Journal of Mathematics *Tạp chí Toán học* **20(1)(1992)**, 48—62

PARALLEL IMPLICIT RUNGE-KUTTA-NYSTROM METHODS OF DIRECT COLLOCATION TYPE FOR STIFF INITIAL-VALUE PROBLEMS

NGUYEN HUU CONG

Abstract. *In this paper, we study parallel iteration of predictor-corretor methods (Parallel PC methods) with direct collocation-based implicit Runge-Kutta-Nystrom correctors for solving stiff initial-value problems for special second-order, ordinary differential equations (ODEs) on parallel computers. These iteration mehtods are such that in each step, the iterated method belongs to the class of diagonally implicit Runge-Kutta-Nystrom methods (DIRKN methods). By a suitable choice of the parameters in the iteration process, the parallel methods constructed in this paper are rather efficient for solving stiff initial-value problems on parallel computers.*

1. INTRODUCTION

We shall be concerned with the integrating the initial-value problem for systems of special second-order, ordinary equations (ODEs) of dimension d

$$y''(t) = g(y(t)),\ y(t_0) = y_0, y'(t_0) = y'_0,\ y : \mathrm{R} \to \mathrm{R}^d,\ g : \mathrm{R}^d \to \mathrm{R}^d,\ t_0 \le t \le t_{end}.$$

Important test examples from this class of problems originate from structural mechanics. Such problems belong usually to the stiff class. Ideal method for solving this class of problems is the method which have, in addition to a high step point order and the property of A-stability, a high stage orders, in order to avoid the effect of order reduction (cf. [1]).

In this paper, we consider parallel integration methods based on parallel iteration of fully implicit Runge-Kutta-Nystrom (RKN) methods of direct collocation type. Such RKN methods possess the largest possible stage order, so that we automatically achieve high stage orders if the method is used is sufficiently accurate (for an extensive investigation of such RKN methods see [8]).

We shall investigate *parallel iteration* methods which are such that, after a finite number of iterations, the method belongs to the class of diagonally implicit Runge-Kutta- Nystrom methods (DIRKN methods). Adopting the terminology used for iterating implicit linear multistep methods, we shall call the implicit RKN method the *corrector* and the method used for starting the iteration the *predictor*. The parallel iteration process will be called *parallel predictor – corrector*(PC) method.

The large number of stages of this *parallel PC method* increases with the number of iterations and may vary from step to step depending on the convergence criterion but can be computed in parallel on multi-processor computers. An other advantage is that on each processor, the method behaves as a *single – diagonal – implicit* RKN method (SDIRKN), that is, only one LU decomposition per processor is required. Thirdly, we can reduce the number of iterations per step by a suitable choice of the iteration parameters of the iteration process. For three single A stable IRKN correctors derived in [8] by direct collocation techniques, we have constructed parallel PC methods which are very efficient for stiff initial-value problems. The use of direct collocation-based *A*-stable corrector methods guarantees high stage order, so that the phenomenon of order reduction, exhibited in many problems with large Lipschitz constants, does not deteriorate the accuracy of the methods. The numerical experiments clearly show that the parallel iteration methods proposed in this paper are much more efficient than sequential SDIRKN methods.

RKN method

We consider RKN correctors of the form

$$y_{n+1} = y_n + hy'_n + b_0h^2g(y_n) + h^2\sum_{i=1}^{k} b_ig(Y_i),$$

$$y'_{n+1} = y'_n + d_0hg(y_n) + h\sum_{i=1}^{k} d_ig(Y_i), \tag{1.2}$$

$$Y_i = y_n + c_ihy'_n + a_ih^2g(y_n) + h^2\sum_{j=1}^{k} a_{ij}g(Y_j),\ i = 1,\dots,k,$$

