ELSEVIER

Applied Numerical Mathematics 15 (1994) 357–374

APPLIED NUMERICAL MATHEMATICS

Butcher–Kuntzmann methods for nonstiff problems on parallel computers

P.J. van der Houwen *, B.P. Sommeijer

CWI, P.O. Box 94079, 1090 GB Amsterdam, Netherlands

Abstract

From a theoretical point of view, the Butcher–Kuntzmann Runge–Kutta methods belong to the best step-by-step methods for nonstiff problems. These methods integrate first-order initial-value problems by means of formulas based on Gauss–Legendre quadrature, and combine excellent stability features with the property of superconvergence at the step points. Like the IVP itself, they only need the given initial value without requiring additional starting values, and therefore are a natural discretization of the initial-value problem. On the other hand, from a practical point of view, these methods have the drawback of requiring in each step an approximation to the solution of a system of equations of dimension sd, s and d being the number of stages and the dimension of the initial-value problem, respectively. However, parallel computers have changed the scene and enable us to design parallel iteration methods for approximating the solution of the implicit systems such that the Butcher–Kuntzmann methods become efficient step-by-step methods for integrating initial-value problems. In this contribution, we address nonstiff initial-value problems and we investigate the possibility of introducing preconditioners into the iteration method. In particular, the iteration error will be analysed. By a number of numerical experiments it will be shown that the Butcher–Kuntzmann method, in combination with the preconditioned, parallel iteration scheme, performs much more efficiently than the best sequential methods.

Keywords: Numerical analysis, Runge–Kutta methods, Preconditioning, Parallelism

1. Introduction

From a theoretical point of view, the Butcher–Kuntzmann Runge–Kutta methods belong to the best step-by-step methods available in the literature. These methods integrate the initial-value problem (IVP)

$$y'(t) = f(y(t)), \qquad y(t_0) = y_0, \qquad y : \mathbb{R} \to \mathbb{R}^d \tag{1.1}$$

* Corresponding author. E-mail: P.J.van.der.Houwen@cwi.nl.

SSDI 0168-9274(94)00032-C

by means of formulas based on Gauss–Legendre quadrature which are, as is stated in [3, p. 75] "best in the sense that they integrate exactly polynomials of as high degree as possible". Like the IVP itself, the Butcher–Kuntzmann methods only need the given initial value y_0 without requiring additional starting values, and therefore are a natural discretization of the IVP. Excellent stability features are combined with the property of superconvergence at the step points. As was proved by Kuntzmann [7] and Butcher [1], s-point Butcher–Kuntzmann methods possess order $2s$.

From a practical point of view however, these methods have the drawback that in each integration step a system of equations has to be solved. The dimension of this system equals sd, where s denotes the number of stages and d is the dimension of the initial-value problem. The amount of computational work involved in solving this huge system prevented the Butcher–Kuntzmann methods to become widely used. This is especially the case for nonstiff problems, where we do not need the robust stability behaviour of these methods; therefore, on traditional (i.e., sequential) computers the Butcher–Kuntzmann methods are never used to solve nonstiff problems, since explicit methods are much more appropriate.

However, parallel computers have changed the scene. Using a predictor–corrector approach with a Butcher–Kuntzmann method as the corrector, it is the high-accuracy property of this corrector that we can exploit; a feature which is of course also nice for nonstiff problems. The resulting method is explicit and, owing to the large amount of inherent parallelism, very efficient. This predictor–corrector approach based on an (implicit) Runge–Kutta corrector has been studied in several papers (see e.g. [5,6,8,9,12]). Once the corrector has been fixed, the main concern in obtaining optimal efficiency is to reduce the number of iterations, that is, the number of f-evaluations.

A first, and rather obvious, step in this direction is to use a predictor of high order (see also [8] for a few numerical experiments). For this purpose the properties of the Butcher–Kuntzmann corrector can be exploited as well. For example, if the stage value components calculated in the previous step are used to generate a prediction, then we obtain an initial iterate of relatively high order. This is due to the collocation principle underlying the Butcher–Kuntzmann methods.

Another, more sophisticated, technique to increase the convergence of the predictor–corrector iteration is studied in this paper. By introducing preconditioners into the iteration process, a significant reduction is obtained in the number of iterations needed to reach the accuracy of the corrector. Here, we distinguish two approaches: one technique is based on the knowledge of the spectrum of the Jacobian matrix and is quite useful if this information is available. The second approach uses the Jacobian itself within the iteration process and is beneficial in the case of problems for which a Jacobian-times-vector multiplication is cheap compared to an evaluation of the right-hand side function.

In conclusion, this contribution to the SCADE conference on the occasion of the sixtieth birthday of John Butcher is an attempt to design a parallel iteration method for solving the implicit systems in such a way that the Butcher–Kuntzmann methods become efficient step-by-step methods for integrating *nonstiff* initial-value problems. The format of the parallel iteration including the proposed preconditioners allows for a natural extension to the treatment of *stiff* problems. Similar (implicit) parallel iteration methods can be designed that fastly converge to the corrector solution. In a forthcoming paper, we shall report on the performance of these parallel iteration methods applied to stiff problems.

2. Parallel iteration methods

We shall study parallel iterative methods for solving the stage vector equation in the s-stage Butcher–Kuntzmann method

$$\begin{aligned} &Y = e \otimes y_n + h(A \otimes I_d)F(Y), \\ &y_{n+1} = y_n + h(b^{\mathrm{T}} \otimes I_d)F(Y) = y_n + (b^{\mathrm{T}}A^{-1} \otimes I_d)(Y - e \otimes y_n). \end{aligned} \tag{2.1}$$

Here, Y is the sd-dimensional stage vector with s vector components Y_i of dimension d, $F(Y)$ is the sd-dimensional vector $(f(Y_i))$, $i = 1, 2, \ldots, s$, b and e are s-dimensional vectors, A is a nonsingular $s \times s$ matrix, I_d is the $d \times d$ identity matrix, and $\otimes$ denotes the Kronecker product. The vector e is the s-dimensional vector with unit entries, and b and A contain the Gauss–Legendre quadrature weights. Introducing the residual function

$$R_n(h, Y) := Y - e \otimes y_n - h(A \otimes I_d)F(Y), \tag{2.2a}$$

the iteration methods investigated in this paper fit into the following family of methods:

$$Y^{(j+1)} = Y^{(j)} - P_j R_n(h, Y^{(j)}) + h^2 Q_j(Y^{(j)} - Y^{(j-1)}), \quad j = 0, 1, \ldots, m-1, \tag{2.2b}$$

where $Y^{(-1)} = Y^{(0)}$ is a given initial iterate, and P_j and Q_j are $sd \times sd$ matrices whose entries may depend on the stepsize h and on the Jacobian matrix $J_n = \partial f(y_n)/\partial y$. It will be assumed that P_j and Q_j are bounded with respect to h and J_n. Evidently, if (2.2b) converges, then it converges to the stage vector Y. The s stage vector components of $Y^{(j+1)}$ defined by (2.2b) can be evaluated in parallel provided that s processors are available. Hence, the sequential computational effort per iteration does not depend on s.

