1991

P.J. van der Houwen, B.P. Sommeijer, Nguyen huu Cong

Parallel diagonally implicit Runge-Kutta-Nyström methods

Department of Numerical Mathematics Report NM-R9103 January

CWI, nationaal instituut voor onderzoek op het gebied van wiskunde en informatica

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

Parallel Diagonally Implicit Runge-Kutta-Nyström Methods

P.J. van der Houwen & B.P. Sommeijer

Centre for Mathematics and Computer Science
Post box 4079, 1009 AB Amsterdam, The Netherlands

Nguyen huu Cong
Faculty of Mathematics, Mechanics and Informatics
University of Hanoi, Thuong dinh, Dong Da, Hanoi, Vietnam

In this paper, we study diagonally implicit iteration methods for solving implicit Runge-Kutta-Nyström methods on parallel computers. These iteration methods are such that in each step, the iterated method belongs to the class of diagonally implicit Runge-Kutta-Nyström methods (DIRKN methods). The number of stages of this DIRKN method depends on the number of iterations and may vary from step to step. Since a large number of these stages can be computed in parallel, and since the total number of stages can be kept small by a suitable choice of the parameters in the iteration process, the resulting variable-stage DIRKN methods are rather efficient on parallel computers. By using implicit Runge-Kutta-Nyström methods with high stage order, the phenomenon of order reduction exhibited in many problems with large Lipschitz constants does not deteriorate the accuracy of these variable-stage DIRKN methods.

1980 Mathematics Subject Classification: 65M10, 65M20
1982 CR Categories: 5.17
Key Words and Phrases: Diagonally implicit Runge-Kutta-Nyström methods, Predictor-corrector methods, parallelism.

Note: These investigations were supported by the University of Amsterdam who provided the third author with a research grant for spending a total of two years in Amsterdam.

1. Introduction

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

$$(1.1) \qquad y''(t) = f(y(t)), \quad y(t_0) = y_0, \quad y'(t_0) = y'_0, \quad y : \mathbb{R} \to \mathbb{R}^d, \quad f : \mathbb{R}^d \to \mathbb{R}^d, \quad t_0 \le t \le t_{end}.$$

Important test examples from this class of problems originate from structural mechanics. Such problems possess periodic solution components with frequencies ranging from small to large of which the lower harmonics are of interest and the higher harmonics are not, that is, only the solution components corresponding to eigenvalues of the Jacobian matrix $\partial f/\partial y$ close to the origin are of interest. In such cases, the ideal method would be a method without dissipation of the lower harmonics (i.e., nonempty periodicity interval), high order of dispersion, and damping of the higher harmonics. The presence of unwanted high harmonics (a form of stiffness) may reduce the step point order considerably. In many stiff problems, it is the stage order that determines the accuracy, rather than the step point order (cf. [2]). In order to avoid the effect of order reduction we need methods that have, in addition to a high step point order and the property of A-stability, a high stage order.

In this paper, we consider integration methods based on iteration of fully implicit Runge-Kutta-Nyström (RKN) methods of collocation type. Such RKN methods possess the largest possible stage order, so that we automatically achieve high stage orders if the RKN method is solved sufficiently accurate. Furthermore, after only a few iterations, the step point order of the iterated method equals that of the underlying implicit RKN method. Since there are A-stable RKN methods available of arbitrarily high step point order, the iterated methods possess the requirements stated above. For an extensive set of suitable RKN methods with high stage orders we refer to [9].

We shall investigate *diagonal-implicit* iteration methods which, after a finite number of iterations, belong to the class of diagonally implicit Runge-Kutta-Nyström methods (DIRKN methods). We remark that the step point order p of these DIRKN methods can be made arbitrarily high by iterating an RKN method with step point order p, where p is sufficiently large. Hence, the restriction $p \le 4$ which applies to the DIRKN methods available in the literature is easily relaxed. Adopting the terminology used for iterating implicit linear multistep methods, we shall call the underlying implicit RKN method the *corrector* and the method used for starting the iteration the *predictor*. The iteration process will be called *predictor-corrector* (PC) method.

The number of stages of this PC method increases with the number of iterations and may vary from step to step depending on the convergence criterion. Because of the nature of diagonal-implicit PC methods, a large number of the

stages of the resulting variable-stage DIRKN method can be computed in parallel, so that the number of stages that have to be computed sequentially is substantially reduced when implemented on multi-processor computers. A second advantage is that on each processor, the method behaves as a *singly-implicit* DIRKN method (SDIRKN method), that is, only one LU decomposition per processor is required. Thirdly, we shall reduce the number of iterations per step by a suitable choice of the parameters in the iteration process. In this paper, our approach of choosing the iteration parameters is based on the minimization of the spectral radius of the stage vector iteration matrix. For a number of RKN correctors generated by collocation-based RK methods, we have calculated the iteration parameters with this minimizing property. However, fast convergence of the PC iteration is useless if the overall stability is insufficient. Therefore, from the various PC methods, we selected those methods that are A-stable for a minimal number of iterations per step. Finally, the use of collocation-based 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.

A number of numerical examples clearly demonstrate that the high-order parallel SDIRKN methods proposed in this paper are by far superior to the sequential SDIRKN methods from the literature.

1.1. RKN methods

We consider RKN correctors of the form

$$
\begin{aligned}
y_{n+1} &= y_n + hy'_n + b_0h^2f(y_n) + h^2\sum_{i=1}^{k} b_i\, f(Y_i),\\
y'_{n+1} &= y'_n + d_0hf(y_n) + h\sum_{i=1}^{k} d_i\, f(Y_i), \qquad (1.2)\\
Y_i &= y_n + c_ihy'_n + a_ih^2f(y_n) + h^2\sum_{j=1}^{k} a_{ij}\, f(Y_j), \quad i = 1, \ldots, k,
\end{aligned}
$$

or using the Butcher array notation (cf., e.g., [5]),

$$
\begin{array}{c|cc}
0 & 0 & \mathbf{0}^T \\
\mathbf{c} & \mathbf{a} & A \\
\hline
 & b_0 & \mathbf{b}^T \\
 & d_0 & \mathbf{d}^T
\end{array}, \qquad (1.3)
$$

where $\mathbf{b}=(b_i)$, $\mathbf{c}=(c_i)$ and $\mathbf{d}=(d_i)$ are k-dimensional vectors, $A=(a_{ij})$ is a k-by-k matrix and $\mathbf{0}$ is a k-dimensional vector with zero entries. We always assume that the matrix A is nonsingular. Scheme (1.2) presents an (s=k+1)-stage RKN method requiring k *implicit* stages and one explicit stage. In the case where $\mathbf{a}$, b_0 and d_0 vanish, the explicit stage is not needed and (1.2) reduces 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, we refer to the literature (e.g., [4], [9]).