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

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

where $b = (b_i)$, $c = (c_i)$ and $d = (d_i)$ are k-dimensional vectors, and $A = (a_{ij})$ is an k-by-k matrix. We always assume that the matrix A is nonsingular. If the vector a does not vanish, then (1.2) presents an $(s = k + 1)$-stage RKN method requiring k *implicit* stages and one explicit stage. If $a = 0$, then (1.2) reducex to the general $(s = k)$- stage RKN method with s implicit stages. For a discussion of the order of accuracy p and the stage order r of RKN methods, the reader is referred 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) and Dr(-1/4, 0, 9/10, 19/20, 1). Here (3/4, 1), (-1/5, 9/10, 1) and (-1/4, 0, 9/10, 19/20, 1) are the vectors of collocation points (more detail about this methods and direct collocation-based IRKN methods, please see [8]). The use of non-A-stable RKN correctors will be investagated in an other paper in order to exploit the performance in high step point order.

2. PARALLEL ITERATION OF PC METHODS

We shall construct integration 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, the stage vector $Y = (Y_i)$. Our aim is to construct solution methods that run fast on parllel computers. In the case where all eigenvalues of the Jacobian matrix are close to the origin, the stage vector equation in (1.2) can be solved by fixed point iteration which is well suited for implementation on parallel computer. For first-order ODEs this has been discussed in [12], [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 leads to the same degree of implicitness as occuring in SDIRKN methods. These processes are similar to the *stiff iteration method* applied in [7] and parallel PC iteration applied in [9].

2.1. Iteration of the stage-vector equation

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

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

We shall compute iterates $X_i^{(\mu)}$, rather than the iterates $Y_i^{(\mu)}$, 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 in (1.2) reads

$$X_i = h^2 \sum_{j=1}^{k} a_{ij} g(X_j + x_j), \; i = 1, \ldots, k. \tag{1.2'}$$

For each of these equation, we define the iteration process

$$X_i^{(\mu)} - h^2 \delta_i g(X_i^{(\mu)} + x_i) = X_i^{(\mu-1)} - \delta_i h^2 g(X_i^{(\mu-1)} + x_i) - \omega_\mu [X_i^{(\mu)} - h^2 \sum_{j=1}^{k} a_{ij} g(X_j^{(\mu)} + x_j)], \tag{2.2}$$

where $i = 1, \ldots, k$; $\mu = 1, \ldots, m$. Here, the ω_μ are relaxation parameters and the δ_i are iteration parameters which are assumed to be positive. The follwing lemma is evident.

Lemma 2.1. *Suppose that the iteration process (2.2) converges, then $X_i^{(\mu)}$ converges to the i-th component of the solution of the stage vector equation (1.2') i. e. to X_i.*

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

Definition of the step values. Suppose that we adopt $Y_i^{(m)} = X_i^{(m)} + x_i$ as a sufficiently accurate approximation to the exact stage vector solution Y of the corrector (1.2). Then, the most natural way to approximate the step values y_{n+1} and y'_{n+1} in (1.2) is to define these values according to the formulas (see [8])

$$y_{n+1} = y_n + h y'_n + b_0 h^2 g(y_n) + h^2 \sum_{i=1}^{k} b_i g(Y_i^{(m)}),$$

$$y'_{n+1} = y'_n + d_0 h g(y_n) + h \sum_{i=1}^{k} d_i g(Y_i^{(m)}), \tag{2.3}$$

(in order to avoid confusion, we shall from now on denote the corrector solution values obtained from y_n and y'_n by u_{n+1} and u'_{n+1}). However, the presence of the righthand side evaluations in these formulas may give rise to loss of accuracy in the case of stiff problems (see [13]). This difficulty can be overcome by applying a similar approach as proposed in [7] for the implementation of implicit RK methods. For simplicity, we describe this approach for a scalar equation $y'' = g(y)$.

Lemma 2.2. *Defining $Y = (Y_i)$ and $G = (g(Y_i))$, the corrector (1.2) can be written in the form*

$$u_{n+1} = y_n + hy'_n + b_0h^2g(y_n) + h^2b^TG,$$
$$u'_{n+1} = y'_n + d_0hg(y_n) + hd^TG, \text{ with}$$
$$G = h^{-2}A^{-1}[Y - ey_n - chy'_n - ah^2g(y_n)],$$

where e is k-dimensional vector with unit entries.

