

A-stable diagonally implicit Runge–Kutta–Nyström methods for parallel computers

Nguyen huu Cong *

*Centre for Mathematics and Computer Science, P. O. Box 4079,
1009 AB Amsterdam, The Netherlands and
Faculty of Mathematics, Mechanics and Informatics, University of Hanoi,
Thuong dinh, Dong Da, Hanoi, Vietnam*

Communicated by C. Brezinski

Received 21 April 1992; revised 12 November 1992

In this paper, we study diagonally implicit Runge–Kutta–Nyström methods (DIRKN methods) for use on parallel computers. These methods are obtained by diagonally implicit iteration of fully implicit Runge–Kutta–Nyström methods (corrector methods). The number of iterations is chosen such that the method has the same order of accuracy as the corrector, and the iteration parameters serve to make the method at least A-stable. Since a large number of the stages can be computed in parallel, the methods are very efficient on parallel computers. We derive a number of A-stable, strongly A-stable and L-stable DIRKN methods of order p with $s^*(p)$ sequential, singly diagonal-implicit stages where $s^*(p) = [(p + 1)/2]$ or $s^*(p) = [(p + 1)/2] + 1$, $[\cdot]$ denoting the integer part function.

Keywords: Diagonally implicit Runge–Kutta–Nyström methods, predictor–corrector methods, parallelism.

Subject classification: 65M12, 65M20.

1. Introduction

Consider the initial-value problem for systems of special second-order, ordinary differential equations (ODEs) of dimension d

$$\begin{aligned} y''(t) &= f(y(t)), & y(t_0) &= y_0, & y'(t_0) &= y'_0, \\ y : \mathbb{R} &\rightarrow \mathbb{R}^d, & f : \mathbb{R}^d &\rightarrow \mathbb{R}^d, & t_0 &\leq t \leq t_{\text{end}}. \end{aligned} \quad (1.1)$$

One possibility for solving such problems is the use of singly diagonal-implicit Runge–Kutta–Nyström methods (SDIRKN methods). Compared with linear mul-

* These investigations were supported by the University of Amsterdam with a research grant to enable the author to spend a total of two years in Amsterdam.

timestep methods (LM methods), SDIRKN methods have the disadvantage of requiring the solution of a sequence of implicit systems of dimension d per step, whereas LM methods require the solution of only one such system per step. On the other hand, a number of SDIRKN methods available in the literature possess excellent stability properties (cf. [17]), which are much better than those of the LM methods derived from the backward differentiation methods for first-order ODEs. In spite of that, LM methods are still more popular than SDIRKN methods, because of their lower costs on a sequential computer. However, on parallel computers, this situation may change. In this paper, we shall construct DIRKN methods tuned to parallel computers, such that each processor has to compute relatively few stages sequentially. We require that on each processor, these stages are *singly diagonal-implicit*, so that effectively the sequential costs of the parallel DIRKN method (PDIRKN method) are equal to those of an SDIRKN method. In fact, these methods are based on a *fixed* number of iterations of k -stage *indirect* RKN methods of Radau IIA and Gauss–Legendre type (methods of *indirect* type are understood to be methods that are derived by applying an RK method for first-order ODEs to the first-order form of (1.1)). Furthermore, the iteration parameters are chosen such that A-stability is obtained as soon as the order of the corrector is reached. The resulting methods require $k = [(p + 1)/2]$ processors, where p denotes the order and $[\cdot]$ denotes the integer part function. We present a number of A-stable, strongly A-

Table 1
DIRKN methods of order p requiring s^* singly diagonal-implicit, sequential stages on k processors.

Method	p	s^*	k	Main properties	Type
Nørsett [15]	3	$p - 1$	1	A-stable	indirect
Crouzeix [6]	3	$p - 1$	1	Strongly A-stable	indirect
Sharp et al. [17]	3	$p - 1$	1	A-stable, reduced phaselag	direct
Cash [3], Cash and Liem [4]	3	p	1	S-stable	indirect
Burrage [1]	3	$p + 1$	1	A-stable, B-convergent	direct
Nørsett and Thomsen [16]	3	$p + 1$	1	L-stable	indirect
Iserles and Nørsett [12]	4	$p - 2$	2	L-stable	indirect
Nørsett [15]	4	$p - 1$	1	A-stable	indirect
Sharp et al. [17]	4	$p - 1$	1	A-stable, reduced phaselag	direct
Cash [3], Cash and Liem [4]	4	$p + 1$	1	S-stable	indirect
Cooper and Sayfy [5]	5	p	1	A-stable	indirect
Van der Houwen et al. [9]	5	p	3	L-stable	indirect
Cooper and Sayfy [5]	6	$p - 1$	1	A-stable	indirect
Sommeijer [18]	6	$p - 1$	3	A-stable	indirect
Van der Houwen et al. [9]	6	p	3	L-stable	indirect
Van der Houwen et al. [9]	7	$p + 1$	4	L-stable	indirect
	8	p	4	L-stable	indirect

stable and L-stable PDIRKN methods of order p with $s^*(p)$ sequential, singly diagonal-implicit stages, where $s^*(p) = [(p + 1)/2]$ or $s^*(p) = [(p + 1)/2] + 1$.

In order to appreciate these methods, we have summarized in table 1 the characteristics of a number of SDIRKN-type methods of orders $p = 3$ until $p = 8$. We included DIRKN methods of both *direct* and *indirect* type (for a specification of indirect RKN methods we refer to [10] and to the appendix of [14]). Furthermore, we also listed a few indirect parallel DIRKN methods derived from *parallel* DIRK methods. Both the sequential and parallel methods are (effectively) *singly* diagonal-implicit, so that the number of sequential stages s^* refers to the number of *singly* diagonal-implicit stages to be computed on each of the k processors.

By means of numerical experiments we will compare the performance of the methods constructed in this paper with that of a number of the methods listed in table 1.

2. Diagonal-implicit iteration

Our starting point is a fully implicit Runge–Kutta–Nyström (RKN) method of the form

$$\begin{aligned}
 y_{n+1} &= y_n + hy'_n + h^2 \sum_{i=1}^k b_i f(Y_i), \\
 y'_{n+1} &= y'_n + h \sum_{i=1}^k d_i f(Y_i), \\
 Y_i &= y_n + c_i hy'_n + h^2 \sum_{j=1}^k a_{ij} f(Y_j), \quad i = 1, \dots, k,
 \end{aligned}
 \tag{2.1a}$$

where $b = (b_i)$, $c = (c_i)$ and $d = (d_i)$ are k -dimensional vectors, and $A = (a_{ij})$ is a nonsingular k -by- k matrix. This method will be referred to as the corrector method.

