IMA Journal of Numerical Analysis (1999) **19**, 335–348

Stability and performance analysis of a block elimination solver for bordered linear systems

PLAMEN Y. YALAMOV[†]
Centre of Applied Mathematics and Informatics, University of Rousse, 7017 Rousse, Bulgaria

AND

MARCIN PAPRZYCKI[‡]
Department of Computer Science and Statistics, University of Southern Mississippi, Hattiesburg, MS 39406, USA

[Received 24 February 1998 and in revised form 23 October 1998]

A new block elimination method for bordered systems is proposed and its numerical properties are analysed. In the case where the leading principal block is ill-conditioned or singular and the method becomes unstable a perturbation approach is used to enhance the stability. Results of experiments performed on the SGI Power Challenge 8000 and on the Cray J-9x illustrate the performance of the new algorithm and compare it with the current best approach. It is shown that the new method works faster while preserving stability.

1. Introduction

Consider a linear system of equations $Mz = b$, where $M \in \mathcal{R}^{(n+m)\times(n+m)}$ is non-singular, $z, b \in \mathcal{R}^{(n+m)\times 1}$, where

$$M = \begin{array}{c c} & \begin{array}{cc} n & m \end{array} \\ \begin{array}{c} n \\ m \end{array} & \begin{pmatrix} A & B \\ C & D \end{pmatrix} \end{array}, \quad z = \begin{array}{c} n \\ m \end{array}\begin{pmatrix} x \\ y \end{pmatrix}, \quad b = \begin{array}{c} n \\ m \end{array}\begin{pmatrix} f \\ g \end{pmatrix}.$$

We are primarily interested in the case where m is very small in comparison with n and the matrix A is sparse, structured and ill-conditioned, or singular. Such problems arise in e.g. the parallel solution of banded systems (Hajj & Skelboe, 1990), numerical continuation and bifurcation (Chan (1984), Decker & Keller (1980), Keller (1983), Werner (1996)), symmetry-breaking bifurcations (Werner & Spence, 1984), constrained optimization (Gill *et al.*, 1981) and domain decomposition methods (Barlow & Vemulapati, 1990).

Typically, a standard black-box solver is applied to matrix A. When A has a special structure (e.g. banded, block diagonal, sparse) an appropriate structure-oriented linear solver can be applied, thus significantly accelerating the solution process. However, when A is ill-conditioned or singular a standard solver is likely to produce incorrect results.

[†]Email: yalamov@ami.ru.acad.bg
[‡]Email: m.paprzycki@usm.edu

A stable solver can be applied to the whole matrix M but the computational cost (time) increases considerably because the structure of matrix A is not utilized.

Several algorithms have been proposed (Chan & Resasco (1986), Govaerts (1991), Govaerts & Pryce (1993)) that utilize information about the structure of A and allow a stable computation of the solution z. A method proposed in Chan & Resasco (1986) involves computing several singular vectors of A. In addition, at least $3m + 1$ back-solves with A are needed (in the case where A has maximum rank deficiency, see Chan & Resasco (1986) for more details). The algorithm developed by Govaerts & Pryce (1993) requires 2 solves involving A and 1 solve involving A^T at each bordering step (the number of bordering steps is equal to m). This latter algorithm is faster than the former and produces quite accurate results.

In the present paper, we propose a new algorithm for the solution of the bordered system which is based on applying perturbations to the decomposition process (similar to the methods introduced in Balle & Hansen (1993), Hansen & Yalamov (1995), for instance). This algorithm is more efficient than that of Govaerts and Pryce while having similar stability properties. The basic idea of the proposed method is to add relatively small perturbations in the factorization whenever a division by a small value is to be performed. In this way, the blowup of the solution is prevented. However, a perturbed problem is solved instead of the original one. To recover the solution of the original system iterative refinement (Golub & Van Loan (1996, §3.5.3)) is applied. This step is inexpensive because a factorization of A (e.g. LU factorization) has already been computed. In practice, one step of iterative refinement is enough to recover the solution. Its cost is equal to the cost of three triangular solves and a back-solve for a system with matrix Δ (for the definition of Δ see below), which is assumed to be of relatively small size. Hence, a solution involving A^T is not required, and the most time-consuming part of the algorithm is the LU decomposition of A. Even if additional steps of iterative refinement are needed, this does not lead to a significant increase in the computational time. The speedup of the new algorithm arises also from the fact that the level 3 BLAS (Anderson *et al.*, 1992) kernels can be applied to block operations whenever possible. The main advantage of the new approach is that it is stable when the block operations are applied, while the blocked version of the Govaerts–Pryce algorithm is quite unstable.

The outline of the paper is as follows. In Section 2 we summarize the Govaerts–Pryce algorithm. Then in Section 3 we present the new algorithm. A roundoff error analysis of this algorithm is given in Section 4. Section 5 presents numerical experiments.

2. The Govaerts–Pryce algorithm

This algorithm was originally developed for bordering of width 1 (i.e. $m = 1$ in our notation). For systems with wider borders the algorithm is applied recursively. The algorithm can be summarized as follows:

```
Step 1.  Solve $A^T V^* = C^T$.
Step 2.  Compute $\gamma^* = D - V^{*T} B$.
Step 3.  Solve $AV = B$.
Step 4.  Compute $\delta = D - CV$.
Step 5.  Compute $y_1 = (g - V^{*T} f)/\gamma^*$.
Step 6.  Compute $f_1 = f - By_1$.
Step 7.  Compute $g_1 = g - Dy_1$.
Step 8.  Solve $Aw = f_1$.
Step 9.  Compute $y_2 = (g_1 - Cw)/\gamma$.
Step 10. Compute $x = w - Vy_2$.
Step 11. Compute $y = y_1 + y_2$.
```

An analysis of the stability of the algorithm and a number of examples are given in Govaerts & Pryce (1993). The results presented there show that the algorithm is quite stable even for almost singular matrices A. But stability is guaranteed only for borders with width 1 (i.e., for wider borders the algorithm is applied recursively). It is not difficult to find numerical examples in which any block bordering leads to strong instability.

3. The perturbation approach

Our approach is based on the block LU decomposition of the matrix M:

$$M = \begin{pmatrix} PLU & 0 \\ C & I_m \end{pmatrix} \begin{pmatrix} I_n & V \\ 0 & \Delta \end{pmatrix}, \tag{1}$$

where $V = A^{-1}B$, $\Delta = D - CV$. Here we calculate $A = PLU$ by LU decomposition with partial pivoting (this is the most time-consuming part of the algorithm). If A is ill-conditioned (or singular) the matrix U has small (or zero) elements along its principal diagonal. Division by these elements would lead to the blowup of the error in the final solution.

To overcome this difficulty we perturb the diagonal entries u_{ii} of U by a small number η which will be specified later. We have

$$\tilde{u}_{ii} = u_{ii} + \delta u_{ii} = u_{ii} + \operatorname{Sgn}(u_{ii})\eta,$$

where

$$\operatorname{Sgn}(a) = \begin{cases} \operatorname{sign}(a), & \text{if } a \neq 0, \\ 1, & \text{if } a = 0. \end{cases}$$

So the growth of intermediate results is bounded to some extent, and usually we can expect some correct digits in the solution. After the perturbation is applied, the solution is computed in an obvious way by block decomposition (1).

The algorithm can be summarized as follows:

```
Step 1. Compute $PLU = A$.
Step 2. If $|u_{ii}| < \eta$, then perturb $u_{ii} = u_{ii} + \operatorname{Sgn}(u_{ii})\eta$.
Step 3. Solve $AV = B$ by $LU$ decomposition.
Step 4. Compute $\Delta = D - CV$.
Step 5. Solve $Ax_1 = f$ by $LU$ decomposition.
Step 6. Compute $y_1 = g - Cx_1$.
Step 7. Solve $\Delta y_\eta = y_1$.
Step 8. Compute $x_\eta = x_1 - Vy_\eta$.
```

The result of the algorithm is the perturbed solution $z_\eta = (x_\eta^T \; y_\eta^T)^T$.

