Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 401059, 12 pages
doi:10.1155/2012/401059

Research Article

A Preconditioned Iteration Method for Solving Sylvester Equations

Jituan Zhou,[1] Ruirui Wang,[2] and Qiang Niu[3]

[1] *School of Mathematics and Computational Science, Wuyi University, Guangdong, Jiangmen 529000, China*
[2] *College of Science, China University of Mining and Technology, Xuzhou 221116, China*
[3] *Mathematics and Physics Centre, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China*

Correspondence should be addressed to Ruirui Wang, doublerui612@gmail.com and Qiang Niu, kangniu@gmail.com

Received 25 May 2012; Accepted 20 June 2012

Academic Editor: Jianke Yang

A preconditioned gradient-based iterative method is derived by judicious selection of two auxiliary matrices. The strategy is based on the Newton's iteration method and can be regarded as a generalization of the splitting iterative method for system of linear equations. We analyze the convergence of the method and illustrate that the approach is able to considerably accelerate the convergence of the gradient-based iterative method.

1. Introduction

In this paper, we consider preconditioned iterative methods for solving Sylvester equations of form

$$AX + XB = C, \tag{1.1}$$

where $A \in \mathcal{R}^{m\times m}$, $B \in \mathcal{R}^{n\times n}$, and $X = [x_1 \cdots x_2] \in \mathcal{R}^{m\times n}$, with $m \gg n$ generally. A Lyapunov equation is a special case of (1.1) with $m = n$, $B = A^T$, and $C = C^T$. Such kind of problems frequently arise from many areas of applications in control and system theory [1], stability of linear systems [2], analysis of bilinear systems [3], power systems [4], signal and image processing [5], and so forth.

Throughout the paper, we assume that Sylvester equation (1.1) possess a unique solution, that is,

$$\lambda(A) \cap \lambda(B) = \emptyset, \tag{1.2}$$

where $\lambda(A)$ and $\lambda(B)$ denote the spectra of A and B, respectively [6]. In theory, the exact solution of (1.1) can be computed by “linearization,” that is, by solving an equivalent system of linear equations of form

$$\mathcal{A}\,\mathrm{vec}(X) = \mathrm{vec}(C), \tag{1.3}$$

where $\mathcal{A} = I_n \otimes A + B^T \otimes I_m \in \mathcal{R}^{mn\times mn}$, $\mathrm{vec}(X) = [x_1^T, \dots, x_n^T]^T$ with $X = [x_1, \dots, x_n] \in \mathcal{R}^{m\times n}$, and $\otimes$ is the Kronecker product. However, the above direct method requires considerable computational efforts, due to the high dimension of the problem.

For small-to-medium-scale problems of the form (1.1), direct approaches such as Bartels-Stewart method [7, 8] and Hessenberg-Schur method [6, 9] have been the methods of choice. The main idea of these approaches is to transform the original linear system into a structured system that can be solved efficiently by forward or backward substitutions.

In the numerical linear community, iterative methods are becoming more and more popular. Several iterative schemes for Sylvester equations have been proposed; see, for example, [10–15]. Recently, Some gradient based iterative methods [3–5, 16–26] have been investigated for solving general coupled matrix equations and general matrix equations. For Sylvester equations of form (1.1), the gradient based iterative methods use a hierarchical identification principle to compute the approximate solution. The convergence condition of these methods is investigated in [16–18]. It is proved that the gradient based iterative methods are convergent under certain conditions. However, we observe that the convergence speed of the gradient based iterative methods is generally very slow, which is similar to the behavior of classical iterative methods applied to systems of linear equations. In this paper, we consider preconditioning schemes for solving Sylvester equations of form (1.1). We illustrated that the preconditioned gradient based iterative methods can be derived by selecting two auxiliary matrices. The selection of preconditioners is natural from the view point of splitting iteration methods for systems of linear equations. The convergent property of the preconditioned method is proved and the optimal relaxation parameter is derived. The performance of the method is compared with the original method in several examples. Numerical results show that preconditioning is able to considerably speed up the convergence of the gradient based iterative method.