It will be assumed that the eigenvalues of the Jacobian matrix $\partial f/\partial y$ in (1.1) are negative. This means that the integration step should satisfy the stability condition

$$
h^2 \le \frac{\beta_{stab}}{\rho(\partial f/\partial y)}, \qquad (1.4)
$$

where $\rho(\partial f/\partial y)$ is the spectral radius of the Jacobian matrix $\partial f/\partial y$ and β_{stab} denotes the stability boundary of the RKN method. Thus, if we have a stiff problem where $\rho(\partial f/\partial y)$ is extremely large, then we should apply an A-stable RKN method, i.e., $\beta_{stab}=\infty$ (we remark that such methods were termed R-stable in [17]). Unfortunately, the RKN methods with maximal stage order possess finite stability boundaries (cf. [10], [9]). In this connection, we remark that for certain classes of problems it is possible to use non-A-stable RKN methods for stiff problems by preconditioning the equation in (1.1). Then, instead of integrating (1.1), we integrate the equation [9]

$$
\begin{aligned}
&y''(t) = g(y(t)), \quad g(y) := T^{-1}(J_n)\, f(y), \quad y(t_n) = y_n, \quad y'(t_n) = y'_n, \quad t_n \le t \le t_n + h,\\
&T(x) := 1 + \varepsilon(-x)^{\sigma}, \quad \sigma := \lfloor (p+1)/2 \rfloor, \quad J_n := \frac{\partial f(y_n)}{\partial y},
\end{aligned} \qquad (1.5)
$$

where p is the order of the RKN method and ε is a small parameter. The advantage is that, irrespective the size of the (negative) eigenvalue interval of $\partial f/\partial y$, the eigenvalues of $\partial g/\partial y$ are in a finite interval $[-\rho^*,0]$, with

$$(1.6) \qquad \rho^* := \frac{\sigma - 1}{\sigma\,[(\sigma - 1)\varepsilon]^{1/\sigma}} .$$

Hence, for the preconditioned equation (1.5) the stability condition (1.4) can be written as

$$(1.7) \qquad \varepsilon \geq \frac{1}{\sigma - 1}\left(\frac{\sigma - 1}{\sigma\beta_{stab}}\right)^{\sigma} h^{2\sigma},$$

where h denotes the step one wants to use. This condition shows that ε can be chosen of order $O(h^{2\sigma})$, so that (1.5) can be interpreted as a perturbed problem in which the perturbation is of order 2σ in h, that is, at least of order p.

In this paper, we shall concentrate on the iteration of A-stable RKN correctors. However, we shall present all formulas for equation (1.5), so that the use of non-A-stable RKN correctors is included in the subsequent analysis (notice that by setting ε=0, we recover the original equation (1.1)). In a future paper, we intend to study the performance of non-A-stable RKN correctors with increased stage order.

1.2. Sequential SDIRKN methods from the literature

On *parallel* computers the computational complexity of the methods constructed in this paper is comparable with that of SDIRKN methods on *sequential* computers. In order to facilitate a comparison of our parallel methods with already available sequential SDIRKN methods, we shall list a few of such SDIRKN methods from the literature.

Firstly, we remark that SDIRKN methods can be generated starting from SDIRK methods for first-order ODEs. Writing (1.1) in first-order form and application of an SDIRK method straightforwardly yields an SDIRKN method. Such methods will be called *indirect* SDIRKN methods. In particular, we mention the two-stage and three-stage A-stable SDIRKN methods of orders p=3 and p=4, respectively, based on the SDIRK methods of Nørsett [13]. These indirect methods will be denoted by Nørsett_2 and Nørsett_3, the subscript referring to the number of implicit stages per step. Since these methods do not possess an explicit stage, they have vanishing $\mathbf{a}$, b_0 and d_0. Therefore, their Butcher arrays will be presented in the condensed form

$$(1.3') \qquad \begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \\ & \mathbf{d}^T \end{array} .$$

Using this format, the indirect Nørsett methods are now defined by

$$\begin{array}{c|cc} \lambda & \lambda^2 & 0 \\ 1-\lambda & 2\lambda(1-2\lambda) & \lambda^2 \\ \hline & (1-\lambda)/2 & \lambda/2 \\ & 1/2 & 1/2 \end{array} , \qquad \begin{array}{c|ccc} \xi & \xi^2 & 0 & 0 \\ 1/2 & \xi(1-2\xi) & \xi^2 & 0 \\ 1-\xi & 8\xi^2 - 3\xi + 1/2 & 2\xi(1-4\xi) & \xi^2 \\ \hline & 5\xi\eta - \xi - \eta + 1/2 & -6\xi\eta + \xi + \eta & \xi\eta \\ & \eta & 1-2\eta & \eta \end{array} ,$$

where

$$\lambda := \frac{3 + \sqrt{3}}{6}, \quad \xi := \frac{3 + 2\sqrt{3}\cos(\pi/18)}{6}, \quad \eta := \frac{1}{6(1-2\xi)^2} .$$

Furthermore, we mention the indirect SDIRKN method generated by the third-order A-stable SDIRK method of Burrage [1]. This four-stage method has the special property that its order of B-convergence equals 3 for semi-linear problems. In the format (1.3'), its Butcher array reads

$$\begin{array}{c|cccc} 0.7886751346 & 0.6220084679 & & & \\ 3.1742957030 & 3.7629592451 & 0.6220084679 & & \\ -0.0195951646 & -1.2749253740 & 0.0 & 0.6220084679 & \\ 1.0830184350 & 0.7564996127 & -0.0739506877 & -0.2182664410 & 0.6220084679 \\ \hline & -0.1353633836 & -0.0473517944 & 0.2862835400 & 0.3964316380 \\ & 0.0763188000 & -0.0301592919 & 0.4511853166 & 0.5026551753 \end{array} .$$

This method will be denoted by B_4.

In addition to the aforementioned indirect SDIRKN methods, we mention two *direct* SDIRKN methods. By 'direct' we mean that they do not originate from an SDIRK method for first-order ODEs, but are constructed directly for the special second-order equation (1.1). In [17], Sharp, Fine and Burrage proposed two-stage and three-stage A-stable *direct* SDIRKN methods. In the form (1.3'), their Butcher tableaux are given by

$$\begin{array}{c|cc} \frac{17}{14} & \frac{289}{392} & \\ \frac{23}{60} & -\frac{234179}{352800} & \frac{289}{392} \\ \hline & -\frac{21}{698} & \frac{185}{349} \\ & \frac{49}{349} & \frac{300}{349} \end{array}, \qquad \begin{array}{c|ccc} \frac{3}{5} & \frac{9}{50} & & \\ \frac{9}{10} & \frac{9}{40} & \frac{9}{50} & \\ \frac{6}{37} & \frac{234657}{1266325} & -\frac{891891}{2532650} & \frac{9}{50} \\ \hline & \frac{115}{729} & \frac{55}{2457} & \frac{42439}{132678} \\ & \frac{575}{1458} & \frac{550}{2457} & \frac{50653}{132678} \end{array}.$$

These methods have step point orders p=3 and p=4, respectively, and possess an increased order of dispersion which makes these methods highly accurate for oscillatory problems. They will be denoted by SFB_2 and SFB_3.

