Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE DIRECT®

ELSEVIER Journal of Computational and Applied Mathematics 164–165 (2004) 145–158

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

The BiM code for the numerical solution of ODEs☆

Luigi Brugnano*, Cecilia Magherini

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

Received 22 July 2002; received in revised form 4 June 2003

Dedicated to Prof. J.C. Butcher on the occasion of his 70th birthday

Abstract

In this paper we present the code BiM, based on *blended implicit methods* (J. Comput. Appl. Math. 116 (2000) 41; Appl. Numer. Math. 42 (2002) 29; Recent Trends in Numerical Analysis, Nova Science Publ. Inc., New York, 2001, pp. 81.), for the numerical solution of stiff initial value problems for ODEs. We describe in detail most of the implementation strategies used in the construction of the code, and report numerical tests comparing the code BiM with some of the best codes currently available. The numerical tests show that the new code compares well with existing ones. Moreover, the methods implemented in the code are characterized by a diagonal nonlinear splitting, which makes its extension for parallel computers very straightforward.

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

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

1. Introduction

After about 50 years, the numerical solution of initial value problems for ODEs is still an active field of investigation, even though the required properties for the numerical methods underwent a significant qualitative evolution across the years. Indeed, until the 1950's accuracy requirements were considered as the most important for the methods. After that, stability requirements became focal, in particular in connection with the numerical solution of stiff problems. More recently, attention has been devoted to methods well suited for particular differential problems (e.g. DAEs, Hamiltonian problems, and so forth), and to methods well suited for an efficient implementation on computers

☆ Work supported by Italian CNR and GNCS-INdAM.
* Corresponding author. Fax: +39-055-422-2605.
E-mail addresses: brugnano@math.unifi.it (L. Brugnano), magherini@math.unifi.it (C. Magherini).

0377-0427/$ - see front matter
doi:10.1016/j.cam.2003.09.004

as well as to the definition of efficient implementation techniques for the existing methods. As a consequence, a number of reliable numerical codes have been developed for solving stiff problems: as an example, the codes DASSL [1], GAM [16], MEBDFDAE [9], RADAU5 and RADAU [11].

In this paper we describe a new code, called BiM, which is based on *blended implicit methods* [2,3,7]. The latter are methods defined in order to favorably meet implementation requirements. The code compares well with the above mentioned codes; moreover, the diagonal nonlinear splitting on which it is based naturally induces its extension for parallel computers, which will be the subject of future researches.

The organization of the paper is as follows: in Section 2 we recall the main facts about the *blended implicit methods* implemented in the code BiM; in Section 3 we describe in detail the implementation of the nonlinear iteration; Section 4 is devoted to the order and stepsize variation strategies; in Section 5 we report some numerical tests comparing the code BiM with the above mentioned ones and, finally, some concluding remarks. Concerning the problem of the eventual re-evaluation of the Jacobian and/or the factorization involved in the nonlinear splitting, we refer to [4].

2. Blended implicit methods

The numerical solution of the ODE problem

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

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.

Let us assume, for the moment, that a uniform partition with stepsize h is used, $t_n = t_0 + nh$, where $n = 0, \dots, N$ and $Nh = T - t_0$.

We will be concerned with the discrete problem generated by a *block implicit method*, namely

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

where A and B are $r \times r$ nonsingular matrices defining the method, the block vectors $\mathbf{y}_n = (y_{n+1}, \dots, y_{n+r})^{\mathrm{T}}$ and $\mathbf{f}_n = (f_{n+1}, \dots, f_{n+r})^{\mathrm{T}}$, where $f_j = f(t_j, y_j)$, contain the discrete solution, and the vector $\boldsymbol{\eta}_n$ only depends on already known quantities. Instances of methods falling in this class are the majority of implicit Runge–Kutta methods, a number of General Linear methods [8,10,11] and, more recently, block BVMs [6]. In particular, the block methods we shall deal with are such that

$$[\mathbf{a} \,|\, A] \equiv \left(\begin{array}{c|ccc} \alpha_0^{(1)} & \alpha_1^{(1)} & \dots & \alpha_r^{(1)} \\ \vdots & \vdots & & \vdots \\ \alpha_0^{(r)} & \alpha_1^{(r)} & \dots & \alpha_r^{(r)} \end{array}\right), \quad [\mathbf{b} \,|\, B] \equiv \left(\begin{array}{c|ccc} \beta_0^{(1)} & \beta_1^{(1)} & \dots & \beta_r^{(1)} \\ \vdots & \vdots & & \vdots \\ \beta_0^{(r)} & \beta_1^{(r)} & \dots & \beta_r^{(r)} \end{array}\right), \tag{3}$$

where the coefficients on the ith row of the two matrices define a suitable r-step LMF, and $\boldsymbol{\eta}_n = -\mathbf{a} \otimes y_n + h\mathbf{b} \otimes f_n$. Our goal is now that of deriving an efficient nonlinear splitting for solving (2).

