ELSEVIER

Applied Numerical Mathematics 28 (1998) 127–141

APPLIED NUMERICAL MATHEMATICS

Parallel implementation of block boundary value methods on nonlinear problems: theoretical results ☆

Luigi Brugnano [a,*], Donato Trigiante [b,1]

[a] *Dipartimento di Matematica "U. Dini", Viale Morgagni 67/A, I-50134 Firenze, Italy*

[b] *Dipartimento di Energetica "S. Stecco", Via C. Lombroso 6/17, I-50134 Firenze, Italy*

Received 4 September 1997; received in revised form 16 September 1997; accepted 16 September 1997

Dedicated to Professor Mario R. Occorsio in occasion of his 65th birthday

Abstract

Recently a new parallel ODE solver implementing a "parallelism across the steps" has been proposed (Amodio and Brugnano, 1997; Brugnano and Trigiante, 1998). In the mentioned references, the attention was devoted to some essential features of the parallel method, which are already present in the case where it is used to approximate linear continuous problems. In this paper, the previous analysis is completed by discussing questions which typically arise when approximating nonlinear continuous problems. In particular, we shall study, for the parallel solver, the problem of the mesh selection and the convergence of the nonlinear iteration.

Keywords: Parallel methods for ODEs; Nonlinear Gauss–Seidel iteration; Mesh selection; Modified Newton method; Boundary Value Methods

1. Introduction

The study of new methods for the numerical solution of initial value problems (IVPs) for ODEs,

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

suitable for parallel computers has been the object of much research in the last thirty years (see [11] for a complete overview). In particular, methods trying to approximate the solution of problem (1) simultaneously at different grid points, are classified as having a *parallelism across the steps*. A new method implementing this kind of parallelism has been recently proposed in [1,2,9]. The parallel solver is based on block Boundary Value Methods (B_2VMs), recently introduced by the authors for

☆ Work supported by CNR (contract no. 96.00243.CT01) and MURST.

* Corresponding author. E-mail: na.brugnano@na-net.ornl.gov.

[1] E-mail: na.dtrigiante@na-net.ornl.gov.

0168-9274/98/$19.00
PII: S0168-9274(98)00040-3

the approximation of Hamiltonian problems [8–10]. The same methods have been also considered for constructing a very efficient sequential code [12], which compares well with the most reliable existing ones (for example, RADAU5).

The analysis carried out in [1,2] concerns the application of the methods to linear continuous problems. In such case, in fact, one may not consider, in a simplified analysis, matters which arise when dealing with nonlinear problems. Among them, the most important are the mesh selection and the choice of the starting approximation to the solution for the iterative process involved. The aim of the present paper is the study of such problems. As a result, a complete scheme for the parallel solver will be derived. The actual implementation of the proposed method on a parallel computer, along with the analysis of its parallel performances, will be considered in a companion paper [3].

The structure of the paper is the following: in Section 2 we recall the main facts about B_2VMs and their parallel implementation; Section 3 is devoted to the mesh selection and the derivation of the starting approximation for the nonlinear iteration, whose convergence is studied in Section 4. Finally, Section 5 contains some numerical tests carried out by using a Matlab prototype of the parallel solver.

2. Block Boundary Value Methods (B_2VMs)

For a complete description of BVMs we refer to the book [9]. Here we shall confine ourselves to a very short introduction.

2.1. *Boundary Value Methods (BVMs)*

The shortest way of introducing Boundary Value Methods (BVMs) is to consider the approximation of problem (1) by using a k-step LMF,

$$\sum_{i=0}^{k} \alpha_i y_{n+i} = h \sum_{i=0}^{k} \beta_i f_{n+i}, \tag{2}$$

where, as usual, $h = (T - t_0)/N$ is the stepsize, y_n is the discrete approximation to $y(t_n)$ and $f_n = f(t_n, y_n)$, $t_n = t_0 + nh$. To get a discrete problem from Eq. (2), k independent conditions must be imposed. They are chosen by fixing the first $k_1 \leqslant k$ initial points, $y_0, y_1, \ldots, y_{k_1-1}$, of the discrete solution, and $k_2 = k - k_1$ final ones, $y_{N-k_2+1}, \ldots, y_N$. In other words, we are approximating the continuous IVP (1) by means of a discrete boundary value problem. The latter defines a BVM with (k_1, k_2)-*boundary conditions*, which may be stable and have arbitrarily high order [6]. As matter of fact, the Dahlquist barriers are overcome, when $k_1 < k$.

In particular, we shall consider the so called *Generalized Adams Methods* (GAMs) [4,7,9], which are methods having the following form:

$$y_n - y_{n-1} = h \sum_{i=-\nu}^{k-\nu} \beta_{i+\nu} f_{n+i}, \quad n = \nu, \ldots, N - k + \nu, \tag{3}$$

where

$$\nu = \begin{cases} (k+1)/2, & \text{for odd } k, \\ k/2, & \text{for even } k. \end{cases}$$

For all $k \geqslant 1$, the coefficients $\{\beta_i\}$ are uniquely determined by imposing that the formula has order $k+1$. The resulting method must be used with $(\nu, k-\nu)$-boundary conditions. We observe that for $k=1$ we get the usual trapezoidal rule.

In principle, the boundary conditions for (3) require the following values:

$$y_0,\ y_1,\ \ldots,\ y_{\nu-1},\ y_{N-k+\nu+1},\ \ldots,\ y_N,$$

of the discrete solution. Of such values, only y_0 is available from the initial condition of the continuous problem. However, we can treat the remaining values as unknowns, provided that an equal number of equations is introduced. This is done by means of a suitable set of *additional initial* and *final methods*. For GAMs they are conveniently chosen in the following form:

$$y_j - y_{j-1} = h\sum_{i=0}^{k} \beta_i^{(j)} f_i, \qquad j = 1, \ldots, \nu-1, \tag{4}$$

$$y_j - y_{j-1} = h\sum_{i=0}^{k} \beta_{k-i}^{(j)} f_{N-i}, \quad j = N-k+\nu+1, \ldots, N. \tag{5}$$

The coefficients of the additional methods (4), (5) are uniquely determined by imposing that each formula has the same order $k+1$ of the *main method* (3).

