]) \tag{2.10}$$

is the stability function of the parllel methods.

DEFINITION 2.1. The parallel RKN methods are said to be A-stable for m_0 if $R_{m_0}(z) \leq 1$ for any $z \leq 0$. Charactezing the stability of parallel RKN methods we introduce the critical value of m defined as minimal value of m such that the parallel methods are A-stable for all m equal to or greater than this value. Let us denote this value by m_{crit}.

2.4. Determination of the number of iterations

Theorem 2.1 gives us the condition for the convergence of the parallel methods (2.2). In these parallel methods we suppose that m is an interger such that $X^{(m)}$ approximates X with sufficient accuracy. In pratical use of the parallel methods (2.2) we have to study some criterion for the detemination of m, i.e.,

some criterion to stop the iteration (2.2c)per step and per processor (for the stability of the parallel methods the number m has to be equal or greater than m_{crit}.)

For this purpose we shall from now denote the step values of the corrrector solution by $\boldsymbol{u}_{n+1}$ and $\boldsymbol{u}'_{n+1}$ (the solution obtained by the parallel method (2.2) by $\boldsymbol{y}_{n+1}$ and $\boldsymbol{y}'_{n+1}$.

DEFINITION 2.2: We shall say that the order of the iteration error for the step values of the slution obtained by the parallel methods (2.2) equals to q^* if $\boldsymbol{u}_{n+1} - \boldsymbol{y}_{n+1} = O(h^{q^*+1}), \boldsymbol{u}'_{n+1} - \boldsymbol{y}'_{n+1} = O(h^{q^*+1})$.

THEOREM 2.2. *For the parallel methods (2.2), if the order of the iteration error for the stage-vector solution equals to q^*, then the iteration error for the step values of the solution is at least of order q^*, i.e.,*

$$X - X^{[m]} = O(h^{q^*+1}) \Rightarrow \boldsymbol{u}_{n+1} - \boldsymbol{y}_{n+1} = O(h^{q^*+1}), \boldsymbol{u}'_{n+1} - \boldsymbol{y}'_{n+1} = O(h^{q^*+1}) \tag{2.11}$$

PROOF. For the scalar equation, if we use the notation in which for any vector $\boldsymbol{v} = (v_i)$, $f(\boldsymbol{v})$ denotes the vector with entries $f(v_i)$, then the corrector (1.2) can be written in the following form

$$u_{n+1} = y_n + hy'_n + b_0 h^2 f(t_n, y_n) + h^2 b^T f(\boldsymbol{e}t_n + \boldsymbol{c}h, Y), \tag{1.2a'}$$

$$u'_{n+1} = y'_n + d_0 h f(t_n, y_n) + h d^T f(\boldsymbol{e}t_n + \boldsymbol{c}h, Y), \tag{1.2b'}$$

$$Y = y_n + \boldsymbol{c}hy'_n + h^2 \boldsymbol{a} f(t_n, y_n) + h^2 A f(\boldsymbol{e}t_n + \boldsymbol{c}h, Y). \tag{1.2c}$$

From (1.2c) and (2.1) we derive

$$f(\boldsymbol{e}t_n + \boldsymbol{c}h, Y) = h^{-2} A^{-1}[Y - \boldsymbol{e}y_n - \boldsymbol{c}hy'_n - h^2 \boldsymbol{a} f(t_n, y_n)] = h^{-2} A^{-1} X \tag{2.12}$$

Relations (1.2), (2.1), (2.2) and (2.12) give us

$$u_{n+1} = y_n + hy'_n + b_0 h^2 f(t_n, y_n) + \sum_{i=1}^{k} \alpha_i X_i,$$

$$u'_{n+1} = y'_n + d_0 h f(t_n, y_n) + h^{-1} \sum_{i=1}^{k} \beta_i X_i \tag{2.13}$$

$$y_{n+1} = y_n + hy'_n + b_0 h^2 f(t_n, y_n) + \sum_{i=1}^{k} \alpha_i X_i^{[m]}$$

$$y'_{n+1} = y'_n + d_0 hf(t_n, y_n) + h^{-1} \sum_{i=1}^{k} \beta_i X_i^{[m]} \tag{2.14}$$

where α_i and β_i are the components of the vectors $\boldsymbol{\alpha} := \boldsymbol{b}^T A^{-1}, \boldsymbol{\beta} := \boldsymbol{d}^T A^{-1}$. In many cases, the corrector satisfies the ralations of stiff accuracy, i.e., $c_k = 1, b_0 = a_k$, and $\boldsymbol{b}^T A^{-1} = e_k^T$. In such cases, the step value u_{n+1} produced by the corrector is given by the last component of the stage vector, i.e., Y_k. This leads us to replacing the formula for u_{n+1} and y_{n+1} in (2.13), (2.14) by

$$u_{n+1} = y_n + hy'_n + b_0 h^2 f(t_n, y_n) + X_k$$
$$y_{n+1} = y_n + hy'_n + b_0 h^2 f(t_n, y_n) + X_k^{(m)}. \tag{2.15}$$

The three last formulas prove Theorem 2.2.