Since the solution is perturbed we are probably not very close to the exact solution. Therefore, we apply the standard iterative refinement procedure (Golub & Van Loan (1996), Higham (1996)) to recover an accurate solution. (Since this procedure is very well known we omit its description.) Usually one step of iterative refinement is enough to obtain a solution with accuracy close to machine precision.

Now we can estimate the influence of the perturbation. At first glance the influence is not clear because we perturb the intermediate results. But from the equation

$$u_{ii} = a_{ii} - \sum_{j=1}^{i-1} l_{ij} u_{ji},$$

which defines the element u_{ii},

$$\tilde{u}_{ii} = u_{ii} + \delta u_{ii} = a_{ii} + \delta a_{ii} - \sum_{j=1}^{i-1} l_{ij} u_{ji}.$$

So, the perturbation in u_{ii} is equivalent to an equal perturbation in a_{ii}. Note, that a_{ii} is a diagonal entry in the permuted matrix A but it will be denoted a_{ii} for simplicity of notation. (This renaming does not influence our conclusions.)

Summarizing, the effect of perturbing the elements u_{ii} is equivalent to solving the original problem but with a perturbed matrix $\tilde{M} = M + \delta M$, where

$$\|\delta M\|_\infty \leqslant \eta. \tag{2}$$

The matrix δM is diagonal, with some diagonal entries nonzero and equal to $\pm\eta$. Now, we estimate the influence of the perturbation on the error and develop the roundoff error analysis. From this analysis we will derive a rule for selecting the value of η.

4. Roundoff error analysis

We consider the roundoff error analysis of our proposed algorithm. Initial results have been reported in Yalamov & Paprzycki (1998); here we present a complete analysis. We assume that backward stable solvers (in either componentwise or normwise sense) are applied to the matrices A and Δ. Here, componentwise backward stability will be assumed, but normwise backward stability leads to similar results. We have

$$(A + \varepsilon_A)\widetilde{x} = c, \quad |\varepsilon_A| \leqslant K_1 \, |A| \, \rho_0, \tag{3}$$

$$(\widetilde{\Delta} + \varepsilon_\Delta)\,\widetilde{y} = d, \quad |\varepsilon_\Delta| \leqslant K_2 \, |\widetilde{\Delta}| \, \rho_0, \tag{4}$$

where K_1 and K_2 are moderate constants (see Higham (1996, §9.2)), and $\tilde{x}$ and $\tilde{y}$ are the computed solutions. We analyze the individual stages of the algorithm, separately, i.e. the block LU decomposition and the solution of the triangular systems. Then, these results are combined to give a backward analysis of the complete algorithm.

LEMMA 1 In the presence of roundoff errors we have $L\widetilde{U} = M + E$, where

$$L = \begin{pmatrix} A & 0 \\ C & I \end{pmatrix}, \quad \widetilde{U} = \begin{pmatrix} I & \widetilde{V} \\ 0 & \widetilde{\Delta} \end{pmatrix},$$

and

$$|E| \leqslant |M| \begin{pmatrix} 0 & |\widetilde{V}| \\ 0 & I \end{pmatrix} \mu_1, \quad \mu_1 = \max\{K_1\rho_0, \gamma_{n+1}\}, \quad \gamma_{n+1} = \frac{(n+1)\,\rho_0}{1-(n+1)\,\rho_0}.$$

Proof. From

$$L\widetilde{U} = \begin{pmatrix} A & 0 \\ C & I \end{pmatrix} \begin{pmatrix} I & \widetilde{V} \\ 0 & \widetilde{\Delta} \end{pmatrix} = \begin{pmatrix} A & A\widetilde{V} \\ C & C\widetilde{V} + \widetilde{\Delta} \end{pmatrix} = \begin{pmatrix} A & B + A\delta V \\ C & D + \delta\Delta \end{pmatrix}$$
$$= M + \begin{pmatrix} 0 & A\delta V \\ 0 & \delta\Delta \end{pmatrix},$$

we obtain

$$E = \begin{pmatrix} 0 & A\delta V \\ 0 & \delta\Delta \end{pmatrix}$$

where δV is the error $\widetilde{V} - V$ and $\delta\Delta$ is the error from computing $\widetilde{\Delta}$, i.e. $\delta\Delta = \widetilde{\Delta} - D + C\widetilde{V}$.

Now we bound the entries of E. Denote the ith columns of V and B by V_i and B_i, respectively. Then from (3),

$$\left(A + \varepsilon_A^{(i)}\right)\widetilde{V}_i = B_i,$$

and after a standard manipulation we obtain

$$A\delta V_i = -\varepsilon_A^{(i)}\widetilde{V}_i, \quad |A\delta V_i| \leqslant K_1\,|A|\left|\widetilde{V}_i\right|\rho_0, \tag{5}$$

where $\delta V_i = \widetilde{V}_i - V_i$. As a result,

$$|A\delta V| \leqslant K_1\,|A|\left|\widetilde{V}_i\right|\rho_0.$$

The error $\delta\Delta$ is the result of one matrix multiplication and one matrix summation. Standard roundoff error analysis (see Higham (1996, §3.5)) produces:

$$|\delta\Delta| \leqslant \left(|D| + |C|\left|\widetilde{V}\right|\right)\gamma_{n+1}, \quad \gamma_{n+1} = \frac{(n+1)\,\rho_0}{1-(n+1)\,\rho_0}.$$

Finally,

$$|E| \leqslant \begin{pmatrix} 0 & K_1\,|A|\left|\widetilde{V}\right|\rho_0 \\ 0 & \left(|D| + |C|\left|\widetilde{V}\right|\right)\gamma_{n+1} \end{pmatrix} \leqslant |M| \begin{pmatrix} 0 & \widetilde{V} \\ 0 & I \end{pmatrix} \mu_1.$$

□

LEMMA 2 The computed solution of the block triangular system $Lu = b$ satisfies $(L + \Delta L)\,\widetilde{u} = b$, where

$$|\Delta L| \leqslant |L|\,\mu_2, \quad \mu_2 = \max\{K_1\rho_0, \gamma_n\}.$$

Proof. The solution of the block triangular system has two stages. First, the system with matrix A is solved, then a simple forward substitution is applied. For the first step, condition (3) holds. For the second part we have (see Higham (1996, §8.1))

$$|\Delta C| \leqslant |C|\,\gamma_n, \quad |\Delta I| \leqslant I\gamma_n,$$

so,

$$\Delta L = \begin{pmatrix} \Delta A & 0 \\ \Delta C & \Delta I \end{pmatrix},$$

and

$$|\Delta L| \leqslant \begin{pmatrix} K_1\,|A|\,\rho_0 & 0 \\ |C|\,\gamma_n & I\gamma_n \end{pmatrix} \leqslant |L|\,\mu_2.$$

□

LEMMA 3 The computed solution of $\widetilde{U}z = \widetilde{u}$ satisfies $(\widetilde{U} + \Delta U)\widetilde{z} = \widetilde{u}$, where

$$|\Delta U| \leqslant |\widetilde{U}|\,\mu_3, \quad \mu_3 = \max\{K_2\rho_0, \gamma_m\}.$$

Proof. The proof is similar to that of the previous lemma except that we apply condition (4) instead of (3). □

Now we combine these results to obtain the backward roundoff error analysis of the complete algorithm.