2.2. *Block version of the methods and parallel implementation*

From the above arguments one has that a BVM can be considered as a composite method made up by a main method coupled with an appropriate set of additional methods. An instance is given by (3)–(5). In this formulation, the method only requires one value, i.e., y_0. This feature has led to define a block version of BVMs [8], which has been successfully used for approximating Hamiltonian problems.

Thus, the block version consists in discretizing the interval $[t_0, T]$ by using two different meshes: a coarser one and a finer one. Let the coarser mesh contain the $p+1$ points

$$\tau_i = \tau_{i-1} + \widehat{h}_i, \quad i = 1, \ldots, p, \qquad \tau_0 \equiv t_0, \qquad \tau_p \equiv T.$$

Then, on each subinterval $[\tau_{i-1}, \tau_i]$, $i = 1, \ldots, p$, we apply the same (composite) BVM as described above, by using the *finer stepsize* $h_i = \widehat{h}_i / s$.

As a consequence, the points in the finer mesh belonging to the subinterval $(\tau_{i-1}, \tau_i]$, are given by

$$t_{ji} = \tau_{i-1} + jh_i, \quad j = 1, \ldots, s,\ i = 1, \ldots, p. \tag{6}$$

They are called *internal steps* and the rightmost lower index of t_{ji} identifies the ith subinterval. In the following it will be convenient to use the notation t_{0i} to denote τ_{i-1}. In this case, it must be observed that

$$t_{01} \equiv \tau_0, \qquad t_{0i} \equiv \tau_{i-1} \equiv t_{s,i-1}, \quad i = 1, \ldots, p. \tag{7}$$

The global discrete problem is then given by

$$(A \otimes I_m)\boldsymbol{y} - (H \cdot B \otimes I_m)\boldsymbol{f} = \boldsymbol{\eta}, \tag{8}$$

where the following notation has been used: I_m is the identity matrix of size m (the size of the continuous problem), $\boldsymbol{\eta} = \boldsymbol{e}_1 \otimes \eta$, $\boldsymbol{e}_1$ is the first unit vector in $\mathbb{R}^{ps+1}$, and

$$\boldsymbol{y} = (\, y_0^{\mathrm{T}} \quad \boldsymbol{y}_1^{\mathrm{T}} \quad \ldots \quad \boldsymbol{y}_p^{\mathrm{T}}\,)^{\mathrm{T}}, \qquad \boldsymbol{f} = (\, f_0^{\mathrm{T}} \quad \boldsymbol{f}_1^{\mathrm{T}} \quad \ldots \quad \boldsymbol{f}_p^{\mathrm{T}}\,)^{\mathrm{T}},$$

where

$$\boldsymbol{y}_i = \begin{pmatrix} y_{1i}^{\mathrm{T}} & \dots & y_{si}^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}}, \qquad \boldsymbol{f}_i = \begin{pmatrix} f_{1i}^{\mathrm{T}} & \dots & f_{si}^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}}, \quad f_{ji} = f(t_{ji}, y_{ji}).$$

Finally,

$$A \equiv A_{p\cdot s} = \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix}_{(ps+1)\times(ps+1)}, \qquad H \equiv H_p = \begin{pmatrix} 0 & & & \\ & h_1 I_s & & \\ & & \ddots & \\ & & & h_p I_s \end{pmatrix},$$

$$B = \begin{pmatrix} 0 & & & & & & & \\ b_s & \widehat{B}_s & c_s & & & & & \\ & b_s & \widehat{B}_s & c_s & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & & b_s & \widehat{B}_s & c_s & \\ & & & & & b_s & \widehat{B}_s & c_s \end{pmatrix}_{(ps+1)\times(ps+1)}, \tag{9}$$

where (see (3)–(5)),

$$[b_s \mid \widehat{B}_s \mid c_s]$$

$$= \left(\begin{array}{c|ccc|ccc|c} \beta_0^{(1)} & \beta_1^{(1)} & \dots & \beta_k^{(1)} & & & & \\ \vdots & \vdots & & \vdots & & & & \\ \beta_0^{(\nu-1)} & \beta_1^{(\nu-1)} & \dots & \beta_k^{(\nu-1)} & & & & \\ \beta_0 & \beta_1 & \dots & \beta_k & & & & \\ & \beta_0 & & & \ddots & & & \\ & & \ddots & & & \ddots & & \\ & & & \ddots & & & \ddots & \\ & & & & \beta_0 & \dots & \beta_k & \\ & & & & \beta_0^{(s-k+\nu+1)} & \dots & \beta_k^{(s-k+\nu+1)} & \\ & & & & \vdots & & \vdots & \\ & & & & \beta_0^{(s)} & \dots & \beta_k^{(s)} & \end{array}\right)_{s\times(s+1)}. \tag{10}$$

We observe that, because of consistency conditions, the above matrix has unit row sums.

When the continuous problem (1) is linear, the discrete problem (8) is a lower block bidiagonal linear system, whose solution can be efficiently obtained in parallel on p processors. This is, in fact, the basic idea exploited in [1,2], where it is shown that an almost perfect speed-up can be obtained, provided that

the number of steps k of the main method and/or the blocksize s are large enough. This is welcome since the higher k, the higher the order of the method.

When problem (1) is nonlinear, a corresponding linear system having the same structure as in the linear case can still be obtained by applying, for example, the Newton method for solving (8). In this case, however, the following two problems arise:

(1) determine an efficient way for choosing the appropriate mesh (i.e., to define the finer stepsizes $h_1, \ldots, h_p$);

(2) get a good starting point, say $y^{(0)}$, for the nonlinear iteration.

As matter of fact, both requirements are crucial for the convergence of the nonlinear iteration.

3. Diagonally Linearized Gauss–Seidel Method (DLGSM)

In order to solve the above mentioned problems, we propose to use a very cheap sequential method. This allows to get the mesh and the starting approximation $y^{(0)}$. After that, a more accurate discrete solution is obtained by applying the simplified Newton iteration,

$$Mz^{(i)} = g^{(i)}, \quad y^{(i+1)} = y^{(i)} - z^{(i)}, \quad i = 0, 1, \ldots, \tag{11}$$

where (see (8))

$$g^{(i)} = g\big(y^{(i)}\big) \equiv (A \otimes I_m)y^{(i)} - (H \cdot B \otimes I_m)f^{(i)} - \eta,$$
$$M = (A \otimes I_m) - (H \cdot B \otimes I_m)J^{(0)},$$

and $J^{(0)}$ is the block diagonal matrix with the Jacobians of the function f at $y^{(0)}$.