We employ a similar iteration technique as applied in [11] which automatically leads to DIRKN methods. Let $Y_i^{(\mu)}$ denote the μ th iterate to Y_i , and define the transformed stage vector quantities X_i and $X_i^{(\mu)}$

$$X_i := Y_i - x_i, \quad X_i^{(\mu)} := Y_i^{(\mu)} - x_i, \quad x_i := y_n + c_i hy'_n, \quad i = 1, \dots, k.
 \tag{2.1b}$$

These new variables are introduced in order to reduce round-off errors (cf. [8, p. 128]). In terms of X_i and x_i , the stage vector equation in (2.1a) reads

$$X_i = h^2 \sum_{j=1}^k a_{ij} f(X_j + x_j), \quad i = 1, \dots, k.
 \tag{2.1'a}$$

For each of these equations, we define the iteration process

$$X_i^{(\mu)} - \delta_i h^2 f(X_i^{(\mu)} + x_i) = h^2 \left(\sum_{j=1}^k a_{ij} f(X_j^{(\mu-1)} + x_j) - \delta_i f(X_i^{(\mu-1)} + x_i) \right), \quad (2.2a)$$

where $i = 1, \dots, k; \mu = 1, \dots, m$, the δ_i are positive iteration parameters, and where the initial approximations $X_i^{(0)}$ are to be provided by means of a predictor formula.

In this paper, we shall try to determine the iteration parameters such that the method is A-stable, strongly A-stable or L-stable as soon as the order of the corrector is reached. As we will see in sections 3 and 4, this can be achieved for a number of correctors derived from classical collocation correctors for first-order equations (indirect collocation correctors, specified in the appendix of the institute report [14]) using one-step predictor formulas of the form

$$X_i^{(0)} = \theta \delta_i h^2 f(X_i^{(0)} + x_i), \quad i = 1, \dots, k, \quad (2.2b)$$

where either $\theta = 0$ or $\theta = 1$. These formulas will be referred to as predictor formulas of type I and II, respectively. The type I predictor $Y_i^{(0)} = x_i = y_n + c_i h y'_n$ is the trivial “last step value” predictor, which does not introduce amplification of stiff error components and does not require any additional computational effort. The type II predictor $Y_i^{(0)} = y_n + c_i h y'_n + \delta_i h^2 f(Y_i^{(0)})$ is implicit and may be considered as a “backward Euler type” predictor. Its strong stability properties may have a stabilizing effect on the whole method (strong damping of stiff components). For example, in the case of Radau correctors, it is possible to achieve L-stability by using type II predictors (see section 4). However, the price to be paid is an additional system of k implicit equations, the computational costs of which may be computed as an additional iteration (notice that the predictor formula of type II can use the same LU decomposition as needed in the subsequent iterations). Both types of predictors are first-order accurate. Within the class of one-step predictors, it is possible to achieve second-order accuracy. For example, we may define the explicit predictor

$$X_i^{(0)} = h^2 \sum_{j=1}^k a_{ij} f(y_n), \quad i = 1, \dots, k.$$

However, such predictor formulas give rise to amplification of stiff components and is not suitable for our purposes. Since we preferred to stay within the class of one-step predictor–corrector methods, we did not investigate multistep predictors.

In [11] it was shown that the formulas for the step values defined in the corrector (2.1) can be presented in the form

$$y_{n+1} = y_n + h y'_n + \sum_{i=1}^k \alpha_i X_i, \quad y'_{n+1} = y'_n + h^{-1} \sum_{i=1}^k \beta_i X_i,$$

body0›where α_i and β_i are the components of the vectors $\mathbf{a} := \mathbf{b}^T A^{-1}$, $\mathbf{\beta} := \mathbf{d}^T A^{-1}$. This suggests defining the step values y_{n+1} and y'_{n+1} corresponding to the iterated method as

$$y_{n+1} = y_n + h y'_n + \sum_{i=1}^k \alpha_i X_i^{(m)}, \quad y'_{n+1} = y'_n + h^{-1} \sum_{i=1}^k \beta_i X_i^{(m)}. \quad (2.3)$$

Since \mathbf{a} and $\mathbf{\beta}$ are not available in the literature, we have listed these vectors for the indirect collocation RKN correctors to be used in our numerical experiments (table 2). For stiffly accurate RKN correctors as Radau IIA, $\mathbf{a} = \mathbf{e}_k^T$.

We remark that for m fixed the method $\{(2.2), (2.3)\}$ fits into the class of DIRKN methods that can be characterized by the Butcher array

$$\begin{array}{c|ccccccc}
 X^{(0)} & \theta D & & & & & & \\
 X^{(1)} & A - D & D & & & & & \\
 X^{(2)} & O & A - D & D & & & & \\
 \cdot & & \cdot & \cdot & \cdot & & & \\
 \cdot & & & \cdot & \cdot & \cdot & & \\
 \cdot & & & & \cdot & \cdot & \cdot & \\
 X^{(m)} & O & O & O & \dots & O & A - D & D \\
 \hline
 & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \dots & \mathbf{0}^T & \mathbf{b}^T A^{-1}(A - D) & \mathbf{b}^T A^{-1} D \\
 & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \dots & \mathbf{0}^T & \mathbf{d}^T A^{-1}(A - D) & \mathbf{d}^T A^{-1} D
 \end{array} \quad (2.4)$$

where D is the diagonal matrix with diagonal entries δ_i . However, in an actual implementation, we shall use the representation $\{(2.2), (2.3)\}$ which avoids f -evaluations in the step point formula.

Table 2
 Vectors \mathbf{a} and $\mathbf{\beta}$ for various indirect collocation RKN correctors.

Correctors	p	\mathbf{a} and $\mathbf{\beta}$
Radau IIA	3	$\mathbf{\beta} = (-9/2, 5/2)^T$
Gauss-Legendre	4	$\mathbf{a} = (-1.732050807569, 1.732050807569)^T$ $\mathbf{\beta} = (-16.392304845413, 4.392304845413)^T$
Radau IIA	5	$\mathbf{\beta} = (5.531972647422, -7.531972647422, 5)^T$
Gauss-Legendre	6	$\mathbf{a} = (5/3, -4/3, 5/3)^T$ $\mathbf{\beta} = (32.909944487358, -16, 7.090055512642)^T$
Radau IIA	7	$\mathbf{\beta} = (-6.923488256444, 6.595237669626, -12.171749413180, 17/2)^T$
Gauss-Legendre	8	$\mathbf{a} = (-1.640705321739, 1.214393969799, -1.214393969799, 1.640705321739)^T$ $\mathbf{\beta} = (-54.681428514064, 26.155201475250, -22.420557316693, 10.946784355507)^T$

Since the k systems that are to be solved in each iteration step of (2.2) can be solved *in parallel* and each has a dimension equal to that of the system of ODEs, the iteration process (2.2) is, on a k -processor computer, of the same computational complexity as an $(m + \theta)$ -stage SDIRKN method on a one-processor computer. Thus, the method $\{(2.2), (2.3)\}$ has only $s^* := m + \theta$ *sequential, singly diagonal-implicit* stages.

