HOUWEN, P. J. VAN DER; SOMMEIJER, B. P.

Block Runge-Kutta Methods on Parallel Computers

In der vorliegenden Arbeit werden Block-Methoden zur Lösung von gewöhnlichen Differentialgleichungen auf Parallel-Computern konstruiert. Die meisten in der Literatur zu findenden Block-Methoden erzeugen Näherungen an die exakte Lösung in äquidistanten Punkten. Hier wird erlaubt, daß die Näherungen nicht-äquidistanten Punkten entsprechen können, wie das bei den Zwischennäherungen der Fall ist, die mit Runge-Kutta-Methoden berechnet werden. Dieser Ansatz gestattet die Verbesserung der Genauigkeitsordnung. Wir konzentrieren uns auf explizite Methoden, die zur Anwendung auf Parallel-Computern geeignet sind.

In this paper block methods for solving ODEs on parallel computers are constructed. Most block methods found in the literature produce approximations to the exact solution at equidistant points. Here, we allow that the approximations correspond to nonequidistant points like the intermediate approximations computed in Runge-Kutta methods. This approach enables us to improve the order of accuracy. We concentrate on explicit methods such that they are suitable for use on parallel computers.

В этой работе устанавливаются блочные методы для решения обыкновенных дифференциальных уравнений на параллельных ЭВМ. Большинство блочных методов встречающих в литературе дает приближения точных решений в равноостоящих точках. Здесь позволяем что приближения соответствуют в неравноостоящих точках как и промежуточные приближения вычисленные при помощи методов Рунге-Кутта. Этот модход облегчает улучшение порядка точности. Сосредоточным на явные методы так что они примеными на параллельныйх ЭВМ.

MS (1980): 65M10, 65M20

1. Introduction

Block methods turned out to be efficient methods for solving the initial-value problem for the system of ordinary differential equations (ODEs)

$$dy(t)/dt = f(y(t))$$

on parallel computers (cf. e.g. WORLAND [10] and CHU and HAMILTON [3]). Most block methods occurring in the literature can be interpreted as *block linear multistep methods* (BLM methods), that is, they are derived from the linear multistep (LM) method

$$\varrho(E) y_n = h\sigma(E) f(y_n),$$

in which y_n is replaced by an m-dimensional vector $Y_n := (y_{nm}, y_{nm+1}, \ldots, y_{nm+m-1})^\mathsf{T}$ and where the (scalar) coefficients of the polynomials ϱ and σ are replaced by matrices. Thus, in BLM methods the components of the block vector Y_n represent approximations to the exact solution at equidistant points.

In this paper, we consider block methods where the components of the block vector represent approximations to the exact solution at not necessarily equidistant points. In this way, we obtain additional parameters for increasing the order of accuracy of the method. In the derivation of these methods it turns out to be convenient to start with a Runge-Kutta (RK) method, and, in analogy with BLM methods, to replace the y-values generated by the method by vectors the components of which represent approximations to the exact solution. If these vectors are k-dimensional, then the RK parameters are replaced by $k \times k$ matrices. We shall call these methods *block Runge-Kutta methods* (BRK methods).

In Section 2, we give a precise definition of BRK methods and we give examples of methods from the literature which can be written as BRK methods. The representation in BRK form provides a unifying way of describing all sorts of methods (including BLM methods) and is particularly convenient for describing block methods for use on parallel computers. In Section 3 the order conditions for explicit one-stage methods and implicit two-stage methods are given, and Section 4 is devoted to the construction of these BRK methods with $k = 2, 3, 4$. We shall particularly be interested in explicit methods. For explicit methods with given k we tried to maximize the order and to minimize the number of processors without increasing the number of sequential righthand side evaluations per step (we shall call this minimal number of processors the *optimal* number of processors). It is possible to derive explicit one-stage methods of order $2k - 1$, using not more than 2 processors. However, if the requirement of zero-stability is imposed (which is crucial if the method is to be used as a method on its own), then the order reduces to $k + 1$. We also derive zero-stable, explicit two-stage methods of order $2k$ for two-processor computers. In Section 5, the various methods are compared for a few test problems from the literature.

It turned out that, like for all block methods, stability is a critical aspect of BRK methods. In this paper, we did not concentrate on stability aspects. Only when free parameters were available which could not be used for increasing the order, we have employed them to increase the stability of the method. A more systematic construction of BRK methods with large stability regions is the subject of a forthcoming paper.

2. Block Runge-Kutta methods

Let us start with the conventional s-stage RK method

$$y_{n+1}^{(i)} = y_n + h \sum_{j=1}^{s} b_{ij} f(y_{n+1}^{(j)}), \qquad i = 1, \ldots, s + 1 ; \qquad y_{n+1} = y_{n+1}^{(s+1)}, \qquad n = 0, 1, \ldots. \tag{2.1}$$

The general structure of the block Runge-Kutta methods considered in this paper is a direct generalization of this conventional method. We introduce block vectors Y_n, the components of which are numerical approximations to the exact solution values at k points. To be more precise, let Y_{n+1} be defined by

$$Y_{n+1} := (y_{n,c_1}, y_{n,c_2}, \ldots, y_{n,c_k})^\mathsf{T}, \qquad c_k = 1,$$

where $y_{n,c}$ denotes a numerical approximation to the exact solution value $y(t)$ at $t = t_{n+c} : t_n + ch$. For scalar ODEs, we now define the s-stage block RK (BRK) method

$$Y_{n+1}^{(i)} = A_i Y_n + h \sum_{j=1}^{s} B_{ij} f(Y_{n+1}^{(j)}), \qquad i = 1, \ldots, s + 1 ; \qquad Y_{n+1} = Y_{n+1}^{(s+1)}, \qquad n = 0, 1, \ldots, \tag{2.1'}$$

where A_i and B_{ij} are $k \times k$ matrices and where we use the convention that for any given vector $v = (v_j)$, $f(v)$ denotes the vector with entries $f(v_j)$. This method can be considered as the block analogue of (2.1). It is straightforwardly extended to systems of ODEs and therefore also to nonautonomous equations. In order to start the method, one needs the initial vector Y_0, which requires as many starting values as there are distinct values c_j, $j = 1, \ldots, k$.

In analogy with the Butcher array for describing the RK methods (2.1), i.e., the $(s + 1) \times (s + 1)$ array

$$
\begin{array}{c|ccc}
 & b_{11} & \ldots & b_{1s} \\
 & \vdots & \ldots & \vdots \\
 & b_{s1} & \ldots & b_{ss} \\
\hline
 & b_{s+1,1} & \ldots & b_{s+1,s}
\end{array},
$$

we may describe the BRK methods (2.1') by the $k(s + 1) \times k(s + 1)$ array

$$
\begin{array}{c|ccc}
A_1 & B_{11} & \ldots & B_{1s} \\
\vdots & \vdots & \ldots & \vdots \\
A_s & B_{s1} & \ldots & B_{ss} \\
\hline
A_{s+1} & B_{s+1,1} & \ldots & B_{s+1,s}
\end{array}.
$$

This notation is particularly convenient when more than two stages are involved. It frequently happens that the two last rows of this array are identical. In such cases, we shall omit the last row in order to save space.

We call the method *explicit* if the matrices B_{ij} vanish for $j \geq i$ and *implicit* otherwise. In this paper, we are mainly interested in e x p l i c i t methods. For explicit methods, the k components of the blocks $f(Y_{n+1}^{(j)})$ can be computed in parallel; hence, if k processors are available, then (explicit) BRK methods require not more than s (sequential) righthand side evaluations per step. However, the required number of processors is often less than k, without causing the number of (sequential) righthand side evaluations per step to exceed s. For instance, it may happen that in the formula for a particular component of Y_{n+1} no righthand side evaluations occur, that is, all rows in the matrices B_{ij} corresponding to this component vanish. In such cases, the processor assigned to this component is not needed. Similarly, if the r-th column of all matrices B_{ij} vanishes, then the computation of the corresponding component of Y_{n+1} does not require any righthand side evaluation not already occurring in the formulas for the other components, so that there is no need to assign a processor to this component. We define the *optimal number of processors* as the number of processors for which the number of (sequential) righthand side evaluations per step is minimal. In the explicit case, the representation (2.1') is very convenient for implementing the method on a computer, because the actual code is a direct translation of the formula (2.1') and the instructions for the computer in order to exploit the built-in parallelism of the method are obvious.

