Applied Numerical Mathematics 11 (1993) 115–124
North-Holland

APNUM 360

Parallel implementation of BVM methods *

L. Brugnano, F. Mazzia and D. Trigiante

Dipartimento di Matematica, Università di Bari, Campus Universitario, Trav. 200 Re David. 4, I-70125 Bari, Italy

Abstract

Brugnano, L., F. Mazzia and D. Trigiante, Parallel implementation of BVM methods, Applied Numerical Mathematics 11 (1993) 115–124.

In a companion paper Lopez and Trigiante [4] introduced BVM methods for solving linear ODEs. In this paper we describe the implementation of a particular method in this class on a parallel computer. The corresponding discrete problem requires the solution of an unsymmetric block tridiagonal linear system, which is solved by means of an iterative method. A brief survey of such iterative methods is given. Comparisons of the parallel method with the LSODE package are reported, both in terms of precision of the numerical solution and speed-up, showing the effectiveness of the parallel method.

Keywords. Linear ODE; parallel computers; parallel algorithms.

1. Introduction

A class of methods (BVM methods), for solving the Cauchy problem

$$y'(t) = Ay(t) + b, \quad t \in [t_0, T],$$
$$y(t_0) = y_0, \tag{1.1}$$

is presented in [4] ($A \in \mathbb{R}^{m\times m}$). In this paper we shall see, in more detail, the implementation of a particular BVM method on a parallel computer. The chosen method utilizes the midpoint method and the implicit Euler method to handle the right boundary condition. If we use constant stepsize

$$h = \frac{T - t_0}{N},$$

and denote by y_i the approximation of $y(t_0 + ih)$, $i = 1, \ldots, N$, the discrete problem corresponding to (1.1) is easily shown to be:

$$Gy = f, \tag{1.2}$$

Correspondence to: L. Brugnano, Dipartimento di Matematica, Università di Bari, Campus Universitario, Trav. 200 Re David. 4, I-70125 Bari, Italy. Fax: (+39-80) 242722. E-mail: 00110570@vm.csata.it.

* Work supported by the European Community (P.C.A. contract #4040) and C.N.R. (P.F. "Calcolo Parallelo", sottoprogetto 1 and contract of research).

where $y=[y_1, y_2,\ldots, y_N]^T$, $f=[y_0+2hb, 2hb,\ldots,2hb, hb]^T$, and the matrix G is the following block tridiagonal one:

$$G=\begin{pmatrix} 2\hat{A} & I & & \\ -I & 2\hat{A} & \ddots & \\ & \ddots & \ddots & I \\ & & -I & I+\hat{A} \end{pmatrix}_{Nm\times Nm}, \quad \hat{A}=-hA. \tag{1.3}$$

We shall assume that the eigenvalues of A lie in the left half of the complex plane. In order to define an iterative method for solving the linear system (1.2), we need to have information about the spectrum $\sigma(G)$ of the matrix G. Let us consider the following approximation of G (which differs in the last block on the last row):

$$\tilde{G}=I_N\otimes 2\hat{A}+H\otimes I,$$

where

$$H=\begin{pmatrix} & 1 & & \\ -1 & & \ddots & \\ & \ddots & & 1 \\ & & -1 & \end{pmatrix}_{N\times N}.$$

If $\sigma(A)=\{\lambda_i\}_{i=1,\ldots,m}$, then the eigenvalues of $\tilde{G}$ are given by:

$$\mu_{ij}=-2h\lambda_i+2\mathrm{i}\cos\frac{j\pi}{N+1}, \qquad i=1,\ldots,m, \quad j=1,\ldots,N.$$

It follows that $\sigma(\tilde{G})$ lies in the right half of the complex plane. Moreover, $\sigma(G)$ and $\sigma(\tilde{G})$ are very similar: the difference of the last block of the last row makes the smallest eigenvalues in $\sigma(G)$ shift slightly to the right, moving away from the origin, while the largest eigenvalues are slightly shifted towards 1 (see Fig. 1, where the eigenvalues of G and those of $\tilde{G}$ are reported, for a given A). This implies a slightly better conditioning for the matrix G. The linear system