THEOREM 2.1

Let p be the order of the k -stage corrector method (2.1) and let $m := [(p + 1)/2]$. Then the method $\{(2.2), (2.3)\}$ is an s -stage DIRKN method of order p with s^* sequential, singly diagonal-implicit stages, where s and s^* are defined by $s = k[(p + 1)/2] + 1 + \theta(k - 1)$ and $s^* = [(p + 1)/2] + \theta$.

Proof

The expressions for s and s^* immediately follow from the Butcher array (2.4). The order of the method is obtained by considering the iteration error of the method. Since (2.2b) defines a first-order predictor formula, we have $X_i^{(0)} - X_i = O(h^2)$. Furthermore, subtracting (2.1'a) and (2.2a) yields

$$\begin{aligned} X_i^{(\mu)} - X_i - \delta_i h^2 (f(X_i^{(\mu)} + x_i) - f(X_i + x_i)) &= -\delta_i h^2 (f(X_i^{(\mu-1)} + x_i) \\ &\quad - f(X_i + x_i)) + h^2 \sum_{j=1}^k a_{ij} (f(X_j^{(\mu-1)} + x_j) - f(X_j + x_j)). \end{aligned}$$

Assuming that f has a bounded Lipschitz constant, it follows that $X_i^{(\mu)} - X_i = O(h^2)(X_i^{(\mu-1)} - X_i)$, so that

$$X_i^{(m)} - X_i = O(h^{2+2m}). \quad (2.5)$$

In order to avoid confusion, let us denote the step values associated with the corrector by u_{n+1} and u'_{n+1} . Subtracting the corrector step values and the iterated step values shows that

$$u_{n+1} - y_{n+1} = \sum_{i=1}^k \alpha_i (X_i - X_i^{(m)}) = O(h^{2+2m}),$$

$$u'_{n+1} - y'_{n+1} = h^{-1} \sum_{i=1}^k \beta_i (X_i - X_i^{(m)}) = O(h^{1+2m}).$$

Let $y(t)$ be the local exact solution. Then the local truncation error is given by

$$\begin{aligned} y(t_{n+1}) - y_{n+1} &= y(t_{n+1}) - u_{n+1} + u_{n+1} - y_{n+1} = O(h^{p+1}) + O(h^{2+2m}), \\ y'(t_{n+1}) - y'_{n+1} &= y'(t_{n+1}) - u'_{n+1} + u'_{n+1} - y'_{n+1} = O(h^{p+1}) + O(h^{1+2m}), \end{aligned} \quad (2.6)$$

where p is the order of the corrector. Thus, we need only $m = [(p + 1)/2]$ iterations to reach the order of the corrector, so that $s^* := m + \theta = [(p + 1)/2] + \theta$. \square

It follows from (2.6) that there are three sources of local errors which together constitute the global error, i.e., the truncation error of the corrector (of order $p + 1$) and the iteration errors corresponding to y_{n+1} and y'_{n+1} (of orders $2m + 2$ and $2m + 1$). In addition to these orders, the order *constants* also play a role. The magnitude of the order constant associated with the corrector is usually rather small. The order constants of the iteration errors decrease with m and are expected to be rather large for small values of m (see also table 3). As the value of m is relatively small, the iteration errors may easily dominate the global error, so that the order of the corrector is not always shown in actual computation. For example, if the iteration error corresponding to y_{n+1} dominates, then the effective order p^* is given by $p^* = 2m + 1 = 2[(p + 1)/2] + 1$. Likewise, if the iteration error corresponding to y'_{n+1} dominates, then $p^* = 2m = 2[(p + 1)/2]$. However, if the integration stepsize h is sufficiently small, then the iteration errors should become negligible, so that the truncation error of the corrector method dominates, and the theoretical order of the corrector should be shown (see table 7).

3. Stability

The linear stability of the method $\{(2.2), (2.3)\}$ is determined by applying it to the scalar test equation $y'' = \lambda y$, where λ runs through the eigenvalues of $\partial f / \partial y$, which are supposed to be negative. Defining the matrix

$$Z(z) := z[I - zD]^{-1}[A - D], \quad P_\theta(z) := z[I - zA]^{-1}[A - \theta D][I - \theta zD]^{-1},$$

$$z := \lambda h^2, \tag{3.1}$$

and the vectors

$$w_{n+1} := \begin{pmatrix} u_{n+1} \\ hu'_{n+1} \end{pmatrix}, \quad v_{n+1} := \begin{pmatrix} y_{n+1} \\ hy'_{n+1} \end{pmatrix}, \tag{3.2}$$

it can be shown (cf. [11]) that the following recursions hold:

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

$$E_m(z) := \begin{pmatrix} b^T A^{-1} Z^m(z) P_\theta(z) e & b^T A^{-1} Z^m(z) P_\theta(z) c \\ d^T A^{-1} Z^m(z) P_\theta(z) e & d^T A^{-1} Z^m(z) P_\theta(z) c \end{pmatrix}, \tag{3.3}$$

$$w_{n+1} = M(z)v_n,$$

$$M(z) := \begin{pmatrix} 1 + zb^T(I - Az)^{-1}e & 1 + zb^T(I - Az)^{-1}c \\ zd^T(I - Az)^{-1}e & 1 + zd^T(I - Az)^{-1}c \end{pmatrix}. \tag{3.4}$$

Hence, by eliminating the corrector values w_{n+1} from (3.3) and (3.4), we find the recursion

$$v_{n+1} = [M(z) - E_m(z)]v_n. \quad (3.5)$$

We shall call the matrix $M(z) - E_m(z)$ the *stability matrix* of the method and its spectral radius the *stability function*, i.e., the function:

$$R_m(z) := \rho([M(z) - E_m(z)]).$$

The method $\{(2.2), (2.3)\}$ is called A-stable if $R_m(z)$ assumes values in $(-1, 1)$ for $z < 0$, strongly A-stable if it is A-stable with $R_m(z)$ bounded away from 1 outside the neighbourhood of the origin, and L-stable if it is A-stable with $R_m(\infty) = 0$.

