

APNUM 446

A parallel preconditioning technique for boundary value methods *

L. Brugnano and D. Trigiante

Dipartimento di Energetica, Via Lombroso 6 / 17, 50134 Firenze, Italy

Abstract

Brugnano, L. and D. Trigiante, A parallel preconditioning technique for boundary value methods, Applied Numerical Mathematics 13 (1993) 277–290.

The boundary value methods (BVMs) are a class of numerical methods for solving initial value problems for ODEs [3,7,14]. One reason that prevented their diffusion in the past years was their higher cost, with respect to the standard initial value methods. We show that BVMs may become competitive when they are efficiently implemented on parallel or vector computers.

Since they require the solution of large, sparse block linear systems, usually obtained by using an iterative method, a preconditioning technique is needed for their efficient implementation. In this paper we introduce and study a new preconditioning technique.

Some numerical tests on a distributed memory parallel computer are reported.

1. Introduction

Let us consider the problem of solving the initial value problem:

$$y'(t) = Ly(t) + b(t), \quad y(t_0) = y_0, \quad t \in [t_0, T], \quad (1)$$

where $y(t)$, $b(t): [t_0, T] \rightarrow \mathbb{R}^m$ and $L \in \mathbb{R}^{m \times m}$ is a constant matrix. Moreover, we shall assume that all the eigenvalues of L have negative real part and $b(t)$ is a smooth and uniformly bounded function.

A three-point BVM [2–4,7,13,14] is obtained by first considering a partitioning of the interval $[t_0, T]$, $t_0 < t_1 < t_2 < \dots < t_k = T$, such that $t_i = t_{i-1} + h_i$, $i = 1, \dots, k$. The problem (1) is then discretized by using a two-step method (*main method*), while in the last step it is discretized by using an implicit one-step method (*last-point method*).

In [5] the stability properties of three particular BVMs have been examined. They utilize, as main and last-point method, the couples:

Correspondence to: L. Brugnano, Dipartimento di Energetica, Via Lombroso 6/17, 50134 Firenze, Italy. E-mail: udini@vm.cnuce.cnr.it.

* Work supported by MURST (40% project), and CNR (Contract No. 92.00535.CT01 and P.F. “Calcolo Parallelo”, sottoprogetto 1).

- mid-point method and implicit Euler method,
- Simpson's method and trapezoidal rule,
- Adams' method and trapezoidal rule.

The stability properties were studied in the case of variable stepsizes defined by

$$h_{i+1} = rh_i, \quad i = 1, \dots, k-1, \quad (2)$$

where $r > 1$ is a fixed parameter, and $h_1 < \|L\|^{-1}$ (when not specified, $\|\cdot\|$ may be $\|\cdot\|_1$, or $\|\cdot\|_2$, or $\|\cdot\|_\infty$).

The discrete problem originated by a BVM for problem (1) is the linear system

$$Ay = c. \quad (3)$$

The matrix A is block tridiagonal for the three BVMs considered above. In fact, if the integration steps are chosen according to (2), A can be written as

$$A = T_1 \otimes I - T_2 \otimes h_1 L, \quad (4)$$

where \otimes denotes the right Kronecker product (see [11]), and the matrices T_1 and T_2 both depend on the main and last-point methods chosen. For example

$$T_1 = \begin{pmatrix} (1-r^{-2}) & r^{-2} & & & \\ -1 & \ddots & \ddots & & \\ & \ddots & (1-r^{-2}) & r^{-2} & \\ & & & -1 & 1 \end{pmatrix} \quad (5)$$

and

$$T_2 = D \begin{pmatrix} (1+r^{-1}) & & & & \\ & \ddots & & & \\ & & (1-r^{-1}) & & \\ & & & & 1 \end{pmatrix}, \quad (6)$$

where

$$D = \text{diag}(1, r, r^2, \dots, r^{k-1}), \quad (7)$$

define the BVM which utilizes the mid-point method as main method and the implicit Euler method as last-point method. If the Simpson method is used as main method and the trapezoidal rule as last-point method, then T_1 and D are the matrices defined above and

$$T_2 = D \begin{pmatrix} \frac{2}{3}(1+r^{-1}) & \frac{1}{3}r^{-1} & & & \\ \frac{1}{3} & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{1}{3} & \frac{2}{3}(1+r^{-1}) & \frac{1}{3}r^{-1} \\ & & & \frac{1}{2} & \frac{1}{2} \end{pmatrix}. \quad (8)$$

Lastly, if the Adams method is used as main method and the trapezoidal rule is used as last-point method, then one obtains:

$$T_1 = \begin{pmatrix} 1 & & & & \\ -1 & & & & \\ & \ddots & & & \\ & & \ddots & & \\ & & & -1 & 1 \end{pmatrix} \tag{9}$$

and

$$T_2 = (6r(r+1))^{-1} D \begin{pmatrix} (1+3r)(r+1) & -1 & & & \\ r(2+3r) & & & & \\ & \ddots & & & \\ & & r(2+3r) & (1+3r)(r+1) & -1 \\ & & & 3r(r+1) & 3r(r+1) \end{pmatrix}, \tag{10}$$

where D is the same matrix as defined in (7).

