Applied Numerical Mathematics 9 (1992) 267–281
North-Holland

A-stable parallel block methods for ordinary and integro-differential equations

B.P. Sommeijer, W. Couzy and P.J. van der Houwen
Centre for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, Netherlands

Abstract

Sommeijer, B.P., W. Couzy and P.J. van der Houwen, A-stable parallel block methods for ordinary and integro-differential equations, Applied Numerical Mathematics 9 (1992) 267–281.

In this paper we study the stability of a class of block methods which are suitable for integrating ordinary and integro-differential equations on parallel computers. A-stable methods of orders 3 and 4 and A(α)-stable methods with $\alpha > 89.9°$ of order 5 are constructed. On multiprocessor computers these methods are of the same computational complexity as implicit linear multistep methods on one-processor computers.

1. Introduction

Many algorithms for numerically solving initial-value problems for ordinary differential equations (ODEs):

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = f(t, y(t)), \quad y(t_0) = y_0, \tag{1.1}$$

or Volterra integro-differential equations (VIDEs):

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = f\left(t, y(t), \int_{t_o}^{t} k(t, x, y(x))\,\mathrm{d}x\right), \quad y(t_0) = y_0, \tag{1.2}$$

are based on implicit linear multistep methods (LM methods), in particular on backward differentiation methods (BDF methods). The main reason for their popularity is the relatively low computational effort per step, at least when compared with other suitable methods for stiff equations, such as implicit Runge–Kutta methods. However, the BDFs have one serious disadvantage: they are subject to the so-called "second Dahlquist barrier", which says that the order cannot exceed two if the method has to be A-stable. Thus the higher-order BDFs lack the property of A-stability. This means that if a high-order formula is selected (dictated by accuracy considerations), then it may happen that—for certain types of stiff ODEs or VIDEs—the algorithm encounters stability problems which usually results in a dramatical degradation of the performance. To circumvent this behaviour it is highly desirable to have A-stable methods of high order without increasing the computational effort per step.

0168-9274/92/$05.00 © 1992 – Elsevier Science Publishers B.V. All rights reserved

It is our aim to construct such methods. They are most easily formulated as so-called block methods. Block methods can be considered as a set of simultaneously applied linear multistep methods to obtain several numerical approximations within one application. Numerous block methods have been proposed in the literature including high-order A-stable ones (see e.g. Watts and Shampine [16]). However, these implicit methods require in each application an amount of work which by far exceeds the computational effort required by a BDF. In recent papers (cf. e.g. Chu and Hamilton [3]), block methods have been given which solve the huge implicit relations on a *parallel* computer which indeed significantly reduces the computational costs. However, all these techniques follow the approach of predictor–corrector iteration, which in fact restricts their application to nonstiff problems.

Like Chu and Hamilton, we will employ parallelism to obtain the aforementioned goals. We shall construct A-stable methods of orders three and four, and A(α)-stable methods of order five with $\alpha \approx \pi/2$. Furthermore, by carefully segmenting the total work per step into a few subtasks of approximately equal computational length, these methods require an amount of work which is very similar to what a BDF requires when implemented on a uni-processor machine. In Section 5.3 we will see that a high degree of parallelization is obtained. Since the implicit relations are solved by a Newton-type process (as is the case in BDF implementations) rather than a predictor–corrector fashion, the property of A-stability is preserved.

In Sections 2 and 3, we present the construction of block methods for ODEs, in Section 4, block methods for VIDEs employing these block ODE solvers are discussed, and in Section 5, numerical experiments are reported. The way of construction is based on extremely simple tools: firstly, certain order-conditions are imposed such that a number of parameters are left free, and secondly, a numerical search over the free parameters is carried out to give the method the optimal stability characteristics. So far, we did not succeed in developing more sophisticated search techniques by analytical means.

2. Parallel block methods for ODEs

In order to simplify the formulas, we present the derivations of the block methods for scalar, autonomous ODEs. The extension of these methods to systems of ODEs, and therefore also to non-autonomous equations, is straightforward.

The block methods studied in this paper are a direct generalization of the implicit one-step method

$$y_{n+1} = ay_n + hbf(y_n) + hdf(y_{n+1}), \quad n = 0, 1, \ldots, \tag{2.1}$$

where h is the stepsize and y_n an approximation to $y(t_n)$. By introducing block vectors

$$\boldsymbol{Y}_{n+1} := (y_{n,1}, \ldots, y_{n,k})^{\mathrm{T}}, \qquad \boldsymbol{c} := (c_1, \ldots, c_k)^{\mathrm{T}}, \quad c_k = 1, \tag{2.2}$$

where $y_{n,i}$ denotes a numerical approximation to the exact solution value $y(t_n + c_i h)$, and assuming that (1.1) is a scalar equation, we can define the block method

$$\boldsymbol{Y}_{n+1} = A\boldsymbol{Y}_n + hBf(\boldsymbol{Y}_n) + hDf(\boldsymbol{Y}_{n+1}), \tag{2.3}$$

where A, B and D are k-by-k matrices. Here 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). A characteristic of these methods is that, unlike conventional block methods based on linear multistep methods, the block point vector c is allowed to have $k-1$ *non-integer* components. In order to start the method, one needs the initial vector Y_0, which requires, in general, as many starting values as there are distinct values c_j $(j = 1, \ldots, k)$. Notice that the last component of Y_{n+1} contains the step point value y_{n+1}. Furthermore, we remark that, in general, $y_{n,i} \neq y_{m,j}$, even if $n + c_i = m + c_j$.