2. Diagonal-Implicit 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 equation for 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 in (1.2) 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 [14], [11] and [7]. If there are also largely negative eigenvalues, then fixed point iteration would dictate rather small stepsizes in order to get convergence. Therefore, we consider a more powerful class of parallel iteration processes which leads to the same degree of implicitness as occurring in SDIRKN methods. These processes are similar to the *stiff iteration method* applied in [8] for solving the stage vector equation associated with RK methods for first-order ODEs. In order to include RKN correctors that are not A-stable, the analysis will be presented for the preconditioned problem (1.5) (recall that (1.5) reduces to the original problem (1.1) if ε tends to zero).

2.1. Iteration of the stage-vector equation

Let $Y_i^{(\mu)}$ denote the μth iterate to Y_i, and define

$$(2.1) \qquad X_i := Y_i - x_i, \quad X_i^{(\mu)} := Y_i^{(\mu)} - x_i, \quad x_i := y_n + c_i h y'_n + a_i h^2 g(y_n), \quad i = 1, ..., k.$$

Following [6] 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

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

For each of these equations, we define the iteration process

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

where $i=1, \ldots, k$ and $\mu = 1, \ldots, m$. Here, the ω_μ are relaxation parameters and the δ_i are iteration parameters which are assumed to be positive. In order to start the iteration (2.2), we need a predictor to compute the initial approximations $X_i^{(0)}$. The choice of a suitable predictor will be discussed in Section 2.3.

Evidently, if (2.2) converges, then $X_i^{(\mu)}$ converges 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 a dimension equal to that of the system of ODEs, the iteration process (2.2) is on an k-processor computer of the same computational complexity as an m-stage SDIRKN method on a one-processor computer.

We remark that, if nonzero values for ε in (1.5) are used, then the implementation of the iteration formula (2.2) can be simplified by premultiplying with the matrix $T(J_n)$:

$$(2.2') \qquad T(J_n)X_i^{(\mu)} - \delta_i h^2 f(X_i^{(\mu)} + x_i) = T(J_n)X_i^{(\mu-1)} - \delta_i h^2 f(X_i^{(\mu-1)} + x_i) - \omega_\mu \Big[T(J_n)X_i^{(\mu-1)} - h^2 \sum_{j=1}^{k} a_{ij}\, f(X_j^{(\mu-1)} + x_j) \Big].$$

This recursion shows that the preconditioning hardly complicates the implicit relations to be solved.

2.1.1. Definition of the step values. Suppose that we adopt $\mathbf{Y}_i^{(m)}=\mathbf{X}_i^{(m)}+\mathbf{x}_i$ 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) defines these values according to the formulas (cf. [8])

$$(2.3)\qquad y_{n+1} = y_n + hy'_n + b_0h^2g(y_n) + h^2\sum_{i=1}^{k} b_i\, g(Y_i^{(m)}), \qquad y'_{n+1} = y'_n + d_0hg(y_n) + h\sum_{i=1}^{k} d_i\, g(Y_i^{(m)})$$

(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 (cf. [16]). This difficulty can be overcome by applying a similar approach as proposed in [6] for the implementation of implicit RK methods. For simplicity, we describe this approach for the scalar equation y"=g(y). Defining $\mathbf{Y}=(Y_i)$ and $\mathbf{G}=(g(Y_i))$, the corrector (1.2) can be written in the form

$$u_{n+1} = y_n + hy'_n + b_0h^2g(y_n) + h^2\mathbf{b}^T\mathbf{G}, \qquad u'_{n+1} = y'_n + d_0hg(y_n) + h\mathbf{d}^T\mathbf{G},$$

$$\mathbf{G} = h^{-2}A^{-1}[\mathbf{Y} - \mathbf{e}y_n - \mathbf{c}hy'_n - \mathbf{a}h^2g(y_n)],$$

provided that A is nonsingular. 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 $\mathbf{Y}^{(m)}$. Returning to systems of ODEs and to the notation $\mathbf{X}_i^{(m)}$, we obtain

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

where α_i and β_i are the components of the vectors

$$(2.4b)\qquad \alpha := \mathbf{b}^TA^{-1}, \quad \beta := \mathbf{d}^TA^{-1}.$$

In many cases the corrector satisfies the relations of stiff accuracy, i.e., $c_k=1$, $b_0=a_k$ and $\mathbf{b}^TA^{-1}=\mathbf{e}_k^T$. 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

$$(2.4')\qquad y_{n+1} = y_n + hy'_n + b_0h^2g(y_n) + X_k^{(m)}.$$

2.2. The iteration error

We shall say that the *order of the iteration error* of the PC method {(2.1), (2.2), (2.4)} equals q if

$$(2.5)\qquad u_{n+1} - y_{n+1} = O(h^{q+1}), \qquad u'_{n+1} - y'_{n+1} = O(h^{q+1}),$$

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. In first approximation, the iteration error associated with {(2.1), (2.2), (2.4)} can be studied by applying it to the scalar test equation y"=λy, where λ runs through the eigenvalues of $\partial g/\partial y$. Defining the error

$$(2.6)\qquad \varepsilon^{(\mu)} := \mathbf{X} - \mathbf{X}^{(\mu)}, \qquad \mathbf{X} := (X_i), \qquad \mathbf{X}^{(\mu)} := (X_i^{(\mu)}),$$

we deduce from (2.2) that the iteration error (2.6) satisfies the recursion

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

where D is the diagonal matrix with diagonal entries δ_i. Hence,

$$(2.7)\qquad \varepsilon^{(m)} = P_m(H(z))\varepsilon^{(0)}, \qquad P_m(x) = \prod_{\mu=1}^{m} (1 - \omega_\mu x).$$

The matrix $P_m(H(z))$ will be called the *stage vector iteration matrix.*

In the following, we use the notation

(2.8a) $$w_{n+1} := \begin{pmatrix} u_{n+1} \\ hu'_{n+1} \end{pmatrix}, \quad v_{n+1} := \begin{pmatrix} y_{n+1} \\ hy'_{n+1} \end{pmatrix}.$$

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

(2.8b) $$w_{n+1} - v_{n+1} = E_m(z)v_n,$$

where the matrix $E_m(z)$ is a 2-by-2 matrix determined by the RKN parameters and the matrix D. This matrix will be called the *iteration matrix of the diagonal-implicit PC method.* From the formulas (2.4) and (2.4') for the step values it follows that

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

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

We shall first give an order result for the PC method. The actual choice of the predictor will be discussed in Section 2.3. The following theorem holds:

Theorem 2.1. Let the predictor be of order p*, i.e.,

$$\varepsilon^{(0)} = \mathbf{X} - \mathbf{X}^{(0)} = O(h^{p^*+1}).$$

Let

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

Then, for any choice of the matrix D, the order q of the iteration error of the PC method {(2.1), (2.2), (2.4)} is given by $q = 2q^* + p^* - 1$.

