ELSEVIER

Journal of Computational and Applied Mathematics 116 (2000) 41–62

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.nl/locate/cam

Blended block BVMs (B_3VMs): A family of economical implicit methods for ODEs ☆

Luigi Brugnano

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

Received 2 March 1998; received in revised form 2 July 1999

Abstract

We analyze some properties of block BVMs for ODEs and introduce blended block BVMs. The latter are implicit methods characterized by a cheap iterative implementation, which makes them computationally very appealing.

MSC: 65L06; 65L05; 65L20; 65H10; 65F10

Keywords: Numerical methods for ODEs; Boundary value method; Iterative solution of linear systems

1. Introduction

In the past years, many attempts have been made in order to derive numerical methods for ODEs obtained by combining simpler methods. A classical instance is given by the popular θ-method. Another one is given by the so-called *blended methods* [7,12]. In both cases, the resulting scheme is obtained as a suitable combination of two basic methods belonging to the class of linear multistep formulae (LMF).

In the above-mentioned examples, the main reason for combining different methods was that of getting better qualitative behavior for the resulting formula, than that of the single-component methods. This is evident in the case of the θ-method, where one tries to get a numerical scheme

☆ Work supported by C.N.R. (contract n.96.00243.CT01) and M.U.R.S.T.

E-mail address: na.brugnano@na-net.ornl.gov (L. Brugnano)

0377-0427/00/$ - see front matter
PII: S0377-0427(99)00280-0

which well performs, when applied to the usual test equation, both for $q \approx 0$ and for $q \to \infty$. Similar considerations hold true for blended methods.

In this paper we shall consider a similar approach by using block boundary value methods (B_2VMs), recently introduced by Brugnano and Trigiante [3–5], which have been used in the code GAM written by Iavernaro and Mazzia [11]. In particular, we shall show that suitable combinations of B_2VMs may produce methods which:

1. may have better properties than those of the two component B_2VMs;
2. allow a cheaper implementation, than that of the single-component methods.

Both aspects are, indeed, very important and will be examined in the sequel. In particular, the use of the presented methods is intended to overcome some of the problems raised in the final section in [1], as we shall see in more details in Section 6.

The structure of the paper is the following: Section 2 is devoted to state the main facts about B_2VMs. In Section 3 we introduce the new class of methods, called Blended B_2VMs (B_3VMs). In Sections 4 and 5 we study two families of methods in this class. Finally, in Section 6 some numerical tests are reported.

2. Block boundary value methods (B_2VMs)

Block boundary value methods (B_2VMs) have been derived as a particular implementation of boundary value methods (BVMs), which is a relatively new class of numerical methods for differential equations (see [5] for a complete treatment of this subject). In order to briefly sketch such methods, let us consider the following initial value problem for ODEs,

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

to be approximated over the discrete set of points $t_i = t_0 + ih$, $i = 0, \ldots, N$, where $h = (T - t_0)/N$ is the stepsize. One may then obtain an approximation over the first $s \leqslant N$ points by using a B_2VM based on k-step formulae, which defines the following discrete problem:

$$A \otimes I_m \boldsymbol{y} - hB \otimes I_m \boldsymbol{f} = h\boldsymbol{b} \otimes f(t_0, \eta) - \boldsymbol{a} \otimes \eta, \tag{2}$$

where I_m is the identity matrix of size m (the dimension of the continuous problem), and the vectors $\boldsymbol{y}$ and $\boldsymbol{f}$ contain the discrete solution and the corresponding values of the function f, respectively: evidently, once problem (2) has been solved, the last block entry of $\boldsymbol{y}$ will provide the initial condition for the subsequent integration, and so on. Finally, the augmented

matrices

$$\hat{A} \equiv [\boldsymbol{a} \,|\, A] = \left(\begin{array}{c|cccc} \alpha_0^{(1)} & \alpha_1^{(1)} & \dots & \alpha_k^{(1)} & \\ \vdots & \vdots & & \vdots & \\ \alpha_0^{(k_1-1)} & \alpha_1^{(k_1-1)} & \dots & \alpha_k^{(k_1-1)} & \\ \alpha_0^{(k_1)} & \alpha_1^{(k_1)} & \dots & \alpha_k^{(k_1)} & \\ & \alpha_0^{(k_1)} & \alpha_1^{(k_1)} & \dots & \alpha_k^{(k_1)} \\ & \ddots & \ddots & & \ddots \\ & \alpha_0^{(k_1)} & \alpha_1^{(k_1)} & \dots & \alpha_k^{(k_1)} \\ & \alpha_0^{(k_1+1)} & \alpha_1^{(k_1+1)} & \dots & \alpha_k^{(k_1+1)} \\ & \vdots & \vdots & & \vdots \\ & \alpha_0^{(k)} & \alpha_1^{(k)} & \dots & \alpha_k^{(k)} \end{array}\right)_{s\times(s+1)} \tag{3}$$

and

$$\hat{B} \equiv [\boldsymbol{b} \,|\, B] = \left(\begin{array}{c|cccc} \beta_0^{(1)} & \beta_1^{(1)} & \dots & \beta_k^{(1)} & \\ \vdots & \vdots & & \vdots & \\ \beta_0^{(k_1-1)} & \beta_1^{(k_1-1)} & \dots & \beta_k^{(k_1-1)} & \\ \beta_0^{(k_1)} & \beta_1^{(k_1)} & \dots & \beta_k^{(k_1)} & \\ & \beta_0^{(k_1)} & \beta_1^{(k_1)} & \dots & \beta_k^{(k_1)} \\ & \ddots & \ddots & & \ddots \\ & \beta_0^{(k_1)} & \beta_1^{(k_1)} & \dots & \beta_k^{(k_1)} \\ & \beta_0^{(k_1+1)} & \beta_1^{(k_1+1)} & \dots & \beta_k^{(k_1+1)} \\ & \vdots & \vdots & & \vdots \\ & \beta_0^{(k)} & \beta_1^{(k)} & \dots & \beta_k^{(k)} \end{array}\right)_{s\times(s+1)} \tag{4}$$

are defined such that the corresponding LMF,

$$\sum_{i=0}^{k} \alpha_i^{(j)} y_{n+i} = h \sum_{i=0}^{k} \beta_i^{(j)} f_{n+i}, \quad j = 1,\dots,k, \tag{5}$$

all have a $O(h^{p+1})$ truncation error, i.e. order p. In such a case, it is not difficult to prove that the block method has order of convergence at least p (see [10,13]). Among the previous formulae, the k_1th one is repeated $s-k+1$ times in the two matrices (3) and (4) (even though, each repetition acts on different components of the vectors $\boldsymbol{y}$ and $\boldsymbol{f}$). This method is called *main method*, whereas the remaining ones are called *additional methods*. Concerning the value of k_1, we shall here consider the following choice:

$$k_1 = v \quad \text{for } k \in \{2v-1, 2v\}. \tag{6}$$

From (3)–(4), one obtains that any *blocksize* $s \geqslant k$ is allowed for a B_2VM based on k-step formulae. So far, blocksizes strictly greater than k have been successfully used (for example, in the code GAM). Nevertheless, we shall often consider the value $s = k$, which is the minimum value allowed.

Definition 1. A B_2VM based on k-step methods and having blocksize $s = k$ is called a *minimal* B_2VM.

We observe that, for minimal B_2VMs, the two matrices (3) and (4) become, respectively,

$$\hat{A} \equiv [\boldsymbol{a} \,|\, A] = \left(\begin{array}{c|ccc} \alpha_0^{(1)} & \alpha_1^{(1)} & \dots & \alpha_k^{(1)} \\ \vdots & \vdots & & \vdots \\ \alpha_0^{(k)} & \alpha_1^{(k)} & \dots & \alpha_k^{(k)} \end{array}\right)_{k\times(k+1)}, \tag{7}$$

$$\hat{B} \equiv [\boldsymbol{b} \,|\, B] = \left(\begin{array}{c|ccc} \beta_0^{(1)} & \beta_1^{(1)} & \dots & \beta_k^{(1)} \\ \vdots & \vdots & & \vdots \\ \beta_0^{(k)} & \beta_1^{(k)} & \dots & \beta_k^{(k)} \end{array}\right)_{k\times(k+1)}. \tag{8}$$

Concerning the matrices $\hat{A}$ and $\hat{B}$, the following result holds true.

Theorem 2. *Let the two matrices $\hat{A}$ and $\hat{B}$ be defined according to* (3)–(4), *and let the corresponding* B_2 VM *have a* $O(h^{p+1})$ *truncation error. Then one has*

$$W_p^{(0:s)}\hat{A}^{\mathrm{T}} = H_p W_p^{(0:s)}\hat{B}^{\mathrm{T}}, \tag{9}$$

where

$$W_p^{(0:s)} = \begin{pmatrix} 0^0 & 1^0 & \dots & s^0 \\ 0^1 & 1^1 & \dots & s^1 \\ \vdots & \vdots & & \vdots \\ 0^p & 1^p & \dots & s^p \end{pmatrix}, \qquad H_p = \begin{pmatrix} 0 & 0 & \dots & \dots & 0 \\ 1 & 0 & \dots & \dots & 0 \\ 0 & 2 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & p & 0 \end{pmatrix}.$$

Proof. In fact, (9) is nothing but the expression of the order conditions for each method corresponding to a given row of $\hat{A}$ and $\hat{B}$. □

Other simple properties, to be used later, can be derived from the above result.