P r o o f. If we use the notation in which for any vector $v = (v_i)$, the $f(v)$ denotes the vector with entries $f(v_i)$ then the corrector (1.2) can be written in the following form:

$$u_{n+1} = y_n + hy'_n + b_0h^2g(y_n) + h^2b^Tg(Y),$$
$$u'_{n+1} = y'_n + d_0hg(y_n) + hd^Tg(Y), \tag{1.2''}$$
$$Y = ey_n + chy'_n + h^2ag(y_n) + h^2Ag(Y)$$

From last relation we derive $g(Y) = G = h^{-2}A^{-1}[Y - ey_n - chy'_n - ah^2g(y_n)]$ that leads us to lemma 2.2. □

This representation shows that we can eliminate the righthand side evaluations and that u_{n+1} and u'_{n+1} can be expressed solely in terms of the stage vector Y. Now we will compute y_{n+1} and y'_{n+1} according to these formulas with Y replaced by $Y^{(m)}$. Returning to systems of ODEs and to the notation $X_i^{(m)}$, we obtain.

$$y_{n+1} = y_n + hy'_n + b_0h^2g(y_n) + \sum_{i=1}^{k} \alpha_i X_i^{(m)},$$
$$y'_{n+1} = y'_n + d_0hg(y_n) + h^{-1}\sum_{i=1}^{k} \beta_i X_i^{(m)}, \tag{2.4}$$

where the α_i and β_i are the components of the vectors

$$\alpha := b^TA^{-1}, \ \beta := d^TA^{-1}.$$

In many cases the corrector satisfies the relation of stiff accuracy, i.e., $c_k = 1, b_0 = a_k$ and $b^TA^{-1} = e_k^T$. 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. by Y_k. This leads us to replacing the formula for y_{n+1} in (2.4) by

$$y_{n+1} = y_n + hy'_n + b_0 h^2 g(y_n) + X_k^{(m)}. \tag{2.4'}$$

2.2. The predictor

In order to start the iteration (2.2), we need a predictor to compute the initial approximations $X_i^{(0)}$. For more details about the predictors we refer to [9]. In this paper we follow two more attractive predictors below:

Explicit predictor I with $X_i^{(0)} = -a_i h^2 g(y_n)$

Implicit predictor II with $X_i^{(0)} = -a_i h^2 g(y_n) + h^2 \delta_i g(X_i^{(0)} + x_i)$.

In view of stability, an important property of these predictors is the degree of amplification of stiff components (here, stiff components are understood to be eigenvector components corresponding to large, negative eigenvalues of $\partial g/\partial y$). We study this property by applying it to the scalar test equation $y'' = \lambda y$.

Definition 2.1. We say that the predictor $X^{(0)}$ is of order p^* if $X - X^{(0)} = O(h^{p^*+1})$

Theorem 2.1. *The predictor I and predictor II are of order one, i. e.,*

$$\varepsilon^{(0)} = X - X^{(0)} = O(h^2)$$

P r o o f. From stage vector equation (1.2'), we derive

$$X = zA(X + ey_n + chy'_n + zay_n) \text{ with } z = \lambda h^2, \text{ or}$$
$$X = [I - zA]^{-1} zA(ey_n + chy'_n + zay_n). \tag{2.5}$$

From (2.5) and observing that $z = O(h^2)$, Theorem 2.1. easily follows. □

2.3. The iteration error

Definition 2.2. We shall say that the order of the iteration error of the PC method (2.1)-(2.4) equals q if

$$u_{n+1} - y_{n+1} = O(h^{q+1}), u'_{n+1} - y'_{n+1} = O(h^{q+1}), \tag{2.6}$$

where (u_{n+1}, u'_{n+1}) and (y_{n+1}, y'_{n+1}) denote the step values obtained from the values (y_n, y'_n) by respectively solving the corrector equation and by performing a finite number of iterations.