The paper is organized as follows. In Section 2, a gradient based iterative method is recalled, and the preconditioned gradient based method is introduced and analyzed. In Section 3, performance of the preconditioned gradient based method is compared with the unpreconditioned one, and the influence of an iterative parameter is experimentally studied. Finally, we conclude the paper in Section 4.

2. A Brief Review of the Gradient Based Iterative Method

We firstly recall an iterative method proposed by Ding and Chen [18] for solving (1.1). The basic idea is regarding (1.1) as two linear matrix equations:

$$AX = C - XB, \qquad XB = C - AX. \tag{2.1}$$

Then define two recursive sequences

$$X_k^{(1)} = X_{k-1}^{(1)} + \kappa A^T \left(C - AX_{k-1}^{(1)} - X_{k-1}^{(1)} B \right), \tag{2.2}$$

$$X_k^{(2)} = X_{k-1}^{(2)} + \kappa \left(C - AX_{k-1}^{(2)} - X_{k-1}^{(2)} B \right) B^T, \tag{2.3}$$

where κ is the iterative step size. The above procedures can be regarded as two separate iterative procedures for solving two matrix equations in (2.1).

With $X_k^{(1)}$ and $X_k^{(2)}$ at hand, then the kth approximate solution X_k can be defined by taking the average of two approximate solutions, that is,

$$X_k = \frac{X_k^{(1)} + X_k^{(2)}}{2}. \tag{2.4}$$

By selecting an appropriate initial approximate solution X_0, and using X_{k-1} to substitute $X_{k-1}^{(1)}$ in (2.2) and $X_{k-1}^{(2)}$ in (2.3), then the above (2.2)-(2.3) constitute the gradient based iterative method proposed in [18]. It is shown [18] that the gradient based iterative algorithm converges as long as

$$0 < \kappa < \frac{2}{\lambda_{\max}\left(AA^T\right) + \lambda_{\max}\left(B^T B\right)}, \tag{2.5}$$

where $\lambda_{\max}(AA^T)$ is the largest eigenvalue of AA^T.

3. Preconditioned Gradient Based Iterative Method

We start with a general algebraic equation [27]

$$f(x) = 0, \quad x \in \mathcal{R}. \tag{3.1}$$

Suppose x_n is an approximate solution for (3.1), for example, $f(x_n) \approx 0$ ($f(x_n) \neq 0$). Then it follows that the accuracy of x_n can be improved by the following scheme:

$$x_{n+1} = x_n + \lambda f(x_n) \tag{3.2}$$

with λ being a Lagrangian multiplier. It is well known that the optimal value of λ is determined by

$$\frac{dx_{n+1}}{dx_n} = 0. \tag{3.3}$$

From the above condition, the Newton's iteration follows:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{3.4}$$

Now, let us consider a general system of linear equations of form

$$Kx = b, \tag{3.5}$$

where $K \in \mathcal{R}^{n\times n}$ is a general square matrix. Let $f(x) = b - Ax$, and x_0 be an initial approximate solution. Then by one-step Newton's iteration (3.4), we can theoretically find the exact solution

$$x_1 = x - K^{-1}(b - Kx) = K^{-1}b. \tag{3.6}$$

However, the previous procedure has the same difficulty as solving (3.5) by direct inverse.

By utilizing the idea of preconditioning, the iteration formula (3.6) can be rebuilt by replacing K^{-1} by an approximate matrix M^{-1}. Taking x_0 as a starting vector and $r_0 = b - Ax_0$, then it follows that

$$x_{k+1} = x_k + M^{-1}r_k, \quad k = 0, 2, \ldots, \tag{3.7}$$

which is the preconditioned iterative method. Formula (3.7) can also be derived from the splitting iteration [28] with

$$A = M - N. \tag{3.8}$$

More generally, a relaxation parameter can be introduced as follows:

$$x_{k+1} = x_k + \kappa M^{-1}r_k, \quad k = 0, 2, \ldots. \tag{3.9}$$

Recall that Sylvester equations (1.1) can be rewritten as the following two fictitious matrix equations:

$$\begin{aligned} AX &= E_1, \\ XB &= E_2, \end{aligned} \tag{3.10}$$

where $E_1 = C - XB$ and $E_2 = C - AX$.