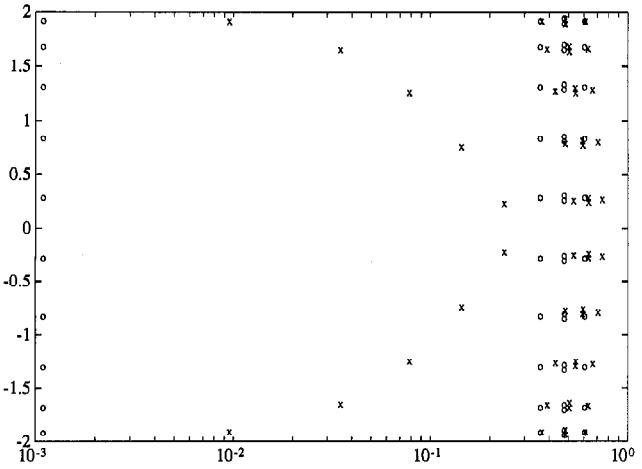

Fig. 1. Eigenvalues of G denoted by ($\times$) and $\tilde{G}$ denoted by ($\circ$).

(1.2) is solved by using an iterative method which takes advantage of this particular distribution of the eigenvalues of G. Nevertheless, in order to improve the convergence we need to cluster the spectrum of the iteration matrix. This is done by means of the following steps:

(1) first, we perform one step of block odd–even cyclic reduction on the matrix G; therefore we obtain a problem of halved size (let us suppose $N = 2K$):

$$G_{\mathrm{R}}\, y_{\mathrm{even}} f_{\mathrm{even}}, \tag{1.4}$$

where $y_{\mathrm{even}} = [y_2, y_4, \ldots, y_K]^{\mathrm{T}}$, $f_{\mathrm{even}} = [f_2, f_4, \ldots, f_K]^{\mathrm{T}}$, and

$$G_{\mathrm{R}} = \begin{pmatrix} 4\hat{A}^2 + 2I & -I & & & \\ -I & 4\hat{A}^2 + 2I & & & \\ & \ddots & \ddots & \ddots & \\ & & & 4\hat{A}^2 + 2I & -I \\ & & & -I & 2\hat{A}(\hat{A} + I) + I \end{pmatrix};$$

(2) then, we scale G_{R} by the block diagonal, obtaining the matrix $\hat{G}_{\mathrm{R}}$, whose eigenvalues are clustered around 1.

In order to examine the eigenvalue distribution of $\hat{G}_{\mathrm{R}}$ we shall consider the following approximation:

$$G_{\mathrm{R}} = I_N \otimes I - \tilde{H} \otimes (4\hat{A}^2 + 2I)^{-1},$$

where

$$\tilde{H} = \begin{pmatrix} & 1 & & \\ 1 & & \ddots & \\ & \ddots & & 1 \\ & & 1 & \end{pmatrix}_{N \times N}.$$

It follows that if $\lambda_i \in \sigma(A)$, then the eigenvalues of $\tilde{G}_{\mathrm{R}}$ (which approximate the eigenvalues of $\hat{G}_{\mathrm{R}}$) are given by:

$$\eta_{ij} = 1 - \frac{\cos(j\pi/(N+1))}{1 + 2(h\lambda_i)^2}, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, N. \tag{1.5}$$

In order to have fast convergence, we should have:

$$\sigma(\tilde{G}_{\mathrm{R}}) \subset B_{1/\delta}(1), \quad \delta \geqslant 1, \tag{1.6}$$

where $B_{1/\delta}(1)$ is the open ball with center in 1 and radius δ^{-1}: the larger the value of δ, the faster is the convergence.