The iteration error associated with (2.1)-(2.4) can be studied by applying it to the scalar test equation $y'' = \lambda y$, where λ runs through the eigenvalues of $\partial g \partial y$.

Theorem 2.2 *Defining the iteration error*

$$\varepsilon^{(\mu)} := X - X^{(\mu)}, \quad \text{where} \quad X := (X_i), X^{(\mu)} := (X_i^{(\mu)}), \tag{2.7}$$

we have following error relation:

$$\varepsilon^{(m)} = P_m(H(z))\varepsilon^{(0)} \text{ with}$$
$$P_m(x) = \prod_{\mu=1}^{m}(1 - \omega_\mu x) \text{ and } H(z) := [I - zD]^{-1}[I - zA],\ z := \lambda h^2 \tag{2.8}$$

P r o o f. We can deduce from (2.2) that the iteration error (2.7) satisfies the recursion

$$\varepsilon^{(\mu)} = [I - \omega_\mu H(z)]\varepsilon^{(\mu-1)}, H(z) := [I - zD]^{-1}[I - zA],\ z := \lambda h^2, \mu = 1, ..., m,$$

where D is the diagonal matrix with diagonal entries δ_i. Hence, relation (2.8) easily follows. □

Definition 2.3. The matrix $P_m(H(z))$ in (2.8) will be called the stage vector iteration matrix.

Theorem 2.3. *Let the predictor be of order p^* and let*

$$P_m(x) = (1 - x)^{q^*} Q_{m-q^*}(x),\ Q_{m-q^*}(1) \neq 0.$$

Then, for any choice of the matrix D, the order q of the iteration error of the PC method (2.1)-(2.4) is given by $q = 2q^* + p^* - 1$.

The proof of this theorem is given in [9]

In the following, we use the notation

$$w_{n+1} := \begin{pmatrix} u_{n+1} \\ hu'_{n+1} \end{pmatrix},\ v_{n+1} := \begin{pmatrix} y_{n+1} \\ hy'_{n+1} \end{pmatrix}. \tag{2.9}$$

In terms of these vectors, we can derive an error equation of the form

$$w_{n+1} - v_{n+1} = E_m(z)v_n, \tag{2.10}$$

where the error matrix $E_m(z)$ is a 2-by-2 matrix determined by the RKN parmeters and the matrix D.

Definition 2.4. The matrix $E_m(z)$ defined in (2.10) will be called the iteration error matrix of the parallel PC method (2.1)-(2.4).

In the following section we need specified structure of iteration error matrix $E_m(z)$ for deriving the stability function of parallel PC methods. We have the following theorem.

Theorem 2.4. *The iteration error matrix of parallel PC methods $E_m(z)$ from relation (2.10) assumes the form below:*

$$E_m(z) := \begin{pmatrix} p^T P_m(H(z))k_1(z) & p^T P_m(H(z))k_2(z) \\ d^T A^{-1} P_m(H(z))k_1(z) & d^T A^{-1} P_m(H(z))k_2(z) \end{pmatrix} \tag{2.11}$$

where $p^T = b^T A^{-1}$ for nonstiffly accurate correctors, and $p^T = e_k^T$ for stiffly accurate correctors.

For predictor I, $k_1(z) = [I - zA]^{-1} zA[e + za] + za$, $k_2(z) = [I - zA]^{-1} zAc$.

For predictor II, $k_1(z) = [I - zA]^{-1}[e + za] - [I - zD]^{-1}[zD(e + za) - za]$, $k_2(z) = \{[I - zA]^{-1} - [I - zD]^{-1}\}c$.

