International Journal of Pure and Applied Mathematics

Volume 49 No. 4 2008, 475-487

APPLYING PRECONDITIONED METHODS IN SOLUTION OF SYSTEMS RELATED TO 2-D PARABOLIC EQUATIONS

A. Shayganmanesh (Golbabai)[1 §], M.M. Arabshahi[2]

[1,2]Department of Applied Mathematics
Faculty of Mathematics
Iran University of Science and Technology
Narmak, Tehran, 16844, IRAN
[1]e-mail: golbabai@iust.ac.ir
[2]e-mail: molavi@iust.ac.ir

Abstract: The aim of this paper is applying suitable preconditioned methods for solving systems from the discretization of fourth-order formula in solution of parabolic equations, $\alpha u_{xx} + \beta u_{yy} = \Psi(x, y, t, u, u_x, u_y, u_t)$, where α and β are constants. Numerical results performed on model problem to confirm the efficiency of our approach.

AMS Subject Classification: 35Kxx
Key Words: Krylov subspace methods, preconditioner, fourth-order approximation, parabolic partial differential equations

1. Introduction

One of the most powerful tools for solving large and sparse systems of linear algebraic equations is a class of iterative methods called Krylov subspace methods. Their significant advantages like low memory requirements and good approximation properties make them very popular, and they are widely used in applications throughout science and engineering [4]. In order to be effective and obtaining faster convergence, these methods should be jointed with a suitable preconditioner. The rate of convergence generally depends on the condition number of the corresponding matrix. Preconditioners based on in-

Received: August 14, 2008

§Correspondence author

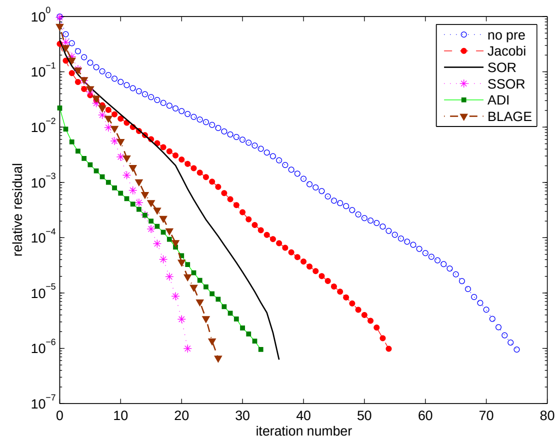

Figure 1: Convergence plot of GMRES

complete factorization are not effective for discretization of fourth order approximation [6] because they need to assume that the coefficient matrix is a M or H-matrix. The ADI method is a preconditioner for non-symmetric systems that can be very effective for the 2-D problems but this method is not effective for more general block tridiagonal systems arising from the fourth-order approximations. Comparison of preconditioning techniques for solving linear systems arising from the fourth-order approximation of the elliptic equations is presented in [5, 7]. In this article, we compare different preconditioners for solving linear systems arising from the fourth-order approximation of 2-D parabolic equation.

The outline of the paper is as follows:

In Section 2, we briefly introduce some available preconditioners. In Section 3, we consider Krylov subspace methods and in Section 4, we present an example arising from the fourth-order approximations. In Section 5, we report a brief conclusion.

2. Preconditioner Based on Relaxation Technique

The convergence rate of iterative methods depends on spectral properties of the coefficient matrix. Hence we will attempt to transform the linear system into

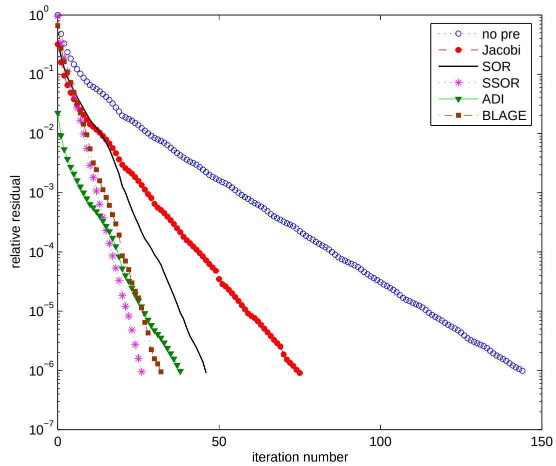

Figure 2: Convergence plot of GMRES(10)

