



ELSEVIER

Applied Numerical Mathematics 25 (1997) 169–184



APPLIED
NUMERICAL
MATHEMATICS

On the potentiality of sequential and parallel codes based on extended trapezoidal rules (ETRs)[☆]

Luigi Brugnano^{a,*}, Donato Trigiante^b

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

^b *Dipartimento di Energetica "S. Stecco", Via Lombroso 6/17, 50134 Firenze, Italy*

Abstract

The Boundary Value Methods (BVMs) is a class of numerical methods for solving ODEs proposed and analyzed in the last few years. They are based on Linear Multistep Formulae (LMF) and do not suffer from the theoretical order limitations due to the Dahlquist barriers. In previous papers some families of BVMs have been proposed and studied. In this paper we exploit the possibility of using the family of Extended Trapezoidal Rules (ETRs) to construct both a sequential and a parallel code. Such methods are used in a block form which improves their flexibility, even though in this form some stability problems arise. The potentiality of the resulting codes are shown through comparison on some test problems taken from the literature. © 1997 Elsevier Science B.V.

Keywords: Boundary Value Methods; Numerical solution of ODEs; Parallel methods for ODEs

1. Introduction

In the last four years, the class of numerical methods for ODEs called *Boundary Value Methods* (BVMs) has been studied by the research group of the authors. Such methods are based on LMF, although used in an unusual way, which permits, for example, to overcome the two well-known Dahlquist barriers.

The class is a very wide one and contains methods suitable for all known specific problems arising in the applications (e.g., dissipative, Hamiltonian, method of lines, etc.). In previous papers some families of BVMs have been proposed such as GBDF [8], GAMs [13], ETRs [4], ETR_{2s} [6], unsymmetric ETR_{2s} [7] and TOMs [1]. In particular, the family of TOMs contains stable methods of highest possible order, i.e., $2k$ for k -step methods. A more global analysis which takes into account the implementation difficulties suggests, however, the use of ETRs as the most promising for the

[☆] Work supported by CNR, Contract No. 96.00243.CT01.

* Corresponding author.

construction of a general purpose code. We also mention that ETRs can be also used to approximate the solution of continuous BVPs [9].

In order to improve the flexibility of the methods chosen, a block version of them seems to be more appropriate. Such block form, previously introduced for Hamiltonian problems [10], is here discussed in detail. In particular, we briefly review the main facts concerning this implementation of the methods and we study some stability problems, due to the specific block form, not yet analyzed so far.

At the present, a sequential version of the code for IVPs [18] based on ETRs is almost completed. Such code compares with the most efficient existing codes, such as RADAU5. For BVPs there exists a Matlab prototype, working on linear problems, which implements a new mesh selection strategy [11,12,14]. This prototype has been extensively tested [20] on several test problems taken from the literature, most of them in the class of singular perturbation problems. Finally, a parallel prototype using ETRs is also ready.

In order to briefly describe BVMs, let us consider the continuous IVP

$$y' = f(t, y), \quad t \in [t_0, T], \quad y(t_0) = \eta, \quad (1)$$

to be approximated by means of a linear k -step formula,

$$\sum_{i=0}^k \alpha_i y_{n+i} = h \sum_{i=0}^k \beta_i f_{n+i}, \quad h = (T - t_0)/N. \quad (2)$$

Eq. (2) needs k independent conditions to be imposed, in order to get the discrete solution. The BVM approach essentially consists in relaxing the usual practice of assigning the first k values of the discrete solution, thus replacing the continuous IVP (1) by a discrete IVP. Let then k_1 , k_2 be two natural numbers, $k_1 \geq 1$, $k_1 + k_2 = k$. The discrete problem is defined by fixing the first k_1 values of the discrete solution, y_0, \dots, y_{k_1-1} , and the last k_2 ones, y_{N-k_2+1}, \dots, y_N . That is, the continuous IVP is approximated by means of a discrete BVP. This defines a BVM with (k_1, k_2) -boundary conditions. We observe that the usual way of using LMF corresponds to $k_1 = k$ and $k_2 = 0$.

The idea has been considered sometimes in the past years (see, for example, [5,16]). Nevertheless, it has been deeply studied only recently, by generalizing the usual stability notions [8,14]. Considering, in fact, the linear stability, one has that stability requirements for a discrete BVP differ from those for an IVP. As an example, the former requires the characteristic polynomial to have part of the roots inside the unit circle and part outside it. In particular, the number of initial conditions, k_1 , needs to be equal to the number of roots inside the unit circle and the number of final conditions k_2 must be equal to the number of roots of modulus greater than one. Of course, when $k_2 = 0$, one returns to the classical situation where the characteristic polynomial is required to be a Schur polynomial. The stability notions have been then changed accordingly. In particular, A-stability corresponds to the situation where the stability polynomial has exactly k_1 roots inside the unit circle and k_2 roots outside, for all $h\lambda \in \mathbb{C}^-$. For convenience, in order to distinguish between the stability for BVPs and that for IVPs, the notion of A_{k_1, k_2} -stability has been introduced [8,14].