The points t_n and $t_n + c_j h$, $j \neq k$, will respectively be called *step points* and *block points*. Block points coincide with step points if the corresponding value of c_j is an integer. Upon completion of the integration process, the order of accuracy of the numerical solution obtained is not necessarily the same at all points $t_n + c_j h$. Points where the corresponding components of Y_{n+1} do have the same order as the components corresponding to the step points t_n will be called *output points*.

The *general explicit one-* and *two-stage methods* are respectively given by

$$\begin{array}{c|c} A_1 & O \\ \hline A_2 & B_{21} \end{array}, \quad \text{i.e.,} \quad Y_{n+1} = A_2 Y_n + h B_{21} f(A_1 Y_n),$$

and

$$\begin{array}{c|cc} A_1 & O & O \\ A_2 & B_{21} & O \\ \hline A_3 & B_{31} & B_{32} \end{array}, \quad \text{i.e.,} \quad Y_{n+1} = A_3 Y_n + h B_{31} f(A_1 Y_n) + h B_{32} f(A_2 Y_n + h B_{21} f(A_1 Y_n)).$$

Here, O denotes the $k \times k$ matrix with zero entries.

As a numerical example of an (explicit) 3-stage method, we present the modified multistep method of BUTCHER [1] of order 5 as a BRK method: the block point vector is given by $c = (0, 1)^T$ and the Butcher array assumes the form:

$$\begin{array}{cc|cccccc}
1 & 0 & & & & & & \\
0 & 1 & & & & & & \\
1 & 0 & 3/8 & 9/8 & & & & \\
0 & 1 & 0 & 0 & & & & \\
-23/5 & 28/5 & -26/15 & 0 & 32/15 & -4 & & \\
0 & 1 & 0 & 0 & 0 & 0 & & \\
\hline
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1/31 & 32/31 & -1/93 & 12/93 & 64/93 & 0 & 15/93 & 0
\end{array}, \quad c = (0, 1)^T.$$

The construction of higher-order BRK methods is rather difficult in the general case. In this paper, we shall construct high-order methods of a special form which are obtained by using the predictor-corrector (PC) technique. Our starting point is the special implicit two-stage method

$$\begin{array}{c|cc|cc} I & O & O & I & O & O \\ A & B & C & A & B & C \\ \hline A & B & C & & & \end{array} = \begin{array}{c|cc} & & \\ & & \end{array}, \quad \text{i.e.,} \quad Y_{n+1} = A Y_n + h B f(Y_n) + h C f(Y_{n+1}). \tag{2.2}$$

If C does not vanish, then we can use this method as corrector and if $C = O$, then it can be used as (a one-stage) predictor formula, e.g.,

$$\begin{array}{c|c} I & O \\ \hline A & B \end{array}, \quad \text{i.e.,} \quad Y_{n+1} = A Y_n + h B f(Y_n). \tag{2.2'}$$

From this pair we can generate higher-stage BRK methods by PC iteration provided that the block point vectors $c := (c_1, \ldots, c_k)^T$ are identical. For example, in PECE mode we obtain the special two-stage BRK method

$$\begin{array}{c|cc} I & O & O \\ D & E & O \\ \hline A & B & C \end{array}, \quad \text{i.e.,} \quad Y_{n+1} = A Y_n + h B f(Y_n) + h C f(D Y_n + h E f(Y_n)). \tag{2.3}$$

Finally, it should be remarked that (2.2) is also the representation of the so-called *general linear methods* introduced by Butcher in 1966 (see BUTCHER [2]). Most methods from the literature (including the general BRK method (2.1')) can be cast into the form (2.2). However, although the original method is explicit, the general linear method version is often implicit. For example, the explicit two-stage BRK method (2.3) can be rewritten in the form (2.2) by redefining the matrices A, B, and C in (2.2), but C will not be a zero matrix. Thus, for implementation of higher-stage BRK methods on parallel computers, the representation (2.2) is less suitable.

In the following subsections, we present in BRK form a number of methods which have been proposed for use on parallel computers. In particular, we give examples of the predictor-corrector methods of MIRANKER and LINIGER [8] and SHAMPINE and WATTS (cf. WORLAND [10]), and the multi-block methods of CHU and HAMILTON [3]. A discussion of block methods for parallel computation may be found in GEAR [5].

2.1 Methods of Miranker and Liniger

The methods of MIRANKER and LINIGER [8] can be presented as explicit, one-stage BRK methods. For example, their second-order method can be represented by the array

$$
\begin{array}{cc|cc}
1 & 0 & & \\
0 & 1 & & \\
\hline
0 & 1 & 2 & 0 \\
0 & 1 & 1/2 & 1/2
\end{array}
\quad, \quad c = (2, 1)^{\mathsf{T}},
\tag{2.4}
$$

and their fourth-order method by

$$
\begin{array}{cccc|cccc}
1 & 0 & 0 & 0 & & & & \\
0 & 1 & 0 & 0 & & & & \\
0 & 0 & 1 & 0 & & & & \\
0 & 0 & 0 & 1 & & & & \\
\hline
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1/3 & 4/3 & 8/3 & -5/3 \\
0 & 0 & 0 & 1 & 1/24 & -5/24 & 9/24 & 19/24
\end{array}
\quad, \quad c = (-1, 0, 2, 1)^{\mathsf{T}}.
\tag{2.5}
$$

Both methods require only two processors and respectively two and four starting values when implemented in BRK form.

2.2 Predictor-Corrector method of Shampine and Watts

The PC method of SHAMPINE and WATTS [9] is based on the block method of CLIPPINGER and DIMSDALE [1958], which can be presented in the form (2.2) by defining

$$
\begin{array}{cc|cccc}
1 & 0 & & & & \\
0 & 1 & & & & \\
0 & 1 & 0 & 5/24 & 1/3 & -1/24 \\
0 & 1 & 0 & 1/6 & 2/3 & 1/6
\end{array}
\quad, \quad c = (1/2, 1)^{\mathsf{T}},
\tag{2.6}
$$

and on the predictor method defined by

$$
\begin{array}{cccc|cccc}
1 & 0 & 0 & 0 & & & & \\
0 & 1 & 0 & 0 & & & & \\
0 & 0 & 1 & 0 & & & & \\
0 & 0 & 0 & 1 & & & & \\
\hline
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1/3 & 1/3 & 1/3 & 0 & 1/4 & -1/3 & 13/12 \\
0 & 1/3 & 1/3 & 1/3 & 0 & 29/24 & -3 & 79/24
\end{array}
\quad, \quad c = (-1/2, 0, 1/2, 1)^{\mathsf{T}}.
\tag{2.7}
$$

Method (2.6) is one of the oldest block methods proposed in the literature. SHAMPINE and WATTS proved that this corrector method is fourth-order accurate at the step points. They also proved that the predictor method is third-order accurate and possesses favourable stability properties. This predictor can also be applied as a method on its own and requires four starting values and one processor.

In order to apply the PC pair (2.7) − (2.6) using the BRK format, we rewrite the corrector in the form

$$
\begin{array}{cccc|cccccccc}
1 & 0 & 0 & 0 & & & & & & & & \\
0 & 1 & 0 & 0 & & & & & & & & \\
0 & 0 & 1 & 0 & & & & & & & & \\
0 & 0 & 0 & 1 & & & & & & & & \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 5/24 & 0 & 0 & 1/3 & -1/24 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1/6 & 0 & 0 & 2/3 & 1/6
\end{array}
\tag{2.6'}
$$

$$
, \quad c = (-1/2, 0, 1/2, 1)^{\mathsf{T}}.
$$

The PC method of Shampine and Watts was implemented by WORLAND [10] on two processors.