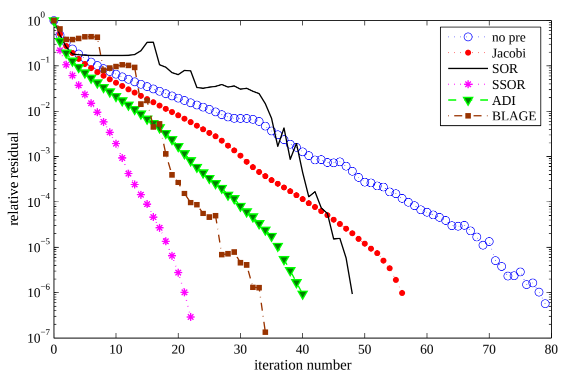

Figure 3: Convergence plot of QMR

another equivalent system in the sense that it has the same solution, but has more favorable spectral properties. A preconditioner is a matrix that effects such as a transformation [1]. If the preconditioner be as $M = M_1 M_2$ then the preconditioned system is as

$$M_1^{-1} A M_2^{-1} (M_2 x) = M_1^{-1} b. \tag{2.1}$$

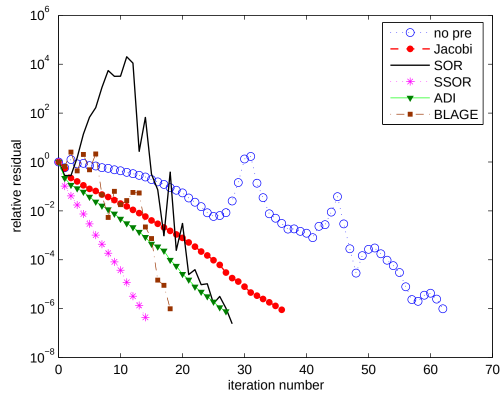

Figure 4: Convergence plot of CGS

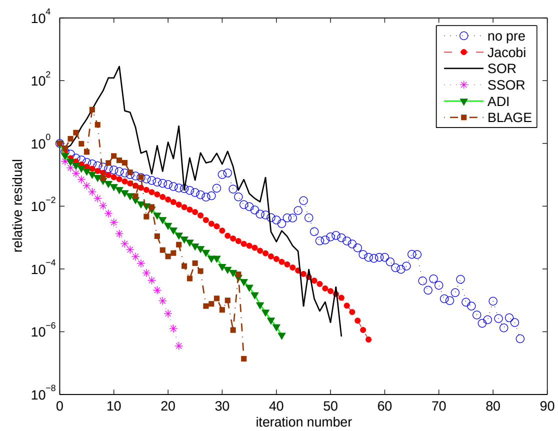

Figure 5: Convergence plot of BiCG

The matrices M_1 and M_2 are called the left and right preconditioners, respectively. Now, we describe briefly preconditioners that we use for solving linear systems.

Let $A = D + L + U$ such that D, L and U are diagonal, lower and upper triangular block matrices, respectively. A splitting of the coefficient matrix is

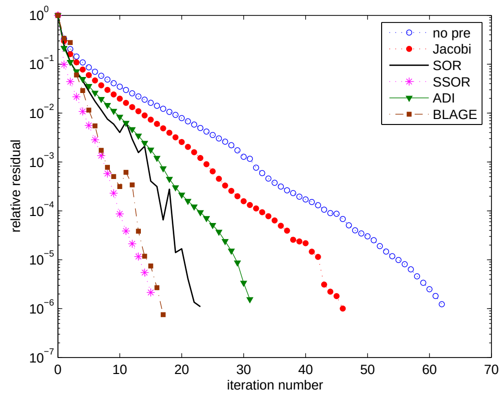

Figure 6: Convergence plot of BiCGSTAB

as $A = M - N$ where the stationary iteration for solving a linear system is as

$$x_{k+1} = M^{-1}Nx_k + M^{-1}b. \tag{2.2}$$

If the preconditioner M be as $M \equiv D$, then this preconditioner is called Jacobi. Also, if $M = \frac{1}{\omega}(D + \omega L)$ then we have SOR preconditioner where for $\omega = 1$, we have Gauss-Seidel preconditioner. If $M = \frac{1}{\omega(2-\omega)}(D + \omega L)D^{-1}(D + \omega U)$, we get $SSOR$ preconditioner. In the above notation, ω is called the relaxation parameter. The optimal value of the parameter ω, like the parameter in the SOR method will reduce the number of iterations to a lower order [2].