In the parallel iteration (2.2), in each step, per processor we have to solve a (general) system of nonlinear equations by Newton's method for dertermination of the stage-vector solution $X_i^{(\mu)}$ which approximates X_i so that we can control the iteration error by the following approximation

$$\| X^{(\mu)} - X \| = \| X^{(\mu)} - X^{(\mu-1)} \| . \tag{2.16}$$

In practice in order to economize the volume of computer memory and computational costs, we need the following theorem.

THEOREM 2.3. *Let the iteration process $X_i^{(\mu)}$ defined by (2.2c) converge to X_i (for $i = 1, 2, \ldots, k$) as μ tends to $+\infty$, then $[X_i^{(\mu)} - h^2 \sum_{j=1}^{k} a_{ij} f(t_n + c_j h, X_j^{(\mu)} + \boldsymbol{x}_j)]$ and $[X_i^{(\mu+1)} - X_i^{(\mu)}]$ (for $i = 1, 2, \ldots, k$) converge to 0 as μ tends to $+\infty$ of the same order.*

PROOF. Applying iteration process (2.2c) to the scalar test equation (**) we have

$$[X_i^{(\mu+1)} - X_i^{(\mu)}][1 - z\delta_i] = \omega_\mu [X_i^{(\mu)} - h^2 \sum_{j=1}^{k} a_{ij} f(t_n + c_j h, X_j^{(\mu)} + \boldsymbol{x}_j)],$$

$$i = 1, \ldots, k; \quad z = h^2\lambda \leq 0 \tag{2.17}$$

$$\| [X_i^{(\mu+1)} - X_i^{(\mu)}] \| = C \| X_i^{(\mu)} - h^2 \sum_{j=1}^{k} a_{ij} f(t_n + c_j h, X_j^{(\mu)} + x_j) \| \tag{2.18}$$

The relation (2.18) proves Theorem 2.3.

3. Construction of parallel RKN methods with high stability

In this section we will consider the construction of parallel methods which have high stability. In fact, the parallel RKN methods have been theoretically constructed in section 2. In this section we will choose the iteration parameters $\delta_i (i = 1, 2, \ldots, k)$ and $\omega_\mu (\mu = 1, 2, \ldots, m)$ of the iteration process (2.2c) in order to get some properties needed. The minimal requirement for these properties is the sufficient condition for the convergence of the iteration process (Theorem 2.1). We will set all relaxation parameters equal to one. Furthermore, we denote the spectral radius of the matrix M by $\rho(M)$, and define

$$\sigma(z) = \rho(I - H(z)) = \rho(I - [I - zD]^{-1}[I - zA]) \tag{3.1}$$

In [11] (also [9]), by setting all relaxation parameters ω_μ equal to one, we followed two iteration approaches: *Zaratonello* iteration and *stiff* iteration.

Zarantonello iteration mode requires the maximum by $z \leq 0$ of $\sigma(z)$ is minimized by the choice of $\delta_i (i = 1, \ldots, k)$. Let this min-max value be denoted by ρ.

Stiff iteration mode requires $\sigma(-\infty)$ is minimized by the choice of $\delta_i (i = 1, \ldots, k)$. Let this minimal value be denoted by $\rho(\infty)$. The convergence condition of the iteration process in this case is $\rho \leq C_0 < 1$.

By these two iteration modes, for three single direct collocation-based IRKN methods derived in [8] Dr(3/4,1), Dr(-1/5, 9/10) and Dr(-1/4,0,9/10,19/20,1) we have constructed three corresponding parallel methods with rather small magnitudes of ρ and $\rho(\infty)$ which ensure in some sence, fast convergence. However these parallel methods do not have good enough stability properties. Table 3.1. below lists the minimal value of m such that the parallel methods are A-stable for m equal to or greater than this critical value which is denoted by m_{crit}.

Table 3.1. Values of m_{crit} with explicit predictor I and implicit predictor II

Correctors	$(\delta)_i$-Iteration parameter	m_{crit}	
		Predictor I	Preditor II
Dr(3/4,1)	(0.16875,0.27777777778)	2	1
Dr(-1/5,9/10,1)	(0.0154593016,01137764435,0.1301060266)	4	5
Dr(-1/4,0,9/10,19/20,1)	(0.0073679545,0.0465886564, 0.0508800912,0.0553578642)	> 10	8

Now we choose the iteration parameters in such a way that the parallel RKN methods are convergent and A-stable for any number of outer iterations to be performed. The derivation of these iteration parameters is very difficult, and it is not feasible to obtain them by analytical means, so that we apply numerical search techniques. Using these search techniques we have successfully derived, for a number of A-stable IRKN methods of direct collocation type (the method $Dr(-1/5, 9/10, 19/20, 1)$ differs from [11]), optimal iteration parameters which ensure the convergence of the iteration process and give our parallel RKN methods (2.2) A-stability for any number of outer iterations per step and per processor. Table 3.2 and Table 3.3 below report the results of the search.