When the size m of the matrix L is small, the solution of the linear system (3) may be conveniently obtained by using a direct method [2]. Conversely it is more convenient to use an iterative solver. In both cases, it is possible to efficiently implement the method on a parallel computer. In this paper we shall analyze the iterative solution of (3). We stress that in general the iterative solver converges very slowly, making the BVMs not competitive with respect to other known ODE solvers. In order to improve the convergence of the iterative solver we need to precondition the linear system (3). For this reason a new preconditioner will be introduced in Section 2. Its properties will be examined in Section 3. In Section 4 its parallel implementation will be outlined, while in Section 5 a device to reduce the computational cost of the corresponding algorithm in the case of L constant is presented. Finally, in Section 6 some numerical tests are reported, from which one realizes that the BVMs are competitive especially for very stiff problems.

2. Preconditioning techniques

To precondition the matrix A in (3) two ways may be essentially followed:

- looking for an approximation of the inverse matrix itself;
- using a suitable approximation of the differential operator from which the matrix A originates.

In a previous paper [4] a preconditioner of the first type was discussed. In this paper we have chosen the latter way to obtain a cheaper preconditioner. In particular, when all the eigenvalues of the matrix L in (1) have negative real part, the simplest numerical scheme which approximates the differential operator while preserving the qualitative properties of the solution is the implicit Euler method. The preconditioner is then obtained by applying this method on the same partitioning of the interval of integration considered for the original BVM.

The corresponding matrix is the block lower bidiagonal matrix:

$$P = \begin{pmatrix} I - h_1 L & & & & \\ -I & I - h_2 L & & & \\ & \ddots & \ddots & & \\ & & & -I & I - h_k L \end{pmatrix}. \quad (11)$$

This matrix is easily invertible. Therefore, we shall solve the linear problem

$$P^{-1}Ay = P^{-1}c, \quad (12)$$

if a *left* preconditioner is used. Alternately, a *right* preconditioner may be used: in this case the linear systems

$$AP^{-1}z = c, \quad y = P^{-1}z$$

need to be solved. Since the two approaches are practically equivalent [6], we shall only analyze left preconditioning.

If variable stepsizes in (2) are used, then the preconditioner (11) can be written as:

$$P = B \otimes I - D \otimes h_1 L, \quad (13)$$

where B is the lower bidiagonal matrix defined in (9), and the matrix D is the diagonal matrix defined in (7).

We observe that the term $D \otimes h_1 L$ in (13) is a good approximation of the term $T_2 \otimes h_1 L$ in (4). Moreover, if the mid-point method or Simpson's method are used as main method, then the larger r is, the better B approximates T_1 (see (9) and (5)), while this is always true for the Adams method. It follows that we should expect smaller values for $\kappa(P^{-1}A)$, the condition number of $P^{-1}A$, when r is large. The worst case is $r = 1$, corresponding to the use of a constant integration step.

3. Properties of the preconditioned matrix

In this section we shall study the properties of the preconditioned matrix $P^{-1}A$. In particular, we are interested in the conditioning of this matrix and in the distribution of its eigenvalues. The spectrum of the preconditioned matrix is important since an iterative solver belonging to the class of oblique projection methods is used for solving (12): to have fast convergence it is desirable to have the spectrum contained in the right half of the complex plane, and clustered in a region which does not contain the origin [16,17].

Concerning the conditioning of the matrix $P^{-1}A$, we shall prove that it does not depend on the number k of the integration steps, but only on the matrix L which defines problem (1). In fact, the following result holds:

Theorem 3.1. *If all the eigenvalues of the matrix L have negative real part, then, for the BVMs defined by (4)–(10) and the preconditioning matrix defined by (13), the matrix $P^{-1}A$ is well-conditioned $\forall r > 1$, provided that the imaginary part of each complex eigenvalue is "sufficiently" small.*

The proof will follow as an easy consequence of the next two lemmas. In the above theorem *well-conditioned* has the following meaning:

Definition 3.2. A nonsingular matrix is said to be *well-conditioned* if its condition number is bounded by a quantity independent on the size of the matrix. If this quantity depends as a polynomial of low degree (1 or 2) on the size of the matrix, then the matrix is said to be *weakly well-conditioned*.

In [5] the following result is proved:

Lemma 3.3. *If all the eigenvalues of the matrix L have negative real part, then, for the BVMs defined by (4)–(10), the matrix $(D^{-1} \otimes I)A$ is well-conditioned $\forall r > 1$, provided that the imaginary part of each complex eigenvalue is “sufficiently” small.*

Since

$$\begin{aligned} \kappa(P^{-1}A) &= \kappa((P^{-1}(D \otimes I)(D^{-1} \otimes I)A) \\ &\leq \kappa((D^{-1} \otimes I)P)\kappa((D^{-1} \otimes I)A), \end{aligned}$$

it follows that Theorem 3.1 will be true if the following result holds:

Lemma 3.4. *If all the eigenvalues of the matrix L have negative real part and P is the matrix defined by (13), then the matrix $(D^{-1} \otimes I)P$ is well-conditioned $\forall r > 1$.*