After each iteration, we define the step point values

$$\begin{aligned} &y^{(j+1)} = y_n + (b^{\mathrm{T}}A^{-1} \otimes I_d)(Y^{(j+1)} - e \otimes y_n), \quad j = 0, 1, \ldots, m-1, \\ &y_{n+1} = y^{(m)}, \end{aligned} \tag{2.2c}$$

where the step value $y_{n+1} = y^{(m)}$ denotes the accepted approximation to the corrector solution at t_{n+1}.

If $P_j = I_{sd}$, and $Q_j = O$, then the iteration method (2.2) reduces to functional iteration. For Runge–Kutta correctors, such iteration methods were studied in [5,6,8,9,12]. The matrices P_j in front of the residual function R_n may be considered as preconditioning matrices. Together with Q_j, these matrices will be used for improving the damping of the iteration error components (see Sections 3 and 4).

In order to analyse the convergence of (2.2) we define the stage vector iteration error

$$\varepsilon^{(j)} := Y^{(j)} - Y,$$

and we write (2.2b) in the form

$$\varepsilon^{(j+1)} = [I_{sd} - P_j]\varepsilon^{(j)} + hP_j(A \otimes I_d)[F(Y^{(j)}) - F(Y)] + h^2 Q_j(\varepsilon^{(j)} - \varepsilon^{(j-1)}). \tag{2.2b'}$$

For sufficiently smooth right-hand side functions f we have

$$F(U + \delta) - F(U) = J(U)\delta + O(\delta^2),$$

where $J(U)$ is an $sd \times sd$ block-diagonal matrix whose diagonal blocks consist of the Jacobian matrices $\partial f(U_i)/\partial y$, U_i being the components of U. On substitution into (2.2b′) we straightforwardly derive the error recursion

$$\begin{aligned} &\varepsilon^{(j+1)} = (I_{sd} - P_j C + h^2 Q_j)\varepsilon^{(j)} - h^2 Q_j \varepsilon^{(j-1)} + O\big((\varepsilon^{(j)})^2\big), \\ &C := I_{sd} - h(A \otimes I_d) J(Y), \quad j = 0, \ldots, m-1. \end{aligned} \tag{2.3}$$

Suppose that $Q_j = O$ for all *even* values of j, and let us define the iteration matrices

$$Z_j := \begin{cases} I_{sd} - (P_j + P_{j-1})C + (P_j C - h^2 Q_j) P_{j-1} C, & j \text{ odd}, \\ Z_j := I_{sd} - P_j C, & j \text{ even}. \end{cases} \tag{2.4a}$$

Then we obtain

$$\varepsilon^{(j+1)} = \begin{cases} Z_j \varepsilon^{(j-1)} + O\big((\varepsilon^{(j)})^2\big) + O\big((\varepsilon^{(j-1)})^2\big), & j \text{ odd}, \\ Z_j \varepsilon^{(j)} + O\big((\varepsilon^{(j)})^2\big), & j \text{ even}. \end{cases} \tag{2.4b}$$

Let us define the *error amplification matrix*

$$H_m := \begin{cases} Z_{m-1} \cdot Z_{m-3} \cdot Z_{m-5} \cdots Z_3 \cdot Z_1, & m \text{ even}, \\ Z_{m-1} \cdot Z_{m-2} \cdot Z_{m-4} \cdots Z_3 \cdot Z_1, & m \text{ odd}. \end{cases} \tag{2.5a}$$

Then, neglecting higher-order terms, the iteration error satisfies

$$\varepsilon^{(m)} = H_m \varepsilon^{(0)}. \tag{2.5b}$$

In order to compare the rate of convergence of the iteration scheme (2.2) for various predictor–corrector pairs, we consider the iteration error at the step points. To that end, we write the step point formula defined in (2.2c) in the form

$$\begin{aligned} y_{n+1} &= y_n + (b^{\mathrm{T}} A^{-1} \otimes I_d)(Y^{(m)} - e \otimes y_n) \\ &= y_n + (b^{\mathrm{T}} A^{-1} \otimes I_d)(Y - e \otimes y_n) + (b^{\mathrm{T}} A^{-1} \otimes I_d)\varepsilon^{(m)}. \end{aligned}$$

Let us introduce the (exact) corrector solution

$$u_{n+1} := y_n + (b^{\mathrm{T}} A^{-1} \otimes I_d)(Y - e \otimes y_n), \tag{2.6}$$

and let us anticipate (see Theorems 3.1 and 4.1) that H_m can be written in the form

$$H_m = h^{\theta m}[K_m \otimes L_m + \Delta H_m], \quad \theta \geqslant 1, \tag{2.7a}$$

where the $s \times s$ matrix K_m is determined by the corrector matrix A, the $d \times d$ matrix L_m by $J(Y)$, and where ΔH_m denotes an $sd \times sd$ matrix which vanishes for linear problems. Using (2.5b) and (2.7a), the iteration error at the step point t_{n+1} reads

$$\begin{aligned} y_{n+1} - u_{n+1} &= (b^{\mathrm{T}} A^{-1} \otimes I_d)\varepsilon^{(m)} \\ &= h^{\theta m}[b^{\mathrm{T}} A^{-1} K_m \otimes L_m + (b^{\mathrm{T}} A^{-1} \otimes I_d)\Delta H_m]\varepsilon^{(0)}. \end{aligned} \tag{2.7b}$$

We now assume that the predictor formula is based on extrapolation of y_n and on preceding stage values, i.e.,

$$Y^{(0)} = w \otimes y_n + (E \otimes I_d)X, \tag{2.8}$$

where X is the stage vector computed in the preceding step, and the vector w and the matrix E contain the extrapolation weights defining the predictor. Notice that—at least formally—this predictor transforms the method into a multistep format. However, the influence of the "history" is very mild in this set-up; firstly, because only one previous step is involved, and, more importantly, because the basic formula that we apply (i.e., the corrector) is still of one-step nature. This precludes all the difficulties in stepsize variation that are usually encountered when a "real" multistep method is applied (for example, a predictor–corrector method of Adams type where also the corrector is of multistep form).

Next we want to obtain an expression for the *local* behaviour of the iteration error in terms of the locally exact solution through (t_n, y_n). This is the standard approach in (one-step) Runge–Kutta theory. However, since the predictor (2.8) is of multistep type, we shall make the additional assumption that the numerical stage vector X is on this locally exact solution. Then we can formulate the following theorem:

Theorem 2.1. *Let the error amplification matrix H_m be written in the form* (2.7a), *let the stage order of the corrector* (2.1) *be r, and define the vectors*

$$c := Ae, \qquad v_0 := w + Ee - e, \qquad v_j := \frac{1}{j!}E(c-e)^j - \frac{1}{(j-1)!}Ac^{j-1}, \quad j \geqslant 1. \tag{2.9}$$

If $v_j = 0$ for $j = 0, \ldots, q$ with $q \leqslant r$, then the iteration error at the step points is given by

$$y_{n+1} - u_{n+1} = h^{\theta m+q+1}\left[C_m L_m y^{(q+1)}(t_n) + O(\Delta H_m) + O(h)\right],$$
$$C_m := b^{\mathrm{T}}A^{-1}K_m v_{q+1}.$$

