ELSEVIER

Applied Numerical Mathematics 42 (2002) 29–45

APPLIED NUMERICAL MATHEMATICS

www.elsevier.com/locate/apnum

Blended implementation of block implicit methods for ODEs ☆

Luigi Brugnano *, Cecilia Magherini

Dipartimento di Matematica "U. Dini", Viale Morgagni 67/A, 50134 Firenze, Italy

Abstract

In this paper we further develop a new approach for naturally defining the nonlinear splittings needed for the implementation of block implicit methods for ODEs, which has been considered by Brugnano [J. Comput. Appl. Math. 116 (2000) 41] and by Brugnano and Trigiante [in: Recent Trends in Numerical Analysis, Nova Science, New York, 2000, pp. 81–105]. The basic idea is that of defining the numerical method as the combination (*blending*) of two suitable component methods. By carefully choosing such methods, it is shown that very efficient implementations can be obtained. Moreover, some of them, characterized by a diagonal splitting, are well suited for parallel computers. Some numerical tests comparing the performances of the proposed implementation with other existing ones are also presented, in order to make evident the potential of the approach. © 2001 IMACS. Published by Elsevier Science B.V. All rights reserved.

Keywords: Numerical methods for ODEs; Iterative solution of linear systems; Nonlinear splittings

1. Introduction

The numerical solution of the ODE problem

$$y' = f(t, y), \quad t \in [t_0, T], \qquad y(t_0) = y_0 \in \mathbb{R}^m, \tag{1}$$

is usually carried out by formally executing the following three steps:

(1) the definition of a suitable partition of the integration interval $[t_0, T]$,
(2) the construction of a discrete problem defined on such a discrete set,
(3) the solution of the discrete problem.

☆ Work supported by Italian M.U.R.S.T. and C.N.R.

* Corresponding author.
E-mail addresses: brugnano@math.unifi.it (L. Brugnano), magherini@math.unifi.it (C. Magherini).

PII: S0168-9274(01)00140-4

Our aim now is to devise an efficient procedure for solving the discrete problems, so that we shall hereafter confine ourselves to the uniform partition with stepsize h:

$$t_n = t_0 + nh, \quad n = 0, \ldots, N, \qquad Nh = T - t_0,$$

where N is a suitable multiple of an integer r to be defined in a while. Concerning the discrete problem, we shall refer to that generated by a *block implicit method*, namely a method generating a discrete problem in the form

$$F(\mathbf{y}_n) \equiv A \otimes I_m \mathbf{y}_n - hB \otimes I_m \mathbf{f}_n - \boldsymbol{\eta}_n = \mathbf{0}, \tag{2}$$

where the matrices $A, B \in \mathbb{R}^{r\times r}$ define the method, the block vectors

$$\mathbf{y}_n = (y_{n+1}, \ldots, y_{n+r})^{\mathrm{T}}, \qquad \mathbf{f}_n = (f_{n+1}, \ldots, f_{n+r})^{\mathrm{T}}, \quad f_j = f(t_j, y_j),$$

contain the discrete solution, and the vector $\boldsymbol{\eta}_n$ only depends on already known quantities. Instances of methods falling in this class are Runge–Kutta methods, a number of General Linear methods [8–10] and, more recently, block BVMs [3].

In the following we shall always assume the two matrices A and B to be nonsingular, so that the underlying method is an implicit one. Consequently, an iterative procedure is usually carried out in order to solve Eq. (2). The most straightforward one is the simplified Newton method which, however, requires the factorization of the $rm \times rm$ Jacobian matrix of F.

A useful simplification, from the computational point of view, can be obtained when the two matrices A and B are diagonalized by the same transformation matrix [7]: in such a case, in fact, only the solution of (eventually complex) $m \times m$ linear systems is required. This kind of implementation is very popular for Runge–Kutta methods and is, indeed, used in the code RADAU5 [10].

An alternative approach is obtained by defining a suitable splitting for Eq. (2). Roughly speaking, instead of solving (2), one solves an *inner-outer* iteration in the form

$$\begin{aligned} &A^* \otimes I_m \mathbf{y}_n^{(i+1)} - hB^* \otimes I_m \mathbf{f}_n^{(i+1)} \\ &\quad = \big(A^* - A\big) \otimes I_m \mathbf{y}_n^{(i)} - h\big(B^* - B\big) \otimes I_m \mathbf{f}_n^{(i)} + \boldsymbol{\eta}_n, \quad i = 0, 1, \ldots, \end{aligned} \tag{3}$$

where the two matrices A^* and B^* have a much simpler structure than A and B, respectively. This implies that the nonlinear system to be solved at each iteration in (3) is in general much simpler than solving the original problem (2). As an example, the matrices A^* and B^* can be chosen lower triangular with constant entries on the main diagonal. In such a case, the simplified Newton iteration for solving (3) only requires us to factor one $m \times m$ matrix. By definition, the procedure is convergent if $\mathbf{y}_n^{(i)} \to \mathbf{y}_n$, as $i \to \infty$. Such an approach, used for example in the code GAM [13], may be very competitive, provided that suitable matrices A^* and B^* for defining the splitting can be obtained (see also [1,11,12]). Nevertheless, their derivation may be, in general, very difficult, when satisfactory convergence properties are required. We shall see that this problem can be much more easily handled via the *blending* of methods, namely by defining a numerical method as the combination of two methods. Concerning this point, we mention that in the past years many attempts have been made to derive numerical methods for ODEs as the combination of two methods. A well-known example is the popular θ-method. Additional examples are provided by the *blended linear multistep formulas* of Skeel and Kong [17] and by the *blended block BVMs* [2]. However, slightly different aims were pursued in doing this:

- in the case of the θ-method and of the blended linear multistep formulas, the only aim was that of getting a method with better stability properties than the two component ones;

- in the case of blended block BVMs, the above aim was coupled with that of getting an efficient implementation of the resulting method.

Moreover, we want also to stress that the implementation issue has become focal for numerical methods for ODEs: indeed, since a number of stable, high order methods are currently available, one of the main reasons to use a method in place of another one is given by its computational cost. As matter of fact, many methods in the class of both Runge–Kutta and General Linear methods have been defined for reducing such cost (see, e.g., [5,15], for Runge–Kutta methods). For this reason, the implementation issue has become paramount in [4], where the main idea has been that of blending different discrete problems derived from the same method, rather than blending different methods. In such a case, we shall speak about a *blended implementation* of the basic method. We now shall further investigate this approach, by making evident its properties and potentialities.

The paper is organized as follows: in Section 2 we give a detailed presentation of the basic idea of the proposed approach, together with its main features; then, in Section 3, we recall the main facts about the methods to be combined. In Section 4 we examine some relevant properties needed for the actual implementation of the methods and, finally, in Section 5 some numerical tests are reported along with some concluding remarks.