P r o o f. From the step value formulae (2.4), (2.4'), (1.2") and Theorem 2.2 (formula (2.8)) it follows that

$$u_{n+1} - y_{n+1} = p^T P_m(H(z))\varepsilon^{(0)},\ hu'_{n+1} - hy'_{n+1} = d^T A^{-1} P_m(H(z))\varepsilon^{(0)}, \tag{2.12}$$

where $p^T = b^T A^{-1}$ for nonstiffly accurate correctors and $p^T = e_k^T$ for stiffly accurate correctors. For the predictor I, $X^{(0)} = -zay_n$, using formula (2.5) we have

$$\varepsilon^{(0)} = X - X^{(0)} = [I - zA]^{-1} zA(ey_n + chy'_n + zay_n) + zay_n = k_1(z)y_n + k_2(z)hy'_n, \tag{2.13}$$

with $k_1(z) = [I - zA]^{-1} zA[e + za] + za$, $k_2(z) = [I - zA]^{-1} zAc$. For the predictor II, $X^{(0)} = -zay_n + zD[X^{(0)} + x]$ with $x = ey_n + chy'_n + zay_n$. Hence,

$$X^{(0)} = [I - zD]^{-1}[zD(ey_n + chy'_n + zay_n) - zay_n]$$

$$\begin{aligned} \varepsilon^{(0)} = X - X^{(0)} &= [I - zA]^{-1} zA[ey_n + chy'_n + zay_n] - \\ &- [I - zD]^{-1}[zD(ey_n + chy'_n + zay_n) - zay_n] = \\ &= k_1(z)y_n + k_2(z)hy'_n \end{aligned} \tag{2.13'}$$

with $k_1(z) = [I - zA]^{-1}[e + za] - [I - zD]^{-1}[zD(e + za) - za]$, $k_2(z) = \{[I - zA]^{-1} - [I - zD]^{-1}\}c$. Relations (2.12), (2.13), (2.13') prove the theorem 2.4. □

2.4. Choise of iteration parameters

Before discussing the choise of iteration parameters (matrix D), we denote the spectrum of $H(z)$ by $\Lambda(H(z))$, and we define

$$\rho(z) := \text{Max}\{|\lambda - 1| : \lambda \in \Lambda(H(z))\},$$
$$\rho := \text{Max}\{|\lambda - 1| : \lambda \in \Lambda(H)\},$$
$$\Lambda(H) := \{\Lambda(H(z)) : -\beta \leq z \leq 0\}. \tag{2.14}$$

From results reported in Subsection 2.3 (Theorems 2.2, 2.3, 2.4) we choose the relaxation parameters ω_μ equal to 1 and follow two iteration mode: *stiff iteration approach and Zarantonello iteration* approach. (see[7], [9]). □

2.4.1. **Stiff iteration.** In this case the matrix D is chosen is such a way that $\Lambda(H(-\infty))$ is contained in a circle with minimal radius $\rho(-\infty)$ and centered at 1. The following theorem holds for $k = 2$:

Theorem 2.5. *Let $k = 2$, then the following assertions hold for the stiff iteration method:*

(a) if $det(A) > 0$ and if either $\{a_{12}a_{21} \leq 0$ and $a_{22} > 0\}$ or $\{a_{11} > 0$ and $a_{22} \leq 0\}$, then there exists a matrix D with positive entries such that $\rho(-\infty) = 0$.

(b) if (a) holds, then one eigenvalue of $H(z)$ equals 1 for all z.

(c) if (a) holds and if $Tr(A) > -2det(A)$, then the eigenvalues of $H(z)$ are real and positive for all negative z.

P r o o f. (a) For $k = 2$ the value of $\rho(-\infty)$ vanishes if the matrix $H(-\infty) - I = D^{-1}A - I$ has zero eigenvalues. This can be achieved by choosing

$$\delta_1 = \frac{\det A}{a_{22}}(1 + \sqrt{1 - \frac{a_{11}a_{22}}{\det A}}),\ \delta_2 = \frac{2\det A - \delta_1 a_2 2}{a_{11}}.$$

By an elementary calculation assertion (a), (b), (c) can now be verified (more detail see [9]). □