Proof. Let L be a normal matrix and $L = QAQ^T$ its Schur decomposition, where $A = \text{diag}(\lambda_1, \dots, \lambda_m)$ and Q is a unitary matrix. Then, instead of the matrix P , we can consider the following matrix:

$$(I \otimes Q^T)P(I \otimes Q) = B \otimes I - D \otimes h_1\lambda =: P_N. \tag{14}$$

Denoting $q_i = h_1\lambda_i$, it follows that we can study the conditioning of P_N by analyzing the $k \times k$ lower bidiagonal matrices $B - q_iD$, $i = 1, \dots, m$. For this reason, let us consider the generic matrix

$$P_q = B - qD,$$

where q is a complex number with negative real part. It is straightforward to verify that, in this case, the matrix $D^{-1}P_q$ is diagonally dominant on both rows and columns for every $r \geq 1$. This implies that both $D^{-1}P_q$ and $(D^{-1} \otimes I)P_N$ are well-conditioned.

This result can be extended to more general matrices L . In fact, let $L = Q(\Lambda + N)Q^T$ be the Schur decomposition of L , where N is strictly upper triangular and nilpotent of order m (the size of the matrix L). Then, instead of the matrix (13), we can consider the matrix

$$\begin{aligned} (I \otimes Q^T)P(I \otimes Q) &= B \otimes I - D \otimes h_1\Lambda - D \otimes h_1N \\ &= P_N(I \otimes I - P_N^{-1}(D \otimes h_1N)), \end{aligned}$$

where P_N is the matrix defined in (14). The matrix $(D^{-1} \otimes I)P_N$ is well-conditioned, as seen above. Moreover one has

$$\begin{aligned} P_N^{-1}(D \otimes h_1 N) &= (B \otimes I - D \otimes h_1 \Lambda)^{-1}(D \otimes h_1 N) \\ &= (I \otimes I - B^{-1}D \otimes h_1 \Lambda)^{-1}(B^{-1}D \otimes h_1 N) \\ &= \left(\sum_{i=0}^{km-1} b_i (B^{-1}D)^i \otimes \Lambda^i \right) (B^{-1}D \otimes h_1 N), \end{aligned}$$

for some scalars b_0, \dots, b_{km-1} (see [11, Chapter 9]), and then

$$\begin{aligned} P_N^{-1}(D \otimes h_1 N) &= h_1 \sum_{i=0}^{km-1} b_i (B^{-1}D)^{i+1} \otimes \Lambda^i N \\ &=: h_1 \sum_{i=0}^{km-1} b_i (B^{-1}D)^{i+1} \otimes \tilde{N}_i. \end{aligned}$$

It follows that the matrix is nilpotent of order m , since the matrices \tilde{N}_i are strictly upper triangular. As a consequence, we have that the matrix $(I \otimes I - P_N^{-1}(D \otimes h_1 N))$ is invertible (since all its eigenvalues are equal to one) and well-conditioned. \square

The spectrum of the preconditioned matrix $P^{-1}A$ will not include the origin, since the matrix is well-conditioned. The following considerations will provide some asymptotic (for $k \gg 0$) bounds for the location of the eigenvalues. By using arguments similar to those used in the proof of Lemma 3.4, one could show that it is sufficient to analyze the eigenvalue distribution of the matrix obtained from problem (1) with $L = \lambda$, where λ is a complex number with negative real part. In this case, by posing $q = h_1 \lambda$, we obtain the matrix (see (4) and (13))

$$G(q, r, k) = (B - qD)^{-1}(T_1 - qT_2),$$

where q has negative real part. We know that $G(0, r, k) = B^{-1}T_1$ has a multiple eigenvalue of multiplicity k which is equal to 1, since this matrix is similar to

$$T_1 B^{-1} = \begin{pmatrix} 1 & r^{-2} & & & \\ & 1 & \cdot & & \\ & & \cdot & \cdot & \\ & & & \cdot & r^{-2} \\ & & & & 1 \end{pmatrix}.$$

Moreover, since we choose (see [5]) $h_1 < \|L\|^{-1}$, it follows that $|q| < 1$ must hold. Although it is difficult to locate exactly each eigenvalue of $G(q, r, k)$ for $q \neq 0$, it is nevertheless possible to show that, for $r > 1$ and k “large”, most of the eigenvalues are clustered near 1. In fact, if μ is an eigenvalue of matrix $G(q, r, k)$, then the relation

$$D^{-1}(T_1 - qT_2)x = \mu(D^{-1}B - qI)x \tag{15}$$

holds for some nonzero vector $x = (x_1, \dots, x_k)^T$. For the mid-point method the i th row of (15) is given by:

$$\begin{aligned} -r^{1-i}x_{i-1} + (r^{1-i}(1 - r^{-2}) + q(1 + r^{-1}))x_i + r^{-i-1}x_{i+1} \\ = \mu(-r^{1-i}x_{i-1} + (r^{1-i} + 1)x_i). \end{aligned}$$

Let us suppose now that $k \gg 0$. Then, since $r > 1$, for i greater than a suitable index i^* , since the previous relation can be approximated by

$$q(1 + r^{-1})x_i = q\mu x_i,$$

which shows that $\mu = (1 + r^{-1})$ is an eigenvalue of $G(q, r, k)$ and

$$e_i = \left(0, \dots, 0, \underbrace{1}_i, 0, \dots, 0\right)^T$$

is the corresponding eigenvector.