Hereafter, I and O will denote the identity and the zero matrix of size r. For simplicity, let us consider the first application of the method, in order to omit the index n, and assume that the given problem is the test equation,

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

for which the discrete problem assumes the simpler form

$$(A - qB)\mathbf{y} = \boldsymbol{\eta}, \quad q = h\mu. \tag{5}$$

By setting $C = A^{-1}B$, such an equation is equivalent to the following:

$$(I - qC)\mathbf{y} = \boldsymbol{\eta}_1 \equiv A^{-1}\boldsymbol{\eta}, \quad (C^{-1} - qI)\mathbf{y} = \boldsymbol{\eta}_2 \equiv B^{-1}\boldsymbol{\eta}. \tag{6}$$

By introducing the weighting function $\theta(q) = (I - q\gamma I)^{-1}$, where $\gamma > 0$, one verifies that $\theta(0) = I$ and $\theta(q) \to O$, as $q \to \infty$. As a consequence, the following equation:

$$\begin{aligned} M(q)\mathbf{y} - \boldsymbol{\eta}(q) &\equiv (A(q) - qB(q))\mathbf{y} - \boldsymbol{\eta}(q) \\ &\equiv ((\theta(q)I + (I - \theta(q))\gamma C^{-1}) - q(\theta(q)C + (I - \theta(q))\gamma I))\mathbf{y} \\ &\quad -(\theta(q)\boldsymbol{\eta}_1 + (I - \theta(q))\gamma\boldsymbol{\eta}_2) = \mathbf{0}, \end{aligned} \tag{7}$$

has the same solution as (5). The previous equation defines a *blended implicit method* associated with the block method (2), due to the fact that the discrete problem is obtained as the "blending" of two equivalent forms of the same basic block method.

The key point concerning a blended implicit method is that its structure naturally induces the choice of a splitting for iteratively solving (7). In fact, one easily verifies that $M(q) = I + \mathrm{O}(q) \approx I$, when $q \approx 0$, and $M(q) = -\gamma q(I + \mathrm{O}(q^{-1})) \approx -q\gamma I$, when $q \to \infty$. Consequently, instead of solving (7), one may think to solve iteratively

$$N(q)\mathbf{y}^{(i+1)} = (N(q) - M(q))\mathbf{y}^{(i)} + \boldsymbol{\eta}(q), \quad i = 0, 1, \ldots, \tag{8}$$

where

$$N(q) = I - q\gamma I \equiv \theta(q)^{-1}. \tag{9}$$

This is because $N(0) = M(0)$, and $N(q) \approx M(q)$, for $|q| \gg 1$. We shall call (8) the *blended iteration* associated with the blended method (7). Obviously, such an iteration will converge if and only if the spectral radius of the iteration matrix

$$I - N(q)^{-1}M(q), \tag{10}$$

say $\rho(q)$, is smaller than 1. Following [12,13], the iteration is said to be *A-convergent* if $\rho(q) < 1$ for all $q \in \mathbb{C}^-$. Since $\gamma > 0$, A-convergence is equivalent to require that the *maximum amplification factor*, $\rho^* = \max_{x>0} \rho(\mathrm{i}x)$, with i denoting the imaginary unit, is smaller than 1. We observe that, from (7)–(9), one obtains that $\rho(0) = 0$ and $\rho^{(\infty)} \equiv \lim_{q\to\infty} \rho(q) = 0$ since, in both cases, the iteration

matrix is the zero matrix. Consequently, because of the second property, iteration (8) is well-suited for stiff problems, since the *stiff amplification factor* $\rho^{(\infty)}$ is 0 [12,13]. In such a case, an A-convergent iteration is said to be *L-convergent*. Moreover, if the iteration matrix is well-defined in a neighborhood of $q=0$, a first-order expansion shows that

$$\rho(q) \approx \tilde{\rho} q, \quad \text{for} \quad q \approx 0, \tag{11}$$

where $\tilde{\rho}$ is the *nonstiff amplification factor*. In [3] the parameter γ has been chosen in order to minimize the maximum amplification factor ρ^*, thus giving L-convergent iterations for all methods implemented in the code. For such methods, the matrix C is uniquely defined by imposing that all the LMFs associated with the two matrices in (3) have an $O(h^{r+1})$ truncation error, and by fixing its characteristic polynomial, say $d(z)$, as the reciprocal, and scaled, polynomial at the denominator of the (v,r) Padé approximation to the exponential,

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

In particular, the following values of r and v have been considered: $v=r-1$ for $r=3$ and $v=r-2$ for $r \geqslant 4$, thus resulting in L-stable methods, whose order (hereafter denoted by p) is the one specified in the Table 1 (see, for example, [18]). For such methods, for which the results of Section 4 in [3] apply, the following result holds true.

Theorem 1. *Let $\sigma(C)$ be the spectrum of the matrix C, then the spectral radius of iteration matrix (10) is given by*

$$\rho(q) = \left| \frac{q(\lambda_1 - \gamma)^2}{\lambda_1(1-q\gamma)^2} \right| \quad \textit{where} \quad |\lambda_1| = \min_{\lambda \in \sigma(C)} |\lambda|. \tag{12}$$

Moreover, again from the arguments in [3] and by denoting with ζ_1 the argument of λ_1, the following results easily follow:

$$\gamma = |\lambda_1|, \quad \rho^* = 1 - \cos\zeta_1, \quad \tilde{\rho} = 2|\lambda_1|\rho^*.$$

In Table 1 we list such parameters for the considered values of r. The reason for considering methods of different order (and, obviously, of different computational cost per step) is due to the fact that the code `BiM` implements a variable order-variable stepsize strategy for the methods. From Table 1 one obtains that all iterations are A-convergent ones and, then, also L-convergent, since $\rho^{(\infty)}=0$, as previously stated.