By (3.9),we define preconditioned iterations as follows:

$$\widetilde{X}_k = X_{k-1} + \kappa M_1^{-1}(C - AX_{k-1} - X_{k-1}B), \tag{3.11}$$

$$\widehat{X}_k = X_{k-1} + \kappa(C - AX_{k-1} - X_{k-1}B)M_2^{-1}, \tag{3.12}$$

where M_1 and M_2 are well-chosen preconditioners for A and B, respectively. The kth approximate solution can be defined by taking the average of two approximate solutions [16, 18], for example,

$$X_k = \frac{\widetilde{X}_k + \widehat{X}_k}{2}. \tag{3.13}$$

By selecting two initial approximate solutions, the above procedures (3.11)–(3.13) constitute the framework of the Newton's iteration method.

The above process can be accomplished by the following algorithm.

Algorithm 3.1. Preconditioned gradient based iterative algorithm (PGBI) can be given as follows.

```
(1) Give two initial approximate solutions $\widetilde{X}_0$ and $\widehat{X}_0$.
(2) for $k = 1, 2, \ldots$, until converges
(3) $X_{k-1} = (\widetilde{X}_{k-1} + \widehat{X}_{k-1})/2$
(4) $\widetilde{X}_k = X_{k-1} + \kappa M_1^{-1}(C - AX_{k-1} - X_{k-1}B)$
(5) $\widehat{X}_k = X_{k-1} + \kappa(C - AX_{k-1} - X_{k-1}B)M_2^{-1}$
(6) end.
```

Remark 3.2. (1) The above algorithm follows the framework of the gradient based iterative method proposed in [16–18]. However, in [16, 18] the matrix M_1^{-1} is chosen as A^T ($M_2^{-1} = B^T$), and in [17] the matrix M_1^{-1} is chosen as A^TA ($M_2^{-1} = B^TB$). The previous selection of the matrices is obviously not wise. In this paper, we have illustrated that M_1 and M_2 should be the reasonable preconditioner for A and B, respectively.

(2) Replacing $C - AX_{k-1} - X_{k-1}B$ by $C - AX_{k-1}B - X_{k-1}$, then the above algorithm can be used to solve generalized Sylvester equation of form

$$AXB + X = C. \tag{3.14}$$

We will present an example of form (3.14) in the last section of the paper.

Lemma 3.3 (see [29]). *Leting $A \in \mathcal{R}^{n\times n}$, then A is convergent if and only if $\rho(A) < 1$.*

Based on Theorem 3.4 in [18], we have the following theorem.

Theorem 3.4. *Suppose Sylvester equation* (1.1) *has a unique solution X, and*

$$\max_i \left\{ \left| 1 - \frac{\kappa\lambda_i}{2} \right| \right\} < 1 \tag{3.15}$$

then the iterative sequence X_k generated by Algorithm 3.1 converges to X, that is, $\lim_{k\to\infty} X_k = X$ *converges to zero for any initial value $X(0)$.*

Proof. Since $X_k = 1/2(\widetilde{X}_k + \widehat{X}_k)$, it follows that

$$X_k = X_{k-1} + \frac{\kappa}{2} M_1^{-1}(C - AX_{k-1} - X_{k-1}B) + \frac{\kappa}{2}(C - AX_{k-1} - X_{k-1}B)M_2^{-1}. \tag{3.16}$$

Letting $E_k = X - X_k$ and submitting $C = AX - XB$ into the above formula, we have

$$X_k = X_{k-1} + \frac{\kappa}{2} M_1^{-1}(AE(k-1) - E(k-1)B) + \frac{\kappa}{2}(AE(k-1) - E(k-1)B)M_2^{-1}. \tag{3.17}$$

Using X to subtract both sides of the above equation, we have

$$E_k = E_{k-1} - \frac{\kappa}{2} M_1^{-1}(AE_{k-1} - E_{k-1}B) - \frac{\kappa}{2}(AE_{k-1} - E_{k-1}B)M_2^{-1}. \tag{3.18}$$

Let vec(X) be defined as the vector formed by stacking the columns of X on the top of one another, that is,

$$\operatorname{vec}(X) = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}. \tag{3.19}$$

