CWI
Centrum voor Wiskunde en Informatica
REPORTRAPPORT

Butcher-Kuntzmann methods for nonstiff problems on parallel computers

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

Department of Numerical Mathematics

Report NM-R9305 March 1993

P.O. Box 4079, 1009 AB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

Butcher-Kuntzmann Methods for Nonstiff Problems on Parallel Computers

P.J. van der Houwen & B.P. Sommeijer
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Dedicated to John Butcher on the occasion of his sixtieth birthday

Abstract. 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 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 efficient than the best sequential methods.

CR Subject Classification (1991): G.1.7
Keywords and Phrases: 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)

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

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 [9] 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. [10], [11], [8], [7], and [6]). 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 [10] 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

Report NM-R9305
ISSN 0169-0388
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

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 righthand 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

$$(2.1)\qquad \mathbf{Y} = \mathbf{e}\otimes\mathbf{y}_n + h(A\otimes I_d)\mathbf{F}(\mathbf{Y}),\quad \mathbf{y}_{n+1} = \mathbf{y}_n + h(\mathbf{b}^T\otimes I_d)\mathbf{F}(\mathbf{Y}) = \mathbf{y}_n + (\mathbf{b}^T A^{-1}\otimes I_d)\,(\mathbf{Y} - \mathbf{e}\otimes\mathbf{y}_n).$$

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

$$(2.2a)\qquad \mathbf{R}_n(h,\mathbf{Y}) := \mathbf{Y} - \mathbf{e}\otimes\mathbf{y}_n - h(A\otimes I_d)\mathbf{F}(\mathbf{Y}),$$

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

$$(2.2b)\qquad \mathbf{Y}^{(j+1)} = \mathbf{Y}^{(j)} - P_j\mathbf{R}_n\big(h,\mathbf{Y}^{(j)}\big) + h^2Q_j\big(\mathbf{Y}^{(j)} - \mathbf{Y}^{(j-1)}\big),\ j = 0, 1, \dots, m-1,$$

where, $\mathbf{Y}^{(-1)} = \mathbf{Y}^{(0)}$ is a given initial iterate, and P_j and Q_j are sd-by-sd matrices whose entries may depend on the stepsize h and on the Jacobian matrix $J_n = \partial\mathbf{f}(\mathbf{y}_n)/\partial\mathbf{y}$. 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 $\mathbf{Y}$. The s stage vector components of $\mathbf{Y}^{(j+1)}$ 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

$$(2.2c)\qquad \mathbf{y}^{(j+1)} = \mathbf{y}_n + (\mathbf{b}^T A^{-1}\otimes I_d)\big(\mathbf{Y}^{(j+1)} - \mathbf{e}\otimes\mathbf{y}_n\big),\ j = 0, 1, \dots, m-1;\quad \mathbf{y}_{n+1} = \mathbf{y}^{(m)},$$

where the step value $\mathbf{y}_{n+1} = \mathbf{y}^{(m)}$ 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 [10], [11], [8], [7], and [6]. The matrices P_j in front of the residual function $\mathbf{R}_n$ 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

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

and we write (2.2b) in the form

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

For sufficiently smooth righthand side functions $\mathbf{f}$ we have

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

where J(U) is an sd-by-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

(2.3) $$\varepsilon^{(j+1)} = \big(I_{sd} - P_jC + h^2Q_j\big)\,\varepsilon^{(j)} - \,h^2Q_j\,\varepsilon^{(j-1)} + O(\varepsilon^{(j)})^2,\;\; C := I_{sd} - h(A\otimes I_d)J(Y),\; j = 0, \ldots , m\text{-}1.$$

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

(2.4a) $$Z_j := I_{sd} - \big(P_j + P_{j-1}\big)C + \big(P_jC - h^2Q_j\big)P_{j-1}C,\; j \text{ odd};\quad Z_j := I_{sd} - P_jC,\; j \text{ even}.$$

Then we obtain

(2.4b) $$\varepsilon^{(j+1)} = Z_j\,\varepsilon^{(j-1)} + O(\varepsilon^{(j)})^2 + O(\varepsilon^{(j-1)})^2,\;\; j = \text{odd};\quad \varepsilon^{(j+1)} = Z_j\,\varepsilon^{(j)} + O(\varepsilon^{(j)})^2,\;\; j = \text{even}.$$