Proof. Let $y(t)$ denote the locally exact solution at the point t_n and let the stage vector X in (2.8) satisfy this solution. On substitution of $y(t)$ into the right-hand side of (2.8), the predictor formula reads

$$Y^{(0)} = w \otimes y(t_n) + (E \otimes I_d)y(t_{n-1}e + hc),$$

where $y(t_{n-1}e + hc)$ is defined by its components $y(t_{n-1} + hc_i)$, $i = 1, 2, \ldots, s$ (componentwise notation). Furthermore, by expressing the stage vector in the right-hand side of the stage vector equation in (2.1) in terms of the exact solution, and again using componentwise notation, we obtain

$$\begin{aligned} Y &= e \otimes y_n + h(A \otimes I_d)F(Y) \\ &= e \otimes y(t_n) + h(A \otimes I_d)F(y(t_n e + hc) + O(h^{r+1})) \\ &= e \otimes y(t_n) + h(A \otimes I_d)y'(t_n e + hc) + O(h^{r+2}). \end{aligned}$$

Taylor expansion of $Y^{(0)}$ and Y yields

$$Y^{(0)} = (w + Ee) \otimes y(t_n) + (E \otimes I_d)\big(h(c-e) \otimes y'(t_n) + \frac{1}{2!}h^2(c-e)^2 \otimes y''(t_n) + \cdots\big),$$
$$Y = e \otimes y(t_n) + (A \otimes I_d)\big(he \otimes y'(t_n) + h^2 c \otimes y''(t_n) + \frac{1}{2!}h^3 c^2 \otimes y'''(t_n) + \cdots\big) + O(h^{r+2}).$$

Thus, in terms of the vectors v_j, the predictor error is given by

$$\varepsilon^{(0)} = Y^{(0)} - Y = \sum_{j=0} v_j h^j \otimes y^{(j)}(t_n) + O(h^{r+2}).$$

The proof is completed by substitution of this expression into (2.7b) and taking into account the conditions of the theorem. □

Notice that the Butcher–Kuntzmann correctors allow for a prediction of order s (i.e., $q = s$), since they are based on collocation points in the open interval $(0, 1)$. For Radau IIA and Lobatto correctors, which all have $c_s = 1$, (2.8) allows for predictions of at most order $s - 1$, since for such methods, y_n is already contained in X.

The constant C_m may be interpreted as the *principal iteration error constant* after m iterations.

3. Spectral fitting

We shall determine the preconditioning matrices P_j and Q_j such that, for the test equation $y'(t) = \lambda y(t)$, the error amplification matrix H_m defined in (2.5) vanishes at m prescribed points $\{\lambda_k : k = 1, 2, \ldots, m\}$ in the complex λ-plane for all values of h. The iteration method will be said to be fitted at the points λ_k.

We remark that in [11], spectral fitting in *real* intervals $[a, b]$ has been considered. Since in that paper only one-step iteration processes were considered ($Q_j = O$ for all j), it was not possible to achieve spectral fitting at complex points. By introducing the two-step iteration method (2.2), it is possible to achieve spectral fitting at points that are either on the real axis or complex conjugate.

3.1. The preconditioner

In this section, we consider preconditioners completely determined by the m fitting points $\{\lambda_k\}$ and the matrix A.

Theorem 3.1. *Let S_m be the polynomial of degree m defined by*

$$S_m(x) = \begin{cases} (\pi_1 - \sigma_1 x + x^2)(\pi_3 - \sigma_3 x + x^2)\cdots(\pi_{m-1} - \sigma_{m-1}x + x^2), & m \text{ even}, \\ (\pi_1 - \sigma_1 x + x^2)(\pi_3 - \sigma_3 x + x^2)\cdots(\pi_{m-2} - \sigma_{m-2}x + x^2)(x - \lambda_m), & m \text{ odd}, \end{cases} \tag{3.1}$$

where σ_j, π_j and λ_m are real coefficients, and let the matrices P_j and Q_j be defined by the expressions

$$\begin{aligned} &P_{j-1} = I_{sd}, \quad P_j = \big(I_s - \sigma_j hA + \pi_j h^2 A^2\big)^{-1} \otimes I_d, && j \text{ odd}, \quad 1 \leqslant j \leqslant m-1, \\ &Q_{j-1} = O, \quad Q_j = -\pi_j P_j (A^2 \otimes I_d), && \\ &P_{m-1} = \big(I_s - h\lambda_m A\big)^{-1} \otimes I_d, \quad Q_{m-1} = O, && m \text{ odd}. \end{aligned} \tag{3.2}$$

Then, the error amplification matrix H_m is given by

$$H_m = [S_m(h^{-1}A^{-1})]^{-1} \otimes S_m(J_n) + O(h^{m+1}\Delta J_n), \qquad J_n := \frac{\partial f(y_n)}{\partial y}, \tag{3.3}$$

where ΔJ_n vanishes if J_n does not depend on y_n.

Proof. The matrix C defined in (2.3) can be written as

$$C = I_{sd} - h(A \otimes I_d)[(I_s \otimes J_n) + h\Delta J_n], \qquad J_n := \frac{\partial f(y_n)}{\partial y}, \tag{3.4}$$

where ΔJ_n is the block-diagonal matrix $h^{-1}[J(Y) - (I_s \otimes J_n)]$ which is bounded as $h \to 0$ and vanishes if J_n does not depend on y_n. Since we assumed $P_{j-1} = I_{sd}$ for odd values of j less than m, it follows from (2.4a) that

$$Z_j = h^2\left(-Q_j + M_j[(I_s \otimes J_n) + O(h\Delta J_n)] + P_j(A^2 \otimes I_d)[(I_s \otimes J_n^2) + O(h\Delta J_n)]\right),$$
$$M_j := h^{-1}(I_{sd} - P_j + h^2 Q_j)(A \otimes I_d), \quad j \text{ odd}, \quad 1 \leqslant j \leqslant m-1.$$

It is easily verified that the relations (3.2) imply that

$$Z_j = h^2 P_j(A^2 \otimes [\pi_j I_d - \sigma_j J_n + J_n^2]) + O(h^3 \Delta J_n), \quad j \text{ odd}, \quad 1 \leqslant j \leqslant m-1. \tag{3.5a}$$

For even values of j we derive

$$Z_j := I_{sd} - P_j(I_{sd} - h(A \otimes I_d)(I_s \otimes J_n)) + O(h^2 \Delta J_n).$$

From (3.2) it follows that

$$I_{sd} = P_{m-1}(I_{sd} - h(A \otimes \lambda_m I_d)), \quad m \text{ odd},$$

hence, the iteration matrix Z_{m-1} takes the form

$$Z_{m-1} = hP_{m-1}(A \otimes [J_n - \lambda_m I_d]) + O(h^2 \Delta J_n), \quad m \text{ odd}. \tag{3.5b}$$

The relations (3.5), together with (3.1) and (3.2) lead to (3.3). Finally, in the case of constant Jacobian matrices, the order term $O(h^{m+1}\Delta J_n)$ vanishes. □

If the method defined by (2.2) and (3.2) is applied to the test equation $y'(t) = \lambda y(t)$, then this theorem shows that H_m vanishes for all zeros of the *spectral fitting polynomial* S_m. Since the zeros of S_m can be chosen arbitrarily, we can achieve spectral fitting at any prescribed set of m fitting points $\{\lambda_k\}$. Ideally, these zeros should be chosen in the region where $J_n = \partial f(y_n)/\partial y$ has its spectrum and such that $S_m(J_n)$ is minimal on the spectrum of J_n (*spectral fitting*). The resulting method will be denoted by PIRK$\{m, \lambda_k\}$ (Parallel Iterated Runge–Kutta method with m fitting points $\{\lambda_k\}$).

