
ELSEVIER

Journal of Computational and Applied Mathematics 101 (1999) 153–165

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Parallel Adams methods

P.J. van der Houwen*, E. Messina

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

Received 15 February 1998; received in revised form 8 September 1998

Abstract

In the literature, various types of parallel methods for integrating nonstiff initial-value problems for first-order ordinary differential equation have been proposed. The greater part of them are based on an implicit multistage method in which the implicit relations are solved by the predictor–corrector (or fixed point iteration) method. In the predictor–corrector approach the computation of the components of the stage vector iterate can be distributed over s processors, where s is the number of *implicit* stages of the corrector method. However, the fact that after each iteration the processors have to exchange their just computed results is often mentioned as a drawback, because it implies frequent communication between the processors. Particularly on distributed memory computers, such a fine grain parallelism is not attractive.

An alternative approach is based on implicit multistage methods which are such that the implicit stages are already parallel, so that they can be solved independently of each other. This means that only after completion of a step, the processors need to exchange their results. The purpose of this paper is the design of a class of parallel methods for solving *nonstiff* IVPs. We shall construct *explicit* methods of order $k+1$ with k parallel stages where each stage equation is of Adams–Bashforth type and *implicit* methods of order $k+2$ with k parallel stages which are of Adams–Moulton type. The abscissae in both families of methods are proved to be the Lobatto points, so that the Adams–Bashforth type method can be used as a predictor for the Adams–Moulton-type corrector.

Keywords: Numerical analysis; General linear methods; Parallelism

1. Introduction

We consider parallel methods for nonstiff initial-value problems (IVPs) for the first-order ordinary differential equation (ODE)

$$\frac{dy}{dt} = f(y), \quad y, \quad f \in \mathbb{R}^d, \quad t \geqslant t_0. \tag{1.1}$$

In the literature, various types of parallel methods for integrating such IVPs have been proposed. The greater part of them are based on an implicit method, usually a classical Runge–Kutta (RK) method or a multistep RK method, in which the implicit relations are solved by the predictor–corrector

* Corresponding author. E-mail: senna@cwi.nl.

0377-0427/99/$ – see front matter
PII: S0377-0427(98)00214-3

(or fixed point iteration) method. Within each iteration, the predictor–corrector approach is highly parallel. The parallel aspects of the predictor–corrector approach using RK-type correctors were analysed in e.g. Lie [15], Nørsett and Simonsen [18], Jackson and Nørsett [14], van der Houwen and Sommeijer [11], Burrage [3], and in Burrage and Suhartanto [5]. More general correctors for parallel computation were constructed in e.g. Miranker and Liniger [17], Chu and Hamilton [8], Birta and Osman Abou-Rabia [2]. The correctors in these last three papers are based on block methods, in which the blocks consist of solution values corresponding with equally spaced abscissae. Extensions to nonequidistant abscissae were studied in [12, 13, 21]. An extensive survey of parallel predictor–corrector methods can be found in the text book of Burrage [4].

In all parallel approaches indicated above, the computation of the components of the stage vector iterate can be distributed over s processors, where s is the number of *implicit* stages of the corrector method. However, the fact that after each iteration the processors have to exchange their just computed results is often mentioned as a drawback, because it implies frequent communication between the processors. Particularly on distributed memory computers, such a fine grain parallelism is not attractive.

An alternative approach is based on implicit multistage methods which are such that the implicit stages are already parallel, so that they can be solved independently of each other. This means that only after completion of a full integration step, the processors need to exchange their results. An example of an implicit method with only parallel stages is an RK method with a diagonal Butcher matrix. Unfortunately, such methods have a low order of accuracy. Higher orders can be obtained in the class of General Linear Methods (GLMs) of Butcher (see [6]). GLMs with parallel stages have been constructed in [20, 7]. As an example, consider the method [20]

$$\begin{aligned} y_{n+21/10} &= y_n + \frac{1}{660}h[541f(y_{n+11/10}) + 483f(y_n) + 462f(y_{n+21/10})], \\ y_{n+1} &= y_n + \frac{1}{660}h[-1000f(y_{n+11/10}) + 230f(y_n) + 1430f(y_{n+1})], \end{aligned} \tag{1.2}$$

where $y_{n+21/10}$ and y_{n+1} provide a 2nd-order and a 3rd-order approximation to $y(t_{n+21/10})$ and $y(t_{n+1})$, respectively. Evidently, the two associated implicit relations can be solved concurrently. Hence, effectively the method behaves as a one-implicit-stage method, provided that two processors are available. However, the methods of [20] and [7] are meant for *stiff* IVPs and great care was taken to make them A-stable. For a given number of stages, this of course limits the order of accuracy.

The purpose of this paper is the design of a class of parallel GLMs for solving *nonstiff* IVPs. Since the stability region is allowed to be finite, we can derive methods such that for a given number of stages, the orders of accuracy are greater than those of the A-stable methods derived in [20] and [7]. In fact, we shall construct *explicit* GLMs of order $k+1$ with k parallel stages where each stage equation is of Adams–Bashforth type and *implicit* GLMs of order $k+2$ with k parallel stages which are of Adams–Moulton type. The abscissae in both families of GLMs are proved to be the Lobatto points, so that the Adams–Bashforth type GLM can be used as a predictor for the Adams–Moulton-type corrector.