Let us define the *error amplification matrix*

(2.5a) $$H_m := Z_{m-1} \,.\, Z_{m-3} \,.\, Z_{m-5} \,.\, \ldots .\, Z_3 \,.\, Z_1,\;\; m \text{ even};\qquad H_m := Z_{m-1} \,.\, Z_{m-2} \,.\, Z_{m-4} \,.\, \ldots .\, Z_3 \,.\, Z_1,\;\; m \text{ odd}.$$

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

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

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

$$y_{n+1} = y_n + (\mathbf{b}^TA^{-1}\otimes I_d)\big(Y^{(m)} - e\otimes y_n\big) = y_n + (\mathbf{b}^TA^{-1}\otimes I_d)(Y - e\otimes y_n) + (\mathbf{b}^TA^{-1}\otimes I_d)\varepsilon^{(m)}.$$

Let us introduce the (exact) corrector solution

(2.6) $$u_{n+1} := y_n + (\mathbf{b}^TA^{-1}\otimes I_d)\,(Y - e\otimes y_n),$$

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

(2.7a) $$H_m = h^{\theta m}[K_m \otimes L_m + \Delta H_m],\;\; \theta \geq 1,$$

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

(2.7b) $$y_{n+1} - u_{n+1} = (\mathbf{b}^TA^{-1}\otimes I_d)\,\varepsilon^{(m)} = h^{\theta m}\,\big[\mathbf{b}^TA^{-1}K_m\otimes L_m + (\mathbf{b}^TA^{-1}\otimes I_d)\,\Delta H_m\big]\;\varepsilon^{(0)}.$$

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

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

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 onestep 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 (onestep) 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

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

If $v_j = \mathbf{0}$ for $j = 0, \dots, q$ with $q \leq r$, then the iteration error at the step points is given by

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

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 righthand 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 $y(t_{n-1} + hc_i)$, $i = 1, 2, \dots, s$ (componentwise notation). Furthermore, by expressing the stage vector in the righthand side of the stagevector equation in (2.1) in terms of the exact solution, and again using componentwise notation, we obtain

$$Y \ = e \otimes y_n + h(A \otimes I_d)F(Y) = e \otimes y(t_n) + h(A \otimes I_d)F(y(t_n e + hc) + O(h^{r+1}))$$

$$= e \otimes y(t_n) + h(A \otimes I_d)y'(t_n e + hc) + O(h^{r+2}).$$

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

$$Y^{(0)} \ = (w + Ee) \otimes y(t_n) + (E \otimes I_d)\big(h(c\text{-}e) \otimes y'(t_n) + \frac{1}{2!}h^2(c\text{-}e)^2 \otimes y''(t_n) + \dots\big)$$

$$Y \quad = e \otimes y(t_n) + (A \otimes I_d)\big(he \otimes y'(t_n) + h^2 c \otimes y''(t_n) + \frac{1}{2!}h^3 c^2 \otimes y'''(t_n) + \dots\big) + O(h^{r+2}).$$

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 an 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 $\{\lambda_k\colon k = 1, 2, \dots, m\}$ 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 [5], 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

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

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

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

Then, the error amplification matrix H_m is given by

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

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

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

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

$$\begin{aligned} &Z_j = h^2\Big(-Q_j + M_j\big[(I_s \otimes J_n) + O(h\Delta J_n)\big] + P_j(A^2 \otimes I_d)\big[(I_s \otimes J_n{}^2) + O(h\Delta J_n)\big]\Big), \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad j \text{ odd},\ 1 \le j \le m-1. \\ &M_j := h^{-1}(I_{sd} - P_j + h^2 Q_j)(A \otimes I_d), \end{aligned}$$

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

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

For even values of j we derive

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

From (3.2) it follows that

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

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

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

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 $\theta = 1$, $K_m = A^m$, 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

$$(3.6)\qquad y_{n+1} - u_{n+1} = C_m S_m(J_n)\, h^{m+q+1}\, y^{(q+1)}(t_n) + O(h^{m+q+2}),\quad C_m = \mathbf{b}^T A^{m-1} v_{q+1}.\ \Box$$

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

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

(3.7b) Last-stage-vector predictor: {(2.8), (2.9)} with $v_j = \mathbf{0}$, $j = 0, \dots, r$, 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

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

where $C_m = -\,1/(m+1)!$ for $m \le p-1$, 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 $\mathbf{b}^T A^{m-1}\mathbf{c} = 1/(m+1)!$ for $m \le p-1$ follows from the order conditions for RK methods. □

For a number of Butcher-Kuntzmann correctors, Table 3.1a 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 $C_0 := -\,\mathbf{b}^T A^{-1}\mathbf{c}$ 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 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 $\gamma = -1/1728$).