Table 1
Values of various parameters for the methods implemented in the code `BiM`

r	Padé	p	γ	ρ^*	$\tilde{\rho}$	$\tilde{\rho}_r^{(\infty)}$	`maxit`	`faterr`
3	(2,3)	4	0.7387	0.3398	0.5021	0.9201	10	7
4	(2,4)	6	0.8482	0.5291	0.8975	1.2476	12	6
6	(4,6)	8	0.7285	0.6299	0.9177	1.7295	14	5
8	(6,8)	10	0.6745	0.6885	0.9288	2.0413	16	4
10	(8,10)	12	0.6433	0.7276	0.9361	2.2621	18	3
12	(10,12)	14	0.6227	0.7560	0.9415	2.4282	20	—

3. The nonlinear iteration

We now analyze in detail the nonlinear iteration generated by a blended implicit method applied to problem (1). In fact, in such a case, the blended iteration (8) becomes

$$\begin{aligned}\Delta\mathbf{y}^{(i)} = &-\theta(\theta((I-\gamma C^{-1})\otimes I_m\mathbf{y}^{(i)} - h(C-\gamma I)\otimes I_m\mathbf{f}^{(i)})\\ &+\gamma(C^{-1}\otimes I_m\mathbf{y}^{(i)} - hI\otimes I_m\mathbf{f}^{(i)}) - \boldsymbol{\eta}),\end{aligned} \tag{13}$$

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} + \Delta\mathbf{y}^{(i)}, \quad i = 0, 1, \dots,$$

where $\mathbf{y}^{(i)} = (y_1^{(i)}, \dots, y_r^{(i)})^{\mathrm{T}}$, $\mathbf{f}^{(i)} = (f_1^{(i)}, \dots, f_r^{(i)})^{\mathrm{T}}$, $f_j^{(i)} = f(t_j, y_j^{(i)})$, the vector $\boldsymbol{\eta}$ only depends on the initial condition, and, if J denotes the Jacobian of f at (t_0, y_0),

$$\theta = I\otimes\Omega^{-1}, \quad \Omega = (I_m - h\gamma J). \tag{14}$$

Consequently, if v iterations are performed to obtain convergence, the overall computational cost is approximately given by:

- the evaluation of the Jacobian matrix J,
- the factorization of the $m\times m$ matrix Ω in (14),
- rv function evaluations, and
- $2rv$ system solvings with the factors of the matrix Ω.

Let us now briefly sketch the choice of the starting vector $\mathbf{y}^{(0)}$ and the stopping criterion for iteration (13). Concerning the first point, the adopted strategy is similar to that used in most of the available codes: the default profile is obtained by using the interpolating polynomial over the previous block of points; alternatively, we use a constant initial vector (namely, the starting point repeated r times) in either one of the following cases:

- when we integrate over the very first block,
- after a failure of the iteration,
- when the solution is very slow varying.

This last condition is recognized when, on the last block (whose size is r, if the order has not been changed), the following test is true:

$$\forall j = 1,\dots,m : \frac{|y_{rj} - y_{0j}|}{1+|y_{0j}|} < \min\{10^{-2}, 10^2 * \mathtt{tol}_j\} \quad \text{and} \quad \|f_r\|_\infty < 0.5, \tag{15}$$

where $\mathtt{tol}_j \equiv \mathtt{rtol}$ (the prescribed relative tolerance) if $|y_{0j}| > 10^{-1}$, $\mathtt{tol}_j \equiv \mathtt{atol}$ (the prescribed absolute tolerance) if $|y_{0j}| \le 10^{-1}$ and, in general, $y_{\ell j}$ is the jth entry of y_ℓ.

Let us now analyze the stopping criterion for iteration (13). Let us consider the vector $\Delta\mathbf{y}^{(i)}$, as defined in that equation, and introduce the norm

$$\|\Delta\mathbf{y}^{(i)}\| \equiv \max_{\ell=1,\dots,r} |\Delta\mathbf{y}_\ell^{(i)}| \equiv \max_{\ell=1,\dots,r} \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(\frac{\Delta y_{\ell j}}{1+\mathtt{ratol}|y_{0j}|}\right)^2}, \tag{16}$$

where $\texttt{ratol} = \texttt{rtol}/\texttt{atol}$ is the ratio between the specified relative (rtol) and absolute (atol) tolerances, and y_0 is the starting point for the current block. Then, the iteration ends as soon as the following condition is satisfied,

$$\|\Delta \mathbf{y}^{(i)}\| \leqslant \max\left\{c, \frac{\texttt{uround}}{\texttt{rtol}}\right\} * \texttt{atol}, \tag{17}$$

where uround is the machine precision (on input, $\texttt{rtol} > \texttt{uround}$) and the parameter $c = 0.1$. Moreover, in order to make more restrictive the stopping criterion when the solution has small entries and/or is slowly varying, the value of the parameter c may be decreased as follows:

- when $\|y_0\|_{-\infty} \equiv |y_{0s}| < 10^{-2}$, $|f_{0s}| < 10^{-4}$ and $\|f_0\|_\infty < 10^{-3}$, then $c = 5 \cdot 10^{-3}$;
- when (15) holds true, then $c = \min\{c, 5 \cdot 10^{-2}\}$.

Iteration (13) fails if condition (17) is not satisfied within maxit iterations, where this parameter depends on the method currently used, according to Table 1. The iteration also fails if $i > 2$ and $\rho^{(i)} > 0.99$, where $\rho^{(i)}$ is the estimate of the spectral radius of the iteration matrix at the ith iterate. Such an estimate is obtained, after at least two iterations, as follows:

$$\rho^{(1)} = \frac{\|\Delta \mathbf{y}^{(1)}\|}{\|\Delta \mathbf{y}^{(0)}\|}, \quad \rho^{(i)} = \sqrt{\rho^{(i-1)} \frac{\|\Delta \mathbf{y}^{(i)}\|}{\|\Delta \mathbf{y}^{(i-1)}\|}} \quad \text{if} \quad i \geqslant 2. \tag{18}$$

In case of failure of iteration (13) the order of the method is decreased (if $r > 3$) and the stepsize is halved.