A comparison with (3.3) reveals that H_m is of the form (2.7a) with $q = 1$, $K_m = A^m$, and $L_m = S_m(J_n)$. By means of Theorem 2.1 we can prove:

Theorem 3.2. *Let the conditions of Theorem 2.1 be satisfied. Then the iteration error of the PIRK$\{m, \lambda_k\}$ method is given by*

$$\begin{aligned} &y_{n+1} - u_{n+1} = C_m S_m(J_n) h^{m+q+1} y^{(q+1)}(t_n) + O(h^{m+q+2}), \\ &C_m = b^T A^{m-1} v_{q+1}. \end{aligned} \tag{3.6}$$

Expression (3.6) shows that the convergence accelerating effect of the spectral fitting polynomial S_m and of the underlying predictor–corrector pair are more or less factorized, so that the determination of an appropriate fitting polynomial S_m (see Section 5) can be addressed independently of the choice of the predictor–corrector arrays A, b, c, w and E.

We shall consider the principal iteration error constant C_m associated to the

Last-step-value predictor:
(2.8) with $E = O$, $w = e$, (3.7a)

Last-stage-vector predictor:
$\{(2.8), (2.9)\}$ with $v_j = 0$, $j = 0, \ldots, r$, (3.7b)
r being the stage order of the corrector.

Theorem 3.3. *For the Last-step-value predictor* (3.7a), *the iteration error of the PIRK*$\{m, \lambda_k\}$ *method is given by*

$$\begin{aligned} &y_{n+1} - u_{n+1} = C_m S_m(J_n) h^{m+1} y'(t_n) + O(h^{m+2}), \\ &C_m := -b^T A^{m-1} c, \end{aligned} \tag{3.6a}$$

where $C_m = -1/(m+1)!$ *for* $m \leqslant p-1$, p *denoting the order of the corrector.*

Proof. If $E = O$, then $q = 0$, and it is easily verified that (3.6) reduces to (3.6a). The relation $b^T A^{m-1} c = 1/(m+1)!$ for $m \leqslant p-1$ follows from the order conditions for Runge–Kutta methods. □

For a number of Butcher–Kuntzmann correctors, Table 1 lists the values by which the principal iteration error constant C_m is reduced in each iteration (i.e., the values of C_m/C_{m-1} where $C_0 := -b^T A^{-1} c$ corresponds to the predictor error). These values show that—for the first few iterations—the choice of the corrector is irrelevant; low-order and high-order correctors will generate iterations

Table 1
Values of C_m/C_{m-1} for Butcher–Kuntzmann correctors using the Last-step-value predictor

p	s	C_0	m											
			1	2	3	4	5	6	7	8	9	10	...	∞
2	1	−1	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	...	1/2
4	2	−1	1/2	1/3	1/4	1/6	0	∞	1/2	1/3	1/4	1/6	...	0.29
6	3	−1	1/2	1/3	1/4	1/5	1/6	3/20	1/6	7/30	2/7	11/40	...	0.22
8	4	−1	1/2	1/3	1/4	1/5	1/6	1/7	1/8	23/210	2/23	1/56	...	0.17
10	5	−1	1/2	1/3	1/4	1/5	1/6	1/7	1/8	1/9	1/10	23/252	...	0.14

Table 2
Values of C_m/C_{m-1} for Butcher–Kuntzmann correctors using the Last-stage-vector predictor

p	s	C_0	m											
			1	2	3	4	5	6	7	8	9	10	...	∞
2	1	0.75	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	...	0.50
4	2	0.36	0.31	0.23	0.14	0.11	1.25	0.43	0.31	0.23	0.14	0.11	...	0.29
6	3	0.13	0.25	0.21	0.18	0.16	0.17	0.22	0.26	0.27	0.24	0.22	...	0.22
8	4	0.038	0.22	0.18	0.16	0.14	0.12	0.10	0.07	0.10	0.64	0.29	...	0.17
10	5	0.009	0.19	0.16	0.14	0.13	0.11	0.10	0.10	0.09	0.11	0.14	...	0.14

with an equal iteration error. This implies that high-order correctors are also appropriate for generating low-order results. Note that C_m/C_{m-1} converges to the spectral radius of A as m tends to infinity. From this table we observe a periodic behaviour (with period of length 6) for the fourth-order Butcher–Kuntzmann method. This can easily be explained by observing that for this method $A^6 = \gamma I_s$ (with $\gamma = -1/1728$).

Theorem 3.4. *Let U and V be $s \times s$ matrices whose columns are respectively given by the vectors $\{jc^{j-1},\ j = 1, \ldots, s\}$ and $\{(c-e)^j,\ j = 1, \ldots, s\}$, and let V be nonsingular and $E = AUV^{-1}$. Then, for the Last-stage-vector predictor* (3.7b) *with $E = AUV^{-1}$ and $w = e - Ee$, the iteration error of the PIRK$\{m, \lambda_k\}$ method is given by*

$$y_{n+1} - u_{n+1} = C_m S_m(J_n) h^{m+r+1} y^{(r+1)}(t_n) + O(h^{m+r+2}),$$
$$C_m := \frac{1}{(r+1)!} b^T A^m [UV^{-1}(c-e)^{r+1} - (r+1)c^r]. \tag{3.6b}$$

Proof. From Theorem 2.1 it follows that we can always achieve $q = r$ if w and E satisfy the relations

$$w = e - Ee, \qquad E(c-e)^j = jAc^{j-1}, \quad j = 1, \ldots, s. \tag{3.8}$$

It is easily verified that E can be represented in the form $E = AUV^{-1}$ with U and V as defined in the theorem. From the Theorems 2.1 and 3.2 the expression (3.6b) is readily obtained. □

The analogue of Table 1 is given by Table 2. Taking into account the rather small values of C_0, this table clearly shows that the Last-stage-vector predictor should converge faster than the Last-step-value predictor. It also shows that it is now more efficient to generate low-order results by high-order correctors than by low-order correctors.