The method (2.3) is suitable for direct use on parallel computers if the matrix D is *diagonal*, since such a form uncouples the various components as far as implicitness is concerned; the corresponding methods will be called *parallel block methods*. Using k processors, each processor has to evaluate a component of $f(Y_n)$ and to solve a system of equations whose dimension is that of the system of ODEs (1.1). If Newton's method is used for solving the system of equations, then each processor needs the Jacobian matrix $I - hd_{jj}\partial f/\partial y$ and its LU-decomposition. Either the various processors have to compute the data they need themselves, or one may consider the use of additional processors for computing the Jacobian matrices and their LU-decompositions. Let us consider the second strategy. As soon as the additional processors have completed an update of the matrix $\partial f/\partial y$ and computed the LU-decompositions of the k matrices $I - hd_{jj}\partial f/\partial y$, then the first k processors can replace their data by the new data. However, usually the computational job of computing Jacobian matrices and LU-decompositions is so substantial that the speed of updating may not be great enough. In such cases, the use of matrices D with equal diagonal elements is recommendable, because then the Jacobian matrices $I - hd_{jj}\partial f/\partial y$ are all identical, so that only one instead of k decompositions are required. Therefore, methods where D is of the form dI, I being the identity matrix, have some advantage.

If D is a full matrix, then the block method is not directly suitable for use on parallel computers. However, (2.3) allows the application of an iteration process that has a high degree of parallelism. This iteration method is of the one-level form

$$\left[I - hC\frac{\partial f(y_n)}{\partial y}\right]Y^{(j+1)} - hEf(Y^{(j+1)})$$

$$= AY_n + hBf(Y_n) - hC\frac{\partial f(y_n)}{\partial y}Y^{(j)} + h[D-E]f(Y^{(j)}),$$

where C and E are suitable iteration matrices. There are several possibilities for choosing these matrices in order to achieve parallelism and to preserve stability. We mention: (i) C diagonal and $E = O$ (linear diagonal iteration), (ii) $C = O$ and E diagonal (nonlinear diagonal iteration), and (iii) $C = D$, $E = O$ combined with diagonalization of C (diagonalized Newton). A survey of properties of diagonal iteration in the case where (2.3) corresponds to Runge–Kutta methods can be found in [10]. The diagonalized Newton process was proposed by Lubich [12]. In passing we remark, that one might also consider higher-level iteration methods. For example, the "pipeline" iteration proposed by Feldstein [5] fits into the family of three-level iteration methods.

In a forthcoming paper, we will study the above iteration process if the matrix D in (2.3) is a full matrix. In this paper we always assume that D is diagonal.

The conditions for pth-order consistency for methods of the form (2.3) are extremely simple and read (cf. [9])

$$\begin{aligned} &C_j = \mathbf{0}, \quad j = 0, 1, \ldots, p. \\ &C_0 := Ae - e; \qquad C_1 := A(c-e) + Be + De - c; \\ &C_j := A(c-e)^j + j\left[B(c-e)^{j-1} - Dc^{j-1}\right] - c^j, \quad j = 2, 3, \ldots, \end{aligned} \tag{2.4}$$

where e denotes the vector with unit entries and where powers of vectors are meant to be componentwise powers.

In order to compare the components of these vectors with the error constants corresponding to conventional linear multistep methods, we introduce the *normalized* error vectors [8]

$$E_j := \frac{C_j}{j!(B+D)e}, \tag{2.5}$$

where the division of vectors is meant componentwise. When a linear k-step method is written in the form (2.3) with $c = (-k+2, \ldots, -2, -1, 0, 1)^T$, then the last component of E_j equals the normalized error constant of the linear k-step method. Since these block methods are in fact a composition of k conventional linear multistep methods, the theory developed for the latter class of methods (see Henrici [8] or Hairer, Nørsett and Wanner [7]), is to a large extent also applicable in the case of block methods. In particular, this theory can be used to determine the order of convergence of the block methods, that is the behaviour of $Y_{n+1} - Y(t_{n+1}) := (y(t_n + c_1 h), y(t_n + c_2 h), \ldots, y(t_n + h))^T$ for $h \to 0$ and $t_n = nh$ fixed (see also the paper by Cooper [4]).

3. Stability

The (linear) stability of block methods can be investigated by applying the method to the test equation $y' = \lambda y$. This will lead to a recursion of the form

$$Y_{n+1} = M(z)Y_n, \qquad M(z) := [I - zD]^{-1}[A + zB], \quad z := \lambda h. \tag{3.1}$$

M will be called the *amplification matrix* and its eigenvalues the *amplification factors*. Here we observe that, by requiring the elements of the diagonal matrix D to be positive, the matrix $I - zD$ is nonsingular for all z on the negative real axis. Therefore, in the sequel we will assume that the (diagonal) elements of D are positive.

In our stability analysis we shall use the following result on the power of a matrix N (cf. [15, p. 65]).

$$\| N^n \| = O\left(n^{q-1}[\rho(N)]^n\right) \quad \text{as } n \to \infty, \tag{3.2}$$

where $\| \cdot \|$ and $\rho(N)$ are the spectral norm and radius of N and where all diagonal submatrices of the Jordan normal form of N which have spectral radius $\rho(N)$ are at most q-by-q. If $\rho(N) < 1$ or $\rho(N) = q = 1$, then we call N *power bounded*.

Following the familiar stability definitions used for RK and LM methods, we shall call the region where the amplification matrix $M(z)$ is power bounded, the *stability region* of the block

method. If the stability region contains the origin, then the method is called *zero-stable*. The region where $\| M^n \|$ tends to zero will be called the *strong stability region*. If the (strong) stability region of a block method contains the left half plane, then the block method is called *(strongly) A-stable*. Furthermore, if the amplification matrix of an A-stable method has vanishing eigenvalues at infinity, then the method is called *L-stable*. For some methods (i.e., the BDF methods) a less demanding definition of stability is more appropriate. Therefore the notion of A(α)-stability has been introduced. The angle α defines a wedge in the left half plane and the method is stable if z lies inside this wedge. This is, however, a rather crude way to describe the stability region, since for the higher-order BDF methods the part of the left half plane which is not included in the stability region is a small lobe near the imaginary axis. To provide more detailed information on the stability region, we introduce two additional parameters leading to the notion of A(α, β, γ)-stability:

Definition 3.1. A method is said to be A(α, β, γ)-*stable* if (i) its region of stability contains the infinite wedge $\{z\colon -\alpha < \pi - \arg(z) < \alpha\}$, $0 < \alpha \leqslant \pi/2$, and all points in the nonpositive half plane with $|z| > \beta$, and (ii) $1 + \gamma$ is the maximum value of the spectral radius of $M(z)$ when z runs through the region of instability lying in the nonpositive half plane.