Then (3.18) is equivalent to

$$\operatorname{vec}(E_k) = \left(I - \frac{\kappa}{2}\Phi\right)\operatorname{vec}(E_{k-1}), \tag{3.20}$$

where $\Phi = I \otimes (M_1^{-1}A) - B^T \otimes M_1^{-1} + M_2^{-T} \otimes A - (M_2^{-T}B^T) \otimes I$. Therefore, the iteration is convergent if and only if $\rho(I - (\kappa/2)\Phi) < 1$, that is,

$$\max_i\left\{\left|1 - \frac{\kappa\lambda_i}{2}\right|\right\} < 1. \tag{3.21}$$

The proof is complete. □

Remark 3.5. The choice of parameter κ is an important issue. We will experimentally study its influence on the convergence. However, the parameter is problem dependent; so seeking a parameter that is suitable for a broad range of problems is a difficult task.

4. Numerical Examples

In the following tests, the parameter κ is set to be

$$\frac{1}{\max(\operatorname{eig}(AA^T) + \max(\operatorname{eig}(BB^T)))} \tag{4.1}$$

in the gradient based iterative method (GBI), and κ is chosen experimentally in the preconditioned gradient based iterative method (PGBI). In all the test, the stopping criterion is set to be $1e{-}6$. The exact solution is set as $X = \text{rand}(m,n){+}\text{speye}(m,n){*}2$ such that the right hand side $C = AX{+}XB$, and the initial approximate solution is set to be $X_0 = \text{ones}(m,n){*}1e{-}6$. The relative error norms are recorded and plotted by Matlab command semilogy [30]. We always choose ILU(0) [31] as the preconditioner. However, other suitable preconditioners can also be adapted.

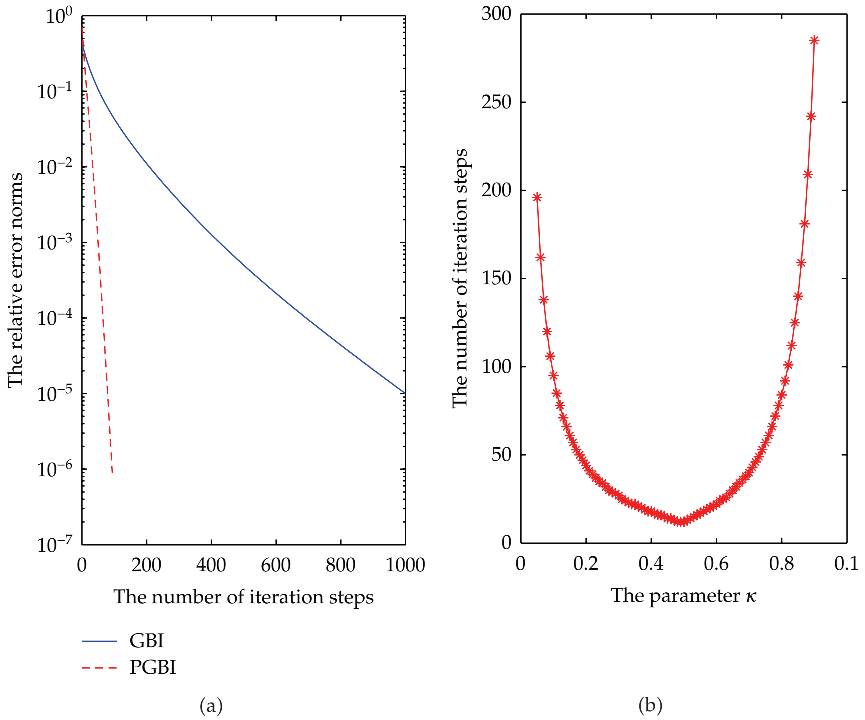


Figure 1: Comparison of convergence curves using GBI and PGBI (a); the influence of parameter κ on PGBI method (b).