In Section 4 we study the convergence of the *outer iteration* (11). For the moment, we analyze the starting procedure for determining the mesh points and the vector $y^{(0)}$. This is achieved by using the trapezoidal rule, which has stability properties similar to those of higher order GAMs, implemented in block form. Namely (see (6)), by considering s consecutive steps with the same stepsize h_i. Since this is a sequential procedure, we can study it over any subinterval $[\tau_{i-1}, \tau_i]$ of the coarse mesh. For this reason, in order to simplify the notation, we shall skip the index for the subinterval itself.

Let τ be a generic point where the approximate solution η_τ is known, and let h be the stepsize. The discrete problem defined by the trapezoidal rule at the points $t_i = \tau + ih$, $i = 0, \ldots, s$, is then given by

$$(A_s \otimes I_m)y_\tau - h(B_s \otimes I_m)f_\tau = e_1 \otimes \eta_\tau, \tag{12}$$

where e_1 is the first unit vector in $\mathbb{R}^{s+1}$, and

$$y_\tau = \begin{pmatrix} y_0 \\ \vdots \\ y_s \end{pmatrix}, \qquad f_\tau = \begin{pmatrix} f_0 \\ \vdots \\ f_s \end{pmatrix}, \qquad B_s = \frac{1}{2}\begin{pmatrix} 0 & & & \\ 1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \end{pmatrix}_{(s+1)\times(s+1)}.$$

Eq. (12), which we assume to have a unique solution, will be solved by using a Gauss–Seidel method with diagonal linearization, DLGSM hereafter (Diagonally Linearized Gauss–Seidel Method). By introducing the matrices

$$\widetilde{I}_s = I_{s+1} - e_1 e_1^{\mathrm{T}}, \qquad L_s = 2B_s - \widetilde{I}_s,$$

and denoting by J_0 the Jacobian of f at the initial point, it is defined by the following splitting:

$$\left(A_s \otimes I_m - \frac{h}{2}\widetilde{I}_s \otimes J_0\right) \boldsymbol{y}_\tau - \frac{h}{2}(L_s \otimes I_m)\boldsymbol{f}_\tau = e_1 \otimes \eta_\tau + \frac{h}{2}\left((\widetilde{I}_s \otimes I_m)\boldsymbol{f}_\tau - (\widetilde{I}_s \otimes J_0)\boldsymbol{y}_\tau\right). \tag{13}$$

We then obtain the iterative scheme

$$\begin{aligned}&\left(A_s \otimes I_m - \frac{h}{2}\widetilde{I}_s \otimes J_0\right) \boldsymbol{y}_\tau^{(j)} - \frac{h}{2}(L_s \otimes I_m)\boldsymbol{f}_\tau^{(j)} \\ &\quad = e_1 \otimes \eta_\tau + \frac{h}{2}\left((\widetilde{I}_s \otimes I_m)\boldsymbol{f}_\tau^{(j-1)} - (\widetilde{I}_s \otimes J_0)\boldsymbol{y}_\tau^{(j-1)}\right), \quad j = 1, \ldots, g,\end{aligned} \tag{14}$$

where the vectors $\boldsymbol{y}_\tau^{(j)}$ and $\boldsymbol{f}_\tau^{(j)}$ contain the approximations to $\boldsymbol{y}_\tau$ and $\boldsymbol{f}_\tau$ (see (12)) at the jth iterate, and g is the maximum number of iterations allowed (see later).

3.1. Convergence of DLGSM

Let us now study the convergence of the iteration (14). By defining

$$\boldsymbol{e}_\tau^{(j)} = \boldsymbol{y}_\tau - \boldsymbol{y}_\tau^{(j)}, \qquad \delta\boldsymbol{f}_\tau^{(j)} = \boldsymbol{f}_\tau - \boldsymbol{f}_\tau^{(j)},$$

from (13) and (14) we obtain

$$\begin{aligned}&\left(A_s \otimes I_m - \frac{h}{2}\widetilde{I}_s \otimes J_0\right) \boldsymbol{e}_\tau^{(j)} - \frac{h}{2}(L_s \otimes I_m)\delta\boldsymbol{f}_\tau^{(j)} = \frac{h}{2}\left((\widetilde{I}_s \otimes I_m)\delta\boldsymbol{f}_\tau^{(j-1)} - (\widetilde{I}_s \otimes J_0)\boldsymbol{e}_\tau^{(j-1)}\right), \\ &\quad j = 1, \ldots, g.\end{aligned} \tag{15}$$

By the way, we observe that if the function $f(t, y)$ is independent of y, namely (1) is a pure quadrature problem, then $\delta\boldsymbol{f}_\tau^{(j)} = \boldsymbol{0}$ and $J_0 = 0$. Consequently, one has $\boldsymbol{e}_\tau^{(1)} = \boldsymbol{0}$, independently of the stepsize h. The same happens for linear autonomous problems.

Assuming now that the function f is suitably smooth, it follows that

$$\delta\boldsymbol{f}_\tau^{(j)} = \begin{pmatrix} J_0 & & \\ & \ddots & \\ & & J_s \end{pmatrix} \boldsymbol{e}_\tau^{(j)} + \mathrm{O}(\|\boldsymbol{e}_\tau^{(j)}\|^2) = (I_{s+1} \otimes J_0 + hD_s \otimes J_0' + \mathrm{O}(h^2))\boldsymbol{e}_\tau^{(j)} + \mathrm{O}(\|\boldsymbol{e}_\tau^{(j)}\|^2),$$

where J_i is the Jacobian of f at $(t_i, y(t_i))$, J_0' is the derivative of the Jacobian at $(t_0, y(t_0))$, and $D_s = \mathrm{diag}(0, 1, \ldots, s)$. One then concludes that, if the procedure converges, Eq. (15) can be approximated as

$$(A_s \otimes I_m - hB_s \otimes J_0)\boldsymbol{e}_\tau^{(j)} = \frac{h}{2}\left((hD_s \otimes J_0')\boldsymbol{e}_\tau^{(j-1)} + \mathrm{O}(\|\boldsymbol{e}_\tau^{(j-1)}\|^2)\right), \quad j = 1, \ldots, g. \tag{16}$$

The coefficient matrix $(A_s \otimes I_m - hB_s \otimes J_0)$ is nonsingular and has a bounded inverse, when J_0 has no eigenvalues with positive real part. The same applies when J_0 has eigenvalues with positive real part, provided that h is small enough. For sake of simplicity, we shall then assume that the above mentioned matrix is nonsingular and with bounded inverse. By considering norms, we finally obtain

$$\|\boldsymbol{e}_\tau^{(j)}\| \leqslant c_1 h^2 \|\boldsymbol{e}_\tau^{(j-1)}\| + c_2 h \|\boldsymbol{e}_\tau^{(j-1)}\|^2, \quad j \geqslant 1, \tag{17}$$

for suitable constants c_1 and c_2 independent of h.