From (1.5) and for $x = -\sqrt{2}h\lambda$, it follows that (1.6) will be satisfied if

$$\frac{1}{|1 + x^2|^2} < \delta^{-2},$$

or, equivalently,

$$(\alpha^2 + \beta^2)^2 + 2(\alpha^2 - \beta^2) - (\delta^2 - 1) > 0,$$

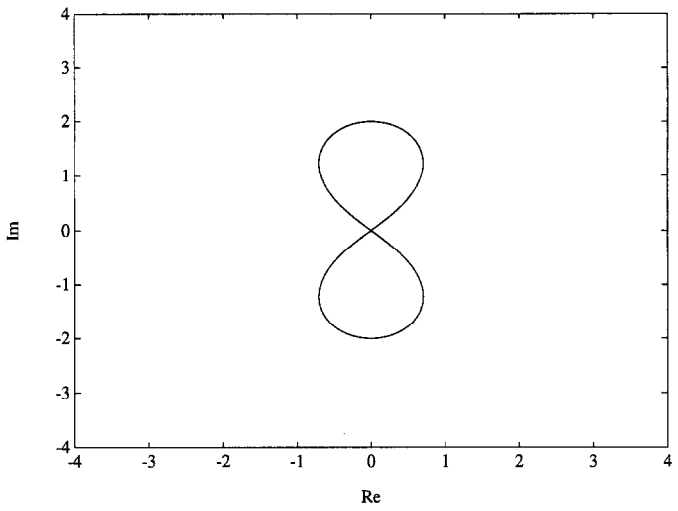

Fig. 2. Case $\delta = 1$.

where α and β are the real and the imaginary part of x, respectively. For $\delta = 1$, this means that x must lie outside the two ovals shown in Fig. 2 (in the right half plane). For $\delta > 1$, x must lie outside a region like that shown in Fig. 3. This region becomes wider as δ increases. From these pictures, we find that $|x| = |\sqrt{2}h\lambda|$ should be quite large, even for reasonably small values of δ. For example, when the eigenvalues of the matrix A are all large, the convergence of many iterative methods becomes fast (even the residual method is incredibly fast). But when matrix A has both large and small eigenvalues, it is difficult to meet the above requirement

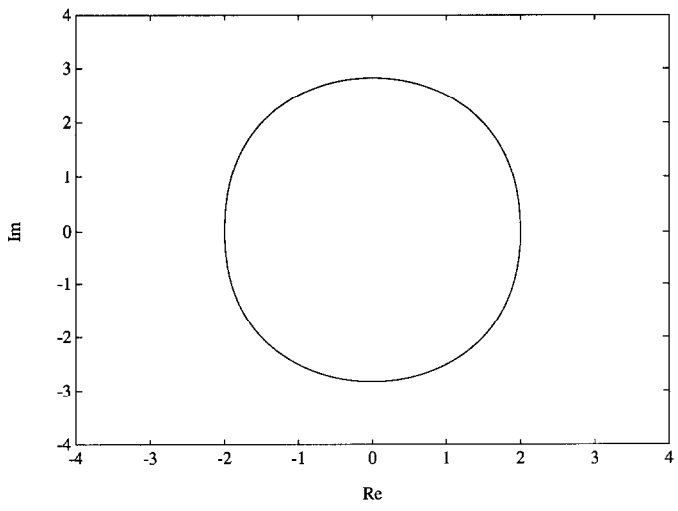

Fig. 3. Case $\delta = 3$.

(i.e. if the problem is stiff), because a large integration step will produce large local errors in the initial layer. It follows that in order to obtain both an accurate solution and fast convergence we *must* use a variable stepsize. The stepsize is then increased as indicated in [4–6]. We start with a *small* stepsize, $h_0 \approx \| A \|^{-1}$. Then, we increase it in an exponential manner as

$$h_i = \gamma^i h_0, \tag{1.7}$$

where $1 \leqslant \gamma$. With this choice, the iterative method used converges in a reasonably small number of steps, as we shall see later.

In [4] it was shown that because of the low order of the implicit Euler method, the error due to the last point may be large, even if this error doesn't have a disastrous backpropagation. This effect may be still reduced if we integrate problem (1.1) to steady state. It follows that the time interval over which we must integrate is as large as

$$\left(\max_{\lambda \in \sigma(A)} Real(\lambda) \right)^{-1}.$$

This interval may be quite large and, if the problem is stiff, h_0 must be quite small. Nevertheless, from (1.7) it follows that we can choose λ to be large enough to cover the interval of integration with a reasonably small number of steps.

2. Iterative method for the solution of unsymmetric linear systems

The conjugate gradient (CG) method is one of the most efficient iterative methods for solving large, sparse linear systems, with a symmetric positive-definite coefficient matrix, especially when used with a suitable preconditioning technique. However, in the present case the matrix of our problem is not symmetric. It follows that the algorithm is not applicable.