2. Extended Trapezoidal Rules (ETRs)

Let us consider the k -step methods having the form

$$y_n - y_{n-1} = h \sum_{i=-\nu}^{\nu-1} \beta_{\nu+i} f_{n+i}, \quad i = \nu, \dots, N - \nu + 1, \quad (3)$$

where $k = 2\nu - 1$, and the coefficients are uniquely determined by imposing the formula to have the maximum possible order $k + 1$. When $\nu = 1$, we get the trapezoidal rule, from which the name of the methods derives [4]. For each value of ν , the methods turn out to be $A_{\nu, \nu-1}$ -stable. In particular, the boundary of the stability region coincides with the imaginary axis (perfect A-stability). That is, the stability properties are similar to those of the basic trapezoidal rule. The methods must be used with $(\nu, \nu - 1)$ -boundary conditions, namely the values

$$y_0, y_1, \dots, y_{\nu-1}, y_{N-\nu+2}, \dots, y_N$$

should in principle be provided. Obviously, of such values only $y_0 = \eta$ is inherited from the continuous problem (1), while the remaining ones need to be found. In order to avoid the latter undesirable request, such values are treated as unknowns. This is done by introducing an appropriate set of equations independent of those provided by the *main method* (3). These equations are conveniently derived by the following set of *initial additional methods*,

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

and *final additional ones*,

$$y_j - y_{j-1} = h \sum_{i=0}^k \beta_{k-i}^{(j)} f_{N-i}, \quad j = N - \nu + 2, \dots, N. \quad (5)$$

The coefficients $\{\beta_i^{(j)}\}$ in each of the above formulae are uniquely determined by imposing the same order $k + 1$ of the main method (3).

Consequently, the discrete problem only needs one condition to be imposed, which is the one provided by the continuous problem. For initial value problems, this condition is the initial condition, although it can be replaced by more general ones, such as boundary or multipoint conditions, in case of continuous boundary value problems [9,14].

The properties of the *composite method* (3)–(5) are better described by introducing the matrices

$$[\mathbf{a}_N \mid A_N] = \left(\begin{array}{c|cccc} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{array} \right)_{N \times (N+1)} \quad (6)$$

and

$$[\mathbf{b}_N \mid B_N] = \left(\begin{array}{c|cccc} \beta_0^{(1)} & \beta_1^{(1)} & \cdots & \beta_k^{(1)} \\ \vdots & \vdots & & \vdots \\ \beta_0^{(\nu-1)} & \beta_1^{(\nu-1)} & \cdots & \beta_k^{(\nu-1)} \\ \beta_0 & \beta_1 & \cdots & \beta_k \\ & \ddots & \ddots & \ddots \\ & & \beta_0 & \beta_1 & \cdots & \beta_k \\ & & \beta_0^{(N-\nu+2)} & \beta_1^{(N-\nu+2)} & \cdots & \beta_k^{(N-\nu+2)} \\ & & \vdots & \vdots & & \vdots \\ & & \beta_0^{(N)} & \beta_1^{(N)} & \cdots & \beta_k^{(N)} \end{array} \right)_{N \times (N+1)} \quad (7)$$

The discrete problem is then given by

$$(A_N \otimes I_m)\mathbf{y} - h(B_N \otimes I_m)\mathbf{f} = -(\mathbf{a}_N \otimes \eta - h\mathbf{b}_N \otimes f(\eta)), \quad (8)$$

where

$$\mathbf{y} = (y_1, \dots, y_N)^T, \quad \mathbf{f} = (f_1, \dots, f_N)^T, \quad f_j = f(t_j, y_j).$$

In this form, the notions of linear stability (e.g., A-stability) can be derived directly by posing as usual $f(t, y) = \lambda y$. This will be discussed in Section 3. By the way, we note that, although the notion of $A_{k_1 k_2}$ -stability will not be used anymore, nevertheless it has had a central role in the derivation of the main methods.

Among the properties of the composite method (3)–(5), we quote the following result, which will be used later. Let consider the application of the method to the linear Hamiltonian problem

$$y' = Ly, \quad t \in [t_0, T], \quad y(t_0) = y_0,$$

where $L \in \mathbb{R}^{2m \times 2m}$ is a Hamiltonian matrix, namely

$$L = \begin{pmatrix} & -I_m \\ I_m & \end{pmatrix} S, \quad S = S^T.$$

It is known that for any matrix C satisfying

$$L^T C + CL = O, \quad (9)$$

the function

$$V(t, C) = y^T C y$$

is invariant, i.e., $V'(t, C) \equiv 0$. A similar result can be proved for the discrete problem.

Theorem 2.1. *Let C satisfy (9). Then*

$$y_0^T C y_0 = y_N^T C y_N,$$

independently of the stepsize h .

Proof. See [10,14]. \square

That is, the continuous invariants are exactly preserved at the initial point and at the last point of the discrete solution. In the remaining points it differs for $O(h^{k+1})$ (the order of the method) [6].

3. Block version

The result of the previous Theorem 2.1 suggests a different implementation of ETRs (and, in general, of BVMs), which increases the flexibility of the methods. The idea is to discretize the integration interval by using two meshes:

- a *coarse mesh*, containing the points

$$\tau_j = \tau_{j-1} + \widehat{h}_j, \quad j = 1, \dots, p, \quad \tau_0 = t_0, \quad \tau_p = T; \tag{10}$$

- a *fine mesh*, obtained by applying on each subinterval $[\tau_{j-1}, \tau_j]$ the composite method, e.g., (3)–(5), with stepsize $h_j = \widehat{h}_j/s$, where $N \equiv s$ is kept fixed.

Therefore we can decrease the stepsize by increasing the number p of subintervals in the coarse mesh, while s is kept fixed. The above approach defines a *block BVM* (B_2VM).

