ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 9, No. 1, 2014, *pp*. 054-066

Matrix technology for a class of fourth-order difference schemes in solution of hyperbolic equations

A. Golbabai
Islamic Azad university, Karaj Branch, Karaj, Iran
E-mail addresses: golbabai@iust.ac.ir
(*Received April 20, 2013, accepted October 19, 2013*)

Abstract. In this article, We apply Krylov subspace methods in combination of the ADI, BLAGE,... method as a preconditioner for a class of linear systems arising from fourth-order finite difference schemes in solution of hyperbolic equations $\alpha u_{tt} - \beta(x,t)u_{xx} = F(x,t,u,u_x,u_t)$ subject to appropriate initial and Dirichlet boundary conditions, where α is constant. We show The BLAGE preconditioner is extremely effective in achieving optimal convergence rates for the class of fourth-order difference schemes considered in this paper. Numerical results performed on model problem to confirm the efficiency of our approach.

Keywords: Fourth-order approximation; Hyperbolic equations; Krylov subspace methods; Preconditioner.

1. Introduction

In Solution of PDE's by means of numerical methods one often has to deal with large systems of linear equations, especially if the PDE's is time-independent or if the time-integrator is implicit. For real life problems, these large systems can often only be solved by means of some iterative method. Even if the system are preconditioned, the basic iterative method often converges slowly or even diverges. The numerical solution of one space second order hyperbolic equations with nonlinear first derivative terms in Cartesian, cylindrical and spherical coordinates are of great importance in many fields of engineering and sciences.

Many computational models give rise to large sparse linear systems. For such systems iterative methods are usually preferred to direct methods which are expensive both in memory and computing requirements. When the iterative method is based on Krylov subspaces, there is a need to use preconditioning techniques in order to achieve convergence in a reasonable number of iteration steps. Since the preconditioner plays a critical role in preconditioned Krylov subspace methods, many preconditioners have been proposed and studied [22, 5, 11]. These, preconditioners based on incomplete factorization such as ILU preconditioner that have been proposed and studied by many of researchers [17, 10, 11]. The ADI method is a preconditioner for non-symmetric systems that can be very effective but this method is not effective for more general block tri-diagonal systems arising from the fourth-order approximations. Bhuruth and Evans [3] proposed BLAGE method as a preconditioner for a class of non-symmetric linear systems arising from the fourth-order finite difference schemes. In this article, we compare different preconditioned methods for solving linear systems arising from the fourth-order approximation of hyperbolic equation

$$\alpha u_{tt} - \beta(x,t)u_{xx} = F(x,t,u,u_x,u_t)$$

defined in the region $W\times[0<t<T]$, where $W=\{x\,|\,0<x<1\}$ and α is constant. The initial and boundary conditions consists of

$$u(x,0)=g_1(x), \qquad u_t(x,0)=g_2(x), \qquad 0\le x \le 1,$$

$$u(0,t)=h_0(t), \qquad u(1,t)=h_1(t), \qquad t\ge 0,$$

Where $u=u(x,t)$. The resulting block tri-diagonal linear system of equations is solved by using Krylov subspace methods. The outline of this paper is as follows: In Section 2, we describe Krylov subspace methods. In Section 3, we briefly introduce some available preconditioners. In Section 4 we present a class of fourth-order finite difference operators and in Section 5, we present an example to illustrate the accuracy of our method. In Section 6, we report a brief conclusion.

DOI https://doi.org/2024-JICS-22597 | Generated on 2025-04-20 18:19:54
OPEN ACCESS

2. Krylov subspace methods

Let x_0 be an arbitrary initial guess for linear systems given by $Ax = b$ and let $r_0 = b - Ax_0$ be the corresponding residual vector. A Krylov subspace of order m that is shown with $K_m(A,r)$ is defined as follows:

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

For un-symmetric matrix A, different Krylov methods can be used such as GMRES, GMRES(m), QMR, CGS, BiCG, BiCGSTAB [18, 24]. Now, we briefly describe some Krylov subspace methods:

2.1. Generalized Minimal Residual(GMRES) method

In 1986, Saad and Schultz [19] introduced GMRES method for solving non-symmetric 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 linearly rises with the iteration number. The usual way for overcome this problem is to restart after m iteration.