Table 3.1a. Values of C_m / C_{m-1} for Butcher-Kuntzmann correctors using the Last-step-value predictor.

p	s	C_0	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8	m=9	m=10	...	m=∞
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

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

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

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

$$(3.8) \qquad \mathbf{w} = \mathbf{e} - E\mathbf{e}, \ \ E(\mathbf{c}-\mathbf{e})^j = jA\mathbf{c}^{j-1}, \ \ j = 1, \dots, s.$$

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 3.1a is given by Table 3.1b. 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.

Table 3.1b. Values of C_m / C_{m-1} for Butcher-Kuntzmann correctors using the Last-stage-vector predictor.

p	s	C_0	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8	m=9	m=10	...	m=∞
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

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:

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

$$R_m(z,h) = 1 + \beta_1(h)z + \beta_2(h)z^2 + \dots + \beta_m(h)z^m, \ \ \beta_j(h) := \frac{1}{j!}\mathbf{b}^T\frac{\partial^j M}{\partial z^j}(0,h)\ \mathbf{e}, \ \ z := h\lambda,$$

$$M(z,h) := z\left[I + zA + z^2A^2 + \dots\right]\left[I_s - S_m(h^{-1}z)[S_m(h^{-1}A^{-1})]^{-1}\right].$$

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

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

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

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

it follows that

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

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

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

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

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

For a few values of p and m, the real and imaginary stability boundaries $\{\beta_{real}, \beta_{imag}\}$ are given in Table 3.2. The three-digits 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 boundary, the given values may be considered as effective boundaries.

Table 3.2. Stability boundaries $\{\beta_{real}, \beta_{imag}\}$ for PIRKJ$\{m,\lambda_k\}$ 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*}

For nonzero fitting points, the polynomial (3.9) is approximated as h → 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 3.1. As an illustration, we derive the stability polynomial for the case where $p = m = 2$ and

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

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

$$R_2(z,h) = 1 + z\,[1 - h^2\lambda_0{}^2\mathbf{b}^T(I_s - h\lambda_0 A)^{-2}A^2\mathbf{e}\,] + \frac{1}{2}\,z^2\,[1 + 2h\lambda_0\mathbf{b}^T(2I_s - h\lambda_0 A)(I_s - h\lambda_0 A)^{-2}A^2\mathbf{e}\,].$$

For h → 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 $A = 1/2$ and $\mathbf{b} = 1$. 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 $h\lambda_0 \leq 3/2$ and $-2(1 - h\lambda_0) \leq z \leq 0$. Let $\rho(J_n)$ denote the spectral radius of the Jacobian J_n. For $\lambda_0 \leq -\rho(J_n)/2$, we have unconditional stability. For $\lambda_0 > -\rho(J_n)/2$, the stability condition becomes $h \leq 2[2\lambda_0 + \rho(J_n)]^{-1}$. 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 $h \leq 4\lambda_0[3\lambda_0{}^2 + \rho(J_n)^2]^{-1}$, whereas Runge's method in 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

$$(4.1) \qquad S^*_{2m}(x) = (\pi_0 - \sigma_0 x + x^2)(\pi_1 - \sigma_1 x + x^2). \; \dots \; .(\pi_{m-1} - \sigma_{m-1} x + x^2),$$

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

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

Then, the error amplification matrix H_m is given by

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

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

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

On substitution of (4.2) into W_j we obtain

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

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 righthand 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