2.3 Multi-block methods of Chu and Hamilton

CHU and HAMILTON [3] generalized the cyclic linear multistep methods of DONELSON and HANSEN [4]. Families of third- and fourth-order multi-block methods were derived. We give two examples of their $k = 2$ methods which can be represented in the form (2.2) or (2.2'). The first example is the explicit third-order method

$$
\begin{array}{cc|cc}
1 & 0 & & \\
0 & 1 & & \\
\hline
5 & -4 & 1 & 2 \\
28 & -27 & 6 & 9
\end{array}, \quad c = (1/2, 1)^\mathsf{T},
\tag{2.8}
$$

and the second example is the fourth-order implicit method

$$
\begin{array}{cc|cccc}
1 & 0 & & & & \\
0 & 1 & & & & \\
0 & 1 & -1/48 & 13/48 & 13/48 & -1/48 \\
0 & 1 & 0 & 1/6 & 2/3 & 1/6
\end{array}, \quad c = (1/2, 1)^\mathsf{T}.
\tag{2.9}
$$

2.4 Parallel MRK methods

An example of methods which can be written in the form (2.3), and which do not originate from PECE methods, is the family of first-order, explicit parallel MRK methods (cf. VAN DER HOUWEN et al. [6])

$$
\begin{array}{ccc|ccc}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 - a_1 & a_1 & 0 & 0 & 0 \\
\hline
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & \dfrac{b_3}{(1 - a_1)} & 0 \\
0 & 0 & 1 & 1 - c - b_1 & b_1 & c
\end{array}, \quad c = (0, c, 1)^\mathsf{T},
\tag{2.10}
$$

where a_1, b_1, b_3 and c are free parameters. Third-order is obtained by setting

$$
a_1 := \frac{1}{2} + \frac{5}{6c} \pm \frac{1}{2} \sqrt{1 + \frac{10}{3c}}, \qquad b_1 = \frac{7}{3} - c a_1, \qquad b_3 = -\frac{5}{6c},
$$

with c as free parameter. These methods require three starting values and only one sequential righthand side evaluation on two processors. Notice that (2.10) is of the general explicit one-stage form in which the matrix A_1 has not been replaced by the identity matrix as was the case in (2.2').

3. Order conditions

In this section, we restrict our considerations to parameter arrays of the form (2.2) either with $C = O$ or $C \neq O$. Let the exact solution be substituted into (2.2). Then, in general, the order conditions are derived by requiring that the residual vector is of order h^{p+1} for all components (that is, we require that all components of Y_{n+1} are pth-order approximations to the corresponding exact solution values). In this way, we obtain the following condition for pth-order consistency:

$$
(I - zC) \exp(zc) - (A + zB) \exp(zc - ze) = O(z^{p+1}), \qquad e := (1, 1, \ldots, 1)^\mathsf{T}.
$$

By defining the error vectors

$$
C_0 := Ae - e; \qquad C_1 := A(c - e) + Be + Ce - c;
$$

$$
C_j := A(c - e)^j + j[B(c - e)^{j-1} + Cc^{j-1}] - c^j, \qquad j = 2, 3, \ldots,
\tag{3.1a}
$$

the conditions for pth-order consistency take the form

$$
C^j = 0, \qquad j = 0, 1, \ldots, p.
\tag{3.1b}
$$

Here, powers of vectors are meant to be componentwise powers.

In the construction of high-order formulas it is convenient to specify the matrix A in (2.2) in advance, because the eigenvalues of A should lie in a zero-stable configuration, that is, they should be on the unit disk, those on the unit circle being simple (such a zero-stability condition is difficult to satisfy simultaneously with the order conditions unless k is sufficiently small). A natural choice for the matrix A is suggested by observing that

$$Y_{n+1} - y_n e \approx \left(\int_{t_n}^{t_n+c_j} f(y(t)) \, dt \right).$$

Replacing the integral term by a quadrature formula, we obtain a method where A is of the form

$$A := \begin{bmatrix} 0 & \dots & 0 & 1 \\ . & \dots & . & . \\ 0 & \dots & 0 & 1 \end{bmatrix}. \tag{3.2}$$

This matrix has one eigenvalue 1 and $k - 1$ zero eigenvalues, so that a reasonable stability region may be expected (cf. the analogous situation for linear multistep methods of Adams type). BRK methods possessing a matrix A of the form (3.2) will be called *Adams-type methods*.

Assuming that A is given and is such that $Ae = e$, the most simple way to derive high-order formulas is to specify the vector c. This leaves us with a linear system of p equations for each component formula of the corrector formula. However, in this approach, the free parameters in the vector c are not exploited. These free parameters may be used for minimizing the error vector C_{p+1}. For instance, we may add to the order conditions (3.1) the condition that c is such that $\|C_{p+1}\|$ is minimal for some norm $\|.\|$. Alternatively, one may sacrifice the linearity of the order conditions and choose c such that certain components of the error vector vanish, that is, it is not necessary that all components of Y_{n+1} are pth-order approximations. As an example, in the Adams-type BRK methods with matrix A of the form (3.2), the first $k - 1$ components of Y_n only occur in the righthand side as argument of the function f, so that these components are allowed to be of one order less than the order of y_n, without decreasing the order of the approximations at the points t_n. To be more general, we denote the order of consistency of the formula for y_{n,c_i} by p_i and we define the set $J_q = \{i \in \{1, 2, \dots, k\} \mid p_i = q\}$. Now we introduce the following property.

Property 3.1: (i) $J_p \cup J_{p-1} = \{1, 2, \dots, k\}$ and (ii) *for each $i \in J_p$ the matrix A has vanishing elements a_{ij} for all* $j \in J_{p-1}$.

If this property is satisfied, then the method (2.2) produces pth-order accurate values at the points t_{n+c_i} for $n = 1, 2, \dots$ and $a \| i \in J_p$. One may interpret this as a form of superconvergence.

We recall that from an explicit and implicit BRK method with identical block point vector $c := (c_1, \dots, c_k)^T$, we can derive higher-stage BRK methods by PC iteration. By requiring that the explicit method (predictor) and the implicit method (corrector) provide approximations to $y(t_n + c_j h)$, respectively of orders q and p, for all j, we obtain after r iterations a method which provides approximations of order $p^* = \min \{p, q + r\}$. Since the predictor need not to be stable, one can employ the full freedom of the generating matrices, so that q is usually sufficiently large to get the maximal attainable order p of the corrector in just one correction (PECE mode). If not, then one may decide to continue the iteration.

4. Construction of BRK methods

Since the implementational complexity of the BRK method is mainly determined by the number of starting values and the associated storage needed to implement the method, we shall distinguish the various methods by their number of starting values. The methods constructed in the following subsections will be compared with methods from the literature.

4.1 Methods requiring two starting values

In this subsection we consider methods where the block vector Y_n is defined by

$$Y_{n+1} := (y_{n,c}, y_{n,1})^T.$$

At first sight, it would be natural to choose $c = 1/2$. However, as we shall see, a more judicious choice is possible.

4.1.1. Explicit one-stage methods

We shall construct the family of second-order BRK methods of Adams type and the general family of third-order methods.