Corollary 3. *Suppose that the minimal* B_2 VM *given by the two matrices* (7)–(8) *has a* $O(h^{k+2})$ *truncation error. Then A is nonsingular if and only if B is nonsingular.*

Proof. In fact, from Theorem 2, for $s = k$ and $p = k + 1$ one obtains that

$$W_{k+1}^{(0:k)}\hat{A}^{\mathrm{T}} = H_{k+1}W_{k+1}^{(0:k)}\hat{B}^{\mathrm{T}}.$$

By neglecting the first two equations, one has then

$$\begin{aligned} O &= W_{k-1}^{(0:k)}(D^{(0:k)})^2\hat{A}^{\mathrm{T}} - D^{(2:k+1)}W_{k-1}^{(0:k)}D^{(0:k)}\hat{B}^{\mathrm{T}} \\ &= W_{k-1}^{(1:k)}(D^{(1:k)})^2A^{\mathrm{T}} - D^{(2:k+1)}W_{k-1}^{(1:k)}D^{(1:k)}B^{\mathrm{T}}, \end{aligned}$$

where, in general, $D^{(i:j)} = \mathrm{diag}(i, i+1, \ldots, j)$. From the above relations, it then follows that

$$A^{\mathrm{T}} = (D^{(1:k)})^{-2}(W_{k-1}^{(1:k)})^{-1}D^{(2:k+1)}W_{k-1}^{(1:k)}D^{(1:k)}B^{\mathrm{T}}, \tag{10}$$

from which the thesis follows, by considering that $W_{k-1}^{(1:k)}$ is a Vandermonde matrix. □

We observe that B_2VMs have the matrix A nonsingular by construction (this is a minimal requirement, in order to reproduce constant solutions). If they are also consistent (which we obviously assume) they can be also rewritten as Runge–Kutta methods. In fact, in (2) multiplication on the left by $A^{-1} \otimes I_m$ gives

$$I_s \otimes I_m \boldsymbol{y} - hA^{-1}B \otimes I_m \boldsymbol{f} = hA^{-1}\boldsymbol{b} \otimes f(t_0, \eta) + \boldsymbol{e} \otimes \eta, \tag{11}$$

where $\boldsymbol{e} \equiv -A^{-1}\boldsymbol{a}$ is the vector with all unit entries, from consistency. The expression (11) defines the following RK method with $s + 1$ stages [5] used with stepsize sh,

$$\begin{array}{c|c} 0 & \mathbf{0}_{s+1}^{\mathrm{T}} \\ 1/s & \\ \vdots & (1/s)A^{-1}\hat{B} \\ s/s & \\ \hline & \boldsymbol{e}_s^{\mathrm{T}}(1/s)A^{-1}\hat{B} \end{array} \tag{12}$$

where $\boldsymbol{e}_s$ is the last unit vector in $\mathbb{R}^s$, and $\mathbf{0}_{s+1}$ is the zero vector in $\mathbb{R}^{s+1}$.

Remark 4. From the previous arguments, it follows that the matrix $A^{-1}\hat{B}$ uniquely defines tableau (12). Actually, because of consistency conditions, the above method is uniquely determined once the matrix $A^{-1}B$ is given. In such a case, in fact, the vector $A^{-1}\boldsymbol{b}$ (see (4)) is fixed. Consequently, we may consider as *equivalent* B_2VMs having the same matrix $A^{-1}B$. It must be stressed, however, that the discrete problems generated by equivalent B_2VMs are different. As a consequence, the methods for their efficient solution may greatly vary, depending on the particular problem, i.e. on the specific form of the matrices A and B.

The above consideration allows us to state the following result (see also [13]).

Corollary 5. *The maximum order of the k-step formulae* (5) *defining a minimal* B_2 VM *is* $p = k+1$. *All minimal* B_2VMs *satisfying such a property have the same matrix* $A^{-1}B$. *They are, therefore, equivalent.*

Proof. In the above hypothesis, it follows that the truncation error of the B_2VM is $O(h^{k+2})$. Consequently, the result of Corollary 3 applies and, from (10), one obtains

$$A^{-1}B = (D^{(1:k)})^2 (W_{k-1}^{(1:k)})^{\mathrm{T}} (D^{(2:k+1)})^{-1} (W_{k-1}^{(1:k)})^{-\mathrm{T}} (D^{(1:k)})^{-1}. \tag{13}$$

The thesis then follows by considering that the right-hand side is independent of the particular method. □

Moreover, it can be shown (see [13]) that the order of convergence of minimal B_2VMs having a $O(h^{k+2})$ truncation error is $k+1$ for k odd, and $k+2$ for k even. When the B_2VM is nonminimal, the order is $k+1$ for all $k \geqslant 1$.

Equality (13) allows us to get some additional consideration concerning stability issues for minimal B_2VMs. In fact, let us consider the usual test equation,

$$y' = \lambda y, \qquad y(t_0) = \eta. \tag{14}$$

The application of (2) then gives the following discrete problem:

$$(A - qB)\boldsymbol{y} = (q\boldsymbol{b} - \boldsymbol{a})\eta, \qquad q = h\lambda. \tag{15}$$

The study of the above equation permits to derive the linear stability properties of B_2VMs [5,10]. In particular, the method is A-stable when $|\boldsymbol{e}_s^{\mathrm{T}}\boldsymbol{y}| < |\eta|$, for all $q \in \mathbb{C}^-$. Nevertheless, this requires the above problem to be well posed for all such q. For this reason, we give the following definition.

Definition 6. A B_2VM is said to be *pre-stable* if the spectrum of the corresponding matrix pencil $A - \mu B$ is contained in $\mathbb{C}^+$.

By direct inspection, from (13) one may verify that the corresponding minimal B_2VMs have the spectrum of the matrix $A^{-1}B$ contained in $\mathbb{C}^+$ up to $k = 8$; in particular, it is possible to prove that such methods are perfectly A-stable (see, for example [13]). In general, for nonminimal B_2VMs we refer to [10].

3. Blended block boundary value methods (B_3VMs)

In this section, we introduce methods obtained as the combination of a couple of B_2VMs. For sake of simplicity, we shall at first describe them when they are applied to the test equation (14). Then, we look for methods generating a discrete problem in the following form:

$$M(q)\boldsymbol{y} \equiv (A(q) - qB(q))\boldsymbol{y} = (q\boldsymbol{b}(q) - \boldsymbol{a}(q))\eta, \tag{16}$$

where, being $\theta = \theta(q)$ a suitable "weight" function, $M(q)$ is a $s \times s$ matrix,

$$A(q) = \theta A_1 + (1-\theta)A_2, \qquad B(q) = \theta B_1 + (1-\theta)B_2, \tag{17}$$

$$\boldsymbol{a}(q) = \theta \boldsymbol{a}_1 + (1-\theta)\boldsymbol{a}_2, \qquad \boldsymbol{b}(q) = \theta \boldsymbol{b}_1 + (1-\theta)\boldsymbol{b}_2, \tag{18}$$

and the couples of augmented matrices

$$\hat{A}_1 = [\boldsymbol{a}_1 \,|\, A_1], \qquad \hat{B}_1 = [\boldsymbol{b}_1 \,|\, B_1], \qquad \hat{A}_2 = [\boldsymbol{a}_2 \,|\, A_2], \qquad \hat{B}_2 = [\boldsymbol{b}_2 \,|\, B_2], \tag{19}$$

define two suitable B_2VMs. Namely, the method is obtained as the combination of two *component* B_2VMs, with the weights of the combination depending on q. In analogy with [12], we call such a method *blended block boundary value method* (*B_3VM*). We observe that the B_3VM (16) is uniquely defined by the following couple of augmented matrices depending on q:

$$\hat{A}(q) = \theta\hat{A}_1 + (1-\theta)\hat{A}_2, \qquad \hat{B}(q) = \theta\hat{B}_1 + (1-\theta)\hat{B}_2. \tag{20}$$

The previous Definition 6 of pre-stability for B_2 VMs generalizes to B_3VMs as well, by requiring the matrix $M(q)$ in (16) to be nonsingular for all $q \in \mathbb{C}^-$.