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

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

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

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

Theorem 4.4. Let U and V be s-by-s matrices whose columns are respectively given by the vectors $\{jc^{j-1}, j = 1, \dots, s\}$ and $\{(c-e)^j, j = 1, \dots, s\}$, 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

$$\begin{aligned} &y_{n+1} - u_{n+1} = C_m S^*{}_{2m}(J_n)\, h^{2m+r+1}\, y^{(r+1)}(t_n) + O(h^{2m+r+1}\Delta J_n) + O(h^{2m+r+2}), \\ (4.5)\qquad & \\ &C_m := \frac{1}{(r+1)!}\, b^T A^{2m} [UV^{-1}(c-e)^{r+1} - (r+1)c^r].\ \square \end{aligned}$$

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

$$R^*{}_{2m}(z,h) = 1 + \beta_1(h)z + \beta_2(h)z^2 + \dots + \beta_{2m}(h)z^{2m}, \quad \beta_j(h) := \frac{1}{j!} b^T \frac{\partial^j M}{\partial z^j}(0,h)\, e, \quad z := h\lambda,$$

$$M(z,h) := z\, [I + zA + z^2A^2 + \dots]\, [I_s - S^*{}_{2m}(h^{-1}z)[S^*{}_{2m}(h^{-1}A^{-1})]^{-1}].\ \square$$

5. Spectral fitting polynomials

From the 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)

$$(5.1)\qquad S_m(z) = \frac{(b-a)^m}{2^{2m-1}}\, T_m\Big(\frac{a+b-2z}{a-b}\Big).$$

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:

$$(5.2)\qquad T_n\Big(\frac{a+b-2x}{a-b}\Big) = \frac{2^{2n-1}}{(b-a)^n} \prod_{k=1}^{n} (x - \omega_{nk}), \quad \omega_{nk} = \frac{1}{2}\Big[a + b - (a-b)\cos\Big(\frac{(2k-1)\pi}{2n}\Big)\Big], \quad k = 1, \dots, n.$$

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

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

where $\| \,.\, \|_R$ 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

$$(5.4)\qquad R_1(r,\phi) := \{\lambda = |\lambda|\, e^{i(\pi-\psi)}: 0 \le |\lambda| \le r, |\psi| \le \phi\}, \quad R_2(r,\phi) := \{\lambda = |\lambda|\, e^{i(\pi/2+\psi)}: 0 \le |\lambda| \le r, 0 \le \psi \le \phi\}.$$

It is easily verified that

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

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

Thus, it suffices to compute the constants characterizing S_m. These constants are given in Tables 5.1 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 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 axis yield smaller convergence factors for $\phi \le 60^o$ and $\phi \le 30^o$, respectively.

Table 5.1. Constants $\sigma_{R_j(1,\phi)}$ in formula (5.3) defined by (5.5).

	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]				
m	$\phi = 0^o$	$\phi = 15^o$	$\phi = 30^o$	$\phi = 60^o$	$\phi = 90^o$	$\phi = 0^o$	$\phi = 15^o$	$\phi = 30^o$	$\phi = 60^o$	$\phi = 90^o$
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

6. Numerical experiments

We illustrate the efficiency of the iterated Butcher-Kuntzmann methods on parallel computers by integrating the Arenstorf orbit problem given in [4, p.127] (see also the Appendix to this paper).

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-digits 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 we more or less copied from the code PIRK8 described in [6]. 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. Additional experiments are reported in the Appendix to this paper.

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 [12] 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 6.1 showing that the high corrector order and Jacobian-dependent preconditioners improve the efficiency drastically.

Table 6.1. Comparison with DOPRI8 for the Arenstorf orbit [4, p.127].

Method	Order of Corrector	Order of resulting method	Δ=3	Δ=4	Δ=5	Δ=6	Δ=7	Δ=8	averaged speed-up w.r.t. DOPRI8
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

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, in general, a corrector of high order, in combination with a predictor based on extrapolation, is 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 [10,11,8,7,6]. 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:

(i) 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.

(ii) 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-times-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. By several numerical examples (see also the Appendix), the efficiency-increasing effect of both acceleration techniques is illustrated 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] Butcher, J.C. (1964): Implicit Runge-Kutta processes, Math. Comp. 18, 50-64.

[2] Coleman, J.P. & Smith, R.A. (1987): The Faber polynomials for circular sectors, Math. Comp. 49, 81-84, 231-241.