4. Stepsize and order variation

In this section we describe the strategies for the variation of both the stepsize of integration h and the order of the method, which rely on the estimate of the local error, obtained through deferred correction (see, for example, [6, Chapter 10]) as follows. As previously said, the basic block method (2) is defined in order to have the equations on each row with an $\mathrm{O}(h^{r+1})$ truncation error. That is, if $\hat{\mathbf{y}}$ denotes the vector obtained by projecting the continuous local solution at the discrete points, and $\hat{\mathbf{f}}$ contains the corresponding values of f, then the local truncation errors for the equivalent forms (6) are defined as

$$\boldsymbol{\tau}_1 \equiv I \otimes I_m \hat{\mathbf{y}} - hC \otimes I_m \hat{\mathbf{f}} - \boldsymbol{\eta}_1, \qquad \boldsymbol{\tau}_2 \equiv \gamma C^{-1} \otimes I_m \boldsymbol{\tau}_1. \tag{19}$$

We recall that, due to the features of the basic block method (2), and provided that $y(t)$ is suitably regular, $\boldsymbol{\tau}_1$ admits the expansion:

$$\boldsymbol{\tau}_1 = \mathbf{v}_{r+1} \otimes h^{r+1} y^{(r+1)}(t_0) + \mathbf{v}_{r+2} \otimes h^{r+2} y^{(r+2)}(t_0) + \cdots. \tag{20}$$

In particular, the last entry of the vector $\mathbf{v}_{r+1}$ is zero. Consequently (see, for example [11, p. 123]) a first-order approximation to the local error is given by (see (14))

$$\mathbf{e} = \theta \boldsymbol{\tau}_1. \tag{21}$$

Then, it follows that we can obtain an efficient estimate of the local error once an estimate of the truncation error τ_1 is available. For this purpose, recall that it is possible to uniquely define $r\times(r+1)$ matrices, $\tilde{A}$ and $\tilde{B}$,

$$[\tilde{\mathbf{a}}\,|\,\tilde{A}] \equiv \left(\begin{array}{c|ccc} -1 & 1 & & \\ \vdots & & \ddots & \\ -1 & & & 1 \end{array}\right), \quad [\tilde{\mathbf{b}}\,|\,\tilde{B}] \equiv \left(\begin{array}{c|ccc} \tilde{\beta}_0^{(1)} & \tilde{\beta}_1^{(1)} & \dots & \tilde{\beta}_r^{(1)} \\ \vdots & \vdots & & \vdots \\ \tilde{\beta}_0^{(r)} & \tilde{\beta}_1^{(r)} & \dots & \tilde{\beta}_r^{(r)} \end{array}\right), \tag{22}$$

such that the coefficients on each row of the two matrices define an r-step LMF with an $\mathrm{O}(h^{r+2})$ truncation error (see, e.g., [6,18]). For such method, the following result holds true [5].

Theorem 2. *Let* $\mathbf{y}$, $\mathbf{f}$ *be the vectors computed by the basic block method* (2). *Let us consider* (*see* (22)) $\tilde{F}(\mathbf{y}) \equiv \tilde{A}\otimes I_m\mathbf{y} - h\tilde{B}\otimes I_m\mathbf{f} - \tilde{\boldsymbol{\eta}}$, *where* $\tilde{\boldsymbol{\eta}} = -\tilde{\mathbf{a}}\otimes y_0 + h\tilde{\mathbf{b}}\otimes f_0$. *It follows then that* (*see* (19)–(20)),

$$\tilde{F}(\mathbf{y}) = -\mathbf{v}_{r+1}\otimes(h\Delta^r f_0). \tag{23}$$

One easily realizes that $-\tilde{F}(\mathbf{y})$ provides a first-order approximation to the leading term in (20). However, in such a case the last block entry is 0, whereas it is known that it must be $\mathrm{O}(h^{p+1})$, if p (see Table 1) is the order of the method. In order to obtain a corresponding approximation also for $\mathbf{e}_r$, the last block entry of $\mathbf{e}$, we then consider the last block entry of the vector (see (14), (19)–(20))

$$\theta(I-\theta)^s(\gamma C^{-1}\mathbf{v}_{r+1}\otimes h\Delta^r f_0), \tag{24}$$

where $s=1$, when $r=3$, and $s=2$, otherwise. This entry turns out to be the one of the largest norm: this feature will be useful for what we shall see in Section 4.1. As a consequence, on one hand, since the norm used to measure the error is the same norm defined in (16), from (14), (21), (23) and (24) one obtains that

$$\|\mathbf{e}\| = \max\{\omega_r\,|\Omega^{-1}g^{(r)}(f_0)|, |\mathbf{e}_r|\} = \mathrm{O}(h^{r+1}), \tag{25}$$

where $\omega_r = \|\mathbf{v}_{r+1}\|_\infty$ and $g^{(r)}(f_0) = h\Delta^r f_0$. On the other hand, the quantity

$$|\mathbf{e}_r| = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(\frac{e_{rj}}{1+\mathtt{ratol}|y_{0j}|}\right)^2} = \mathrm{O}(h^{p+1}), \tag{26}$$

already computed to obtain (25), provides an estimate for $\|\mathbf{e}_{\mathrm{up}}\|$, namely the error corresponding to the use of the next higher-order method. This feature will be conveniently exploited when we shall speak about the order variation strategy. Before that, let us consider the problem of the stepsize variation in more detail. First of all, if `rtol` and `atol` are the prescribed relative and absolute tolerances, then the current solution is accepted provided that (see (25))

$$\|\mathbf{e}\| \leqslant \mathtt{atol}. \tag{27}$$

The new stepsize to be used by the same method is then obtained through extrapolation

$$h_{\text{new}} = h\left(\texttt{sftyerr} * \frac{\texttt{atol}}{\|\mathbf{e}\|}\right)^{1/r+1}, \tag{28}$$

where $\texttt{sftyerr} = \frac{1}{20}$ if (27) holds true and $\texttt{sftyerr} = \frac{1}{10}$ otherwise. Similarly, if $r < 12$ the stepsize to be used by the next higher-order method would be

$$h_{\text{up}} = h\left(\texttt{sftyup} * \frac{\texttt{atol}}{\|\mathbf{e}_{\text{up}}\|}\right)^{1/p+1}, \tag{29}$$

where the approximation $\|\mathbf{e}_{\text{up}}\| = |\mathbf{e}_r|$ has been used (see (26)) and, moreover, we have set $\texttt{sftyup} = \texttt{sftyerr}/2$. We shall use such an estimate for the stepsize of the higher-order method when discussing the order variation strategy. Moreover, we set

$$h_{\text{new}} = \min\{\max\{h_{\text{new}}, 0.12 \times h\}, 10 \times h, (T - t_0)/8\}.$$