2. Blending of block methods

In this paper we shall be concerned with the definition of numerical methods for which a suitably splitting (3) is naturally defined. In order to present the methods, and to carry out the linear analysis of convergence, we shall consider the application of the methods to the classical test equation

$$y' = \mu y, \qquad y(t_0) = y_0, \qquad \mathrm{Re}(\mu) < 0, \tag{4}$$

for which, by setting as usual $q = h\mu$, the discrete problem (2) assumes the form (let us discard, for sake of brevity, the index n for the block vectors):

$$(A - qB)\mathbf{y} = \boldsymbol{\eta}. \tag{5}$$

We observe that the solution of the previous equation is not affected by left-multiplication by A^{-1} or B^{-1} of both sides of the equation,

$$\big(I - qA^{-1}B\big)\mathbf{y} = A^{-1}\boldsymbol{\eta}, \qquad \big(B^{-1}A - qI\big)\mathbf{y} = B^{-1}\boldsymbol{\eta}. \tag{6}$$

The basic idea for the blended implementation of the method (5) relies on the fact that, by combining equations in the form (6), the discrete solution does not change. In more detail, let A_1 be a nonsingular matrix with a "simple" structure. By multiplying on the left both sides of the first equation in (6), we then obtain

$$(A_1 - qB_1)\mathbf{y} = \boldsymbol{\eta}_1, \tag{7}$$

where

$$B_1 = A_1A^{-1}B, \qquad \boldsymbol{\eta}_1 = A_1A^{-1}\boldsymbol{\eta}. \tag{8}$$

Similarly, by considering another nonsingular and "simple structured" matrix B_2, by multiplying on the left the second equation in (6) we obtain

$$(A_2 - qB_2)\mathbf{y} = \boldsymbol{\eta}_2, \tag{9}$$

where

$$A_2 = B_2 B^{-1} A, \qquad \boldsymbol{\eta}_2 = B_2 B^{-1} \boldsymbol{\eta}. \tag{10}$$

Obviously, both Eqs. (7) and (9) do have the same solution as Eq. (5), since they are derived from the same method.

In addition to this, let us define a suitable *weighting function* $\theta(q)$ such that

$$\theta(0) = I, \qquad \theta(q) \to O, \quad \text{as } q \to \infty, \tag{11}$$

where I and O are, respectively, the $r \times r$ identity and the zero matrix. Then, also the following equation,

$$\begin{aligned} M(q)\mathbf{y} - \boldsymbol{\eta}(q) &\equiv \big(A(q) - qB(q)\big)\mathbf{y} - \boldsymbol{\eta}(q) \\ &\equiv \big(\big(\theta(q)A_1 + \big(I - \theta(q)\big)A_2\big) - q\big(\theta(q)B_1 + \big(I - \theta(q)\big)B_2\big)\big)\mathbf{y} \\ &\quad - \big(\theta(q)\boldsymbol{\eta}_1 + \big(I - \theta(q)\big)\boldsymbol{\eta}_2\big) \\ &= \mathbf{0}, \end{aligned} \tag{12}$$

does have the same solution as (5): as matter of fact, the latter is obtained by "blending" the same Eq. (5) written in the two different, though equivalent, forms (7) and (9).

The advantage of using such an approach consists in the fact that, from (11), one obtains that

- for $q \approx 0$: $M(q) \approx A_1 - qB_1 \approx A_1$;
- for $q \to \infty$: $M(q) \approx A_2 - qB_2 \approx -qB_2$.

Consequently, instead of solving (12), one may think to solve iteratively

$$N(q)\mathbf{y}^{(i+1)} = \big(N(q) - M(q)\big)\mathbf{y}^{(i)} + \boldsymbol{\eta}(q), \quad i = 0, 1, \ldots, \tag{13}$$

where

$$N(q) = A_1 - qB_2. \tag{14}$$

Obviously, the iteration (13) is a convergent one iff the spectral radius of the iteration matrix $I - N(q)^{-1}M(q)$, say $\rho(q)$, is smaller than 1. Following [11,12], the iteration is said to be A-convergent if $\rho(q) < 1$ for all $q \in \mathbb{C}^-$. If the matrix pencil (14) has no eigenvalues having negative real part, and the function $\theta(q)$ is analytic in $\mathbb{C}^-$, A-convergence is equivalent to requiring that the maximum amplification factor,

$$\rho^* = \max_{x \geqslant 0} \rho(\mathrm{i}x), \tag{15}$$

with i denoting the imaginary unit, is smaller than 1. We observe that, from (14), one obtains that

- $\rho(0) = 0$,
- $\rho^{(\infty)} \equiv \lim_{q\to\infty} \rho(q) = 0$,

since, in both cases the iteration matrix is the zero matrix. Consequently, one has that, because of the second property, the iteration (13) is well-suited for stiff problems, since the *stiff amplification factor* $\rho^{(\infty)}$ [11,12] is 0. Moreover, if the iteration matrix is well-defined in a neighborhood of $q = 0$, the first property implies that

$$\rho(q) \approx q\tilde{\rho}, \quad \text{for } q \approx 0, \tag{16}$$

where $\tilde{\rho}$ is the *nonstiff amplification factor*.

Concerning the choices of the two "simple structured" matrices A_1 and B_2, we shall here consider the following choice, though different ones are possible,

$$A_1 = I + L_A, \qquad B_2 = D + L_B, \tag{17}$$

where L_A and L_B are strictly lower triangular matrices, and D is a diagonal matrix with positive entries. With such assumptions, we have that the linear systems required by the iteration (13) are lower triangular (block lower triangular when the method is applied to systems of equations). Moreover, in the case of systems, one only needs to factorize matrices having the same size of the continuous problem, and the number of matrices to be actually factored equals the number of distinct diagonal entries of the matrix D.

Finally, in order to keep low the computational cost, the weight function $\theta(q)$ is defined as

$$\theta(q) = (I - qD)^{-1}, \tag{18}$$

so that the properties (11) are satisfied, the iteration (13) is well-defined for all $q \in \mathbb{C}^-$, and, in the case of systems, no additional factorizations are required, besides those needed for $N(q)$.

With such assumptions, the only key-points which we need to clarify are the following ones:

(1) the choice of appropriate methods (5),
(2) the choice of the corresponding "simple structured" matrices A_1 and B_2 in (17) (the remaining matrices B_1 and A_2 being defined by (8) and (10), respectively).

The first point will be discussed in the next section, whereas the second one will be addressed in Section 4.

3. Choice of the component methods

Let now introduce the methods that we shall implement in blended form, according to what has been said in the previous section. Even though different choices can be made, we shall here consider methods which have been already introduced in the past years by Watts and Shampine [19]. Such methods are block methods characterized by the fact that each one of the r equations which define the method itself corresponds to a linear multistep formula. Even though the methods could be also derived in the framework of Runge–Kutta methods (by means of the "V-transform" [5,6,10]) we prefer to use the same framework originally used in [19] (see also [4]).