Example 4.1. The coefficient matrices used in this example are taken from [18]. We outline the matrix for completeness:

$$\begin{gathered} m = 50, \quad n = 30, \quad \text{rand}(\text{"state"}, 0), \\ A = \text{triu}(\text{rand}(m, m), 1) + \text{diag}(\alpha + \text{diag}(\text{rand}(m))), \\ B = \text{triu}(\text{rand}(n, n), 1) + \text{diag}(\alpha + \text{diag}(\text{rand}(n))). \end{gathered} \tag{4.2}$$

In the matrices, there is a parameter α which can be used to change the weight of the diagonal entries. The case with $\alpha = 3$. For $\kappa = 0.1$, the behavior of the error norms is recorded in Figure 1(a). From this figure, we can see that the convergence of GBI method tends to slow down after some iteration steps. The PGBI method converges linearly, and much faster than GBI method. In Figure 1(b), the convergence curves with different parameter κ are recorded. For this problem, we can see that the optimal value of κ is very close to 0.5. By comparing the convergence of GBI method, we can see that PGBI is able to converge within 300 iteration steps, whereas GBI needs more than 1000 iteration steps. Therefore, even not with the optimal κ, the convergence of PGBI method is also much faster than that of GBI method.

Example 4.2. In this example, we test a problem with $B = A$, where coefficient matrix A is generated from the discretization of the following two-dimensional Poisson problem:

$$-\Delta u = f \tag{4.3}$$

posed on the unit square $\Omega = 0 \leq x, y \leq 1$ with Dirichlet boundary conditions

$$u(x, y) = 0. \tag{4.4}$$

Discretizing this problem by the standard second-order finite difference (FE) scheme on a uniform grid with step size $h = 1/(m+1)$ in each direction, we obtain a coefficient matrix

$$A = I_m \otimes D + \kappa_2 S \otimes I_m \tag{4.5}$$

with

$$\begin{aligned} D &= \operatorname{tridiag}(-\kappa_1, d, -\kappa_1), \\ S &= \operatorname{tridiag}(-1, 0, -1), \end{aligned} \tag{4.6}$$

and $d = 2(\kappa_1 + \kappa_2)$, $\kappa_1 = \kappa_2 = 1$. By choosing $m = 30$, we compared the performance of GBI method and the preconditioned GBI method. The convergence curves are recorded in Figure 2(a). Figure 2(b), the influence of parameter κ is investigated. From this figure we can see that the optimal κ is close to 0.4. By comparing the convergence of GBI method, it is easy to see that for a wide range of κ the preconditioned GBI method is much better than GBI method.

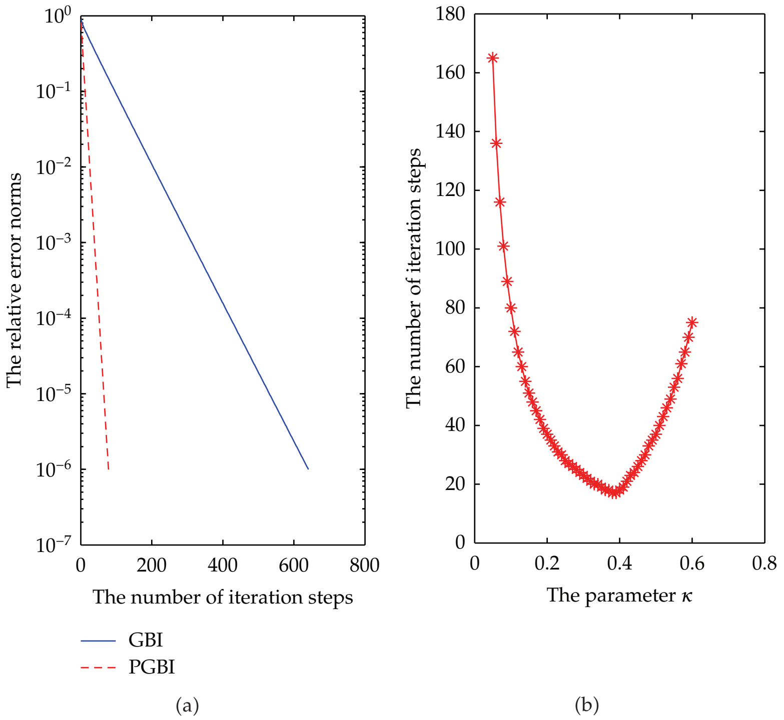
Example 4.3. In this example, we consider the convection diffusion equation with Dirichlet boundary conditions [32]

$$-\Delta u + 2\nu u_x + 2\nu u_y = f \quad \text{in } [0,1]^2, \qquad u = g \quad \text{on } \partial[0,1]^2. \tag{4.7}$$