Note that A($\pi/2$, 0, 0)-stability implies A-stability. The degree of instability of the method is measured by γ.

If we set $A = D = I$ and $B = O$ in (2.3), then the method reduces to a set of k completely uncoupled one-step methods of the backward Euler type, each advancing the solution from $t_{n-1} + c_i h$ to $t_n + c_i h$ $(i = 1, 2, \ldots, k)$. Evidently, these k formulas can be efficiently implemented on a k-processor machine (in fact, they could equally well run on k separate computers). Such methods have excellent stability properties (e.g., the property of L-stability), but are only of first order. However, by using full matrices A and B, that is the k formulas of the block method share the same information from the previous step, the order can be considerably increased. In the next two subsections, we investigate for $k = 2$ ("two-dimensional block methods") and $k = 3$ ("three-dimensional block methods") to what values the order can be raised while preserving the favourable stability properties of backward Euler (stability plots may be found in [14]).

3.1. Two-dimensional block methods

First we consider the case $k = 2$ and choose the coefficient matrices of the form

$$A = \begin{pmatrix} a_1 & 1 - a_1 \\ a_2 & 1 - a_2 \end{pmatrix}, \qquad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}, \qquad D = \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix}, \qquad \boldsymbol{c} = (c, 1)^{\mathrm{T}}. \tag{3.3}$$

Imposing the conditions for second-order consistency we can express the entries of the matrix B in terms of the five free parameters c, a_1, a_2, d_1 and d_2:

$$b_{j1} = \tfrac{1}{2}(1 - c)a_j + \frac{c_j(2d_j - c_j)}{2(1 - c)}, \quad b_{j2} = c_j + (1 - c)a_j - b_{j1} - d_j, \qquad j = 1, 2, \tag{3.4a}$$

where $c_1 = c$ and $c_2 = 1$. The components C_{ij} of the vectors $\boldsymbol{C}_i$ $(i \geqslant 3)$ are given by

$$C_{ij} = (1 - \tfrac{1}{2}i)(c-1)^i a_j + i c_j^{i-1} d_j + \tfrac{1}{2} i c_j (c_j - 2d_j)(c-1)^{i-2} - c_j^i, \quad j = 1, 2.$$

An elementary calculation shows that C_{3j} vanishes if

$$a_j = \frac{c_j}{(c-1)^3}\left[3(c-1)(c_j - 2d_j) + 2c_j(3d_j - c_j)\right], \tag{3.4b}$$

and that C_{4j} also vanishes if, in addition,

$$d_1 = \frac{c}{2(c+1)}, \qquad d_2 = \frac{c-2}{2(c-3)}. \tag{3.4c}$$

The characteristic equation of the amplification matrix in (3.1) can be written in the form

$$\begin{aligned} p(\zeta, z) &:= \det[A + zB - \zeta(I - zD)] \\ &= \det\begin{pmatrix} a_1 + b_{11}z - \zeta(1 - d_1 z) & 1 - a_1 + b_{12}z \\ a_2 + b_{21}z & 1 - a_2 + b_{22}z - \zeta(1 - d_2 z) \end{pmatrix} = 0. \end{aligned} \tag{3.5}$$

We shall determine the z-region where this polynomial has its roots ζ within the unit circle, that is, the region of strong stability. In addition, we should impose the condition of zero-stability, i.e., the condition that the two eigenvalues $\alpha = 1$ and $\alpha = a_1 - a_2$ of A are on the unit disk hose on the unit circle being simple, i.e.,

$$-1 \leqslant a_1 - a_2 < 1. \tag{3.6}$$

A further restriction on the range of the free parameters is obtained by imposing the "stability at infinity" condition. By this we mean that the roots of the polynomial $P(\zeta, \infty)$ are on the unit disk (which is of course anyhow a necessary condition for A-stability). By virtue of the Hurwitz criterion we obtain (recall that d_1 and d_2 are assumed to be positive)

$$|b_{11}d_2 + b_{22}d_1| \leqslant d_1 d_2[d_1 d_2 + \det(B)], \quad \det(B) \leqslant d_1 d_2. \tag{3.7}$$

3.1.1. Second-order methods

If we are satisfied with second-order accuracy, then we may choose the free parameters a_j and d_j in (3.4a) such that the matrix B vanishes while preserving the property of A-stability. For example, if $c = 0$ then the method is equivalent with the familiar two-step backward differentiation formula generated by

$$A = \begin{pmatrix} 0 & 1 \\ -\frac{1}{3} & \frac{4}{3} \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \qquad D = \begin{pmatrix} 0 & 0 \\ 0 & \frac{2}{3} \end{pmatrix}, \qquad \boldsymbol{c} = (0, 1)^{\mathrm{T}}. \tag{3.8}$$

3.1.2. Third-order methods

Third-order accuracy is achieved by choosing $C_{31} = C_{32} = 0$, leaving us with three free

parameters for monitoring the stability of the method. We find

$$a_1 = \frac{c(c^2 - 3c + 6d_1)}{(c-1)^3}, \qquad a_2 = \frac{3c + 12d_2 - 6cd_2 - 5}{(c-1)^3},$$
$$b_{11} = \frac{c^2 - 2cd_1 - c^2 d_1}{(c-1)^2}, \qquad b_{12} = \frac{c - 2cd_1 - d_1}{(c-1)^2}, \tag{3.9}$$
$$b_{21} = \frac{2 - 5d_2 - c + 2cd_2}{(c-1)^2}, \qquad b_{22} = \frac{(c-2)^2 - d_2(c^2 - 6c + 8)}{(c-1)^2},$$

leaving c, d_1 and d_2 as the free parameters. Taking into account the conditions of zero-stability and "stability at infinity" (conditions (3.6) and (3.7)), we performed a numerical search in the (c, d_1, d_2)-space. It turned out that the regions of A-stable (c, d_1, d_2)-values are so small that A-stable points and strongly unstable points are close together, that is, a small perturbation of these values causes the method to violate the A-stability conditions. For example, the values

$$c = 0.917387, \qquad d_1 = 0.319523, \qquad d_2 = 0.347067, \tag{3.10}$$

generate such a "marginally" A-stable method. There is, however, an alternative approach. It is easily verified that putting $a_2 = C_{32} = 0$ yields methods providing third-order approximations at the step points t_n and second-order approximations at the points $t_n + ch$. It turns out that in the space of free parameters the regions of A-stable methods are larger so that it is easier to find A-stable methods by a numerical search. For example, we found the A-stable, third-order method