3.2. Stability

Next, we address the stability of PIRK$\{m, \lambda_k\}$ methods. Confining our considerations to the Last-step-value predictor, we have the following theorem:

Table 3
Stability boundaries $\{\beta_{\text{real}}, \beta_{\text{imag}}\}$ for PIRK$\{m, \lambda_k\}$ methods with Butcher–Kuntzmann correctors

p	$m = p$	$m = p + 2$	$m = p + 4$
4	{2.78, 2.82}	{3.54, 3.46}	{3.01, 0.6*}
6	{3.55, 0.5*}	{3.99, 3.68}	{3.89, 1.0*}
8	{4.31, 3.39}	{5.27, 1.5*}	{4.99, 1.6*}

Theorem 3.5. *Let the predictor be defined by the Last-step-value predictor* (3.7a) *and let* S_m *be defined by* (3.1). *Then, for the test equation* $y' = \lambda y$, *the stability polynomial of the PIRK*$\{m, \lambda_k\}$ *method is given by*

$$R_m(z,h) = 1 + \beta_1(h)z + \beta_2(h)z^2 + \ldots + \beta_m(h)z^m,$$
$$\beta_j(h) := \frac{1}{j!}\boldsymbol{b}^{\mathrm{T}}\frac{\partial^j M}{\partial z^j}(0,h)\boldsymbol{e}, \quad z := h\lambda,$$
$$M(z,h) := z\left[I + zA + z^2A^2 + \cdots\right]\left[I_s - S_m(h^{-1}z)\left[S_m(h^{-1}A^{-1})\right]^{-1}\right].$$

Proof. For the test equation $y' = \lambda y$, we derive from (2.7), (3.3), (2.8) and (2.1)

$$\begin{aligned} y_{n+1} &= u_{n+1} + (\boldsymbol{b}^{\mathrm{T}}A^{-1}K_m \otimes L_m)(\boldsymbol{Y}^{(0)} - \boldsymbol{Y}) \\ &= \left[1 + h\lambda\boldsymbol{b}^{\mathrm{T}}(I - h\lambda A)^{-1}\boldsymbol{e}\right]y_n + S_m(\lambda)\boldsymbol{b}^{\mathrm{T}}A^{-1}\left[S_m(h^{-1}A^{-1})\right]^{-1}\left[I - (I - h\lambda A)^{-1}\right]\boldsymbol{e}y_n. \end{aligned}$$

Defining the stability polynomial $R_m(\lambda, h)$ according to

$$y_{n+1} = R_m(z,h)y_n, \qquad z := h\lambda,$$

it follows that

$$R_m(z,h) = 1 + z\boldsymbol{b}^T(I - zA)^{-1}\left[I_s - S_m(h^{-1}z)\left[S_m(h^{-1}A^{-1})\right]^{-1}\right]\boldsymbol{e}.$$

If zA has its eigenvalues within the unit circle, we may write

$$R_m(z,h) = 1 + \boldsymbol{b}^T M(z,h)\boldsymbol{e},$$
$$M(z,h) := z\left[I + zA + z^2A^2 + \cdots\right]\left[I_s - S_m(h^{-1}z)\left[S_m(h^{-1}A^{-1})\right]^{-1}\right].$$

Since $R_m(z,h)$ necessarily is a polynomial of degree m in z, the assertion of the theorem follows. □

Let us first consider conventional predictor–corrector iteration where all fitting points are at the origin, i.e., $S_m(x) = x^m$. Then, Theorem 3.5 shows that the stability function takes the form

$$R_m(z,h) = 1 + z + \frac{1}{2!}z^2 + \cdots + \frac{1}{p!}z^p + \beta_{p+1}z^{p+1} + \cdots + \beta_m z^m, \quad \beta_j = \boldsymbol{b}^{\mathrm{T}}A^{j-2}\boldsymbol{c}. \tag{3.9}$$

For a few values of p and m, the real and imaginary stability boundaries $\{\beta_{\text{real}}, \beta_{\text{imag}}\}$ are given in Table 3. The three-digit numbers refer to stability intervals where $|R_m|$ does not exceed 1, whereas, in the case of β_{imag}, the two-digit numbers (carrying an $*$) refer to imaginary stability intervals

where $|R_m|$ is less than $1 + 10^{-6}$. Although the latter cases formally indicate zero imaginary stability boundaries, the given values may be considered to be effective boundaries.

For nonzero fitting points, the polynomial (3.9) is approximated as $h \to 0$. However, if $h \neq 0$, this polynomial, and hence the stability region, may change considerably. Although we did not introduce the preconditioners for improving stability, we want to remark that they can also be used to relax the stability conditions.

Example 3.6. As an illustration, we derive the stability polynomial for the case where $p = m = 2$ and

$$S_2(x) = (x - \lambda_0 I_s)^2,$$

for λ_0 a real, fixed parameter. The corresponding stability polynomial is given by

$$\begin{aligned} R_2(z,h) = {} & 1 + z\left[1 - h^2\lambda_0^2 \boldsymbol{b}^{\mathrm{T}}(I_s - h\lambda_0 A)^{-2}A^2\boldsymbol{e}\right] \\ & + \tfrac{1}{2}z^2\left[1 + 2h\lambda_0 \boldsymbol{b}^{\mathrm{T}}(2I_s - h\lambda_0 A)(I_s - h\lambda_0 A)^{-2}A^2\boldsymbol{e}\right]. \end{aligned}$$

For $h \to 0$, the stability region of this polynomial converges to that of the stability polynomial of Runge's method. However, if $h \neq 0$, the stability region changes considerably. For example, let the corrector be defined by the one-stage Butcher–Kuntzmann method with $A = \frac{1}{2}$ and $\boldsymbol{b} = 1$. Then,

$$R_2(z,h) = 1 + z\frac{1 - h\lambda_0}{(1 - h\lambda_0/2)^2} + \frac{1}{2}z^2\frac{1}{(1 - h\lambda_0/2)^2}.$$

If z is negative, then $R_2(z,h)$ assumes values in $[-1,+1]$ for $h\lambda_0 \leqslant \frac{3}{2}$ and $-2(1 - h\lambda_0) \leqslant z \leqslant 0$ Let $\rho(J_n)$ denote the spectral radius of the Jacobian J_n. For $\lambda_0 \leqslant -\frac{1}{2}\rho(J_n)$, we have unconditiona. stability. For $\lambda_0 > -\frac{1}{2}\rho(J_n)$, the stability condition becomes $h \leqslant 2[2\lambda_0 + \rho(J_n)]^{-1}$. From this expression we conclude that for negative λ_0, the stability condition is always less stringent than that of Runge's method. For imaginary values of z, the stability condition takes the form $\lambda_0 > 0$ and $h \leqslant 4\lambda_0[3\lambda_0^2 + \rho(J_n)^2]^{-1}$, whereas Runge's method is always unstable. □

4. Spectral fitting using the Jacobian matrix

In this section, we construct preconditioners by using the Jacobian matrix of the IVP. This enables us to achieve convergence factors of $O(h^2)$, and at the same time, to apply spectral fitting at $2m$ points in the complex plane (m being the total number of iterations). The use of the Jacobian matrix is quite uncommon in nonstiff problem solvers. However, in many problems, the increased rate of convergence and the observation that an update of the entries of the Jacobian is probably needed only once in a few steps and, moreover, can be done in parallel, justifies the use of the Jacobian-dependent preconditioners in nonstiff problems.