This block implementation of BVMs has relevant consequences in practice. For example, it allows a simple stepsize variation, by changing only the stepsizes in the coarse mesh. Moreover, for ETRs applied to linear Hamiltonian problems, by considering that $y_{j \cdot s}$ is the approximation to $y(\tau_j)$, $j = 0, \dots, p$, from Theorem 2.1 we obtain that

$$y_0^T C y_0 = y_{j \cdot s}^T C y_{j \cdot s}, \quad j = 1, \dots, p,$$

for all matrices C satisfying (9). That is, the continuous invariants are exactly preserved at the points of the coarse mesh. Finally, B_2VM s can be efficiently implemented on parallel computers, as described in Section 4.

We now briefly discuss the stability of the block version of ETRs. The application of the B_2VM derived from (3)–(5) over the j th subinterval of the coarse mesh, $j = 1, \dots, p$, can then be written as (see (8) with $N \equiv s$)

$$(A_s \otimes I_m) \mathbf{y}_j - h_j (B_s \otimes I_m) \mathbf{f}_j = -(\mathbf{a}_s \otimes y_{(j-1) \cdot s} - h_j \mathbf{b}_s \otimes f_{(j-1) \cdot s}), \tag{11}$$

where

$$\mathbf{y}_j = (y_{(j-1) \cdot s+1}, \dots, y_{j \cdot s})^T, \quad \mathbf{f}_j = (f_{(j-1) \cdot s+1}, \dots, f_{j \cdot s})^T \tag{12}$$

are the vectors containing the approximations over the considered subinterval and the vectors $y_{(j-1) \cdot s}$, $f_{(j-1) \cdot s}$ are inherited from the previous subinterval. It is obvious that, since both the main method and the additional methods have k -steps, one must assume $s \geq k$. Suppose now to apply the method to the test equation,

$$y' = \lambda y, \quad \text{Re}(\lambda) \leq 0,$$

with stepsize h . Then, by setting as usual $q = h\lambda$, the discrete problem (11) over the first subinterval becomes

$$(A_s - qB_s) \mathbf{y}_1 = -(\mathbf{a}_s - q\mathbf{b}_s) y_0.$$

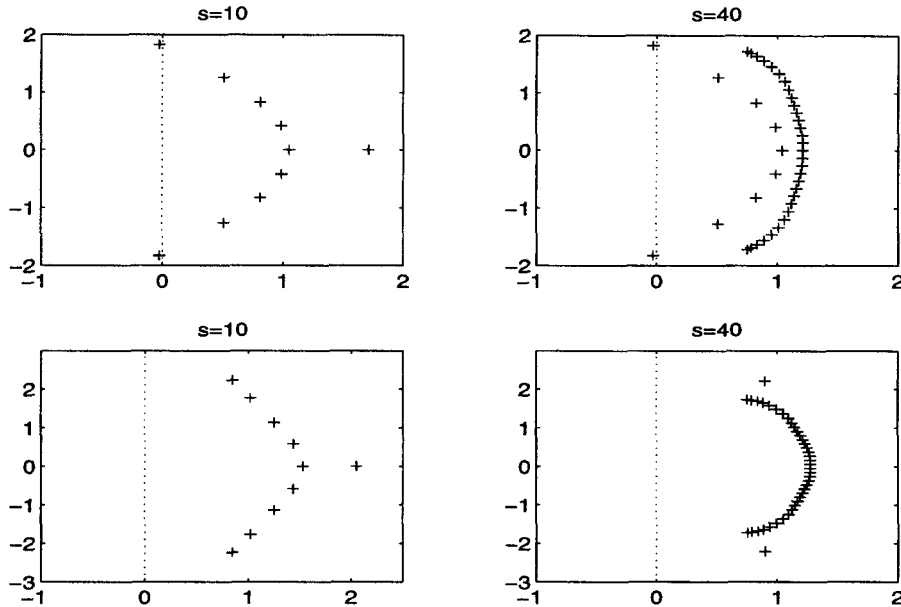


Fig. 1. Eigenvalues of the pencil (13) corresponding to the tenth order ETR, $s = 10, 40$, before (plots 1 and 2) and after (plots 3 and 4) the introduction of the auxiliary points ($r = 2$).

As a consequence, the discrete solution is defined for all $q \in \mathbb{C}^-$ iff the matrix pencil

$$(A_s - qB_s) \tag{13}$$

has all the eigenvalues with positive real part. This is in fact the case for B_2 VMs derived from ETRs of order 2, 4, 6 and 8. However, for $k \geq 9$ some eigenvalues of the pencil (13) may enter \mathbb{C}^- , as shown in the first and second plots of Fig. 1 for the tenth order ETR, $s = 10, 40$. It is not difficult to slightly modify the methods in order to avoid this problem. A possible modification consists in taking the points inside each block not equally spaced. We choose such points, i.e.,

$$t_0, t_1, \dots, t_r, t_{r+1}, \dots, t_{s-r-1}, t_{s-r}, \dots, t_{s-1}, t_s, \tag{14}$$

so that

$$t_0, t_r, \dots, t_{s-r}, t_s$$

are still equally spaced with stepsize h , while the remaining points are defined as

$$t_i - t_{i-1} = t_{s-i+1} - t_{s-i} = \gamma^{r+1-i}h, \quad i = 1, \dots, r, \tag{15}$$

where γ is a positive parameter such that $\sum_{i=1}^r \gamma^i = 1$. The points (15) will be called *auxiliary points*. The choice (15) implies that the points (14) are symmetrically distributed in the interval $[t_0, t_s]$. As a consequence, it can be shown that the properties of the methods on Hamiltonian problems continue to hold (see [14]).