For Simpson's method and Adams' method similar arguments hold. Let us briefly sketch them for the Simpson method. The i th row of (15) is now given by:

$$\begin{aligned} & \left(\frac{1}{3}q - r^{1-i}\right)x_{i-1} + \left(r^{1-i}(1 - r^{-2}) + \frac{2}{3}q(1 + r^{-1})\right)x_i + \left(\frac{1}{3}r^{-1}q + r^{-i-1}\right)x_{i+1} \\ & = \mu\left(-r^{1-i}x_{i-1} + (r^{1-i} + q)x_i\right). \end{aligned}$$

Let us suppose now that $k \gg 0$. Then, since $r > 1$, for i greater than a suitable index i^* , the previous relation can be approximated by

$$\frac{1}{3}qx_{i-1} + \frac{2}{3}q(1 + r^{-1})x_i + \frac{1}{3}r^{-1}qx_{i+1} = q\mu x_i,$$

that is

$$\frac{1}{3}x_{i-1} + \frac{2}{3}(1 + r^{-1})x_i + \frac{1}{3}r^{-1}x_{i+1} = \mu x_i.$$

This means that most of the eigenvalues will be contained in the pseudospectrum of the matrix $D^{-1}T_2$ (see (8)), which is given by the region enclosed in the ellipse (see [12, p. 170])

$$\frac{1}{3}(1 + r^{-1})(2 + \cos(\theta)) + i\frac{1}{3}(1 - r^{-1}) \sin(\theta), \quad 0 \leq \theta \leq 2\pi. \tag{16}$$

Similarly, for the Adams method one obtains that, for $r > 1$ and $k \gg 0$, most of the eigenvalues of the preconditioned matrix lie in the region enclosed in the ellipse given by (see (10))

$$\frac{1 + 3r}{6r} + \frac{r(2 + 3r) - 1}{6r(r + 1)} \cos(\theta) + i \frac{r(2 + 3r) + 1}{6r(r + 1)} \sin(\theta), \quad 0 \leq \theta \leq 2\pi.$$

To give an idea of the effectiveness of the above bounds, we shall consider the Simpson method plus the trapezoidal rule applied to problem (1) with

$$L = - \begin{pmatrix} 1e4 & 1e3 & 1e2 \\ 0 & 1e1 & 1e0 \\ 0 & 0 & 1e-5 \end{pmatrix}.$$

The parameters used are $h_1 = 5e-5$, $r = 1.2$, $k = 150$. In Fig. 1 the spectrum of the preconditioned matrix $P^{-1}A$ is plotted, along with the ellipse given by (16): 273 of the 450 eigenvalues lie inside the ellipse. Moreover, for larger values of k , the number of eigenvalues which lie outside the ellipse (177) does not increase: for example, for $k = 200$ one finds 423 eigenvalues (of 600) inside the ellipse. For comparison, in Fig. 2 the spectrum of the non-preconditioned matrix A is shown.

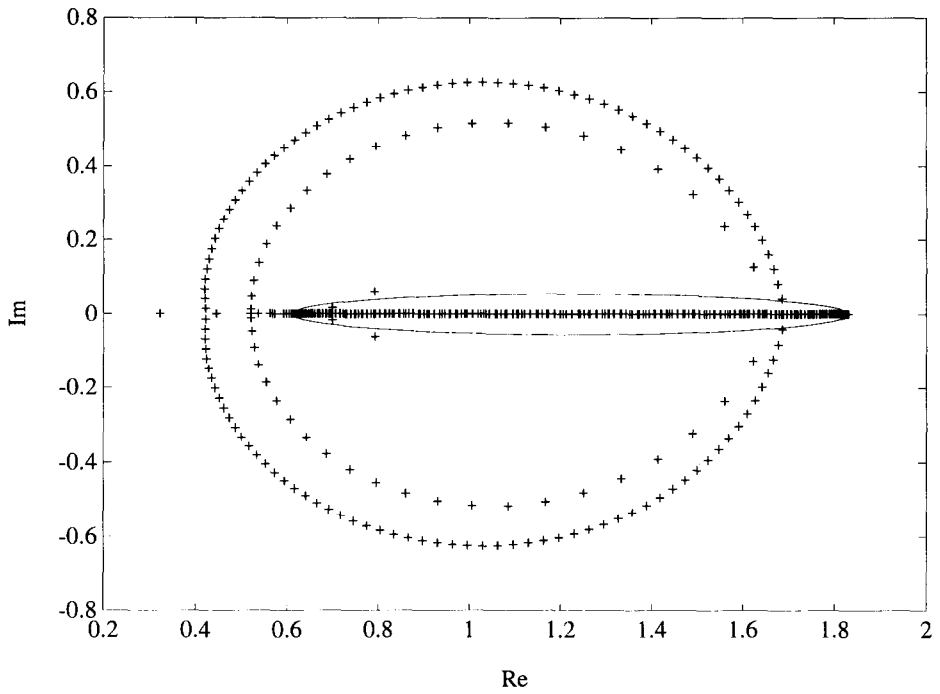


Fig. 1. Eigenvalues of the preconditioned matrix; the ellipse is the one described by (16).