Proposition 2.1. Assume that A is a diagonalizable matrix and let $A = XDX^{-1}$ where $D = \text{diag}\{\lambda_1, ..., \lambda_n\}$ is the diagonal matrix of eigenvalues. Define,

$\varepsilon^m = \min\_\{p \varepsilon P_m, p(0) = 1\} \max\_\{i = 1, ..., n\} | p(\lambda_i) |$.

Then, the residual norm achieved by the m-th step of GMRES satisfies the inequality

$\| r_m \|_2 \leq K(X) \epsilon^m \| r_0 \|_2$,

where , $K(X) = \| X \|_2 \| X^{-1} \|_2$.

If A is positive real with symmetric part M , the following error bound can be derived from the proposition,

$\| r_m \| \leq [1 - \alpha / \beta]^{m/2} \| r_0 \|$,

with $\alpha = (\lambda_{\min}(M))^2$, $\beta = \lambda_{\max}(A^T A)$. This proves the convergence of the GMRES(m) for all m when A is positive real [18].

2.2. Bi-Conjugate Gradient (BiCG) method

DOI https://doi.org/2024-JICS-22597 | Generated on 2025-04-20 18:19:54

OPEN ACCESS

Bi-conjugate gradient (BiCG) method was suggested by Fletcher in 1977, is applied to non-symmetric matrices. BiCG method needs matrix-vector products with A and A^T. Also, BiCG method is sensitive to possible breakdowns and numerical instabilities.

Proposition2.2 [18] The vectors produced by the Bi-conjugate Gradient algorithm satisfy the following orthogonality properties:

$$(r_j, r_i^*) = 0, i \neq j$$

$$(Ap_j, p_i^*) = 0, i \neq j$$

2.3. 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 [9]. This method sometimes avoids the break down of BiCG method. Also, QMR method has a regular convergence behavior than other Krylov subspace methods.

Proposition 2.3 [18]The residual norm of the approximate solution x_m of QMR method satisfies the relation

$$\| b - Ax_m \| \leq \| V_{m+1} \|_2 | s_1 \ldots s_m | \| r_0 \|_2 .$$

2.4. Conjugate Gradient Squared (CGS) method

In 1989, Sonneveld presented the conjugate gradient squared (CGS) method for nonsymmetric systems [21]. 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 of Preconditioned Conjugate Gradient Squared method is presented in [21].

2.5. 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 [20]. Algorithm of BiCGSTAB method that applied to the preconditioned system (2.1) is presented in [2].

3. Preconditioner

The convergence rate of iterative methods depends on spectral properties of the coefficient matrix. Hence we will attempt to transform the linear system into another equivalent system in the sense that it has the same

DOI https://doi.org/2024-JICS-22597 | Generated on 2025-04-20 18:19:54
OPEN ACCESS

solution, but has more favorable spectral properties. A preconditioner is a matrix that effects such as a transformation [2, 4]. If the preconditioner be as $M = M_1 M_2$ then the preconditioned system is as

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

The matrices M_1 and M_2 are called the left and right preconditioners, respectively. Now, we briefly describe preconditioners that we use for solving linear systems and let us take A matrix arising from fourth order approximations that is block tri-diagonal.

3.1. Preconditioner based on relaxation technique

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

$$x^{(k+1)} = M^{-1} N x^{(k)} + M^{-1} b .$$

In Table 1, we briefly show preconditioners based on relaxation technique.

Table 1 Preconditioners based on relaxation technique Preconditioner

Preconditioner	M
Jacobi	D
Gauss-Seidel	(D+L)
SOR	$\frac{1}{\omega}(D+\omega L)$
SSOR	$\frac{1}{\omega(2-\omega)}(D+\omega L)D^{-1}(D+\omega U)$

In the above notation, ω is called the relaxation parameter. The optimal value of the parameter ω reduces the number of iterations to a lower order [1]. We have chose M in Jacobi, G-S, SOR as a left preconditioner and in SSOR preconditioner, we have chose $M_1 = \frac{1}{\omega(2-\omega)}(D+\omega L)$ as a left preconditioner and $M_2 = D^{-1}(D+\omega U)$ as a right preconditioner. Also, we take $\omega_{opt} = \frac{2}{1+\sqrt{1-\rho_J^2}}$

3.2. ADI preconditioner

Peaceman and Rachford [16] in 1955 presented the ADI method for solving linear systems. Let A=H+V and in the form