Our concern is now that of appropriately choosing the two component B_2VMs, along with the function $\theta(q)$. They will be chosen by looking for methods which allow to solve linear systems in the form (see (16))

$$M(q)\boldsymbol{x} = \boldsymbol{c} \tag{21}$$

by using an iterative procedure,

$$N(q)\boldsymbol{x}_{r+1} = (N(q) - M(q))\boldsymbol{x}_r + \boldsymbol{c}, \quad r = 0, 1, \ldots, \tag{22}$$

where the linear system with the matrix $N(q)$ is much simpler to solve than (21). The above iteration converges to the solution of (21) if and only if the spectral radius, say $\rho(q)$, of the iteration matrix

$$(I_s - N(q)^{-1}M(q))$$

is smaller than 1. According to [8,9], the *region of convergence* of the iteration (22) is given by

$$\Gamma = \{q \in \mathbb{C}\colon \rho(q) < 1\}.$$

Moreover, the iteration is said to be *A-convergent* if $\mathbb{C}^- \subseteq \Gamma$ ($A(\alpha)$-convergence is similarly defined). A-convergence is a very remarkable property, as also stated by the following result.

Theorem 7. *Suppose that iteration* (22) *is A-convergent. Then the corresponding* B_3 VM (16) *is pre-stable.*

Proof. In fact, suppose that for a given $q \in \mathbb{C}^-$ the matrix $M(q)$ defined in (16) is singular. This means that the linear system (21) has either no solution or more than one solution. In both cases, this contradicts the fact that iteration (22) converges. □

Another important property which iteration (22) should enjoy is that

$$\rho(q) \to 0, \quad \text{as } q \to \infty. \tag{23}$$

Such a property, in fact, is desirable in order to have iteration (22) rapidly converging when the method is applied to stiff problems [8,9].

Taking into account all the above facts, we now give some practical criteria to define the component methods and the function $\theta(q)$ of the B_3VM (16). The latter function will be chosen so that

$$\theta(0) = 1, \qquad \theta(\infty) = 0. \tag{24}$$

Namely, for q "small" the B_3VM behaves essentially as the method defined by the matrices $\hat{A}_1$ and $\hat{B}_1$, whereas the method defined by $\hat{A}_2$ and $\hat{B}_2$ is the most effective when q is "large". From (24) one has that a good candidate for the weighting function $\theta(q)$ is in the form

$$\theta(q) = (1 - \beta q)^{-1}, \quad \beta > 0. \tag{25}$$

The fact that β is strictly positive, implies that the matrix $M(q)$ is well defined for all $q \in \mathbb{C}^-$, which is a necessary condition for A-convergence.

Let us now consider the problem of choosing the two B_2VMs defined by the couples $(\hat{A}_1, \hat{B}_1)$ and $(\hat{A}_2, \hat{B}_2)$. From (16)–(19) and (25), it follows that for q "small" the main contribute comes from the matrix $\hat{A}_1$. Consequently, for the first method, which should be more "active" in this range of q, we choose a B_2VM having such a matrix as simple as possible. Good candidates for this purpose are then given by Block GAMs [5,10,11], for which (see (3)) one has

$$\hat{A}_1 \equiv [\boldsymbol{a}_1 \,|\, A_1] = \left(\begin{array}{c|ccccc} -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & & -1 & 1 \end{array}\right)_{s\times(s+1)}, \tag{26}$$

and the coefficients of the corresponding matrix $\hat{B}_1$ (see (4) and (5)) are uniquely determined so that the truncation error on each equation defining the method is $O(h^{k+2})$. For what said in Section 2, the order of convergence of the method turns out to be $k+2$, for minimal methods with k even, and $k+1$ otherwise.

Hereafter, again because of what stated in Section 2, we shall consider Block GAMs up to $k = 8$.

Similarly, for q "large", the main contribute in (16) comes from the matrix $\hat{B}_2$. Consequently, for the second component method we shall choose such matrix as simple as possible. In the following, we shall consider two possibilities:

1. the first choice is

$$\hat{B}_2 \equiv [\boldsymbol{b}_2 \,|\, B_2] = [B_1^{-1}\boldsymbol{b}_1 \,|\, I_s], \qquad \hat{A}_2 = B_1^{-1}\hat{A}_1 \tag{27}$$

and the corresponding parameter in (25) is $\beta = 1$. We call the obtained B_3VMs *blended block GAMs* (*B_2GAMS*), either minimal ($s = k$) or nonminimal ($s \geqslant k+1$). Among nonminimal B_2GAMs, we shall study in more detail the case $s = k+1$;

2. the second choice that we consider is

$$\hat{B}_2 \equiv [\boldsymbol{b}_2 \,|\, B_2] = \left(\begin{array}{c|ccccc} 1-\beta & \beta & & & & \\ & 1-\beta & \beta & & & \\ & & \ddots & \ddots & & \\ & & & & 1-\beta & \beta \end{array}\right)_{s\times(s+1)}, \quad \beta \in (0.5, 1], \tag{28}$$

where β is the same parameter in (25), and the entries of the matrix $\hat{A}_2$, see (3)–(6), are uniquely determined in order that each equation of this B_2VM has a $O(h^{k+1})$ truncation error. We call the corresponding B_3VMs *hybrid blended block GAMs* (*HB_2GAMs*). In particular, we shall consider minimal HB_2GAMs, obtained for $s = k$.

Corresponding to each one of the above possibilities, the matrix $N(q)$ in (22) is chosen as

$$N(q) = A_1 - qB_2. \tag{29}$$

Such a choice, in fact, has several advantages:

- the matrix $N(q)$ (see (26)–(28)) is Toeplitz lower bidiagonal (block Toeplitz lower bidiagonal when the continuous problem is not scalar) and, therefore, easily and cheaply invertible;
- the matrix (29) defines a first-order implicit method, which can be used as a predictor for iteration (22). In more details, such method is the B_2VM defined by the two matrices $\hat{A_1}$ in (26) and $\hat{B}_2$ in (28), where it is assumed $\beta = 1$ in the case of B_2GAMs;
- property (23) for iteration (22) holds true. As a matter of fact, from (16), (17), (25) and (29) one has that

$$(I_s - N(q)^{-1}M(q)) \to O \quad \text{as } q \to \infty. \tag{30}$$

3.1. Evaluation parameters

In order to measure the convergence properties of iteration (22), we consider the following parameters [8] associated with it. The first, obvious, parameter is the maximum amplification factor for $q \in \mathbb{C}^-$, which, considering that the matrix $N(q)^{-1}M(q)$ is defined for all such values of q (see (16), (17), (25) and (29)), can be defined as

$$\rho^* = \sup_{\arg(q)=\pi/2} \rho(q). \tag{31}$$

Evidently, $\rho^* \leqslant 1$ for A-convergent methods. Moreover, again following [8], we introduce a couple of parameters which describe the convergence properties of the iteration (22) for $q \approx 0$ and $q \to \infty$.

In particular, for $q \approx 0$, one has that

$$I_s - N(q)^{-1}M(q) = N(q)^{-1}(N(q) - M(q)) \approx qA_1^{-1}(B_1 - B_2 + \beta A_2).$$

Consequently, we define the following *non stiff amplification factor*,

$$\tilde{\rho} = \max\{|\lambda|: \lambda \text{ eigenvalue of } A_1^{-1}(B_1 - B_2 + \beta A_2)\}. \tag{32}$$

Finally, we consider the *stiff amplification factor*,

$$\rho^{(\infty)} = \lim_{q\to\infty} \rho(q). \tag{33}$$

From (30) it follows that $\rho^{(\infty)} = 0$.

The previous parameters are defined through the eigenvalues of the involved matrices. Consequently, they describe the properties of the iteration (22) as $r \to \infty$. In order to have more information when a finite (possibly small) number of iterations are performed, the following *averaged factors* corresponding to μ iterations are defined [8]:

$$\rho_\mu^* = \sup_{\arg(q)=\pi/2} \rho_\mu(q), \tag{34}$$

where, given a suitable matrix norm, $\rho_\mu(q) = \sqrt[\mu]{\| (I_s - N(q)^{-1}M(q))^\mu \|}$,

$$\tilde{\rho}_\mu = \sqrt[\mu]{\| (A_1^{-1}(B_1 - B_2 + \beta A_2))^\mu \|} \tag{35}$$

and, finally,

$$\rho_\mu^{(\infty)} = \lim_{q\to\infty} \rho_\mu(q). \tag{36}$$

Again from (30), one obtains that

$$\rho_\mu^{(\infty)} = 0, \quad \text{for all } \mu = 1,2,3,\dots .$$

We observe that parameter (34) can be regarded as a particular instance of the following more general one:

$$\rho_\mu^*(\alpha) = \sup_{\arg(q)=\pi-\alpha} \rho_\mu(q), \tag{37}$$

since, evidently, $\rho_\mu^*(\pi/2) \equiv \rho_\mu^*$. The latter parameter may be useful in the case where the iteration (22) is A-convergent, even though $\rho_\mu^* > 1$, for a given finite μ. In such a case, in fact, if one obtains that $\rho_\mu^*(\pi/(2+\delta)) < 1$ for a small $\delta > 0$, this means that the iteration is "almost" A-convergent, with the given number of iterations μ.