When $r = 1$ then $\gamma = 1$ and no auxiliary point is introduced. The value $r = 2$ is sufficient to avoid eigenvalues with nonnegative real part for the B_2 VM based on the tenth order ETR, as shown in the third and fourth plots of Fig. 1.

The above choice of the points (14) is able to guarantee A-stability for B_2 VMs derived from ETRs, as stated by the next result.

Theorem 3.1. *Let the method (11) be derived from ETRs and suppose that the corresponding pencil (13) has all the eigenvalues with positive real part. The method is then A-stable.*

Proof. Suppose, in fact, to apply the method to the test equation

$$z' = \alpha iz, \quad z(t_0) = 1, \quad (16)$$

where $\alpha \in \mathbb{R}$ and i is the imaginary unit. Since the pencil (13) has all the eigenvalues with positive real part, from the maximum-modulus theorem, it follows that the thesis holds true iff $|z_{j,s}| \leq 1$, $j = 1, \dots, p$, independently of α and of the stepsize h used. Let us prove the thesis for $j = 1$, since the result can be readily generalized by induction. For this purpose, let us denote

$$z = \xi + i\zeta, \quad \xi, \zeta \in \mathbb{R}.$$

Consequently, Eq. (16) can be recast in the equivalent real form

$$y' = \alpha \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} y, \quad y(t_0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad y = \begin{pmatrix} \xi \\ \zeta \end{pmatrix}. \quad (17)$$

It is easily verified that problem (17) is Hamiltonian, with Hamiltonian function

$$y^T y = \xi^2 + \zeta^2 = |z|^2.$$

For B_2 VMs based on ETRs, Theorem 2.1 applies, thus giving ($N \equiv s$)

$$|z_s|^2 = y_s^T y_s = y_0^T y_0 = |z_0|^2 = 1,$$

for all values of α and for any stepsize h . \square

Remark 3.1. The previous result can be generalized to B_2 VMs based on *symmetric schemes*, as defined in [10,14]. In particular, ETRs belong to this wider family of methods.

4. Parallel implementation

The purpose of this section is to present the approaches followed for deriving an efficient parallel implementation of ETRs in both cases of IVPs and BVPs. A detailed presentation would be too long to be contained in this paper, so we confine ourselves to sketch the main ideas (see [2,3,14,15]). In order to simplify the presentation, we consider a linear autonomous problem of size m ,

$$y' = Ly + g(t), \quad t \in [t_0, T], \quad y(t_0) = \eta, \quad (18)$$

to describe the parallel algorithm.

Supposing that the coarse mesh (10) is fixed, the use of a B_2 VM over the entire interval $[t_0, T]$ leads to the following lower block bidiagonal linear system:

$$M^{(p)} \mathbf{y}^{(p)} = \mathbf{g}^{(p)}, \quad (19)$$

where

$$M^{(p)} = \begin{pmatrix} I_m & & & & & \\ \mathbf{v}_1 & M_1 & & & & \\ & V_2 & M_2 & & & \\ & & \ddots & \ddots & & \\ & & & V_p & M_p & \end{pmatrix}, \quad \mathbf{y}^{(p)} = \begin{pmatrix} y_0 \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_p \end{pmatrix}, \quad \mathbf{g}^{(p)} = \begin{pmatrix} \eta \\ h_1 \mathbf{g}_1 \\ h_2 \mathbf{g}_2 \\ \vdots \\ h_p \mathbf{g}_p \end{pmatrix},$$

the block vector \mathbf{y}_j (see (12)) contains the discrete approximation to the solution over the j th subinterval of the coarse mesh, and \mathbf{g}_j is a block vector containing suitable combinations of the inhomogeneity in (18) at the points in the same subinterval. The block matrices M_j are defined as (see (6)–(7))

$$M_j = A_s \otimes I_m - h_j B_s \otimes L, \quad j = 1, \dots, p.$$

Finally,

$$V_j = [O \mid \mathbf{v}_j], \quad j = 2, \dots, p,$$

where O is the $sm \times (s - 1)m$ zero matrix and

$$\mathbf{v}_j = \mathbf{a}_s \otimes I_m - h_j \mathbf{b}_s \otimes L, \quad j = 1, \dots, p.$$

The parallel algorithm, working on p parallel processors, then consists in assigning the subproblem over the j th subinterval to the j th processor, $j = 1, \dots, p$. This is done by considering the following factorization of the coefficient matrix:

$$M^{(p)} = D_1^{(p)} D_2^{(p)}, \tag{20}$$

where

$$D_1^{(p)} = \begin{pmatrix} I_m & & & & & \\ & M_1 & & & & \\ & & \ddots & & & \\ & & & & & \\ & & & & & M_p \end{pmatrix} \quad \text{and} \quad D_2^{(p)} = \begin{pmatrix} I_m & & & & & \\ \mathbf{w}_1 & \hat{I}_s & & & & \\ & W_2 & \hat{I}_s & & & \\ & & \ddots & \ddots & & \\ & & & & & W_p & \hat{I}_s \end{pmatrix}.$$

Hereafter, for any integer r we set $\hat{I}_r = I_r \otimes I_m$, and for all allowed j , $W_j = [O \mid \mathbf{w}_j]$, where \mathbf{w}_j is the solution of the linear system

$$M_j \mathbf{w}_j = \mathbf{v}_j, \quad j = 1, \dots, p.$$

Of course, each block M_j , which represents the application of the method over the subinterval $[\tau_{j-1}, \tau_j]$ of the coarse mesh, must be nonsingular. When L has no eigenvalues with positive real part, the previous Theorem 3.1 implies that each block M_j is nonsingular, whatever is the stepsize h_j used.