In more detail, let define the following $r \times (r+1)$ matrices,

$$\hat{A} = [\mathbf{a} \mid A] \equiv \left(\begin{array}{c|ccc} \alpha_0^{(1)} & \alpha_1^{(1)} & \dots & \alpha_r^{(1)} \\ \vdots & \vdots & & \vdots \\ \alpha_0^{(r)} & \alpha_1^{(r)} & \dots & \alpha_r^{(r)} \end{array}\right), \qquad \widehat{B} = [\mathbf{b} \mid B] \equiv \left(\begin{array}{c|ccc} \beta_0^{(1)} & \beta_1^{(1)} & \dots & \beta_r^{(1)} \\ \vdots & \vdots & & \vdots \\ \beta_0^{(r)} & \beta_1^{(r)} & \dots & \beta_r^{(r)} \end{array}\right), \tag{19}$$

where the coefficients on the ith row of the two matrices define a suitable r-step LMF. Assuming for simplicity that the first r points are to be approximated, the further relation with (5) is that

$$\boldsymbol{\eta} = -(\mathbf{a} - q\mathbf{b})\, y_0.$$

It is not difficult to prove the following result.

Theorem 1. *Let all LMFs defining* (19) *have an* $\mathrm{O}(h^{p+1})$ *truncation error. Then*

$$A\mathbf{q}_i = i\,B\mathbf{q}_{i-1}, \quad i = 2, \ldots, p,$$

where $\mathbf{q}_i = (1^i, \ldots, r^i)^{\mathrm{T}}$, *and, moreover*

$$\mathbf{a} = -A\mathbf{q}_0, \qquad \mathbf{b} = A\mathbf{q}_1 - B\mathbf{q}_0.$$

Then the previous result tells us that, provided all LMFs in (19) are consistent, we can concentrate our attention on the matrices A and B alone. Moreover, since we assume both of them to be nonsingular, one obtains that the method is uniquely defined by the matrix $C = A^{-1}B$. It can be shown that we can uniquely define such a matrix by imposing $p = r$, and by fixing its characteristic polynomial,

$$d(z) = \sum_{i=0}^{r} d_i z^i, \quad d_r = 1. \tag{20}$$

As matter of fact, one obtains that (see [4])

$$C = QG^{-1}FGQ^{-1},$$

where $Q = (\mathbf{q}_1, \ldots, \mathbf{q}_r)$, $G = \mathrm{diag}(1!, \ldots, r!)$, and

$$F = \begin{pmatrix} & & -d_0 \\ 1 & & -d_1 \\ & \ddots & \vdots \\ & 1 & -d_{r-1} \end{pmatrix}.$$

Concerning the choice of the characteristic polynomial (20), a necessary requirement for having a stable method, is to have its roots with positive real part. In order to meet this requirement, and to ensure good stability properties for the corresponding method, we choose the polynomial (20) as the reciprocal, and scaled, polynomial at the denominator of the (ν, r) Padé approximation to the exponential,

$$z^r d\left(z^{-1}\right) = \sum_{i=0}^{r} \frac{(\nu + r - i)!r!}{(\nu + r)!i!(r - i)!}(-rz)^i.$$

In such a case, in fact, one obtains A-stable methods for all r and $\nu = r - 2, r - 1, r$, which are also L-stable for $\nu < r$ [19] (see also [18]). Moreover, for all $r \geqslant 3$, it can be shown that the local error of the methods has the first $r - 1$ entries which are $\mathrm{O}(h^{r+1})$, and the rth one which is (for general nonlinear problems)

- $\mathrm{O}(h^{r+3})$, when r is even,
- $\mathrm{O}(h^{r+2})$, when r is odd,

in which case, the global order of convergence of the corresponding method is, respectively, $r + 2$ or $r + 1$ [4].

In the present case, we look for L-stable methods and, consequently, we need to choose appropriate values for the couples (ν, r), $\nu \in \{r-2, r-1\}$. In order to make the proper choice, we observe that the last entry of the vector $\mathbf{y}$ in (5) is essentially given by the (ν, r) Padé approximation to the exponential. We know that such an approximation is exact at $q = 0$ and as $q \to \infty$ (due to the L-stability of the methods). In addition to this, we also require that, for $\mu < 0$ (see (4)), the discrete solution has the same sign as the continuous one (which is the sign of y_0), whatever the stepsize h used. By considering that the Padé approximation, for $\nu = r-2,\ r-1$, is analytic in $\mathbb{C}^-$, with no real and negative zeros when

- r is even and $\nu = r-2$,
- r is odd and $\nu = r-1$,

we shall hereafter consider the methods obtained in correspondence of the $(2,3)$, $(2,4)$, $(4,6)$, $(6,8)$, $(8,10)$, and $(10,12)$ Padé approximations to the exponential (having orders 4, 6, 8, 10, 12 and 14, respectively). This in view of a variable order, variable stepsize implementation of the methods themselves.

4. Properties and implementation details

In this section we shall study in more detail particular choices of appropriate matrices A_1 and B_2, as defined in (17). As we have said, this uniquely defines the whole blended implementation of the considered method, since the weight function θ and the remaining matrices B_1 and A_2 are defined according to (18), (8) and (10), respectively. We start considering the simpler case where

$$L_A = L_B = O, \qquad D = \gamma I, \quad \gamma > 0, \tag{21}$$

since in such a case a complete spectral analysis can be carried out. In fact, one has that

$$A_1 = I, \qquad A_2 = \gamma B_1^{-1}, \qquad B_2 = \gamma I, \qquad \theta(q) = (1-q\gamma)^{-1} I, \tag{22}$$

which allow us to easily derive the following result.

Theorem 2. *Assume that for the blended method* (12) *the previous equalities* (21) *hold true. Then, the eigenvalues of the iteration matrix corresponding to* (13) *and* (14) *are given by*

$$\frac{q(\lambda-\gamma)^2}{\lambda(1-q\gamma)^2}, \quad \lambda \in \sigma(B_1). \tag{23}$$

Proof. Since (21) are satisfied, then also (22) do. Consequently, by taking into account (13) and (14), one obtains that the iteration matrix is given by

$$\begin{aligned} I - N(q)^{-1}M(q) &= I - (1-q\gamma)^{-2}\big(I - qB_1 - q\gamma^2\big(B_1^{-1} - qI\big)\big) \\ &= (1-q\gamma)^{-2}\big((1-q\gamma)^2 I - I + qB_1 + q\gamma^2\big(B_1^{-1} - qI\big)\big) \\ &= (1-q\gamma)^{-2}B_1^{-1}\big(\big(q^2\gamma^2 - 2q\gamma\big)B_1 + qB_1^2 + q\gamma^2(I - qB_1)\big) \\ &= q(1-q\gamma)^{-2}B_1^{-1}\big(B_1^2 - 2\gamma B_1 + \gamma^2 I\big) = q(1-q\gamma)^{-2}B_1^{-1}(B_1 - \gamma I)^2, \end{aligned}$$

from which the thesis follows. □

The above result allows us to easily characterize the value of the two parameters ρ^* and $\tilde{\rho}$ as defined in (15) and (16), respectively. In fact, by expanding (23) at $q = 0$, one readily obtains that

$$\tilde{\rho} = \max_{\lambda\in\sigma(B_1)} \frac{|\lambda-\gamma|^2}{|\lambda|}. \tag{24}$$