4.1. The preconditioner

The analogue of Theorem 3.1 reads:

Theorem 4.1. *Let ΔJ_n be defined as in Theorem 3.1, let S_{2m}^* be the polynomial of degree $2m$ defined by*

$$S_{2m}^*(x) = (\pi_0 - \sigma_0 x + x^2)(\pi_1 - \sigma_1 x + x^2)\cdots(\pi_{m-1} - \sigma_{m-1}x + x^2), \tag{4.1}$$

where σ_j and π_j are real coefficients, and let the matrices P_j and Q_j be defined by the expressions

$$\begin{aligned} &P_j = I_{sd} - \pi_j h^2 V_j(h) \otimes I_d + hV_j(h)A^{-1} \otimes J_n, \\ &Q_j = O, \\ &V_j(h) = A^2(I_s - \sigma_j hA + \pi_j h^2 A^2)^{-1}, \\ &\quad j = 0, 1, \ldots, m-1, \quad J_n := \frac{\partial f(y_n)}{\partial y}. \end{aligned} \tag{4.2}$$

Then, the error amplification matrix H_m is given by

$$H_m(h, J_n) = [S_{2m}^*(h^{-1}A^{-1})]^{-1} \otimes S_{2m}^*(J_n) + O(h^{2m}\Delta J_n), \quad J_n := \frac{\partial f(y_n)}{\partial y}. \tag{4.3}$$

Proof. Using that all Q_j vanish and substituting (3.4), the matrix Z_j defined in (2.4a) takes the factorized form

$$\begin{aligned} &Z_j = \begin{cases} W_j(h, J_n)W_{j-1}(h, J_n), & j \text{ odd}, \\ W_j(h, J_n), & j \text{ even}, \end{cases} \\ &W_j(h, J_n) := I_{sd} - P_j(I_{sd} - h(A \otimes I_d)[(I_s \otimes J_n) + h\Delta J_n]). \end{aligned} \tag{4.4}$$

On substitution of (4.2) into W_j we obtain

$$\begin{aligned} &W_j(h, J_n) \\ &\quad = \pi_j h^2 V_j(h) \otimes I_d + h(A - V_j(h)A^{-1} - \pi_j h^2 V_j(h)A) \otimes J_n \\ &\qquad + h^2 V_j(h) \otimes J_n^2 + O(h^2 \Delta J_n) \\ &\quad = \pi_j h^2 V_j(h) \otimes I_d - \sigma_j h^2 V_j(h) \otimes J_n + h^2 V_j(h) \otimes J_n^2 + O(h^2 \Delta J_n) \\ &\quad = h^2(A^2(I_s - \sigma_j hA + \pi_j h^2 A^2)^{-1} \otimes (J_n^2 - \sigma_j J_n + \pi_j I_d)) + O(h^2 \Delta J_n), \\ &j = 0, 1, 2, \ldots, m-1. \end{aligned}$$

By substitution of $W_j(h, J_n)$ into (4.4) and using the spectral fitting polynomial (4.1), the matrix H_m can be written in the form (4.3). □

Like the PIRK$\{m, \lambda_k\}$ method of the preceding section, the method defined by (2.2) and (4.2), when applied to the test equation $y'(t) = \lambda y(t)$, has the property that H_m vanishes for all zeros of the *spectral fitting polynomial* S_{2m}^*. However, firstly, we now have $2m$ fitting points $\{\lambda_k\}$, and secondly, in each iteration we have a factor h^2 instead of a factor h. On the other hand, for IVPs where ΔJ_n does not vanish, the eigenvalues of the error multiplication matrix H_m are always $O(h^{2m}\Delta J_n)$ including eigenvalues that coincide with the fitting points. In this connection, it should be remarked that Theorem 4.1 also applies to nonautonomous IVPs provided that J_n and ΔJ_n are understood to correspond to the Jacobian of the nonautonomous right-hand side function. As a consequence,

for linear nonautonomous systems, the eigenvalues of the error multiplication matrix vanish if they coincide with the fitting points.

The method defined by (2.2) and (4.2) will be denoted by PIRKJ$\{2m, \lambda_k\}$ (Parallel Iterated Runge–Kutta method using the Jacobian matrix and $2m$ fitting points $\{\lambda_k\}$). The analogues of the Theorems 3.2–3.5 become:

Theorem 4.2. *Let the conditions of Theorem 2.1 be satisfied. Then the iteration error of the PIRKJ$\{2m, \lambda_k\}$ method is given by*

$$\begin{aligned} &y_{n+1} - u_{n+1} = h^{2m+q+1}[C_m S^*_{2m}(J_n) y^{(q+1)}(t_n) + O(\Delta J_n) + O(h)], \\ &C_m := b^T A^{2m-1} v_{q+1}. \end{aligned} \tag{4.5}$$

Theorem 4.3. *For the Last-step-value predictor* (3.7a), *the iteration error of the PIRKJ$\{2m, \lambda_k\}$ method is given by*

$$\begin{aligned} &y_{n+1} - u_{n+1} = C_m S^*_{2m}(J_n) h^{2m+1} y'(t_n) + O(h^{2m+1}\Delta J_n) + O(h^{2m+2}), \\ &C_m := -b^T A^{2m-1} c, \end{aligned}$$

where $C_m = -1/(2m+1)!$ for $2m \leqslant p-1$, p denoting the order of the corrector.

Theorem 4.4. *Let U and V be $s \times s$ matrices whose columns are respectively given by the vectors $\{jc^{j-1},\ j = 1, \ldots, s\}$ and $\{(c-e)^j,\ j = 1, \ldots, s\}$, and let V be nonsingular and $E = AUV^{-1}$. Then, for the Last-stage-vector predictor* (3.7b) *with $E = AUV^{-1}$ and $w = e - Ee$, the iteration error of the PIRKJ$\{2m, \lambda_k\}$ method is given by*

$$\begin{aligned} &y_{n+1} - u_{n+1} = C_m S^*_{2m}(J_n) h^{2m+r+1} y^{(r+1)}(t_n) + O(h^{2m+r+1}\Delta J_n) + O(h^{2m+r+2}), \\ &C_m := \frac{1}{(r+1)!} b^T A^{2m}[UV^{-1}(c-e)^{r+1} - (r+1)c^r]. \end{aligned} \tag{4.6}$$

Theorem 4.5. *Let the predictor be defined by the Last-step-value predictor* (3.7a) *and let S^*_{2m} be defined by* (4.1). *Then, for the test equation $y' = \lambda y$, the stability polynomial of the PIRKJ$\{2m, \lambda_k\}$ method is given by*

$$\begin{aligned} &R^*_{2m}(z, h) = 1 + \beta_1(h) z + \beta_2(h) z^2 + \cdots + \beta_{2m}(h) z^{2m}, \\ &\beta_j(h) := \frac{1}{j!} b^T \frac{\partial^j M}{\partial z^j}(0, h) e, \quad z := h\lambda, \\ &M(z, h) := z[I + zA + z^2 A^2 + \cdots][I_s - S^*_{2m}(h^{-1}z)[S^*_{2m}(h^{-1}A^{-1})]^{-1}]. \end{aligned}$$

5. Spectral fitting polynomials

From Theorems 3.1 and 4.1 it follows that for small h, the best we can do is to minimize the spectral fitting polynomial in some sense over the eigenvalue spectrum of J_n. In particular, we consider eigenvalue spectra located on the line segment $[a, b]$ in the complex plane (e.g., intervals on the real axis or on the imaginary axis). It is easily shown that the maximum norm of the spectral

fitting polynomial can be minimized on $[a, b]$ by means of shifted Chebyshev polynomials. For the preconditioners (3.2), this minimax spectral fitting polynomial is given by (T_m denotes the first-kind Chebyshev polynomial of degree m)

$$S_m(z) = \frac{(b-a)^m}{2^{2m-1}} T_m\left(\frac{a+b-2z}{a-b}\right). \tag{5.1}$$

The minimax spectral fitting polynomial $S^*_{2m}(z)$ for preconditioner (4.2) can be obtained by setting $S^*_{2m}(z) = S_{2m}(z)$.