3.2. The general case

Let us now derive the general expression of the B_3VM (16) by applying the method to problem (1). The resulting discrete problem turns out to be

$$\tilde{A}\boldsymbol{y} - h\tilde{B}\boldsymbol{f} = h\tilde{\boldsymbol{b}} f(t_0,\eta) - \tilde{\boldsymbol{a}}\eta, \tag{38}$$

where (see (19)), denoting by J the Jacobian of the function f at (t_0,η),

$$\tilde{A} = (I_s \otimes (I_m - h\beta J)^{-1})(A_1 \otimes I_m - A_2 \otimes h\beta J),$$

$$\tilde{B} = (I_s \otimes (I_m - h\beta J)^{-1})(B_1 \otimes I_m - B_2 \otimes h\beta J),$$

$$\tilde{\boldsymbol{a}} = (I_s \otimes (I_m - h\beta J)^{-1})(\boldsymbol{a}_1 \otimes I_m - \boldsymbol{a}_2 \otimes h\beta J),$$

$$\tilde{\boldsymbol{b}} = (I_s \otimes (I_m - h\beta J)^{-1})(\boldsymbol{b}_1 \otimes I_m - \boldsymbol{b}_2 \otimes h\beta J).$$

From the above expressions, it is evident that the application of the method requires the factorization of the matrix

$$I_m - h\beta J. \tag{39}$$

It is customary to solve Eq. (38) by using the modified Newton method, then solving linear systems with the matrix

$$M = \tilde{A} - h\tilde{B}(I_s \otimes J).$$

In place of such linear systems, we solve an *inner iteration* similar to (22), thus involving only linear systems with the matrix

$$N = A_1 \otimes I_m - hB_2 \otimes J.$$

The latter matrix is block Toeplitz lower bidiagonal, and its diagonal block is given by (39), which has already been factored to form the right-hand side of the linear system. We then conclude that,

leaving aside for simplicity function and Jacobian evaluations, the arithmetic complexity for solving (38) when r Newton iterations are performed, each requiring μ inner iterations, amounts to

$$\tfrac{2}{3}m^3 + \mathrm{O}(r\mu s m^2) \quad \text{floating operations.}$$

The leading term is obviously due to the factorization of matrix (39). From this fact, one concludes that the proposed methods do have a cheap implementation, at least for large-size problems.

4. Blended block GAMs (B_2GAMs)

In such a case, the B_3VM is obtained as the combination of equivalent B_2VMs (see (27)). Consequently, it has the same order and stability properties as the underlying Block GAM. Nevertheless, the discrete problem (16) generated by the B_3VM differs from those generated by the single-component methods. As a matter of fact, the former may be solved by using iteration (22), which turns out to be A-convergent, as we are going to see. Finally, we recall that, when $s = k$, minimal Block GAMs are perfectly A-stable, like the usual trapezoidal rule, up to $k = 8$. Nonetheless, they suffer the same drawback of this formula, as $q \to \infty$. In fact, from (16)–(18), (25) and (27), it follows that

$$\boldsymbol{y} \approx -B_1^{-1}\boldsymbol{b}_1\eta \quad \text{for } q \gg 0.$$

In particular, the last component of the discrete solution turns out to be given by

$$y_k \approx -\boldsymbol{e}_k^{\mathrm{T}}B_1^{-1}\boldsymbol{b}_1\eta \equiv (-1)^k\eta.$$

Consequently, there is no damping as $\mathrm{Re}(q) \to -\infty$, and it is well known that this is undesirable for stiff problems.

Conversely, by considering the blocksize $s = k+1$ (i.e., both the component methods are nonminimal), we obtain that the last entry of the solution vector, for $q \gg 0$, is

$$y_{k+1} \approx -\boldsymbol{e}_{k+1}^{\mathrm{T}}B_1^{-1}\boldsymbol{b}_1\eta \equiv g_k\eta,$$

where g_k is listed below (the values for k even are rounded to the second decimal digit).

k	1	2	3	4	5	6	7	8
g_k	1	−0.59	1	−0.74	1	−0.81	1	−0.85

Consequently, now there is some damping for k even. Moreover, also in this case, the order and stability properties of the B_2GAM coincide with those of the underlying Block GAM, namely they are perfectly A-stable for k odd [10] and A-stable for k even (in Fig. 1 the corresponding boundaries of the absolute stability regions are plotted, for completeness).

In Table 1 we report the evaluation parameters (31)–(33) for B_2GAMs, both minimal and nonminimal. In both cases, one concludes that the corresponding iteration (22) is A-convergent. This implies (see Theorem 7) that B_2GAMs, both minimal and nonminimal, are pre-stable methods.

In Table 2 we also list the evaluation parameters (34)–(37) for different values of μ (hereafter, the norm used is the ∞-norm). From this table, one concludes that only $A(\alpha)$-convergence could be practically expected for the highest-order methods, even though $\alpha > 2\pi/5$. Nevertheless, we observe that, when a high-order method is used, usually the requested error tolerances are very small. Consequently, on the imaginary axis one should "work" only for, say, $q \in (-\gamma i, \gamma i)$, with $\gamma \ll 1$. In such

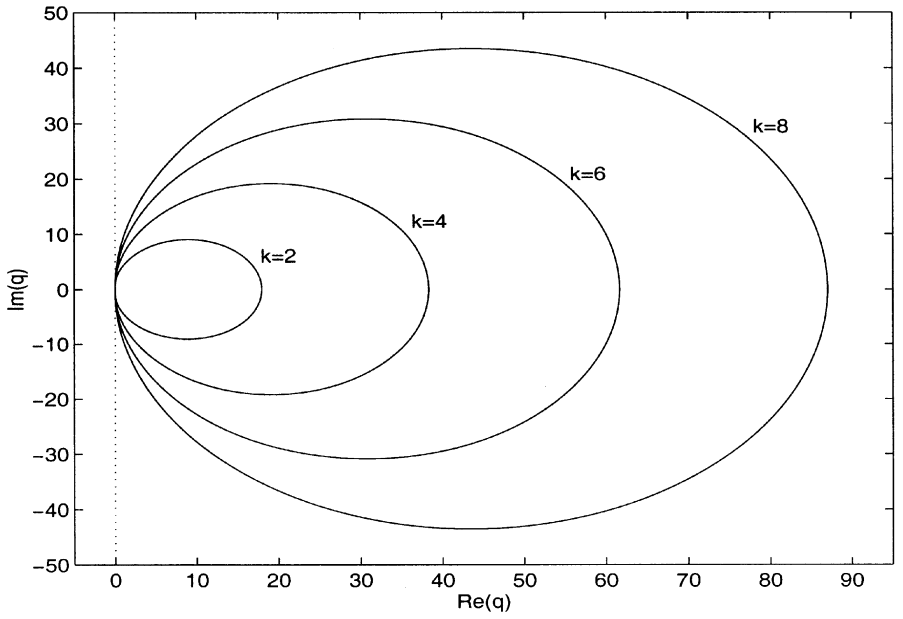

Fig. 1. Boundaries of the absolute stability regions of nonminimal Block GAMs ($s = k + 1$), $k = 2, 4, 6, 8$.

Table 1
Evaluation parameters (31)–(33) for B_2GAMs, both minimal (first column) and nonminimal (second column)

k	ρ^*		$\tilde{\rho}$		$\rho^{(\infty)}$	
1	0.25	0.25	1.50	1.50	0	0
2	0.27	0.27	1.22	1.36	0	0
3	0.40	0.40	1.19	1.29	0	0
4	0.53	0.54	1.10	1.21	0	0
5	0.66	0.66	1.09	1.26	0	0
6	0.78	0.78	1.04	1.18	0	0
7	0.88	0.88	1.04	1.26	0	0
8	0.97	0.97	1.01	1.17	0	0

interval, iteration (22) based on B_2GAMs turns out to be convergent, even though the parameter ρ_μ^* may be greater than 1.

The coefficients of Block GAMs may be found in [10]. The entries of $\hat{A}_2$ and $\hat{B}_2$ are then determined according to (27).

Remark 8. It is worth noting that B_2GAMs turn out to be A-convergent for all $s \geqslant k$. As matter of fact, in Table 3 we list the value of the parameter (31) for $k = 1, \ldots, 8$, and $s = k, \ldots, k + 10$. As one can see, in each row of the table the value of the parameter ρ^* is almost constant, after the first few values of s. From Theorem 7 one then concludes that B_2GAMs are pre-stable methods, for all allowed values of s.