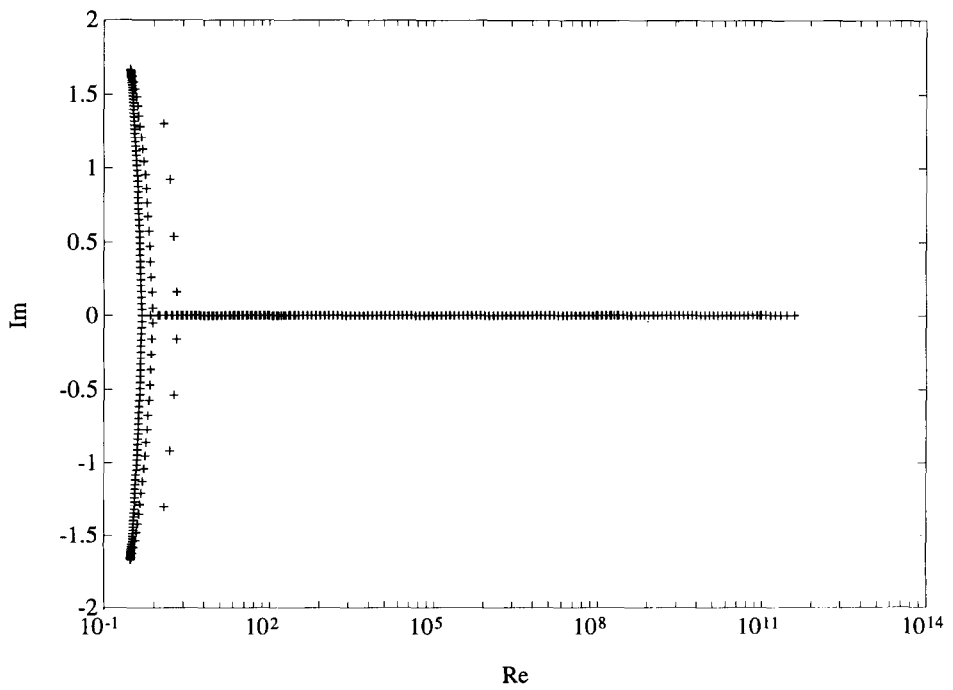


Fig. 2. Eigenvalues of the non-preconditioned matrix.

4. Parallel implementation of the preconditioner

Each application of the preconditioner (11) requires the solution of a block lower bidiagonal linear system. Therefore, the natural implementation of the preconditioner is not parallel, since the multiplication $v := P^{-1}u$ is equivalent to the solution of the following block recurrence:

$$\begin{aligned} v_1 &:= (I - h_1 L)^{-1} u_1, \\ v_i &:= (I - h_i L)^{-1} (u_i + v_{i-1}), \quad i = 2, \dots, k. \end{aligned}$$

To get a parallel preconditioner, we consider the following splitting of the matrix (11):

$$P = \hat{P} - E, \quad (17)$$

where \hat{P} is a block diagonal matrix and E is a nilpotent matrix. The number of diagonal blocks of \hat{P} equals the number of parallel processing units, say p , while matrix E has just $p - 1$ non-zero block entries on the lower subdiagonal. From (17) it follows that

$$P^{-1} = (I - \hat{P}^{-1}E)^{-1} \hat{P}^{-1} = \hat{P}^{-1} (I - E\hat{P}^{-1})^{-1}. \quad (18)$$

The matrix \hat{P}^{-1} is easily obtainable in parallel, while the two matrices $(I - E\hat{P}^{-1})^{-1}$ and $(I - \hat{P}^{-1}E)^{-1}$ are obtainable with expansions which have only the first p terms (since both $(\hat{P}^{-1}E)$ and $(E\hat{P}^{-1})$ are nilpotent of order p). Therefore, if $s \leq p - 1$, (18) is approximated by

$$P^{-1} \approx \sum_{i=0}^s (\hat{P}^{-1}E)^i \hat{P}^{-1} = \hat{P}^{-1} \sum_{i=0}^s (E\hat{P}^{-1})^i. \quad (19)$$

The choice between the two approximations depends on the implementation. For the numerical tests reported in Section 6, the first one is used.

The computational cost of the preconditioner increases with s . However, the minimum value of s in (19) needed to obtain an effective approximation is an increasing function of p . It follows that an optimal balance between the size of the problem and the number of parallel processors needs to be found. A good choice seems to be $s \approx p/2$.

5. Tridiagonalization of the transformation matrix

We observe that the count of scalar operations needed for the product of A by a vector (matvec) is $O(km^2)$. Moreover, the construction of the preconditioner requires $O(km^3)$ flops, and the multiplication by \hat{P}^{-1} (see (19)) has a cost of $O(km^2)$ flops.