Eq. (17) allows us to derive the following conclusions concerning the convergence of DLGSM:

1. When $\|\boldsymbol{e}_\tau^{(j-1)}\|$ is large with respect to h, then the first term on the right hand side is negligible. Quadratic convergence is then to be expected in the initial iterations.

2. Conversely, as soon as the first term dominates, the iteration converges linearly.

By taking into account that the local error of the trapezoidal rule is $O(h^3)$, the above arguments suggest that the number g of iterations for the procedure (14) must be of moderate size. In fact, if we consider the starting vector

$$\boldsymbol{y}_\tau^{(0)} = (1,\ 1,\ \ldots,\ 1)^{\mathrm{T}} \otimes \eta_\tau, \tag{18}$$

the assumptions on the function f imply that $\|\boldsymbol{e}_\tau^{(0)}\| \approx O(h)$. Consequently, from (17) we obtain $\|\boldsymbol{e}_\tau^{(j)}\| \approx O(h^{2j+1})$, $j = 0, 1, \ldots, g$. Therefore, by using $g = 2$ we can expect that the error associated with the convergence of iteration (14) is smaller than the local error. However, the stepsize control described later, based on the estimates

$$\|\boldsymbol{e}_\tau^{(j)}\| \approx \|\boldsymbol{y}_\tau^{(j+1)} - \boldsymbol{y}_\tau^{(j)}\|, \tag{19}$$

needs $g = 3$ iterations for the DLGSM.

Concerning the convergence of DLGSM, it can be established by using the comparison principle [13]. The following sufficient condition is then derived.

Theorem 1. *For all sufficiently small $h > 0$, the DLGSM iteration is convergent.*

In order to meet the requirements of Theorem 1, we start DLGSM by using a suitably small initial stepsize. This stepsize will be then changed, in the subsequent subintervals, depending on the convergence of the method. This is discussed in the following section.

3.2. Convergence of DLGSM and stepsize selection

In this section the problem of stepsize selection is analyzed in more detail. Namely, we study the determination of the new stepsize h_{new}, depending on the current stepsize h and on the convergence of the DLGSM iteration. The following arguments assume (see (18)) that, for a suitable $\beta > 0$, $\|\boldsymbol{e}_\tau^{(0)}\| \approx \beta h$. In our analysis, we shall distinguish the two cases: (i) $\beta \approx 1$, and (ii) $\beta \ll 1$. Moreover, by considering the comparison equation associated with (17), namely

$$x_j = c_1 h^2 x_{j-1} + c_2 h x_{j-1}^2, \quad j \geqslant 1, \qquad x_0 = \|\boldsymbol{e}_\tau^{(0)}\|, \tag{20}$$

we shall suppose that (see (19)), $x_j \approx \|\boldsymbol{e}_\tau^{(j)}\| \approx \|\boldsymbol{y}_\tau^{(j+1)} - \boldsymbol{y}_\tau^{(j)}\|$.

In case (i), by performing $g = 3$ iterations of DLGSM, we obtain estimates for x_0, x_1 and x_2. Consequently, from (20) we get

$$x_3 \approx \beta c_1^2 (c_1 + \beta c_2) h^7 \approx \frac{x_2^2}{x_1}.$$

Having fixed a tolerance ε for x_3, one then obtains that the stepsize

$$h_{\mathrm{new}} = h \left(\varepsilon \frac{x_1}{x_2^2} \right)^{1/7} \tag{21}$$

can be used for the DLGSM iteration in the subsequent subinterval (or to repeat the current iteration, if $x_3 > \varepsilon$).

In case (ii), again from (20), we obtain that, for $j \geqslant 1$, $x_j \approx c_1 h^2 x_{j-1}$. Consequently, one has

$$x_3 \approx c_1^3 h^6 x_0 \approx \frac{x_1 x_2}{x_0}. \tag{22}$$

This allows to predict the new stepsize as

$$h_{\text{new}} = h\left(\varepsilon \frac{x_0}{x_1 x_2}\right)^{1/6}. \tag{23}$$

We observe that, if x_0 is small, one could have $x_j \leqslant \varepsilon$ for $j \leqslant 2$. In the case $j = 2$, the estimate (22) allows to predict the new stepsize by means of (23). Obviously, we must perform at least two iterations, in order to predict the new stepsize. This may be not possible when both c_1 and c_2 are close to zero, as in the case, for example, of pure quadrature problems (i.e., when $c_1 = c_2 = 0$). This matter will be discussed later. For the moment, we assume that at least two iterations are performed. In such a case, we obtain

$$x_3 \approx \frac{x_1^3}{x_0^2},$$

and, consequently, the new stepsize is predicted as

$$h_{\text{new}} = h\left(\varepsilon \frac{x_0^2}{x_1^3}\right)^{1/6}. \tag{24}$$

In the previous relations, by considering $\varepsilon = \text{tol} \cdot x_0$, one obtains a control of the relative growth of the error. Namely, the stepsize is predicted by imposing

$$\frac{x_3}{x_0} = \text{tol}. \tag{25}$$

3.3. Accuracy control

The most commonly used strategy for the stepsize selection is based on the control of the truncation error. In the case of the trapezoidal rule, it amounts to determine the stepsize in order to have

$$\|T_\tau\| \leqslant \widetilde{\text{tol}}, \tag{26}$$

where T_τ is the vector whose entries are the truncation errors at $t_0, \ldots, t_s$, and $\widetilde{\text{tol}}$ is the given tolerance. By using the estimate

$$\|T_\tau\| \approx \frac{s}{12} \max_i \left\|y^{(3)}(t_i)\right\| h^3,$$

we then obtain the new stepsize

$$\widetilde{h}_{\text{new}} = \left(\frac{12 \cdot \widetilde{\text{tol}}}{s \cdot \max_i \|y^{(3)}(t_i)\|}\right)^{1/3}. \tag{27}$$

Approximations for the values $y^{(3)}(t_i)$ are obtainable from componentwise second divided differences of the f_i, $i = 0, \ldots, s$, computed during the DLGSM iteration.