The equation is discretized by using central finite differences on $[0,1]^2$, with mesh size $h = 1/(m+1)$ in the X-direction, and $p = 1/(n+1)$ in the Y-direction, produces two tridiagonal matrices A and B given by

$$\begin{aligned} A &= \frac{1}{h^2} \operatorname{tridiag}\{1 + \nu h, 2, 1 - \nu h\}, \\ B &= \frac{1}{p^2} \operatorname{tridiag}\{1 + \nu p, 2, 1 - \nu p\}. \end{aligned} \tag{4.8}$$

By taking $\nu = 3$, $m = 60$, and $n = 40$, we have tested the performance of GBI method and the preconditioned GBI method. The convergence curves are recorded in Figure 3(a). From this figure, we can see that the GBI method converges very slowly and nearly stagnate. The preconditioned GBI method has much better performance. We also investigate the influence



(a)

(b)

Figure 2: Comparison of convergence curves using GBI and PGBI (a); the influence of parameter κ on PGBI method (b).

of parameter κ on this problem. The convergence behavior with different κ is recorded in Figure 3(b). From this figure, we can see that the parameter becomes very sensitive when it is larger than the optimal value. Therefore, a relative small parameter is more reliable.

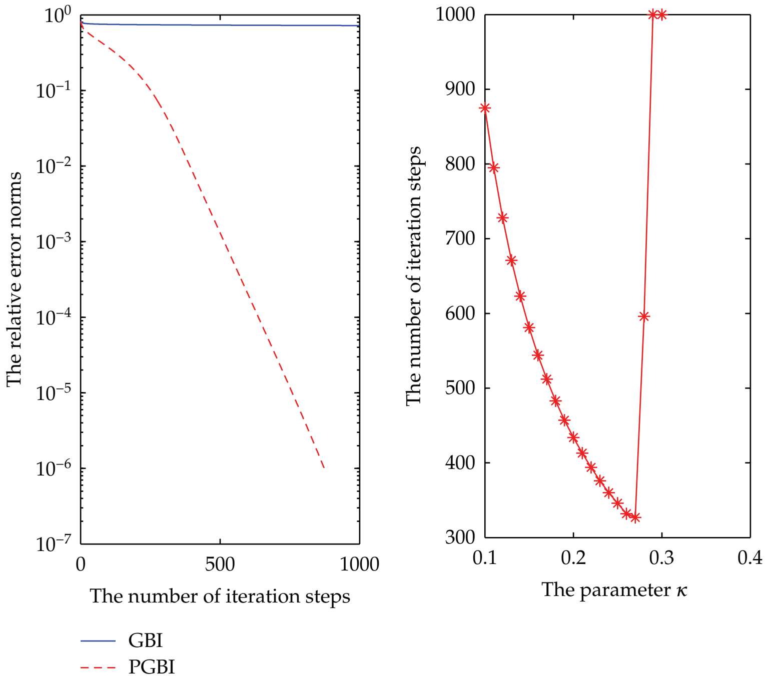
Example 4.4. In this example, we intend to test the algorithm for solving generalized Sylvester equation (3.14). All the initializations are the same except setting $C = AXB+X$. The coefficient matrix A has the following structure:

$$A = \text{tridiag}(1 - d, 4, 1 + d) \tag{4.9}$$

and $B = A$. We set $d = 3$ in this example. The tested results are shown in Figure 4. The faster convergence behavior is observable from Figure 4(a).