Putting $m = \lceil (p+1)/2 \rceil$, we obtain p th-order accuracy for any D . We shall exploit the matrix D to obtain p th-order A-stable, strongly A-stable or L-stable methods. However, it turns out that various choices of D generate such highly stable methods. From these methods we selected the methods with smallest truncation error. Recalling that the truncation error of the PDIRKN method will usually be dominated by the iteration error, we are led to consider the iteration error defined by (3.3). Since the nonstiff error components in the iteration error corresponding to small values of $|z|$ are sufficiently damped by the matrix $E_m(z)$ (note that $E_m(z) = O(z^{m+1})$), we shall concentrate on the *stiff* error components. From (3.2), (3.3) and (3.4) it follows that

$$\begin{aligned} w_{n+1} - v_{n+1} &= E_m(z)v_n \\ &= E_m(z)[M(z) - E_m(z)]v_{n-1} \\ &= E_m(z)[M(z) - E_m(z)]^n v_0. \end{aligned}$$

Restricting our considerations to the iteration error associated with y_{n+1} , we deduce that $u_{n+1} - y_{n+1}$ can be bounded by

$$\begin{aligned} \|u_{n+1} - y_{n+1}\| &= \|e_1^T E_m(z)[M(z) - E_m(z)]^n v_0\| \\ &\leq \|e_1^T E_m(z)\| \| [M(z) - E_m(z)]^n \| \|v_0\| \\ &\approx \text{const.} n^{\nu-1} [R_m(z)]^n \|e_1^T E_m(z)\| \|v_0\| \quad \text{as } n \rightarrow \infty, \end{aligned} \quad (3.6)$$

where ν denotes the maximum dimension of the Jordan box corresponding to the maximum-modulus-eigenvalues of the matrix $M(z) - E_m(z)$. This estimate shows that the stiff error components can be suppressed if the stability function $R_m(z)$ is small for large $|z|$ -values. We remark that a similar estimate can be derived for $u'_{n+1} - y'_{n+1}$. The following theorem may be helpful in selecting methods possessing this property:

THEOREM 3.1

Let the predictor be given by (2.2b) and let the corrector (2.1) be obtained from a consistent RK method for first-order equations given by the parameter arrays $\{A^*, b^*, c\}$, then the following assertions hold:

(a) If $\theta = 0$, then

$$R_m(\infty) = \rho \begin{pmatrix} 1 - (\mathbf{b}^*)^T A^* Q_m \mathbf{e} & 1 - (\mathbf{b}^*)^T A^* Q_m \mathbf{c} \\ -(\mathbf{b}^*)^T Q_m \mathbf{e} & 1 - (\mathbf{b}^*)^T Q_m \mathbf{c} \end{pmatrix},$$

$$Q_m := (A^*)^{-2} [I - [I - D^{-1}(A^*)^2]^m].$$

(b) If $\theta = 1$, then $R_m(\infty) = |1 - (\mathbf{b}^*)^T (A^*)^{-1} \mathbf{e}|$ for all m and D , and if the RK method $\{A^*, \mathbf{b}^*, \mathbf{c}\}$ is stiffly accurate, then $R_m(\infty) = 0$ for all m and D .

Proof

If the corrector (2.1) is obtained from an RK method for first-order equations $\{A^*, \mathbf{b}^*, \mathbf{c}\}$, then

$$A = (A^*)^2, \quad \mathbf{b} = (A^*)^T \mathbf{b}^*, \quad \mathbf{c} = A^* \mathbf{e}, \quad \mathbf{d} = \mathbf{b}^*. \quad (3.7)$$

Furthermore, we have that $Z(\infty) = I - D^{-1}A$ and $P_\theta(\infty) = (\theta - 1)I$, where θ is either 0 or 1. Hence,

$$M(\infty) - E_m(\infty) = \begin{pmatrix} 1 - \mathbf{b}^T Q_{m\theta} \mathbf{e} & 1 - \mathbf{b}^T Q_{m\theta} \mathbf{c} \\ -\mathbf{d}^T Q_{m\theta} \mathbf{e} & 1 - \mathbf{d}^T Q_{m\theta} \mathbf{c} \end{pmatrix},$$

$$Q_{m\theta} := A^{-1} [I + (\theta - 1)[I - D^{-1}A]^m]. \quad (3.8)$$

(a) On substitution of $\theta = 0$ and (3.7) into (3.8), part (a) is immediate.

(b) For $\theta = 1$ and using (3.7), we see that (3.8) reduces to

$$M(\infty) - E_m(\infty) = \begin{pmatrix} 1 - (\mathbf{b}^*)^T (A^*)^{-1} \mathbf{e} & 1 - (\mathbf{b}^*)^T \mathbf{e} \\ -(\mathbf{b}^*)^T (A^*)^{-2} \mathbf{e} & 1 - (\mathbf{b}^*)^T (A^*)^{-1} \mathbf{e} \end{pmatrix}. \quad (3.8')$$

Because of the consistency we have that $(\mathbf{b}^*)^T \mathbf{e} = 1$, so that the eigenvalues of $M(\infty) - E_m(\infty)$ are given by $1 - (\mathbf{b}^*)^T (A^*)^{-1} \mathbf{e}$. If the corrector $\{A^*, \mathbf{b}^*, \mathbf{c}\}$ is stiffly accurate, then

$$\mathbf{e}_k^T \mathbf{c} = 1, \quad (\mathbf{b}^*)^T = \mathbf{e}_k^T A^*, \quad (3.9)$$

so that $R_m(\infty)$ vanishes for all m and D .

This theorem shows that for *explicit* predictors of type I ($\theta = 0$), the behaviour of the stability function at infinity depends on D , so that we can exploit the matrix D by selecting methods with the smallest value $R_m(\infty)$. It is interesting to note that we obtained strongly A-stable PDIRKN methods although the corrector is only A-stable (e.g., in the case of Gauss–Legendre correctors listed in table 3).

For *implicit* predictors of type II ($\theta = 1$), the behaviour of the stability function at infinity is completely determined by the corrector, so that D cannot be used for selecting small values of $R_m(\infty)$ in the estimate (3.6). However, (3.6) indicates that the iteration error is also influenced by the magnitude of $\|\mathbf{e}_1^T E_m(z)\|$. Since

Table 3

PDIRKN methods of order p requiring s^* singly diagonal-implicit, sequential stages on k processors.