Table 2
Evaluation parameters (34)–(37) for B_2GAMs, both minimal (first column) and nonminimal (second column)

k	ρ_1^*		ρ_3^*		ρ_5^*		ρ_9^*	
1	0.25	1.38	0.25	0.61	0.25	0.47	0.25	0.38
2	1.51	2.95	0.45	0.43	0.35	0.41	0.30	0.33
3	3.84	6.94	0.80	0.93	0.58	0.67	0.49	0.53
4	6.97	11.00	1.20	1.26	0.83	0.89	0.69	0.71
5	12.52	22.32	1.51	1.88	1.06	1.13	0.86	0.89
6	22.04	34.80	1.76	2.21	1.23	1.26	1.01	1.01
7	38.32	68.02	2.03	3.30	1.32	1.38	1.10	1.10
8	67.33	111.54	2.50	4.02	1.40	1.49	1.14	1.13
k	$\rho_1^*(\pi/2.5)$		$\rho_3^*(\pi/2.5)$		$\rho_5^*(\pi/2.5)$		$\rho_9^*(\pi/2.5)$	
1	0.19	1.05	0.19	0.47	0.19	0.36	0.19	0.29
2	1.15	2.25	0.34	0.31	0.26	0.30	0.22	0.24
3	2.88	5.37	0.59	0.67	0.42	0.49	0.36	0.38
4	5.33	8.38	0.86	0.86	0.59	0.63	0.49	0.50
5	9.60	17.01	1.07	1.34	0.73	0.78	0.61	0.62
6	16.83	26.68	1.24	1.59	0.84	0.86	0.70	0.69
7	29.34	52.13	1.41	2.45	0.90	0.94	0.75	0.74
8	51.59	85.31	1.69	2.98	0.97	1.03	0.77	0.76
k	$\tilde{\rho}_1$		$\tilde{\rho}_3$		$\tilde{\rho}_5$		$\tilde{\rho}_9$	
1	1.50	3.50	1.50	2.56	1.50	2.25	1.50	1.99
2	4.25	6.85	2.27	3.06	1.71	2.29	1.43	1.78
3	7.71	15.01	3.11	4.73	2.13	3.18	1.53	2.24
4	14.39	20.67	4.06	5.45	2.60	3.42	1.75	2.25
5	24.45	42.05	5.26	7.56	3.14	4.36	1.99	2.71
6	42.35	67.84	6.67	8.93	3.74	4.82	2.24	2.83
7	74.48	130.01	8.46	11.89	4.42	5.92	2.51	3.28
8	127.49	215.35	10.65	14.22	5.20	6.62	2.81	3.48
k	$\rho_1^{(\infty)}$		$\rho_3^{(\infty)}$		$\rho_5^{(\infty)}$		$\rho_9^{(\infty)}$	
1–8	0	0	0	0	0	0	0	0

Table 3
Values of ρ^* for B_2GAMs, different values of s

$k \setminus s-k$	0	1	2	3	4	5	6	7	8	9	10
1	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
2	0.27	0.27	0.28	0.29	0.30	0.30	0.31	0.31	0.31	0.32	0.32
3	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
4	0.53	0.54	0.54	0.55	0.55	0.55	0.56	0.56	0.56	0.56	0.56
5	0.66	0.66	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67
6	0.78	0.78	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79
7	0.88	0.88	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89
8	0.97	0.97	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98

5. Hybrid-blended block GAMs (HB$_2$GAMs)

We now study the properties of minimal HB$_2$GAMs. First of all, let us consider the truncation error of the methods, by considering again the test equation (14), for simplicity. By setting $\hat{\boldsymbol{y}}$ the vector with the values of the continuous solution at the grid points, for the first component method (the minimal Block GAM based on k-step formulae) one has

$$(\hat{A}_1 - q\hat{B}_1)\hat{\boldsymbol{y}} = \mathrm{O}(h^{k+2}), \quad (q = h\lambda),$$

whereas for the second component method one has (see (28)),

$$(\hat{A}_2 - q\hat{B}_2)\hat{\boldsymbol{y}} = \mathrm{O}(h^{k+1}).$$

However, when considering the overall method (see (16)–(20) and (25)), one obtains

$$(\hat{A}(q) - q\hat{B}(q))\hat{\boldsymbol{y}} \equiv \frac{1}{1-\beta q}(\hat{A}_1 - q\hat{B}_1 - \beta q(\hat{A}_2 - q\hat{B}_2))\hat{\boldsymbol{y}} = \mathrm{O}(h^{k+2}). \tag{40}$$

That is, the corresponding minimal HB$_2$GAM has still a $\mathrm{O}(h^{k+2})$ truncation error, and it can be seen that the order of convergence is $k+1$. Alternatively, one may define nonminimal HB$_2$GAMs by using the Block GAM of order $k+1$ and blocksize $s = k+1$ as the first component method. In such a case, the entries of the matrices $\hat{A}_2$ and $\hat{B}_2$ (see (28)) can be uniquely determined in order to have

$$(\hat{A}_2 - q\hat{B}_2)\hat{\boldsymbol{y}} = \mathrm{O}(h^{k+2})$$

as well (namely, by using $(k+1)$-step LMF in each equation defining the second component method). As a consequence, one obtains that (see (40)) the principal term of the truncation error of the corresponding nonminimal HB$_2$GAM coincides with that of the Block GAM, which can be efficiently approximated via deferred correction [5,11].

Another interesting property of HB$_2$GAMs (both minimal and nonminimal) is the qualitative behavior of the discrete solutions as $q \to \infty$, in which case one has that

$$\boldsymbol{y} \to -B_2^{-1}\boldsymbol{b}_2\eta.$$

From (28), one then obtains that for $q \gg 0$ the last entry of the solution vector is given by

$$y_s \approx \gamma^s\eta, \quad \gamma = -\frac{1-\beta}{\beta}.$$

Since $\frac{1}{2} < \beta \leqslant 1$, it follows that there is a considerable potential damping, as s grows. In more detail, we have chosen β according to the following values, where we also list the corresponding damping factors γ^s: the first rows are relative to minimal HB$_2$GAMs ($s = k$), whereas the second rows refer to the nonminimal methods ($s = k+1$).

k	1	2	3	4	5	6	7	8
β	0.53	0.53	0.58	0.63	0.68	0.75	0.81	0.86
	0.53	0.57	0.60	0.66	0.71	0.80	0.85	0.88
γ^s	−8.9e − 1	7.9e − 1	−3.8e − 1	1.2e − 1	−2.3e − 2	1.4e − 3	−3.9e − 5	4.9e − 7
	7.9e − 1	−4.3e − 1	2.0e − 1	−3.6e − 2	4.6e − 3	−6.1e − 5	9.4e − 7	−1.6e − 8

Table 4
Evaluation parameters (31)–(33) for minimal (first column) and nonminimal (second column) HB_2GAMs

k	ρ^*		$\tilde{\rho}$		$\rho^{(\infty)}$	
1	0.03	0.06	0.50	0.50	0	0
2	0.07	0.11	0.50	0.58	0	0
3	0.20	0.29	0.50	0.50	0	0
4	0.36	0.46	0.50	0.55	0	0
5	0.54	0.65	0.50	0.50	0	0
6	0.72	0.80	0.50	0.53	0	0
7	0.86	0.93	0.50	0.51	0	0
8	0.98	1.02	0.51	0.52	0	0

It is evident that the damping factors of HB_2GAMs compare almost always favorably with those of B_2GAMs, in particular for the highest-order methods.

Concerning the choice of the parameter β, it has been done in order to have satisfactory convergence properties for the corresponding iteration (22). Indeed, for all $k = 1, \ldots, 8$, it turns out that minimal HB_2GAMs are A-convergent, as one may infer from Table 4, where we list the corresponding evaluation parameters (31)–(33). From Theorem 7 one then concludes that minimal HB_2GAMs are pre-stable methods. Nonminimal HB_2GAMs are A-convergent up to $k = 7$: for $k = 8$, the method can be seen to be at least $A(89^\circ)$-convergent. In Table 5 we also report the corresponding parameters (34)–(37) for both minimal and nonminimal methods.

For completeness, in Figs. 2 and 4 we plot the boundaries of the absolute stability regions of minimal and nonminimal HB_2GAMs with k odd, respectively. In Figs. 3 and 5 there are the corresponding plots for k even. From the two figures one may infer that all methods are $A(\alpha)$-stable. The corresponding angles α are listed (rounded to the second decimal digit) in the following table: the first row is relative to minimal HB_2GAMs, while the second row is relative to the nonminimal methods.

k	1	2	3	4	5	6	7	8
α	90°	90°	90°	89.88°	89.46°	89.16°	88.80°	88.08°
	90°	90°	89.87°	89.78°	89.39°	89.33°	88.66°	87.42°

6. Numerical tests

We here report some numerical tests on well-known problems taken from the literature. We first consider a modified version of the sequential code *GAM* in [11]. The latter code is based on Block GAM (BGAM, hereafter) formulae, whereas the modified code is based on the corresponding B_2GAMs introduced in Section 4. The discrete problems generated by the methods are solved by a splitting-Newton iteration in the code *GAM* (see [11] for details), while an iteration similar to (22) is used in the modified code. By comparing the two iterations, it turns out that the iteration used for B_2GAMs is more robust and faster (in terms of iterations needed for convergence) than