Our numerical experiments reveal that by virtue of the quite accurate Adams–Bashforth-type predictor, usually only one and at most two corrections are needed (see Section 4). Due to this low number of corrections, the most efficient strategy turns out to perform the same number of

corrections for all stage values. Hence, when implemented on a parallel computer system, there is a perfect load balancing, so that with respect to a sequential implementation of these methods, the speed-up factor is about k, i.e. the number of stages of the predictor–corrector pair.

2. General linear methods

In 1966 Butcher proposed the general linear method

$$Y_{n+1} = (R \otimes I)Y_n + h(S \otimes I)F(Y_n) + h(T \otimes I)F(Y_{n+1}), \quad n = 1,2,\ldots . \tag{2.1}$$

Here R, S and T denote $k \times k$ matrices, $\otimes$ the Kronecker product, h is the stepsize $t_{n+1} - t_n$, and each of the k components $y_{n+1,i}$ of the kd-dimensional solution vector Y_{n+1} represents a numerical approximation to $y(t_n + a_i h)$, to $hy'(t_n + a_i h)$ or "to any other quantity which enables us to construct and describe useful methods" (see Butcher [6, p. 339]). The vector $a := (a_i)$ is called the *abscissa vector*, the quantities Y_n the *stage vectors* and their components y_{ni} the *stage values*. Furthermore, for any vector $Y_n = (y_{ni})$, $F(Y_n)$ contains the right-hand-side values $(f(y_{ni}))$.

The GLM (2.1) is completely determined by the arrays $\{R, S, T\}$ and the starting vector Y_1. This starting vector should be computed by some one-step starting method (in the experiments reported in Section 4, we used the 8th-order Runge–Kutta method of Dormand–Prince). Thus, given the arrays $\{Y_1, R, S, T\}$, (2.1) completely defines the sequence of vectors $Y_2, Y_3, Y_4, \ldots$.

In this paper, we shall assume that $a_k = 1$ and that all components of Y_{n+1} represent numerical approximations to *solution* values $y(t_n + a_i h)$. Furthermore, we restrict our considerations to the case where T is a *diagonal* matrix with *nonnegative* diagonal entries δ_i. Such GLMs will be referred to as GLMs with parallel stages. It should be remarked that more general GLMs like the so-called DIMSIM methods of Butcher employ stage values that are not only solution values, but also derivative values of various orders. By choosing T diagonal, the stages in such methods become parallel. This is a subject of investigation in the group of Butcher in Auckland.

2.1. Solution of the implicit relations

If the matrix T has one or more nonzero diagonal entries δ_i, then $y_{n+1,i}$ has to be obtained by solving an (uncoupled) implicit relation of the form

$$y - \delta_i h f(y) = v_{ni}, \tag{2.2}$$

where v_{ni} is the corresponding d-dimensional vector component of $V_n := (R \otimes I)Y_n + h(S \otimes I)F(Y_n)$. Note that Eq. (2.2) can be solved concurrently. The convenional way of solving Eq. (2.2) in nonstiff situations is a fixed point iteration (briefly FP iteration) process of the form

$$y^{(j)} = \delta_i h f(y^{(j-1)}) + v_{ni}, \quad j \geqslant 1, \tag{2.3}$$

where $y^{(0)}$ represents an initial stage value iterate. These initial iterates can be generated by the GLM (2.1) with $T = O$ and with the same abscissa vector as the underlying implicit GLM. A sufficient

condition for the convergence of the process (2.3) is

$$h < \frac{1}{\delta_i \|\partial \boldsymbol{f}/\partial \boldsymbol{y}\|}. \tag{2.4}$$

Thus, in the construction of GLMs with parallel stages, we should choose δ_i sufficiently small.

2.2. Consistency

Consistency is defined by substitution of the exact solution into the GLM and by requiring that the residue vanishes as h tends to zero. The rate by which the residue tends to zero determines the *order of consistency*. We shall call the GLM (and the stage vector Y_{n+1}) *consistent of order p* if the residue upon substitution of the exact solution values $y(t_n + a_i h)$ into Eq. (2.1) is of order h^{p+1}. This leads to a set of order conditions to be satisfied by the matrices R, S and T. In addition, in order to have convergence, the GLM should satisfy the *necessary* condition of *zero-stability*, that is, the matrix R should have its eigenvalues on the unit disk and the eigenvalues of modulus one should have multiplicity not greater than 1.