[3] Davis, P.J. & Rabinowitz, P. (1975): Methods of numerical integration, Academic Press, New York.

[4] Hairer, E., Nørsett, S.P. & Wanner, G. (1987): Solving ordinary differential equations I. Nonstiff Problems, Springer Series in Comp. Math., vol. 8, Springer-Verlag, Berlin.

[5] Houwen, P.J. van der (1992): Preconditioning in implicit initial value problem methods on parallel computers, Report NM-R9216, Centre for Mathematics and Computer Science, Amsterdam.

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

[7] Jackson, K.R., Kværnø, A. & Nørsett, S.P. (1992): Order of Runge-Kutta methods when using Newton-type iteration, Technical Report No. 1/91, Division of Math. Sciences, University of Trondheim.

[8] Jackson, K.R. & Nørsett, S.P. (1990): The potential for parallelism in Runge-Kutta methods, Part I: RK formulas in standard form, Technical Report No. 239/90, Dept. of Computer Science, University of Toronto.

[9] Kuntzmann, J. (1961): Neuere Entwicklungen der Methoden von Runge und Kutta, Z. Angew. Math. Mech. 41, T28-T31.

[10] Lie, I. (1987): Some aspects of parallel Runge-Kutta methods, Report 3/87, Dept. of Mathematics, University of Trondheim

[11] 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, LNM 1386, Springer-Verlag, Berlin.

[12] Prince, P.J. & Dormand, J.R. (1981): High order embedded Runge-Kutta formulae, J. Comput. Appl. Math. 7, 67-75.

Appendix

A.1 Numerical experiments

We present a few examples illustrating the efficiency of the iterated Butcher-Kuntzmann methods on parallel computers. The methods tested are the PIRK$\{m,\lambda_k\}$ and PIRKJ$\{2m,\lambda_k\}$ methods defined in the Sections 3.1 and 4.1, respectively. In both methods, the underlying corrector is the eighth-order Butcher-Kuntzmann method ($p = 2s = 8$). Hence, the PIRK and PIRKJ methods need (at least) 4 processors. The predictor is given by (2.8) with $E = O$, i.e., by $Y^{(0)} = e \otimes y_n$. The fitting points $\{\lambda_k\}$ and the number of iterations are specified in the tables of results.

As reference methods, we shall use:

PIRK8:	eighth-order, parallel method proposed in [6]
DOPRI8:	8(7)-method of Prince and Dormand [12].

The PIRK8 method is identical with PIRK$\{m,\lambda_k\}$ if we set all $\lambda_k = 0$ and if we choose $m = 8$. The 8(7)-method of Prince and Dormand [12] is nowadays generally considered as one of the most efficient sequential methods.

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 observed order of accuracy is denoted by p^*. The computational effort is measured by the total number N of sequential right-hand side evaluations performed during the integration process.

Below, we list the test problems used, together with the regions where the eigenvalues of the Jacobian (along the exact solution curve) are located.

Euler problem (Problem JACB from [4, p. 236])

$$\begin{aligned} &y_1' = y_2y_3, && y_1(0) = 0, && \\ &y_2' = -y_1y_3, && y_2(0) = 1, && 0 \le t \le 60. \\ &y_3' = -.51y_1y_2, && y_3(0) = 1, && \end{aligned} \tag{A.1}$$

For this problem, the Jacobian possesses one real eigenvalue running through the interval [- 0.55, + 0.55], and two complex conjugated eigenvalues, located in the rectangles [- 0.27, + 0.27] x [0.21i, 1.23i], and [- 0.27, + 0.27] x [-1.23i, - 0.21i]. Since the spectrum is more or less clustered around the origin, suitable fitting points are just the points $\lambda_k = 0$.

Orbit problem (Problem TWOB from [4, p. 236])

$$\begin{aligned} &y_1' = y_3, && y_1(0) = 1 - \varepsilon,\ \varepsilon = \frac{3}{10}, && \\ &y_2' = y_4, && y_2(0) = 0, && \\ &y_3' = \frac{-y_1}{(y_1^2 + y_2^2)^{3/2}}, && y_3(0) = 0, && 0 \le t \le 20. \\ &y_4' = \frac{-y_2}{(y_1^2 + y_2^2)^{3/2}}, && y_4(0) = \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}, && \end{aligned} \tag{A.2}$$

The real eigenvalues of the Jacobian are in the intervals [- 2.42, - 0.95] and [+0.95, +2.42]; the purely imaginary eigenvalues are in the intervals [0.67i, 1.71i] and [- 1.71i, - 0.67i]. As in the preceding problem, the eigenvalue spectrum is clustered around the origin, indicating that the fitting points can best be chosen at the origin.