{Predictor–Corrector}	Iteration parameters δ_i	p	s^*	k	Stability	E_{\max}	E_{∞}
{I - Radau IIA}	(11/200, 107/225)	3	$p - 1$	2	Strongly A-stable	0.35	0.06
{II - Radau IIA}	(1/5, 1/5)	3	p	2	L-stable	0.14	0.00
{I - Gauss-Legendre}	(1/5, 11/20)	4	$p - 2$	2	Strongly A-stable	1.35	1.35
{II - Gauss-Legendre}	(223/10000, 311/1000)	4	$p - 1$	2	A-stable	0.25	0.00
{I - Radau IIA}	(1/40, 1/4, 3/5)	5	$p - 2$	3	Strongly A-stable	0.73	0.16
{II - Radau IIA}	(639/5000, 17/1250, 409/2500)	5	$p - 1$	3	L-stable	0.51	0.00
{I - Gauss-Legendre}	(1/5, 1/2, 3/4)	6	$p - 3$	3	Strongly A-stable	1.44	0.51
{II - Gauss-Legendre}	(1/100, 1/5, 9/20)	6	$p - 2$	3	A-stable	1.32	0.00
{I - Radau IIA}	(1/5, 4/5, 4/5, 19/20)	7	$p - 3$	4	Strongly A-stable	1.43	0.77
{II - Radau IIA}	(9/200, 1/40, 9/40, 91/200)	7	$p - 2$	4	L-stable	1.09	0.00
{I - Gauss-Legendre}	(13/20, 13/20, 3/4, 19/20)	8	$p - 4$	4	Strongly A-stable	1.60	1.60
{II - Gauss-Legendre}	(1/10, 1/5, 3/10, 2/5)	8	$p - 3$	4	A-stable	1.55	0.00

$e_1^T E_m(z)$ vanishes at infinity, we selected methods with a small value of $\|e_1^T E_m(z)\|$ in the whole interval $(-\infty, 0)$.

Finally, we remark that the preceding discussion of the error $u_{n+1} - y_{n+1}$ can also be given for the derivative error $u'_{n+1} - y'_{n+1}$, presumably leading to other matrices D . As a consequence, the PDIRKN methods using the D matrices indicated above aim at problems where our first interest is in an accurate computation of the solution $y(t)$, rather than $y'(t)$.

4. Survey of PDIRKN methods

In table 3, we list the main characteristics of the A-stable, strongly A-stable and L-stable PDIRKN methods we found by means of the approach described in the preceding sections. In this table, E_{\max} denotes the maximum value of $\|e_1^T E_m(z)\|_{\infty}$ in the interval $(-\infty, 0)$ and E_{∞} denotes the value of $\|e_1^T E_m(\infty)\|_{\infty}$. The predictors are of the form (2.2b) with $\theta = 0$ (predictor I) and $\theta = 1$ (predictor II), and the correctors used are the indirect collocation-type RKN methods based on the Gauss–Legendre and Radau IIA RK methods for first-order equations. Specification of the parameters of the resulting methods can be found in the appendix to [14].

Comparing the main characteristics of the methods listed in table 3 with those listed in table 1, we conclude that the computational costs per step of the lower-order methods (order three or four) are comparable, but the higher-order methods in table 3 are much cheaper. On the other hand, the error constant E_{\max} of the itera-

tion error associated with y_{n+1} is relatively large. However, as we have shown in the discussion of theorem 2.1, the order in h of these iteration errors is also larger, which may compensate the large error constants. Hence, we may hope for improved efficiency for the new PDIRKN methods.

5. Numerical experiments

We shall numerically investigate the following aspects of the PDIRKN methods: (i) the stability, in particular, the damping of perturbations of the initial conditions, (ii) the effective order, in relation to the order of the generating corrector, (iii) the predictor, mutual comparison of the explicit and implicit predictor formula, and (iv) the efficiency, in comparison with available sequential SDIRKN methods from the literature.

All problems are taken from the literature and possess exact solutions in closed form. Initial (and boundary) conditions are taken from the exact solution. Most experiments are performed on a 14 digit computer. Only the results reported in table 7 are performed in double precision (28 digits). Furthermore, because of round-off errors, we cannot expect 14 digits or 28 digits accuracy. As a consequence, the tables of results do contain empty spots whenever the corresponding numerical result was in the neighbourhood of the accuracy-limits of the machine and therefore considered as unreliable.

5.1. STABILITY TEST

We first test the stability properties of the various PDIRKN methods by integrating a nonautonomous problem with varying stiffness:

$$\begin{aligned}
 y''(t) &= \begin{pmatrix} -2\alpha(t) + 1 & -\alpha(t) + 1 \\ 2(\alpha(t) - 1) & \alpha(t) - 2 \end{pmatrix} y(t), \\
 y(0) &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad y'(0) = \begin{pmatrix} -1 \\ 2 \end{pmatrix}, \quad 0 \leq t \leq T, \\
 \alpha(t) &= \sqrt{1 + t^3} + \frac{1}{\sqrt{1 + t^3}}.
 \end{aligned} \tag{5.1}$$

The Jacobian matrix of the system has the eigenvalues -1 and $-\alpha(t)$, so that the spectral radius, and therefore the stiffness, increases with t . We compared the numerical solution of (5.1) with the numerical solution obtained by perturbing the initial conditions, i.e., instead of the initial conditions $y(0)$ and $y'(0)$ we used the initial conditions $y(0) + \epsilon e$ and $y'(0) + \epsilon e$. Denoting the numerical solutions by y_n and y_n^* , we may expect from any stable method that $\|y_n - y_n^*\|$ does not increase with n . For various PDIRKN methods, table 4 lists the values

Table 4

Values of the amplification factor C_n for problem (5.1) with $T = 4000$, $n = 4000$ and with $T = 6000$, $n = 6000$ for various Predictor–Corrector pairs.

Type I Methods	p	C_{4000}	C_{6000}	Type II Methods	p	C_{4000}	C_{6000}
I-Radau IIA	3	0.30E-23	0.84E-36	II-Radau IIA	3	0.43E-11	0.35E-17
I-Gauss–Legendre	4	0.44E-12	0.22E-18	II-Gauss–Legendre	4	0.44E-01	0.38E-01
I-Radau IIA	5	0.53E-29	0.48E-44	II-Radau IIA	5	0.64E-01	0.67E-02
I-Gauss–Legendre	6	0.17E-23	0.12E-35	II-Gauss–Legendre	6	0.34E-13	0.93E-14
I-Radau IIA	7	0.88E-21	0.63E-31	II-Radau IIA	7	0.40E-09	0.61E-15
I-Gauss–Legendre	8	0.53E-25	0.13E-38	II-Gauss–Legendre	8	0.44E-12	0.51E-13

$$C_n := \frac{\|y_n - y_n^*\|}{\|y_0 - y_0^*\|} \\ = \|y_n - y_n^*\|/\epsilon \quad \text{for } n = 4000 \text{ and } n = 6000.$$

The methods are specified by the generating Predictor–Corrector pair where the predictor is indicated by its type. It turned out that C_n is almost independent of ϵ for $\epsilon \leq 1/10$. The results in table 4 demonstrate the strong damping of the initial perturbation by all PDIRKN methods.