3. Krylov Subspace Methods

Consider the linear system:

$$Ax = b, \tag{3.1}$$

where A is a large sparse nonsymmetric matrix. Let x_0 present an arbitrary initial guess to x and $r_0 = b - Ax_0$ be a corresponding residual vector. An iterative scheme for solving (2.1) is called a Krylov subspace method if for any choice of w, it produces approximate solutions of the form $x = x_0 + w$. The subspace K_m is the Krylov subspace

$$K_m(A, r_0) = span\{r_0, Ar_0, ..., A^{m-1}r_0\}. \tag{3.2}$$

Now, we briefly describe Krylov subspace methods:

3.1. Generalized Minimal Residual (GMRES) Method

In 1986, Saad and Schultz [9] introduced GMRES method for solving nonsymmetric systems. This method has the property of minimizing the norm of the residual vector over the Krylov subspace method at every step. The major drawback for GMRES method is that the amounts of work and storage required per iteration rises linearly with the iteration number [1]. The usual way for overcome this problem is to restart after m iteration.

Algorithm 2.2. Preconditioned Generalized Minimal Residual (GMRES (m)) [8]

1. Start: Choose x_0 and a dimension m of the Krylov subspace. Define an $(m+1) \times m$ matrix $\overline{H}_m$ and initialize all the entries $h_{i,j}$ to zero.

2. Arnoldi process:

a) Compute $r_0 = M_1^{-1}(b - Ax_0)$, $\beta = \|r_0\|_2$ and $\nu_1 = r_0/\beta$.

b) For j=1,...,m do

Compute $z_j = M_1^{-1}AM_2^{-1}\nu_j$,

Compute $w = z_j$,

For i=1,...,j do

$h_{i,j} = (w, \nu_i)$,

$w = w - h_{i,j}\nu_i$,

Compute $h_{j+1,j} = \|w\|_2$ and $\nu_{j+1} = w/h_{j+1,j}$.

c) Define $V_m = [\nu_1, ..., \nu_m]$.

3. Form the approximate solution, compute $x_m = x_0 + M_2^{-1}V_m y_m$ where $y_m = argmin_y\|\beta e_1 - \overline{H}_m y\|_2$ and $e_1 = [1, 0, ..., 0]^T$.

4. Restart: If convergence has been reached then stop, else set $x_0 \leftarrow x_m$ and go to 2.

3.2. Bi-Conjugate Gradient (BiCG) Method

Bi-conjugate gradient (BiCG) method is applied to nonsymmetric matrices. BiCG method needs matrix-vector products with A and A^T. Also, BiCG method is sensitive to possible breakdowns and numerical instabilities [1].

Algorithm 2.3. Preconditioned Bi-Conjugate Gradient Method

1. $r_0 = b - Ax_0$, $\tilde{r}_0 = r_0$.

2. For $i = 1, 2, ...$

(a) solve $Mz_{i-1} = r_{i-1}$, $M^T\tilde{z}_{i-1} = \tilde{r}_{i-1}$,
(b) $\rho_{i-1} = z_{i-1}^T\tilde{r}_{i-1}$,
(c) if $\rho_{i-1} = 0$ stop
(d) if $i = 1$, $p_i = z_{i-1}$, $\tilde{p}_i = \tilde{z}_{i-1}$,
(e) else $\beta_{i-1} = \frac{\rho_{i-1}}{\rho_{i-2}}$, $p_i = z_{i-1} + \beta_{i-1}p_{i-1}$,
$\tilde{p}_i = \tilde{z}_{i-1} + \beta_{i-1}\tilde{p}_{i-1}$,
(f) $q_i = Ap_i$, $\tilde{q}_i = A^T\tilde{p}_i$,
(g) $\alpha_i = \frac{\rho_{i-1}}{\tilde{p}_i^T q_i}$,
(h) $x_i = x_{i-1} + \alpha_i p_i$,
(i) $r_i = r_{i-1} - \alpha_i q_i$,
(j) $\tilde{r}_i = \tilde{r}_{i-1} - \alpha_i\tilde{q}_i$,
(k) $\gamma_i^2 = r_i^T r_i$,
(l) if $\gamma_i^2 \leq \epsilon$ then stop.