THEOREM 1 The computer solution of $Mz = b$ satisfies $(M + \Delta M)\widetilde{z} = b$, where

$$|\Delta M| \leqslant 2\,|M| \begin{pmatrix} I & |\widetilde{V}| \\ 0 & I \end{pmatrix} \mu, \quad \mu = \mu_1 + \mu_2 + \mu_3 + \mu_2\mu_3.$$

Proof. Combining Lemmas 1, 2 and 3 we obtain

$$(M + \Delta M)\widetilde{z} = (M + E + \Delta L\widetilde{U} + L\Delta U + \Delta L\Delta U)\widetilde{z} = b.$$

Applying the bounds for E, ΔL, and ΔU calculated earlier

$$|\Delta M| \leqslant |M| \begin{pmatrix} 0 & |\widetilde{V}| \\ 0 & I \end{pmatrix} \mu_1 + |L||\widetilde{U}|\,(\mu_2 + \mu_2 + \mu_2\mu_3). \tag{6}$$

Now,

$$\begin{aligned} |L|\,|\widetilde{U}| &= \begin{pmatrix} |A| & 0 \\ |C| & I \end{pmatrix} \begin{pmatrix} I & |\widetilde{V}| \\ 0 & I\widetilde{\Delta} \end{pmatrix} \\ &= \begin{pmatrix} |A| & |A||\widetilde{V}| \\ |C| & |\widetilde{\Delta}| + |C||\widetilde{V}| \end{pmatrix} \\ &\leqslant \begin{pmatrix} |A| & |A||\widetilde{V}| \\ |C| & |D| + 2\,|C||\widetilde{V}| \end{pmatrix} \\ &\leqslant 2 \begin{pmatrix} |A| & |B| \\ |C| & |C| \end{pmatrix} \begin{pmatrix} I & |\widetilde{V}| \\ 0 & I \end{pmatrix} = 2\,|M| \begin{pmatrix} I & |\widetilde{V}| \\ 0 & I \end{pmatrix}. \end{aligned} \tag{7}$$

The backward analysis in Theorem 1 holds for any matrix M. We solve the system with the matrix $M + \delta M$, $\|\delta M\|_\infty \leqslant \eta$. So, we have

$$(M + \delta M + \Delta M)\widetilde{z}_\eta = b, \tag{8}$$

where $\widetilde{z}_\eta$ is the solution computed with perturbations.

THEOREM 2 The relative forward error of the solution of (8) satisfies

$$\frac{\left\|\widetilde{z}_\eta - z\right\|_\infty}{\left\|\widetilde{z}_\eta\right\|_\infty} \leqslant 2\kappa\,(M)\,(\eta + r\mu)\,, \tag{9}$$

where the condition number $\kappa(M)$ is defined by

$$\kappa\,(M) = \max\left\{\|M^{-1}\|_\infty, \left\||M^{-1}|\,|M|\right\|_\infty\right\}, \quad r = \left\|\begin{matrix} I & |\widetilde{V}| \\ 0 & I \end{matrix}\right\|_\infty,$$

and μ is defined in Theorem 1.

Proof. From (8)

$$\widetilde{z}_\eta - z = -M^{-1}\,(\delta M + \Delta M)\,\widetilde{z}_\eta,$$

so from (2),

$$\left\|\widetilde{z}_\eta - z\right\|_\infty \leqslant \left\|M^{-1}\right\|_\infty \eta + \left\||M^{-1}|\,|\Delta M|\right\|_\infty \left\|\widetilde{z}_\eta\right\|_\infty. \tag{10}$$

We introduce the notation

$$r = \left\|\begin{matrix} I & |\tilde{V}| \\ 0 & I \end{matrix}\right\|_\infty.$$

From Theorem 1 and (10)

$$\begin{aligned}\left\|\widetilde{z}_\eta - z\right\|_\infty &\leqslant (\left\|M^{-1}\right\|_\infty \eta + 2\left\||M^{-1}|\,|M|\right\|_\infty r\mu)\left\|\widetilde{z}_\eta\right\|_\infty \\ &\leqslant 2\kappa\,(M)\,(\eta + r\mu)\left\|\widetilde{z}_\eta\right\|_\infty. \end{aligned} \tag{11}$$

□

Theorem 2 shows that the bound on the forward error is proportional to $\eta + r\mu$. The value r measures the growth of the intermediate results when computing $\widetilde{V}$. Since we have assumed that the solver for A is backward stable, the constant μ is $O\,(n\rho_0)$. In this case, the source of any large forward error must be a large r, which can happen when the matrix A is ill-conditioned. It is difficult to find exact bounds on r. Note:

1. We have assumed that A is scaled appropriately, so that it does not contain large elements.
2. We solve a system with matrix A by LU decomposition with partial pivoting. In the general case, equation (3) is (see Higham (1996, §9.2))

$$(A + \Delta A)\,\widetilde{X} = b, \quad |\Delta A| \leqslant 2\gamma_n\,|\widetilde{L}|\,|\widetilde{U}|\,,$$

where $\widetilde{L}$ and $\widetilde{U}$ are the computed factors. Because the solver is backward stable the product $|\widetilde{L}||\widetilde{U}|$ is not much larger than $|A|$, and so $|\widetilde{L}|$ and $|\widetilde{U}|$ do not have very large entries.

3. The essential growth of elements when solving a system with the matrix A arises from division by small elements on the principal diagonal of the U factor.

We can model the growth factor r by:

$$r \simeq \frac{1}{\eta^s}, \tag{12}$$

where s is difficult to estimate. We shall see in the next section that the value $s = 1$ works very well in practice.

Now, we can choose an appropriate value for η. From (9) and (12),

$$\frac{\|\tilde{z}_\eta - z\|}{\|\tilde{z}_\eta\|_\infty} \leqslant 2\kappa(\mu)\left(\eta + \frac{\mu}{\eta^s}\right) \tag{13}$$

which shows that we would have a minimal bound on the error in the perturbed solution when $\eta + \mu/\eta^s$ is minimal. The minimum is attained when $\eta = (s\mu)^{1/s+1}$. For the choice $s = 1$,

$$\eta \approx \sqrt{\mu} \approx \sqrt{n\rho_0}.$$

In the numerical experiments we have chosen η equal to $\sqrt{\rho_0}$. As a result, in most cases, we needed only one step of iterative refinement to achieve almost full precision. Thus the algorithm was stabilized cheaply.

5. Numerical tests

We present results from tests with random matrices using single precision ($\rho_0 \approx 10^{-7}$ for the SGI 8000, and $\rho_0 \approx 10^{-14}$ for the Cray J-9x). The exact solution in all examples is chosen to be $z = (1, \ldots, 1)^T$, and the value of η is the square root of the machine precision (i.e. $\approx 10^{-4}$ for the SGI 8000, and $\approx 10^{-7}$ for the Cray J-9x). We have run our experiments also on the SGI 10000. Since the results were quite similar to those obtained on the SGI 8000 only the latter ones will be reported.

The matrices A are generated randomly as follows:

$$A = H_1 \ldots H_{100}\,\mathrm{diag}(0, 0, 0, 0{\cdot}7 + 0{\cdot}04n, 0{\cdot}7 + 0{\cdot}04(n-1), \ldots, 0{\cdot}86)H_{101} \ldots H_{200},$$

where

$$H_i = I - 2h_i h_i^T,$$

and h_i is a vector of unit length whose entries are random and uniformly distributed in [0,1]. Matrices of this type are used in the experiments reported in Govaerts & Pryce (1993). The sparsity and the structure of A can vary in practice. Here, we consider the two (almost) limiting cases: a dense matrix and a tridiagonal matrix.