Table 5
Evaluation parameters (34)–(37) for minimal (first column) and nonminimal (second column) HB_2GAMs

k	ρ_1^*		ρ_3^*		ρ_5^*		ρ_9^*	
1	0.03	0.07	0.03	0.07	0.03	0.07	0.03	0.06
2	0.32	0.37	0.11	0.17	0.09	0.14	0.08	0.13
3	0.66	1.78	0.32	0.56	0.26	0.42	0.23	0.35
4	1.73	3.33	0.64	0.94	0.51	0.70	0.44	0.58
5	3.20	6.08	1.05	1.36	0.80	1.02	0.67	0.83
6	6.30	14.14	1.34	1.51	1.05	1.18	0.88	0.99
7	13.61	26.73	1.47	1.85	1.19	1.26	1.03	1.11
8	25.83	50.26	1.78	2.36	1.24	1.35	1.13	1.18
k	$\rho_1^*(\pi/2.5)$		$\rho_3^*(\pi/2.5)$		$\rho_5^*(\pi/2.5)$		$\rho_9^*(\pi/2.5)$	
1	0.02	0.06	0.02	0.05	0.02	0.05	0.02	0.05
2	0.23	0.25	0.07	0.11	0.06	0.09	0.05	0.08
3	0.46	1.04	0.20	0.34	0.16	0.26	0.14	0.22
4	0.99	2.24	0.39	0.56	0.31	0.43	0.27	0.36
5	2.12	4.72	0.63	0.80	0.49	0.61	0.41	0.51
6	4.86	10.62	0.79	0.97	0.63	0.73	0.55	0.63
7	10.18	20.51	0.93	1.30	0.73	0.83	0.65	0.74
8	19.83	38.45	1.26	1.60	0.85	0.93	0.76	0.83
k	$\tilde{\rho}_1$		$\tilde{\rho}_3$		$\tilde{\rho}_5$		$\tilde{\rho}_9$	
1	0.50	0.60	0.50	0.58	0.50	0.57	0.50	0.56
2	0.70	1.14	0.65	0.89	0.62	0.80	0.61	0.72
3	1.46	1.69	0.96	1.01	0.82	0.84	0.71	0.71
4	1.65	6.26	1.00	1.79	0.85	1.25	0.72	0.93
5	6.22	6.40	1.76	2.22	1.23	1.52	0.91	1.09
6	6.79	24.31	2.40	3.79	1.62	2.25	1.15	1.44
7	24.83	40.01	3.76	5.40	2.23	2.94	1.44	1.75
8	38.02	89.23	5.40	7.40	2.95	3.72	1.76	2.08
k	$\rho_1^{(\infty)}$		$\rho_3^{(\infty)}$		$\rho_5^{(\infty)}$		$\rho_9^{(\infty)}$	
1–8	0	0	0	0	0	0	0	0

that used for BGAMs (this can be also deduced by comparing the amplification factors in Tables 1–3 with the corresponding amplification factors in [11]). As a matter of fact, the modified code generally requires less function and Jacobian evaluations, than the original code. Nevertheless, each iteration for B_2GAMs has an $O(m^2)$ complexity section (we recall that m denotes the dimension of the continuous problem) which is approximately three times more expensive than that of the iteration used for the corresponding BGAM. Moreover, the variable order strategy of the code *GAM* is very well tuned for the BGAMs used. As a result, it turns out that the modified code, used with variable order, is generally 30%–70% slower than the original code *GAM*. Things partially change when the two codes are used with a fixed order, in particular the highest one, that is 9 ($k = 8$). In such a case, in fact, one has that the iteration of the 9th-order B_2GAM is much more robust than that used in the original code. As a matter of fact, the former iteration is A-convergent (see Section 4), whereas the latter does not [11]. In Tables 6 and 7 we report the obtained results, for the original

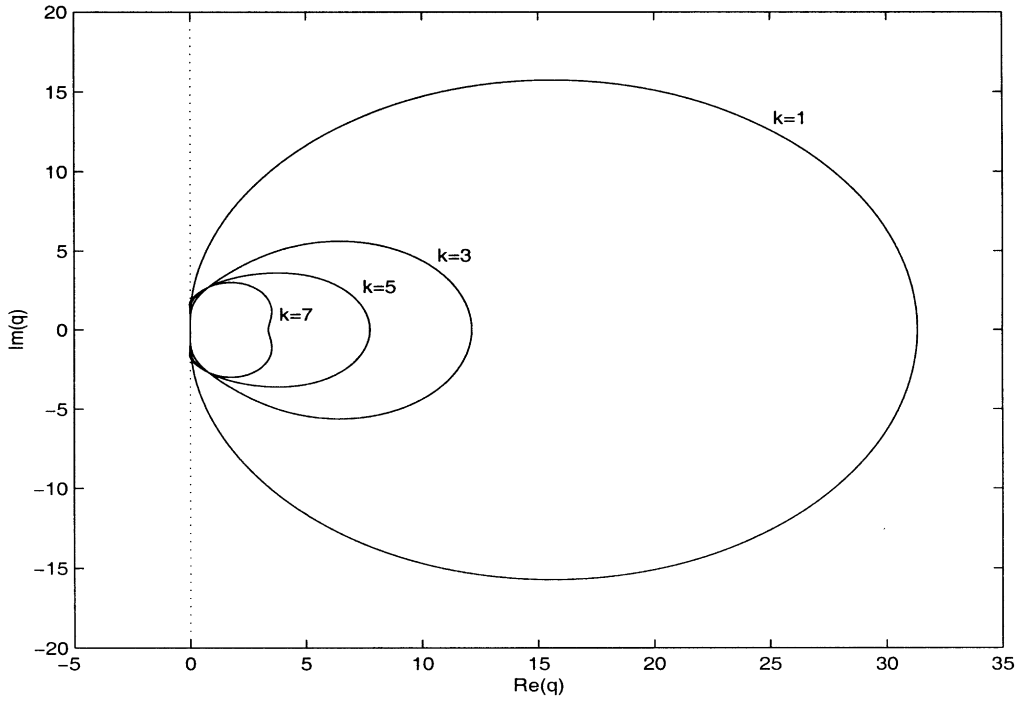

Fig. 2. Boundaries of the absolute stability regions of minimal HB_2GAMs, $k = 1, 3, 5, 7$.

Table 6
van der Pol problem

	Tol	1e − 02	1e − 03	1e − 04	1e − 05	1e − 06	1e − 07	1e − 08	1e − 09
	Error	1.4E − 03	5.5E − 04	3.3E − 05	4.7E − 06	5.1E − 07	5.4E − 08	3.4E − 09	1.9E − 10
B	# steps	5457	4411	537	396	382	117	96	113
G	# accept	4223	4329	521	384	371	99	79	92
A	# f-eval	260963	259850	32317	23869	22878	8632	7645	10287
M	# Jac-eval	2940	4253	519	386	373	97	73	77
9	# LU-decomp	4233	4338	532	396	381	113	95	108
	time (sec/100)	360	371	46	34	32	12	11	14
	Error	5.7E − 04	3.2E − 04	8.6E − 06	3.4E − 06	5.5E − 07	3.8E − 08	2.0E − 08	1.3E − 11
B	# steps	773	287	151	86	97	108	94	107
2	# accept	521	207	110	62	69	75	72	84
G	# f-eval	15399	8245	5421	3996	4732	6727	6112	6961
A	# Jac-eval	265	98	65	36	44	55	59	57
M	# LU-decomp	525	216	127	84	96	102	93	103
9	time (sec/100)	45	23	16	11	13	18	17	19

fixed-order code (BGAM9) and the fixed-order modified one (B2GAM9), on the van der Pol and Robertson problems, respectively. In all cases, the parameters $atol = rtol = h0 = \text{tol}$ have been used. Asterisks in the "error" field mean that the (fixed) order method fails. As one can see, in such a case the modified code generally performs better.