Proof. If P_m has a zero at x=1 of order q*, then it follows from (2.9) that for $z \to 0$

$$u_{n+1} - y_{n+1} = z^{q^*} Q_{m-q^*}(1)\, \mathbf{p}^T (A-D)^{q^*} O(h^{p^*+1}),$$

$$u'_{n+1} - y'_{n+1} = z^{q^*} Q_{m-q^*}(1)\, \mathbf{d}^T A^{-1} (A-D)^{q^*} h^{-1} O(h^{p^*+1}).$$

Recalling definition (2.5) and observing that $z=O(h^2)$, the theorem easily follows. □

2.3. The predictor

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 $\partial g/\partial y$). Therefore, apart from the usual approach to choose an explicit predictor, we will also consider some implicit predictors. Notice that, as a consequence of this choice, the number of implicit relations to be solved per step in increased by one.

In Table 2.1 we have collected various possibilities for choosing a one-step predictor. Here, $\gamma_i = \frac{1}{2} c_i^2$. Observe that the predictors III and IV are of order 2, whereas the first two predictors are only of first order. Furthermore, we remark that the predictors II and IV are recommended in cases where extra damping of stiff components is desirable.

To compare the computational costs required by the various predictors, we also list the number of systems of dimension d to be solved in each step on each processor, and the number of sequential LU decompositions (LUDs) per step. On the basis of this table, the predictors I and II seem to be the most attractive ones.

Table 2.1. Survey of one-step predictors

Predictor	$X_i^{(0)}$, i = 1, ... , k	p*	Systems	LUDs
I	$- a_i h^2 g(y_n)$	1	m	1
II	$- a_i h^2 g(y_n) + \delta_i h^2 g(X_i^{(0)}+x_i)$	1	m+1	1
III	$- a_i h^2 g(y_n) + h^2[\delta_i g(X_i^{(0)}+x_i) + (\gamma_i - \delta_i) g(y_n)]$	2	m+1	1
IV	$- a_i h^2 g(y_n) + \gamma_i h^2 g(X_i^{(0)}+x_i)$	2	m+1	2

The predictors listed in Table 2.1 are such that we can write

(2.10) $\varepsilon^{(0)} = X - X^{(0)} = \mathbf{k}_1(z)\, y_n + \mathbf{k}_2(z)\, hy'_n,$

where the vectors $\mathbf{k}_1(z)$ and $\mathbf{k}_2(z)$ are determined by the RKN parameters and, in case of the predictors II and III, also by the matrix D. The iteration matrix $E_m(z)$ in (2.8b) assumes the form

(2.8c) $$E_m(z) := \begin{pmatrix} \mathbf{p}^T P_m(H(z))\mathbf{k}_1(z) & \mathbf{p}^T P_m(H(z))\mathbf{k}_2(z) \\ \mathbf{d}^T A^{-1} P_m(H(z))\mathbf{k}_1(z) & \mathbf{d}^T A^{-1} P_m(H(z))\mathbf{k}_2(z) \end{pmatrix}.$$

This matrix will be used in deriving the stability function of the PC methods (see Section 2.5.3).

2.4. The rate of convergence

Ideally, the overall rate of convergence should be based on some norm of the iteration matrix $E_m(z)$ for all z-values that are relevant for the problem (1.5). However, this would lead to iteration parameters that depend on the predictor and on m. By observing that the entries of $E_m(z)$ are small if the magnitude of the stage vector iteration matrix $P_m(H(z))$ is small, we are led to minimize, in some sense, the magnitude of $P_m(H(z))$ for negative values of z. In this paper, we consider the case where the magnitude of $P_m(H(z))$ is estimated by its spectral radius. By minimizing the spectral radius of $P_m(H(z))$, the iteration parameters can be determined independently of the predictor and of the number of iterations m. Denoting the spectral radius of a matrix M by $\rho(M)$, we characterize the rate of convergence of the stage vector iteration by

(2.11) $$r_m(z) := \sqrt[m]{\rho(P_m(H(z)))}, \quad r_m := \max_{-\beta \le z \le 0} r_m(z), \quad r := (r_1, r_2, \dots)^T, \quad \beta := h^2\rho^*.$$

where ρ^* is the parameter occurring in (1.6).

Furthermore, we denote the spectrum of H(z) by $\Lambda(H(z))$, and we define

(2.12) $$\rho(z) := \max\{\, |\lambda - 1| : \lambda \in \Lambda(H(z))\}, \quad \rho := \max\{\, |\lambda - 1| : \lambda \in \Lambda(H)\,\}, \quad \Lambda(H) := \{\, \Lambda(H(z)) : -\beta \le z \le 0\}.$$

2.5. Choice of iteration parameters

In the following subsections, we shall discuss a few special cases for choosing the relaxation parameters ω_μ and the matrix D. We start with a discussion of the *stiff iteration* approach which was investigated in [8] for solving implicit RK methods for first-order ODEs.

2.5.1. Stiff iteration. In this case the matrix D is such that $\Lambda(H(-\infty))$ is contained in a circle with minimal radius $\rho(-\infty)$ and centered at 1, and the relaxation parameters are all equal to 1, so that $r_m = \rho$. Stiff iteration preassumes that the corrector is A-stable, hence we set $\beta=\infty$ in (2.15). The following theorem holds for k=2:

Theorem 2.2. Let k=2, then the following assertions hold for the stiff iteration method:

(a) if $\det(A) > 0$ and if either $\{a_{12}a_{21} \le 0$ and $a_{22} > 0\}$ or $\{a_{11} > 0$ and $a_{22} \le 0\}$, 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 $\mathrm{Tr}(A) > -2\,[\det(A)]^{1/2}$, then the eigenvalues of H(z) are real and positive for all negative z.

Proof. (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

$$\delta_1 = \frac{\det(A)}{a_{22}}\left(1 \pm \sqrt{1 - \frac{a_{11}a_{22}}{\det(A)}}\right), \quad \delta_2 = \frac{2\det(A) - \delta_1 a_{22}}{a_{11}}.$$

By an elementary calculation assertion (a) can now be verified.

(b) Assertion (b) is satisfied if there exists a vector v, such that $H(z)v=v$ for all z, i.e., if $(I - zA)v = (I - zD)v$. This relation is true for all z if $D^{-1}Av=v$. Evidently, if (a) holds, then $D^{-1}A$ has only eigenvalues 1 which proves (b).

(c) Since the entries of H(z) are real for all negative z, we deduce from (b) that H(z) has real eigenvalues for $z<0$. Hence, by showing that $\det(H(z)) = \det(I - zD)^{-1}\det(I - zA) = \det(I - zD)^{-1}\,[\det(A)z^2 - \mathrm{Tr}(A)z + 1]$ is positive for $z<0$, we can prove assertion (c). This leads us to assertion (c). □

Table 2.2. Stiff matrices D and corresponding vectors **r**.