The previous estimate (27) should then be compared with the stepsize predicted by the convergence of DLGSM (see (25)). Since the convergence must be always guaranteed, it should be

$$h_{\text{new}} \leqslant \widetilde{h}_{\text{new}}, \tag{28}$$

for a suitable tolerance $\widetilde{\text{tol}}$. If this were always the case, the cost for computing $\widetilde{h}_{\text{new}}$ could be avoided. Unfortunately, there are some cases where the above condition is not satisfied. For example, when the function f is independent of y, or it is autonomous and linear, one obtains $x_1 = 0$, independently of the stepsize used, whereas there is a restriction on $\widetilde{h}_{\text{new}}$. It is then evident that, in such cases, the strategy (27) is more restrictive.

In the remaining part of this section, we shall study the cases where (28) holds true. Let us first suppose, for simplicity, that $f = f(y)$ and nonlinear. Then, from the previous analysis, (25) implies that

$$c_1^3 h^6 \approx \frac{x_3}{x_0} = \text{tol},$$

where, for h sufficiently small (see (16) and (17)),

$$c_1 \approx \left\| \left(A_s \otimes I_m - hB_s \otimes J_0\right)^{-1} \right\| \frac{1}{2} \| D_s \otimes J_0' \| \approx \frac{s^2 m}{2} \| J_0' \|.$$

It follows that

$$\frac{s^6 m^3}{8} h^6 \left\| \frac{\partial J_0}{\partial y} f_0 \right\|^3 \approx \text{tol},$$

that is,

$$h^3 \approx \frac{\sqrt{8}}{s^3 m^{3/2}} \left\| \frac{\partial J_0}{\partial y} f_0 \right\|^{-3/2} \sqrt{\text{tol}}. \tag{29}$$

On the other hand, (26) would approximately require

$$\frac{s}{12} h^3 \left\| \left(\frac{\partial J_0}{\partial y} f_0 \right) f_0 + J_0^2 f_0 \right\| \leqslant \widetilde{\text{tol}},$$

which, from (29), gives

$$\frac{\sqrt{2}}{6 \cdot s^2 m^{3/2}} \frac{\| ((\partial J_0/\partial y) f_0) f_0 + J_0^2 f_0 \|}{\| (\partial J_0/\partial y) f_0 \|^{3/2}} \sqrt{\text{tol}} \leqslant \widetilde{\text{tol}}.$$

It is then evident that the previous inequality is satisfied with $\widetilde{\text{tol}} \approx \sqrt{\text{tol}}$, provided that

$$\frac{\sqrt{2}}{6 \cdot s^2 m^{3/2}} \frac{\| ((\partial J_0/\partial y) f_0) f_0 + J_0^2 f_0 \|}{\| (\partial J_0/\partial y) f_0 \|^{3/2}} \tag{30}$$

is not too large.

It can be shown that the above result continues to hold when f is not autonomous, but with a "moderate" dependence on t. We then conclude that inequality (25) is sufficient for selecting the stepsize when problem (1) is autonomous (or "almost" autonomous), provided that (30) is not very large. In all other cases, the new stepsize will be that given by (27).

The above arguments can be summarized as follows:

(1) when both

$$\frac{\mathrm{d}f}{\mathrm{d}t}(t, y) \approx J(t, y) f(t, y), \qquad \frac{\| ((\partial J_0/\partial y) f_0) f_0 + J_0^2 f_0 \|}{\| (\partial J_0/\partial y) f_0 \|^{3/2}} \leqslant \nu_1, \tag{31}$$

hold true, where ν_1 is suitably chosen, we use the requirement (25) for the mesh selection;

(2) conversely, we use (26).

In practice, (31) is assumed to hold true when both

$$\frac{\|f_1 - f_0\|}{h} \leqslant 1.1\,\|J_0 f_0\|, \qquad \frac{\|f_0''\|}{(\|f_0'' - J_0^2 f_0\|/\|f_0\|)^{3/2}} \leqslant \nu_1, \tag{32}$$

are satisfied. Here f_0'' is the second derivative of f at (t_0, y_0). Conversely, (31) is assumed to be false. As a matter of fact, for the numerical tests in Section 5 the two above requirements are fulfilled at almost all grid points. Consequently, the mesh is essentially always selected through the convergence of DLGSM.

We observe that the checks (32) only require to know information at the initial point. This implies that, in the actual implementation of the above procedure on a parallel computer [3], the corresponding computational load can be done in parallel by a different processor. On the contrary, the computation of $\widetilde{h}_{\text{new}}$ needs to be done sequentially.

4. Convergence of the nonlinear iteration

Next theorem is a useful tool in studying the convergence of the iteration (11), used for solving

$$\boldsymbol{g}(\boldsymbol{y}) \equiv (A \otimes I_m)\boldsymbol{y} - (H \cdot B \otimes I_m)\boldsymbol{f} - \boldsymbol{\eta} = \boldsymbol{0}.$$

Theorem 2. *Assume that*

(a) $\boldsymbol{g}$ *is differentiable in* $\mathcal{D}_\alpha = \{\boldsymbol{x}\colon \|\boldsymbol{x} - \overline{\boldsymbol{y}}\| \leqslant \alpha\}$;

(b) $\boldsymbol{y}^{(0)} \in \mathcal{D}_\alpha$;

(c) $\|M^{-1}(\boldsymbol{g}'(\boldsymbol{y}) - \boldsymbol{g}'(\boldsymbol{x}))\boldsymbol{v}\| \leqslant \gamma\,\|\boldsymbol{y} - \boldsymbol{x}\|\,\|\boldsymbol{v}\|$, *for all* $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}_\alpha$ *and* $\boldsymbol{v}$ *such that* $\|\boldsymbol{y} - \boldsymbol{x} - \boldsymbol{v}\| \leqslant \alpha$;

(d) $\theta \equiv \frac{5}{2}\alpha\gamma < 1$.