In addition to this, if $0.1 \times h \leqslant t \times \texttt{uround}$, then the execution ends because the selected stepsize is too small. Finally, we also use the following heuristics: if `nfail` consecutive failures have occurred (either for the convergence of the iteration or for the accuracy) before the last successful step, then the stepsize is increased only after at least $\texttt{nfail} + 1$ consecutive successful steps occur.

Let us now consider the problem of the order variation. The aim is that of reducing the global computational cost for getting a discrete solution with a prescribed accuracy. For this purpose, we normalize the cost with respect to the width of the covered interval. By neglecting, for sake of simplicity, Jacobian and function evaluations, whose cost in general is strongly problem dependent, we then introduce the following *specific cost per step* function for the method with blocksize r:

$$c_{\text{tot}}(v, r, m, h) = \frac{c_{\text{fatt}} + c_{\text{it}} + c_{\text{err}}}{rh}, \tag{30}$$

where c_{fatt} is the cost for the factorization of the matrix Ω in (14), c_{it} is the number of flops required by v iterations in (13), and c_{err} is the cost for computing estimate (25) of the local error. In particular, in case of full $m \times m$ Jacobians, $c_{\text{fatt}} \approx 2m^3/3$, $c_{\text{it}} \equiv c_{\text{it}}(r, v, m) \approx 4rvm^2$, and $c_{\text{err}} \approx 4$ m^2, if $r = 3$, or 6 m^2, otherwise. Corresponding formulae are used in case of banded Jacobians. Therefore, the next higher-order method, with blocksize r_{up} (see the first column in Table 1), requiring v_{up} iterations for satisfying the same stopping criterion, and using a stepsize h_{up} for getting the same accuracy, would be preferable in the subsequent step provided that

$$c_{\text{tot}}(v_{\text{up}}, r_{\text{up}}, m, h_{\text{up}}) < c_{\text{tot}}(v_{\text{new}}, r, m, h_{\text{new}}), \tag{31}$$

where h_{new} and v_{new} are the stepsize and the number of expected iterations for the current-order method. Therefore, the problem is easily solved, once we have an estimate for the above quantities. We have already seen how to get estimates for h_{new} and h_{up} (see (28) and (29), respectively). It remains to obtain estimates for v_{new} and v_{up}. We observe that, if the same stopping criterion has to be satisfied, then the following equalities should approximately hold,

$$\rho^v = (\rho_{\text{new}})^{v_{\text{new}}} = (\rho_{\text{up}})^{v_{\text{up}}}.$$

In the above equation, ρ is the spectral radius of the current iteration matrix (estimated by (18)), v is the (known) number of iterations carried out to satisfy the convergence criterion (17), and ρ_{new}, ρ_{up} are the spectral radii of the iteration matrices of the current-order method, by using the new stepsize h_{new}, and of the next higher-order method, respectively. By taking into account that the stiff amplification factor of both methods is 0, and considering (11), we then obtain the following estimates,

$$v_{\mathrm{new}} = v\frac{\log\rho}{\log\rho(h_{\mathrm{new}}/h)}, \quad v_{\mathrm{up}} = v\frac{\log\rho}{\log\rho(\tilde{\rho}_{\mathrm{up}}/\tilde{\rho})(h_{\mathrm{up}}/h)}, \tag{32}$$

where $\tilde{\rho}$ and $\tilde{\rho}_{\mathrm{up}}$ are the nonstiff amplification factors of the current and next higher-order methods, respectively (see Table 1). Finally, in order to prevent erratic behaviour in some pathological cases, the previous strategy is applied provided that all the following three conditions are satisfied:

(1) $0.8h \leqslant h_{\mathrm{new}} \leqslant 1.25h$,
(2) at least $\max\{2, \mathtt{nfail}\}$ successful consecutive steps have been carried out with the current-order method, when the previous `nfail` steps failed to satisfy the accuracy requirement (27),
(3) the (estimated) spectral radius of the current iteration, say ρ, is "adequately small". The latter condition is assumed to be fulfilled, provided that $\rho < \rho_p$, where the parameter ρ_p is defined so that all methods do have a prescribed absolute cost to obtain convergence. In more detail, by setting

$$\rho_4 = 10^{-2}|\log_{10}\min\{10^{-1}, \mathtt{rtol}\}|, \tag{33}$$

we impose that, for all allowed orders p, the quantity $c_{\mathrm{it}}(r_p, v_p, m)$ (see Table 1 and (30)) is constant, for the same stopping criterion, where r_p and v_p are the blocksize and the number of iterations required by the pth order method, $p = 4, 6, 8, 10, 12, 14$. This leads to the equalities,

$$r_p v_p = r_{p-2} v_{p-2}, \quad \rho_p^{v_p} = \rho_{p-2}^{v_{p-2}}, \quad p = 6, 8, 10, 12, 14,$$

which provide the following recursion with starting value given by (33)

$$\rho_p = (\rho_{p-2})^{r_p/r_{(p-2)}}, \quad p = 6, 8, 10, 12, 14. \tag{34}$$

We observe that the sequence $\{\rho_p\}$ is a decreasing one.