The solution of Eq. (19) is then obtained by first solving the block diagonal system

$$D_1^{(p)} \mathbf{x}^{(p)} = \mathbf{g}^{(p)}, \tag{21}$$

and then the block lower bidiagonal one

$$D_2^{(p)} \mathbf{y}^{(p)} = \mathbf{x}^{(p)}. \tag{22}$$

For convenience, we partition the auxiliary vector $\mathbf{x}^{(p)}$ as done for $\mathbf{y}^{(p)}$,

$$\mathbf{x}^{(p)} = (x_0^T, \mathbf{x}_1^T, \dots, \mathbf{x}_p^T)^T, \quad x_0 \in \mathbb{R}^m, \quad \mathbf{x}_j \in \mathbb{R}^{sm}.$$

Owing to its block diagonal structure, the solution of Eq. (21) can easily be obtained in parallel on p processors. This is not the case for Eq. (22), even though some parallelism is still present. To make this evident, we consider a further partition of the block vectors \mathbf{x}_j , \mathbf{y}_j and \mathbf{w}_j ,

$$\mathbf{x}_j = \begin{pmatrix} \hat{\mathbf{x}}_j \\ x_{j,s} \end{pmatrix}, \quad \mathbf{y}_j = \begin{pmatrix} \hat{\mathbf{y}}_j \\ y_{j,s} \end{pmatrix}, \quad \mathbf{w}_j = \begin{pmatrix} \hat{\mathbf{w}}_j \\ w_{j,s} \end{pmatrix}, \quad j = 1, \dots, p,$$

where $x_{j,s}, y_{j,s} \in \mathbb{R}^m$ and $w_{j,s} \in \mathbb{R}^{m \times m}$. Moreover, we define the $(ps + 1)m \times (ps + 1)m$ permutation matrix Q such that

$$Q\mathbf{x}^{(p)} = (\hat{\mathbf{x}}_1^T, \hat{\mathbf{x}}_2^T, \dots, \hat{\mathbf{x}}_p^T, x_0^T, x_{1,s}^T, \dots, x_{p,s}^T)^T.$$

Then, we consider the permuted system

$$QD_2^{(p)}Q^TQ\mathbf{y}^{(p)} = Q\mathbf{x}^{(p)},$$

which is a 2×2 upper block triangular system (let O_m be the $m \times m$ zero matrix),

$$\left(\begin{array}{ccc|ccc} \hat{I}_{s-1} & & & \hat{\mathbf{w}}_1 & O_m & \\ & \hat{I}_{s-1} & & \hat{\mathbf{w}}_2 & O_m & \\ & & \ddots & & \ddots & \ddots \\ & & & \hat{I}_{s-1} & & \hat{\mathbf{w}}_p & O_m \\ \hline & & & I_m & & & \\ & & & w_{1,s} & I_m & & \\ & & & & w_{2,s} & I_m & \\ & & & & & \ddots & \ddots \\ & O & & & & & w_{p,s} & I_m \end{array} \right) \begin{pmatrix} \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \\ \vdots \\ \hat{\mathbf{y}}_p \\ y_0 \\ y_{1,s} \\ \vdots \\ y_{p,s} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \\ \vdots \\ \hat{\mathbf{x}}_p \\ x_0 \\ x_{1,s} \\ \vdots \\ x_{p,s} \end{pmatrix}. \quad (23)$$

Consequently, the solution is obtained by first solving the *reduced system*

$$\begin{pmatrix} I_m & & & \\ w_{1,s} & I_m & & \\ & \ddots & \ddots & \\ & & w_{p,s} & I_m \end{pmatrix} \begin{pmatrix} y_0 \\ y_{1,s} \\ \vdots \\ y_{p,s} \end{pmatrix} = \begin{pmatrix} x_0 \\ x_{1,s} \\ \vdots \\ x_{p,s} \end{pmatrix}, \quad (24)$$

and then updating in parallel the right hand side in (23). Only the reduced system (24) is not directly solvable in parallel. However, if needed, its solution could be also parallelized, by using block cyclic reduction.

We observe that the size of the reduced system only depends on the number p of points in the coarse mesh and on the size m of the continuous problem. It is then independent of both the number k of steps of the main method and the blocksize s in its block version.

When block cyclic reduction is considered for the solution of the reduced system (24), we obtain (see [2,3,14]) that the parallel complexity of the algorithm on p processors is approximately given by $sk^2m^3 + \mu m^3 \log_2 p$ flops ($\mu = 2$) plus $\log_2 p$ communications of length m . On the other hand,

Table 1
Pollution problem

Solver	rtol	atol	h0	scd	steps	accept	# f	# Jac	# LU	CPU
ETR	10^{-4}	10^{-4}	10^{-4}	1.36	12	12	265	6	12	0.01
	10^{-6}	10^{-6}	10^{-6}	3.56	19	19	615	12	19	0.02
	10^{-9}	10^{-9}	10^{-9}	8.04	49	49	2385	42	49	0.08
RADAU5	10^{-4}	10^{-4}	10^{-4}	1.23	22	18	156	15	21	0.01
	10^{-6}	10^{-6}	10^{-6}	2.84	29	24	201	19	29	0.02
	10^{-9}	10^{-9}	10^{-9}	6.79	49	48	366	28	42	0.04