Arenstorf orbit (cf. [4, p.127])

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

In the greater part of the integration interval, the eigenvalues of the Jacobian are located in the neighbourhood of the origin. In this 'nonstiff part', the intervals [- 4.26, + 4.26] and [- 3.19i, + 3.19i], respectively contain the real and the purely imaginary eigenvalues. Similar to the preceding Orbit equation, this problem has a 'cross spectrum', so that the fitting points will again be chosen at the origin.

Lagrange problem (Problem LAGR from [4, p. 237])

$$
\text{(A.4)}\qquad
\begin{aligned}
y'_j &= y_{j+10},\ j = 1, 2, \dots, 10,\\
y'_{11} &= y_2 - y_1,\\
y'_{j+10} &= (j-1)y_{j-1} - (2j-1)y_j + jy_{j+1},\ j = 2, 3, \dots, 9; \qquad 0 \le t \le 10,\\
y'_{20} &= 9y_9 - 19y_{10},\\
y_j(0) &= 0 \text{ for } j \ne 8,\ \ y_8(0) = 1.
\end{aligned}
$$

The Jacobian of this problem has purely imaginary eigenvalues located in the intervals [+ 0.37i, + 5.47i] and [- 5.47i, - 0.37i]. Therefore, the optimal fitting polynomial should be the Chebyshev polynomial shifted to the interval [- 5.47i, + 5.47i].

Problem with eigenvalues in the left half-plane. With the exception of the Euler problem, the preceding test problems originated from systems of second-order ODEs, and, as a consequence, the eigenvalues were not restricted to the left half-plane which often occurs in genuine first-order systems. In order to see what spectral fitting can do for us in the case of eigenvalues that are located in disks centered at the negative axis, we chose the following problem (cf. [5]):

$$
\text{(A.5)}\qquad y'(t) = A(y(t))y(t) - \begin{pmatrix} 1 \\ 1 \end{pmatrix},\ A(y) := \begin{pmatrix} -1 & \cos(y_1) \\ -\cos(y_2) & -2 \end{pmatrix},\ y(0) = \mathbf{0},\ \ 0 \le t \le 2.
$$

According to Gerschgorin's disk theorem, this problem has its eigenvalues in two disks centered at - 1 and - 2 of radius at most 1. This suggests the identification of the fitting polynomial with the Chebyshev polynomial shifted to an interval on the negative real axis. We shall give results for the intervals [- 3, 0] and [- 2, - 1].

A.2. **Effect of introducing the Jacobian matrix into the preconditioner.** In order to examine the effect of introducing the Jacobian matrix into the preconditioner on the number of iterations needed to obtain the accuracy of the corrector, we compared the methods PIRKJ{2m,0} and PIRK{m,0} (which is identical to PIRK8 if we choose m = 8). Here, we used the test problems (A.1) - (A.3) which can only profit from a Jacobian-based preconditioner. The results listed in the Tables A.1a, A.1b and A.1c clearly show that PIRKJ{2m,0} reaches the accuracy of the corrector solution (indicated in bold) after fewer iterations than PIRK{m,0}. Roughly speaking, two to four iterations can be saved per step by using PIRKJ{2m,0}. We did not include Lagrange's problem (A.4), because in this problem we have a right-hand side function **f** of the form **f**(**y**) = J**y** with J constant. Hence, both variants are equally efficient.

Since the order of the PIRKJ{2m,0} method equals min(p,2m) and the order p of the corrector is equal to 8, four PIRKJ iterations suffice to reach this order. However, for the stepsizes used in the experiments, it is clear that four iterations are insufficient to solve the underlying corrector. Performing one additional iteration, usually leads to a more efficient algorithm. This observation has been used in Section 6, where we give results for an automatic version of the PIRKJ algorithm.