Actually, the last condition is relaxed when $v \leqslant 3$ and both the conditions of *stepsize stagnation* and *convergence stagnation*, as described in Section 4.1 below, are verified.

So far, we have dealt with the strategy for increasing the order of the method to be used at the subsequent step of numerical integration. However, it may be convenient to decrease the order of the method as well. Obviously, the criterion based on the minimization of the specific cost per step (30) could be, in principle, also used to decrease the order of the method, provided that an estimate for h_{low}, namely the stepsize to be used by the next lower-order formula, is available. Its computation, based on a procedure similar to that required for evaluating h_{new}, would require an additional linear system with the matrix Ω to be solved. Nevertheless, we decided not to systematically resort to such a criterion for decreasing the order, because there is numerical evidence that it is seldom effective.

Instead, we chose to lower the order p to $p-2$ (when $r>3$, see Table 1), in either one of the following two situations:

- a failure of the nonlinear iteration (13) occurs (in such a case, $h_{\text{new}}=h/2$, as we have already said at the end of Section 3);
- all the following four conditions hold true:
  (1) in the last step the current-order method has been successful,
  (2) the nonlinear iteration (13) has required more than three iterations,
  (3) the (estimated) spectral radius of the iteration matrix, ρ, satisfies $\rho>\rho_p$, where ρ_p is again defined according to (34), but with the initial condition, in place of (33),

$$\rho_4 = 0.5; \tag{35}$$

  (4) if condition (37) below is satisfied, then $h_{\text{low}} \geqslant h_{\text{new}}$.

In this case, the new stepsize is set equal to

$$\begin{cases} h_{\text{low}} & \text{if (37) and } h_{\text{low}} \geqslant h_{\text{new}} \text{ hold true,} \\ \min\{h_{\text{low}}, h_{\text{new}}\} & \text{otherwise.} \end{cases}$$

4.1. Order reduction recovery

A particular handling is required in order to get rid of the so-called *order reduction phenomenon* (see, for example, [11, Chapter IV, 15]). Such a phenomenon occurs when, in test equation (4), $h\to 0$ but $q=h\mu$ is large. In such a case, in fact, expansion (20) of the truncation error becomes

$$\boldsymbol{\tau}_1 = q^{r+1}\mathbf{v}_{r+1}y_0 + q^{r+2}\mathbf{v}_{r+2}y_0 + \cdots,$$

and the local error is given by $(I-qC)^{-1}\boldsymbol{\tau}_1$. The latter expression admits different expansions, depending on the "size" of q. In particular,

- when $|q|$ is small, then

$$(I-qC)^{-1}\boldsymbol{\tau}_1 = q^{r+1}\mathbf{v}_{r+1}y_0 + q^{r+2}(\mathbf{v}_{r+2}+C\mathbf{v}_{r+1})y_0 + \cdots,$$

  and the principal term of each entry behaves like q^{r+1}, with the exception of the last one, which depends on higher-order terms;
- when $|q|$ is large, then

$$(I-qC)^{-1}\boldsymbol{\tau}_1 \approx -q^r C^{-1}\mathbf{v}_{r+1}y_0 + \cdots. \tag{36}$$

  In such a case, the principal term of each entry behaves like q^r, including the last one.

The conclusions in the latter case make evident the fact that $|\mathbf{e}_r|$ (see (26)) is no more an estimate for $\|\mathbf{e}_{\text{up}}\|$. On the other hand, when q is large, it happens that, see (25) and (26),

$$|\mathbf{e}_r| \equiv \|\mathbf{e}\|, \tag{37}$$

i.e., the norm of the last (block) entry of the vector defined in (24). Moreover, the latter vector turns out to be an approximation to the principal term of expansion (36). In conclusion, if the order reduction phenomenon occurs, then the strategy for the order variation previously described, which relies on the higher-order accuracy of the last entry of the local error, may fail. Indeed, this actually

happens for the well-known Prothero–Robinson problem (see [11]). In such a case, also the stepsizes stagnate. In the code `BiM`, the order reduction phenomenon is recognized when (37) holds true or all the following conditions are satisfied:

Order stagnation: the order of the method has not been increased by the above mentioned strategy;

Error stagnation: $|\mathbf{e}_r|\,\texttt{faterr} \geqslant \|\mathbf{e}\|$, where the parameter `faterr` is chosen according to Table 1. When such a condition holds true, this means that the last entry of the local error is "not too small", with respect to the remaining ones. This is, indeed, usually the case, when it correctly estimates the error for the next higher-order method. The parameter `faterr` is, at the moment, chosen in a heuristic way;

Stepsize stagnation: the ratio between the new stepsize, h_{new}, and the current one, h, belongs to the interval $[0.95, 1.05]$;

Convergence stagnation: the ratio between the current estimated spectral radius, ρ (see (18)), and the one of the previous iteration, ρ_{old}, belongs to the interval $[0.95, 1.05]$.

Once the error reduction phenomenon is recognized, it is possible to get rid of it, as explained in the sequel. The basic idea is to obtain an estimate for $\|\mathbf{e}_{\text{up}}\|$ in a form similar to (25):

$$\|\mathbf{e}_{\text{up}}\| \approx \omega_{r_{\text{up}}} |\Omega^{-1} g^{(r_{\text{up}})}(f)|. \tag{38}$$

Indeed, the quantity $\omega_{r_{\text{up}}}$ is known. Concerning the second term, $g^{(r_{\text{up}})}(f)$ can be approximated by suitable first (in the case $r = 3$) or second (in the case $r > 3$) differences of $g^{(r)}(f)$, since this function has already been computed at the previous blocks. Once estimate (38) is available, the usual formula (29) can then be used, in order to predict h_{up}.