Table 3.2. "Optimal" iteration parameter(δ_i) and corresponding $\rho(\infty)$ and ρ

Correctors	p	r	k	(δ_i)-Iteration parameters	$\rho(\infty)$	ρ
Dr(3/4,1)	2	2	2	(0.175,0.275)	0.430	0.450
Dr(-1/5,9/10,1)	3	3	3	(0.12,0.144,0.18)	0.910	0.910
Dr(-1/5,9/10,19/20,1)	4	4	4	(0.75,0.8,0.85,0.9)	0.988	0.988
Dr(-1/4,0,9/10,19/20,1)	5	5	4	(0.125,0.45,0.65,0.8)	0.987	0.987

Note that the high stability of these parallel methods was shown on the numerical experiment in [12,pp 18-19]

Table 3.3. Values of m_{crit} with explicit predictor I and implicit predictor II

Correctors	p	r	k	(δ_i)-Iteration parameters	m_{crit}	
					Predictor I	Predictor II
Dr(3/4,1)	2	2	2	(0.175,0.275)	1	1
Dr(-1/5,9/10,1)	3	3	3	(0.12,0144,0.18)	3	1
Dr(-1/5,9/10,19/20,1)	4	4	4	(0.75,0.8,0.85,0.9)	>10	1
Dr(-1/4,0,9/10,19/20,1)	5	5	4	(0.125,0.45,0.65,0.8)	> 10	1

4. Numerical comparisons

In the numerical experiments we restrict the consideration to the new optimal parallel PC methods listed in Table 4.1 below and the already available parallel RKN and seequential SDIRKN methods in the literature (see [9],[11],[14])

Table 4.1. Survey of new high stability parallel PC methods

PC Methods	RKN corrector	k	p	r	Predictor	Iteration parameters (δ_i)	m_{crit}
D2(IO1)	Dr(3/4,1)	3	5	3	Predictor II	(0.175,0.275)	1
D3(IO1)	Dr(-1/5,9/10,1)	3	6	4	Predictor II	(0.12,0.144,0.18)	1
D3(IO1)	Dr(-1/5,9/10,19/20,1)	4	7	4	Predictor II	(0.75,0.8,0.85,0.9)	1
D5(IO1)	Dr(-1/4,0,9/10,19/20,1)	4	5	6	Predictor II	(0.125,0.45,0.65,0.8)	1

Since the new parallel methods are A-stable for any m, we drop the stability criterion $m \geq m_{\text{crit}}$ used in [9] and [11]. Using Theorem 2.2 and Theorem 2.3 the number of iterations m is determinated dynamically only by the condition for the iteration error as follows:

$$\text{Max}_{\,i} \parallel X_i^{(m)} - h^2 \sum_{j=1}^{k} a_{ij} f(t_n + c_j h, X_j^{(m)} + x_j) \parallel_\infty \leq C h^{p+1}, \qquad (4.1)$$

where p is the order of the local error of the corrector methods. The constant C is parameter independent on stepsize h but problem and method dependent.

Furthermore, in the table of results (as in [9] and [11]) M denotes the averaged number of sequential systems to be solved per unit interval and NCD denotes the number of minimal correct digits which is defined by

$$\text{NCD}\ (h) := -\log\ (\parallel \text{ global error at the endpoint of the integration interval } \parallel_\infty) \qquad (4.2)$$

We consider the model linear stiff Kramarz-type problem (see [2])

$$\frac{d^2y(t)}{dt^2} = \begin{pmatrix} \alpha-2 & 2(\alpha-1) \\ 1-\alpha & 1-2\alpha \end{pmatrix} y(t), y(0) = \begin{pmatrix} 2 \\ -1 \end{pmatrix}, y'(0) = (0), 0 \le t \le 100 \tag{4.3}$$

with exact solution $y(t) = (2cos(t), -cos(t))^T$. For this linear problem, where per processor, in each step only one Newton-iteration is required, the value of M may serve as computional costs. The matrix of the system has two negative eigenvalues -1 and $-\alpha$ so that when α is a large positive number, the system is really stiff . In our experiment we take $\alpha = 2500$. The results in Table 4.2 show that the new methods for $k = 3, 4, 5$ are more efficient than the methods of the same order from the literature.