Second-order methods of Adams type. The conditions (3.1) with $C = O$ and A defined by (3.2) can be satisfied for $p = 2$ and yield

$$
\begin{array}{cc|cc}
1 & 0 & & \\
0 & 1 & & \\
\hline
0 & 1 & \dfrac{-c^2}{2(1-c)} & \dfrac{c(2-c)}{2(1-c)} \\
0 & 1 & \dfrac{-1}{2(1-c)} & \dfrac{3-2c}{2(1-c)}
\end{array}
\,, \qquad c = (c, 1)^\mathsf{T}, \qquad c \neq 1 ,
\tag{4.1}
$$

with error vector

$$
C_3 = \frac{1}{2} \begin{pmatrix} c^2(c-3) \\ 3c - 5 \end{pmatrix}.
\tag{4.2}
$$

The following special cases of (4.1) will be tested in the numerical experiments at the end of this section:

$c = 0$	(4.1) reduces to the Adams-Bashforth method	$C_3 = (0.0, -2.5)^\mathsf{T}$
$c = 1/2$	'natural choice'	$C_3 \approx (-0.3, -1.8)^\mathsf{T}$
$c = 5/3$	Local error at t_{n+1} is $O(h^4)$	$C_3 \approx (-1.9, +0.0)^\mathsf{T}$
$c = 2$	(4.1) reduces to Miranker-Liniger method (2.4)	$C_3 = (-2.0, +0.5)^\mathsf{T}$
$c = 1 + 4^{1/3}$	$\|C_3\|_\alpha$ minimized	$C_3 \approx (-1.4, +1.4)^\mathsf{T}$
$c = 3$	Local error at t_{n+c} is $O(h^4)$	$C_3 = (0.0, +2.0)^\mathsf{T}$

We observe that the case $c = 5/3$ will raise the order to 3 at all step points t_n, in spite of the second-order accuracy of y_{n+c}, because of the special form of the matrix A (cf. Property 3.1).

Third-order methods. Next we construct the family of one-stage BRK methods in which all components are at least of third order. We find the method

$$
\begin{array}{cc|cc}
1 & 0 & & \\
0 & 1 & & \\
\hline
\dfrac{c^2(3-c)}{(1-c)^3} & \dfrac{1-3c}{(1-c)^3} & \dfrac{c^2}{(1-c)^2} & \dfrac{c}{(1-c)^2} \\
\dfrac{5-3c}{(1-c)^3} & \dfrac{-c^3+3c^2-4}{(1-c)^3} & \dfrac{2-c}{(1-c)^2} & \dfrac{(2-c)^2}{(1-c)^2}
\end{array}
\,, \qquad c = (c, 1)^\mathsf{T}, \qquad c \neq 1 ,
\tag{4.3}
$$

with error vector

$$
C_4 = \begin{pmatrix} -c^2 \\ -(2-c)^2 \end{pmatrix}.
$$

This method is zero-stable for all values of c for which the eigenvalues of A are on the unit disk and are not both equal to 1. Since A has the eigenvalues 1 and $(c^2 - 2c - 5)/(c - 1)^2$, we obtain the condition

$$
-1 \leq \lambda < 1, \qquad \lambda := \frac{c^2 - 2c - 5}{(c - 1)^2}.
$$

This leads to the necessary condition

$$
c \leq 1 - \sqrt{3}, \qquad c \geq 1 + \sqrt{3}.
\tag{4.4}
$$

The parasitic eigenvalue λ vanishes for $c = 1 \pm \sqrt{6}$. If $c = 1/2$, then the method reduces to the method (2.8) of Chu and Hamilton.

A number of experiments were carried out in order to illustrate the effect of c on the accuracy of the methods (4.1) and (4.3). We chose the nonlinear initial-value problem

$$y'(t) = \sin(y^5) - \sin(\sin^5(t)) + \cos(t), \qquad y(0) = 0, \qquad 0 \leq t \leq 1, \tag{4.5}$$

with exact solution $y(t) = \sin(t)$.

In the following table the results are given. In this table, the absolute error obtained at the end point of the integration interval is written in the form 10^{-d} (d may be interpreted as the number of correct decimal digits). Each column contains results which required the same number of sequential righthand sides. In these and subsequent experiments, the starting values incorporated in the initial vector Y_0 are taken from the exact solution.

Table 4.1. Correct decimal digits at $t = 1$ for problem (4.5) obtained by BRK methods with $k = 2$ and $s = 1$

Sequential righthand sides	6	12	24	48	96	order
Adams-Bashforth method	1.8	2.4	3.0	3.6	4.2	2
Miranker-Liniger method (2.4)	2.7	3.2	3.7	4.3	4.9	2
BRK method (4.1): $c = 1/2$	2.0	2.5	3.1	3.7	4.4	2
BRK method (4.1): $c = 1 + 4^{1/3}$	2.1	2.7	3.3	3.9	4.5	2
BRK method (4.1): $c = 3$	1.9	2.5	3.1	3.7	4.3	2
BRK method (4.1): $c = 5/3$	3.1	4.0	5.0	5.9	6.8	3
BRK method (4.3): $c = 1 + \sqrt{6}$	3.1	4.0	4.9	5.8	6.7	3
BRK method (4.3): $c = 1 - \sqrt{6}$	3.3	4.1	4.9	5.8	6.7	3

These results show the theoretical order of accuracy. It is clear that the choice $c = 1/2$ is not the best possible. Furthermore, the value $c = 1 + 4^{1/3}$ (minimal norm value) does not improve the accuracy, so that we refrain from considering this special case in subsequent sections. Notice that the method (4.1) with $c = 5/3$ produces results which are comparable with the results of the method (4.3) with $c = 1 \pm \sqrt{6}$.

4.1.2. Implicit two-stage methods of Adams-type

The conditions (3.1) with nonvanishing matrix C can be satisfied for $p = 4$ by

$$\begin{array}{cc|cccc}
1 & 0 & & & & \\
0 & 1 & & & & \\
0 & 1 & \dfrac{-c^3}{12(1-c)} & \dfrac{c(c^2 - 6c + 6)}{12(1-c)} & \dfrac{c(c^2 - 6c + 6)}{12(1-c)} & \dfrac{-c^3}{12(1-c)} \\
0 & 1 & \dfrac{(1-2c)}{12(1-c)(2-c)} & \dfrac{-6c^2 + 10c - 3}{12c(1-c)} & \dfrac{3-2c}{12c(1-c)} & \dfrac{6c^2 - 14c + 7}{12(1-c)(2-c)}
\end{array}$$

$$c = (c, 1)^{\mathsf{T}}, \qquad c \neq 0, 1, 2. \tag{4.6}$$

The corresponding error vector is given by

$$C_5 = -\frac{1}{6} \begin{pmatrix} c^3(c^2 - 5c + 5) \\ 5c^2 - 10c + 4 \end{pmatrix}.$$

The following special cases of (4.6) will be considered:

$c = \dfrac{1}{2}$	(4.6) is equivalent with the corrector (2.9)	$C_5 = \left(-\dfrac{11}{192}, -\dfrac{8}{192}\right)^{\mathsf{T}}$
$c = 1 - \dfrac{\sqrt{5}}{5}$	Local error at t_{n+1} is $O(h^6)$	$C_5 = \left(-\dfrac{4\sqrt{5}}{125}, 0\right)^{\mathsf{T}}$

4.1.3. Predictor-corrector methods

In order to 'solve' the corrector equation defined by (4.6) one may use a PC method with predictor defined by (4.3). The PC methods determined by the matrices (4.3)−(4.6) require two starting values and, in PECE mode, they have all at least order 4. For $c = 1 - \sqrt{5}/5$, we achieve order 5 in $\mathrm{PE(CE)^2}$ mode. We remark that for the predictor formula, the value of c is not required to satisfy the inequalities (4.4).

We illustrate the performance of the PC method (4.3) − (4.6) by comparing it with the 2-step Adams PC method (notice that the BRK method (4.3)−(4.6) with $c = 1/2$ is equivalent with the Chu-Hamilton pair (2.8)−(2.9)). In the Tables 4.2, the correct decimal digits at $t = 1$ and the total numbers of sequential righthand sides are listed for the various methods in PECE mode and in $PE(CE)^2$ mode.