An additional question needs to be considered, at this point, by observing that, when q is not small, then (11) is not valid. The latter approximated equality, in turn, was used in order to predict ρ_p and ρ_{up} from the knowledge of ρ, h, h_{new}, h_{up} (see (32)). However, when q is large, from (12) it is not difficult to prove the following result.

Theorem 3. *For $|q| \gg 1$ the spectral radius of the iteration matrix (10) is approximately given by*

$$\rho(q) \approx \frac{|\lambda_1 - \gamma|^2}{|\lambda_1|\gamma^2|q|} \equiv \frac{\tilde{\rho}^{(\infty)}}{|q|}, \quad \tilde{\rho}^{(\infty)} = \frac{\tilde{\rho}}{|\lambda_1|^2} \equiv \frac{2\rho^*}{|\lambda_1|}.$$

The corresponding values of the parameter $\tilde{\rho}^{(\infty)}$ are listed in Table 1. The previous result allows us to derive the following estimates for ν_{new} and ν_{up}, alternative to (32):

$$\nu_{\text{new}} = \nu \frac{\log \rho}{\log \rho(h/h_{\text{new}})}, \quad \nu_{\text{up}} = \nu \frac{\log \rho}{\log \rho(\tilde{\rho}_{\text{up}}^{(\infty)}/\tilde{\rho}^{(\infty)})(h/h_{\text{up}})},$$

where $\tilde{\rho}^{(\infty)}$ is the parameter of the current-order method, and $\tilde{\rho}_{\text{up}}^{(\infty)}$ is that of the next higher-order one. The above estimates are then used to check (31), in order to decide whether to increase the order of the method used in the subsequent step, when the order reduction phenomenon is diagnosed.

Finally, we mention that, for robustness, when (37) holds true, the order is not increased when the following two conditions are fulfilled:

- $h_{\mathrm{up}} \geqslant h$,
- the estimated spectral radius for the higher-order method,

$$\rho_{\mathrm{up}} = \rho \frac{\tilde{\rho}_{\mathrm{up}}^{(\infty)}}{\tilde{\rho}^{(\infty)}} \frac{h}{h_{\mathrm{up}}}, \tag{39}$$

is larger than the corresponding maximum allowed value, as defined by (34) and (35).

Indeed, the first condition ensures that approximation (39), derived from Theorem 3, is appropriate also for the next higher-order method.

5. Numerical tests and conclusions

In this section we report the results obtained on a few test problems taken from [11] and from the former CWI testset [17], now available at the University of Bari [14]. In the tests, we compare the code `BiM` [15] with some of the best codes currently available for the numerical integration of ODE-IVPs: the codes `DASSL` [1], `GAM` [16], `MEBDFDAE` [9], `RADAU5` and `RADAU` [11]. In Fig. 1we report the work-precision diagrams for the following problems: (a) Van der Pol; (b) Robertson; (c) Plate; (d) Bruss (1D diffusion); (e) Beam; (f) Emep; (g) Hard Ring Modulator. All executions have been carried out on a Pentium based computer, under Linux, and by using the same compilation options (`-O3`). The execution times (on the y-axis) are in seconds. As usual, on the x-axis there is the number of significant computed digits (scd) in the numerical solution. The parameters listed in the following table have been used for all codes (`h0` is the initial stepsize; `rtol` and `atol` have been already defined in Section 3).

Problem	(a)	(b)	(c)	(d)	(e)	(f)	(g)
`h0`	$10^{-(2+\ell/2)}$	$10^{-(2+\ell/2)}$	$10^{-(2+\ell/2)}$	$10^{-(2+\ell/2)}$	$10^{-(2+\ell/4)}$	10^{-7}	$10^{-(4+\ell/2)}$
`rtol`	$10^{-(2+\ell/2)}$	$10^{-(2+\ell/2)}$	$10^{-(2+\ell/2)}$	$10^{-(2+\ell/2)}$	$10^{-(2+\ell/4)}$	$10^{-(2+\ell/2)}$	$10^{-(4+\ell/2)}$
`atol`	$10^{-(2+\ell/2)}$	$10^{-(2+\ell/2)}$	$10^{-(2+\ell/2)}$	$10^{-(2+\ell/2)}$	$10^{-(2+\ell/4)}$	1	$10^{-(4+\ell/2)}$
ℓ-range	$0, \ldots, 22$	$0, \ldots, 24$	$0, \ldots, 22$	$0, \ldots, 24$	$0, \ldots, 20$	$0, \ldots, 20$	$0, \ldots, 16$

It is worth mentioning that for the code `BiM` we have always got a correct answer, whereas, for the other codes, possible failures have occurred when using either the coarsest or the finest accuracy requirements.

From the results obtained, we can conclude that the code `BiM` turns out to be a robust and reliable one. Moreover, it is competitive with some of the best existing codes. Finally, its parallelization, naturally induced by the fact that the splitting is block-diagonal, seems to be very promising, by considering that the code `BiM` implements methods with blocksize r ranging from 3 to 12. Future extensions of this research will deal with such a parallelization, as well as with the extension of the code to deal with implicit differential equations (IDEs) and differential algebraic equations (DAEs).