2.4.2. **Zarantonello iteration.** When all relaxation parameters equal to 1, the optimal choice of the set $\Lambda(H)$ is a circle centered at 1 with minimal radius ρ. This follows from a lemma of Zarantonello (cf.[15]), stating that the spectral radius of $P_m(H(z))$ is minimized if P_m has all its zeros at the center of the circle containing the eigenvalues of $H(z)$. We shall call this iteration mode *Zarantonello iteration.*

Using Theorem 2.5 for Dr(3/4, 1) we can derive matrix D with $\rho(-\infty) = 0$. For two remaining correctors Dr(-1/5, 9/10, 1) and Dr(-1/4, 0, 9/10, 19/20, 1) we did not succed in deriving the optimal matrix D by analytical methods, so that we used the numerical search techniques. The results of search for both two iterations modes are listed in Table 2.1. In this table, we list the optimal D-matrix,

the values of ρ and $\rho(-\infty)$ for each corrector.

Table 2.1. "Optimal" iteration parameters δ and corresponding $\rho(-\infty), \rho$ values

Correctors	P	R	K	δ^T – iteration parameters	$\rho(-\infty)$	ρ
Dr (3/4, 1)	2	2	2	(0.16875, 0.2777777778)	0.000	0.500
Dr(-1/5, 9/10, 1)	3	3	3	(0.0154593016, 0.1137764435, 0.1301060266)	0.009	0.367
Dr(-1/4, 0,9/10, 19/20, 1)	5	5	4	(0.0073679545, 0.0465886564, 0.0508800912, 0.0553578642)	0.279	0.380

2.5. Stability

Regarding the stability of parallel PC methods, we have the following theorem

Theorem 2.6. *Using the notation in Subsection 2.3. i. e.* $v_{n+1} := \begin{pmatrix} y_{n+1} \\ hy'_{n+1} \end{pmatrix}$ *where* y_{n+1} *is an approximate solution by parallel PC method, the stability relation below holds*

$$v_{n+1} = [M(z) - E_m(z)]v_n, \tag{2.15}$$

where matrix $M(z) := \begin{pmatrix} 1 + zb_0 + zb^T(I - Az)^{-1}[e + za] & 1 + zb^T(I - Az)^{-1}c \\ zd_0 + zd^T(I - Az)^{-1}[e + za] & 1 + zd^T(I - Az)^{-1}c \end{pmatrix}$ *is stability matrix of IRKN methods and matrix* $E_m(z)$ *is defined by (2.11)*

P r o o f. If $w_{n+1} := (u_{n+1}, hu'_{n+1})$ where u_{n+1} is a solution by RKN method, then $w_{n+1} = M(z)v_n$ (see [8]). By substituting into (2.10) we obtain (2.15). □

Definition 2.5. We call the matrix $M(z) - E_m(z)$ the *stability matrix* of parallel PC methods and its spectral radius its *stability function* $R_m(z)$, i. e.

$$R_m(z) := \rho([M(z) - E_m(z)]). \tag{2.19}$$

Since stability plays a crucial role in the overall performance of the PC methods, it is of interest to compute the minimal value of m such that the parallel PC method is stable for all z in the interval $(-\infty, 0]$ and for all m equal to or greater than this value, i. e., the PC methods are A-stable for all m greater than or equal this value. Let us denote this critical value of m by m_{crit} and let m_1 denote the minimal number of systems (of dimension d) that are to be solved per step and per processor such that the PC method is A-stable. Thus $m_1 = m_{\text{crit}}$ for the predictors I and, and $m_1 = m_{\text{crit}} + 1$ for the predictors II. In Table 2.2 the values of m_{crit} are listed for three parallel PC methods with three Dr(3/4, 1). Dr(-1/5, 9/10, 1) and Dr(-1/4, 0, 9/10, 19/20, 1) correctors using the predictor I and the predictor II. For each k, the minimal values are indicated by bold face.