Then the simplified Newton iteration (11) *converges to the solution* $\overline{\boldsymbol{y}}$ *at least linearly.*

Proof. We recall that the matrix M in (11) is nothing but $\boldsymbol{g}'(\boldsymbol{y}^{(0)})$. The thesis is then proved by showing that

$$\|\boldsymbol{y}^{(i)} - \overline{\boldsymbol{y}}\| \leqslant \theta^i \|\boldsymbol{y}^{(0)} - \overline{\boldsymbol{y}}\|.$$

For $i = 1$, in fact, we have

$$\begin{aligned}
\|\boldsymbol{y}^{(1)} - \overline{\boldsymbol{y}}\| &= \|\boldsymbol{y}^{(0)} - \overline{\boldsymbol{y}} - M^{-1}(\boldsymbol{g}(\boldsymbol{y}^{(0)}) - \boldsymbol{g}(\overline{\boldsymbol{y}}))\| \\
&= \|M^{-1}(\boldsymbol{g}'(\boldsymbol{y}^{(0)})(\boldsymbol{y}^{(0)} - \overline{\boldsymbol{y}}) - (\boldsymbol{g}(\boldsymbol{y}^{(0)}) - \boldsymbol{g}(\overline{\boldsymbol{y}})))\| \\
&\leqslant \int_0^1 \|M^{-1}(\boldsymbol{g}'(\boldsymbol{y}^{(0)}) - \boldsymbol{g}'(\overline{\boldsymbol{y}} + t(\boldsymbol{y}^{(0)} - \overline{\boldsymbol{y}})))(\boldsymbol{y}^{(0)} - \overline{\boldsymbol{y}})\|\,\mathrm{d}t \\
&\leqslant \int_0^1 \gamma(1-t)\|\boldsymbol{y}^{(0)} - \overline{\boldsymbol{y}}\|^2 \mathrm{d}t \leqslant \frac{1}{2}\alpha\gamma\|\boldsymbol{y}^{(0)} - \overline{\boldsymbol{y}}\| < \theta\|\boldsymbol{y}^{(0)} - \overline{\boldsymbol{y}}\|.
\end{aligned}$$

By induction, we now suppose true the result for i. Then, for $i + 1$, we get

$$\begin{aligned}
\|\boldsymbol{y}^{(i+1)} - \overline{\boldsymbol{y}}\| &= \|\boldsymbol{y}^{(i)} - \overline{\boldsymbol{y}} - M^{-1}(\boldsymbol{g}(\boldsymbol{y}^{(i)}) - \boldsymbol{g}(\overline{\boldsymbol{y}}))\| \\
&= \|M^{-1}(\boldsymbol{g}'(\boldsymbol{y}^{(0)})(\boldsymbol{y}^{(i)} - \overline{\boldsymbol{y}}) - (\boldsymbol{g}(\boldsymbol{y}^{(i)}) - \boldsymbol{g}(\overline{\boldsymbol{y}})))\|
\end{aligned}$$

$$
\begin{aligned}
&\leqslant \|M^{-1}(g'(y^{(0)}) - g'(y^{(i)}))(y^{(i)} - \overline{y})\| \\
&\quad + \|M^{-1}(g'(y^{(i)})(y^{(i)} - \overline{y}) - (g(y^{(i)}) - g(\overline{y}))\| \\
&\leqslant \gamma \|y^{(0)} - y^{(i)}\| \, \|y^{(i)} - \overline{y}\| \\
&\quad + \int_0^1 \|M^{-1}(g'(y^{(i)}) - g'(\overline{y} + t(y^{(i)} - \overline{y})))(y^{(i)} - \overline{y})\| \, \mathrm{d}t \\
&\leqslant \gamma (\|y^{(0)} - \overline{y}\| + \|y^{(i)} - \overline{y}\|) \|y^{(i)} - \overline{y}\| + \frac{1}{2}\gamma \|y^{(i)} - \overline{y}\|^2 \\
&< \frac{5}{2}\gamma \|y^{(0)} - \overline{y}\| \, \|y^{(i)} - \overline{y}\| \leqslant \theta \|y^{(i)} - \overline{y}\|. \qquad \square
\end{aligned}
$$

Remark 1. It is worth noting that, since all GAMs have similar stability properties, the conditioning of the matrix M is almost independent of the particular method considered in this class. Consequently, it is reasonable to expect that the above parameter θ does not vary sensibly for such methods. This fact is actually observed in practice. In fact, the simplified Newton iteration (11) requires approximately the same number of steps to have the same stopping criterion satisfied, independently of the method chosen.

The result of Theorem 2 is of practical interest, since we can get cheap estimates for the two parameters α and γ during the execution of DLGSM (see next section). If necessary, this allows to split the interval of integration into suitable "windows", where the convergence of the simplified Newton iteration (11) is assured in a given number of steps.

In other words, having fixed an upper bound $\theta_{\max}$ for the parameter θ, we use DLGSM until the estimate for θ is smaller than $\theta_{\max}$. As soon as this bound is exceeded, we stop DLGSM, thus determining the first "window", where the simplified Newton iteration (11) (i.e., the parallel section of the method) is carried out until convergence. After that, we start DLGSM again, thus repeating the above procedure until the whole interval of integration is covered.

4.1. Estimates for the parameters α and γ

In this section, we shall get estimates for both parameters α and γ defined in Theorem 2. They are obtained dynamically during the execution of DLGSM. In particular, we show that to each subinterval $[\tau_{i-1}, \tau_i]$ of the coarse mesh we can associate a couple of parameters (α_i, γ_i) which take into account of all the integration up to τ_i.

Suppose, for simplicity, that only one window is needed. We start considering the first parameter, which, in practice, is the measure of the global error up the current subinterval. In fact, by considering the vector $\widehat{y}$ whose (block) entries are the values of the continuous solution at the grid points, one has

$$\alpha = \|\overline{y} - y^{(0)}\| \approx \|\widehat{y} - y^{(0)}\|.$$

This because $\overline{y}$ is the solution of the discrete problem given by a higher order method than the trapezoidal rule, which provides us with the starting vector $y^{(0)}$.