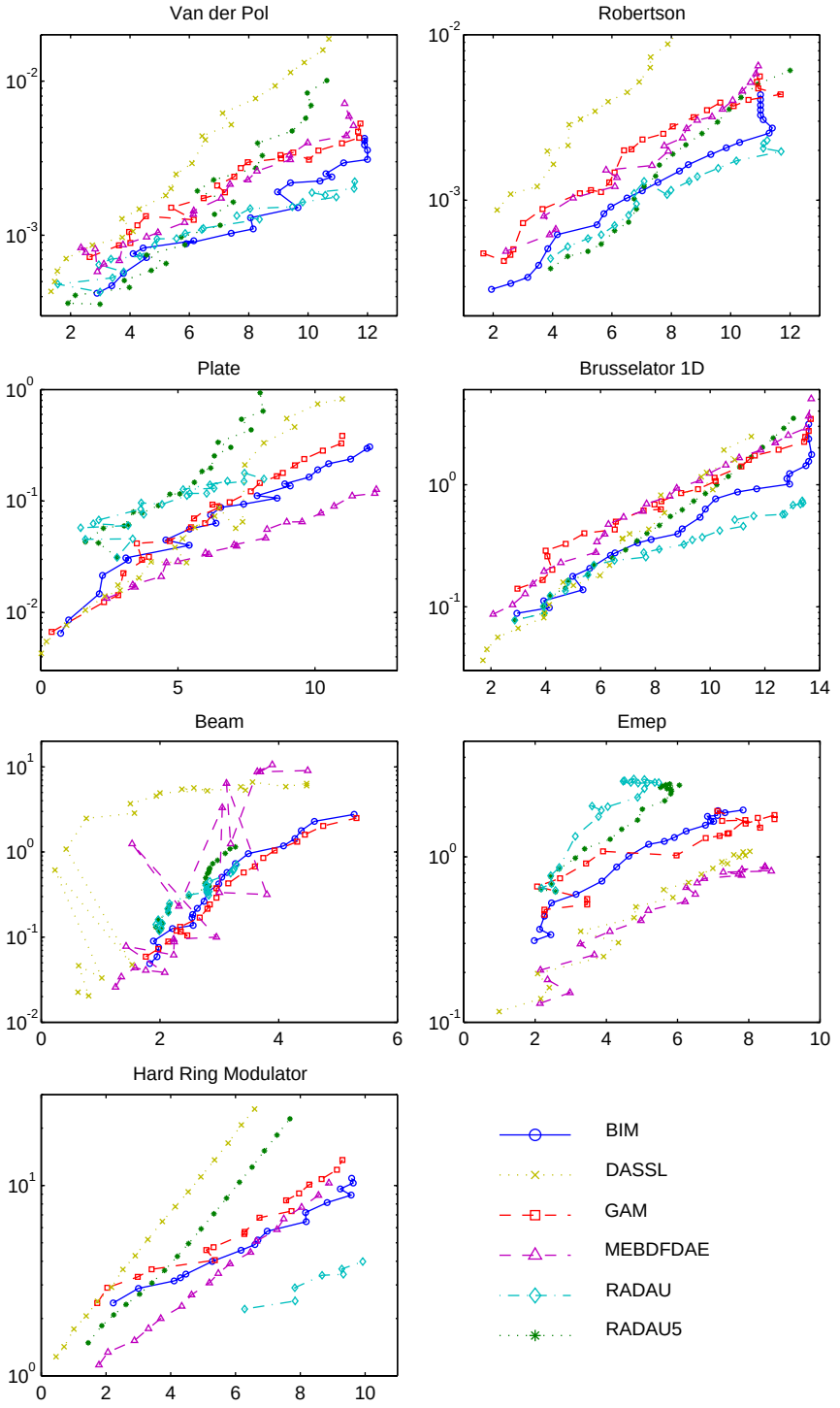

Fig. 1. Computed work-precision diagrams.

Acknowledgements

The authors are very indebted to the referees for their comments and suggestions.

References

[1] K.E. Brenan, S.L. Campbell, L.R. Petzold, Numerical solution of initial-value problems in differential-algebraic equations. Classics in Applied Mathematics, Vol.14, SIAM, Philadelphia, 1996. Code available at: http://www.netlib.org/ode/ddassl.f
[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, C. Magherini, Blended implementation of block implicit methods for ODEs, Appl. Numer. Math. 42 (2002) 29–45.
[4] L. Brugnano, C. Magherini, Some linear algebra issues concerning the implementation of blended implicit methods, Numer. Linear Algebra Appl., accepted for publication.
[5] L. Brugnano, C. Magherini, Economical error estimates for block implicit methods for ODEs via deferred correction, submitted for publication.
[6] L. Brugnano, D. Trigiante, Solving Differential Problems by Multistep Initial and Boundary Value Methods, Taylor & Francis, London, 1998.
[7] L. Brugnano, D. Trigiante, Block implicit methods for ODEs, in: D. Trigiante (Ed.), Recent Trends in Numerical Analysis, Nova Science Publ. Inc., New York, 2001, pp. 81–105.
[8] J.C. Butcher, The Numerical Analysis of Ordinary Differential Equations: Runge–Kutta Methods and General Linear Methods, Wiley, New York, 1987.
[9] J.R. Cash, S. Considine, An MEBDF code for stiff Initial Value Problems, ACM Trans. Math. Software 18(2) (1992) 142–158. Code available at: http://www.ma.ic.ac.uk/∼jcash/IVP_software/readme.html
[10] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, Springer Series in Computational Mathematics, 2nd Edition, Vol.8, Springer, Berlin, 1993.
[11] E. Hairer, G. Wanner. Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems, Springer Series in Computational Mathematics, 2nd Edition, Vol.14, Springer, Berlin, 1996. Codes available at: http://www.unige.ch/math/folks/hairer/software.html
[12] 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.
[13] 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.
[14] http://www.dm.uniba.it/∼testset/
[15] http://www.math.unifi.it/∼brugnano/BiM/
[16] 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
[17] W.M. Lioen, J.J.B. deSwart, W.A. van der Veen, Test set for IVP solvers, Report NM-R96150, CWI, Department of Mathematics, Amsterdam, 1996.
[18] H.A. Watts, L.F. Shampine, *A*-stable block one-step methods, BIT 12 (1972) 252–266.