5. Conclusions

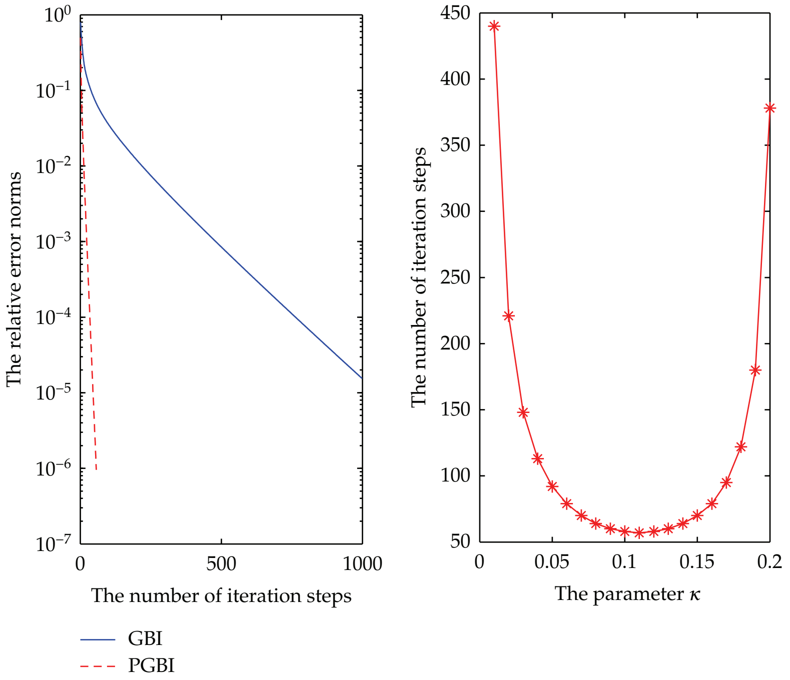
In this paper, a preconditioned gradient based iteration (PGBI) method is proposed. The convergence of PGBI is analyzed. The choice of parameter κ is an important issue, and its influence is experimentally studied. The principle idea of this paper can be extended to the more general setting like coupled Sylvester matrix equations.



(a)

(b)

Figure 3: Comparison of convergence curves using GBI and PGBI (a); the influence of parameter κ on PGBI method (b).



(a)

(b)

Figure 4: Comparison of convergence curves using GBI and PGBI (a); the influence of parameter κ on PGBI method (b).

Acknowledgment

This work is supported by NSF no. 11101204, and XJTLU RDF.

References

[1] R. Bhatia and P. Rosenthal, "How and why to solve the operator equation $AX - XB = Y$," *The Bulletin of the London Mathematical Society*, vol. 29, no. 1, pp. 1–21, 1997.

[2] B. N. Datta, *Numerical Methods for Linear Control Systems*, Elsevier Academic Press, 2003.

[3] L. Xie, Y. J. Liu, and H.-Z. Yang, "Gradient based and least squares based iterative algorithms for matrix equations $AXB + CX^TD$=F," *Applied Mathematics and Computation*, vol. 217, no. 5, pp. 2191–2199, 2010.

[4] L. Xie, J. Ding, and F. Ding, "Gradient based iterative solutions for general linear matrix equations," *Computers & Mathematics with Applications*, vol. 58, no. 7, pp. 1441–1448, 2009.

[5] J. Ding, Y. J. Liu, and F. Ding, "Iterative solutions to matrix equations of the form $A_iXB_i = F_i$," *Computers & Mathematics with Applications*, vol. 59, no. 11, pp. 3500–3507, 2010.

[6] G. H. Golub, S. Nash, and C. Van Loan, "A Hessenberg-Schur method for the problem $AX - XB = C$," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 909–913, 1979.

[7] R. H. Bartels and G. W. Stewart, "Algorithm 432: solution of the matrix equation $AX - XB = C$," *Communications of the ACM*, vol. 15, pp. 820–826, 1972.

[8] D. C. Sorensen and Y. K. Zhou, "Direct methods for matrix Sylvester and Lyapunov equations," *Journal of Applied Mathematics*, vol. 2003, no. 6, pp. 277–303, 2003.

[9] W. H. Enright, "Improving the efficiency of matrix operations in the numerical solution of stiff ordinary differential equations," *ACM Transactions on Mathematical Software*, vol. 4, no. 2, pp. 127–136, 1978.

[10] D. Calvetti and L. Reichel, "Application of ADI iterative methods to the restoration of noisy images," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 1, pp. 165–186, 1996.