Generating RK method	s	k	δ_1	δ_2	δ_3	δ_4	$\rho(-\infty)$	**r**
Radau IIA	2	2	1/18	1/2			0	e/2
Lobatto IIIA (≡Newton-Cotes)	3	2	1/24	1/6			0	e/3
Lagrange with $\mathbf{c} = (3/4,1)^T$	3	2	3/32	1/6			0	e/3
Radau IIA	3	3	8417/16328	255/19799	1483/35645		0.0028	0.77e
Lobatto IIIA	4	3	754/7243	113/12480	999/13576		0.0007	0.52e
Newton-Cotes	4	3	125/8979	988/18531	85/729		0.0035	0.52e
Lagrange: $\mathbf{c} = (7/12,5/6,1)^T$	4	3	362/8683	605/7281	783/6628		0.0019	0.53e
Radau IIA	4	4	2625/7342	1225/7601	76/20731	71/10024	0.023	0.81e
Lobatto IIIA	5	4	3384/40409	25/5154	221/10255	134/3319	0.074	0.58e
Newton-Cotes	5	4	81/12772	493/20960	337/6661	921/10594	0.026	0.64e
Lagrange: $\mathbf{c} = (1/6,7/12,11/12,1)^T$	5	4	71/4500	105/9613	400/7807	1177/18717	0.016	0.55e

For k>2, we did not succeed in deriving the optimal matrix D by analytical methods, so that we resorted to numerical search techniques. For a few RKN correctors generated by classical RK methods, Table 2.2 presents the entries of the matrices D that are optimal for stiff iteration (stiff matrices D). The given entries in this table (and in the subsequent tables) are rational approximations to the decimal numbers we found. Furthermore, we include the RKN correctors generated by the Lagrange methods with collocation vectors $\mathbf{c}=(3/4,1)^T$, $\mathbf{c}=(7/12,5/6,1)^T$ and $\mathbf{c}=(1/6,7/12,11/12,1)^T$ proposed in [8]. For all methods, we also list the vectors **r** as defined in (2.14).

2.5.2. Zarantonello iteration. Assuming that all relaxation parameters equal 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. [18]), 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 (with smallest radius) containing the eigenvalues of H(z). We shall call this iteration mode *Zarantonello iteration.* As for stiff iteration, we have $r_m = \rho$, however, r_m is expected to be smaller.

Table 2.3. Zarantonello matrices D and corresponding vectors **r**.

Generating RK method	s	k	δ_1	δ_2	δ_3	δ_4	$\rho(-\infty)$	**r**
Radau IIA	2	2	1/24	3/8				0.33e
Lobatto IIIA (≡Newton-Cotes)	3	2	611/17603	347/2500				0.20e
Lagrange $\mathbf{c} = (3/4,1)^T$	3	2	391/5000	139/1000				0.20e
Radau IIA	3	3	453/2500	47/2500	547/2500		0.47	0.64e
Lobatto IIIA	4	3	133/1250	1/100	431/5000		0.47	0.47e
Newton-Cotes	4	3	57/5000	441/10000	971/10000		0.36	0.42e
Lagrange with $\mathbf{c} = (7/12,5/6,1)^T$	4	3	43/1250	69/1000	578/5871		0.34	0.44e
Radau IIA	4	4	2625/7342	1225/7601	76/20731	71/10024	0.023	0.81e
Lobatto IIIA	5	4	1/10	1/200	1/50	7/200	0.561	0.57e
Newton-Cotes	5	4	81/12772	493/20960	337/6661	921/10594	0.026	0.64e
Lagrange: $\mathbf{c} = (1/6,7/12,11/12,1)^T$	5	4	71/4500	105/9613	400/7807	1177/18717	0.016	0.55e

A numerical search yields the results listed in Table 2.3. For the 4-stage Radau IIA and the 5-stage Newton-Cotes and Lagrange correctors we could not find a better D matrix than in the stiff case, so that the Zarantonello matrix D is identical with the stiff matrix D yielding identical convergence factors. In all other cases, Zarantonello iteration possesses considerably better convergence factors.

2.5.3. Chebyshev iteration. The PC method can be made more rapidly converging by a more sophisticated choice of the relaxation parameters ω_μ and the iteration parameters δ_i. The optimal choice of the relaxation parameters leads to a minimax problem for the polynomial $P_m(x)$ on the set $\Lambda(H)$. Such minimax problems have been extensively studied in the literature and can be solved by identifying the polynomial P_m in (2.7) with a shifted Chebyshev polynomial, the shift parameters being determined by the ellips-shaped region containing the complex set $\Lambda(H)$ (see [12]). We shall consider this approach for the simplified case where the matrix D is such that $\Lambda(H)$ is contained in a real positive interval [a,b]. The most optimal choice of P_m is then given by the polynomial (see e.g. [18])

$$(2.13) \qquad P_m(x) = \frac{1}{\tau_m} T_m\left(\frac{b+a-2x}{b-a}\right), \quad \tau_m := T_m\left(\frac{b+a}{b-a}\right),$$

where T_m denotes the first-kind Chebyshev polynomial of degree m. From (2.7) it follows that the corresponding relaxation parameters are the inverses of the zeros of the polynomial (2.13), i.e.,

$$\omega_\mu = \frac{2}{b + a - (b - a)\cos\left(\frac{(2\mu-1)\pi}{2m}\right)}, \quad \mu = 1, \ldots, m.$$

Since P_m is bounded by $1/\tau_m$, we may write

$$(2.14) \qquad r_m := \sqrt[m]{\frac{1}{\tau_m}} \approx \sqrt[m]{2}\,\frac{b-a}{2(b+a)} \in \frac{b-a}{b+a}\,[1/2,\,1] \text{ as } a \to b.$$

Evidently, the Chebyshev approach will be more rapidly converging as b/a is closer to 1, hence we determined D such that b/a is minimal (notice that $b \geq 1$). The corresponding iteration method will be called *Chebyshev iteration.*

From Theorem 2.2 it follows that under the conditions of part (a) of the theorem, the matrices D corresponding to stiff iteration can also be used for Chebyshev iteration. It turns out that the conditions of part (a) are fulfilled by a number of RKN correctors generated by classical RK collocation methods for first-order ODEs (for these correctors, we have $\beta_{stab}=\infty$). Moreover, we found that for these correctors the corresponding matrices D minimize the value of b/a. Hence:

Corollary 2.1. For k=2 the matrices D corresponding to stiff iteration are optimal for Chebyshev iteration. □

Table 2.4. Chebyshev matrices D and corresponding vectors **r**.