Table 4.2a. Correct decimal digits at $t = 1$ for problem (4.5) obtained by BRK methods in PECE mode with $k = 2$

Sequential righthand sides	6	12	24	48	96	order
Two-step Adams-PC method	2.1	3.1	4.1	5.0	5.9	3
Chu-Hamilton pair (2.8)−(2.9)	4.3	5.4	6.5	7.6	8.7	4
BRK method (4.3)−(4.6): $c = 1 - \sqrt{5}/5$	4.8	5.4	6.5	7.6	8.8	4

Table 4.2b. Correct decimal digits at $t = 1$ for problem(4.5) obtained by BRK methods in $PE(CE)^2$ mode with $k = 2$

Sequential righthand sides	6	12	24	48	96	order
Two-step Adams-PC method	1.8	3.1	4.2	5.1	6.0	3
Chu-Hamilton pair (2.8)−(2.9)	3.9	5.7	9.3	8.4	9.5	4
BRK method (4.3)−(4.6): $c = 1 - \sqrt{5}/5$	3.9	5.5	7.0	8.5	10.0	5

4.2. Methods requiring three starting values

The block vector Y_n is now defined by

$$Y_{n+1} := (y_{n,c_1}, y_{n,c_2}, y_{n,1})^\mathsf{T}$$

providing us with two free parameters. As before, equidistant output points need not to be the best choice. Because of the rapidly increasing complexity of the derivations if more than 2 starting values are used, we shall not consider the general case as in the preceding section, but we shall restrict our considerations to a few special cases.

4.2.1. Explicit one-stage methods

We consider Adams-type methods and a more general family of zero-stable methods.

Third-order methods of Adams type. If $C = O$, then the following array satisfies the order conditions (3.1) for $p = 3$ and for all (distinct) values of c_1 and c_2 different from 1:

$$
\begin{array}{ccc|ccc}
1 & 0 & 0 & & & \\
0 & 1 & 0 & & & \\
0 & 0 & 1 & & & \\
\hline
0 & 0 & 1 & a_1 - (c_2 - 1)b_1 & (c_1 - 1)b_1 & c_1 - a_1 + (c_2 - c_1)b_1 \\
0 & 0 & 1 & a_2 - (c_2 - 1)b_2 & (c_1 - 1)b_2 & c_2 - a_2 + (c_2 - c_1)b_2 \\
0 & 0 & 1 & a_3 - (c_2 - 1)b_3 & (c_1 - 1)b_3 & c_3 - a_3 + (c_2 - c_1)b_3 \\
\end{array}
\quad, \quad c = (c_1, c_2\ 1)^\mathsf{T},
\tag{4.7}
$$

where

$$a_j := \frac{c_j^2}{2(c_1 - 1)}, \qquad b_j := \frac{c_j^2(2c_j - 3c_1 + 3)}{6(c_1 - 1)(c_2 - 1)(c_2 - c_1)}, \qquad j = 1, 2, 3.$$

We restrict our considerations to the two-processor case, that is, we set $c_1 = 0$. By virtue of the special form of A we obtain order $p = 4$ at the step points if the third formula has order 4 while the first and second formula have order 3. Setting the third error component equal to zero we find $c_2 = 17/10$.

Fourth-order methods. Let us consider methods of the form

$$
\begin{array}{ccc|ccc}
1 & 0 & 0 & & & \\
0 & 1 & 0 & & & \\
0 & 0 & 1 & & & \\
\hline
0 & 0 & 1 & 0 & 0 & 0 \\
a_{21} & a_{22} & a_{23} & b_{21} & b_{22} & b_{23} \\
a_{31} & a_{32} & a_{33} & b_{31} & b_{32} & b_{33} \\
\end{array}
\quad, \quad c = (0, c, 1)^\mathsf{T}.
\tag{4.8}
$$

Solving the conditions (3.1) for $p = 4$ with $c = 1/2$ we obtain

$$
\left[
\begin{array}{ccc|ccc}
1 & 0 & 0 & & & \\
0 & 1 & 0 & & & \\
0 & 0 & 1 & & & \\
\hline
0 & 0 & 1 & 0 & 0 & 0 \\
-9-a & 9 & 1+a & (-10-a)/6 & (-22-4a)/6 & (8-a)/6 \\
-b & 64 & -63+b & (-9-b)/6 & (108-4b)/6 & (99-b)/6
\end{array}
\right], \quad c = (0, 1/2, 1)^\mathsf{T}.
$$

where a and b are free parameters. We could have used these parameters for increasing the order of accuracy to $p = 5$. However, then the method turns out to be zero-unstable. Therefore, we shall employ them for improving the stability of the method. In particular, we choose a and b such that A has zero parasitic roots. The characteristic equation of A is given by

$$(\delta - 1)(\delta^2 + (55 - b)\delta + 9b - 64a - 576) = 0,$$

so that we are led to the values $a = -81/64$ and $b = 55$. The corresponding Butcher array becomes

$$
\left[
\begin{array}{ccc|ccc}
1 & 0 & 0 & & & \\
0 & 1 & 0 & & & \\
0 & 0 & 1 & & & \\
\hline
0 & 0 & 1 & 0 & 0 & 0 \\
-495/64 & 9 & -17/64 & -559/384 & -271/96 & 593/384 \\
-55 & 64 & -8 & -32/3 & -56/3 & 22/3
\end{array}
\right], \quad c = (0, 1/2, 1)^\mathsf{T}. \tag{4.9}
$$

The following table illustrates the performance of the above explicit, one-stage methods.

Table 4.3. Correct decimal digits at $t = 1$ for problem (4.5) obtained by BRK methods with $k = 3$ and $s = 1$

Sequential righthand sides	6	12	24	48	96	order
Adams-Bashforth	3.2	3.9	4.8	5.6	6.5	3
BRK method (4.7): $(c_1, c_2) = (0, 1/2)$	3.4	4.2	5.1	6.0	6.9	3
BRK method (4.7): $(c_1, c_2) = (0, 17/10)$	4.1	5.3	6.5	7.7	8.9	4
BRK method (4.9)	4.0	5.1	6.4	7.6	8.8	4

4.2.2. Implicit two-stage methods

We assume the generating array of the form

$$
\left[
\begin{array}{ccc|cccccc}
1 & 0 & 0 & & & & & & \\
0 & 1 & 0 & & & & & & \\
0 & 0 & 1 & & & & & & \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{21} & a_{22} & a_{23} & b_{21} & b_{22} & b_{23} & 0 & c_{22} & c_{23} \\
a_{31} & a_{32} & a_{33} & b_{31} & b_{32} & b_{33} & 0 & c_{32} & c_{33}
\end{array}
\right], \quad c = (0, c, 1)^\mathsf{T} \tag{4.10}
$$

and we derive a fifth-order method of Adams type. Let us choose $c = 1/2$ and define A according to (3.2), then the order conditions (3.1) can be satisfied for $p = 5$ by

$$
\left[
\begin{array}{ccc|cccccc}
1 & 0 & 0 & & & & & & \\
0 & 1 & 0 & & & & & & \\
0 & 0 & 1 & & & & & & \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 11/1440 & -37/720 & 19/60 & 0 & 173/720 & -19/1440 \\
0 & 0 & 1 & -1/180 & 1/45 & 2/15 & 0 & 31/45 & 29/180
\end{array}
\right], \quad c = (0, 1/2, 1)^\mathsf{T}. \tag{4.11}
$$

If the method is not required to be of Adams type, then a corrector of order 6 can be constructed. In the following subsection, this will be carried out as part of the construction of a PECE method with increased real stability interval.

4.2.3. Predictor-corrector methods

We consider two PC methods which are in PECE mode of orders 5 and 6, respectively.

Method of order 5. The fourth-order predictor (4.9) and the fifth-order corrector (4.11) determine a PC method of order $p = 5$. It requires three starting values and, if two processors are available, then only two sequential righthand side evaluations per step are needed.