DOI https://doi.org/2024-JICS-22597 | Generated on 2025-04-20 18:19:54
OPEN ACCESS

$$A=\begin{pmatrix} B_1 & C_1 & & & \\ A_2 & B_2 & C_2 & & \\ & \ddots & \ddots & \ddots & \\ & & A_{n-1} & B_{n-1} & C_{n-1} \\ & & & A_n & B_n \end{pmatrix}$$

where $A_i = tridiag\{a_{1i}, b_{1i}, c_{1i}\}$, $B_i = tridiag\{a_{2i}, b_{2i}, c_{2i}\}$ and $C_i = tridiag\{a_{3i}, b_{3i}, c_{3i}\}$ of order $N \times N$ where H and V are bounded and include $H = \{0.5B_i, b_{3i}, b_{1i}\}$, $V = \{0.5B_i, a_{1i}, c_{1i}, a_{3i}, c_{3i}\}$. The alternative direction implicit method for solving the linear system Ax=b is in following form:

$$(\mathrm{H}+\mathrm{r}_1\mathrm{I})\mathrm{u}^{(k+1/2)} = \mathrm{b} - (\mathrm{V} - \mathrm{r}_1\mathrm{I})\mathrm{u}^{(k)},$$

$$(\mathrm{V}+\mathrm{r}_2\mathrm{I})\mathrm{u}^{(k+1)} = \mathrm{b} - (\mathrm{H} - \mathrm{r}_2\mathrm{I})\mathrm{u}^{(k+1/2)},$$

The ADI preconditioner is as $\mathrm{M} = (\mathrm{H}+\mathrm{r}_1\mathrm{I})(\mathrm{V}+\mathrm{r}_2\mathrm{I})$ that $\mathrm{M}_1 = (\mathrm{H}+\mathrm{r}_1\mathrm{I})$ and $\mathrm{M}_2 = (\mathrm{V}+\mathrm{r}_2\mathrm{I})$ where Parameters r_1 and r_2 are acceleration parameters. Young and Varga [25,23] proved that the optimum value for r_1 and r_2 is $\sqrt{\alpha\beta}$ where $\alpha \le \mu_i, \nu_i \le \beta$ and μ_i, ν_i are eigenvalues of matrices H and V respectively.

3.3. BLAGE preconditioner

The BLAGE method [3,7] was originally introduced as analogue of the AGE method [6]. The BLAGE uses fractional splitting technique that is applied in two half steps on linear systems with block tri-diagonal matrices of order $\mathrm{N}^2 \times \mathrm{N}^2$ and in the form

$$A=\begin{pmatrix} B_1 & C_1 & & & \\ A_2 & B_2 & C_2 & & \\ & \ddots & \ddots & \ddots & \\ & & A_{n-1} & B_{n-1} & C_{n-1} \\ & & & A_n & B_n \end{pmatrix}$$

where $\mathrm{A}_i, \mathrm{B}_i$ and C_i are tri-diagonal matrices of order $N \times N$. The splitting of matrix A is sum of matrices G_1 and G_2 in which $\mathrm{A} = \mathrm{G}_1 + \mathrm{G}_2$ where G_1 and G_2 are of the form

DOI https://doi.org/2024-JICS-22597 | Generated on 2025-04-20 18:19:54
OPEN ACCESS

$$G_1 = \begin{pmatrix} B_1' & & & & & \\ & B_2' & C_2 & & & \\ & A_3 & B_3' & & & \\ & & & \ddots & & \\ & & & & B_{n-1}' & C_{n-1} \\ & & & & A_n & B_n' \end{pmatrix} \text{ and } G_2 = \begin{pmatrix} B_1' & C_1 & & & & \\ A_2 & B_2' & & & & \\ & & \ddots & & & \\ & & & B_{n-2}' & C_{n-2} & \\ & & & A_{n-1} & B_{n-1}' & \\ & & & & & B_n' \end{pmatrix}$$

for odd values of n and

$$G_1 = \begin{pmatrix} B_1' & C_1 & & & \\ A_2 & B_2' & & & \\ & & \ddots & & \\ & & & B_{n-1}' & C_{n-1} \\ & & & A_n & B_n' \end{pmatrix} \text{ and } G_1 = \begin{pmatrix} B_1' & & & & & & \\ & B_2' & C_2 & & & & \\ & A_3 & B_3' & & & & \\ & & & \ddots & & & \\ & & & & B_{n-2}' & C_{n-2} & \\ & & & & A_{n-1} & B_{n-1}' & \\ & & & & & & B_n' \end{pmatrix}$$