Similarly, for $q = \mathrm{i}x$, one has that the modulus of (23) is given by

$$\frac{x|\lambda-\gamma|^2}{|\lambda|(1+x^2\gamma^2)}, \quad x \geqslant 0,$$

which is strictly monotone increasing in $[0, \gamma^{-1})$, and decreasing in (γ^{-1}, ∞). As a consequence, one readily obtains that, at $x = \gamma^{-1}$,

$$\rho^* = \max_{\lambda\in\sigma(B_1)} \frac{|\lambda-\gamma|^2}{2\gamma|\lambda|}. \tag{25}$$

The above relations allow the derivation of simple criteria for choosing the parameter γ: indeed one may think to choose it in order to minimize either (24), or (25), or a combination of the two. Concerning the minimization of (24) and (25), the following result may be used. Before stating it, let us order the eigenvalues of the matrix B_1, so that

$$\frac{\pi}{2} > \arg(\lambda_1) \geqslant \arg(\lambda_2) \geqslant \cdots \geqslant \arg(\lambda_r) > -\frac{\pi}{2}.$$

Since the matrix is real, this means that we can only consider the first $\ell = \lceil r/2 \rceil$ eigenvalues, in the sequel. Let now assume that the moduli of the eigenvalues are strictly decreasing, that is,

$$|\lambda_1| < |\lambda_2| < \cdots < |\lambda_\ell|.$$

By setting $\lambda_j = \varphi_j \mathrm{e}^{\mathrm{i}\zeta_j}$, $j = 1, \ldots, \ell$, then the previous two equations can be written as

$$0 < \varphi_1 < \cdots < \varphi_\ell, \qquad \frac{\pi}{2} > \zeta_1 \geqslant \cdots \geqslant \zeta_\ell \geqslant 0. \tag{26}$$

We can now state the following preliminary result.

Lemma 3. *Assume that* (21) *holds true and the eigenvalues of the matrix* B_1 *satisfy* (26)*. Then, for all values of* γ *greater than or equal to*

$$\hat{\gamma} \equiv \max_{j\in\{1,\ldots,\ell\}} \Psi_j + \sqrt{\Psi_j^2 + \varphi_1\varphi_j}, \qquad \Psi_j = \frac{\varphi_1\varphi_j(\cos\zeta_1 - \cos\zeta_j)}{\varphi_j - \varphi_1}, \tag{27}$$

one has that

$$\frac{|\lambda_1-\gamma|^2}{|\lambda_1|} = \max_{j\in\{1,\ldots,\ell\}} \frac{|\lambda_j-\gamma|^2}{|\lambda_j|}. \tag{28}$$

Proof. Indeed, in order for (28) to be satisfied, for all $j > 1$ one must have

$$\frac{|\lambda_j-\gamma|^2}{|\lambda_j|} \leqslant \frac{|\lambda_1-\gamma|^2}{|\lambda_1|}.$$

By multiplying both sides by $|\lambda_1\lambda_j|$, and taking into account (26), one then obtains the following second order inequality,

$$\gamma^2(\varphi_j - \varphi_1) - 2\gamma\Psi_j + \varphi_1\varphi_j(\varphi_1 - \varphi_j) \geqslant 0,$$

which, considering that $\Psi_j \leqslant 0$ and the discriminant of the equation is positive, is satisfied for all

$$\gamma \geqslant \Psi_j + \sqrt{\Psi_j^2 + \varphi_1\varphi_j}. \qquad \square$$

The previous lemma allows us to state the desired results.

Theorem 4. *Assume the hypotheses of Lemma* 3 *to be satisfied and, moreover, assume that*

$$\varphi_1 > \hat{\gamma}, \tag{29}$$

where $\hat{\gamma}$ *is defined according to* (27). *It follows that the minimum value of* ρ^* *is obtained at* $\gamma = \varphi_1$, *and it is given by* $\rho^* = 1 - \cos\zeta_1$. *If, in addition,*

$$\varphi_1 \cos\zeta_1 > \hat{\gamma}, \tag{30}$$

then the minimum value of $\tilde{\rho}$ *is obtained at* $\gamma = \varphi_1\cos\zeta_1$, *and it is given by* $\tilde{\rho} = \varphi_1 \sin^2\zeta_1$.

Proof. Let us first consider the first point. By taking into account (25), we want to solve the problem

$$\min_{\gamma>0} \max_{j\in\{1,\dots,\ell\}} \frac{|\varphi_j e^{i\zeta_j} - \gamma|^2}{2\gamma\varphi_j}.$$

If such a minimum would be obtained at a value of $\gamma \geqslant \hat{\gamma}$ (see (27)) then, from Lemma 3, the previous problem would reduce to the following simpler one,

$$\min_{\gamma>0} \frac{\varphi_1^2 + \gamma^2 - 2\varphi_1\gamma\cos\zeta_1}{2\gamma\varphi_1} = \min_{\gamma>0} \frac{1}{2}\left(\frac{\varphi_1}{\gamma} + \frac{\gamma}{\varphi_1} - 2\cos\zeta_1\right) \equiv \min_{\gamma>0} g^*(\gamma).$$

Indeed, by considering that the only stationary point of g^* is given by $\frac{dg^*}{d\gamma}(\varphi_1) = 0$ and, moreover, $\frac{d^2g^*}{(d\gamma)^2}(\varphi_1) > 0$, one then obtains that, from (29), at $\gamma = \varphi_1$, $\rho^* = g^*(\varphi_1) \equiv 1 - \cos\zeta_1$.

Similarly, for the second point we obtain that

$$\min_{\gamma>0} \max_{j\in\{1,\dots,\ell\}} \frac{|\varphi_j e^{i\zeta_j} - \gamma|^2}{\varphi_j} = \min_{\gamma>0} \frac{\varphi_1^2 + \gamma^2 - 2\varphi_1\gamma\cos\zeta_1}{\varphi_1} = \min_{\gamma>0}\left(\varphi_1 + \frac{\gamma^2}{\varphi_1} - 2\gamma\cos\zeta_1\right) \equiv \min_{\gamma>0} \tilde{g}(\gamma),$$

provided that the minimum is obtained at a value of $\gamma \geqslant \hat{\gamma}$. Indeed, by considering that the only stationary point of $\tilde{g}$ is given by $\frac{d\tilde{g}}{d\gamma}(\varphi_1\cos\zeta_1) = 0$ and, moreover, $\frac{d^2\tilde{g}}{(d\gamma)^2}(\varphi_1\cos\zeta_1) > 0$, one then obtains that, from (30), at $\gamma = \varphi_1\cos\zeta_1$, $\tilde{\rho} = \tilde{g}(\varphi_1\cos\zeta_1) \equiv \varphi_1\sin^2\zeta_1$. $\square$

Remark 5. We observe that the above relation (27)–(29) can be also written as

$$\frac{\varphi_j}{\varphi_1} + \frac{\varphi_1}{\varphi_j} < 2(1 + \cos\zeta_j - \cos\zeta_1), \quad j = 2,\dots,\ell. \tag{31}$$

By taking into account (26), the previous inequality implies that all the eigenvalues of the matrix B_1 are contained in a suitably small annulus, whose internal radius is φ_1. A similar conclusion can be obtained from (27)–(30),

$$\frac{\varphi_j}{\varphi_1} + \frac{\varphi_1}{\varphi_j}\cos^2\zeta_1 < 1 + \cos^2\zeta_1 + 2\cos\zeta_1(\cos\zeta_j - \cos\zeta_1), \quad j = 2,\dots,\ell, \tag{32}$$

which, however, is more restrictive than (31).