Method of order 6. Next we consider PC methods where the predictor and corrector are generated by matrices of the form (4.8) and (4.10), and where c is still a free parameter. We try to construct a PC method which is of order 6 in PECE mode by choosing the free parameters such that the corrector formula for y_{n+1} becomes of order $p = 6$, whereas the other corrector formula and the two predictor formulas have order $p = 5$.

In this paper, we have investigated methods where

$$
A := \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ a & 0 & 1-a \end{bmatrix}
$$

(notice that A does not refer to the second component of the block vector so that the corrector formula corresponding to this component may be of one order less than that of the third component). This leads to a one-parameter family of sixth-order PECE methods which can be represented in the form (2.3), i.e.

$$
Y_{n+1} = A Y_n + hBf(Y_n) + hCf(DY_n + hEf(Y_n)) .
$$

The free parameter will be used to improve the (linear) stability of the method. The (linear) stability of this two-stage BRK method can be investigated by applying the method to the test equation $y' = \lambda y$ to obtain the recursion

$$
Y_{n+1} = R(z) Y_n , \qquad R(z) := A + z(B + CD) + z^2 CE , \qquad z := \lambda h ,
$$

and by requiring that the matrix R satisfies the simple von Neumann stability condition, that is, it has its eigenvalues on the unit disk those on the unit circle being simple. Choosing c as the free parameter, we start with determining a range of relevant c-values by requiring that $R(0)$ satisfies the stability condition (zero-stability). Since the eigenvalues of $R(0) = A$ are given by 0, 1 and $-a$, we require $-1 \leqq -a < 1$. It can be shown that imposing the conditions for sixth-order accuracy on the corrector formula for y_{n+1} leads to

$$
a = \frac{15c^2 - 31c + 13}{15c^2 + c - 3} ,
$$

so that c should be not less than $1/2$ in order to ensure zero-stability. As before, we shall not consider the maximization of the general stability boundary. Instead we consider the simpler case of maximizing the real stability boundary. A numerical search reveals that the real stability boundary is maximized for $c \approx 4.16$ and is approximately given by 2.247. In order to obtain rational expressions for the entries of the various matrices, we do not choose this 'optimal' value of c, but we set $c = 4$ yielding the stability boundary 1.766. The predictor is generated by the matrices

$$
\left[\begin{array}{ccc|ccc}
1 & 0 & 0 & & & \\
0 & 1 & 0 & & & \\
0 & 0 & 1 & & & \\
\hline
0 & 0 & 1 & 0 & 0 & 0 \\
\dfrac{27}{2} & \dfrac{-25}{54} & \dfrac{-325}{27} & 5 & \dfrac{25}{9} & \dfrac{100}{9} \\
\dfrac{3}{2} & \dfrac{5}{54} & \dfrac{-16}{27} & 1 & \dfrac{-1}{18} & \dfrac{16}{9}
\end{array} \right], \qquad c = (0, 4, 1)^{\mathsf{T}} ,
$$

(4.12)

and the corrector by

$$
\left[
\begin{array}{ccc|cccccc}
1 & 0 & 0 & & & & & & \\
0 & 1 & 0 & & & & & & \\
0 & 0 & 1 & & & & & & \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & \dfrac{4}{75} & \dfrac{76}{45} & \dfrac{2}{45} & 0 & \dfrac{58}{225} & \dfrac{88}{45} \\
\dfrac{129}{241} & 0 & \dfrac{112}{241} & \dfrac{1141}{7230} & \dfrac{-47}{4338} & \dfrac{2110}{2169} & 0 & \dfrac{26}{10845} & \dfrac{896}{2169}
\end{array}
\right], \qquad c = (0, 4, 1)^{\mathsf{T}}.
\tag{4.13}
$$

The following table is the $k = 3$ analogue of the preceding tables.

Table 4.4. Correct decimal digits at $t = 1$ for problem (4.5) obtained by BRK methods in PECE mode with $k = 3$

Sequential righthand sides	6	12	24	48	96	order
Three-step Adams-PC method	3.6	4.5	5.7	6.9	8.1	4
BRK method (4.9)—(4.11)	4.5	6.0	7.5	9.0	10.5	5
BRK method (4.12)—(4.13)	5.0	6.9	8.9	10.9	13.0	6

4.3. Predictor-Corrector method requiring four starting values

We have searched for two-processor predictors in the class of methods of the form

$$
\left[
\begin{array}{cccc|cccc}
1 & 0 & 0 & 0 & & & & \\
0 & 1 & 0 & 0 & & & & \\
0 & 0 & 1 & 0 & & & & \\
0 & 0 & 0 & 1 & & & & \\
\hline
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
a_{31} & a_{32} & a_{33} & a_{34} & b_{31} & b_{32} & b_{33} & b_{34} \\
a_{41} & a_{42} & a_{43} & a_{44} & b_{41} & b_{42} & b_{43} & b_{44}
\end{array}
\right], \qquad c = (-1, 0, c, 1)^{\mathsf{T}}.
$$

For a given value of c we can achieve order 7 by solving two linear systems of 8 equations each in 8 unknowns.

The corrector was chosen such that

$$
\left[
\begin{array}{cccc|cccccccc}
1 & 0 & 0 & 0 & & & & & & & & \\
0 & 1 & 0 & 0 & & & & & & & & \\
0 & 0 & 1 & 0 & & & & & & & & \\
0 & 0 & 0 & 1 & & & & & & & & \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{31} & a_{32} & 0 & 1-a_{31}-a_{32} & b_{31} & b_{32} & b_{33} & b_{34} & 0 & 0 & c_{33} & c_{34} \\
a_{41} & a_{42} & 0 & 1-a_{41}-a_{42} & b_{41} & b_{42} & b_{43} & b_{44} & 0 & 0 & c_{43} & c_{44}
\end{array}
\right], \qquad c = (-1, 0, c, 1)^{\mathsf{T}}.
$$

By this choice we achieve that the order conditions (3.1) simplify considerably. Given the value of c, this method can be made order 8 accurate in each component equation, again by solving two linear systems of 8 equations in 8 unknowns. These four systems of 8 equations have been solved numerically in terms of the parameter c and for a range of c-values we computed the real stability boundary β_{real} of the PECE mode. We found that β_{real} was maximal for $c \approx 2.58$ ($\beta_{\mathrm{real}} \approx 0.358$). In order to obtain a method with rational parameter values we chose $c = 5/2$ resulting in $\beta_{\mathrm{real}} \approx 0.302$. The corresponding

predictor is generated by

$$
\left[
\begin{array}{cccc|cccc}
1 & 0 & 0 & 0 & & & & \\
0 & 1 & 0 & 0 & & & & \\
0 & 0 & 1 & 0 & & & & \\
0 & 0 & 0 & 1 & & & & \\
\hline
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\dfrac{5975}{224} & \dfrac{1539}{20} & -\dfrac{537}{35} & -\dfrac{2793}{32} & \dfrac{225}{32} & \dfrac{567}{8} & 9 & \dfrac{2205}{32} \\
\dfrac{82}{343} & \dfrac{117}{125} & \dfrac{63232}{128625} & -\dfrac{2}{3} & \dfrac{3}{49} & \dfrac{18}{25} & -\dfrac{128}{1225} & 1
\end{array}
\right], \qquad c = (-1, 0, 5/2, 1)^{\mathsf{T}}.
\tag{4.14}
$$