Table 4.2 Values of NCD/M for problem (4.3)

PC Methods	s	k	p	r	$h = 1/5$	$h = 1/10$	$h = 1/20$	$h = 1/40$
$Norsett_2$[9]	2	1	3	1	0.9/10	1.8/20	2.7/40	3.6/80
SFB_2[9]	2	1	3	1	0.6/10	1.5/20	2.4/40	3.3/80
$Norsett_3$[9]	3	1	4	1	3.1/15	3.1/30	4.1/60	5.2/120
SFB_3[9]	3	1	4	1	2.4/15	3.6/30	4.8/60	6.0/120
B_4[9]	4	1	3	1	0.9/20	1.8/40	2.7/80	3.6/160
D2(IS1)[11]	2	2	2	2	0.9/14	1.4/27	1.7/40	2.4/80
D3(ES4)[11]	3	3	3	3	1.8/20	2.7/42	3.7/80	4.6/160
D4(IS8)[11]	5	4	5	5	5.3/45	6.8/90	8.3/180	9.8/360
D2(I01)	2	2	2	2	0.9/14	1.4/27	1.7/40	2.4/80
D3(I01)	3	3	3	3	1.8/15	2.7/29	3.6/59	4.5/117
D4(I01)	4	4	4	4	3.8/25	5.0/40	6.0/80	7.3/159
D5(I01)	5	4	5	5	5.3/34	6.8/59	8.2/100	9.8/196

REFERENCES

[1] K. Dekker and J.G. Verwer, *Stability of Runge-Kutta methods for stiff nonlinear differential equations*, North-Holland, Amsterdam (1984).

[2] L. Kramarz, *Stability of collocation methods for the numerical solutions of* $y'' = f(x, y)$, BIT **20** (1980), 215-222.

[3] E.Hairer, *Unconditionnally stable methods for second order differential equations*, numer. Math. **32** (1979), 373-379.

[4] E.Hairer, S.P. Norsett and G.Wanner, *Solving ordinary differential equations, I.Nonstiff problems,*, Springer-Verlag, Berlin, 1987 (1987).

[5] P.J. van der Houwen, B.P. Sommeijer, *Parallel iteration of high-order Runge-Kutta methods with stepsize control*, J. Comp. Appl. Math. **29** (1989), 111-127.

[6] P.J. van der Houwen, B.P. Sommeijer and W. Couzy, *Embedded diagonally implicit Runge-Kutta algorithms on parallel computer*, Math. Comp. (to appear) (1992).

[7] P.J. van der Houwen, B.P. Sommeijer, *Iterated Runge -Kutta methods on parallel computers*, SIAM J. Sci. Stat. Comp. **12** (1991), 1000-1028.

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

[9] P.J. van der Houwen, B.P. Sommeijer and Nguyen huu cong, *Parallel diagonally implicit Runge-Kutta-Nyström methods*, Appl. Numer. Math. **9** (1992), 111-131.

[10] I. Lie, *Some aspects of parallel Runge-Kutta-Nyström methods*, report No. 3/87, Division Numerical Mathematics, University of Trondheim.

[11] Nguyen huu Cong, *Parallel implicit Runge-Kutta-Nyström methods of direct collocation type for stiff initial-value problems*, J. Math. (to appear) **2** (1992).

[12] Nguyen huu Cong, *Stability of parallel iteration of direct collocation-based implicit Runge-Kutta-Nyström methods*, Preprint No 6(1991), Institute of Computer Science, National Centre for Scientific Research of Vietnam.

[13] S.P Norsett and H.H. Simonsen, *Semi-explicit Runge-Kutta-Nyström methods*, Report Mathematics and Computation No 6/74 1974, Department of Mathematics, University of Trondheim.

[14] S.P. Norsett and H.H. Simonsen, *Aspects of parallel Runge-Kutta methods, in numerical methods for ordinary differential equations*, A. Bellen, C.W. Gear and E. Russo (eds.), Proceedings of the L'Aquila conference, 1987, Lecture Notes in Mathematics, Springer-Verlag **1386** (1989), 103-117.

[15] I.F. Shapine, *Implementation of implicit formulas for the solution of ODE s*, SIAM J. Sci. Stat. Comp. **1** (1980), 103-118.

[16] P.W. Sharp, J.H. Fine and K. Burrage, *Two-stage diagonally implicit Runge-Kutta-Nyström methods of orders three and four*, IMA J. Numer. Anal. **10** (1990), 489-504.

[17] R.S. Varga, *A comparision of the successive overrelaxation method and semi-iterative methods using Chebyshev polynomials*, SIAM J. Appl. Math. **5** (1975), 39-47.

FACULTY OF MATHEMATICS, MECHANICS AND INFORMATICS
UNIVERSITY OF HANOI