Experiments on the SGI 8000 are reported in Figs 1–4 for dense (Figs 1–2) and tridiagonal (Figs 3–4) matrices A, and for $m = 1, 2, \ldots, 25$ and $n = 500$, and $m = 1, 2, \ldots, 50$, $n = 1000$. In a similar way, Figs 5–8 show the results on the Cray J-9x for $m = 1, 2, \ldots, 25$ and $n = 500$, and $m = 1, 2, \ldots, 50$, $n = 1000$. In each figure, the first diagram compares the computational times for the Govaerts–Pryce algorithm (solid line)

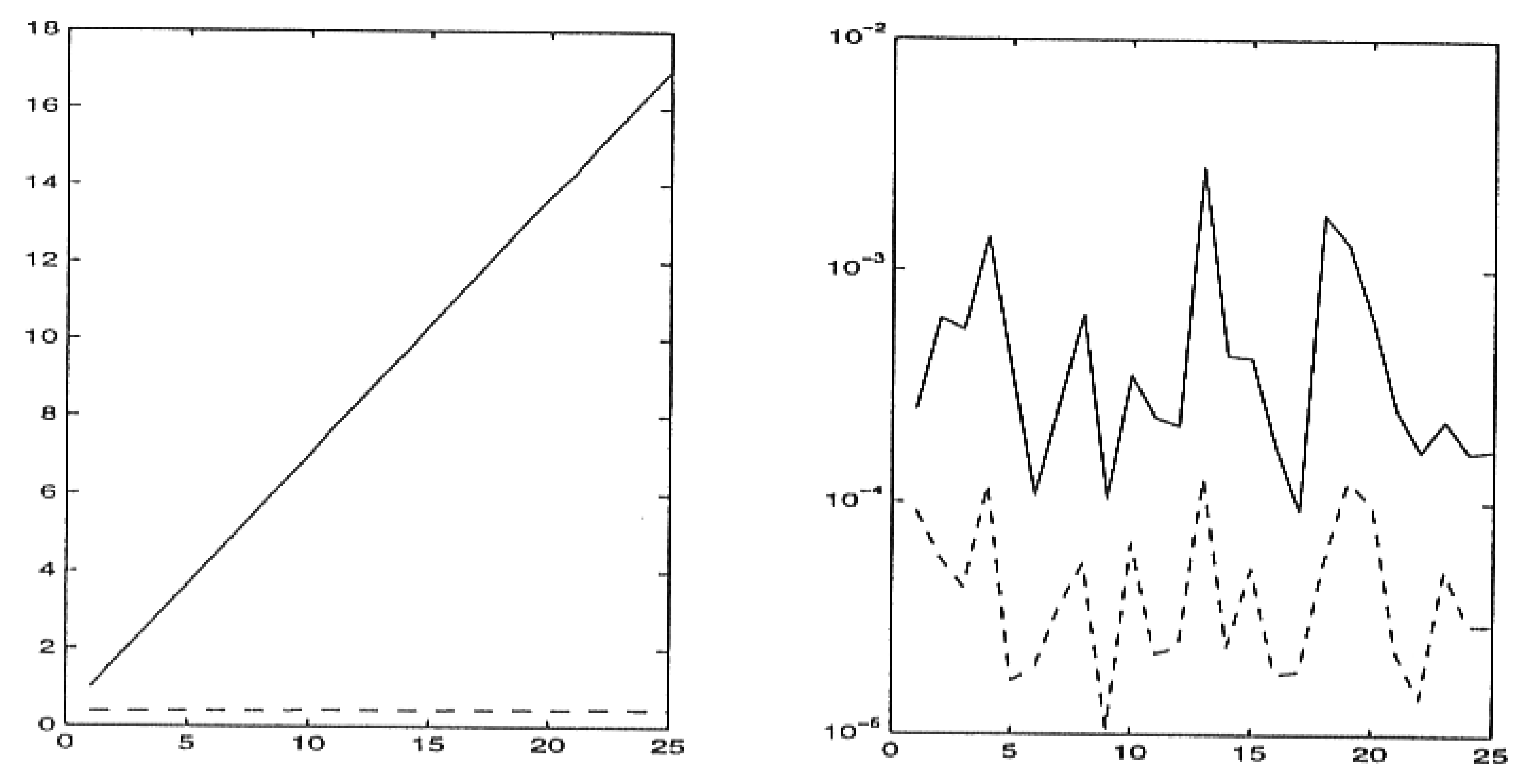

FIG. 1. The computational time (left) and the error (right) on the SGI 8000 for both algorithms: Govaerts–Pryce (continuous line), perturbation approach (dashed line); matrix A dense, $n = 500$, $m = 1, \dots, 25$.

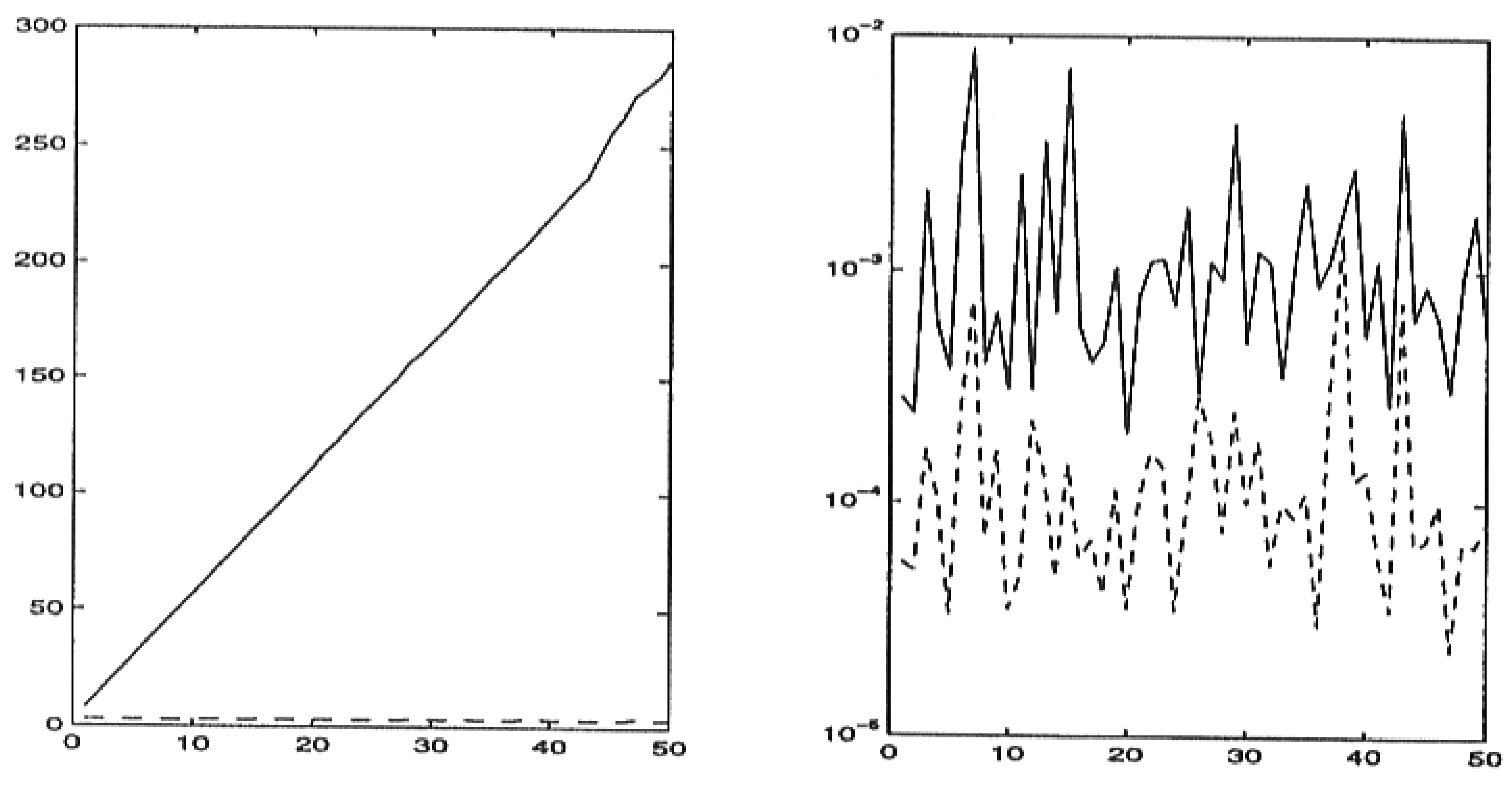

FIG. 2. The computational time (left) and the error (right) on the SGI 8000 for both algorithms: Govaerts–Pryce (continuous line), perturbation approach (dashed line); matrix A dense, $n = 1000$, $m = 1, \dots, 50$.