and the corresponding corrector by

$$
\left[
\begin{array}{cccc|cccccccc}
1 & 0 & 0 & 0 & & & & & & & & \\
0 & 1 & 0 & 0 & & & & & & & & \\
0 & 0 & 1 & 0 & & & & & & & & \\
0 & 0 & 0 & 1 & & & & & & & & \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\dfrac{5^3 \cdot 7^3 \cdot 13 \cdot 83}{30469 \cdot 2^{10}} & \dfrac{3^6 \cdot 5^3 \cdot 263}{30469 \cdot 2^7} & 0 & \dfrac{3^6 \cdot 7^3 \cdot 827}{30469 \cdot 2^{10}} & \dfrac{3^3 \cdot 5^4 \cdot 7^3}{30469 \cdot 2^9} & \dfrac{3^6 \cdot 5^2 \cdot 7 \cdot 17 \cdot 67}{30469 \cdot 2^{10}} & \dfrac{3^5 \cdot 5^2 \cdot 7^3}{30469 \cdot 2^5} & \dfrac{3^5 \cdot 5 \cdot 7^3 \cdot 13}{30469 \cdot 2^5} & 0 & 0 & \dfrac{3^2 \cdot 5 \cdot 7 \cdot 809}{30469 \cdot 2^5} & \dfrac{3^3 \cdot 5^3 \cdot 7^3 \cdot 37}{30469 \cdot 2^{10}} \\
\dfrac{4549}{30469} & \dfrac{3^3 \cdot 1039}{30469} & 0 & \dfrac{3^3 \cdot 79}{30469} & \dfrac{23029}{30469 \cdot 3 \cdot 7} & \dfrac{3^3 \cdot 13 \cdot 1709}{30469 \cdot 5 \cdot 7} & \dfrac{2^8 \cdot 3^2 \cdot 31}{30469 \cdot 5 \cdot 7} & \dfrac{3^2 \cdot 61 \cdot 337}{30469 \cdot 5} & 0 & 0 & \dfrac{2^9 \cdot 11}{30469 \cdot 3 \cdot 5 \cdot 7} & \dfrac{14369}{30469}
\end{array}
\right]
$$

$$ c = (-1, 0, 5/2, 1)^{\mathsf{T}}. \tag{4.15} $$

Table 4.5 compares this method in PECE mode with the four-step Adams and four-step Shampine-Watts method.

Table 4.5. Correct decimal digits at $t = 1$ for problem (4.5) obtained by BRK methods in PECE mode with $k = 4$

Sequential righthand sides	6	12	24	48	96	order
Four-step Adams-PC method	3.3	4.8	6.4	7.9	9.5	5
Shampine-Watts pair $(2.7) - (2.6')$	3.6	4.8	6.0	7.2	8.4	4
BRK pair $(4.14) - (4.15)$	7.3	10.2	12.8			8

5. Summary of methods and numerical examples

The explicit, zero-stable methods and the PC combinations discussed in the preceding sections will be applied to a number of initial-value problems. In addition, we give the results obtained by the classical Adams formulas. First, however, we summarize the main characteristics of the various methods.

5.1. Summary of methods

Below we have listed a few important features such as the block point vector c, the order p, and the number of processors P_{opt} needed to implement the method with only one righthand side evaluation per step.

Table 5.1a. Survey of explicit one-stage BRK methods of the form (2.2')

Reference	c^{T}	P_{opt}	p	Remarks
MIRANKER-LINIGER [8]	$(2, 1)$	2	2	See (2.4)
	$(-1, 0, 2, 1)$	2	4	See (2.5)
SHAMPINE-WATTS [9]	$(-1/2, 0, 1/2, 1)$	1	3	See (2.7)
CHU-HAMILTON [3]	$(1/2, 1)$	2	3	See (2.8)
This paper	$(c, 1)$	2	3	See (4.1) with $c = 5/3$
	$(c, 1)$	2	3	See (4.3)
	$(c_1, c_2, 1)$	2	3	See (4.7)
	$(c_1, c_2, 1)$	2	4	See (4.7) with $(c_1, c_2) = (0, 17/10)$
	$(0, 1/2, 1)$	2	4	See (4.9)
	$(0, 4, 1)$	2	5	See (4.12)
	$(-1, 0, 5/2, 1)$	2	7	See (4.14)

Table 5.1b. Survey of implicit BRK methods of the form (2.2)

Reference	c^T	P_{opt}	p	Remarks
CLIPPINGER-DIMSDALE [1958]	$(1/2, 1)$	2	4	See (2.6)
CHU-HAMILTON [3]	$(1/2, 1)$	2	4	See (2.9)
This paper	$(1 - \sqrt{5}/5, 1)$	2	5	See (4.6)
	$(0, 1/2, 1)$	2	5	See (4.11)
	$(0, 4, 1)$	2	6	See (4.13)
	$(-1, 0, 5/2, 1)$	2	8	See (4.15)

Table 5.1c. Survey of PC pairs in $PE(CE)^r$ mode

Predictor	Corrector	c^T	p	r
(2.7)	(2.6')	$(-1/2, 0, 1/2, 1)$	4	1
(2.8)	(2.9)	$(1/2, 1)$	4	1
(4.3)	(4.6)	$(1 - \sqrt{5}/5, 1)$	5	2
(4.9)	(4.11)	$(0, 1/2, 1)$	5	1
(4.12)	(4.13)	$(0, 4, 1)$	6	1
(4.14)	(4.15)	$(-1, 0, 5/2, 1)$	8	1

5.2. Nonlinear problem with rapidly increasing solution

The first test problem is the nonlinear problem

$$y'(t) = -y^3 + t^9(10 + t^{21}), \qquad y(0) = 0, \qquad 0 \le t \le 1, \tag{5.1}$$

with exact solution $y(t) = t^{10}$. In Table 5.2 the results are listed. Since the number of sequential righthand side evaluations per step varies from 1 to 3 for the various methods, we adapted the stepsize as to obtain that each column of this table contains results with an equal number of sequential righthand side evaluations over the whole integration interval.

Table 5.2. Correct decimal digits at $t = 1$ for problem (5.1)

Sequential righthand sides	6	12	24	48	96	order
Two-step Adams-Bashforth method	0.3	0.8	1.3	1.9	2.5	2
Miranker-Liniger method (2.4)	0.6	1.2	1.9	2.5	3.1	2
BRK method (4.1): $c = 5/3$	2.6	2.4	3.1	3.9	4.8	3
BRK method (4.3): $c = 1 - \sqrt{6}$	0.5	1.2	2.0	2.9	3.8	3
Two-step Adams pair: PECE	0.2	0.9	1.7	2.5	3.4	3
Chu-Hamilton pair (4.3)−(4.6): PECE, $c = 1/2$	1.1	1.9	3.0	4.2	5.5	4
BRK pair (4.3)−(4.6): $PE(CE)^2$, $c = 1 - \sqrt{5}/5$	2.0	2.9	4.1	5.7	7.4	5
Three-step Adams-Bashforth method	0.5	1.1	1.9	2.7	3.6	3
Method (4.7): $(c_1, c_2) = (0, 17/10)$	2.0	2.6	3.7	4.8	6.0	4
Three-step Adams pair: PECE	0.3	1.1	2.1	3.3	4.5	4
BRK pair (4.9)−(4.11): PECE	1.2	2.2	3.6	5.1	6.7	5
BRK pair (4.12)−(4.13): PECE	*	*	1.5	5.3	7.4	6
Four-step Adams-Bashforth method	0.6	1.4	2.5	3.6	4.8	4
Miranker-Liniger method (2.5)	1.1	2.3	3.5	4.7	5.9	4
Four-step Adams pair: PECE	1.3	2.6	4.0	5.5	7.0	5
Shampine-Watts pair (2.7)−(2.6'): PECE	1.1	1.8	2.9	4.1	5.3	4
BRK pair (4.14)−(4.15): PECE	*	1.3	5.6	9.0	11.6	8

A first observation is that most parallel methods behave more efficient than the corresponding one-processor Adams methods, showing that already on two-processor machines parallelism can be exploited. Furthermore, these results clearly demonstrate the superiority of the high-order methods, especially the 6th- and the 8th-order BRK methods. It should be remarked that these two methods produce unstable results (indicated by an '*' in Table 5.2) for large stepsizes, in spite of their large real stability boundary. The reason is that these methods employ a block point $t_n + ch$, with c much larger than 1, viz. $c = 4$ and $c = 5/2$, respectively. Since the modulus of $\partial f / \partial y$, which determines the maximally allowed stepsize, is a rapidly increasing function of t (viz. $3 \cdot t^{20}$), it is clear that an evaluation of f beyond the endpoint $t = 1$ may easily cause instabilities.