for even values of n where $\mathrm{B}_\mathrm{i}' = \frac{1}{2}\mathrm{B}_\mathrm{i}$. The BLAGE preconditioner is as $\mathrm{M} = (\mathrm{G}_1 + \omega_1 \mathrm{I})(\mathrm{G}_2 + \omega_2 \mathrm{I})$ that $\mathrm{M}_1 = (\mathrm{G}_1 + \omega_1 \mathrm{I})$ and $\mathrm{M}_2 = (\mathrm{G}_2 + \omega_2 \mathrm{I})$ where ω_1 and ω_2 are optimal iteration parameters. We have experimentally chosen the relaxation parameter $\omega_1 = \sqrt{\alpha_1 \beta_2}$ and $\omega_2 = \sqrt{\alpha_2 \beta_1}$ where $\alpha_1 = \lambda_{\min}(\mathrm{M}_1)$, $\beta_1 = \lambda_{\max}(\mathrm{M}_1)$ and $\alpha_2 = \lambda_{\min}(\mathrm{M}_2)$, $\beta_2 = \lambda_{\max}(\mathrm{M}_2)$ so that we will have the minimum condition number.

4. Fourth-order approximations

Now let us $p = \frac{k}{h}$, Mohanty et al. [14] have derived finite difference schemes of fourth-order accuracy for equations of the form

$$\mathrm{u}_{\mathrm{tt}} - \mathrm{A}(\mathrm{x},\mathrm{t})\mathrm{u}_{\mathrm{xx}} = \mu(\mathrm{x},\mathrm{t})\mathrm{u}_\mathrm{x} + \nu(\mathrm{x},\mathrm{t})\mathrm{u}_\mathrm{t} + \lambda(\mathrm{x},\mathrm{t})\mathrm{u} + \mathrm{f}(\mathrm{x},\mathrm{t}),$$

the fourth-order scheme for (4.1) can be written in the form

$$\begin{aligned} &(\lambda_1 - T_{i+1}^{k+1})u_{i+1}^{k+1} + (\lambda_2 - T_{i-1}^{k+1})u_{i-1}^{k+1} + (\lambda_3 - T_i^{k+1})u_i^{k+1} + (\lambda_4 - T_{i+1}^{k})u_{i+1}^{k} + (\lambda_5 - T_{i-1}^{k})u_{i-1}^{k} \\ &+ (\lambda_6 - T_i^{k})u_i^{k+1} + (\lambda_7 - T_{i+1}^{k-1})u_{i+1}^{k-1} + (\lambda_8 - T_{i-1}^{k-1})u_{i-1}^{k-1} + (\lambda_9 - T_i^{k-1})u_i^{k-1} = \\ &\frac{k^2}{2}[(r_1 + 8ha_1 Q_i^k)K_{i+1}^k + (r_2 - 8ha_1 Q_i^k)K_{i-1}^k + (1 + 8ka_2 R_i^k)K_i^{k+1} + (1 - 8ka_2 R_i^k)K_i^{k-1} + 8K_i^k)]. \end{aligned} \tag{3.1}$$

Where

DOI https://doi.org/2024-JICS-22597 | Generated on 2025-04-20 18:19:54
OPEN ACCESS

$$T_{i+1}^{k+1} = 4kp^2 b_2 R_i^k + 4hb_1 Q_i^k + \frac{k}{4} R_{i+1}^k (r_1 + 8ha_1 Q_i^k) + \frac{kp}{4} Q_i^{k+1}(1 + 8ha_2 R_i^k), \tag{3.2}$$

$$T_{i-1}^{k+1} = 4kp^2 b_2 R_i^k - 4hb_1 Q_i^k + \frac{k}{4} R_{i-1}^k (r_2 - 8ha_1 Q_i^k) - \frac{kp}{4} Q_i^{k+1}(1 + 8ha_2 R_i^k), \tag{3.3}$$

$$T_{i+1}^{k-1} = -4kp^2 b_2 R_i^k + 4hb_1 Q_i^k - \frac{k}{4} R_{i+1}^k (r_1 + 8ha_1 Q_i^k) + \frac{kp}{4} Q_i^{k-1}(1 - 8ha_2 R_i^k), \tag{3.4}$$