During the last years many generalizations of the CG method for solving unsymmetric problems have been proposed [1–3,8,9,11–13] (see [10] for a survey).

For our problem, the most efficient method proved to be CGSTAB, which we shall briefly describe in a form we think is simpler than the original formulation given in [12]. Let $Ax = b$ be the problem to be solved, where A is a square, nonsingular and unsymmetric matrix, then we consider first the following scheme where the directions are defined similarly to the corresponding ones in the conjugate gradient method:

$r_0 = b - Ax_0$; $p_{-1} = \mathbf{0}$; $q_0 = \hat{r}_0$ such that $\hat{r}_0^{\mathrm{T}} r_0 \neq 0$;
for $j = 0, 1, 2, \ldots$
$p_j = r_j + \beta_j p_{j-1}$
$r_{j+1} = r_j - \alpha_j A p_j$
$x_{j+1} = x_j + \alpha_j p_j$
$q_{j+1} = q_j - \eta_{j+1} A^{\mathrm{T}} q_j$

The α_j and β_j are determined by imposing the following *orthogonality* conditions:

$$q_j^{\mathrm{T}} A p_i = 0 \quad \text{for } j < i,$$
$$q_j^{\mathrm{T}} r_i = 0 \quad \text{for } j < i,$$

it follows then that

$$\alpha_j = \frac{q_j^{\mathrm{T}} r_j}{q_j^{\mathrm{T}} A p_j}, \qquad \beta_j = \frac{q_j^{\mathrm{T}} r_j}{q_{j-1}^{\mathrm{T}} r_{j-1}} \frac{\alpha_{j-1}}{\eta_j}.$$

The η_j will be determined later.

CGSTAB is derived from this scheme by considering that one can avoid to construct the sequence q_j. In fact one has:

$$q_j = \left(I - \eta_j A^{\mathrm{T}}\right) \cdots \left(I - \eta_1 A^{\mathrm{T}}\right) q_0$$

and

$$q_j^{\mathrm{T}} r_j = q_0^{\mathrm{T}} \prod_{k=1}^{j} (I - \eta_k A) r_j,$$

$$q_j^{\mathrm{T}} A p_j = q_0^{\mathrm{T}} A \prod_{k=1}^{j} (I - \eta_k A) p_j.$$

We may consider instead of r_j and p_j the sequences,

$$\tilde{r}_j = \prod_{k=1}^{j} (I - \eta_k A) r_j, \qquad \tilde{p}_j = \prod_{k=1}^{j} (I - \eta_k A) p_j,$$

which can be obtained iteratively by

$$\tilde{r}_j = \left(I - \eta_j A\right)\left(\tilde{r}_{j-1} - \alpha_{j-1} A \tilde{p}_{j-1}\right),$$

$$\tilde{p}_j = \tilde{r}_j + \beta_j \left(I - \eta_j A\right) \tilde{p}_{j-1}.$$

In the CGSTAB algorithm $\tilde{r}_j$ is considered to be the residual and the η_j are determined by minimizing $\| \tilde{r}_j \|_2$. From the orthogonality properties it follows that the resulting algorithm, reported below, ends in a finite number of steps:

$r_0 = b - A x_0$ (x_0 given)
$\tilde{r}_0$ such that $\tilde{r}_0^{\mathrm{T}} r_0 \neq 0$
$p_{-1} = \mathbf{0}$
$v_{-1} = \mathbf{0}$
$\rho_{-1} = 1$
$\alpha_{-1} = 1$
$\eta_0 = 1$
$j = 0$

while (*exit condition not satisfied*) do
$\rho_j = \tilde{r}_0^{\mathrm{T}} r_j$
$\beta_j = (\rho_j/\rho_{j-1})(\alpha_{j-1}/\eta_j)$
$p_j = r_j + \beta_j(p_{j-1} - \eta_j v_{j-1})$
$v_j = Ap_j$
$\sigma_j = \tilde{r}_0^{\mathrm{T}} v_j$
$\alpha_j = \rho_j/\sigma_j$
$s_j = r_j - \alpha_j v_j$
$t_j = As_j$
$\eta_{j+1} = s_j^{\mathrm{T}} t_j / t_j^{\mathrm{T}} t_j$
$x_{j+1} = x_j + \alpha_j p_j + \eta_{j+1} s_j$
$r_{j+1} = s_j - \eta_{j+1} t_j$
$j = j + 1$
end do