It turns out that both results in Theorem 4 apply to the case of the methods considered at the end of Section 3: indeed, according to what was stated in Remark 5, it can be shown that all the eigenvalues of the matrix B_1 are contained in a suitably small annulus (see [4,16]), and (32) (and then (31)) turns out to be satisfied. In Table 1 we list the obtained values of the parameters $\tilde{\rho}$ and ρ^*, with the choice $\gamma = \varphi_1$. We omit to list the case corresponding to the choice $\gamma = \varphi_1 \cos\zeta_1$, since the latter choice does not guarantee the A-convergence of the corresponding iteration (13) and (14).

When the blended implementation does not satisfy (22), then the above analysis cannot be applied, since the involved matrices no more commute. In such a case, one must resort to computational techniques in order to minimize either one of the two parameters (15) and (16). In such a case, it is useful to know that $\tilde{\rho}$ is given by the spectral radius of the following matrix,

$$R = A_1^{-1}\big(B_1 - B_2 + D(A_2 - A_1)\big). \tag{33}$$

Concerning alternative choices for the matrices A_1 and B_2, we have considered the case where

$$A_1 = \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix}, \qquad B_2 = \gamma I, \tag{34}$$

with $A_2 = \gamma B_1^{-1} A_1$, and $\theta(q) = (1 - q\gamma)^{-1} I$. In Table 2 we list the obtained results when choosing γ so as to minimize ρ^* and $\sqrt{\tilde{\rho}\|R\|_2}$, respectively, where R is the matrix defined in (33). As told before, in such a case the parameters have been computed numerically. We observe that the choice of minimizing $\sqrt{\tilde{\rho}\|R\|_2}$ makes the method corresponding to $r = 12$ not A-convergent (though $A(\alpha)$-convergent [12] with $\alpha \approx \pi/2$). Hereafter, we shall refer to the schemes corresponding to the above three choices as the

Table 1
Values of the parameters ρ^* and $\tilde{\rho}$ for the methods satisfying (22)

r	Padé	γ	ρ^*	$\tilde{\rho}$
3	(2,3)	0.7387	0.3398	0.5021
4	(2,4)	0.8482	0.5291	0.8975
6	(4,6)	0.7285	0.6299	0.9177
8	(6,8)	0.6745	0.6885	0.9288
10	(8,10)	0.6433	0.7276	0.9361
12	(10,12)	0.6227	0.7560	0.9415

Table 2
Values of the parameters ρ^* and $\tilde{\rho}$ for the methods satisfying (34)

r	Padé	γ	ρ^*	$\tilde{\rho}$	γ	ρ^*	$\tilde{\rho}$
3	(2,3)	0.6884	0.2672	0.3366	0.5802	0.2998	0.2692
4	(2,4)	0.8351	0.4045	0.4513	0.5960	0.5427	0.3833
6	(4,6)	0.7677	0.5184	0.4747	0.5165	0.6687	0.4310
8	(6,8)	0.6151	0.5428	0.6032	0.4472	0.7858	0.4389
10	(8,10)	0.6046	0.6475	0.6884	0.4088	0.9017	0.4408
12	(10,12)	0.5819	0.7400	0.7462	0.3866	1.0004	0.4583

type 1, 2, and 3 *schemes*, respectively. A comparative analysis of Tables 1 and 2 puts into evidence the type 2 schemes as the ones with the best features, from the point of view of the amplification factors. Nevertheless, the diagonal splitting characterizing the type 1 schemes makes them very appealing for an implementation on parallel computers.

4.1. Order variation

We now briefly examine some details concerning the mesh and the order variation for the methods considered above. First of all, we observe that the iteration (13) and (14) becomes, for problem (1) (again, for simplicity let us consider the first application of the method),

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} - (A_1 \otimes I_m - hB_2 \otimes J)^{-1}\big(\theta\big((A_1 - A_2) \otimes I_m \mathbf{y}^{(i)} - h(B_1 - B_2) \otimes I_m \mathbf{f}^{(i)}\big) + A_2 \otimes I_m \mathbf{y}^{(i)} - hB_2 \otimes I_m \mathbf{f}^{(i)} + \boldsymbol{\eta}\big), \quad i = 1, 2, \ldots, \quad (35)$$

where $\mathbf{f}^{(i)} = (f_1^{(i)}, \ldots, f_r^{(i)})^{\mathrm{T}}$, J is the Jacobian matrix of f at (t_0, y_0), the vector $\boldsymbol{\eta}$ only depends on the initial condition, and (assuming to deal with the type 1, 2 or 3 schemes previously introduced) $\theta = I \otimes (I_m - h\gamma J)^{-1}$. Consequently, if ν iterations are performed to obtain convergence, the overall computational cost is approximately given by:

- the factorization of the $m \times m$ matrix $\Omega = I_m - h\gamma J$,
- $r\nu$ function evaluations, and
- $2r\nu$ system solvings with the factors of the matrix Ω.

Considering that the last point is the most time consuming section, for small-medium size problems, we now shall briefly sketch a variable order strategy with the aim of reducing this cost. As usual, during the ν iterations (35), one is able to get an estimate, say ρ, of the spectral radius of the iteration matrix, which will depend on the stepsize h and on the eigenvalues of the matrix J. Assuming that the stepsize h_{new} is to be used in the next step, one then obtains that the spectral radius of the new iteration matrix is approximately given by (see (16))

$$\rho_{\mathrm{new}} = \rho \frac{h_{\mathrm{new}}}{h}.$$

If the same stopping criterion has to be satisfied, then approximately

$$\nu_{\mathrm{new}} = \nu \frac{\log \rho}{\log \rho_{\mathrm{new}}}$$

iterations will be required. Now, assume that we are able to know the stepsize, say h_{up}, to be used by the next higher order method, among those listed at the end of Section 3. Consequently, the spectral radius of the corresponding iteration matrix, and the number of iterations to get convergence, can be respectively estimated as

$$\rho_{\mathrm{up}} = \rho \frac{h_{\mathrm{up}} \tilde{\rho}_{\mathrm{up}}}{h \tilde{\rho}}, \qquad \nu_{\mathrm{up}} = \nu_{\mathrm{new}} \frac{\log \rho_{\mathrm{new}}}{\log \rho_{\mathrm{up}}},$$

where $\tilde{\rho}$ and $\tilde{\rho}_{\mathrm{up}}$ are the (known) nonstiff amplification factors (see (16)) of the current method and of the higher order one.