3.3. Quasi-Minimal Residual (QMR) Method

In 1991, Freund and Nachtigal proposed the quasi-minimal residual (QMR) method for solving non-Hermitian linear systems. Later in 1994, they presented QMR method based on the coupled two-term recurrences instead of three-term recurrences [5]. This method sometimes avoids the break down of BiCG method. Also, QMR method has a regular convergence behavior than other Krylov subspace methods.

Algorithm 2.4. Preconditioned Quasi-Minimal Residual (QMR) Method

1. $r_0 = b - Ax_0$, $\rho_1 = \|M_1^{-1}r_0\|$ and $\nu_1 = \frac{r_0}{\rho_1}$.
2. $p_0 = d_0 = 0$, $c_0 = \epsilon_0 = 1$, $\vartheta_0 = 0$, $\eta_0 = -1$.
3. For $n = 1, 2, ...$

   (a) if $\epsilon_{n-1} = 0$, then stop
   (b) $\delta_n = \nu_n^T M^{-1}\nu_n$. If $\delta_n = 0$, then stop.
   (c) $p_n = M^{-1}\nu_n - p_{n-1}(\rho_n\delta_n/\epsilon_{n-1})$.
   (d) $\epsilon_n = p_n^T Ap_n$, $\beta_n = \frac{\epsilon_n}{\delta_n}$
   (e) $\tilde{\nu}_{n+1} = Ap_n - \nu_n\beta_n$,
   (f) $\rho_{n+1} = \|M_1^{-1}\tilde{\nu}_{n+1}\|$

(g) $\tilde{\nu}_n = \frac{w_{n+1}\rho_{n+1}}{w_n c_{n-1}|\beta_n|}$, $c_n = \frac{1}{\sqrt{1+\vartheta_n^2}}$, $\eta_n = -\eta_{n-1}\frac{\rho_n c_n^2}{\beta_n c_{n-1}^2}$.

(i) $d_n = p_n\eta_n + d_{n-1}(\vartheta_{n-1}c_n)^2$, $x_n = x_{n-1} + d_n$.

(j) if $\rho_{n+1} = 0$, then stop otherwise $\nu_{n+1} = \tilde{\nu}_{n+1}/\rho_{n+1}$.

3.4. Conjugate Gradient Squared (CGS) Method

In 1989, Sonneveld presented the conjugate gradient squared (CGS) method for nonsymmetric systems [11]. The speed of convergence of this method usually is about twice as fast as BiCG method. Convergence behavior of this method is often quite irregular, which may result loss of accuracy in the updated residual.

Algorithm 2.5. Preconditioned Conjugate Gradient Squared Method

1. $r_0 = M_1^{-1}(b - Ax_0)$, $\tilde{r}_0 = r_0$.
2. $q_0 = p_{-1} = 0$, $\rho_{-1} = 1$.
3. For $k = 0, 1, 2, ...$
   (a) $\rho_k = \tilde{r}_0^T r_k$, $\beta_k = \frac{\rho_k}{\rho_{k-1}}$,
   (b) $u_k = r_k + \beta_k q_k$,
   (c) $p_k = u_k + \beta_k(q_k + \beta_k p_{k-1})$,
   (d) $v_k = M_1^{-1}AM_2^{-1}p_k$,
   (e) $\sigma = \tilde{r}_0^T u_k$, $\alpha_k = \frac{\rho_k}{\sigma_k}$,
   (f) $q_{k+1} = u_k - \alpha_k v_k$,
   (g) $v_k = \alpha_k M_2^{-1}(u_k + q_{k+1})$,
   (h) $x_{k+1} = x_k + v_k$,
   (i) $r_{k+1} = r_k - M_1^{-1}Av_k$,
   (j) $\gamma_{k+1}^2 = r_{k+1}^T r_{k+1}$,
   (k) if $\gamma_{k+1}^2 \leq \epsilon$ then stop