The above computations are the most time-consuming operations needed for the iterative solution of (12). However, their cost can be reduced when the matrix L in (1) is constant. In fact, if L is a tridiagonal matrix, the cost for the matvec, the multiplication by \hat{P}^{-1} , and the construction of \hat{P}^{-1} , is reduced to $O(km)$ flops, since all the blocks of the matrices A and P are tridiagonal. It follows that the computational cost reduces dramatically, if L is transformed to tridiagonal form by means of a similarity transformation. It is known that, when L is symmetric, it can be reduced to tridiagonal form by means of orthogonal similarity transformations with a cost of $O(m^3)$ flops. However, when L is unsymmetric one cannot rely on the use of orthogonal similarity transformations to lead L to tridiagonal form, nor is a general

procedure of tridiagonalization known (see [15]). Nevertheless, in [8] an algorithm which utilizes Gauss similarity transformations for reducing L to tridiagonal form is described, along with a procedure to overcome possible breakdowns that may occur during the basic procedure. The whole algorithm has a cost of $O(m^3)$ flops and it turns out to be quite robust. This algorithm has been considered in the implementation of the BVM used for the numerical tests.

6. Numerical tests

In the following tests we shall consider the BVM which uses the Simpson method as main method and the trapezoidal rule as last-point method. The iterative method Bi-CGSTAB [19] is used for solving the linear system (3). Both left and right preconditioning will be considered.

Let $y^{(i)}$ be the approximation of the solution at the i th step, and $r^{(i)}$ the corresponding residual. Given a tolerance $\varepsilon > 0$, Bi-CGSTAB is stopped as soon as one of the following two conditions is verified:

$$\|r^{(i)}\|_{\infty} < \varepsilon$$

or

$$\max_j \frac{|y_j^{(i)} - y_j^{(i-1)}|}{\max(1, |y_j^{(i)}|)} < \varepsilon.$$

The implementation has been made by using the Fortran programming language with the Express parallel library [21] and executed on a linear array of transputers T800-20. The comparison algorithm used is the solver LSODE from Odepack [10] available on Netlib.

In the implementation of the BVM, the step of tridiagonalization of the matrix L has not been parallelized since it is not too expensive for the size of the considered tests ($m = 50$). Nevertheless, for larger values of m , it will be convenient to parallelize this step too.

The construction of the test problems has been made by fixing the spectrum of L and its departure from normality $\Delta(L)$ [9]. The resulting matrix has been transformed by means of a similarity transformation with a random orthogonal matrix. The vectors b and y_0 in (1) have been chosen at random.

The preconditioner for the BVM will be identified by a three-digit label $l_1 l_2 l_3$ defined as follows:

- l_1 : 1 for left and 2 for right preconditioning;
- l_2 : length of the truncated sum (s in (19)) used to compute the initial guess $y^{(0)} \approx P^{-1}c$;
- l_3 : length of the truncated sum (s in (19)) used in the body of Bi-CGSTAB to approximate P^{-1} .

The reference solutions have been computed by means of the matrix exponential. In all the three test problems reported below, the solution computed by the BVM is at least as accurate as the one computed by LSODE.

A variable stepsize as in (2) is used in all the test problems.

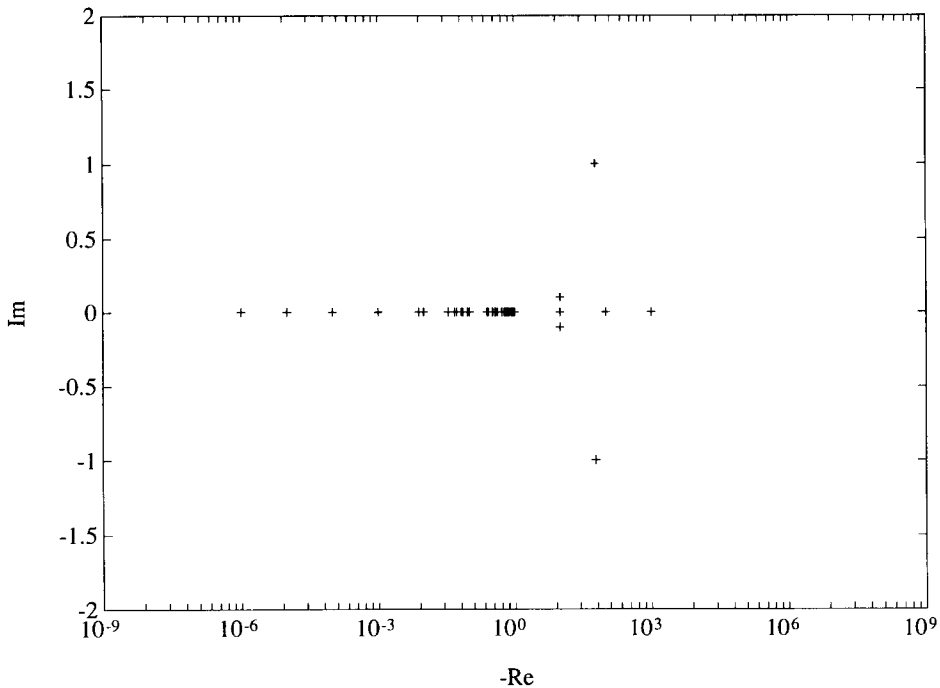


Fig. 3. Eigenvalues of matrix L , problem 1.