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} \frac{147}{220} & \frac{161}{220} \\ -\frac{50}{33} & \frac{23}{66} \end{pmatrix}, \qquad D = \begin{pmatrix} \frac{7}{10} & 0 \\ 0 & \frac{13}{6} \end{pmatrix}, \qquad c = \tfrac{1}{10}(21, 10)^{\mathrm{T}} \tag{3.11}$$

with the normalized error vectors $E_3 \approx (0.19, 0)^{\mathrm{T}}$ and $E_4 \approx (0.20, -0.017)^{\mathrm{T}}$. The amplification factors at the origin equal 0 and 1, and the maximal amplification factor at infinity is ≈ 0.94.

3.1.3. Fourth-order methods

Fourth-order accuracy for both components is obtained by choosing $C_{31} = C_{32} = C_{41} = C_{42} = 0$. Alternatively, replacing $C_{41} = 0$ by $a_2 = 0$, reduces the order of the first component to 3, without affecting the order of the second component. In both approaches we are left with one free parameter for monitoring the stability of the method. Unfortunately, the stability regions of these fourth-order methods are rather limited and do not even allow for A(α)-stability. Thus, in the class (3.3) the fourth-order methods seem to be of no interest.

3.2. Three-dimensional block methods

For $k = 3$ we expect to find A-stable methods of order four and we may hope for A(α)-stable methods of order five. These two cases will be investigated in the following subsections.

3.2.1. Fourth-order methods

Let us choose the matrix A such that $a_{i3} = 1 - a_{i1} - a_{i2}$, $i = 1, 2, 3$, so that C_0 vanishes. The vectors C_j vanish for $j = 1, 2, 3, 4$ if the entries b_{ij} and d_j satisfy the linear systems

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ c_1 - 1 & c_2 - 1 & 0 & c_i \\ (c_1 - 1)^2 & (c_2 - 1)^2 & 0 & c_i^2 \\ (c_1 - 1)^3 & (c_2 - 1)^3 & 0 & c_i^3 \end{pmatrix} \begin{pmatrix} b_{i1} \\ b_{i2} \\ b_{i3} \\ d_i \end{pmatrix} = \begin{pmatrix} c_i - a_{i1}(c_1 - 1) - a_{i2}(c_2 - 1) \\ \frac{1}{2}\left[c_i^2 - a_{i1}(c_1 - 1)^2 - a_{i2}(c_2 - 1)^2\right] \\ \frac{1}{3}\left[c_i^3 - a_{i1}(c_1 - 1)^3 - a_{i2}(c_2 - 1)^3\right] \\ \frac{1}{4}\left[c_i^4 - a_{i1}(c_1 - 1)^4 - a_{i2}(c_2 - 1)^4\right] \end{pmatrix},$$

$$i = 1, 2, 3. \tag{3.12}$$

This shows that there is a family of fourth-order block methods with eight free parameters: a_{i1}, a_{i2} $(i = 1, 2, 3)$, c_1 and c_2.

In order to ensure zero-stability, we require that A has its two parasitic eigenvalues within the unit circle. Writing the characteristic equation of A in the form $(\zeta - 1)(\zeta^2 + q_0\zeta + r_0) = 0$, we find that we have zero-stability if

$$|q_0| < r_0 + 1, \qquad r_0 < 1, \qquad q_0 := a_{31} + a_{32} - a_{11} - a_{22},$$
$$r_0 := a_{11}a_{12} + a_{31}a_{12} + a_{32}a_{21} - a_{11}a_{32} - a_{21}a_{12} - a_{22}a_{31}. \tag{3.13}$$

Taking this constraint into account, we performed a numerical search over the free parameters to obtain the A-stable method

$$A = \begin{pmatrix} a - 1 & \frac{1}{2} & \frac{3}{2} \\ \frac{1}{2} & 1 & -\frac{1}{2} \\ -1 & \frac{1}{2} & \frac{3}{2} \end{pmatrix}, \qquad B = \begin{pmatrix} \frac{5 \cdot 13 \cdot 43}{2^{11}} & \frac{15161}{2^5 \cdot 3^2 \cdot 11} & \frac{29 \cdot 43 \cdot 83}{2^{11} \cdot 3^2 \cdot 5} \\ \frac{-73}{2 \cdot 3^2 \cdot 7} & \frac{-467}{2 \cdot 3^3 \cdot 7} & \frac{-7 \cdot 37}{2 \cdot 3^3 \cdot 13} \\ \frac{5 \cdot 16069}{2^{11} \cdot 3^2 \cdot 7} & \frac{54419}{2^5 \cdot 3^3 \cdot 5 \cdot 7} & \frac{41927}{2^{11} \cdot 3^3} \end{pmatrix},$$

$$D = \begin{pmatrix} \frac{13 \cdot 1303}{2^9 \cdot 5 \cdot 11} & 0 & 0 \\ 0 & \frac{277}{2 \cdot 3^2 \cdot 13} & 0 \\ 0 & 0 & \frac{16001}{2^9 \cdot 3^2 \cdot 5} \end{pmatrix}, \qquad c = (5, 13/4, 1)^{\mathrm{T}} \tag{3.14}$$

with normalized error vector $E_5 \approx (0.13, 0.27, 0.075)^{\mathrm{T}}$. Its amplification factors at the origin are 0, $\frac{1}{2}$ and 1, and at infinity the maximal amplification factor is ≈ 0.92.