Table 2
Emep problem

Solver	rtol	atol	h0	scd	steps	accept	# f	# Jac	# LU	CPU
ETR	10^{-2}	1	10^{-7}	2.29	220	186	8508	218	160	3.38
	10^{-4}	1	10^{-7}	3.59	355	293	16838	282	342	6.41
	10^{-6}	1	10^{-7}	6.20	673	581	39506	482	644	14.82
RADAU5	10^{-2}	1	10^{-7}	2.81	327	239	3312	224	324	3.86
	10^{-4}	1	10^{-7}	2.67	493	407	4726	378	479	5.67
	10^{-6}	1	10^{-7}	4.81	942	821	8138	756	905	10.45

Tables 1 and 2 summarize the numerical results. Figs. 2 and 3 contain the corresponding work-precision diagrams. In each table and figure, scd denotes the number of significant computed digits. For the diagrams, the following parameters have been used:

- *Emep*: $\text{rtol} = 10^{-(2+m/4)}$, $m = 0, \dots, 24$; $\text{atol} = 1$; $\text{h0} = 10^{-7}$;
- *Pollution*: $\text{atol} = \text{rtol} = \text{h0} = 10^{-(4+m/4)}$, $m = 0, \dots, M$, where $M = 24$ for ETR, and $M = 32$ for RADAU5.

Consider now the singularly perturbed BVP,

$$\varepsilon y'' - ty' + y = 0, \quad y(-1) = 1, \quad y(1) = 2, \quad (26)$$

where $\varepsilon = 10^{-4}$. This problem is very difficult to solve, and most of the currently available BVP solvers fail to provide the correct solution, when started from a uniform mesh. For example, the current version of the popular code COLSYS fails to provide a correct answer for $\varepsilon \leq 10^{-2}$. We obtain the approximated solution reported in Fig. 4. The final mesh contains 480 points, and the estimated error is 9×10^{-5} . For $\varepsilon = 10^{-5}$, we obtain a final mesh of 840 points, and an estimated error 10^{-4} .

The second test problem, still in the class of singularly perturbed problems, is the following:

$$\varepsilon y'' - 2ty' = 0, \quad y(-1) = 1, \quad y(1) = 2, \quad (27)$$

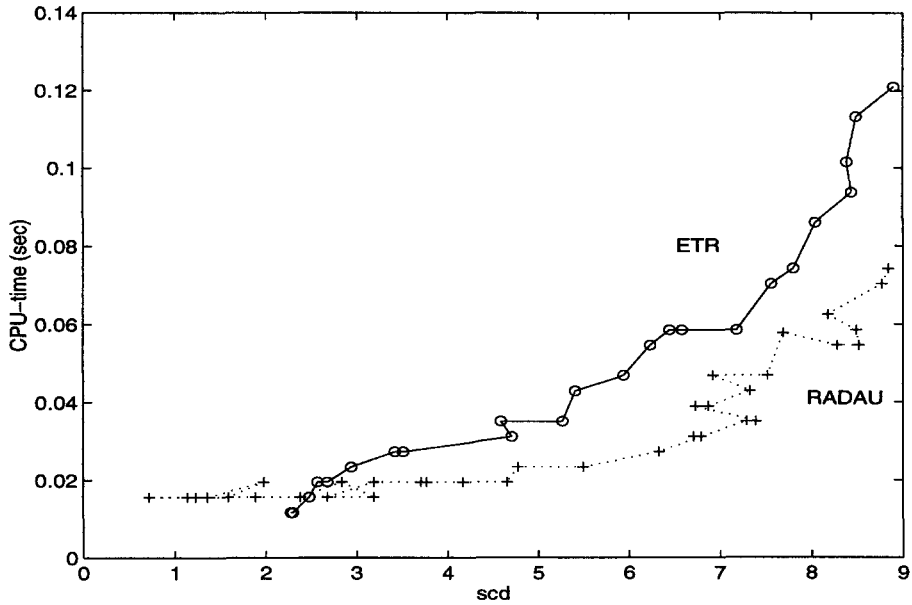


Fig. 2.

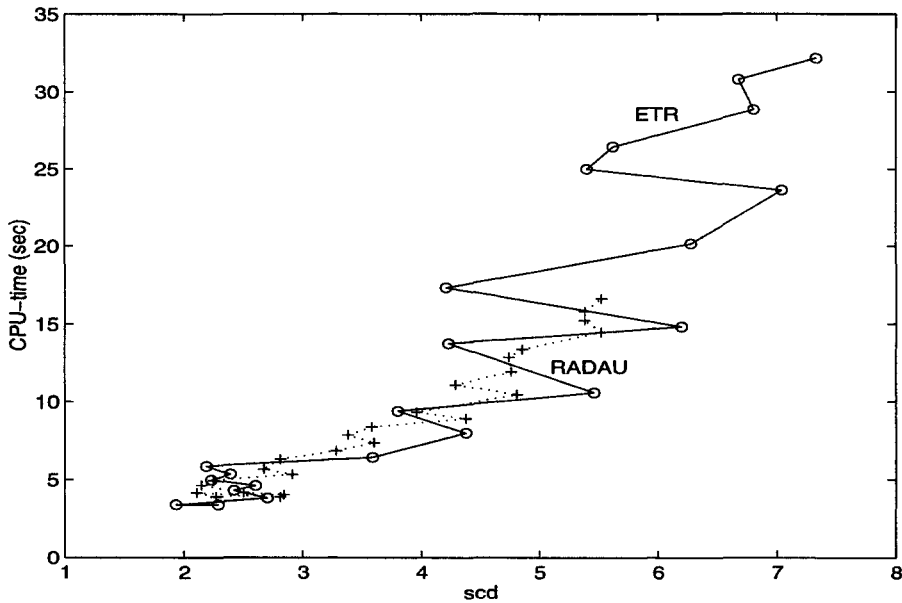


Fig. 3.

where $\varepsilon = 10^{-4}$. As in the previous case, COLSYS fails to provide the correct solution, when started from a uniform mesh. We obtain a final mesh of 480 points, and an estimated error 9×10^{-5} (see Fig. 5). For $\varepsilon = 10^{-5}$, we obtain a final mesh of 760 points, and estimated error 2×10^{-5} .