The BVM has been used with the following parameters:

$$h_1 = \varepsilon = 1e-5, \quad r = 1.1, \quad k = \begin{cases} 400, & \text{for problem 1,} \\ 320, & \text{for problem 2,} \\ 400, & \text{for problem 3.} \end{cases}$$

LSODE has been used on the same mesh, with the following parameters:

$$mf = 21, \quad atol = rtol = 1e-5, \quad mxstep = 20,000.$$

The spectra of the matrices L are shown in Figs. 3–5 for problems 1–3, respectively. The departures from normality of L for the three problems are:

$$\Delta(L) = \begin{cases} 1.83e0, & \text{for problem 1,} \\ 7.00e0, & \text{for problem 2,} \\ 7.00e-2, & \text{for problem 3.} \end{cases}$$

The measured execution times (in secs) of LSODE and of the BVM on 1 and 8 processors are reported in Table 1, along with the labels of the used preconditioners (in brackets, the iterates of Bi-CGSTAB to get convergence are also reported).

In Table 2 the relative performances on 1 processor and 8 processors are reported. On 16 processors a slightly better performance is obtainable, but in this case the step of tridiagonalization (which has not been parallelized) is no longer negligible. Finally, we observe that LSODE on problem 3 terminates with an error and does not compute the last part of the solution correctly.

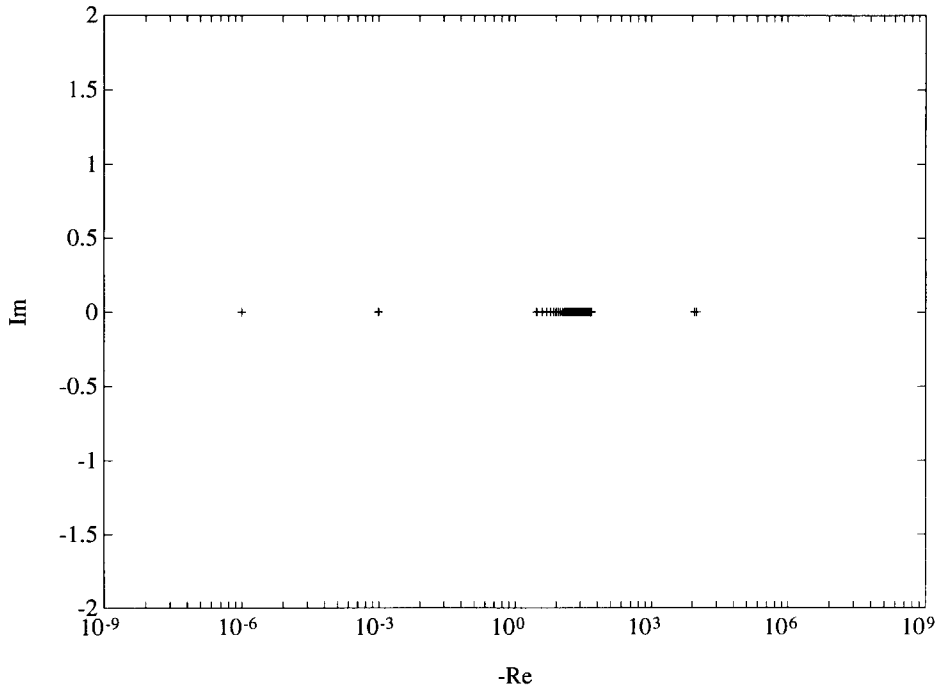


Fig. 4. Eigenvalues of matrix L , problem 2.

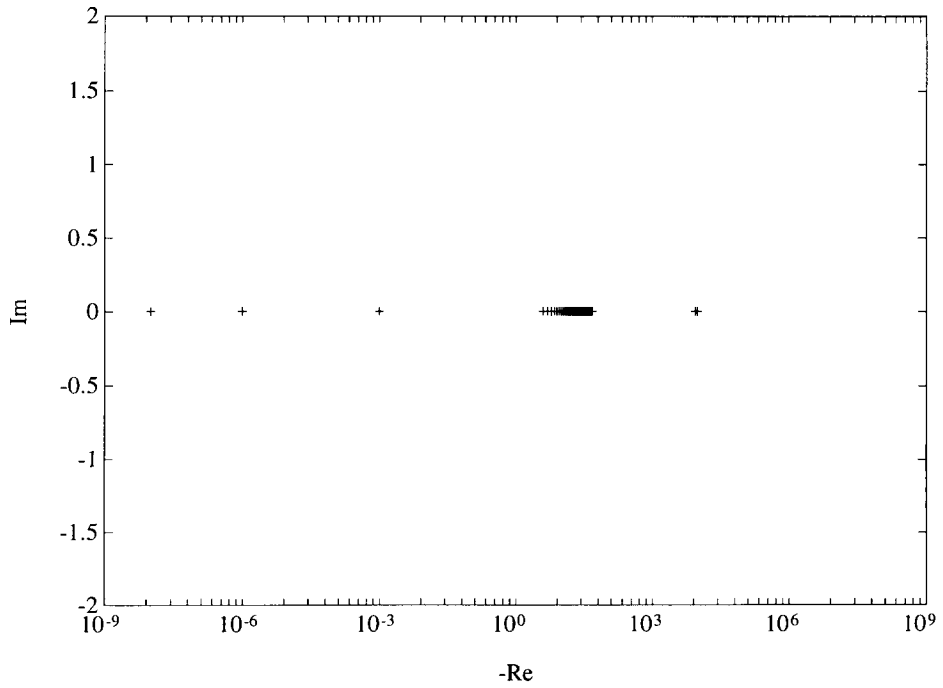


Fig. 5. Eigenvalues of matrix L , problem 3.