3.5. Bi-Conjugate Gradient Stabilized (BiCGSTAB) Method

This method is applied for non-symmetric systems. Bi-conjugate gradient stabilized method is an alternative for CGS method that avoids the irregular convergence behavior of CGS method while maintaining about the same speed of convergence [10]. We present here BiCGSTAB method that applied to the preconditioned system (2.1) as follows:

1) $r_0 = b - Ax_0, p_0 = q_0 = 0$, $\alpha = \omega_0 = 1$.

2) For $j = 1, 2, ...$

   (a) $\beta = \frac{[(r_0, r_{j-1})\alpha]}{[(r_0, r_{j-2})\omega_{j-1}]}$,

   (b) $p_j = M^{-1}(r_{j-1} - \beta\omega_{j-1}q_{j-1}) + \beta p_{j-1}$,

   (c) $q_j = Ap_j$,

   (d) $\alpha = \frac{(r_0, r_{j-1})}{(r_0, q_j)}$,

   (e) $\tilde{s}_j = r_{j-1} - \alpha q_j$,

   (f) check norm of s if small enough: set $x_j = x_{j-1} + \alpha_j p_j$ and stop.

   (g) $s_j = M^{-1}\tilde{s}_j$,

   (h) $t_j = As_j$,

   (i) $\omega_j = \frac{(t_j, \tilde{s}_j)}{(t_j, t_j)}$,

   (j) $x_j = x_{j-1} + \alpha p_j + \omega_j s_j$,

   (k) $r_j = \tilde{s}_j - \omega_j t_j$,

   (l) $\gamma_{j+1}^2 = r_{j+1}^T r_{j+1}$,

   (m) check convergence: if $\gamma_{j+1}^2 \leq \epsilon$ then stop.

4. Numerical Experiment

In this section, we present one numerical example to show the computational efficiency of the preconditioning methods introduced in Section 2. We assume that $\alpha = \beta = 5 \times 10^{-4}$ and $\gamma = 1$. Our initial guess is the zero vector and the iterations are stopped when the error is less of 10^{-6}. We show the number of outer iterations and inner iterations GMRES(m) method with "ou" and "in" respectively in following Tables 1-6. The coefficient matrix is order of $N^2 \times N^2$. The computations have been done on a P.C. with Corw 2 Pue 2.0 Ghz and 1024 MB RAM.

We consider 2-D partial differential equation.

$$\alpha u_{xx} + \beta u_{yy} = u_x + u_y + \gamma u_t - e^{-t}\cos(x + y), \tag{4.1}$$

with Dirichlet boundary conditions, on the unit square where

$$u(x, y, t) = e^{-t}\cos(x)\sin(y). \tag{4.2}$$

N	without preconditioner	Jacobi	SOR	SSOR
19	75	54	36	21
29	111	80	54	32
39	124	89	61	36
49	119	88	63	36
59	119	87	60	35
69	129	94	59	36
79	139	102	58	37
89	151	110	58	39
99	162	119	58	40

Table 1: Number of iterations with GMRES

N	without preconditioner		Jacobi		SOR		SSOR	
	ou	in	ou	in	ou	in	ou	in
19	15	4	8	5	5	6	3	6
29	18	6	11	9	7	8	4	7
39	20	8	10	3	8	4	4	10
49	21	6	10	8	7	10	4	10
59	19	8	12	2	7	9	5	6
69	19	7	11	10	8	8	6	5
79	18	10	14	4	9	3	8	2
89	19	2	15	9	9	9	8	1
99	20	4	16	10	9	9	8	2

Table 2: Number of iterations with GMRES(10

We use the fourth-order approximation for discretization of this equation. We show the iteration numbers using different preconditioned methods in Tables 1-6 for different preconditioners. Also, their convergent plot are shown in Figures $1-6$. It is seen that we arrive in convergence with SSOR preconditioner and also, we save a lot of time. In this test condition number is high and our problem is ill-conditioned.