We remark that with respect to the scalar test equation (see also (3.6)), the estimate

$$\|y_n - y_n^*\| = \|e_1^T (M_m(z) - E_m(z))^n (y_0 - y_0^*)\| \\ \approx \text{const.} n^{\nu-1} [R(z)]^n \epsilon$$

shows that C_n depends on the stability behaviour of the PDIRKN method for a particular value of z , and it is expected that for an A-stable PDIRKN method and a given problem with specified stepsize, C_n will decrease as n increases. This behaviour is demonstrated by the results listed in table 4.

Another observation is that for this linear problem, the explicit predictors give a better damping than the implicit predictors. The damping effect turns out to be strongly problem-dependent as is shown by the following example:

$$y''(t) = -1000(y(t) - \cos(t))^3 - \cos(t), \quad y(0) = 1, y'(0) = 0, 0 \leq t \leq T. \quad (5.1')$$

Applying the same test strategy as before, the results listed in table 5 show that the implicit and explicit predictors give rise to a similar damping effect for this problem. Moreover, the damping is much weaker when compared to the previous example.

5.2. EFFECTIVE ORDER AND EFFICIENCY OF THE EXPLICIT AND IMPLICIT PREDICTOR

In this section, we show that the effective order of the PDIRKN methods may exceed the order of the corrector. In addition, we compare the efficiency of the explicit and implicit predictor. In all experiments the accuracy is given by means of the number of minimal correct digits (NCD) defined by $\text{NCD}(h) = -\log(\| \text{global}$

Table 5
Values of the amplification factor C_n for problem (5.1') with $T = 1000, n = 10000$ for various Predictor–Corrector pairs.

Type I Methods	p	s^*	k	C_n	Type II Methods	p	s^*	k	C_n
I-Radau IIA	3	2	2	0.36E+00	II-Radau IIA	3	3	2	0.63E+00
I-Gauss-Legendre	4	2	2	0.44E+00	II-Gauss-Legendre	4	3	2	0.12E+00
I-Radau IIA	5	3	3	0.82E+00	II-Radau IIA	5	4	3	0.58E+00
I-Gauss-Legendre	6	3	3	0.89E+00	II-Gauss-Legendre	6	4	3	0.10E+01
I-Radau IIA	7	4	4	0.10E+01	II-Radau IIA	7	5	4	0.64E+00
I-Gauss-Legendre	8	4	4	0.10E+01	II-Gauss-Legendre	8	5	4	0.48E+00

error at the endpoint of the integration interval $\| \cdot \|_\infty$), and the computational effort is measured by the number of sequential stages per unit interval. The (fixed) step-size is chosen such that the number of sequential stages per unit interval (approximately) equals a prescribed number M . To be more precise, let N_{steps} denote the total number of integration steps for the integration interval $[t_0, T]$, then $M = N_{steps}s^*/(T - t_0)$, which leads us to

$$N_{steps} = \left\lceil \frac{M(T - t_0)}{s^*} + 0.5 \right\rceil, \quad h = \frac{T - t_0}{N_{steps}},$$

where $[\cdot]$ denotes the integer part function (the effect of the $[\cdot]$ operation causes that the actual number of sequential stages may be slightly different from the prescribed number M).

Table 6 lists results for the linear Kramarz problem (see [13])

$$y''(t) = \begin{pmatrix} 2498 & 4998 \\ -2499 & -4999 \end{pmatrix} y(t), \quad 0 \leq t \leq 100, \quad (5.2)$$

Table 6
Effective order p^* and values of NCD and M for problem (5.2).

Predictor–Corrector	p	s^*	k	$M = 25$	$M = 50$	$M = 100$	$M = 200$	p^*
I-Radau IIA	3	2	2	2.8	3.8	4.7	5.6	3
II-Radau IIA	3	3	2	2.4	3.3	4.2	5.1	3
I-Gauss-Legendre	4	2	2	3.3	4.5	5.7	6.9	4
II-Gauss-Legendre	4	3	2	4.0	5.4	6.7	8.0	4
I-Radau IIA	5	3	3	4.2	6.0	7.8	9.6	6
II-Radau IIA	5	4	3	5.1	6.8	8.5	10.0	5
I-Gauss-Legendre	6	3	3	3.9	5.8	7.6	9.4	6
II-Gauss-Legendre	6	4	3	4.6	6.7	8.8	11.0	7
I-Radau IIA	7	4	4	4.5	6.9	9.3	12.0	8
II-Radau IIA	7	5	4	5.4	8.1	10.8		9
I-Gauss-Legendre	8	4	4	4.4	6.8	9.2	12.8	8
II-Gauss-Legendre	8	5	4	5.2	7.7	10.1		8

Table 7

Effective order p^* and values of NCD and M for problem (5.2) obtained by some specified PDIRKN methods with small stepsizes.

Predictor–Corrector	p	s^*	k	$M = 800$	$M = 1600$	$M = 3200$	$M = 6400$	p^*
I - Radau IIA	5	3	3	13.1	14.8	16.5	18.1	5.3
II - Gauss–Legendre	6	4	3	15.3	17.3	19.2	21.0	6
I - Radau IIA	7	4	4	16.5	18.7	20.9		7.3
II - Radau IIA	7	5	4	18.5	20.7	22.7		6.7

with exact solution $y(t) = (2 \cos(t), -\cos(t))^T$. These results show that for some higher-order methods (indicated in bold face), the measured effective order p^* is greater than p (see the discussion of theorem 2.1). In order to show that this “higher-order behaviour” is caused by a dominance of the iteration error, we applied these “higher-order” PDIRKN methods again to the Kramarz problem (5.2), but now with very small stepsizes. Using a high-precision computer (28 digits), we obtained the results listed in table 7, showing that the corrector-order is more or less retained.

Finally, we observe that usually the implicit predictor (type II) produces better results, in spite of the additional implicit stage. Therefore, in the following, we shall confine our considerations to the type II predictor.

5.3. EFFICIENCY TESTS

In this section, we compare the efficiency of the PDIRKN method with methods from the literature. We selected the following methods from table 1:

Table 8

Values of NCD and M for problem (5.2).