It should be remarked that only for intervals $[a, b]$ on the real axis or intervals with a and b complex conjugate, the coefficients of S_m and $S^*_{2m}(z)$ are real-valued. Furthermore, these Chebyshev-type fitting polynomials are not optimal for more general eigenvalue spectra. For example, if the eigenvalues of J_n are located in a circular sector, then the near-optimal polynomials are the Faber polynomials (cf. [2]). However, in this paper, we shall restrict our considerations to fitting polynomials of the form (5.1).

The parameters σ_j and π_j occurring in the preconditioners (3.2) and (4.2) can now easily be derived from the zeros of the spectral fitting polynomials by using the following expression for shifted Chebyshev polynomials of degree n:

$$\begin{aligned} &T_n\left(\frac{a+b-2x}{a-b}\right) = \frac{2^{2n-1}}{(b-a)^n} \prod_{k=1}^{n} (x - w_{nk}), \\ &w_{nk} = \frac{1}{2}\left[a+b-(a-b)\cos\left(\frac{(2k-1)\pi}{2n}\right)\right], \quad k = 1, \ldots, n. \end{aligned} \tag{5.2}$$

In order to get some insight in the magnitude of $S_m(J_n)$, we set $J_n = \lambda I_d$ and we write

$$\|S_m(\lambda)\|_R = |b-a|^m (\sigma_R)^m, \qquad \|S^*_{2m}(\lambda)\|_R = |b-a|^{2m} (\sigma^*_R)^m, \tag{5.3}$$

where $\|\cdot\|_R$ denotes the maximum norm over some region R in the complex λ-plane. In particular, we consider the two cases $a = -r$, $b = 0$ and $a = -ir$, $b = +ir$ for the sectorial regions

$$\begin{aligned} R_1(r, \phi) &:= \{\lambda = |\lambda| e^{i(\pi - \psi)} : 0 \leqslant |\lambda| \leqslant r, |\psi| \leqslant \phi\}, \\ R_2(r, \phi) &:= \{\lambda = |\lambda| e^{i(\pi/2 + \psi)} : 0 \leqslant |\lambda| \leqslant r, \ 0 \leqslant \psi \leqslant \phi\}. \end{aligned} \tag{5.4}$$

It is easily verified that

$$\begin{aligned} &\sigma_{R_1(1,\phi)} = \tfrac{1}{4}\sqrt[m]{2}\, \sqrt[m]{\|T_m(1+2\lambda)\|_{R_1(1,\phi)}}, \\ &\sigma^*_{R_1(1,\phi)} = \left(\sigma_{R_1(1,\phi)}\right)^2, \end{aligned} \quad \text{if } a = -r, \ b = 0, \tag{5.5a}$$

$$\begin{aligned} &\sigma_{R_2(1,\phi)} = \tfrac{1}{2}\sqrt[m]{2}\, \sqrt[m]{\|T_m(i\lambda)\|_{R_2(1,\phi)}}, \\ &\sigma^*_{R_2(1,\phi)} = \left(\sigma_{R_2(1,\phi)}\right)^2, \end{aligned} \quad \text{if } a = -ir, \ b = +ir. \tag{5.5b}$$

Thus, it suffices to compute the constants characterizing S_m. These constants are given in Table 4 for the region $R_1(r, \phi)$ with fitting points λ_k in $[-r, 0]$ and for the region $R_2(r, \phi)$ with fitting points λ_k in $[-ir, +ir]$. For $\phi > 0$, the constants rapidly converge to a fixed value as m increases. These values are substantially less than 1 for regions $R_j(r, \phi)$ with quite large apertures. Since

Table 4
Constants $\sigma_{R_j(1,\phi)}$ in formula (5.3) defined by (5.5)

m	Region $R_1(r,\phi)$ with fitting points λ_k in $[-r,0]$					Region $R_2(r,\phi)$ with fitting points λ_k in $[-ir,+ir]$				
	$\phi=0°$	$\phi=15°$	$\phi=30°$	$\phi=60°$	$\phi=90°$	$\phi=0°$	$\phi=15°$	$\phi=30°$	$\phi=60°$	$\phi=90°$
1	0.50	0.54	0.62	0.87	1.12	1.00	1.00	1.00	1.00	1.00
2	0.36	0.50	0.67	0.94	1.16	0.71	0.79	0.94	1.16	1.23
3	0.32	0.51	0.67	0.94	1.16	0.63	0.81	0.97	1.15	1.21
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
10	0.27	0.51	0.67	0.94	1.16	0.54	0.82	0.97	1.15	1.21
∞	0.25	0.51	0.67	0.94	1.16	0.50	0.82	0.97	1.15	1.21

the constants corresponding to the conventional fitting polynomial z^m are equal to 1 (for all m), we see that Chebyshev fitting polynomials with zeros on the real and imaginary axes yield smaller convergence factors for $\phi \leqslant 60°$ and $\phi \leqslant 30°$, respectively.

6. Numerical experiments

To test the efficiency of the iterated Butcher–Kuntzmann methods on parallel computers, we applied these methods to various test examples. An extensive discussion of these tests can be found in [13, Appendix]. Here we present results for one, representative, example, i.e., the Arenstorf orbit problem [4, p. 127]

$$
\begin{aligned}
&y_1' = y_3, && y_1(0) = 0.994,\\
&y_2' = y_4, && y_2(0) = 0,\\
&y_3' = y_1 + 2y_4 - \mu'\frac{y_1+\mu}{D_1} - \mu\frac{y_1-\mu'}{D_2}, && y_3(0) = 0,\\
&y_4' = y_2 - 2y_3 - \mu'\frac{y_2}{D_1} - \mu\frac{y_2}{D_2}, && y_4(0) = -2.001585106379082\ldots,\\
&\quad 0 \leqslant t \leqslant 17.06521656015796\ldots,\\
&\mu = 0.012277471, && \mu' = 1-\mu,\\
&D_1 = \left((y_1+\mu)^2 + y_2^2\right)^{3/2}, && D_2 = \left((y_1-\mu')^2 + y_2^2\right)^{3/2}.
\end{aligned}
\tag{6.1}
$$

For the underlying corrector we choose the Butcher–Kuntzmann method of order 8. Since it is also of interest to see the effect of the order of the corrector on the performance of the PIRK methods, we give, in addition, the results of a low-order corrector ($p = 4$) and of a corrector of very high order ($p = 26$). We remark that this last method is not advocated to be of practical interest; it is merely used to show the impact of a high-order corrector. For the predictor we choose the Last-stage-vector predictor (3.7b). The calculations are performed using 15-digit arithmetic.