The above direct search method is rather expensive, and therefore we also applied an alternative approach where

$$\sum_{i=1}^{m} \sum_{j=1}^{k} |\mu_{ij}|^{q_{ij}} \tag{3.15}$$

was minimized over the free parameters b_{i2} and d_i $(i = 1, 2, 3)$, c_1 and c_2. Here, $k = 3$, the q_{ij} are control parameters and μ_{ij}, $j = 1, \ldots, k$, denote the eigenvalues of the amplification matrix

$M(z_i)$ defined in (3.1) with z_i running through a set of m points lying on the imaginary axis. In this way we found the A-stable method

$$A = \tfrac{1}{1600}\begin{pmatrix} 2820 & -183 & -1037 \\ -7100 & -3423 & 12123 \\ -1020 & -1607 & 4227 \end{pmatrix}, \quad B = \tfrac{1}{400}\begin{pmatrix} -398 & -92 & -177 \\ 6282 & -92 & 2143 \\ 1098 & 272 & 507 \end{pmatrix},$$

$$D = \tfrac{1}{5}\begin{pmatrix} 8 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 8 \end{pmatrix}, \quad c = (3, 5, 1)^{\mathrm{T}} \tag{3.16}$$

with normalized error vector $E_5 \approx (3.67, 0.19, 0.064)^{\mathrm{T}}$. At the origin the amplification factors are 0.81, 0.81 and 1, and at infinity the maximal amplification factor is ≈ 0.37.

3.2.2. Fifth-order methods

Along the same lines as we constructed the fourth-order method (3.16), we proceeded with the fifth-order case. Now only five free parameters are available, say d_i $(i = 1, 2, 3)$, c_1 and c_2. Imposing the constraint (3.13), we found a few A(α, β, γ)-stable methods which may be considered as A-stable in most practical applications.

We mention the A(α, β, γ)-stable method with $\alpha = 89.9988°$, $\beta \approx 0.16$ and $\gamma = 2.6 \cdot 10^{-6}$ generated by

$$A = \begin{pmatrix} -0.37354856915573 & 1.3772028209449 & -0.0036542517891531 \\ 0.45636214490330 & 0.58957191150098 & -0.045934056404276 \\ -71.558907928027 & 69.945110840701 & 2.6137970873262 \end{pmatrix},$$

$$B = \begin{pmatrix} -0.089579683013023 & -0.020791477924637 & 0.0023118793010643 \\ 0.037434812789650 & 0.78549538208108 & 0.024702269787981 \\ -18.279469309687 & -29.674965823418 & -1.6401568285440 \end{pmatrix},$$

$$D = \begin{pmatrix} 0.261 & 0 & 0 \\ 0 & 0.581 & 0 \\ 0 & 0 & 0.832 \end{pmatrix}, \quad c = \begin{pmatrix} -2.747 \\ -2.122 \\ 1 \end{pmatrix} \tag{3.17}$$

with normalized error vector $E_6 \approx (0.007, 0.0038, -0.015)^{\mathrm{T}}$. At the origin the amplification factors are 0.92, 0.92, and 1, and at infinity the maximal amplification factor is ≈ 0.993.

Finally, we present the A(α, β, γ)-stable method with $\alpha \approx 89.98°$, $\beta \approx 0.30$ and $\gamma = 6.9 \cdot 10^{-5}$ generated by

$$A = \begin{pmatrix} 0.58694824150708 & -0.042737729478577 & 0.45578948797150 \\ 73.394943213338 & 2.5499812910344 & -74.944924504372 \\ 1.3881897627759 & -0.0035265226034516 & -0.38466324017241 \end{pmatrix},$$

$$B = \begin{pmatrix} 0.78434821208875 & 0.023439431423946 & 0.033345158796322 \\ -30.332265183768 & -1.5938561820999 & -18.934741340575 \\ -0.012761141648945 & 0.0022604702667178 & -0.092097195902230 \end{pmatrix},$$

$$D = \begin{pmatrix} 0.57487 & 0 & 0 \\ 0 & 0.83102 & 0 \\ 0 & 0 & 0.2618 \end{pmatrix}, \quad c = \begin{pmatrix} 1.6153 \\ 4.7871 \\ 1 \end{pmatrix} \tag{3.18}$$

Table 1
Normalized error vectors and values of α, β and γ

Method	Order p	E_{p+1}^{T}	α	β	γ
BDF_3	3	$(0, 0, \frac{1}{4})$	88.4°	1.94	0.046
(3.11)	3	(0.20, −0.017)	90°	0	0
BDF_4	4	$(0, 0, 0, \frac{1}{5})$	73.2°	4.72	0.191
(3.14)	4	(0.13, 0.27, 0.075)	90°	0	0
(3.16)	4	(3.67, 0.19, 0.064)	90°	0	0
BDF_5	5	$(0, 0, 0, 0, \frac{1}{6})$	51.8°	9.94	0.379
(3.17)	5	(0.007, 0.0038, −0.015)	> 89.9°	0.16	0.0000026
(3.18)	5	(0.004, −0.016, 0.007)	> 89.9°	0.30	0.000069

and with normalized error vector $E_6 \approx (0.004, -0.016, 0.007)^T$. At the origin the amplification factors are 0.88, 0.88 and 1, and at infinity the maximal amplification factor is ≈ 0.89.

3.3. *Survey of method characteristics*

We conclude with a survey of the parameters α, β and γ characterizing the stability regions of the block methods derived in this paper (see Definition 3.1) and compare them with those of the BDFs (details about the BDF methods can be found in [6]). In Table 1 these values are listed. In addition, we give the normalized error vectors defined in (2.5) of all methods. For a uniform presentation, we first formulated the BDFs as block methods. We recall that a k-step BDF method can be cast in the form (2.3) with block point vector $c = (2-k, \ldots, -1, 0, 1)^T$.

Finally, we remark that a k-step, kth-order BDF requires k starting values, independent of its formulation, whereas the block methods of this paper need only 2 (for $p = 3$) or 3 (for $p = 4, 5$) starting values.