We saw that using of SSOR and SOR preconditioners, we arrive in convergence in less iteration numbers respectively and also, we save a lot of time using of Jacobi preconditioner.

N	without preconditioner	Jacobi	SOR	SSOR
19	79	56	48	22
29	129	85	80	29
39	152	96	99	33
49	143	100	96	37
59	162	109	95	39
69	170	126	68	41
79	189	137	67	34
89	191	141	65	38
99	225	149	63	39

Table 3: Number of iterations with QMR

N	without preconditioner	Jacobi	SOR	SSOR
19	62	36	28	14
29	91	46	40	18
39	81	50	43	18
49	88	56	44	20
59	93	56	37	20
69	103	64	37	23
79	114	74	41	26
89	129	84	44	28
99	146	NuN	42	31

Table 4: Number of iterations with CGS

N	without preconditioner	Jacobi	SOR	SSOR
19	85	57	52	22
29	131	85	78	30
39	152	99	97	36
49	143	101	76	37
59	162	108	86	39
69	182	127	68	41
79	190	137	67	34
89	188	143	65	38
99	223	131	67	39

Table 5: Number of iterations with BiCG

5. Conclusions

Dehghan and Molavi [3] have already considered the use of preconditioner for solving linear systems of elliptic equation. Here, we compared the different pre-

N	without preconditioner	Jacobi	SOR	SSOR
19	62	46	23	15
29	72	59	31	21
39	81	60	36	20
49	83	53	39	18
59	86	54	38	19
69	97	64	36	20
79	103	72	37	22
89	122	82	34	24
99	135	95	36	26

Table 6: Number of iterations with BiCGSTAB

conditioners in non-symmetric systems for parabolic equation. We saw that in well-conditioned problems the iteration number of the SSOR preconditioner is less and the iteration number of the Jacobi, SOR preconditioners is more but the computing time of SSOR preconditioner is more than other. So we propose using SSOR preconditioner in non-symmetric systems because this preconditioner have the less iteration numbers than other. We can use the parallel machines for better comparison of preconditioners.

References

[1] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. Van Der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM (1994).

[2] A.M. Bruaset, *A Survery of Preconditioned Iterative Methods*, Longman Scientific and Technical, UK (1995).

[3] M. Dehghan, S.M. Molavi-Arabshahi, Comparison of preconditioning techniques for solving linear systems arising from the fourt-order approximation of the three-dimensional elliptic equation, *Applied Mathematics and Computation*, **184** (2007), 156-172.

[4] L.C. Evans, *Partial Differential Equations*, American Mathematical Society Providence, Rhode Island (1999).

[5] R.W. Freund, N.M. Nachtigal, An implementation of the QMR method based on coupled two-term recurrences, *SIAM J. Sci. Statist. Comput.*, **15** (1994), 313-337.

[6] M.K. Jain, R.K. Jain, R.K. Mohanty, Fourth-order finite difference method for 2D parabolic partial differential equations with nonlinear first-derivative terms, *Numerical Methods for Partial Differential Equations*, **8** (1992), 21-31.

[7] S.M. Molavi-Arabshahi, M. Dehghan, Preconditioned techniques for solving large sparse linear systems arising from the discretization of the elliptic partial differential equations, *Applied Mathematics and Computation*, Article In Press (2007).

[8] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Second Edition, PWS Publishing Company, Boston (2000).

[9] Y. Saad, M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM, J. Sci. Statist. Comput.*, **7** (1985), 856-869.

[10] G.L.G. Sleijpen, R.F. Diederik, BICGSTAB(l) for linear equations involving un-symmetric matrices with complex spectrum, *Electronic Transactions on Numerical Analysis*, **1** (1993), 11-32.

[11] P. Sonneveld, CGS, a fast Lanczos-type solver for non symmetric linear systems, *SIAM. J. Sci. Statist. Comput.*, **10** (1989), 36-52.

[12] H.A. Van Der Vorst, *Iterative Krylov Subspace Methods for Large Linear Systems*, Cambridge University Press, Cambridge (2003).