A realistic application of the PIRK$\{m,0\}$ and PIRKJ$\{2m,0\}$ methods requires a dynamic variation of the stepsize h. Therefore, both versions are provided with an automatic stepsize selection which

Table 5
Comparison with DOPRI8 for the Arenstorf orbit problem (6.1)

Method	Order of corrector	Order of resulting method	Δ						Averaged speed-up w.r.t. DOPRI8
			3	4	5	6	7	8	
DOPRI8		8	1564	1900	2459	3215	3944	4817	
PIRK{3,0}	4	4	1753	3138	6270	11516	18875	39311	0.4
PIRKJ{4,0}	4	4	1237	1988	3249	5275	9529	15266	0.7
PIRK{5,0}	8	8	664	812	967	1191	1415	1809	2.6
PIRKJ{6,0}	8	8	403	483	588	698	831	963	4.4
PIRK{5,0}	26	18	450	523	545	619	696	852	4.7
PIRKJ{6,0}	26	19	291	338	373	407	467	760	6.7

we more or less copied from the code PIRK8 described in [12]. Here, we confine ourselves to a fixed number of iterations. Moreover, all fitting points are chosen at the origin. One reason is that, for the greater part of the integration interval, the eigenvalues for this problem are located in the neighbourhood of the origin; another, more important reason is that for this *nonlinear* problem, fitting does not help when it is applied in addition to preconditioning with the Jacobian (see also the discussion following Theorem 4.1). We use values of m that turned out to produce the best results (these values are indicated in the table of results). It is however to be expected that a variable-m strategy (which may result in a variable-order method) will enhance the performance of the codes. In passing, we remark that for the Arenstorf orbit problem, the variable stepsize implementation of PIRKJ{10,0} is 50 to 100 times more efficient than its fixed-h variant. This is due to the fact that in small parts of the integration interval the problem is (mildly) stiff, forcing the fixed-h version to use a small stepsize in the whole interval.

The accuracy is given by the number of correct digits Δ, obtained by writing the maximum norm of the absolute error at the endpoint in the form $10^{-\Delta}$. The computational effort is measured by the total number N of sequential right-hand side evaluations performed during the integration process.

For various values of the local error control parameter, we compared the iterated Butcher–Kuntzmann correctors with the DOPRI8 code given in [4] (DOPRI8 is based on the 8(7)-method of Prince and Dormand [10] which is nowadays considered as one of the most efficient sequential methods). For an easy comparison of the results we computed, for a number of given Δ-values, the corresponding N-values by linear interpolation. The results are listed in Table 5 showing that the high corrector order and Jacobian-dependent preconditioners improve the efficiency drastically.

From this experiment we can draw several conclusions. Let us first compare the PIRK-type methods of order 8 with the eighth-order DOPRI8 code. We observe an averaged speed-up of 2.6 for the PIRK variant and 4.4 for the preconditioned version (recall that these parallel methods require 4 processors). Hence, it is clear that the preconditioning considerably increases the efficiency. Also for the PIRK methods of low and high order, we see that the preconditioned versions are to be preferred.

Furthermore, we observe the significant role of the *order* of the underlying corrector. The fourth-order scheme is clearly of a too low order to compete with DOPRI8, especially in the high-accuracy range. The PIRK methods based on the corrector of order 26 on the other hand, show a speed-up which is even higher than obtained for the eighth-order PIRKs, also in the low-accuracy range. This observation gives evidence to our statement that, for nonstiff problems, a corrector of high order, in combination with a predictor based on extrapolation, is generally most efficient.

7. Summary

In this paper our starting point is a parallel iteration scheme for the approximate solution of the Butcher–Kuntzmann methods. For nonstiff problems, such algorithms have been proposed in [5,6,8,9,12]. These methods are "general-purpose methods" in the sense that they do not take into account any special knowledge about the problem.

In the present paper we analyze the effect on the convergence behaviour of these parallel iteration methods by incorporating some additional information; we distinguish two approaches:

- In the case that information on the spectrum of the Jacobian matrix is available, it is possible to considerably increase the rate of convergence by adapting the iteration parameters to this spectrum. It is shown that this technique is applicable both in the case of real eigenvalues and of complex (conjugate) pairs. An advantage of this approach is that the computational effort of the method is hardly increased by introducing these "fitting parameters". If no spectral information is available, then the best thing to do is fitting at the origin to obtain conventional functional iteration.
- A second approach is to use additional information which is provided by the Jacobian matrix. In this case we have shown that the convergence factor can be made of $O(h^2)$ instead of the usual behaviour of $O(h)$, which implies that roughly half the number of iterations are required. A disadvantage is the additional overhead due to Jacobian evaluations and matrix–vector multiplications. However, often an inaccurate Jacobian does not drastically degrade the performance, so that this matrix needs not be reevaluated in each integration step. Moreover, Jacobian evaluations possess a high degree of parallelism. The Jacobian approach has turned out to be useful in cases where a right-hand side evaluation is substantially more expensive than a Jacobian–vector multiplication. In such cases the additional work introduced in the iteration scheme is negligible. A clear advantage of this approach is that it does not require *a priori* information about the problem.

In this paper, we have focused on the application of the above convergence-acceleration techniques to *nonstiff* problems. Numerical examples have demonstrated (see also [13, Appendix]) the efficiency-increasing effect of both acceleration techniques in the context of fixed stepsizes. Finally, a variable-stepsize implementation of our preconditioned PIRK(J) methods is compared to the best sequential nonstiff solver and shown to be superior.

References

[1] J.C. Butcher, Implicit Runge–Kutta processes, *Math. Comp.* 18 (1964) 50–64.

[2] J.P. Coleman and R.A. Smith, The Faber polynomials for circular sectors, *Math. Comp.* 49 (1987) 81–84, 231–241.
[3] P.J. Davis and P. Rabinowitz, *Methods of Numerical Integration* (Academic Press, New York, 1975).
[4] E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations* I: *Nonstiff Problems*, Springer Series in Computational Mathematics 8 (Springer-Verlag, Berlin, 1987).
[5] K.R. Jackson, A. Kværnø and S.P. Nørsett, Order of Runge–Kutta methods when using Newton-type iteration, Technical Report No. 1/91, Division of Mathematical Sciences, University of Trondheim, Norway (1992).
[6] K.R. Jackson and S.P. Nørsett, The potential for parallelism in Runge–Kutta methods, Part I: RK formulas in standard form, Technical Report No. 239/90, Department of Computer Science, University of Toronto, Toronto, Ont. (1990).
[7] J. Kuntzmann, Neuere Entwicklungen der Methoden von Runge und Kutta, Z. Angew. Math. Mech. 41 (1961) T28–T31.
[8] I. Lie, Some aspects of parallel Runge–Kutta methods, Report 3/87, Department of Mathematics, University of Trondheim, Norway (1987).
[9] S.P. Nørsett and H.H. Simonsen, 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, 1989).
[10] P.J. Prince and J.R. Dormand, High order embedded Runge–Kutta formulae, *J. Comput. Appl. Math.* 7 (1981) 67–75.
[11] P.J. van der Houwen, Preconditioning in implicit initial value problem methods on parallel computers, Report NM-R9216, Centre for Mathematics and Computer Science, Amsterdam (1992); also *Adv. Comput. Math.* (to appear).
[12] P.J. van der Houwen and B.P. Sommeijer, Parallel iteration of high-order Runge–Kutta methods with stepsize control, *J. Comput. Appl. Math.* 29 (1990) 111–127.
[13] P.J. van der Houwen and B.P. Sommeijer, Butcher–Kuntzmann methods for nonstiff problems on parallel computers, Report NM-R9305, Centre for Mathematics and Computer Science, Amsterdam (1993).