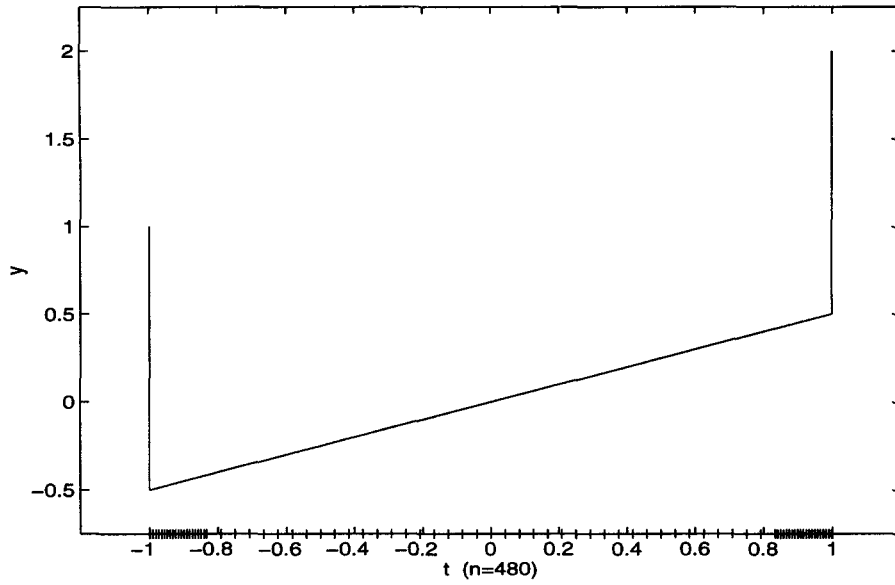


Fig. 4.

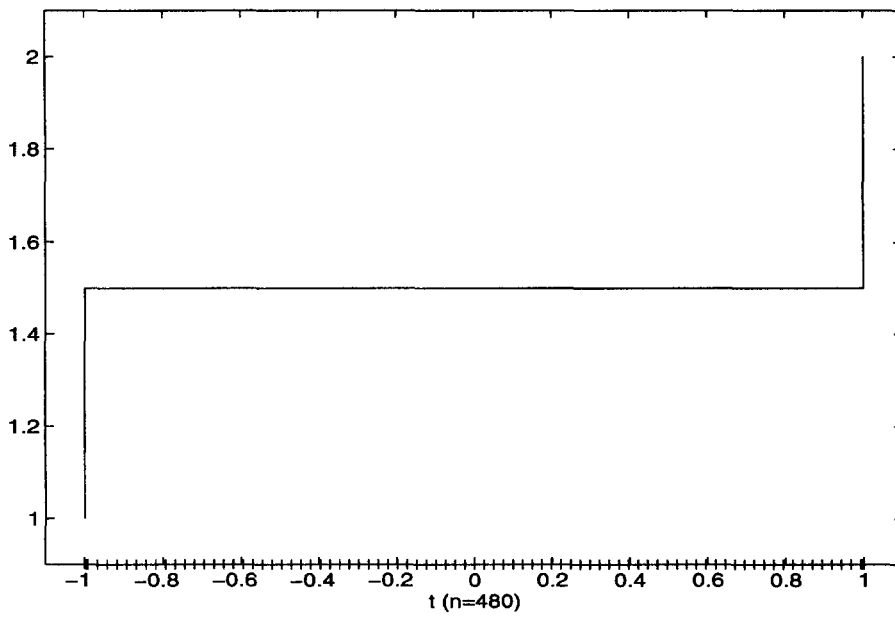


Fig. 5.

For the parallel algorithm, we consider two linear problems, one is an IVP and the other a BVP. In both cases, a uniform mesh has been considered, and the speedup over the sequential implementation of the methods is computed. In this case, the time for the parallel execution includes both computations and communications. The parallel computer used is a transputer based machine. The results have been

Table 3
Measured speedups for problem (28), $s = 40$

$p \setminus k$	3	5	7	9
1	1.00	1.00	1.00	1.00
2	1.94	1.96	1.97	1.98
4	3.74	3.84	3.88	3.92
8	7.11	7.43	7.58	7.72
16	14.03	14.70	14.98	15.24

Table 4
Measured execution times for problem (29)

$p \setminus k$	3	5	7	9
1	4217	6137	8482	10899
2	3955	5822	8571	10993
4	4110	5987	8007	10259
8	4176	6043	8116	10404
16	4265	6030	8151	10551

obtained by using a simplified parallel prototype, even though a more general parallel solver, which is able to efficiently solve nonlinear problems, is under development [15].