[11] D. Y. Hu and L. Reichel, "Krylov-subspace methods for the Sylvester equation," *Linear Algebra and Its Applications*, vol. 172, no. 15, pp. 283–313, 1992.

[12] P. Kirrinnis, "Fast algorithms for the Sylvester equation $AX - XB^T = C$," *Theoretical Computer Science*, vol. 259, no. 1-2, pp. 623–638, 2001.

[13] G. Starke and W. Niethammer, "SOR for $AX - XB = C$," *Linear Algebra and Its Applications*, vol. 154/156, pp. 355–375, 1991.

[14] Q. Niu, X. Wang, and L.-Z. Lu, "A relaxed gradient based algorithm for solving Sylvester equations," *Asian Journal of Control*, vol. 13, no. 3, pp. 461–464, 2011.

[15] Z.-Z. Bai, "On Hermitian and Skew-Hermitian splitting iteration methods for continuous Sylvester equations," *Journal of Computational Mathematics*, vol. 29, no. 2, pp. 185–198, 2011.

[16] F. Ding and T. Chen, "On iterative solutions of general coupled matrix equations," *SIAM Journal on Control and Optimization*, vol. 44, no. 6, pp. 2269–2284, 2006.

[17] F. Ding and T. Chen, "Iterative least-squares solutions of coupled Sylvester matrix equations," *Systems & Control Letters*, vol. 54, no. 2, pp. 95–107, 2005.

[18] F. Ding and T. Chen, "Gradient based iterative algorithms for solving a class of matrix equations," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1216–1221, 2005.

[19] F. Ding, P. X. Liu, and J. Ding, "Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle," *Applied Mathematics and Computation*, vol. 197, no. 1, pp. 41–50, 2008.

[20] F. Ding and T. Chen, "Gradient-based identification methods for Hammerstein nonlinear ARMAX models," *Nonlinear Dynamics*, vol. 45, no. 1-2, pp. 31–43, 2006.

[21] F. Ding and T. Chen, "Gradient and least-squares based identification methods for OE and OEMA systems," *Digital Signal Processing*, vol. 20, no. 3, pp. 664–677, 2010.

[22] Y. J. Liu, J. Sheng, and R. F. Ding, "Convergence of stochastic gradient estimation algorithm for multivariable ARX-like systems," *Computers & Mathematics with Applications*, vol. 59, no. 8, pp. 2615–2627, 2010.

[23] Y. J. Liu, Y. S. Xiao, and X. L. Zhao, "Multi-innovation stochastic gradient algorithm for multiple-input single-output systems using the auxiliary model," *Applied Mathematics and Computation*, vol. 215, no. 4, pp. 1477–1483, 2009.

[24] F. Ding, G. J. Liu, and X. P. Liu, "Parameter estimation with scarce measurements," *Automatica*, vol. 47, no. 8, pp. 1646–1655, 2011.

[25] F. Ding and T. W. Chen, “Performance analysis of multi-innovation gradient type identification methods,” *Automatica*, vol. 43, no. 1, pp. 1–14, 2007.

[26] F. Ding, Y. J. Liu, and B. Bao, “Gradient based and least squares based iterative estimation algorithms for multi-input multi-output systems,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 226, no. 1, pp. 43–55, 2012.

[27] M. Y. Waziri, W. J. Leong, and M. Mamat, “A two-step matrix-free secant method for solving large-scale systems of nonlinear equations,” *Journal of Applied Mathematics*, vol. 2012, Article ID 348654, 9 pages, 2012.

[28] L. A. Hageman and D. M. Young, *Applied Iterative Methods*, Academic Press, New York, NY, USA, 1981.

[29] R. S. Varga, *Matrix Iterative Analysis*, Springer, Berlin, Germany, 2nd edition, 2000.

[30] The MathWorks, Inc. MATLAB 7, September 2004.

[31] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS, Boston, Mass, USA, 1996.

[32] A. Bouhamidi and K. Jbilou, “A note on the numerical approximate solutions for generalized Sylvester matrix equations with applications,” *Applied Mathematics and Computation*, vol. 206, no. 2, pp. 687–694, 2008.