From the consistency definition given above, the order conditions follow immediately. Let for any function g, the vector $g(\boldsymbol{a})$ be defined by $(g(a_i))$. Then, on substitution of the exact solution into Eq. (2.1) and requiring the Taylor series expansion of the residue to be of order $p+1$ in h yields

$$(R + hS)\exp(\boldsymbol{b}h) - (I - hT)\exp(\boldsymbol{a}h) = \mathrm{O}(h^{p+1}), \quad \boldsymbol{b} := \boldsymbol{a} - \boldsymbol{e}, \tag{2.5}$$

where $\boldsymbol{e}$ denotes a vector with unit entries. In the construction of GLMs, we shall start with a given zero-stable matrix R and a diagonal matrix T. The matrix S is then determined by imposing the order conditions. From Eq. (2.5) it easily follows that we obtain order of consistency $p = k$ (also called the *stage order*) if

$$R\boldsymbol{e} = \boldsymbol{e}, \quad S\boldsymbol{b}^{j-1} = \frac{1}{j}(\boldsymbol{a}^j - R\boldsymbol{b}^j) - T\boldsymbol{a}^{j-1}, \quad j = 1, \ldots, k. \tag{2.6}$$

Let us introduce the $k \times k$ matrices V_x and W_x:

$$V_x := (\boldsymbol{x}, \ldots, \boldsymbol{x}^k), \quad W_x := (\boldsymbol{e}, 2\boldsymbol{x}, 3\boldsymbol{x}^2, \ldots, k\boldsymbol{x}^{k-1}). \tag{2.7}$$

The consistency conditions (2.6) can now be expressed as

$$R\boldsymbol{e} = \boldsymbol{e}, \quad SW_b = V_a - RV_b - TW_a. \tag{2.8}$$

Given an abscissa vector $\boldsymbol{a}$ with distinct abscissae, a zero-stable matrix R satisfying the condition $R\boldsymbol{e} = \boldsymbol{e}$ and a matrix T, we obtain a family of GLMs with stage order k by defining

$$S = (V_a - RV_b - TW_a)W_b^{-1}. \tag{2.9}$$

In the case of the predictor formula needed in the FP iteration process, we set $T = O$, so that the S is completely defined as soon as $\boldsymbol{a}$ is prescribed. In the next section, we try to choose the abscissa vector $\boldsymbol{a}$ and the matrix T such that the *step point order* of the corrector, i.e. the order of the step point value $y_{n+1,k}$, is greater than k under the constraint that the diagonal entries of T are nonnegative and sufficiently small.

3. Construction of GLMs with parallel stages

If the GLM (2.1) is consistent of order k, then its error constants are given by the components of the vector

$$C(k) := \frac{1}{(k+1)!}(Rb^{k+1} + (k+1)(Sb^k + Ta^k) - a^{k+1}). \tag{3.1}$$

Hence, given a, R and T, and defining S by (2.9), this error vector can be written as

$$C(k) := \frac{1}{(k+1)!}(Tp(k) - q(k)), \tag{3.2}$$

where the vectors $p(k)$ and $q(k)$ are defined by

$$p(k) := (k+1)(a^k - W_a W_b^{-1} b^k), \quad q(k) := a^{k+1} - Rb^{k+1} - (k+1)((V_a - RV_b)W_b^{-1} b^k). \tag{3.3}$$

The stage order can be raised to $k+1$ by setting $C(k) = 0$, that is, $Tp(k) = q(k)$. Observing that δ_i may be chosen arbitrary whenever $p_i(k)$ and $q_i(k)$ both vanish, we have the following theorem:

Theorem 1. *Let $p(k)$ and $q(k)$ be defined by Eq. (3.3) where the matrix R is a given zero-stable matrix, and let the abscissae vector a be such that if an entry of $p(k)$ vanishes, then the corresponding entry of $q(k)$ also vanishes. Then all stage values in the GLM (2.1) have order of consistency $k+1$ if the matrices S and T are defined by*

$$T = \operatorname{diag}(q(k)p^{-1}(k)), \quad S = (V_a - RV_b - TW_a)W_b^{-1}. \tag{3.4}$$

3.1. Parallel Adams–Moulton and Adams–Bashforth methods

Let us choose the zero-stable matrix $R = ee_k^{\mathrm{T}}$, where e_k is the kth unit vector. Then, each stage value $y_{n+1,k}$ is defined by $y_{n,k}$ and the last k evaluations of f. Such stage equations are very similar to the classical Adams–Moulton formula, and therefore we shall call the special family of GLMs with $R = ee_k^{\mathrm{T}}$ and T diagonal, *parallel Adams–Moulton methods*, or briefly *PAM methods*.

For PAM methods, the quantities p and q defined in Eq. (3.3) are given by

$$p(k) = (k+1)(a^k - W_a W_b^{-1} b^k), \quad q(k) = a^{k+1} - (k+1)V_a W_b^{-1} b^k. \tag{3.5}$$

A second consequence of the choice $R = ee_k^{\mathrm{T}}$ is that we can obtain step point order $k+2$ if the last component of the second error vector

$$C(k+1) := \frac{1}{(k+2)!}(Rb^{k+2} + (k+2)(Sb^{k+1} + Ta^{k+1}) - a^{k+2})$$

vanishes. In order to derive a simple expression for this condition, we consider the last equation in system (2.1), viz.

$$y_{n+1,k} = y_{n,k} + h(e_k^{\mathrm{T}} S \otimes I)F(Y_n) + h\delta_k f(y_{n+1,k}), \tag{3.6}$$

and we compare this equation with the continuous analogue satisfied by the solution of (1.1), i.e.