If we normalize the cost by dividing the number of required linear systems by the covered integration interval, we have that the normalized cost for the current method is given by

$$\frac{2r(\nu_{\text{new}}+1)}{rh_{\text{new}}} = \frac{2(\nu_{\text{new}}+1)}{h_{\text{new}}}. \tag{36}$$

Consequently, the higher order method has to be preferred when

$$\frac{\nu_{\text{new}}+1}{h_{\text{new}}} > \frac{\nu_{\text{up}}+1}{h_{\text{up}}}, \tag{37}$$

which can be readily computed, provided that estimates for h_{new} and h_{up} are available. Concerning this point, let us consider the local error of the currently used method, say $\mathbf{e} = (e_1, \ldots, e_r)^{\mathrm{T}}$: from Section 3, its entries behave as $\mathrm{O}(h^{r+1})$, with the only exception of the last entry, which is $\mathrm{O}(h^{r+2})$, for $r = 3$, and $\mathrm{O}(h^{r+3})$, for $r = 4, 6, 8, 10, 12$. Therefore, if we use the following norm for measuring the error,

$$\max_{i=1,\ldots,r} \frac{\|e_i\|_2}{\sqrt{m}},$$

then $m^{-1/2}\|e_r\|_2$ provides an estimate of the error for the closest higher order method. Consequently, when estimating the local error to predict the new stepsize h_{new}, we are also able to predict, *at no extra cost*, the stepsize h_{up} for the higher order method.

In such a way, we have a simple criterion to decide whether to increase the order of the method. Conversely, the order is reduced when the iteration (35) fails to converge, or when the estimated value

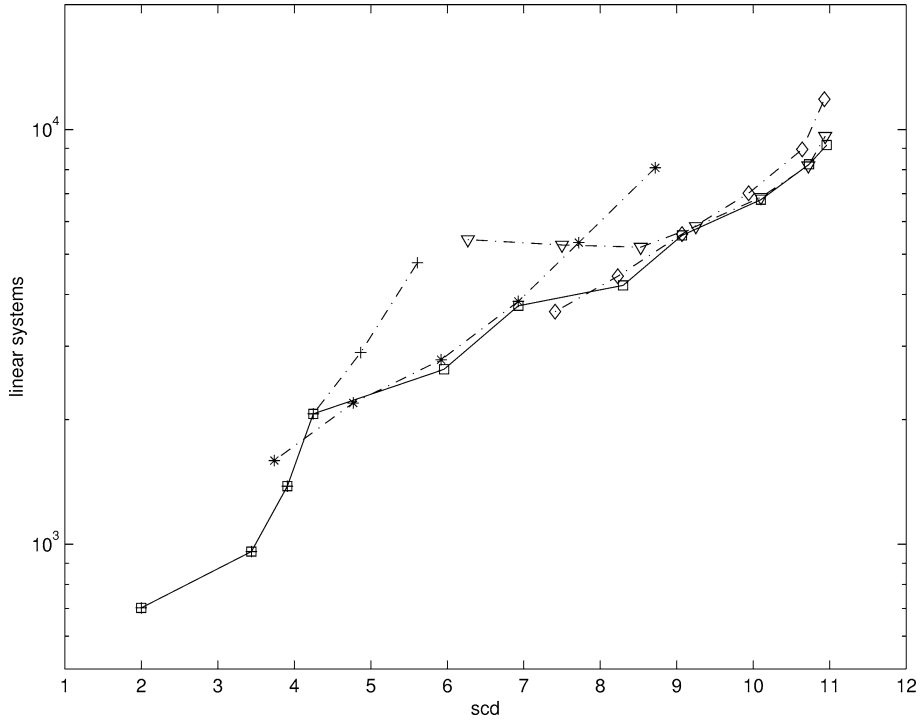

Fig. 1. Variable versus fixed order implementation.

of the spectral radius of the iteration matrix, ρ, is too large. We also mention that the local error can be estimated, by using deferred correction (see, e.g., [3]), at the cost of one iteration. This extra iteration is the reason for the "+1" increment at the numerators in (36) and (37) (full details will be provided in a future paper).

In order to put into evidence the effectiveness of the presented variable order strategy, in Fig. 1 we plot the work precision diagram, with the number of required system solvings with the factors of the matrix Ω, for the type 2 schemes applied to the Robertson problem. It can be seen that the plot of the variable order method (solid line and squares) is almost everywhere below those of the fixed order ones, with order 4 (pluses), 6 (stars), 8 (rhombuses) and 10 (triangles).

5. Numerical tests

We report here a few numerical tests to compare the proposed blended schemes (implemented in Matlab) with one of the best codes currently available, i.e., the Fortran code GAM [13]. This is because the methods used in such codes are implemented by using a nonlinear splitting, still requiring one $m \times m$ factorization per step, and an equal number of system solvings and function evaluations per inner iteration, depending on the (variable) order of the method used. In this respect, we recall that the blended schemes require twice the number of system solvings per inner iteration, with respect to the function evaluations.

In Fig. 2 are the results obtained for the Van der Pol problem, where we plot the work precision diagrams, with the needed linear system solvings and the function evaluations, for the GAM code (solid line and pluses), the type 1 schemes (dashed line and downward triangles), the type 2 schemes (dash-dotted line and rhombuses), and the type 3 ones (dotted line and squares), as defined in Section 4. The symbols are the same for Figs. 3 and 4, where we plot the results for the Robertson problem and the Ring Modulator problem (the latter problem from the CWI testset [14]). For the type 1 schemes we recall that the splitting is diagonal. This implies that the method with blocksize r can be implemented on r parallel processors with a perfect degree of parallelism, for what concerns the system solvings and the function evaluations. For this reason, for the type 1 schemes we also plot the parallel cost (dashed lines and upward triangles), obtained by considering that the parallel complexity per iteration is 2 system solvings and 1 function evaluation.

We observe that all methods require a comparable number of linear system solvings. On the other hand, the blended schemes require approximately half the number of function evaluations as the GAM code. Moreover, we stress that the type 1 schemes, characterized by a diagonal splitting, do have good potentialities for a parallel implementation.

5.1. Conclusions

In this paper we have studied the solution of the discrete problems generated by the application of block implicit methods for ODEs. By suitably *blending* two discrete problems corresponding to the same method, it is indeed possible to naturally define a corresponding nonlinear splitting. By properly choosing the methods, and the discrete problems to combine, it has been shown that very efficient splittings can be obtained. In particular, a diagonal splitting has been defined, which seems to be promising for the implementation on parallel computers.