Methods	p	s^*	k	$M = 25$	$M = 50$	$M = 100$	$M = 200$
Nørsett ₃	3	2	1	2.1	3.0	3.9	4.8
SFB ₃	3	2	1	1.8	2.7	3.6	4.5
B ₃	3	4	1	1.2	2.1	3.0	3.9
II - Radau IIA	3	3	2	2.4	3.3	4.2	5.1
Nørsett ₄	4	3	1	2.8	3.8	4.9	6.1
SFB ₄	4	3	1	3.2	4.5	5.7	6.9
II - Gauss–Legendre	4	3	2	4.0	5.4	6.7	8.0
CS ₅	5	5	1	4.1	5.6	7.1	8.6
II - Radau IIA	5	4	3	5.1	6.8	8.5	10.0
CS ₆	6	5	1	5.5	7.0	8.4	9.0
II - Gauss–Legendre	6	4	3	4.6	6.7	8.8	11.0
II - Radau IIA	7	5	4	5.4	8.1	10.8	
II - Gauss–Legendre	8	5	4	5.2	7.7	10.1	

- Nørsett₃ third-order method of Nørsett;
- Nørsett₄ fourth-order method of Nørsett;
- SFB₃ third-order method of Sharp et al.;
- SFB₄ fourth-order method of Sharp et al.;
- B₃ third-order method of Burrage;
- CS₅ fifth-order method of Cooper and Sayfy;
- CS₆ sixth-order method of Cooper and Sayfy.

5.3.1. Linear Kramarz problem

Table 8 presents results for these sequential methods and for our PDIRKN methods when applied to the Kramarz problem (5.2). In most cases, the PDIRKN methods are by far the most accurate ones. Notice that the CS₆ method does not show its order 6 in the high accuracy range. This is caused by an insufficient accuracy of the method parameters. As a consequence, the CS₆ method may well be competitive with the sixth-order PDIRKN method.

5.3.2. Linear Strehmel–Weiner problem

In [19] we find the following linear, stiff problem:

$$y''(t) = \begin{pmatrix} -20.2 & 0 & -9.6 \\ 7989.6 & -10000 & -6004.2 \\ -9.6 & 0 & -5.8 \end{pmatrix} y(t) + \begin{pmatrix} 150 \cos(10t) \\ 75 \cos(10t) \\ 75 \cos(10t) \end{pmatrix},$$

$$0 \leq t \leq 100, \tag{5.3}$$

Table 9
Values of NCD and M for problem (5.3).

Methods	p	s^*	k	$M = 100$	$M = 200$	$M = 400$	$M = 800$
Nørsett ₃	3	2	1	1.1	2.0	2.9	3.8
SFB ₃	3	2	1	0.8	1.7	2.6	3.5
B ₃	3	4	1	0.3	1.1	2.0	2.9
II - Radau IIA	3	3	2	1.4	2.3	3.2	4.1
Nørsett ₄	4	3	1	1.2	2.5	3.8	5.0
SFB ₄	4	3	1	2.3	3.4	4.7	5.9
II - Gauss–Legendre	4	3	2	3.1	4.9	6.7	7.3
CS ₅	5	5	1	3.0	4.5	5.9	7.4
II - Radau IIA	5	4	3	4.9	6.6	7.6	9.0
CS ₆	6	5	1	3.6	5.5	7.5	8.2
II - Gauss–Legendre	6	4	3	3.2	5.3	7.4	9.4
II - Radau IIA	7	5	4	3.9	6.6	9.4	10.0
II - Gauss–Legendre	8	5	4	4.4	6.5	8.8	10.0

with exact solution

$$y(t) = \begin{pmatrix} \cos(t) + 2 \cos(5t) - 2 \cos(10t) \\ 2 \cos(t) + \cos(5t) - \cos(10t) \\ -2 \cos(t) + \cos(5t) - \cos(10t) \end{pmatrix}.$$

Unlike the Kramarz problem, this problem has slowly and rapidly oscillating solution components (nonstiff and stiff solution components) which are appearing with comparable weights. This implies a severe test for the PDIRKN methods because of the strong damping, and therefore inaccurate approximation, of the stiff solution components. In spite of that, they are generally superior to the sequential methods. Again, taking into account the inaccurate method parameters of CS₆, we see from the results listed in table 9 that this method is competitive.

5.3.3. Nonlinear Strehmel–Weiner problem

In [19] we also find a nonlinear, stiff problem:

$$\begin{aligned} y_1''(t) &= (y_1(t) - y_2(t))^3 + 6368y_1(t) - 6384y_2(t) + 42 \cos(10t), \\ y_2''(t) &= -(y_1(t) - y_2(t))^3 + 12768y_1(t) - 12784y_2(t) + 42 \cos(10t), \\ 0 &\leq t \leq 10, \end{aligned} \tag{5.4}$$

with exact solution $y_1(t) = y_2(t) = \cos(4t) - \cos(10t)/2$. Table 10 demonstrates that the PDIRKN methods similarly compare with the sequential methods as for the linear Kramarz and Strehmel–Weiner problems.

Table 10
Values of NCD and M for problem (5.4).

Methods	p	s^*	k	$M = 100$	$M = 200$	$M = 400$	$M = 800$
Nørsett ₃	3	2	1	2.9	3.9	4.8	5.7
SFB ₃	3	2	1	2.7	3.6	4.5	5.4
B ₃	3	4	1	2.3	3.6	5.2	6.2
II - Radau IIA	3	3	2	3.3	4.1	5.1	6.0
Nørsett ₄	4	3	1	3.0	4.2	5.3	6.5
SFB ₄	4	3	1	3.7	4.9	6.1	7.3
II - Gauss–Legendre	4	3	2	4.8	6.1	7.4	8.7
CS ₅	5	5	1	4.9	6.4	7.9	9.4
II - Radau IIA	5	4	3	5.8	7.6	9.4	11.1
CS ₆	6	5	1	5.9	7.6	9.2	9.9
II - Gauss–Legendre	6	4	3	5.5	7.6	9.7	11.8
II - Radau IIA	7	5	4	6.4	9.0	11.6	
II - Gauss–Legendre	8	5	4	5.8	8.2	10.6	

5.3.4. Fehlberg problem

An often used test problem is the orbit equation (cf. [7])

$$\begin{aligned}
 y_1''(t) &= -4t^2y_1(t) - \frac{2y_2(t)}{\sqrt{y_1^2(t) + y_2^2(t)}}, \\
 y_2''(t) &= -4t^2y_2(t) + \frac{2y_1(t)}{\sqrt{y_1^2(t) + y_2^2(t)}},
 \end{aligned}
 \quad \sqrt{\pi/2} \leq t \leq 3\pi, \quad (5.5)$$

with the exact solution $y_1(t) = \cos(t^2), y_2(t) = \sin(t^2)$. Results are presented in table 11. Usually this type of equations has to be solved with stringent accuracy demands. From table 11 we conclude that the high-order PDIRKN methods are more efficient in the high accuracy range.