Generating RK method	s	k	δ_1	δ_2	δ_3	δ_3	[a,b]	$\mathbf{r}^T$
Radau IIA	2	2	1/18	1/2			[1/2,1]	(.34, .25, ... , .18)
Lobatto IIIA (≡Newton-Cotes)	3	2	1/24	1/6			[2/3,1]	(.20, .15, ... , .11)
Lagrange with $\mathbf{c} = (3/4,1)^T$	3	2	3/32	1/6			[2/3,1]	(.20, .15, ... , .11)
Radau IIA	3	3	18/125	7/500	11/50		[.33,1.48]	(.63, .50, ... , .36)
Lobatto IIIA	4	3	7/40	7/500	31/200		[.27,1.00]	(.57, .44, ... , .32)
Newton-Cotes	4	3	1/50	81/1000	93/500		[.29,1.00]	(.55, .42, ... , .30)
Lagrange: $\mathbf{c} = (7/12,5/6,1)^T$	4	3	1/25	17/200	1/8		[.41,1.74]	(.62, .49, ... , .35)
Radau IIA	4	4	141/1000	7/1000	7/125	31/200	[.21,1.49]	(.75, .63, ... , .46)
Lobatto IIIA	5	4	1/20	9/1000	113/1000	9/50	[.15,1.00]	(.73, .61, ... , .44)
Newton-Cotes	5	4	1/125	43/1250	167/2000	4/25	[.18,1.38]	(.77, .65, ... , .47)
Lagrange: $\mathbf{c} = (1/6,7/12,11/12,1)^T$	5	4	29/1000	1/50	53/50	33/250	[.20,1.00]	(.66, .53, ... , .38)

Table 2.4 presents the matrices D that are optimal for Chebyshev iteration (Chebyshev matrices D) and corresponding vectors **r**. A comparison with the Tables 2.2 and 2.3 reveals that the convergence of Chebyshev stage vector iteration should be substantially faster than that of stiff and Zarantonello iteration. A number of experiments where the rates of convergence in a single step were considered, confirmed this conclusion. However, when the overall performance was considered, it turned out that Chebyshev iteration is by far inferior to stiff and Zarantonello iteration. This is illustrated in the following example.

Example 2.1. Consider the model problem (see Kramarz [10]):

$$(2.15) \qquad y''(t) = \begin{pmatrix} 2498 & 4998 \\ -2499 & -4999 \end{pmatrix} y(t), \quad y(0) = \begin{pmatrix} 2 \\ -1 \end{pmatrix}, \quad y'(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad 0 \leq t \leq 100,$$

with exact solution $y(t)=(2\cos(t),-\cos(t))^T$. For the indirect two-stage Radau IIA corrector Table 2.5 lists the number of minimal correct digits

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

obtained for a few values of h and m. Negative NCD-values are indicated by $*$. This table shows the inferiority of Chebyshev iteration. Since the matrices D in the stiff and Chebyshev iteration mode of the indirect two-stage Radau IIA corrector are identical (see Corollary 2.1), the poor performance of Chebyshev iteration is apparently caused by the choice of the relaxation parameters. □

Table 2.5. NCD-values for problem (2.15) obtained by the PC method with predictor I and indirect two-stage Radau IIA corrector.

Iteration mode	h	m=2	m=3	m=4	m=5	...	m=∞
Stiff	1/10	2.5	2.6				2.6
	1/20	3.4	3.5				3.5
	1/40	4.4					4.4
Zarantonello	1/10	2.5	2.6				2.6
	1/20	3.5					3.5
	1/40	4.4					4.4
Chebyshev	1/10	*	*	0.1	0.9	...	2.6
	1/20	*	*	*	0.3	...	3.5
	1/40	*	*	*	*	...	4.4

The explanation of the poor overall performance of Chebyshev iteration is that, in spite of the rapid Chebyshev convergence in each step, the stability of the integration process requires many more iterations per step than required by the convergence criterion. To see the reasons for this phenomenon we have to define the *stability function* for diagonal-implicit PC methods. The RKN corrector satisfies the relation (cf. [9])

$$(2.16)\qquad \mathbf{w}_{n+1} = M(z)\mathbf{v}_n,\quad M(z) := \begin{pmatrix} 1 + zb_0 + z\mathbf{b}^T(I - Az)^{-1}[\mathbf{e} + z\mathbf{a}] & 1 + z\mathbf{b}^T(I - Az)^{-1}\mathbf{c} \\ zd_0 + z\mathbf{d}^T(I - Az)^{-1}[\mathbf{e} + z\mathbf{a}] & 1 + z\mathbf{d}^T(I - Az)^{-1}\mathbf{c} \end{pmatrix}.$$

On substitution into (2.8b) we obtain

$$(2.17)\qquad \mathbf{v}_{n+1} = [M(z) - E_m(z)]\mathbf{v}_n.$$

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

$$(2.18)\qquad R_m(z) := \rho([M(z) - E_m(z)]).$$

From (2.16) it follows that M(z) approaches a matrix with a double unit eigenvalue for $z \to 0$. Hence, the eigenvalues of the stability matrix $M(z) - E_m(z)$ easily move outside the unit circle, unless the entries of $E_m(z)$ are close to zero as $z \to 0$. The definition of $E_m(z)$ strongly suggests choosing all zeros of the polynomial $P_m(x)$ at x=1, i.e., all relaxation parameters equal to 1. In order to illustrate that unit relaxation parameters improve the performance dramatically, we repeated the experiment in Example 2.1 by iterating the indirect three-stage Radau IIA corrector using relaxation parameters equal to 1 together with the Chebyshev matrix D (*stationary Chebyshev iteration*).

Example 2.2. Table 2.6 compares the Chebyshev and stationary Chebyshev mode of the indirect three-stage Radau IIA corrector for problem (2.15). The superiority of the stationary Chebyshev mode over the 'true' Chebyshev mode is evident. □

Table 2.6. NCD-values for problem (2.15) obtained by the PC method with predictor I and indirect three-stage Radau IIA corrector.

Iteration mode	h	m=2	m=3	m=4	m=5	...	m=∞
Chebyshev	1/10	*	*	*	*	...	6.6
	1/20	*	*	*	*	...	8.1
	1/40	*	*	*	*	...	9.6
Stationary	1/10	*	*	*	6.6		6.6
Chebyshev	1/20	4.6	7.7	8.1			8.1
	1/40	5.9	9.4	9.6			9.6

2.6. Selection of methods

Since stability plays such 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 iteration method is stable for all z in the interval $[-\beta,0]$ and for all m equal to or

greater than this value. Let us denote this critical value of m by m_0 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 stable. From Table 2.1 it follows that $m_1 = m_0$ for the explicit predictor I, and $m_1 = m_0 + 1$ for the implicit predictors II, III, and IV. In Table 2.7, the values of m_1 are listed for a number of RK-generated RKN correctors using the explicit predictor I and the implicit predictor II. For each k, the minimal values are indicated in bold face.

Table 2.7. Values of m_1 for Zarantonello and stationary Chebyshev iteration.