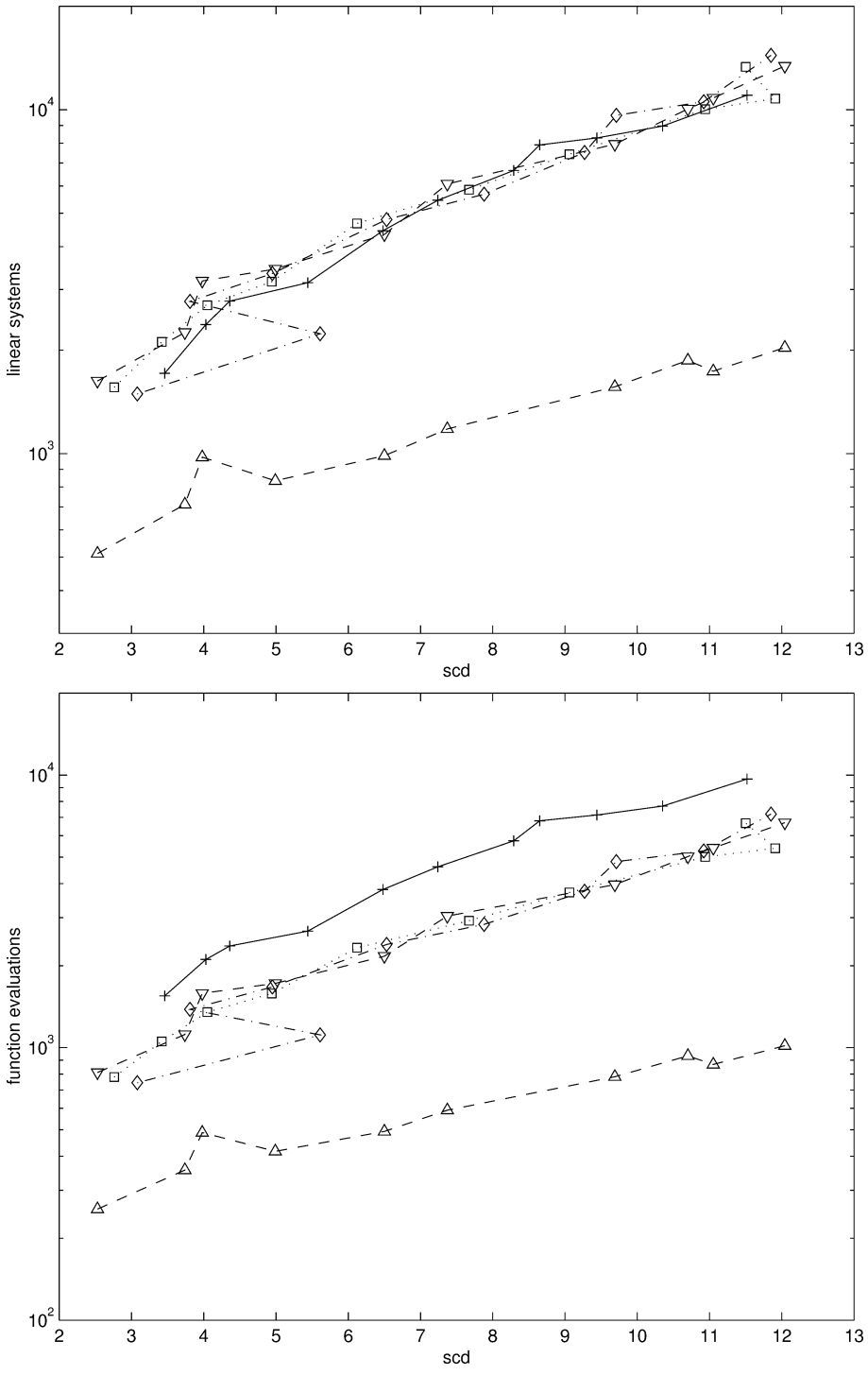

Fig. 2. Results for the Van der Pol problem.

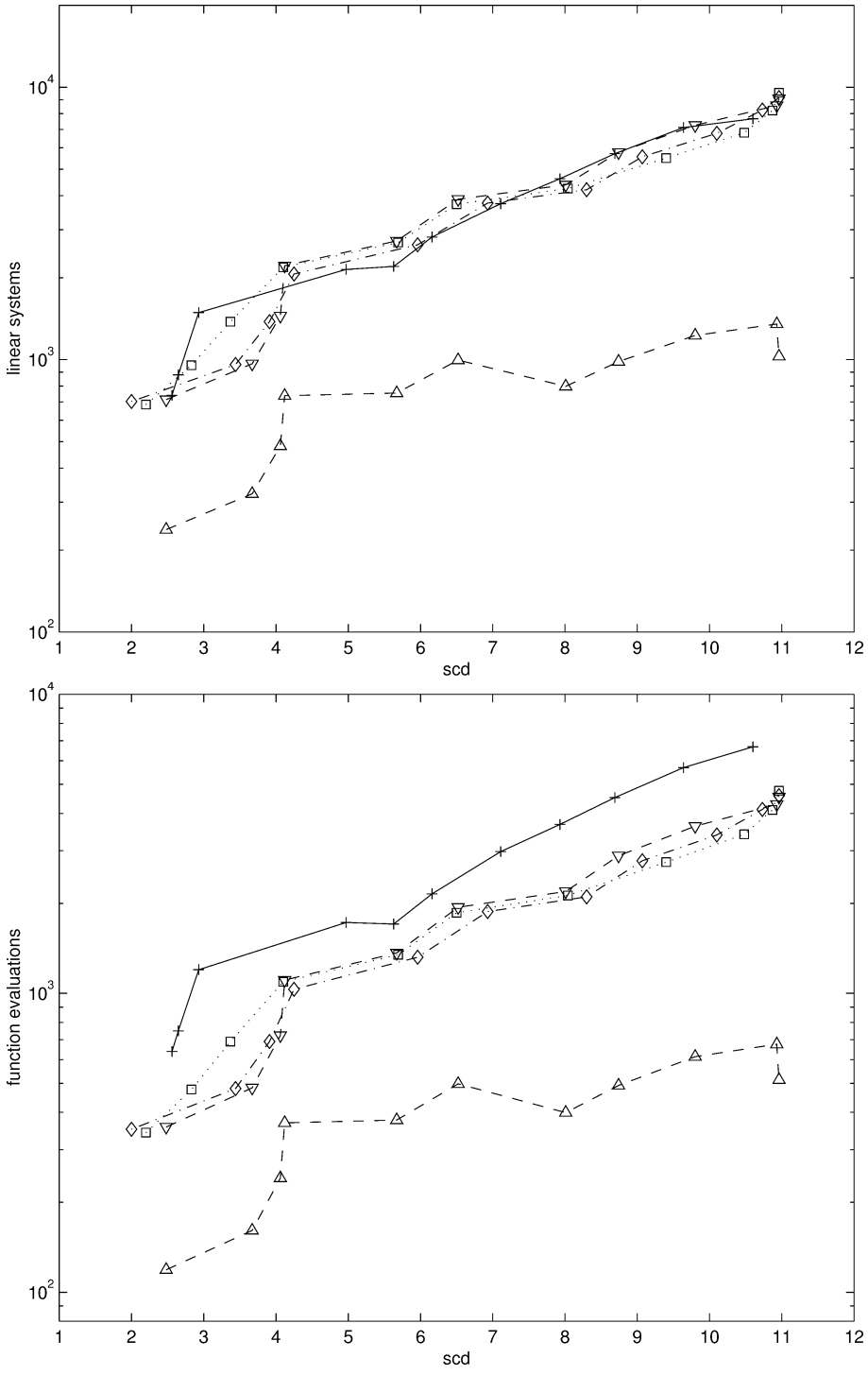

Fig. 3. Results for the Robertson problem.