4. Application to Volterra integro-differential equations

Consider the initial-value problem for VIDEs given by (1.2). The most straightforward way of solving numerically this problem replaces the integral term in (1.2) by a quadrature formula and integrates the resulting ODE by some ODE integrator. This "direct quadrature" method will be indicated by DQ method. The stability of DQ methods strongly depends on the quadrature formula used for approximating the integral term, particularly if the VIDE in (1.2) is stiff. For example, DQ methods using Gregory quadrature formulas become easily unstable (see, e.g., [1]).

A more stable approach is based on the approximation of the integral term by converting it into a differential equation and by integrating this differential equation by an ODE solver. For that purpose, we introduce the function

$$z(t, s) := \int_{t_0}^{s} k(t, x, y(x))\, dx, \tag{4.1}$$

and we write the initial-value problem (1.2) in the form

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = f(t,\, y(t),\, z(t,\, t)), \quad y(t_0) = y_0. \tag{4.2a}$$

The method now consists of the application of an ODE solver to the initial-value problem (4.2a), where the values of $z(t, t)$ needed by the ODE solver are obtained by integrating the initial-value problem

$$\frac{\partial z(t,\, s)}{\partial s} = k(t,\, s,\, y(s)), \quad z(t,\, t_0) = 0 \tag{4.2b}$$

from $s = t_0$ until $s = t$. This method still belongs to the class of DQ methods, however, it uses a special quadrature formula derived from an ODE solver. If the ODE solver is an LM method (ρ, σ), then the quadrature formula is called (ρ, σ)-*reducible* (cf. [13]). Similarly, we shall call the DQ method (ρ, σ)-*reducible* if both initial-value problems (4.2a) and (4.2b) are solved by the same LM method (ρ, σ), and (A, B, D)-*reducible* if (4.2a) and (4.2b) are solved by the same block method (2.3) generated by the matrices A, B and D.

Let us consider the stability of (A, B, D)-reducible DQ methods. Following the usual stability analysis of VIDE solvers (cf., e.g., [2,13]), we shall consider stability with respect to the basic test problem

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = \xi y(t) + \eta \int_{t_0}^{t} y(x)\, \mathrm{d}x, \quad y(t_0) = y_0. \tag{4.3}$$

Using the representation (4.2) and writing $z(t, t) = z(t)$, this problem can be represented in the form

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = \xi y(t) + \eta z(t), \quad y(t_0) = y_0, \qquad \frac{\mathrm{d}z(t)}{\mathrm{d}t} = y(t), \quad z(t_0) = 0. \tag{4.4}$$

Application of the block method (2.3) to each of these equations yields the recursions

$$\begin{aligned} \boldsymbol{Y}_{n+1} &= A\boldsymbol{Y}_n + hB[\xi \boldsymbol{Y}_n + \eta \boldsymbol{Z}_n] + hD[\xi \boldsymbol{Y}_{n+1} + \eta \boldsymbol{Z}_{n+1}], \\ \boldsymbol{Z}_{n+1} &= A\boldsymbol{Z}_n + hB\boldsymbol{Y}_n + hD\boldsymbol{Y}_{n+1}. \end{aligned} \tag{4.5}$$

We shall show that (4.5) is algebraically equivalent with the recursion obtained by applying (2.3) to the system (4.4). Writing (4.4) in the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t) = \begin{pmatrix} \xi & \eta \\ 1 & 0 \end{pmatrix} \boldsymbol{u}(t), \qquad \boldsymbol{u}(t) := \begin{pmatrix} y(t) \\ z(t) \end{pmatrix}, \tag{4.4'}$$

the block method (2.3) takes the form

$$\begin{aligned} &\boldsymbol{U}_{n+1} = A \circ \boldsymbol{U}_n + hB \circ f(\boldsymbol{U}_n) + hD \circ f(\boldsymbol{U}_{n+1}), \\ &\boldsymbol{U}_{n+1} := (y_{n,1},\, z_{n,1};\, \ldots;\, y_{n,k},\, z_{n,k})^{\mathrm{T}}, \\ &f(\boldsymbol{U}_{n+1}) := (\xi y_{n,1} + \eta z_{n,1},\, y_{n,1};\, \ldots;\, \xi y_{n,k} + \eta z_{n,k},\, y_{n,k})^{\mathrm{T}}, \end{aligned} \tag{4.5'}$$

with $y_{n,j}$ and $z_{n,j}$ denoting the components of the (column) vectors $\boldsymbol{Y}_{n+1}$ and $\boldsymbol{Z}_{n+1}$ used in (4.5), and where the tensor products $A \circ \boldsymbol{U}_n$ and $B \circ f(\boldsymbol{U}_n)$ are defined according to

$$A \circ \boldsymbol{U}_n := (\boldsymbol{a}_1 \boldsymbol{Y}_n, \boldsymbol{a}_1 \boldsymbol{Z}_n; \ldots; \boldsymbol{a}_k \boldsymbol{Y}_n, \boldsymbol{a}_k \boldsymbol{Z}_n)^{\mathrm{T}},$$

$$B \circ f(\boldsymbol{U}_n) := (\boldsymbol{b}_1(\xi \boldsymbol{Y}_n + \eta \boldsymbol{Z}_n), \boldsymbol{b}_1 \boldsymbol{Y}_n; \ldots; \boldsymbol{b}_k(\xi \boldsymbol{Y}_n + \eta \boldsymbol{Z}_n), \boldsymbol{b}_k \boldsymbol{Y}_n)^{\mathrm{T}},$$

with $\boldsymbol{a}_j$ and $\boldsymbol{b}_j$ denoting the jth row vectors of the matrices A and B, respectively. It is now readily verified that by reordering the equations occurring in (4.5′) such that the first, third, fifth,... equations come first and the second, fourth, sixth,... equations come next, we obtain the recursions (4.5).

Hence, if λ and μ denote the eigenvalues of the Jacobian matrix associated with (4.4′), then the recursion (4.5) is stable if both $h\lambda$ and $h\mu$ are in the stability region of the block method (2.3). The corresponding region of $(h\xi, h^2\eta) = (h\lambda + h\mu, -h^2\lambda\mu)$-values will be called the *stability region of the (A, B, D)-reducible DQ method*. Furthermore, if this stability region contains the set $\{(h\xi, h^2\eta)\colon \xi < 0, \eta < 0\}$, then the DQ method is called *A_0-stable*. The preceding considerations can be summarized in the following theorem which generalizes a result for LM methods originally given by Brunner and Lambert [2].