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(y(t))\,\mathrm{d}t. \tag{3.7}$$

Thus, $h(e_k^T S \otimes I)F(U_n) + h\delta_k f(u_{n+1,k})$ may be considered as an interpolatory quadrature formula for the integral term in Eq. (3.7) using the k quadrature points $\{t_{ni}, i = 1,\ldots,k\}$ and the additional point $t_{n+1,k}$ with $t_{ji} := t_{j-1} + a_i h$. Assuming that $f(y(t))$ is sufficiently differentiable, such quadrature formulas possess a quadrature error of the form (see e.g. [1, p. 886])

$$Q_{k+1} := \frac{1}{(k+1)!}\int_{t_n}^{t_{n+1}} q_{k+1}(t) g_{k+1}(t)\,\mathrm{d}t,$$

$$q_{k+1}(t) := (t - t_{n+1,k})\prod_{i=1}^{k}(t - t_{ni}), \quad g_{k+1}(t) := \frac{d^{k+1} f(y(\theta(t)))}{dt^{k+1}},$$

where $\theta(t)$ assumes values in the interval (t_n, t_{n+1}). Defining the polynomial B_k and the integral $I_p(q)$

$$B_k(x) := \prod_{i=1}^{k}(x - b_i), \quad I_p(q) := \int_0^1 x^p B_q(x)\,\mathrm{d}x, \tag{3.8}$$

and substituting $t_{n+1,k} = t_n + h$, $t_{ni} = t_n + b_i h$, where $b_i := a_i - 1$, we can write

$$(k+1)!Q_{k+1} = h^{k+2}\int_0^1 (x-1)B_k(x)g_{k+1}(t_n + xh)\,\mathrm{d}x = h^{k+2}([I_1(k) - I_0(k)]g_{k+1}(t_n) + h[I_2(k) - I_1(k)]g'_{k+1}(t_n) + \tfrac{1}{2}h^2[I_3(k) - I_2(k)]g''_{k+1}(t_n) + \ldots). \tag{3.9}$$

Hence, the step point order of the method can be raised by one if $I_0(k) = I_1(k)$. Imposing this superconvergence condition yields a $(k-2)$-parameter family of PAM methods with step point order $p = k+2$ (we recall that $b_k = 0$). For $k = 2$, we find $b_{k-1} = b_1 = \frac{1}{2}$, so that the abscissae $a_1 = \frac{3}{2}$.

For $k \geqslant 3$ we shall exploit the additional degrees of freedom by choosing b such that the predictor formula to be used to start the FP iteration process (see Section 2.1) becomes also superconvergent. Let us define the predictor by the GLM (2.1) with

$$R = ee_k^{\mathrm{T}}, \quad S = V_a W_b^{-1}, \quad T = O, \tag{3.10}$$

where the abscissa vector is the same as in the PAM corrector. The method {(2.1), (3.9)} will be called a parallel Adams–Bashforth method and is denoted by PAB. Proceeding as above, we find for the approximation error the relation

$$k!Q_k = h^{k+1}\int_0^1 B_k(x)g(t_n + xh)\,\mathrm{d}x = h^{k+1}(I_0(k)g_k(t_n) + hI_1(k)g'_k(t_n) + \tfrac{1}{2}h^2 I_2(k)g''_k(t_n) + \cdots). \tag{3.11}$$

Thus, we have the result:

Lemma 2. *The PAB and PAM methods, respectively, have step point order $k+1$ and $k+2$ if the abscissae vector is such that $I_0(k) = I_1(k) = 0$.*

It is easily verified that for $k=3$ these conditions are satisfied by

$$b_1=\frac{6+\sqrt{6}}{10},\quad b_2=\frac{6-\sqrt{6}}{10},\quad b_3=0. \tag{3.12}$$

If $k\geqslant 4$, then we have a $(k-3)$-parameter family of solutions. The following theorem is straightforwardly verified.

Lemma 3. *Let $k\geqslant 4$ and let the shifted abscissae b_i be chosen symmetrically in the interval* $[0,1]$. *Then, $I_0(k)=I_1(k)=0$ for k odd and $I_0(k)-2I_1(k)=0$ for k even.*

From this theorem, we immediately conclude that for k *odd* the *Lobatto points* in the interval $[0,1]$ generate PAB and PAM methods with step point order $k+1$ and $k+2$, respectively. We will show that this is also true if k is *even*, or equivalently, $I_0(k)$ also vanish in the case of an *even* number of Lobatto points in the interval $[0,1]$. These Lobatto points are given by $b_1=1$, $b_k=0$, and by $b_i=z_i$, $i=2,\ldots,k-1$, where z_i is the $(i-1)$st zero of the derivative of the shifted Legendre polynomial $P_{k-1}(2x-1)$ (see e.g. [1, p. 888]). Hence, $B_k(x)=cx(x-1)P'_{k-1}(2x-1)$, where c is some constant. Transformation to the interval $[-1,1]$, integration by parts and observing that $P_1(y)=y$ yields

$$I_0(k)=c\int_0^1 x(x-1)P'_{k-1}(2x-1)\,\mathrm{d}x=-\frac{c}{4}\int_{-1}^1 yP_{k-1}(y)\,\mathrm{d}y=-\frac{c}{4}\int_{-1}^1 P_1(y)P_{k-1}(y)\,\mathrm{d}y$$

which does vanish for $k\geqslant 3$.