3. Numerical results

The numerical tests reported next are relative to the following stiff problem:

$$y'(t) = A_m y(t), \quad t \in [0, T_m],$$
$$y(t_0) = y_0, \tag{3.1}$$

where

$$A_m = \begin{pmatrix} -1800 & 900 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1000 & -2000 \end{pmatrix}_{m\times m}. \tag{3.2}$$

The BVM method previously described (which utilizes CGSTAB for solving the linear system (1.4)), implemented on a network of 32 transputers, is compared with the scalar code LSODE. Both the algorithms have been implemented in Fortran; the parallel code utilizes the Express communication library [7]. The endpoint T_m has been chosen large enough to reach the stationary solution. In all cases, the interval of integration is covered by $N = 64$ steps, so that

Table 1

m	Speed-up	LSODE fails
10	4.4	no
25	8.8	no
50	8.4	no
75	24.4	yes
100	21.1	yes

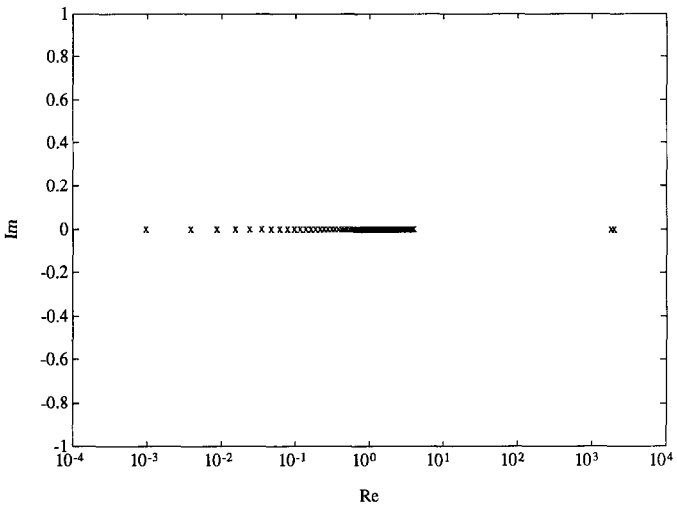

Fig. 4. Eigenvalue distribution of the matrix A ($m = 100$).

each processor handles only one block of the reduced problem (1.4). The step of cyclic reduction described in Section 1 is completely parallel, as well as the operations of the iterative method. Therefore, the code is extremely efficient, even if the used topology of interconnection among the processors is a simple linear array. The parameters used for LSODE are $mf = 21$, $rtol = 1E{-}3$, and $atol = 1E{-}7$.

In Table 1, the speed-ups on problem (3.1) of the parallel method with respect to LSODE, for $m = 10, 25, 50, 75, 100$, are reported. The cases where LSODE did not terminate properly and ended before time T_m was reached (the parallel method always worked) are also indicated. In Fig. 4 the distribution of the eigenvalues of matrix (3.2) is reported for $m = 100$. For $m = 10, 25, 50$ both methods work properly: in Fig. 5 the numerical solution computed by the parallel method (×), by LSODE (○), and—with a large expense of computer time!—by the

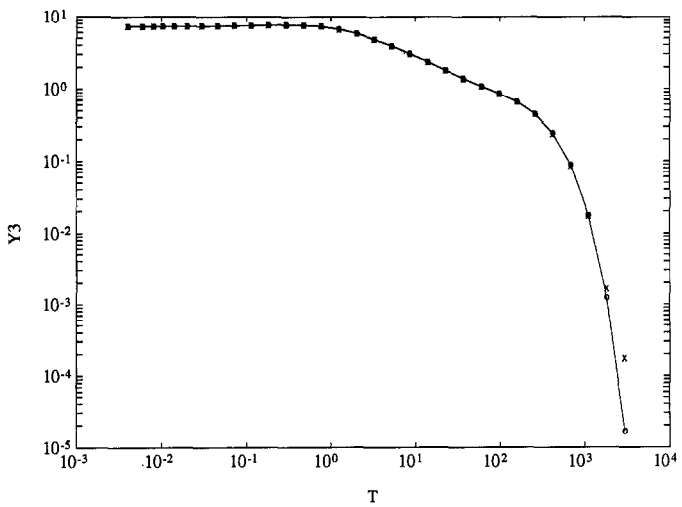

Fig. 5. Computed solution for $m = 50$: LSODE (○), parallel BVM (×), matrix exponential (—).