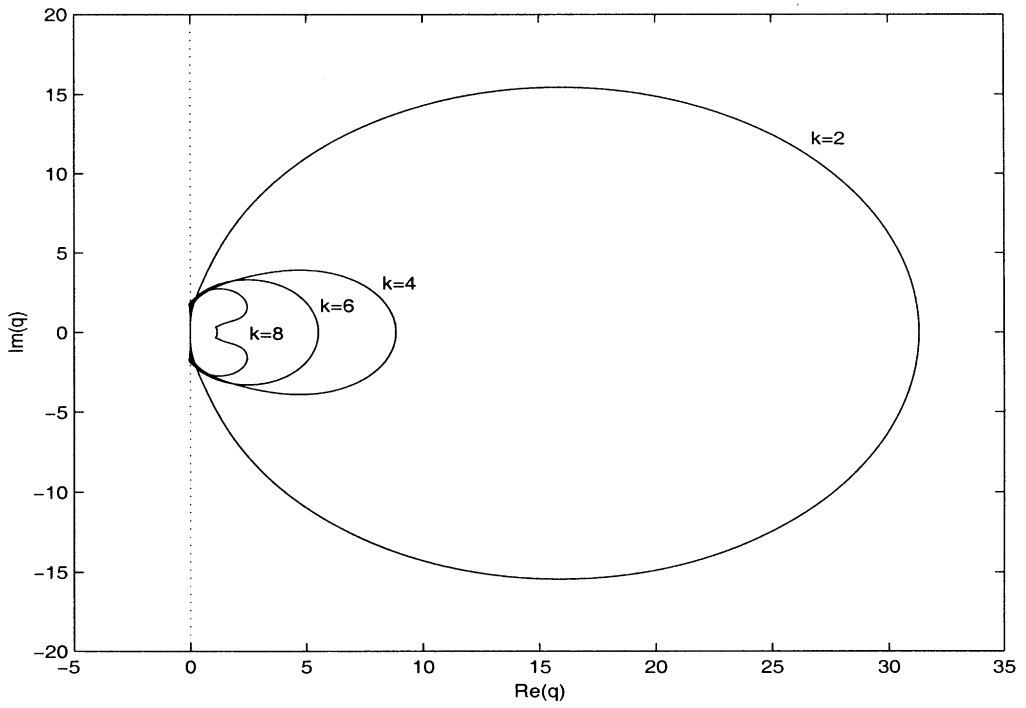

Fig. 3. Boundaries of the absolute stability regions of minimal HB_2GAMs, $k = 2, 4, 6, 8$.

Table 7
Robertson problem

	Tol	1e − 04	1e − 05	1e − 06	1e − 07	1e − 08	1e − 09	1e − 10	1e − 11
	Error	****	1.2E − 5	2.2E − 10	8.3E − 11	1.0E − 11	3.5E − 13	1.6E − 13	1.8E − 14
B	# steps	2611	22509	223	223	119	107	100	94
G	# accept	1546	12988	145	148	88	87	85	86
A	# f-eval	72506	514143	9775	10551	6896	6964	6754	6855
M	# Jac-eval	1032	9525	94	118	79	80	76	76
9	# LU-decomp	1577	12992	165	193	115	106	99	94
	time (sec/100)	125	843	16	18	12	12	11	12
	Error	1.6E − 08	2.7E − 09	6.3E − 10	4.4E − 11	4.3E − 15	3.4E − 14	7.9E − 15	6.3E − 15
B	# steps	1207	668	86	62	52	73	118	82
2	# accept	746	401	63	51	49	64	96	81
G	# f-eval	21969	15495	2998	2671	2759	3611	5038	4105
A	# Jac-eval	472	277	44	35	35	45	62	45
M	# LU-decomp	758	410	66	53	49	67	96	81
9	time (sec/100)	78	55	10	9	10	12	17	14

Finally, we consider the parallel implementation *across the steps* of nonminimal HB_2GAMs by using a two-step procedure as described in [1,6]. The second step of such procedure, in fact, in its original formulation does suffer for a parallel complexity of $O((sm)^3)$ flops (see the concluding

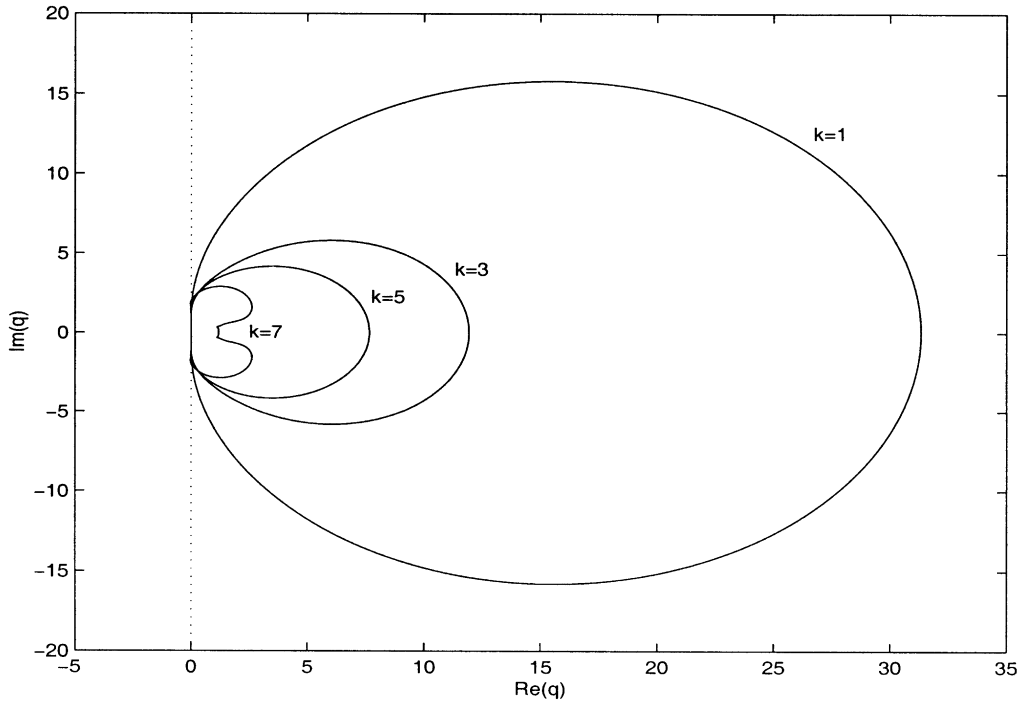

Fig. 4. Boundaries of the absolute stability regions of nonminimal HB_2GAMs, $k = 1, 3, 5, 7$.

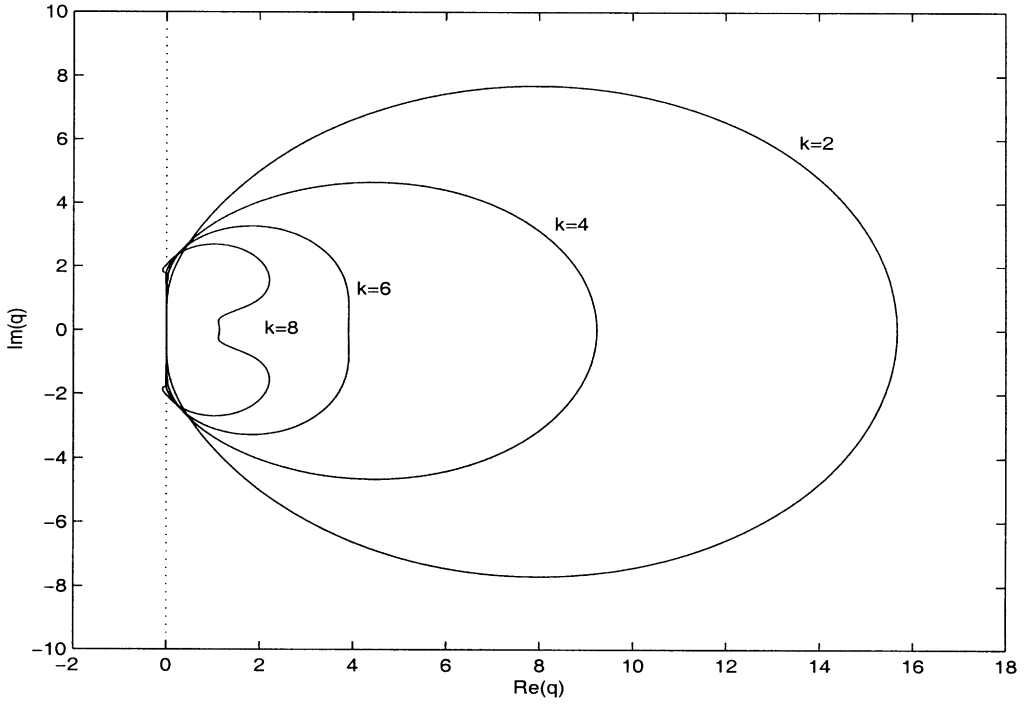

Fig. 5. Boundaries of the absolute stability regions of nonminimal HB_2GAMs, $k = 2, 4, 6, 8$.