Let δ_i be the vector whose entries are the errors over the ith subinterval. It is then not difficult to realize that, at first order approximation, it satisfies the equation (see (8) and (12))

$$\left(A_{i\cdot s}\otimes I_m-(H_iB_{i\cdot s}\otimes I_m)\boldsymbol{J}_i^{(0)}\right)\begin{pmatrix}0\\ \delta_1\\ \vdots\\ \delta_i\end{pmatrix}=\begin{pmatrix}0\\ T_1\\ \vdots\\ T_i\end{pmatrix}, \tag{33}$$

where $\boldsymbol{J}_i^{(0)}$ is the block diagonal matrix with the Jacobians of f at $(t_0, y_0^{(0)}), \ldots, (t_{si}, y_{si}^{(0)})$, and the vectors $\{T_r\}$ contain the truncation errors over the corresponding subintervals of the coarse mesh. Consequently, by considering that (33) is a lower block bidiagonal linear system, one obtains the following recurrence (hereafter, see (7)), let J_{jr} be the Jacobian of f at $(t_{jr}, y_{jr}^{(0)})$, $J_{0r}\equiv J_{s,r-1}$, $r>1$, and $J_{01}\equiv J_0$,

$$\begin{pmatrix}\delta_{s,r-1}\\ \delta_r\end{pmatrix}=\left(A_s\otimes I_m-h_rB_s\otimes I_m\begin{pmatrix}J_{0r}&&\\ &\ddots&\\ &&J_{sr}\end{pmatrix}\right)^{-1}\begin{pmatrix}\delta_{s,r-1}\\ T_r\end{pmatrix},\quad r=1,2,\ldots,i, \tag{34}$$

where $\delta_{s,r-1}$ is the last block entry of δ_{r-1}, and $\delta_{s0}=0$. The vectors $\{\delta_r\}$ are then obtained, provided that estimates for the truncation errors $\{T_r\}$ are available. This can be done, as previously said, by means of componentwise second divided difference of the function f. Alternatively, a *deferred correction approach* as described in [5,9] may be used.

In practice, in order to reduce the computational effort, the above recurrence (34) is approximated by

$$\begin{pmatrix}\delta_{s,r-1}\\ \delta_r\end{pmatrix}=(A_s\otimes I_m-h_rB_s\otimes J_{0r})^{-1}\begin{pmatrix}\delta_{s,r-1}\\ T_r\end{pmatrix},\quad r=1,2,\ldots,i.$$

The coefficient matrix turns out to be block Toeplitz lower bidiagonal, whose diagonal block, $(I_m-\frac{1}{2}h_rJ_{0r})$, has already been factored during the DLGSM iteration.

By using hereafter the infinity norm, we then conclude that to the ith subinterval there corresponds a parameter

$$\alpha_i=\max\{\alpha_{i-1},\ \|\delta_i\|\},\quad i\geqslant 1,\qquad \alpha_0=0.$$

A similar approach can be used for the parameter γ. In fact, one verifies that

$$\boldsymbol{g}'(\boldsymbol{y})-\boldsymbol{g}'(\boldsymbol{x})=(H\cdot B\otimes I_m)\big(\boldsymbol{J}(\boldsymbol{x})-\boldsymbol{J}(\boldsymbol{y})\big)\approx(H\otimes I_m)\big(\boldsymbol{J}(\boldsymbol{x})-\boldsymbol{J}(\boldsymbol{y})\big), \tag{35}$$

where $\boldsymbol{J}(\boldsymbol{x})$ and $\boldsymbol{J}(\boldsymbol{y})$ are the block diagonal matrices, whose diagonal blocks are the Jacobians of the function f evaluated at the entries of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. The last approximation follows from the fact that the matrix B (see (9), (10)) has unit row sums.

We shall then obtain the estimate for γ by fixing a particular direction $\boldsymbol{x}-\boldsymbol{y}$, and a vector $\boldsymbol{v}$ of unit norm. In practice, in the ith block we assume that the Jacobian matrices all coincide with J_{0i}, which corresponds to the initial condition, y_{0i} (i.e., $y_{s,i-1}$), for the current block. Then, after we have carried out the DLGSM iterations, in order to start the procedure in the subsequent block, we need to compute $J_{0,i+1}\equiv J_{si}$, namely the Jacobian at $y_{0,i+1}\equiv y_{si}$. As a consequence, by setting $E_s=(1,\ldots,1)^{\mathrm{T}}\in\mathbb{R}^s$, we choose

$$\boldsymbol{z}=\begin{pmatrix}0\\ z_1\\ \vdots\\ z_p\end{pmatrix}\equiv\big(\boldsymbol{J}(\boldsymbol{x})-\boldsymbol{J}(\boldsymbol{y})\big)\boldsymbol{v}=\begin{pmatrix}0\\ E_s\otimes(J_{01}-J_{s1})v_1\\ \vdots\\ E_s\otimes(J_{0p}-J_{sp})v_p\end{pmatrix},$$

where the $\{v_i\}$ are the block entries of $\boldsymbol{v}$ (let $y_{01} \equiv y_0$), defined as

$$v_0 = 0, \qquad v_i = \frac{y_{0i} - y_{si}}{\|y_{0i} - y_{si}\|}, \quad i = 1, \ldots, p.$$

Consequently, we determine the approximation for γ such that

$$\gamma \geqslant \frac{\|\boldsymbol{w}\|}{\|\boldsymbol{u}\|},$$

where (see (35))

$$\boldsymbol{w} \equiv M^{-1}(H \otimes I_m)z \approx M^{-1}\big(g'(y) - g'(x)\big)\boldsymbol{v},$$

and

$$\boldsymbol{u} = (0,\ u_1,\ \ldots, u_p)^{\mathrm{T}}, \qquad u_i = E_s \otimes (y_{0i} - y_{si}), \quad i = 1, \ldots, p.$$

By using as an approximation for M the matrix computed for the DLGSM iteration, we then obtain the following recurrence for the entries of the vector $\boldsymbol{w}$:

$$\begin{pmatrix} w_{s,r-1} \\ w_r \end{pmatrix} = \big(A_s \otimes I_m - h_r B_s \otimes J_{0r}\big)^{-1} \begin{pmatrix} w_{s,r-1} \\ h_r z_r \end{pmatrix}, \quad r = 1, 2, \ldots, i.$$

Consequently, we choose the following approximation to the parameter γ up to the ith subinterval:

$$\gamma_i = \begin{cases} \max\left\{\gamma_{i-1}, \dfrac{\|w_i\|}{\|y_{s,i-1} - y_{si}\|}\right\}, & \text{if } \|y_{s,i-1} - y_{si}\| > 0, \\ \gamma_{i-1}, & \text{otherwise,} \end{cases} \qquad i \geqslant 1,$$

having set $\gamma_0 = 0$.