Table 2.2. Values of m_{crit} with explicit preddictor I and implicit predictor II

Correctors	δ^T – iteration parameters	$m_{\text{crit}}$for predictor I	$m_{\text{crit}}$for predictor II
Dr(3/4, 1)	(0.16875, 0.2777777778)	2	1
Dr(-1/5, 9/10, 1)	(0.0154593016, 0.1137764435, 0.1301060266)	4	5
Dr(-1/4, 0, 9/10, 19/20, 1)	(0.0073679545, 0.0465886564, 0.0508800912, 0.0553578642)	> 10	8

3. NUMERICAL COMPARISONS

In the numerical experiments we restrict the consideration to only the three

best parallel PC methods listed in Table 3.1 below and the already available sequential SDIRKN methods (see [9], [11], [14]).

Table 3.1. Survey of new selected parallel PC methods.

PC Method	IRKN corrector	k p r	Predictor	Iteration parameters δ^T	m_{crit}
D2(IS1)	Dr(3/4, 1)	2 2 2	Predictor II	(0.16875, 0.277777778)	1
D3(ES4)	Dr(-1/5, 9/10, 1)	3 3 3	Predictor I	(0.01545593016, 0.1137764435, 0.1301060266)	4
D4(IS8)	Dr(-1/4, 0, 9/10, 19/20, 1)	4 5 5	Predictor II	(0.0073679545, 0.0465886564, 0.0508800912, 0.0553578642)	8

In all experiments we used the convergence criterion proposed in [9], i. e., the number of outer iteration m was determinated dynamically by the stability criterion $m \geq m_{\text{crit}}$ together with a condition on the iteration error as following:

$$m \geq m_{\text{crit}} \text{ and } \text{Max}_i \| X_i(m) - h^2 \sum_{j=1}^{k} a_{ij} g(X_j(m) + x_j) \|_\infty \leq Chp^{+1}, \qquad (3.1)$$

where p is order of local error of the corrector methods, constant C is parameter independent of the stepsize h. In our numerical experiments we used $C = 10^{-2}$.

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

$$\text{NCD}(h) := -\log(\|\text{global error at the endpoint of the integration interval}\|_\infty) \qquad (3.2)$$

3.1. Lincar stiff Kramarz problem

Consider the model linear stiff problem (see [2]):

$$y''(t) = \begin{pmatrix} 2498 & 4998 \\ -2499 & -4999 \end{pmatrix} y(t),\ y(0) = \begin{pmatrix} 2 \\ -1 \end{pmatrix},\ y'(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix},\ 0 \le t \le 100, \quad (3.3)$$

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 computational costs. The results in Table 3.2 show that D3(ES4) is more efficient than sequential three-order methods, D4(IS8) is the most efficient.

Table 3.2. Values of NCD/M for problem (3.3)

Methods	k	p	r	h=1/5	h = 1/10	h=1/20	h=1/40	h = 1/80
$Norsett_2$	2	3	1	0.9/10	1.8/20	2.7/40	3.6/80	4.5/160
SFB_2	2	3	1	0.6/10	1.5/20	2.4/40	3.3/80	4.2/160
$Norsett_3$	3	4	1	3.1/15	3.1/30	4.1/60	5.2/120	6.4/240
SFB_3	3	4	1	2.4/15	3.6/30	4.8/60	6.0/120	7.2/240
B_4	4	3	1	0.9/20	1.8/40	2.7/80	3.6/160	4.5/320
D2(IS1)	2	2	2	0.9/14	1.4/27	1.7/40	2.4/80	3.0/160
D3(ES4)	3	3	3	1.8/20	2.7/42	3.7/80	4.6/160	5.5/320
D4(IS8)	4	5	5	5.3/45	6.8/90	8.3/180	9.8/360	11.3/720

3.2. Fehlberg problem

Consider the nonlinear equation (cf.[9]):

$$y'' = Jy(t),\ J = \begin{pmatrix} -4t^2 & -2/r(t) \\ 2/r(t) & -4t^2 \end{pmatrix},\ r(t) = \|y(t)\|_2,\ \sqrt{\pi/2} \le t \le 3\pi, \quad (3.4)$$

where $y(t) = (y_1(t), y_2(t))^T$, $\|y(t)\|_2 = \sqrt{(y_1(t))^2 + (y_2(t))^2}$. Problem (6.4) has the exact solution $y(t) = (\cos(t^2), \sin(t^2))^T$. In this difficult experiment, D3(ES4)

and the three-order sequential methods are comparable, D4(IS8) is again the most efficient.