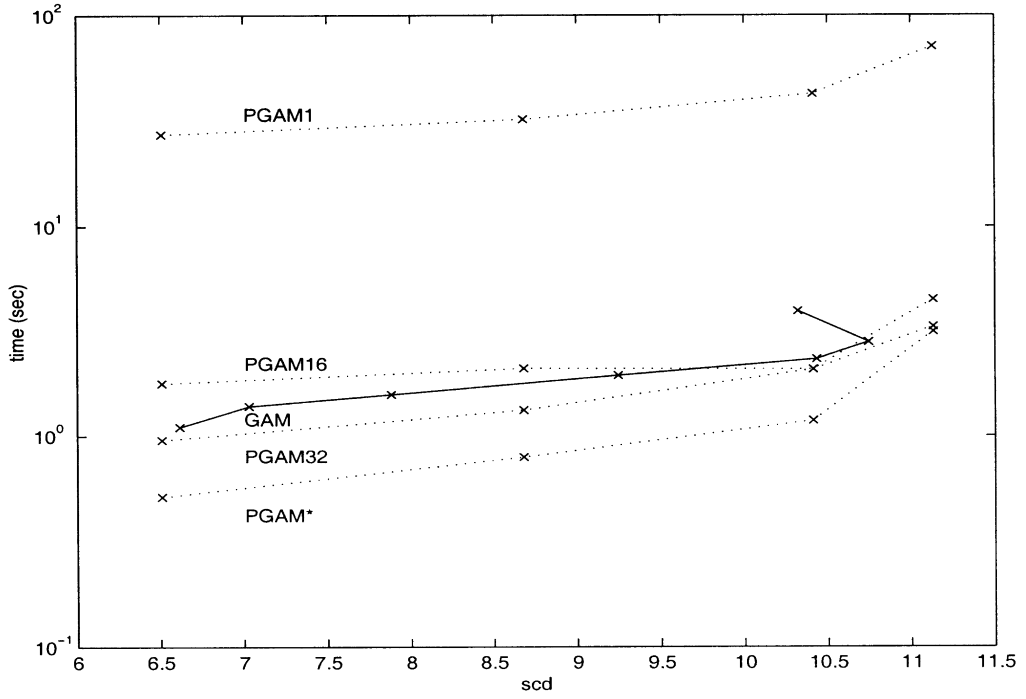

Fig. 6. Work precision diagram, ring modulator problem.

remarks in [1]). By using the inner iteration corresponding to HB_2GAMs, such complexity is lowered to $O(m^3)$ flops, thus overcoming the above mentioned problem. In Fig. 6, there is the work-precision diagram for the "ring modulator" problem, a stiff IVP of dimension 15 from the CWI test-set [14], obtained on a Cray T3E parallel computer, where scd is as usual the number of significant computed digits in the discrete solution. Moreover, in the diagram the continuous line labelled "GAM" denotes the obtained results for the code *GAM* used with decreasing tolerances. The dotted line labelled "PGAMp" denotes the obtained results for the execution, on p processors, of the parallel code based on the 9th-order nonminimal HB_2GAM ($k = 8, s = 9$). The coefficients of the matrix $\hat{A}_2$ of this method are listed in Table 8 ($\alpha_j^{(i)}$ is the $(i, j+1)$th entry of the matrix, $i = 1, \ldots, 9$, $j = 0, \ldots, 9$); the corresponding matrix $\hat{B}_2$ (see (28)) is obtained for $\beta = 0.88$, as it has been already said in Section 5. Finally, the dotted line labelled "PGAM*" corresponds to the best asymptotic parallel performance for the parallel solver (full details will be given in [2]). From the figure, one has that the parallel code, when executed on 32 processors, is always faster than the sequential code GAM and, asymptotically, it is about 2.5 times faster (for such problem). This result could be further improved, by considering that the first step of the parallel procedure, based on a sequential, low-order method which determines the mesh [6], needs to be improved as well. In fact, by looking at Fig. 7, it turns out that at least twice the number of mesh points needed by the code GAM is required by the current version of the parallel code to obtain a comparable accuracy. This problem will be the further investigated. Nevertheless, the results obtained so far seem to confirm that the second step of the procedure, based on nonminimal HB_2GAMs, is quite efficient and reliable.

Table 8
Coefficients of the nonminimal HB_2GAM, $k = 8$, rows of the matrix $\hat{A_2}$ multiplied by $\delta = 63000$

i	$\alpha_0^{(i)}\delta$	$\alpha_1^{(i)}\delta$	$\alpha_2^{(i)}\delta$	$\alpha_3^{(i)}\delta$	$\alpha_4^{(i)}\delta$	$\alpha_5^{(i)}\delta$	$\alpha_6^{(i)}\delta$	$\alpha_7^{(i)}\delta$	$\alpha_8^{(i)}\delta$	$\alpha_9^{(i)}\delta$
1	−27547	−27198	85680	−47040	20580	−3528	−2352	1920	−585	70
2	−70	−26847	−30348	94080	−61740	38220	−18228	6048	−1230	115
3	−115	1080	−32022	−16548	69930	−32760	14070	−4428	873	−80
4	80	−915	4680	−41622	252	49770	−15960	4470	−828	73
5	−73	810	−4200	13440	−56952	18648	34440	−7200	1185	−98
6	98	−1053	5220	−15960	34020	−81648	39228	22680	−2790	205
7	−205	2148	−10278	29820	−59010	85680	−124698	63828	13455	−740
8	740	−7605	35448	−99078	185220	−245490	241080	−213498	97128	6055
9	−6055	61290	−280080	762048	−1370628	1711080	−1517040	967680	−485973	157678

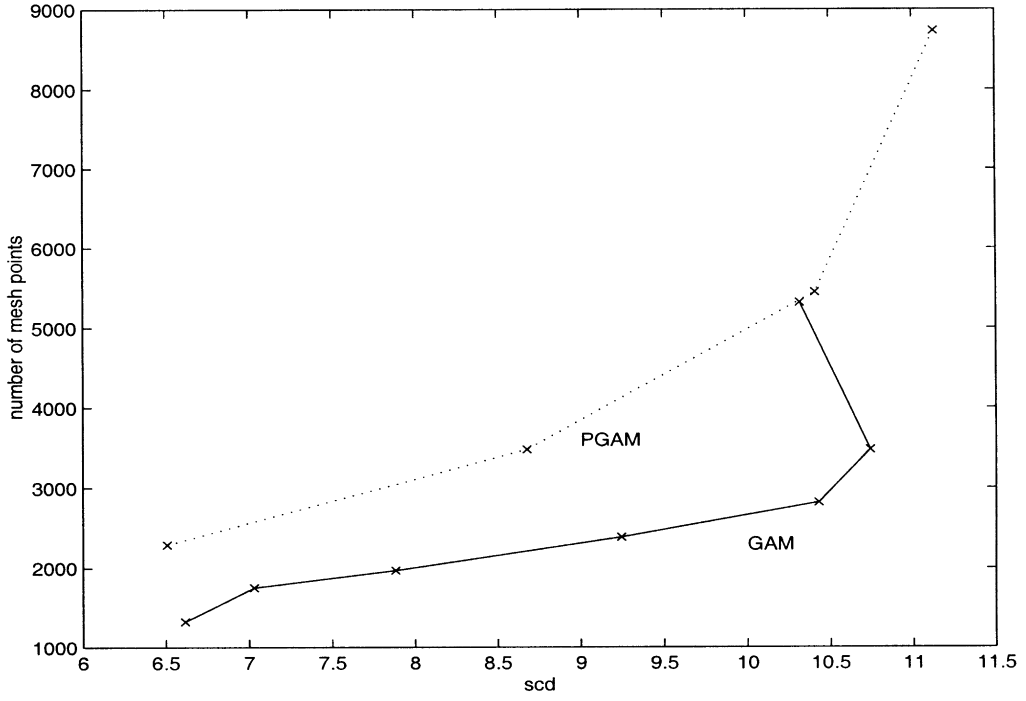

Fig. 7. Number of mesh point versus accuracy, ring modulator problem.

Acknowledgements

The author wishes to thank Professor Donato Trigiante for the helpful discussions and the referees for their comments.

References

[1] P. Amodio, L. Brugnano, ParalleloGAM: a parallel code for ODEs, Appl. Numer. Math. 28 (1998) 127–141.
[2] P. Amodio, L. Brugnano, in progress.

[3] L. Brugnano, D. Trigiante, Block boundary value methods for linear Hamiltonian systems, Appl. Math. Comput. 81 (1997) 49–68.
[4] L. Brugnano, D. Trigiante, On the potentiality of sequential and parallel codes based on extended trapezoidal rules (ETRs), Appl. Numer. Math. 25 (1997) 169–184.
[5] L. Brugnano, D. Trigiante, Solving Differential Problems by Multistep Initial and Boundary Value Methods, Gordon and Breach, Amsterdam, 1998.
[6] L. Brugnano, D. Trigiante, Parallel implementation of block boundary value methods on nonlinear problems: theoric results, Appl. Numer. Math. 28 (1998) 127–141.
[7] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems, Springer, Berlin, 1991.
[8] 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.
[9] P.J. van der Houwen, J.J.B. de Swart, Parallel linear system solvers for Runge–Kutta methods, Adv. Comput. Math. 7 (1-2) (1997) 157–181.
[10] F. Iavernaro, F. Mazzia, Block-boundary value methods for the solution of ordinary differential equations, SIAM J. Sci. Comput., to appear.
[11] F. Iavernaro, F. Mazzia, Solving ordinary differential equations by generalized Adams methods: Properties and implementation techniques, Appl. Numer. Math. 28 (1998) 107–126. See also the URL http://www.dm.uniba.it/mazzia/ode/readme.html
[12] R.D. Skeel, A.K. Kong, Blended linear multistep methods, ACM TOMS 3 (1977) 326–345.
[13] H.A. Watts, L.F. Shampine, *A*-stable block one-step methods, BIT 12 (1972) 252–266.
[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.