5.3.5. Semi-discrete partial differential equation

Consider the following initial-boundary-value problem (see [11]):

$$\frac{\partial^2 u}{\partial t^2} = \frac{4\pi^2 u^2}{1 + 2x - 2x^2} \frac{\partial^2 u}{\partial x^2} + 4\pi^2 u [4 \cos^2(2\pi t) - 1], \quad 0 \leq t \leq 1, \quad 0 \leq x \leq 1, \quad (5.6)$$

with Dirichlet boundary conditions and exact solution $u = (1 + 2x - 2x^2) \cos(2\pi t)$. By using second-order symmetric spatial discretization on a uniform grid with mesh $\Delta x = 1/20$ we obtain a set of 19 ODEs. Table 12 shows that the

Table 11
Values of NCD and M for problem (5.5).

Methods	p	s^*	k	$M = 98$	$M = 196$	$M = 392$	$M = 783$
Nørsett ₃	3	2	1	0.9	1.8	2.7	3.6
SFB ₃	3	2	1	0.6	1.5	2.4	3.3
B ₃	3	4	1	0.2	0.9	1.9	2.7
II - Radau IIA	3	3	2	0.9	2.0	2.9	4.0
Nørsett ₄	4	3	1	0.7	1.5	2.7	4.0
SFB ₄	4	3	1	1.2	2.4	3.6	4.8
II - Gauss-Legendre	4	3	2	1.7	3.2	4.5	5.9
CS ₅	5	5	1	1.7	3.1	4.7	6.2
II - Radau IIA	5	4	3	2.1	3.8	5.6	7.3
CS ₆	6	5	1	1.9	3.5	5.3	7.1
II - Gauss-Legendre	6	4	3	1.2	3.1	5.1	7.2
II - Radau IIA	7	5	4	1.1	3.3	5.9	8.5
II - Gauss-Legendre	8	5	4	1.1	3.2	5.6	8.0

Table 12
Values of NCD and M for problem (5.6).

Methods	p	s^*	k	$M = 200$	$M = 400$	$M = 800$	$M = 1600$
Nørsett ₃	3	2	1	3.5	4.3	5.1	5.9
SFB ₃	3	2	1	3.6	4.5	5.4	6.3
B ₃	3	4	1	*	4.3	5.4	6.4
II - Radau IIA	3	3	2	3.7	5.1	6.0	6.8
Nørsett ₄	4	3	1	3.4	4.2	5.2	5.9
SFB ₄	4	3	1	5.5	6.4	7.6	8.8
II - Gauss-Legendre	4	3	2	5.0	6.3	7.8	9.2
CS ₅	5	5	1	4.0	5.3	6.6	7.7
II - Radau IIA	5	4	3	4.2	5.2	6.3	7.7
CS ₆	6	5	1	3.1	4.4	5.5	6.9
II - Gauss-Legendre	6	4	3	3.8	4.7	6.2	8.1
II - Radau IIA	7	5	4	*	4.7	6.0	8.5
II - Gauss-Legendre	8	5	4	3.6	4.4	5.5	7.0

PDIRKN methods are at least competitive and often more efficient than the sequential methods of the same order.

6. Concluding remarks

In this paper, we have shown that diagonally implicit iteration of fully implicit, p th-order RKN correctors leads to parallel DIRKN methods of order p with relatively few sequential stages. For Radau IIA and Gauss-Legendre correctors, the iteration parameters are determined in such a way that the methods are A-stable, strongly A-stable or L-stable. Numerical experiments clearly demonstrate the superiority of the parallel methods over most of the sequential SDIRKN methods available in the literature.

Acknowledgements

I like to thank Prof. Dr. P.J. van der Houwen and Dr. B.P. Sommeijer for their help during the preparation of this paper. I am also grateful to the referees for their useful comments.

References

- [1] K. Burrage, A study of order reduction for semi-linear problems, Report, University of Auckland (1990).
- [2] J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations, Runge–Kutta and General Linear Methods* (Wiley, New York, 1987).
- [3] J.R. Cash, Diagonally implicit Runge–Kutta formulae with error estimates, *J. Inst. Math. Appl.* 24 (1979) 293–301.
- [4] J.R. Cash and C.B. Liem, On the design of a variable order, variable step diagonally implicit Runge–Kutta algorithm, *J. Inst. Math. Appl.* 26 (1980) 87–91.
- [5] G.J. Cooper and A. Sayfy, Semiexplicit A-stable Runge–Kutta methods, *Math. Comp.* 33 (1979) 146, 541–556.
- [6] M. Crouzeix, Sur l’approximation des équations différentielles opérationnelles linéaires par des méthodes de Runge–Kutta, Ph. D. Thesis, Université de Paris, France (1975).
- [7] E. Fehlberg, Klassische Runge–Kutta–Nyström Formeln mit Schrittweiten-Kontrolle für Differentialgleichungen $x'' = f(t, x)$, *Computing* 10 (1972) 305–315.
- [8] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*, Springer Series in Comp. Math., vol. 14 (Springer, Berlin, 1991).
- [9] P.J. van der Houwen, B.P. Sommeijer and W. Couzy, Embedded diagonally implicit Runge–Kutta algorithms on parallel computers, *Math. Comp.* 58 (1992) 197, 135–159.
- [10] 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.
- [11] P.J. van der Houwen, B.P. Sommeijer and Nguyen huu Cong, Parallel diagonally implicit Runge–Kutta–Nyström methods, *J. Appl. Numer. Math.* 9 (1992) 111–131.
- [12] A. Iserles and S.P. Nørsett, On the theory of parallel Runge–Kutta methods, *IMA J. Numer. Anal.* 10 (1990) 463–488.
- [13] L. Kramarz, Stability of collocation methods for the numerical solution of $y'' = f(x, y)$, *BIT* 20 (1980) 215–222.
- [14] Nguyen huu Cong, A-stable diagonally implicit Runge–Kutta–Nyström methods for parallel computers, Report NM-R9208, Centre for Mathematics and Computer Science, Amsterdam (1992).
- [15] S.P. Nørsett, Semi-explicit Runge–Kutta methods, Report Mathematics and Computation No. 6/74, Dept. of Mathematics, University of Trondheim, Norway (1974).
- [16] S.P. Nørsett and P.G. Thomsen, Embedded SDIRK-methods of basic order three, *BIT* 24 (1984) 634–646.
- [17] P.W. Sharp, J.H. Fine and K. Burrage, Two-stage and three-stage diagonally implicit Runge–Kutta–Nyström methods of orders three and four, *IMA J. Numer. Anal.* 10 (1990) 489–504.
- [18] B.P. Sommeijer, Parallelism in the numerical integration of initial value problems, Thesis, University of Amsterdam (1992).
- [19] K. Strehmel and R. Weiner, Nichtlineare Stabilität und Phasenuntersuchung adaptiver Nyström–Runge–Kutta Methoden, *Computing* 35 (1985) 325–344.