Theorem 4. *Let $k\geqslant 4$ and let the abscissae b_i be defined by the Lobatto points in the interval* $[0,1]$. *Then, the step point order of the PAB methods is $k+1$ and of the PAM methods $k+2$.*

In computing the matrices S and T corresponding to the Lobatto points, it turned out that for $4\leqslant k\leqslant 8$, the last component of both vectors $\boldsymbol{p}(k)$ and $\boldsymbol{q}(k)$ are of quite small magnitude (less than 10^{-18}), so that the value of δ_k is more or less free (cf. (3.2)). In our experiments, we have chosen $\delta_k=0.15$ for $4\leqslant k\leqslant 8$.

The Lobatto-based PAM methods are only useful if the size of the entries of S, T, and of the vector of error constants

$$\boldsymbol{E}(k):=(C_1(k+1),\quad C_2(k+1),\ldots,C_{k-1}(k+1),\quad C_k(k+2)). \tag{3.13}$$

is acceptably small. Table 1 lists the abscissa vector $\boldsymbol{a}$, the diagonal entries of T, $\|\boldsymbol{E}(k)\|_\infty$ and $\|S\|_\infty$ for $2\leqslant k\leqslant 8$ (left part). In all cases, the value of $\boldsymbol{E}\|(k)\|_\infty$ is quite acceptable. However, the value of $\|S\|_\infty$ rapidly increases with k and becomes inconveniently large for $k\geqslant 7$. Therefore, it is of interest to construct PAM methods in which the abscissae are chosen such that $\|\boldsymbol{E}(k)\|_\infty$ is minimized under the constraint that $\|S\|_\infty$ is small.

For that purpose, we first derive the *general* solution to the superconvergence condition of Lemma 2 by expressing the abscissae b_{k-2} and b_{k-1} explicitly in terms of the other abscissae. Let $\sigma:=b_{k-2}+b_{k-1}$ and $\pi:=b_{k-2}b_{k-1}$. Then $I_p(k)$ can be written as

$$I_p(k)=I_{p+3}(k-3)-I_{p+2}(k-3)\sigma+I_{p+1}(k-3)\pi,$$

Table 1
PAB and PAM methods

	Lobatto points				Abscissae obtained by minimization			
k	a	$qp^{-1}=(\delta_i)$	$\|E(k)\|_\infty$	$\|S\|_\infty$	a^*	$qp^{-1}=(\delta_i)$	$\|E(k)\|_\infty$	$\|S\|_\infty$
2	3/2	0.38	0.093	1.1				
	1	0.17						
3	$(16-\sqrt{6})/10$	0.18	0.047	2.2				
	$(16+\sqrt{6})/10$	0.33						
	1	0						
4	2	0.27	0.013	7.1				
	$(15+\sqrt{5})/10$	0.21						
	$(15-\sqrt{5})/10$	0.10						
	1	0.15						
5	2	0.23	$2.8_{10^{-3}}$	28	2	0.23	$3.0_{10^{-3}}$	27
	1.8273268354	0.20			1.808994654	0.20		
	1.5	0.14			1.463543856	0.13		
	$3-a_2$	0.06			1.148203456	0.05		
	1	0.15			1	0.15		
6	2	0.20	$5.0_{10^{-4}}$	118	2	0.20	$5.3_{10^{-4}}$	115
	1.8825276620	0.18			1.885275365	0.18		
	1.6426157582	0.14			1.630711359	0.14		
	$3-a_2$	0.04			1.083979860	0.03		
	$3-a_3$	0.09			1.320847699	0.09		
	1	0.15			1	0.15		
7	2	0.17	$8.1_{10^{-5}}$	522	2	0.10	$8.6_{10^{-5}}$	502
	1.9151119481	0.16			1.922202399	0.17		
	1.7344243967	0.14			1.731517951	0.14		
	1.5	0.10			1.472403452	0.10		
	$3-a_2$	0.03			1.062674121	0.03		
	$3-a_3$	0.07			1.226601360	0.06		
	1	0.15			1	0.15		
8	2	0.16	$1.2_{10^{-5}}$	2386	2	0.15	$1.1_{10^{-5}}$	4094
	1.9358700743	0.15			1.898303060	0.14		
	1.7958500907	0.13			1.801012442	0.13		
	1.6046496090	0.11			1.636888160	0.11		
	$3-a_2$	0.02			1.028569475	0.00		
	$3-a_3$	0.05			1.243500553	0.06		
	$3-a_4$	0.08			1.528856627	0.10		
	1	0.15			1	0.15		

so that the superconvergence condition leads to a linear system for $\sigma = b_{k-2} + b_{k-1}$ and $\pi = b_{k-2}b_{k-1}$. Hence, solving this linear system for σ and π, the shifted abscissae b_{k-2} and b_{k-1} are the solutions of the equation $b^2 - \sigma b + \pi = 0$, provided that this equation has real zeros. Thus, we have:

Table 2a
PAM stability boundaries, Lobatto points a in Table 1

k	2	3	4	5	6	7	8
β_{real}	2.39	1.36	0.88	0.96	0.46	0.36	0.17
β_{imag}	0.12	1.14	0.23	0.84	0.44	0.35	0.17