Theorem 4.1. *Let S be the stability region of the block method* (2.3) *generated by the matrices A, B and D, and let γ and μ be defined by $\lambda + \mu = \xi$, $\lambda\mu = -\eta$. Then the set $\{(h\xi, h^2\eta)\colon h\lambda \in S, h\mu \in S\}$ defines the region of stability of the (A, B, D)-reducible* DQ *method.*

From this theorem it follows that the (A, B, D)-reducible DQ method is A_0-stable if, and only if, the generating block method (A, B, D) is A-stable. Thus, the use of the block methods constructed in this paper avoids the so-called "second Dahlquist barrier" which applies to A_0-stable (ρ, σ)-reducible DQ methods for VIDEs (cf. [13, Theorem 5]).

5. Numerical experiments

5.1. Accuracy test

To verify the order of the various methods we integrated the test problem proposed by Kaps [11]:

$$\frac{dy_1}{dt} = -(2+\varepsilon^{-1})y_1 + \varepsilon^{-1}(y_2)^2, \qquad \frac{dy_2}{dt} = y_1 - y_2(1+y_2),$$

$$y_1(0) = y_2(0) = 1, \qquad 0 \leqslant t \leqslant T, \tag{5.1}$$

with exact solution $y_1 = \exp(-2t)$ and $y_2 = \exp(-t)$ for all values of the parameter ε. In Table 2, we have listed the values Δ, where Δ denotes the number of correct decimal digits at the endpoint (i.e., we write the maximum norm of the error at $t = T$ in the form $10^{-\Delta}$). In all experiments the theoretical order of the method is shown for sufficiently small values of h (if p is the order of the method, then, on halving the step size, the value of Δ should increase by $\approx 0.3p$).

Table 2
Values of Δ for (5.1) with $T = 1$, $\varepsilon = 10^{-8}$

Method	p	h					
		$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{1}{128}$
BDF_3	3	2.8	3.7	4.6	5.5	6.5	7.4
(3.11)	3	2.8	3.6	4.4	5.2	6.1	7.0
BDF_4	4	3.4	4.7	5.9	7.1	8.4	9.6
(3.14)	4	3.8	5.2	9.5	7.9	8.9	10.0
(3.16)	4	3.1	3.9	4.8	5.9	7.1	8.2
BDF_5	5	4.0	5.6	7.2	8.7	10.2	12.0
(3.17)	5	2.6	4.0	5.5	7.3	9.2	10.3
(3.18)	5	4.7	5.4	6.4	7.7	9.2	10.1

Table 3
Values of Δ for (5.2) with $T = 100$, $\alpha = 10$

Method	p	h					
		$\frac{4}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$\frac{1}{10}$	$\frac{1}{20}$	$\frac{1}{40}$
BDF_3	3	2.0	2.9	3.9	*	*	4.9
(3.11)	3	2.1	2.8	3.4	4.0	4.6	5.3
BDF_4	4	2.2	*	*	*	2.9	8.2
(3.14)	4	2.8	4.0	4.9	5.8	6.8	8.0
(3.16)	4	1.6	2.7	3.8	4.9	5.8	6.8
BDF_5	5	−0.1	*	*	*	8.5	10.3
(3.17)	5	1.2	2.0	3.4	4.7	6.2	7.6
(3.18)	5	2.9	3.9	5.1	6.4	7.6	8.6

5.2. *Stability test*

We tested the stability of the methods by integrating a problem in which the Jacobian matrix has purely imaginary eigenvalues:

$$\frac{dy_1}{dt} = -\alpha y_2 + (1+\alpha)\cos(t), \qquad \frac{dy_2}{dt} = \alpha y_1 - (1+\alpha)\sin(t),$$

$$y_1(0) = 0, \qquad y_2(0) = 1, \qquad 0 \leqslant t \leqslant T, \tag{5.2}$$

with exact solution $y_1 = \sin(t)$ and $y_2 = \cos(t)$ for all values of the parameter α.

In Table 3 the results are listed for $T = 100$. Values of Δ corresponding to stepsizes that are theoretically unstable are underlined and overflow is indicated by $*$. The unstable results of the BDFs is in agreement with their regions of instability indicated in Table 1 (the phenomenon that BDF_5 becomes stable again for sufficiently small h is due to the fact that its imaginary interval of instability is given by i[0.71, 9.94]).

Next, we show that the "almost" A-stable fifth-order methods (3.17) and (3.18) behave as A-stable methods in practice. We performed experiments for $\alpha = 1$ and $\alpha = 4$ with $h = \frac{1}{8}$: for $\alpha = 1$ both integration processes are theoretically unstable, and for $\alpha = 4$ the processes are stable. In Table 4 the results are listed for increasing length of the integration interval: these results clearly show that both methods perform perfectly stable for $\alpha = 1$.

Table 4
Values of Δ for problem (5.2) for $h = \frac{1}{8}$

Method	$\alpha = 1$: theoretically unstable			$\alpha = 4$: theoretically stable		
	$T = 10$	$T = 100$	$T = 1000$	$T = 10$	$T = 100$	$T = 1000$
(3.17)	3.6	3.8	3.6	4.0	3.9	3.9
(3.18)	4.5	4.3	4.8	5.4	5.4	5.4

Table 5
Values of Δ for problem (5.3) at $T = 10$

Method	Order p	$\alpha = 1$			$\alpha = 10$		
		$h = \frac{1}{2}$	$h = \frac{1}{4}$	$h = \frac{1}{8}$	$h = \frac{1}{2}$	$h = \frac{1}{4}$	$h = \frac{1}{8}$
BDF_3	3	5.7	6.8	7.9	6.0	6.9	7.8
(3.11)	3	5.5	6.5	7.3	5.4	6.5	7.3
BDF_4	4	5.4	7.0	8.3	6.5	8.1	9.4
(3.14)	4	6.0	8.3	9.1	6.4	8.6	10.9
(3.16)	4	5.2	6.2	7.2	6.7	7.9	8.5
BDF_5	5	5.1	7.2	8.9	6.1	8.2	9.9
(3.17)	5	2.5	5.2	7.2	2.9	5.3	7.5
(3.18)	5	6.0	6.9	8.2	6.8	8.5	9.3