In order to see how the accuracy of the solution is influenced by an inaccurate Jacobian matrix, we integrated problem (A.5) by PIRKJ{2m,0} with the exact Jacobian and by approximating the Jacobian by the matrix A(y). In both cases, the Jacobian was updated in each step point. Table A.1d reveals that the accuracy produced by both PIRKJ{2m,0}-variants is considerably higher than the PIRK8-accuracy. However, for larger N (we observe that h = 2m/N), the PIRKJ{2m,0} method using the approximate Jacobian shows order p* = 3, whereas the exact Jacobian version almost shows its theoretical order min {p,2m} = 8.

Table A.1a. Comparison of PIRKJ{2m,0} and PIRK{m,0} for the Euler problem (A.1).

	Corrector	PIRK{m,0}			PIRKJ{2m,0}				
h	solution	m=4	m=6	m=8	m=4	m=5	m=6	m=7	m=8
1	4.6	0.4	1.8	3.5	1.6	2.6	3.8	**4.8**	**4.6**
1/2	6.9	1.5	3.6	6.0	4.3	5.9	**6.9**	**6.9**	**6.9**
1/4	9.3	2.8	5.6	8.5	7.3	9.8	**9.3**	**9.3**	**9.3**
p*	8.0	4.3	6.7	8.3	10.0	13.0	8.0	8.0	8.0

Table A.1b. Comparison of PIRKJ{2m,0} and PIRK{m,0} for the Orbit problem (A.2).

	Corrector	PIRK{m,0}			PIRKJ{2m,0}				
h	solution	m=4	m=6	m=8	m=4	m=5	m=6	m=7	m=8
1	2.1	0.2	- 0.3	1.0	0.6	**2.2**	**2.2**	**2.1**	**2.1**
1/2	4.6	0.0	1.5	3.3	3.1	5.0	**4.6**	**4.6**	**4.6**
1/4	6.9	1.4	3.4	5.9	5.8	**6.9**	**6.9**	**6.9**	**6.9**
p*	7.7	4.7	6.3	8.7	9.0	6.3	7.7	7.7	7.7

Table A.1c. Comparison of PIRKJ{2m,0} and PIRK{m,0} for the Arenstorf orbit (A.3).

	Corrector	PIRK{m,0}			PIRKJ{2m,0}				
h	solution	m=4	m=6	m=8	m=4	m=5	m=6	m=7	m=8
1/300	1.2	- 0.2	0.3	1.3	2.0	**1.2**	**1.2**	**1.2**	**1.2**
1/600	4.4	0.1	1.8	5.1	3.6	**4.4**	**4.4**	**4.4**	**4.4**
1/1200	6.6	1.0	3.7	6.9	6.1	**6.6**	**6.6**	**6.6**	**6.6**
p*	7.3	3.0	6.3	6.0	8.3	7.3	7.3	7.3	7.3

Table A.1d. Comparison of PIRKJ{2m,0} and PIRK{m,0}for problem (A.5).

Method	Jacobian	N = 16	N = 32	N = 64	p*
PIRK{8,0}		4.2	6.7	9.2	8.3
PIRKJ{8,0}	$A(y_n)$	7.1	9.2	10.1	3.0
PIRKJ{8,0}	$\partial f(y_n)/\partial y$	6.6	9.1	11.4	7.7

A.3. **Effect of introducing fitting points.** Next we compare the methods PIRK{m,0} and PIRK{m,λ_k} in order to see the effect of adapting the method to the spectrum of the Jacobian of the problem. From the location of the eigenvalue spectra specified above, spectral fitting seems to be promising for the problems (A.4) and (A.5) only. Notice that problem (A.4) originates from a second-order differential equation of the form $y'' = Jy$, with J symmetric. Applying Gerschgorin's disk theorem yields the eigenvalue interval [- 34, 0] for the matrix J. Hence, in our tests we identified the spectral fitting polynomial with the Chebyshev polynomial shifted to the interval $[-\sqrt{34}\ i, +\sqrt{34}\ i]$. For problem (A.5) we used the fitting intervals [- 2, - 1] and [- 3, 0].

Table A.2a. Comparison of PIRK{m,0} and PIRK{m,λ_k} for the Lagrange problem (A.4).