Table 2b
PAM stability boundaries, Abscissae a^* in Table 1

k	5	6	7	8
β_{real}	0.92	0.41	0.31	0.15
β_{imag}	0.81	0.40	0.30	0.15

Theorem 5. *Let $k \geqslant 4$ and let σ and π be defined by*

$$I_2(k-3)\sigma - I_1(k-3)\pi = I_3(k-3), \quad I_3(k-3)\sigma - I_2(k-3)\pi = I_4(k-3). \tag{3.14}$$

Then, the PAB and PAM methods, respectively, have step point order $k+1$ and $k+2$ if $\sigma^2 > 4\pi$ and if b_{k-2} and b_{k-1} satisfy the equation $b^2 - \sigma b + \pi = 0$.

Next, we restricted the entries of T to the interval [0, 0.5] and minimized the quantity

$$\|E(k)\|_\infty + c(\|S\|_\infty - k), \tag{3.15}$$

where $c=0$ if $\|S\|_\infty < k$ and $c=1$ otherwise. In this way, we computed for $4 \leqslant k \leqslant 8$ the abscissa vector a^*, the matrices T, $\|E(k)\|_\infty$ and $\|S\|_\infty$. It turned out that for $k=4$, we again obtained the Lobatto points. For $5 \leqslant k \leqslant 8$, we found the results as listed in Table 1 (right part). These figures show that for $5 \leqslant k \leqslant 7$ the values of $\|E(k)\|_\infty$ and $\|S\|_\infty$ are close to those obtained for the Lobatto points. Only for $k=8$, $\|E(k)\|_\infty$ is slightly reduced, however at the cost of a larger value of $\|S\|_\infty$.

3.2. Stability

The *linear stability region* $\mathbb{S}$ of Eq. (2.1) is defined by the set of points in the complex z-plane where

$$M(z) := (I - zT)^{-1}(R + zS) \tag{3.16}$$

has its eigenvalues within the unit circle. Process (2.1) will be called *linearly stable* if the eigenvalues of the matrix $h\partial f/\partial y$ are in $\mathbb{S}$. The maximal length of the negative interval $(-\beta_{\text{real}}, 0)$ and the imaginary interval $(0, i\beta_{\text{imag}})$ which is contained in $\mathbb{S}$ is called the *real* and *imaginary stability boundary*, respectively. Approximate values to these boundaries are listed in the Tables 2a and 2b, respectively, for the PAM methods with the Lobatto abscissae and with the abscissae obtained by minimization (see Table 1). These boundaries are quite acceptable, except for the imaginary stability boundaries of the 2-stage and 4-stage PAM methods. However, if we look at the intersection of the region $\mathbb{S}^*$ defined by the points z where $M(z)$ has eigenvalues of modulus less than say 1.001, then we obtain $\beta^*_{\text{imag}} \approx 0.75$ for $k=2$ and $\beta^*_{\text{imag}} \approx 0.78$ for $k=4$.

4. Numerical experiments

In this section we illustrate the performance of a few of the parallel GLMs constructed in the previous sections. In order to clearly see the algorithmic effects, we used a fixed stepsize strategy. The implicit relations were solved by FP iteration (see Section 2) with a fixed number of iterations for all stages. The tables of results list the total number of sequential righthand sides needed to produce a given number of correct digits Δ at the end point, that is, the maximal absolute end point error is written as $10^{-\Delta}$ (negative values are indicated by $-$).

We tested the k-stage PAM methods with step point order $p=k+2$ for $k=6,7,8$ with Lobatto abscissae. The starting values were obtained by the 8th-order Runge–Kutta method of Dormand–Prince [9] and the predictor needed to start the FP iteration method (2.3) was defined by the PAB method (3.10) with the same Lobatto abscissae. The resulting predictor–corrector method is denoted by PABM. We distinguish the $P(EC)^m$ mode and the $P(EC)^{m-1}E$ mode of the PABM method. On k processors, these modes require m right-hand sides per step that have to be computed sequentially.

We selected the following well-known test problems (cf. [10, p. 174]), viz. the Fehlberg problem:

$$\begin{aligned} y_1' &= 2ty_1 \log(\max\{y_2, 10^{-3}\}), \quad y_1(0)=1, \\ y_2' &= -2ty_2 \log(\max\{y_1, 10^{-3}\}), \quad y_2(0)=e, \end{aligned} \qquad 0 \leqslant t \leqslant 5, \tag{4.1}$$

the Euler problem JACB [10, p. 236]:

$$\begin{aligned} y_1' &= y_2 y_3, & y_1(0)&=0, \\ y_2' &= -y_1 y_3, & y_2(0)&=1, \quad 0 \leqslant t \leqslant 20, \\ y_3' &= -0.51 y_1 y_2, & y_3(0)&=1, \end{aligned} \tag{4.2}$$

and the Orbit problem TWOB [10, p. 236]:

$$\begin{aligned} y_1' &= y_3, \quad y_1(0)=1-\varepsilon, \\ y_2' &= y_4, \quad y_2(0)=0, \\ y_3' &= \frac{-y_1}{(y_1^2+y_2^2)^{3/2}}, \quad y_3(0)=0, \qquad 0 \leqslant t \leqslant 20. \\ y_4' &= \frac{-y_2}{(y_1^2+y_2^2)^{3/2}}, \quad y_4(0)=\sqrt{\frac{1+\varepsilon}{1-\varepsilon}}, \quad \varepsilon=\frac{1}{2}, \end{aligned} \tag{4.3}$$