The first problem is

$$y' = \begin{pmatrix} -21 & 19 & -20 \\ 19 & -21 & 20 \\ 40 & -40 & -40 \end{pmatrix} y, \quad y(0) = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}, \quad t \in [0, 5]. \tag{28}$$

In Table 3 we report the measured speedups on $p = 1, 2, 4, 8, 16$ processors. We have considered ETRs of different order ($k = 3, 5, 7, 9$), having fixed the blocksize $s = 40$.

Finally, we consider the following second order boundary value problem ($\varepsilon = 10^{-3}$),

$$\begin{aligned} \varepsilon u'' &= u, \\ \varepsilon y'' &= t + \frac{t}{2} u' - \frac{t}{2} y' - \varepsilon \pi^2 \cos(\pi t) - \frac{t}{2} \pi \sin(\pi t), \\ u(-1) &= -y(-1) = 1, \quad u(1) = y(1) = e^{-2/\sqrt{\varepsilon}}, \end{aligned} \tag{29}$$

solved after recasting it as a first order system. Also in this case, we have fixed the blocksize $s = 40$ for the methods. Then, we solve the problem on p processors by using a constant stepsize $h = 2(ps)^{-1}$, $p = 1, 2, 4, 8, 16$. This means that when the stepsize is halved, and consequently the size of the discrete problem is doubled, the number of the parallel processors used is also doubled. As a consequence, we expect the execution time to remain approximately constant for increasing number of processors, even if the accuracy of the solution improves, due to the use of a smaller stepsize. In

Table 5
Measured maximum absolute errors for problem (29)

$p \setminus k$	3	5	7	9
1	3.6e-2	1.2e-2	7.5e-03	5.5e-03
2	3.8e-3	9.3e-4	1.2e-04	5.0e-04
4	2.4e-4	1.4e-5	3.5e-06	1.6e-06
8	1.1e-5	1.9e-7	3.9e-09	4.5e-10
16	8.3e-7	3.1e-9	2.1e-11	5.6e-13

Table 4 we report the measured execution times (the unit corresponds to 64 μ sec). In Table 5 we report the maximum absolute errors, showing that, as predicted, the maximum error decreases with the number p of processors.

Acknowledgements

The authors are very indebted with Dr. F. Iavernaro and Dr. F. Mazzia, for providing some of the numerical tests.

References

- [1] P. Amodio, A -stable k -step linear multistep formulae of order $2k$ for the solution of stiff ODEs, Report 24/96, Dipartimento di Matematica, Università degli Studi di Bari (1996).
- [2] P. Amodio and L. Brugnano, Parallel implementation of block boundary value methods for ODEs, *J. Comput. Appl. Math.* 78 (1997) 197–211.
- [3] P. Amodio and L. Brugnano, Parallel ODE solvers based on block BVMs, *Adv. Comput. Math.* 7 (1–2) (1997) 5–26.
- [4] P. Amodio and F. Mazzia, A boundary value approach to the numerical solution of initial value problems by multistep methods, *J. Difference Equations Appl.* 1 (1995) 353–367.
- [5] A.O.H. Axelsson and J.G. Verwer, Boundary value techniques for initial value problems in ordinary differential equations, *Math. Comp.* 45 (1985) 153–171.
- [6] L. Brugnano, Essentially symplectic boundary value methods for linear Hamiltonian systems, *J. Comput. Math.* 15 (1997) 233–252.
- [7] L. Brugnano, *Boundary Value Methods for the Numerical Approximation of Ordinary Differential Equations*, Lecture Notes in Computer Science 1196 (Springer, Berlin, 1997) 78–89.
- [8] L. Brugnano and D. Trigiante, Convergence and stability of boundary value methods for ordinary differential equations, *J. Comput. Appl. Math.* 66 (1996) 97–109.
- [9] L. Brugnano and D. Trigiante, High order multistep methods for boundary value problems, *Appl. Numer. Math.* 18 (1995) 79–94.
- [10] L. Brugnano and D. Trigiante, Block boundary value methods for linear Hamiltonian systems, *Appl. Math. Math. Comput.* 81 (1997) 49–68.
- [11] L. Brugnano and D. Trigiante, On the characterization of stiffness for ODEs, *Dynamics of Continuous, Discrete and Impulsive Systems* 2 (3) (1996) 317–336.

- [12] L. Brugnano and D. Trigiante, A new mesh selection strategy for ODEs, *Appl. Numer. Math.* 24 (1997) 1–21.
- [13] L. Brugnano and D. Trigiante, Boundary value methods: The third way between linear multistep and Runge–Kutta methods, *Comput. Math. Appl.* (to appear).
- [14] L. Brugnano and D. Trigiante, *Solving Differential Problems by Multistep Initial and Boundary Value Methods* (Gordon & Breach, to appear).
- [15] L. Brugnano and D. Trigiante, Parallel implementation of block boundary value methods on nonlinear problems: theoretical results, in preparation.
- [16] J.R. Cash, *Stable Recursions* (Academic Press, London, 1979).
- [17] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II*, Springer Series in Computational Mathematics 14 (Springer, Berlin, 1991).
- [18] F. Iavernaro and F. Mazzia, The ETR code for stiff initial value problems, in preparation.
- [19] W.M. Lioen, J.J.B. de Swart and W.A. van der Veen, Test set for IVP solvers, CWI Report NM-R9615, Amsterdam (1996).
- [20] I. Sgura, Approssimazione numerica di ODE con metodi BVM: proprietà, applicazioni e confronti, Tesi di Dottorato di Ricerca, Pisa (1996).