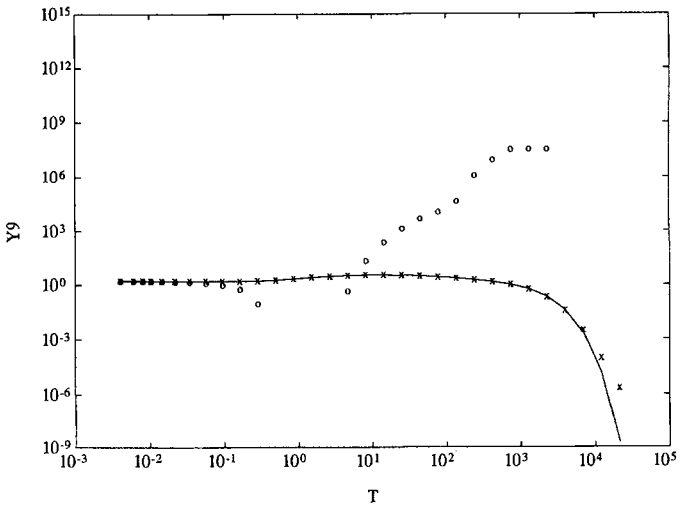

Fig. 6. Computed solution for $m = 100$: LSODE (○), parallel BVM (×), matrix exponential (—).

matrix exponential (solid line) are reported for one of the components for the problem with $m = 50$. For $m = 75, 100$ the parallel method works properly (the only slight error is in correspondence with the last points), while LSODE fails, as is shown in Fig. 6 for one of the components for the case $m = 100$.

From these results, we can conclude that parallel implementation of BVM methods seems to be very reliable and efficient for solving large systems of linear ODEs that must be integrated over a large time interval.

References

[1] O. Axelsson, Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations, *Linear Algebra Appl.* 29 (1980) 1–16.

[2] S.C. Eisenstat, H.C. Elman and M.H. Schultz, Variational iterative methods for nonsymmetric systems of linear equations, *SIAM J. Numer. Anal.* 20 (1983) 345–357.

[3] D.M. Young and K.C. Jea, Generalized conjugate-gradient acceleration of nonsymmetrizable iterative methods, *Linear Algebra Appl.* 34 (1980) 159–194.

[4] L. Lopez and D. Trigiante, Numerical boundary value techniques for initial value problems, in: *Proceedings International Conference on Parallel Methods for Ordinary Differential Equations: The State of the Art*, Grado, Italy (1991).

[5] F. Mazzia and D. Trigiante, Numerical methods for second order singular perturbation problems, *Comput. Math. Appl.* (to appear).

[6] F. Mazzia and D. Trigiante, FDSPP: a solver for singular perturbation problems (submitted).

[7] ParaSoft Corporation, Express: a communication environment for parallel computers (1988).

[8] Y. Saad, The Lanczos biorthogonalization algorithm and the other oblique projection methods for solving large unsymmetric system, *SIAM J. Numer. Anal.* 19 (1982) 485–506.

[9] Y. Saad and M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* 7 (1986) 856–869.

[10] Y. Saad and M.H. Schultz, Conjugate gradient-like algorithms for solving non symmetric linear systems, *Math. Comp.* 44 (1985) 417–424.
[11] P. Sonneveld, CGS a fast Lanczos-type solver for nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* 10 (1989) 36–52.
[12] H.A. van der Vorst and P. Sonneveld, CGSTAB: a more smoothly converging variant of CG-S, Preprint (1990).
[13] O. Widlund, A Lanczos method for a class of non-symmetric systems of linear equations, *SIAM J. Numer. Anal.* 15 (1978) 801–812.