5.3. *Volterra integro-differential equation*

Consider the initial-value problem

$$\frac{dy(t)}{dt} = -\frac{1+\alpha t(1+t)^2}{(1+t)^2} + \frac{\alpha}{y(t)}\ln\left(\frac{2+2t}{2+t}\right) + \alpha\int_0^t \frac{dx}{1+(1+t)y(x)},$$
$$y(2) = \tfrac{1}{3}, \qquad 2 \leqslant t \leqslant T, \qquad \alpha > 0 \tag{5.3}$$

with exact solution $y(t) = 1/(1+t)$. For $\alpha = 1$, this problem has been discussed in [2]. From the expressions

$$\xi := \frac{\partial f}{\partial y} = -\frac{\alpha}{y^2(t)}\ln\left(\frac{2+2t}{2+t}\right), \qquad \eta := \frac{\partial f}{\partial z}\frac{\partial k}{\partial y} = -\alpha\frac{1+t}{(1+(1+t)y)^2}$$

it follows that (5.3) is stable if $t > 0$ and $y \geqslant 0$. Furthermore, we see that in the vicinity of the exact solution we have $\xi \approx -\alpha(1+t)^2$ and $\eta \approx -\alpha(1+t)$, so that the stiffness of this problem increases with α and t. For example, if $\alpha = T = 10$, then an A_0-stable method is highly desirable.

Table 5 lists results for various methods and values of the stepsize h. Notice that the results for the stiff problem ($\alpha = 10$) are not less accurate (even more accurate) than the results for the nonstiff problem ($\alpha = 1$), showing that stiffness does not cause any problem. Similar to the ODE case (cf. Table 2), the method (3.14) performs very accurately, whereas (3.17) is significantly less accurate.

5.4. *Performance test on the ALLIANT FX/4*

Finally, we tested the methods (3.11) and (3.18) on the ALLIANT FX/4 by integrating the problem (5.1) of Kaps. In Table 6, we have listed timings on P processors and the rate of

Table 6
Timings (in seconds) for problem (5.1) at $T = 1$ with $\varepsilon = 10^{-8}$ and $h = \frac{1}{256}$

Method	k	$P = 1$	$P = 2$	$P = 3$	$P = 4$	Efficiency rate
(3.11)	2	0.43	0.23	0.23		0.93
(3.18)	3	0.66	0.45	0.25	0.25	0.88

efficiency of a k-processor method, i.e., the execution time on one processor divided by k times the execution time on k processors. These results show that the gain factor is close to its optimal value.

From Table 6 we conclude that the performance is close to its optimum, that is, the gain factor obtained for a k-processor method is almost equal to k. Table 6 also lists timings in cases where methods have the disposal of one more processor (i.e., $k + 1$) than the number (i.e., k) they are designed for. We see that this additional processor is not utilized, since the k processors (concurrently) solve the k implicit relations and the extra processor is idle. As mentioned before, it could have been exploited for updating the Jacobian matrix, but in this test we did not include such a technique.

It should be noted that the efficiency rate is slightly dependent on implementation strategies, such as how accurately the nonlinear systems are solved. For example, it may happen that the first (or any other) implicit relation requires less Newton iterations than the other implicit relations (e.g., because of a more accurate initial approximation); in such cases this first processor will be idle for some time, which of course, has a bad influence on the efficiency rate.

References

[1] H. Brunner and P.J. van der Houwen, *The Numerical Solution of Volterra Equations*, CWI Monograph 3 (North-Holland, Amsterdam, 1986).

[2] H. Brunner and J.D. Lambert, Stability of numerical methods for Volterra integro-differential equations, *Computing* 12 (1974) 75–89.

[3] M.T. Chu and H. Hamilton, Parallel solution of ODE's by multi-block methods, *SIAM J. Sci. Statist. Comput.* 8 (1987) 342-353.

[4] G.J. Cooper, The order of convergence of general linear methods for ordinary differential equations, *SIAM J. Numer. Anal.* 15 (1978) 643-661.

[5] A. Feldstein, Oral communication at the International Conference on the Numerical Solution of Volterra and Delay Equations, Arizona State University, Tempe, AZ (1990).

[6] C.W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1971).

[7] E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations I. Nonstiff Problems* (Springer, Berlin, 1987).

[8] P. Henrici, *Discrete Variable Methods in Ordinary Differential Equations* (Wiley, New York, 1962).

[9] P.J. van der Houwen and B.P. Sommeijer, Block Runge–Kutta methods on parallel computers, Report NM-R8906, Centre for Mathematics and Computer Science, Amsterdam (1989); also: *Z. Angew. Math. Mech.* (to appear).

[10] P.J. van der Houwen and B.P. Sommeijer, Parallel ODE solvers, in: *Proceedings International Conference on Supercomputing*, Amsterdam (ACM Press, New York, 1990) 71-81.

[11] P. Kaps, Rosenbrock-type methods, Bericht Nr. 9, Inst. für Geometrie und Praktische Mathematik der RWTH Aachen (1981).

[12] C. Lubich, Oral communication at the International Conference on the Numerical Solution of Volterra and Delay Equations, Arizona State University, Tempe, AZ (1990).

[13] J. Matthys, A-stable linear multistep methods for Volterra integro-differential equations, *Numer. Math.* 27 (1976) 85-94.

[14] B.P. Sommeijer, W. Couzy and P.J. van der Houwen, A-stable parallel block methods, Report NM-R8918, Centre for Mathematics and Computer Science, Amsterdam (1989).

[15] R.S. Varga, *Matrix Iterative Analysis* (Prentice-Hall, Englewood Cliffs, NJ, 1962).

[16] H.A. Watts and L.F. Shampine, A-stable block implicit one-step methods, *BIT* 12 (1972) 252-266.