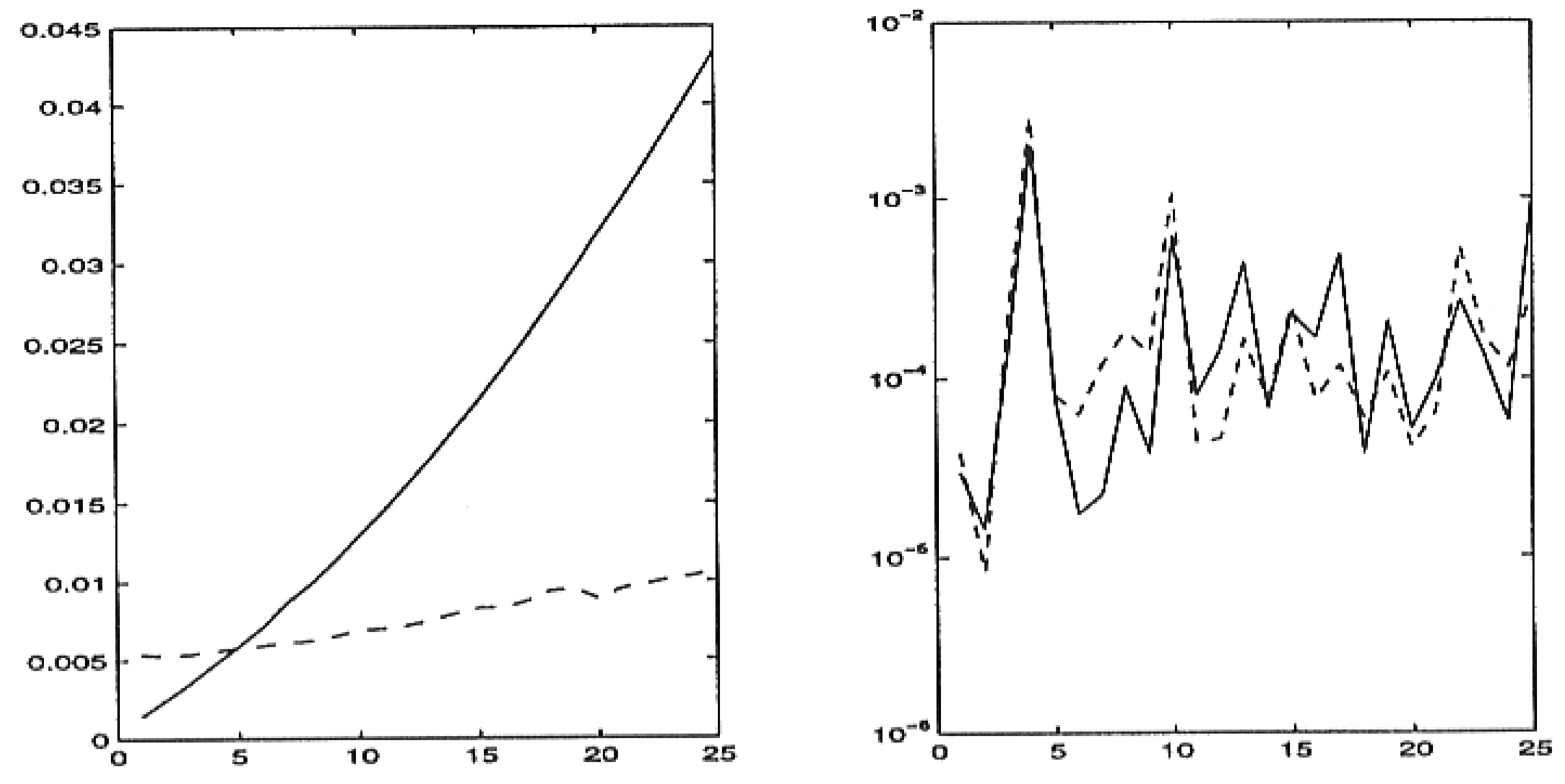

FIG. 3. The computational time (left) and the error (right) on the SGI 8000 for both algorithms: Govaerts–Pryce (continuous line), perturbation approach (dashed line); matrix A tridiagonal, $n = 500$, $m = 1, \ldots, 25$.

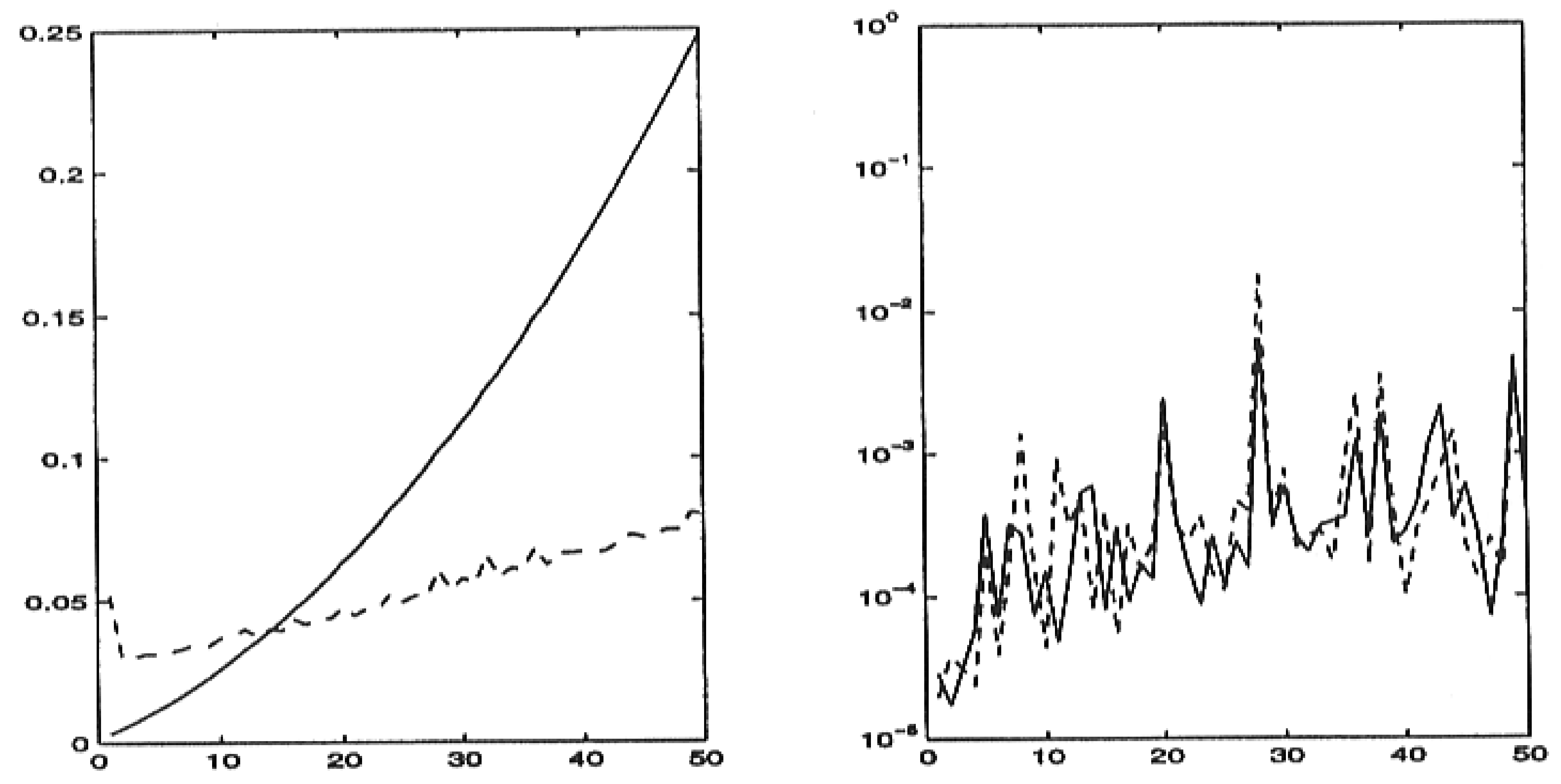

FIG. 4. The computational time (left) and the error (right) on the SGI 8000 for both algorithms: Govaerts–Pryce (continuous line), perturbation approach (dashed line); matrix A tridiagonal, $n = 1000$, $m = 1, \ldots, 50$.