Generating RK method	k	r=s	Zarantonello iteration		Stationary Chebyshev iteration	
			Explicit Predictor	Implicit Predictor	Explicit Predictor	Implicit Predictor
Radau IIA	2	2	4	**2**	**2**	**2**
Lobatto IIIA (≡Newton-Cotes)	2	3	>10	>10	7	7
Lagrange with $\mathbf{c} = (3/4,1)^T$	2	3	3	4	**2**	3
Radau IIA	3	3	8	5	7	**4**
Lobatto IIIA	3	4	>10	>10	>10	>10
Newton-Cotes	3	4	>10	>10	>10	>10
Lagrange: $\mathbf{c} = (7/12,5/6,1)^T$	3	4	6	6	6	7
Radau IIA	4	4	>10	>10	>10	>10
Lobatto IIIA	4	5	>10	>10	>10	>10
Newton-Cotes	4	5	>10	>10	>10	>10
Lagrange: $\mathbf{c} = (1/6,7/12,11/12,1)^T$	4	5	**7**	8	>10	>10

This table shows that for k=2 there are four combinations of predictor, corrector and iteration mode with a minimal m_1-value. From these combinations we have chosen the Lagrange-based method because the stage order r of the indirect Lagrange corrector is higher than that of the indirect Radau corrector. For k=3 and k=4 there is just one 'optimal' combination. Thus, we are led to the following three optimal A-stable combinations:

(2.19)
Explicit - Lagrange - Chebyshev with at least 2 implicit sequential stages (ELC_2)
Implicit - Radau IIA - Chebyshev with at least 4 implicit sequential stages (IRC_4)
Explicit - Lagrange - Zarantonello with at least 7 implicit sequential stages (ELZ_7).

Since the global order of PC methods equals min{p,q}, it follows from Table 2.1 and Theorem 2.1 that the global orders of the methods ELC_2, IRC_4 and ELZ_7 are given by min{p,2m} (recall that q* equals the number of iterations), so that both the order and the stage order of the corrector is already reached for $m \geq p/2$. Hence, by satisfying the stability condition $m \geq m_0$, we are sure that the PC method has the same order p and stage order r of the corrector.

For completeness, we give the correctors selected above, and the corresponding vector β (see (2.4b)). Since these correctors originate from stiffly accurate RK methods, they all have $\alpha = e_k^T$.

The indirect Lagrange corrector with k=2 is defined by

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 3/4 & 17/128 & 5/16 & -21/128 \\ 1 & 11/54 & 14/27 & -2/9 \\ \hline & 11/54 & 14/27 & -2/9 \\ & 5/18 & 8/9 & -1/6 \end{array}, \qquad \beta = (-64/9, 6)^T.$$

The indirect Radau IIA corrector with k=s-1=3 (written in the form (1.3')) reads

$$\begin{array}{c|ccc} 0.155051025722 & 0.021835034191 & -0.019857254099 & 0.010042630197 \\ 0.644948974278 & 0.177190587432 & 0.038164965809 & -0.007375963530 \\ 1.000000000000 & 0.318041381744 & 0.181958618256 & 0.000000000000 \\ \hline & 0.318041381744 & 0.181958618256 & 0.000000000000 \\ & 0.376403062700 & 0.512485826188 & 0.111111111111 \end{array},$$

with $\beta = (5.531972647422, -7.531972647422, 5)^T$.

The indirect Lagrange corrector with k=4 is given by

0.0	0.0	0.0	0.0	0.0	0.0
0.166666666667	0.007240660574	0.008214814815	-0.003273544974	0.004739057239	-0.003032098765
0.583333333333	0.027437044052	0.114192181070	0.040901388889	-0.030997357838	0.018605632716
0.916666666667	0.034578097443	0.229227777778	0.165230621693	-0.032210648148	0.023313040123
1.000000000000	0.037142857143	0.256177777778	0.202057142857	-0.017777777778	0.022400000000
	0.037142857143	0.256177777778	0.202057142857	-0.017777777778	0.022400000000
	0.029870129870	0.325333333333	0.438857142857	0.193939393939	0.012000000000

with $\beta = (-4.8, 3.526530612245, -19.834710743802, 17.6)^T$.

3. Numerical Comparisons

In this section, we restrict our considerations to the methods (2.19) and the two-, three- and four-stage SDIRKN methods of Section 1.2. In the experiments reported below, we dropped the fixed-numbers-of-iterations strategy used in the preceding examples. Instead, the number of iterations m was determined dynamically by the stability criterion $m \geq m_0$ together with a condition on the iteration error. It seems natural to require that the iteration error is of the same (stiff) order in h as the local error of the corrector. This leads us to the convergence criterion

$$(3.1) \qquad m \geq m_0, \quad \text{Max}_i \, \| X_i^{(m)} - h^2 \sum_{j=1}^{k} a_{ij} \, g(X_j^{(m)} + x_j) \|_\infty \leq C \, h^{p+1},$$

where C is a parameter independent of h. In our numerical experiments we always used $C=10^{-2}$.

Furthermore, in the tables of results, M denotes the (averaged) number of sequential systems to be solved per unit interval and N denotes the number of integration steps per unit interval.

3.1. Kramarz problem

Table 3.1 compares the methods specified above when applied to problem (2.15) of Kramarz. For this linear problem, where the Jacobian and its LU-decomposition can be computed once and for all at the beginning of the integration interval, the value of M may serve as a measure of the sequential computational costs. The results clearly show that the parallel methods IRC_4 and ELZ_7 are by far the most efficient ones, in spite of the fact that in this example no order reduction is observed. However, by the same reason, the method ELC_2 is only slightly more efficient than the other third-order methods. We observe that IRC_4 and ELZ_7 do not need more iterations to satisfy the convergence criterion (3.1) than already prescribed by the stability condition $m \geq m_0$.

Table 3.1. Values of NCD / M for problem (2.15).

Method	k	p	r	N = 5	N = 10	N = 20	N = 40	N = 80
$Nørsett_2$	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
$Nørsett_3$	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
ELC_2	2	3	3	2.0 / 19	2.9 / 39	3.8 / 78	4.6 / 156	
IRC_4	3	5	3	4.2 / 20	6.1 / 40	7.7 / 80	9.4 / 160	
ELZ_7	4	5	5	6.6 / 35	8.1 / 70	9.6 / 140	10.9 / 280	

3.2. Nonlinear partial differential equation

We apply the methods to the semidiscretization of the partial differential equation (see also [9])

$$(3.2) \qquad \frac{\partial^2 u}{\partial t^2} = \frac{4\pi^2 u^2}{1 + 2x - 2x^2} \frac{\partial^2 u}{\partial x^2} + u \, [4\cos^2(2\pi t) - 1], \quad 0 \leq t \leq 1, \quad 0 \leq x \leq 1,$$

with initial and Dirichlet boundary conditions such that its exact solution is given by $u=(1+2x-2x^2)\cos(2\pi t)$. By using second-order symmetric spatial discretization on a uniform grid with mesh $\Delta x=1/20$, we obtain a set of 19 ODEs. Notice that, as a consequence of these choices, we have no spatial discretization error. Table 3.2 is the analogue of

Table 3.1. Again, no order reduction is shown. If M is taken as a measure for the sequential computational costs, then only the four-processor method ELZ_7 can beat the one-processor methods. However, in this case of a semidiscrete nonlinear PDE, it is more realistic to consider the evaluations of the Jacobian and the corresponding LU-decompositions as the bulk of the computational work. This implies that all methods require approximately the same computational effort per step. As a consequence, both IRC_4 and ELZ_7 are the most efficient methods, while ELC_2 is only superseded by SFB_3. Notice that the residual condition in (3.1) now plays a dominant role in the determination of the number of iterations needed by the PC methods.