4.1. Comparison of the PE, PEC, PECE and PECEC modes

First we want to know in what mode the PABMs are most efficient. We tested the PE, PEC, PECE and PECEC modes. The PABMs with 6, 7 and 8 stages were run on a lot of examples (not only the three test problems given above) and in most cases the PEC mode turned out to be the most efficient mode. A typical performance is given in the Tables 3a and 3b for the Fehlberg and Euler problems.

Table 3a
Number of sequential right-hand-side evaluations for the Fehlberg problem (4.1)

Method	$\Delta=5$	$\Delta=6$	$\Delta=7$	$\Delta=8$	$\Delta=9$	$\Delta=10$
PAB(PE)	222	274	338	431	622	873
PABM(PEC)	218	267	317	382	585	809
PABM(PECE)	270	349	445	569	697	847
PABM(PECEC)	273	349	447	563	699	867

Table 3b
Number of sequential right-hand-side evaluations for the Euler problem (4.2)

Method	$\Delta=5$	$\Delta=6$	$\Delta=7$	$\Delta=8$	$\Delta=9$	$\Delta=10$
PAB(PE)	96	123	158	210	281	374
PABM(PEC)	88	111	141	180	232	302
PABM(PECE)	103	135	173	221	283	363
PABM(PECEC)	105	137	177	223	283	363

4.2. *Comparison with other methods*

The PAMs in PEC mode were compared with a few methods from the literature, viz.

DOPRI	Dormand-Prince method of order 8 using 13 stages
ABM	Adams Moulton of order 8, 9 and 10 using AB predictors of order 7, 8 and 9
ABR	Adams-Bashforth-Radau methods of order 8 and 9 constructed in [13] and [21]

The ABR methods are also parallel predictor–corrector methods in which the predictor and the corrector are again of the form Eq. (2.1) but based on Radau abscissae. The corrector uses a nondiagonal matrix T and the FP iteration process is applied dynamically.

The Tables 4a–4c list the results produced by DOPRI, ABM, ABR and PABM. The ABR results of order 8 and 9 for the Euler and Fehlberg problem were taken from [13, 21], respectively (these papers do not contain results for the Orbit problem). The ABM and PABM methods were applied in their most effective modes, viz. the PECEC and the PEC mode, respectively. Furthermore, we listed the "speed-down" factors of all methods with respect to the PABM(PEC) method with step point order $p=10$ and the effective step point order of accuracy defined by $p_{\text{eff}} := (\Delta_2 - \Delta_1)\log_{10}(h_1 h_2^{-1})$, where (h_1, Δ_1) and (h_2, Δ_2) correspond with the results of highest and lowest accuracy, respectively.

5. Conclusions

Because of their simple structure, the PABM methods constructed in this paper are easily implemented on a parallel computer system. If k processors are available, then PABM methods with step point order up to $p=k+2$ can be implemented. The results presented above show that in terms of right-hand side evaluations, the speed-up factor of the 10th-order PABM(PEC) method, with respect to the Runge–Kutta and Adams methods DOPRI and ABM (which do not have any

Table 4a
Number of sequential right-hand-side evaluations for the Fehlberg problem (4.1)

Method	p	$\Delta=5$	$\Delta=6$	$\Delta=7$	$\Delta=8$	$\Delta=9$	$\Delta=10$	p_{eff}	Speed-down
DOPRI	8	688	934	1244	1608	2085	2690	8.4	3.7–5.9
ABM(PECEC)	8	548	677	810	1044	1536	2161	8.4	3.0–4.7
ABR	8	240	335	430	532	689	846	9.1	1.3–1.9
PABM(PEC)	8	218	267	317	382	585	809	8.8	1.2–1.8
ABM(PECEC)	9	497	653	853	1110	1437	1850	8.8	2.7–4.0
ABR	9		256	361	466	571	677	9.5	1.1–1.5
PABM(PEC)	9	188	223	276	351	445	558	10.6	1.0–1.2
ABM(PECEC)	10	468	570	719	916	1139	1383	10.6	2.6–3.0
PABM(PEC)	10	184	223	267	318	380	456	12.7	1

Table 4b
Number of sequential right-hand-side evaluations for the Euler problem (4.2)

Method	p	$\Delta=5$	$\Delta=6$	$\Delta=7$	$\Delta=8$	$\Delta=9$	$\Delta=10$	p_{eff}	Speed-down
DOPRI	8	379	505	685	901	1189	1567	8.1	5.3–8.5
ABM(PECEC)	8	221	266	379	575	798	1085	7.2	3.1–5.9
ABR	8	160	192	223	293	379	506	10.0	2.2–2.7
PABM(PEC)	8	88	111	141	180	232	302	9.3	1.2–1.6
ABM(PECEC)	9	198	278	394	514	658	834	8.0	2.6–4.5
ABR	9		117	169	221	273	325	9.0	1.4–1.8
PABM(PEC)	9	76	95	119	148	184	233	10.3	1.1–1.3
ABM(PECEC)	10	203	266	338	404	501	689	9.4	2.8–3.7
PABM(PEC)	10	72	84	101	121	149	185	12.2	1