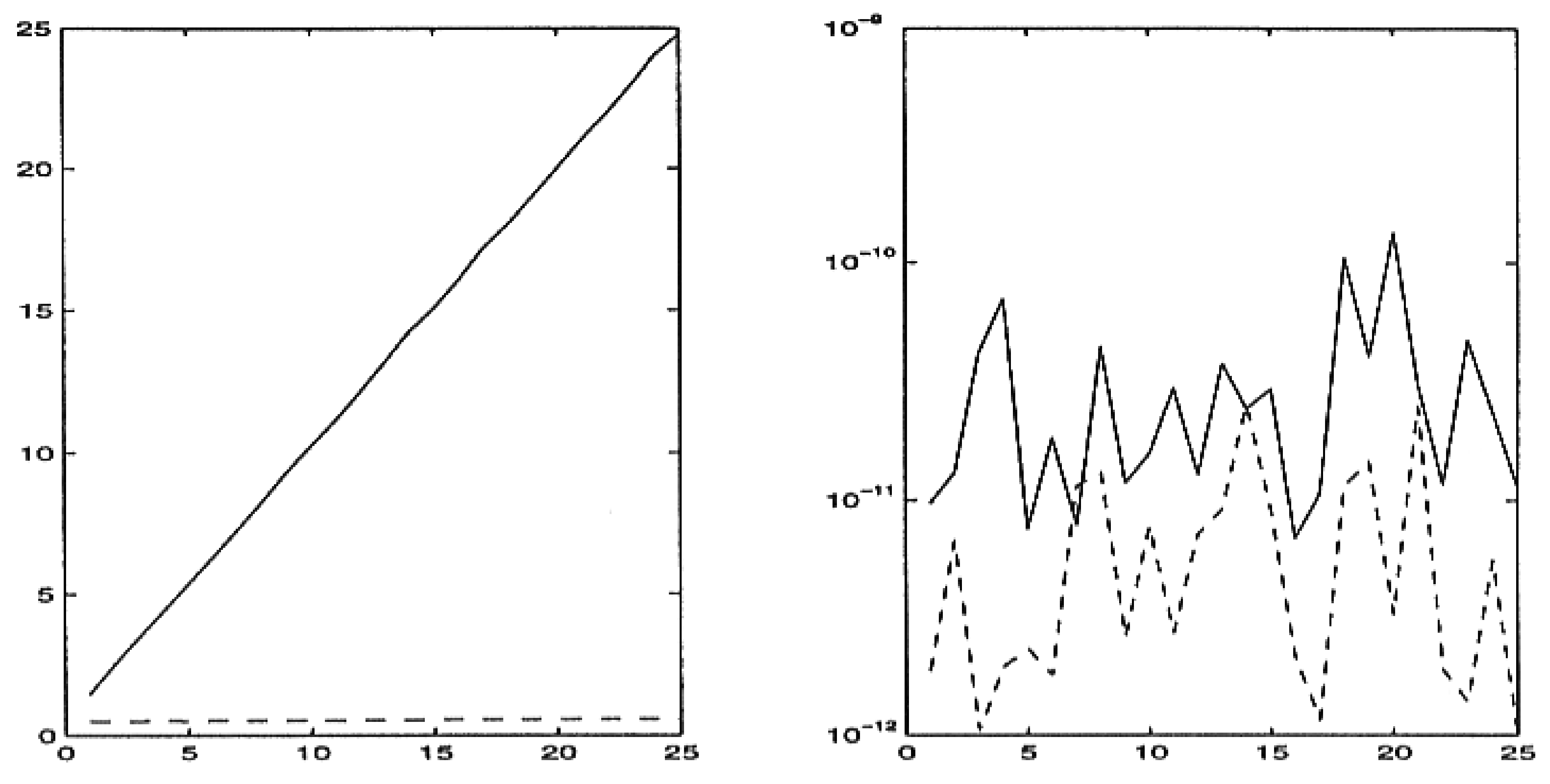

FIG. 5. The computational time (left) and the error (right) on the Cray J-9x for both algorithms: Govaerts–Pryce (continuous line), perturbation approach (dashed line); matrix A dense, $n = 500$, $m = 1, \ldots, 25$.