We then conclude that, during the DLGSM iteration over the ith subinterval in the coarse mesh, we can obtain cheap estimates for α_i and γ_i and, therefore, for the parameter θ_i which determines the speed of convergence of the outer iteration (11). Consequently, when such estimate is larger than a fixed $\theta_{\max}$, we end the sequential procedure based on the DLGSM. Namely, we have determined a *window* where we perform the simplified Newton process (11). As previously said, the whole procedure is then repeated until the entire interval of integration $[t_0, T]$ is covered.

5. Numerical examples

In this section we report some numerical tests obtained by using a Matlab prototype implementing the procedure previously described. In all cases, we use the 9th order GAM (GAM9), after the DLGSM iteration. The tolerance used for the DLGSM iteration is $\mathrm{tol} = 10^{-6}$, with tolerance $\mathrm{tol} = 10^{-3}$ in case of switch to the local error control. Moreover, the value $\theta_{\max}$ is chosen in order to have convergence for the outer iteration (11) in at most four iteration. The stopping criterion for the Newton iteration is

$$\max_j \frac{|y_j^{(i+1)} - y_j^{(i)}|}{1 + |y_j^{(i+1)}|} \leqslant 10^{-9},$$

where $y_j^{(i)}$ is the jth entry of $\boldsymbol{y}^{(i)}$. For comparison, we also report the results on the same problems by using the stiff solver ode23s, taken from the Matlab ODE Suite [15]. In order to have a comparable

Table 1
Numerical results

Number	Robertson		van der Pol		Hires	
of	GAM9	ode23s	GAM9	ode23s	GAM9	ode23s
windows	3	–	9	–	1	–
points	1010	7883	1851	86541	488	18210
fev	2530	23654	6280	259628	1560	54629
Jev	1010	7883	1851	86541	488	18210
flops	3.4×10^6	2.2×10^6	2.7×10^6	14.3×10^6	14.1×10^6	22.4×10^6

accuracy for both solvers, we have used ode23s with the following parameters: atol = rtol = 10^{-11}, and providing analytically the Jacobian. Such parameters have been obtained by comparing the solutions computed for two of the following problems at selected grid points, where the values of the exact solutions are known. For both solvers we report the number of mesh points, function and Jacobian evaluations, and the number of Matlab flops. For the prototype of the parallel solver we also report the number of windows required to cover the integration interval. Finally, we want to mention that the flops count for GAM9 includes some extra work required for getting an approximation to the global error.

The first problem we consider are the Robertson equations

$$
\begin{aligned}
&y_1' = -0.04y_1 + 10^4 y_2 y_3,\\
&y_2' = 0.04y_1 - 10^4 y_2 y_3 - 3 \times 10^7 y_2^2, \quad t \in [0, 10^{15}],\\
&y_3' = 3 \times 10^7 y_2^2,\\
&y_1(0) = 1, \qquad y_2(0) = y_3(0) = 0.
\end{aligned}
$$

The second problem is given by the van der Pol equations

$$
\begin{aligned}
&y_1' = y_2,\\
&y_2' = -y_1 + \mu y_2(1 - y_1^2), \quad t \in [0, \mu],\\
&y_1(0) = 2, \qquad y_2(0) = 0,
\end{aligned}
$$

where $\mu = 10^6$.

Finally, the third problem is the Hires problem, a stiff ODE of dimension 8, taken from [14]. In Table 1 we report the obtained results, as described above. It is worth mentioning that for the van der Pol problem, whose solution has a huge spike near $t = 806853$, more than one window is needed to cover the interval $[0, 10^6]$ (actually 9 of them), with the first window ending at about $t = 806009$. This confirms the reliability of the proposed procedure for detecting the intervals where the Newton iteration does converge.

References

[1] P. Amodio and L. Brugnano, Parallel implementation of block boundary value methods for ODEs, *J. Comput. Appl. Math.* 78 (1997) 197–211.

[2] P. Amodio and L. Brugnano, Parallel ODE solvers based on block BVMs, *Adv. Comput. Math.* 7 (1–2) (1997) 5–26.
[3] P. Amodio and L. Brugnano, ParalleloGAM: a parallel code for ODEs, *Appl. Numer. Math.* 28 (1998) 95–106.
[4] P. Amodio and F. Mazzia, A boundary value approach to the numerical solution of initial value problems by multistep methods, *J. Differential Equations Appl.* 1 (1995) 353–367.
[5] L. Brugnano, Boundary value methods for the numerical approximation of ordinary differential equations, in: *Lecture Notes in Computer Science* 1196 (1997) 78–89.
[6] L. Brugnano and D. Trigiante, Convergence and stability of boundary value methods for ordinary differential equations, *J. Comput. Appl. Math.* 66 (1996) 97–109.
[7] L. Brugnano and D. Trigiante, Boundary value methods: the third way between linear multistep and Runge–Kutta methods, *Comput. Math. Appl.* (to appear).
[8] L. Brugnano and D. Trigiante, Block boundary value methods for linear Hamiltonian systems, *Appl. Math. Comput.* 81 (1997) 49–68.
[9] L. Brugnano and D. Trigiante, *Solving Differential Problems by Multistep Initial and Boundary Value Methods* (Gordon & Breach, Amsterdam, 1998).
[10] L. Brugnano and D. Trigiante, On the potentiality of sequential and parallel codes based on extended trapezoidal rules (ETRs), *Appl. Numer. Math.* 25 (1997) 169–184.
[11] K. Burrage, *Parallel and Sequential Methods for Ordinary Differential Equations* (Clarendon Press, Oxford, 1995).
[12] F. Iavernaro and F. Mazzia, Solving ordinary differential equations by generalized Adams methods: properties and implementation techniques, submitted.
[13] V. Lakshmikantham and D. Trigiante, *Theory of Difference Equations: Numerical Methods and Applications* (Academic Press, San Diego, CA, 1988).
[14] W.M. Lioen, J.J.B. de Swart and W.A. van der Veen, Test set for IVP solvers, CWI Report NM-R9615 (August 1996).
[15] L.F. Shampine and M.W. Reichelt, The Matlab ODE Suite, Report (1995).