Table 3.3. Values of NCD/M for problem (3.4)

PC method	k	p	r	h=1/10	h=1/20	h=1/39	h=1/78	h=1/156
Norsett_2	2	3	1	0.1/20	0.1/39	0.6/78	1.5/157	2.4/313
SFB_2	2	3	1	0.1/20	0.1/39	0.4/78	1.2/157	2.1/313
Norsett_3	3	4	1	-0.1/29	0.4/59	1.6/117	2.7/235	3.9/470
B_4	4	3	1	0.1/39	0.1/78	0.6/157	1.5/313	2.4/627
D2(IS1)	2	2	2	0.0/102	-0.1/130	0.3/188	1.0/312	1.6/552
D3(ES4)	3	3	3	0.1/74	0.7/124	1.5/212	2.5/398	3.4/733
D4(IS8)	4	5	5	1.3/92	2.9/180	4.4/351	5.9/702	7.4/1404

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 solution of $y'' = f(x, y)$, BIT 20,(1980), pp. 215-222.
3. Hairer: Unconditionaly stable method for second order differential equations, Numer. Math. 32,(1979), pp. 373-379.
4. E. Hairer, S.P. Norsctt and G. Wanner: Solving ordinary differential equations I, Nonstiff problems, Springer-Verlag, Berlin, (1987).
5. P. J. van der Houwen and B. P. Sommeijer: Parallel iteration of high-order Runge-Kutta methods with stepsize control, J. Comp, Appl. Math. 29, (1989), pp. 111-127.
6. P. J. van der Houwen, B. P. Sommeijer and W. Couzy: Embedded diagonally implicit Runge-Kutta algorithms on parallel computers, Report NM-8912, Centre for Mathematics and Computer Science, Amsterdam (submitted for publication in Math. Comp.)(1989).
7. P. J. van der Houwen, B. P. Sommeijer: Iterated Runge-Kutta methods on parallel computers, Report NM-R9001, Centre for Mathematics and Computer Science, Amsterdam (to appear in SISSC.) (1990).

8. P. J. van der Houwen, B. P. Sommeijer and Nguyen Huu Cong: Stability of collocation-based Runge-Kutta-Nystrom methods, Report NM-R9016, Centre for Mathematics and Computer Science, Amsterdam.(to appear in BIT)(1990).
9. P. J. van der Houwen, B. P. Sommeijer and Nguyen Huu Cong: Parallel diagonally implicit Runge-Kutta-Nystrom methods, Report NM-R9103, Centre for Mathematics and Computer Science, Amsterdam, (to appear in Apl. Num. Math), (1991).
10. I. Lie: Some aspects of parallel Runge-Kutta methods, Report No. 3/87, Division Numerical Mathematics, University of Trondheim, (1987).
11. S. P. Norsett: Semi-explicit Runge-Kutta methods, Report Mathematics and Computation No. 6/74, Dept. of Mathematics, University of Trondheim, (1974).
12. S.P. Norsett, H. H. Simonsen: Aspects of Differential Equations, L'Aquila, 1987.
13. L. F. Shampine: Implementation of implicit formulas for the solution of ODEs, SIAM J. Sci. Stat. Comp. 1, pp. 103-118, (1980).
14. Sharpe, Fine and Burrage: Two-stage and three-stage R-stable, P-stable and RL-stable diagonally implicit Runge-Kutta-Nystrom methods of orders three and four, (submitted for publication in JIMA), (1990).
15. R. S. Varga: A comparison of the successive overrelaxation method and semi-iterative methods using Chebyshev polynomials, SIAM J. Appl. Math. 5, pp. 39-47, (1957).

Faculty of Mathematics,
Mechanics and Informatics,
University of Hanoi

Received August 20, 1991