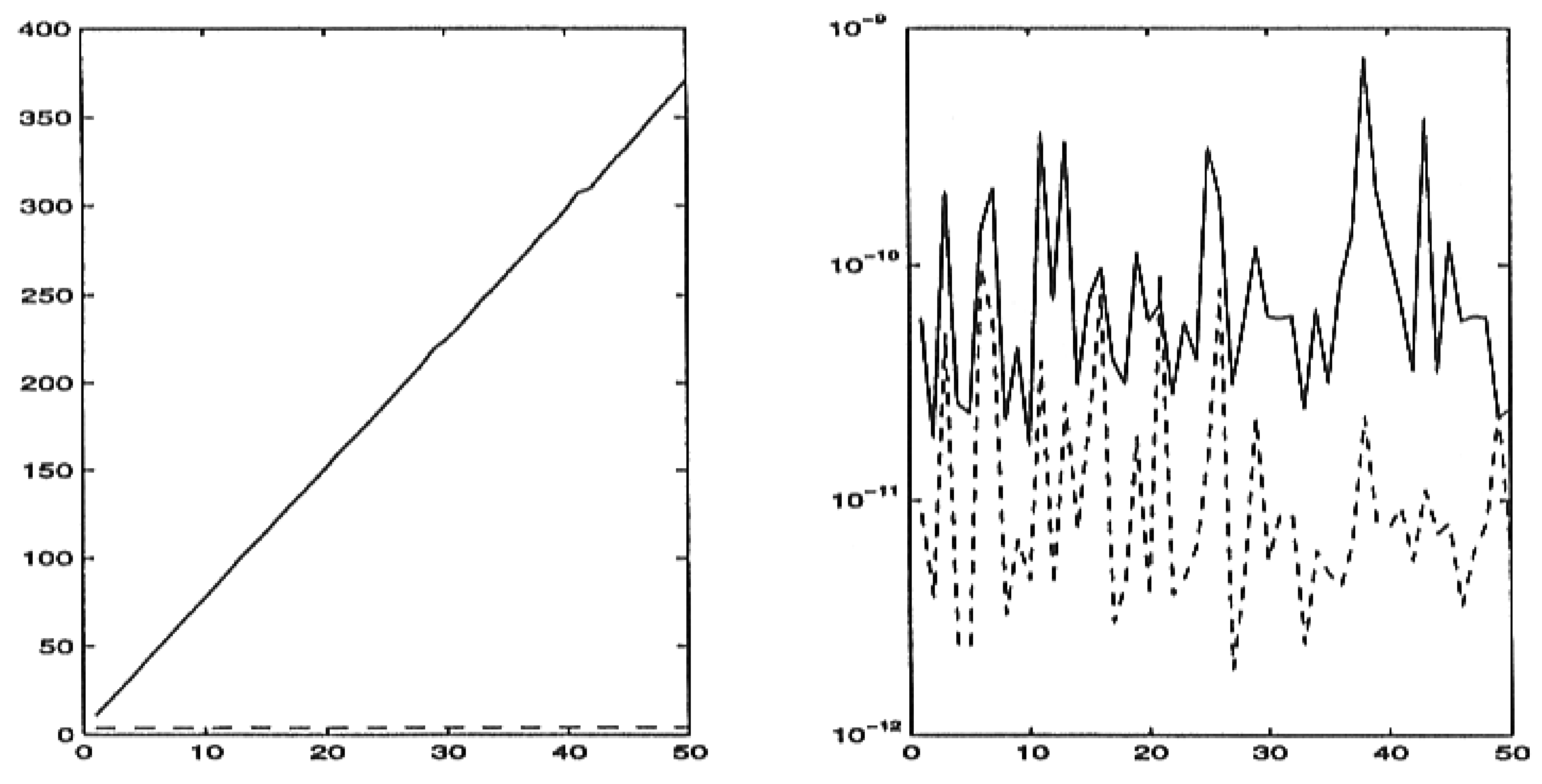

FIG. 6. The computational time (left) and the error (right) on the Cray J-9x for both algorithms: Govaerts–Pryce (continuous line), perturbation approach (dashed line); matrix A dense, $n = 1000$, $m = 1, \ldots, 50$.

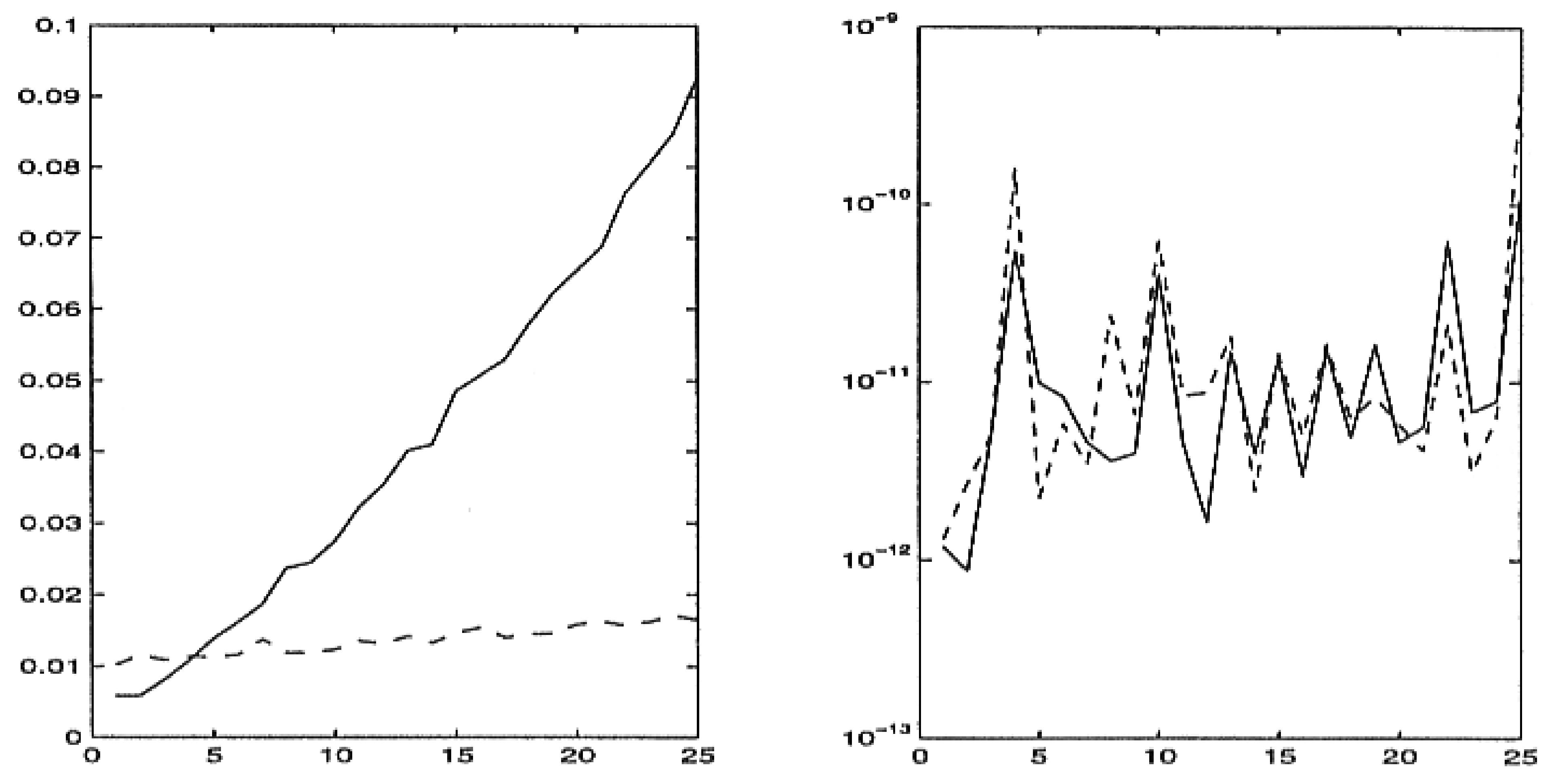

FIG. 7. The computational time (left) and the error (right) on the Cray J-9x for both algorithms: Govaerts–Pryce (continuous line), perturbation approach (dashed line); matrix A tridiagonal, $n = 500$, $m = 1, \ldots, 25$.