Table 1
Absolute performances

Problem	LSODE (1 proc)	BVM (1 proc)		BVM (8 procs)	
	Time	Label	Time	Label	Time
1	61.19	100	70.66 (7)	153	19.00 (7)
		200	76.80 (8)	253	20.25 (8)
2	2,125.52	100	85.71 (12)	154	24.50 (11)
		200	96.36 (14)	254	29.06 (14)
3	84,578.41	100	142.27 (17)	164	45.03 (18)
		200	220.04 (28)	264	27.15 (10)

Table 2
Relative performances

Problem 1	BVM 100/BVM 153	3.72
	LSODE/BVM 153	3.22
	BVM 200/BVM 253	3.79
	LSODE/BVM 253	3.02
Problem 2	BVM 100/BVM 154	3.50
	LSODE/BVM 154	86.75
	BVM 200/BVM 254	3.32
	LSODE/BVM 254	73.14
Problem 3	BVM 100/BVM 164	3.16
	LSODE/BVM 164	1,878.13
	BVM 200/BVM 264	8.10
	LSODE/BVM 264	3,115.19

7. Conclusions

In this paper we have analyzed a preconditioning technique which makes some BVMs, when implemented on a parallel computer, competitive with respect to other known solvers, in particular for stiff problems.

The reported results are relative to the application of the BVMs to a simple autonomous linear system of ODEs. For this problem, the considered mesh strategy is appropriate. Nevertheless, it may result to be unsatisfactory for more general problems: this matter will be the subject of forthcoming papers.

Acknowledgements

We thank one of the referees for his helpful comments and corrections.

References

- [1] P. Amodio, F. Mazzia and D. Trigiante, Stability of some boundary value methods for the solution of initial value problems (submitted).
- [2] P. Amodio and D. Trigiante, A parallel direct method for solving initial value problems for ordinary differential equations, *Appl. Numer. Math.* 11 (1993) 85–93.
- [3] A.O.H. Axelsson and J.G. Verwer, Boundary value techniques for initial value problems in ordinary differential equations, Preprint, Mathematisch Centrum, Amsterdam (1983).
- [4] L. Brugnano, F. Mazzia and D. Trigiante, Parallel implementation of BVM methods, *Appl. Numer. Math.* 11 (1993) 115–124.
- [5] L. Brugnano and D. Trigiante, Stability properties of some boundary value methods, *Appl. Numer. Math.* 13 (1993) 291–304 (this issue).
- [6] L. Brugnano and D. Trigiante, A parallel preconditioning technique for BVM methods, Report, Dipartimento di Energetica dell' Università di Firenze, Firenze, Italy (1992).
- [7] J.R. Cash, *Stable Recursion* (Academic Press, New York, 1976).
- [8] G.A. Geist, Reduction of a general matrix to tridiagonal form, *SIAM J. Matrix Anal. Appl.* 12 (1991) 362–373.
- [9] G.H. Golub and C.F. van Loan, *Matrix Computations* (The Johns Hopkins University Press, Baltimore, MD, 2nd ed., 1989).
- [10] A.C. Hindmarsh, Odepack, a systematized collection of ODE solvers, in: R.S. Stepleman et al., eds., *Scientific Computing* (North-Holland, Amsterdam, 1983).
- [11] P. Lancaster and M. Tismenetsky, *The Theory of the Matrices, with applications* (Academic Press, New York, 2nd ed., 1985).
- [12] V. Lakshmikantham and D. Trigiante, *Theory of Difference Equations: Numerical Methods and Applications* (Academic Press, New York, 1988).
- [13] L. Lopez, Two-step boundary value methods in the solution of ODEs, *Comput. Math. Appl.* (to appear).
- [14] L. Lopez and D. Trigiante, Boundary value methods and BV-stability in the solution of initial value problems, *Appl. Numer. Math.* 11 (1993) 225–239.
- [15] B.N. Parlett, Reduction to tridiagonal form and minimal realizations, Preprint, Center for Pure and Applied Mathematics, Berkeley, CA (1990).
- [16] Y. Saad, Krylov subspace methods for solving large unsymmetric linear systems, *Math. Comp.* 37 (1981) 105–126.
- [17] Y. Saad, The Lanczos biorthogonalization algorithm and other oblique projection methods for solving large unsymmetric systems, *SIAM J. Numer. Anal.* 19 (1982) 485–506.
- [18] G.W. Stewart, The convergence of the method of conjugate gradients at isolated extreme points in the spectrum, *Numer. Math.* 24 (1975) 85–93.
- [19] H.A. van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* 13 (1992) 631–644.
- [20] J.H. Wilkinson, *The Algebraic Eigenvalue Problem* (Clarendon Press, Oxford, England, 1978).
- [21] Express: a communication environment for parallel computers, ParaSoft Corporation (1988).