Method	h=1/2	h=1/4	h=1/8	h=1/16	p*
PIRK{8, 0}	0.4	2.5	5.1	7.6	8.3
PIRK{8, $\sqrt{34}$ i cos((2k-1)π/16)}	2.5	4.5	6.9	9.3	8.0

Table A.2b. Comparison of PIRK{m,0} and PIRK{m,λ_k} for problem (A.5).

Method	h=1	h=1/2	h=1/4	p*
PIRK{8, 0}	4.2	6.7	9.2	8.3
PIRK{8, cos((2k-1)π/16)/2 - 3/2}6.8	9.3	11.5	7.3	
PIRK{8, 3 cos((2k-1)π/16)/2 - 3/2}	5.6	7.9	10.4	8.3

The results, which are given in the Tables A.2, show that the a priori knowledge of the spectrum of the Jacobian can successfully be exploited for these problems: we observe an increase of the Δ-values by roughly 2 (that is a reduction of the global error by a factor 100). Comparing the Tables A.1d and A.2b, we see that both accelerating techniques have a comparable accuracy-increasing effect for problem (A.5).

A.4. Comparisons with DOPRI8. A realistic application of the PIRKJ(2m,λ_k) method requires a dynamic variation of the stepsize h. Therefore, PIRKJ{2m,0} and PIRK{m,λ_k} were provided with an automatic stepsize selection which is very similar to the stepsize strategy of the code PIRK8 described in [6]. In the present paper we confine ourselves to *fixed-order* methods: in the PIRKJ{2m,0} method we set m = 5 (resulting in order 8), since this results in an integration process that is usually more effficient than the process obtained by m = 4 (see also the discussion in Section 6); in the PIRK{m,λ_k} method we choose the minimal value to obtain an eighth-order scheme, i.e., m = 8. However, it is to be expected that a variable-m strategy (which may result a variable-order method) will enhance the performance of the codes. The codes PIRKJ{10,0} and PIRK{8,λ_k} will be compared to the code DOPRI8, which also has the fixed order 8. For various values of the local error control parameter, the codes were applied to the test problems, yielding a sequence of (Δ, N)-pairs (recall that N denotes the number of sequential right-hand side evaluations). 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 the Tables A.3. For the first three test problems we see that PIRKJ{10,0} is more efficient than DOPRI8 by a factor ranging from 2.0 until 3.4.

Table A.3a. Comparison of PIRKJ{10,0} and DOPRI8 for the Euler problem (A.1).

Method	Δ=3	Δ=4	Δ=5	Δ=6	Δ=7	Δ=8
DOPRI8	823	1083	1361	1864	2366	3038
PIRKJ{10,0}	419	509	607	714	904	1094
Speed-up factor	2.0	2.1	2.2	2.6	2.6	2.8

Table A.3b. Comparison of PIRKJ{10,0} and DOPRI8 for the Orbit problem (A.2).

Method	Δ=3	Δ=4	Δ=5	Δ=6	Δ=7	Δ=8
DOPRI8	428	601	729	870	1015	1160
PIRKJ{10,0}	186	224	270	316	385	469
Speed-up factor	2.3	2.7	2.7	2.8	2.6	2.5

Table A.3c. Comparison of PIRKJ{10,0} and DOPRI8 for the Arenstorf orbit (A.3).

Method	Δ=3	Δ=4	Δ=5	Δ=6	Δ=7	Δ=8
DOPRI8	1564	1900	2459	3215	3944	4817
PIRKJ{10,0}	514	601	790	986	1148	1660
Speed-up factor	3.0	3.2	3.1	3.3	3.4	2.9

For Lagrange's problem, where spectral fitting is appropriate, the speed-up of PIRK{8,λ_k} with respect to DOPRI8 is less spectacular; it ranges from a factor 1.2 until 1.6.

Table A.3d. Comparison of PIRK{8,λ_k} and DOPRI8 for the Lagrange problem (A.4).

Method	Δ=3	Δ=4	Δ=5	Δ=6	Δ=7	Δ=8
DOPRI8	429	540	668	841	1161	1498
PIRK{8, $\sqrt{34}$ i cos((2k-1)π/16)}	272	393	533	711	921	1190
Speed-up factor	1.6	1.4	1.3	1.2	1.3	1.3