Table 4c
Number of sequential right-hand-side evaluations for the Orbit problem (4.3)

Method	p	$\Delta=5$	$\Delta=6$	$\Delta=7$	$\Delta=8$	$\Delta=9$	$\Delta=10$	p_{eff}	Speed-down
DOPRI	8	1418	1834	2367	3030	3862	4862	9.3	5.1–5.5
ABM(PECEC)	8	1014	1380	2293	3172	4304	5794	6.6	3.7–6.5
PABM(PEC)	8	409	570	738	945	1207	1554	8.6	1.5–1.7
ABM(PECEC)	9	1144	1548	2014	2586	3291	4188	8.9	4.1–4.7
PABM(PEC)	9	332	386	510	715	946	1227	8.8	1.2–1.4
ABM(PECEC)	10	1046	1314	1585	1783	2365	3590	9.3	3.8–4.0
PABM(PEC)	10	276	336	477	604	741	892	9.8	1

scope for parallelism), ranges from 3.7 until 8.5. By virtue of the diagonal structure of the matrix T in (2.1), implying that only after completion of a full integration step the processors need to communicate with each other, the actual speed-up factors are expected to be close to the speed-up factors in terms of right-hand sides.

References

[1] M. Abramowitz, I.A. Stegun, Handbook of Mathematical Functions, Dover, New york, 1964.
[2] L.G. Birta, A.R. Osman, Parallel predictor–corrector methods for ode's, IEEE Trans. Comput. C36 (1987) 299–231.
[3] K. Burrage, Parallel block-predictor–corrector methods with an Adams type predictor, CSMR Report, Liverpool University, 1990.
[4] K. Burrage, Parallel and Sequential Methods for Ordinary Differential Equations, Clarendon Press, Oxford, 1995.
[5] K. Burrage, H. Suhartanto, Parallel iterated methods based on multistep Runge–Kutta methods of Radau type, Adv. Comput. Math. 7 (1997) 37–57.
[6] J.C. Butcher, The Numerical Analysis of Ordinary Differential Equations, Runge–Kutta and General Linear Methods, Wiley, New York, 1987.
[7] P. Chartier, Parallelism in the numerical solution of initial value problems for ODEs and DAEs, Thesis, Université de Rennes I, France, 1993.
[8] M.T. Chu, H. Hamilton, Parallel solution of ODE's by multi-block methods, SIAM J. Sci. Statist. Comput. 3 (1987) 342–353.
[9] J.R. Dormand, P.J. Prince, A family of embedded Runge–Kutta methods, J. Comput. Appl. Math. 6, (1980) 19–26.
[10] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations, vol. I. Nonstiff problems, Springer, Berlin, 1987.
[11] P.J. van der Houwen, B.P. Sommeijer, Parallel iteration of high-order Runge–Kutta methods with stepsize control, J. Comput. Appl. Math. 29 (1990) 111–127.
[12] P.J. van der Houwen, B.P. Sommeijer, Block Runge–Kutta methods on parallel computers, ZAMM 72 (1991) 3–18.
[13] P.J. van der Houwen, B.P. Sommeijer, J.J.B. Swart, Parallel predictor–corrector methods, J. Comput. Appl. Math. 66 (1996) 53–71.
[14] K.R. Jackson, S.P. Nørsett, The potential for parallelism in Runge–Kutta methods. Part 1: RK formulas in standard form, SIAM J. Numer. Anal. 32 (1995) 49–82.
[15] I. Lie, Some aspects of parallel Runge–Kutta methods, Report No. 3/87, Division Numerical Mathematics, University of Trondheim, 1987.
[16] E. Messina, J.J.B. de Swart, W.A. van der Veen, Parallel iterative linear solvers for multistep Runge–Kutta methods, J. Comput. Appl. Math. 85 (1997) 145–167.
[17] W.L. Miranker, W. Liniger, Parallel methods for the numerical integration of ordinary differential equations, Math. Comp. 21 (1967) 303–320.
[18] S.P. Nørsett, H.H. Simonsen, Aspects of parallel Runge–Kutta methods, in: A. Bellen, C.W. Gear, E. Russo (Eds.), Numerical Methods for Ordinary Differential Equations, Proceedings L'Aquila 1987, Lecture Notes in Mathematics 1386, Springer, Berlin (1989) 103–117.
[19] B. Orel, Parallel Runge–Kutta methods with real eigenvalues, Appl. Numer. Math. 11 (1993) 241–250.
[20] B.P. Sommeijer, W. Couzy, P.J. van der Houwen, A-stable parallel block methods for ordinary and integro-differential equations, Appl. Numer. Math. 9 (1992) 267–281.
[21] J.J.B. de Swart, Efficient parallel predictor–corrector methods, Appl. Numer. Math. 18 (1996) 387–396.