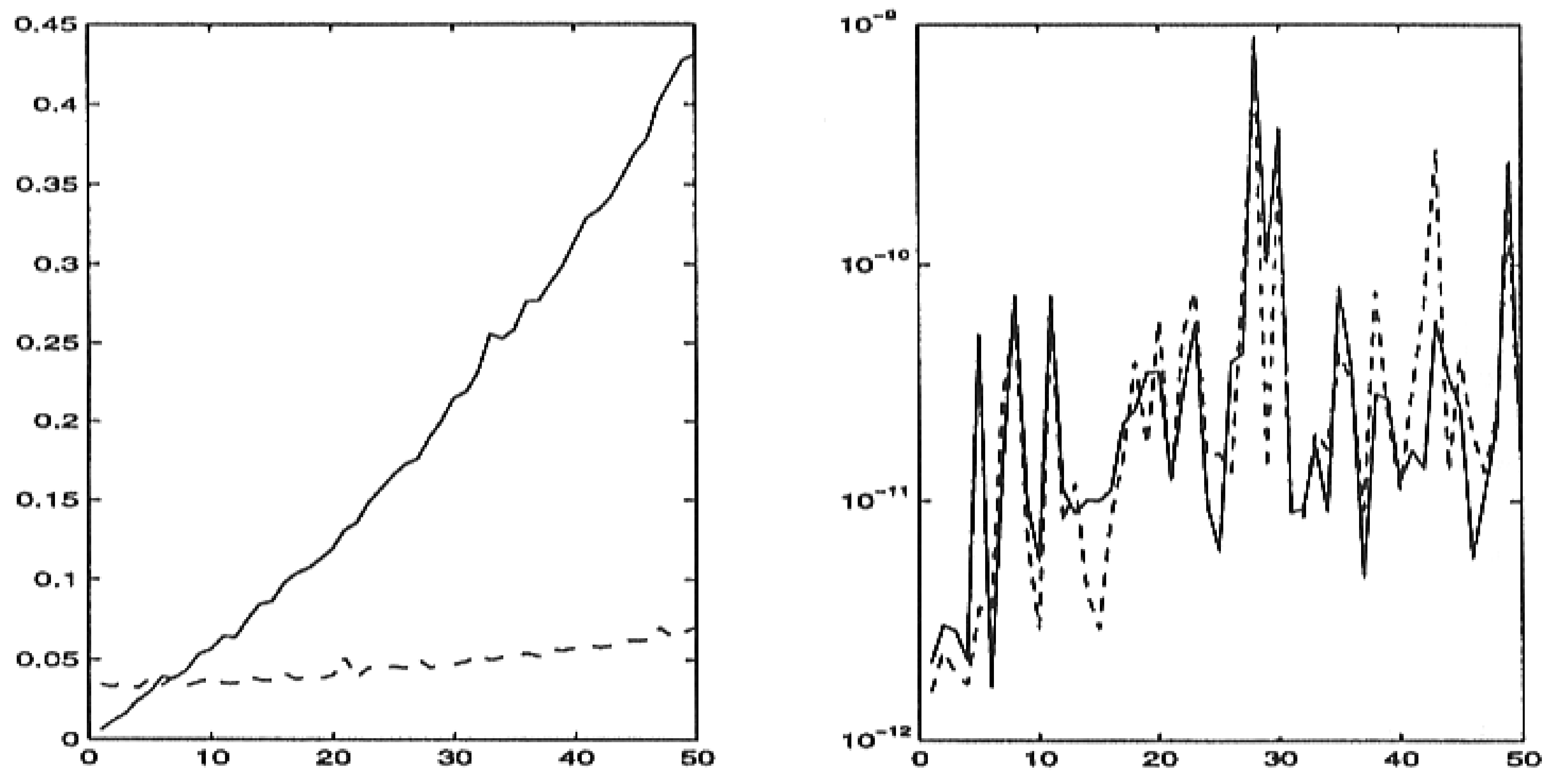

FIG. 8. The computational time (left) and the error (right) on the Cray J-9x for both algorithms: Govaerts–Pryce (continuous line), perturbation approach (dashed line); matrix A tridiagonal, $n = 1000$, $m = 1, \ldots, 50$.

and the perturbation approach (dashed line). The second diagram presents the forward error $\|\tilde{z} - z\|_\infty / \|z\|_\infty$ in the computed solution $\tilde{z}$ for the two algorithms.

From these results we draw the following observations:

1. The computational time for the Govaerts–Pryce algorithm grows linearly with m, while the computational time of our approach is almost constant as m increases. This is due to the speed of the blocked level 3 BLAS-based operations. As the blocksize increases their performance also increases and is able to partially compensate for the computational complexity introduced with the increasing m.
2. The perturbation approach is several times faster for dense matrices A, and the speedup is almost linear with m. In most cases, the perturbation approach needs only one step of iterative refinement. In our experiments there were only few cases when the number of iterations was 2, 4 or 5. Even then the perturbation approach remained several times faster, as can be seen from Figs 1, 2, 5 and 6.
3. For the tridiagonal matrices, the perturbation approach is slightly slower than the Govaerts–Pryce algorithm for small values of m. For larger values of m, we obtain a reasonable speedup. It seems that the critical value of m varies between machines. From Figs 4 and 5 we see that on the Cray this critical value is small, and on the SGI 8000 it is relatively large. This requires further study.
4. The error in both algorithms is similar. In most cases (especially for dense matrices A) the perturbation approach produces a slightly smaller error.
5. Essentially the error does not vary with m.
6. The perturbation approach is essentially independent of the choice of η as long as η is located in a relatively large range around machine precision. This result is slightly surprising and we plan to investigate it further.

6. Conclusions

We have presented a new perturbation-based method for solving bordered linear systems. This approach has been shown to work faster than the well-known algorithm of Govaerts and Pryce while preserving similar numerical stability. Future research will concentrate on several areas: the choice of the value of η; establishing a critical value m for which the new algorithm outperforms the Govaerts–Pryce approach; and applying the approach to the parallel solution of bordered systems.

Acknowledgements

We would like to express our gratitude to Ian Gladwell for his comments which allowed us to significantly improve an earlier version of this paper. We are also grateful to the two referees who helped us to improve the presentation of the paper. Finally, we would like to thank Katarzyna Paprzycka for her help with English.

This work was supported partially by grants MM-707/97 and I-702/97 from the Bulgarian Ministry of Education and Science. The work was initiated under the COBASE program from the National Research Council. Computer time grants from NPACI at UT Austin and NCSA in Urbana-Champain are kindly acknowledged.

REFERENCES

ANDERSON, E., BAI, Z., BISCHOF, C., DEMMEL, J., DONGARRA, J., CROZ, J. D., GREENBAUM, A., HAMMARLING, S., MCKENNEY, A., OSTROUCHOV, S., & SORENSEN, D. 1992 *LAPACK Users' Guide*. Philadelphia: SIAM.

BALLE, S. M. & HANSEN, P. C. 1993 A Strassen-type matrix inversion algorithm for the Connection Machine. *Report UNIC-93-11*.

BARLOW, J. & VEMULAPATI, U. 1990 An improved method for one-way dissection with singular diagonal blocks. *SIAM J. Matrix Anal. Appl.* **11**, 575–588.

CHAN, T. F. 1984 Newton-like pseudo-arclength methods for computing simple turning points. *SIAM J. Sci. Stat. Comput.* **5**, 135–148.

CHAN, T. F. & RESASCO, D. C. 1986 Generalized deflated block elimination. *SIAM J. Numer. Anal.* **23**, 913–924.

DECKER, D. W. & KELLER, H. B. 1980 Multiple limit point bifurcation. *J. Math. Anal. Appl.* **75**, 417–430.

GILL, P. E., MURRAY, W., & WRIGHT, M. 1981 *Practical Optimization*. New York: Academic.

GOLUB, G. & VAN LOAN, C. 1996 *Matrix Computations* 3rd edn. Baltimore, MD: John Hopkins University Press.

GOVAERTS, W. 1991 Stable solvers and block elimination for bordered systems. *SIAM J. Matrix Anal. Appl.* **12**, 469–483.

GOVAERTS, W. & PRYCE, J. D. 1993 Mixed block elimination for linear systems with wider borders. *IMA J. Numer. Anal.* **13**, 161–180.

HAJJ, I. N. & SKELBOE, S. 1990 A multilevel parallel solver for block tridiagonal and banded linear systems. *Parallel Comput.* **15**, 21–45.

HANSEN, P. C. & YALAMOV, P. Y. 1995 Stabilization by perturbation of a $4n^2$ Toeplitz solver. *Preprint N25*, Technical University of Russe, January.

HIGHAM, N. J. 1996 *Accuracy and Stability of Numerical Algorithms*. Philidelphia: SIAM.

KELLER, H. B. 1983 The bordering algorithm and path following near singular points of higher nullity. *SIAM J. Sci. Stat. Comput.* **4**, 573–582.

WERNER, B. 1996 Computation of Hopf bifurcation with bordered matrices. *SIAM J. Numer. Anal.* **33**, 435–455.

WERNER, B. & SPENCE, A. 1984 The computation of symmetry-breaking bifurcations. *SIAM J. Numer. Anal.* **21**, 388–399.

WILKINSON, J. H. 1965 *The Algebraic Eigenvalue Problem*. Oxford: Clarendon.

YALAMOV, P. Y. 1994 The bordering and the block bordering method. *Advances in Parallel Computing* (I. Dimov and O. Tonev, eds). IOS Press, pp 60–65.

YALAMOV, P. Y. & PAPRZYCKI, M. 1998 A new stable solver and block elimination for bordered systems. *Large Scale Scientific Computations of Engineering and Environmental Problems* (*Notes on Numerical Fluid Dynamics* 62) (M. Griebel, O. Iliev, S. Margenov and P. Vassilevski, eds). Vieweg, pp 347–356.