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

Method	k	p	r	N = 40	N = 80	N = 160	N = 320
$Nørsett_2$	2	3	1	2.5 / 80	3.2 / 160	4.1 / 320	4.9 / 640
SFB_2	2	3	1	*	3.3 / 160	4.2 / 320	5.1 / 640
$Nørsett_3$	3	4	1	*	3.6 / 240	4.5 / 480	5.3 / 960
SFB_3	3	4	1	4.4 / 120	5.6 / 240	6.8 / 480	7.9 / 960
B_4	4	3	1	*	3.9 / 320	5.1 / 640	6.3 / 1280
ELC_2	2	3	3	3.8 / 296	4.7 / 566	5.5 / 946	6.4 / 1628
IRC_4	3	5	3	6.4 / 698	7.8 / 1126	9.2 / 1572	10.5 / 2274
ELZ_7	4	5	5	6.7 / 422	8.2 / 584	11.2 / 1120	13.1 / 2240

3.3. Prothero-Robinson-type problem

Consider the system of (uncoupled) second-order Prothero-Robinson-type equations (cf. [15]):

$$(3.3) \qquad y''(t) = J\,[y(t) - g(t)] + g''(t), \quad J := \mathrm{diag}(-100^{j-1}), \quad g(t) = (1 + e^{-jt}), \quad j = 1, \ldots, 6; \quad 0 \le t \le 10,$$

with initial values $y(0)=g(0)$, $y'(0)=g'(0)$, so that its exact solution is given by $y(t)=g(t)$. In this example, the stiffness causes that the classical order p is no longer shown. The results demonstrate the superiority of the methods IRC_4 and ELZ_7. The number of iterations in the PC methods is completely determined by the residual condition in (3.1).

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

Method	k	p	r	N = 1	N = 2	N = 4	N = 8	N = 16
$Nørsett_2$	2	3	1	1.5 / 2	2.3 / 4	3.1 / 8	3.9 / 16	4.0 / 32
SFB_2	2	3	1	1.3 / 2	2.0 / 4	2.9 / 8	3.8 / 16	4.9 / 32
$Nørsett_3$	3	4	1	1.7 / 3	2.3 / 6	3.3 / 12	4.5 / 24	5.0 / 48
SFB_3	3	4	1	1.2 / 3	2.8 / 6	3.3 / 12	3.5 / 24	4.6 / 48
B_4	4	3	1	1.6 / 4	2.3 / 8	3.0 / 16	3.9 / 32	4.8 / 64
ELC_2	2	3	3	2.3 / 4	3.1 / 8	4.2 / 17	4.2 / 34	5.7 / 77
IRC_4	3	5	3	3.7 / 5	5.1 / 13	6.0 / 30	6.5 / 84	
ELZ_7	4	5	5	5.1 / 10	5.3 / 18	7.6 / 33	9.5 / 97	

3.4. Fehlberg problem

Consider the nonlinear orbit equation (cf. [3]):

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

with exact solution $y(t) = (\cos(t^2), \sin(t^2))^T$. In this example, the stiffness causes that the classical order p is no longer shown. As in the preceding examples, the methods IRC_4 and ELZ_7 are considerably more efficient.

Table 3.4. Values of NCD / M for problem (3.4) with M and N rounded to integer values.

Method	k	p	r	N = 10	N = 20	N = 39	N = 78	N = 156
$Nørsett_2$	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
$Nørsett_3$	3	4	1	0.1 / 29	0.2 / 59	0.8 / 117	1.8 / 235	3.1 / 470
SFB_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
ELC_2	2	3	3	0.2 / 86	0.8 / 135	1.7 / 229	2.6 / 413	3.6 / 766
IRC_4	3	5	3	1.2 / 130	2.7 / 173	4.2 / 274	5.7 / 477	7.2 / 867
ELZ_7	4	5	5	2.5 / 75	4.1 / 137	5.7 / 274	7.2 / 548	8.7 / 1096

References

[1] **Burrage, K. (1990)**: A study of order reduction for semi-linear problems, Report, University of Auckland.

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

[3] **Fehlberg, E. (1972)**: Klassische Runge-Kutta-Nyström Formeln mit Schrittweiten-Kontrolle für Differentialgleichungen x"=f(t,x), Computing **10**, 305-315.

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

[5] **Hairer, E., Nørsett, S.P. & Wanner, G. (1987)**: Solving ordinary differential equations I, Nonstiff problems, Springer-Verlag, Berlin.

[6] **Hairer, E. & Wanner, G. (1988)**: RADAU 5 - an implicit Runge-Kutta code, Report, Université de Genève.

[7] **Houwen, P.J. van der & Sommeijer, B.P. (1990)**: Parallel iteration of high-order Runge-Kutta methods with stepsize control, J. Comp. Appl. Math. **29**, 111-127.

[8] **Houwen, P.J. van der & Sommeijer, B.P. (1990)**: Iterated Runge-Kutta methods on parallel computers, Report NM-R9001, Centre for Mathematics and Computer Science, Amsterdam (to appear in SISSC.).

[9] **Houwen, P.J. van der, Sommeijer, B.P. & Nguyen huu Cong (1990)**: Stability of collocation-based Runge-Kutta-Nyström methods, Report NM-R9016, Centre for Mathematics and Computer Science, Amsterdam.

[10] **Kramarz, L. (1980)**: Stability of collocation methods for the numerical solution of y"=f(x,y), BIT **20**, 215-222.

[11] **Lie, I. (1987)**: Some aspects of parallel Runge-Kutta methods, Report No. 3/87, Division Numerical Mathematics, University of Trondheim.

[12] **Manteuffel, T.A. (1977)**: The Tchebyshev iteration for nonsymmetric linear systems, Numer. Math. **28**, 307-327.

[13] **Nørsett, S.P. (1974)**: Semi-explicit Runge-Kutta methods, Report Mathematics and Computation No.6/74, Depart. of Mathematics, University of Trondheim.

[14] **Nørsett, S.P. & Simonsen, H.H. (1989)**: Aspects of parallel Runge-Kutta methods, in: A. Bellen, C.W. Gear and E Russo (Eds.): Numerical Methods for Ordinary Differential Equations, Proceedings L'Aquila 1987, Lecture Notes in Mathematics **1386**, Springer-Verlag, Berlin.

[15] **Prothero, A. & Robinson, A. (1974)**: On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations, Math. Comp. **28**, 145-162.

[16] **Shampine, L.F. (1980)**: Implementation of implicit formulas for the solution of ODEs, SIAM J. Sci. Stat. Comp. **1**, 103-118.

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

[18] **Varga, R.S. (1957)**: A comparison of the successive overrelaxation method and semi-iterative methods using Chebyshev polynomials, SIAM J. Appl. Math. **5**, 39-47.