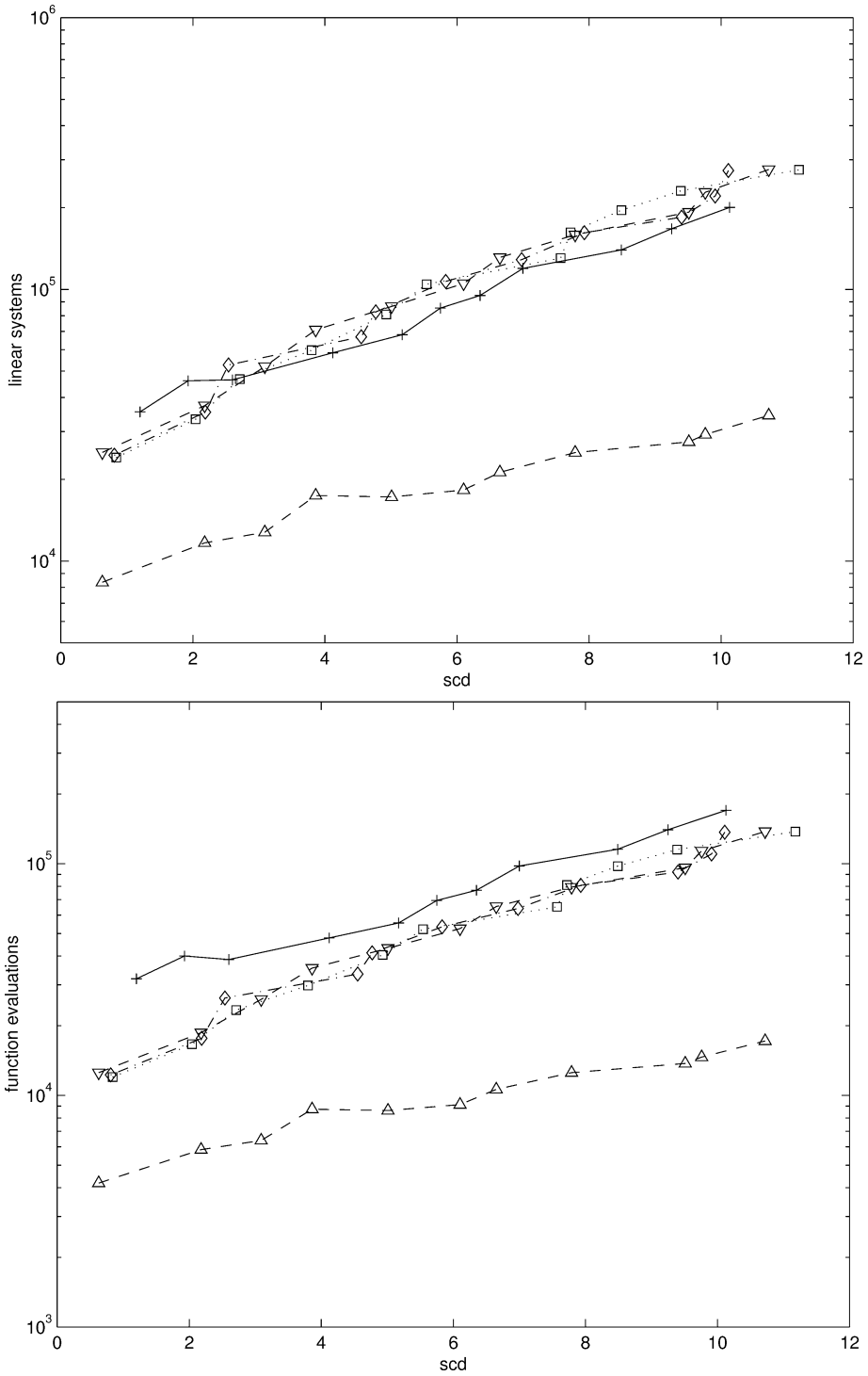

Fig. 4. Results for the Ring Modulator problem.

Acknowledgement

The authors are very indebted to Professor Donato Trigiante for the valuable advice and to the referees for their suggestions.

References

[1] P. Amodio, L. Brugnano, A note on the efficient implementation of implicit methods for ODEs, J. Comput. Appl. Math. 87 (1997) 1–9.
[2] L. Brugnano, Blended block BVMs (B_3VMs): A family of economical implicit methods for ODEs, J. Comput. Appl. Math. 116 (2000) 41–62.
[3] L. Brugnano, D. Trigiante, Solving Differential Problems by Multistep Initial and Boundary Value Methods, Gordon and Breach, London, 1998.
[4] L. Brugnano, D. Trigiante, Block implicit methods for ODEs, in: D. Trigiante (Ed.), Recent Trends in Numerical Analysis, Nova Science, New York, 2000, pp. 81–105.
[5] K. Burrage, A special family of Runge–Kutta methods for solving stiff differential equations, BIT 18 (1978) 22–41.
[6] K. Burrage, High order algebraically stable Runge–Kutta methods, BIT 18 (1978) 373–383.
[7] J.C. Butcher, On the implementation of implicit Runge–Kutta methods, BIT 6 (1976) 237–240.
[8] J.C. Butcher, The Numerical Analysis of Ordinary Differential Equations: Runge–Kutta Methods General Linear Methods, John Wiley, Chichester, 1987.
[9] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, 2nd edn., Springer Ser. Comput. Math., Vol. 8, Springer, Berlin, 1993.
[10] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Stiff and Differential–Algebraic Problems, Springer, Berlin, 1991.
[11] P.J. van der Houwen, J.J.B. de Swart, Triangularly implicit iteration methods for ODE-IVP solvers, SIAM J. Sci. Comput. 18 (1997) 41–55.
[12] P.J. van der Houwen, J.J.B. de Swart, Parallel linear system solvers for Runge–Kutta methods, Adv. Comput. Math. 7 (1997) 157–181.
[13] F. Iavernaro, F. Mazzia, Solving ordinary differential equations by generalized Adams methods: Properties and implementation techniques, Appl. Numer. Math. 28 (1998) 107–126. Code available at http://www.dm.uniba.it/mazzia/ode/readme.html.
[14] W.M. Lioen, J.J.B. de Swart, W.A. van der Veen, Test set for IVP solvers, Report NM-R96150, CWI, Department of Mathematics, Amsterdam, 1996.
[15] S.P. Nørsett, Runge–Kutta methods with a multiple real eigenvalue only, BIT 16 (1976) 388–393.
[16] E.B. Saff, R.S. Varga, On zeros and poles of Padé approximants to e^z. III, Numer. Math. 30 (1978) 241–266.
[17] R.D. Skeel, A.K. Kong, Blended linear multistep methods, ACM Trans. Math. Software 3 (1977) 326–345.
[18] G. Wanner, E. Hairer, S.P. Nørsett, Order stars and stability theorems, BIT 18 (1978) 475–489.
[19] H.A. Watts, L.F. Shampine, *A*-stable block one-step methods, BIT 12 (1972) 252–266.