5.3. Orbit equation

The second problem was taken from the test set of HULL et al. [7]:

$$\begin{aligned}
y_1' &= y_3, & y_1(0) &= 1 - \varepsilon, & \varepsilon &= 0.3, \\
y_2' &= y_4, & y_2(0) &= 0, \\
y_3' &= -y_1(y_1^2 + y_2^2)^{-3/2}, & y_3(0) &= 0, \\
y_4' &= -y_2(y_1^2 + y_2^2)^{-3/2}, & y_4(0) &= \sqrt{\frac{1 + \varepsilon}{1 - \varepsilon}}.
\end{aligned} \tag{5.2}$$

For this example, which describes a system of ODEs, the errors are measured in the maximum norm. Since most methods nicely show their asymptotic order behaviour, the high-order BRK methods are again superior to the low-order ones. Hence, the conclusion can be drawn that the introduction of non-equally spaced block points $t_n + c_j h$ favourably influences the performance of the BRK methods.

Table 5.3. Correct decimal digits at $t = 20$ for problem (5.2)

Sequential righthand sides	240	480	960	1920	3840	order
Two-step Adams-Bashforth method	0.3	0.7	1.2	1.7	2.3	2
Miranker-Liniger method (2.4)	0.5	2.1	2.1	2.5	3.1	2
BRK method (4.1): $c = 5/3$	0.3	1.2	2.1	3.0	3.9	3
BRK method (4.3): $c = 1 - \sqrt{6}$	0.3	1.2	2.1	3.0	3.9	3
Two-step Adams pair: PECE	−0.1	0.6	1.4	2.3	3.2	3
Chu-Hamilton pair (4.3)−(4.6): PECE, $c = 1/2$	−1.5	0.1	3.7	5.2	6.5	4
BRK pair (4.3)−(4.6): PE(CE)^2, $c = 1 - \sqrt{5}/5$	1.4	3.2	4.8	6.4	7.9	5
Three-step Adams-Bashforth method	0.1	1.0	1.9	2.8	3.7	3
Method (4.7): $(c_1, c_2) = (0, 17/10)$	1.9	3.5	4.4	5.5	6.7	4
Three-step Adams pair: PECE	0.4	1.8	3.4	5.0	6.2	4
BRK pair (4.9)−(4.11): PECE	1.3	2.8	4.4	5.9	7.4	5
BRK pair (4.12)−(4.13): PECE	3.3	4.9	6.8	8.6	9.6	6
Four-step Adams-Bashforth method	1.4	2.3	3.4	4.6	5.8	4
Miranker-Liniger method (2.5)	2.0	4.4	4.8	5.8	6.9	4
Four-step Adams pair: PECE	0.8	2.0	3.5	5.0	6.5	5
Shampine-Watts pair (2.7)−(2.6'): PECE	1.1	2.9	4.1	5.1	6.2	4
BRK pair (4.14)−(4.15): PECE	3.9	6.8	9.0			8

5.4. Euler's equation of motion

The third problem is Euler's equation of motion (cf. HULL et al. [7]):

$$
\begin{aligned}
y_1' &= y_2 y_3, & y_1(0) &= 0, \\
y_2' &= -y_1 y_3, & y_2(0) &= 1, \\
y_3' &= -0.51 y_1 y_2, & y_3(0) &= 1.
\end{aligned}
\tag{5.3}
$$

The results in Table 5.4 give rise to the same conclusions as formulated for the previous test problems.

Table 5.4. Correct decimal digits at $t = 20$ for problem (5.3)

Sequential righthand sides	120	240	480	960	1920	order
Two-step Adams-Bashforth method	1.2	1.9	2.5	3.1	3.7	2
Miranker-Liniger method (2.4)	1.6	2.4	3.1	3.8	4.4	2
BRK method (4.1): $c = 5/3$	1.7	2.6	3.5	4.4	5.3	3
BRK method (4.3): $c = 1 - \sqrt{6}$	1.6	2.6	3.5	4.4	5.3	3
Two-step Adams pair: PECE	1.2	2.0	2.9	3.8	4.7	3
Chu-Hamilton pair (4.3)−(4.6): PECE, $c = 1/2$	*	3.3	4.7	6.0	7.3	4
BRK pair (4.3)−(4.6): PE(CE)^2, $c = 1 - \sqrt{5}/5$	2.5	3.9	5.5	7.0	8.5	5
Three-step Adams-Bashforth method	1.5	2.4	3.3	4.2	5.1	3
Method (4.7): $(c_1, c_2) = (0, 17/10)$	2.8	4.1	5.4	6.6	7.9	4
Three-step Adams pair: PECE	1.4	2.7	4.0	5.3	6.5	4
BRK pair (4.9)−(4.11): PECE	2.7	4.1	5.6	7.1	8.6	5
BRK pair (4.12)−(4.13): PECE	3.2	5.1	6.9	8.7	10.7	6
Four-step Adams-Bashforth method	3.3	3.8	4.8	6.0	7.1	4
Miranker-Liniger method (2.5)	3.1	5.0	6.3	7.2	8.3	4
Four-step Adams pair: PECE	2.5	3.4	4.8	6.2	7.7	5
Shampine-Watts pair (2.7)−(2.6'): PECE	1.9	3.3	4.6	5.9	7.2	4
BRK pair (4.14)−(4.15): PECE	2.9	7.4	9.8			8

To sum up, these examples clearly show that, even when only 2 processors are used, a substantial gain in efficiency can be obtained when compared with sequential (uniprocessor) methods. This especially holds for the high-order BRK methods.

References

1 BUTCHER, J. C.: A modified multistep method for the numerical integration of ordinary differential equations. J. ACM **12** (1965), 124–135.

2 BUTCHER, J. C.:The numerical analysis of ordinary differential equations, Runge-Kutta and general linear methods. Wiley, New York 1987.

3 CHU, M. T.; HAMILTON, H.: Parallel solution of ODE's by multi-block methods. SIAM J. Sci. Statist. Comput. **3** (1987), 342–353.

4 DONELSON, J.; HANSEN, E.: Cyclic composite multistep predictor-corrector methods. SIAM J. Numer. Anal. **8** (1971), 137–157.

5 GEAR, C. W.: Parallel methods for ordinary differential equations. Calcolo **25** (1988), 1–20.

6 HOUWEN, P. J. VAN DER; SOMMEIJER, B. P.; MOURIK, P. A. VAN: Note on explicit parallel multistep Runge-Kutta methods. J. Comput. Appl. Math. **27** (1989), 411–420.

7 HULL, T. E.; ENRIGHT, W. H.; FELLEN, B. M.; SEDGWICK, A. E.: Comparing numerical methods for ordinary differential equations. SIAM J. Numer. Anal. **9** (1972), 603–637.

8 MIRANKER, W. L.; LINIGER, W.: Parallel methods for the numerical integration of ordinary differential equations. Math. Comput. **21** (1967), 303–320.

9 SHAMPINE, L. F.; WATTS, H. A.: Block implicit one-step methods. Math. Comput. **23** (1969), 731–740.

10 WORLAND, P. B.: Parallel methods for the numerical solution of ordinary differential equations. IEEE Trans. Comput. **C-25** (1976), 1045–1048.

Address: Prof. Dr. PETER J. VAN DER HOUWEN, Drs. B. P. SOMMEIJER, Centre for Mathematics and Computer Science, Post Box 4079, NL-1009AB Amsterdam, The Netherlands