$$T_{i-1}^{k-1} = -4kp^2 b_2 R_i^k - 4hb_1 Q_i^k - \frac{k}{4} R_{i-1}^k (r_1 + 8ha_1 Q_i^k) - \frac{kp}{4} Q_i^{k-1}(1 - 8ha_2 R_i^k), \tag{3.5}$$

$$\begin{aligned} T_{i+1}^{k} = {} & 2kpQ_i^k + 4k^2 c_1 Q_i^k - 8hb_1 Q_i^k + \frac{k^2}{2} S_{i+1}^k (r_1 + 8ha_1 Q_i^k) \\ & + \frac{3k}{4} pQ_{i+1}^k (r_1 + 8ha_1 Q_i^k) - \frac{kp}{4} Q_{i-1}^k (r_2 - 8ha_1 Q_i^k), \end{aligned} \tag{3.6}$$

$$\begin{aligned} T_{i-1}^{k} = {} & -2kpQ_i^k + 4k^2 c_1 Q_i^k + 8hb_1 Q_i^k + \frac{k^2}{2} S_{i-1}^k (r_2 - 8ha_1 Q_i^k) \\ & + \frac{k}{4} pQ_{i+1}^k (r_1 + 8ha_1 Q_i^k) - \frac{3kp}{4} Q_{i-1}^k (r_2 - 8ha_1 Q_i^k), \end{aligned} \tag{3.7}$$

$$\begin{aligned} T_{i}^{k+1} = {} & 2kR_i^k - 8kp^2 b_2 R_i^k + 4k^2 c_2 R_i^k + \frac{k^2}{2} S_i^{k+1} (1 + 8ka_2 R_i^k) \\ & + \frac{3k}{4} R_i^{k+1} (1 + 8ka_2 R_i^k) - \frac{k}{4} R_i^{k-1} (1 - 8ka_2 R_i^k), \end{aligned} \tag{3.8}$$

$$\begin{aligned} T_{i}^{k-1} = {} & -2kR_i^k + 8kp^2 b_2 R_i^k + 4k^2 c_2 R_i^k + \frac{k^2}{2} S_i^{k-1} (1 - 8ka_2 R_i^k) \\ & + \frac{k}{4} R_i^{k+1} (1 + 8ka_2 R_i^k) - \frac{3k}{4} R_i^{k-1} (1 - 8ka_2 R_i^k), \end{aligned} \tag{3.9}$$

$$\begin{aligned} T_i^k = {} & -8k^2 c_1 Q_i^k - 8k^2 c_2 R_i^k + 4k^2 S_i^k - kpQ_{i+1}^k (r_1 + 8ha_1 Q_i^k) + kpQ_{i-1}^k (r_2 - 8ha_1 Q_i^k) \\ & - kR_i^{k+1} (1 + 8ka_2 R_i^k) + kR_i^{k-1} (1 - 8ka_2 R_i^k), \end{aligned} \tag{3.10}$$

where

$$\lambda_1 = -(L_2 + L_3 + L_4) \quad , \quad \lambda_2 = -(L_2 - L_3 + L_4) \quad , \quad \lambda_3 = 6 + 2L_2 + 2L_4, \tag{3.11}$$

$$\lambda_4 = -(L_1 - 2L_3 - 2L_4) \quad , \quad \lambda_5 = -(L_1 + 2L_3 - 2L_4) \quad , \quad \lambda_6 = -12 + 2L_1 - 4L_4, \tag{3.12}$$

$$\lambda_7 = (L_2 - L_3 - L_4) \quad , \quad \lambda_8 = (L_2 + L_3 - L_4) \quad , \quad \lambda_9 = 6 - 2L_2 + 2L_4, \tag{3.13}$$

If we put above operators in (4.1) we arrive to a system of equations in which the corresponding matrix is tri-diagonal. We can solve this system with well-known iterative methods such as Krylov subspace methods.

5. Numerical experiment

DOI https://doi.org/2024-JICS-22597 | Generated on 2025-04-20 18:19:54
OPEN ACCESS

In this section, we present a numerical example to show the computational efficiency of the preconditioning methods introduced in Section 2. Our initial guess is the zero vector and the iterations are stopped when the norm of relative residual is less than 10^{-6}. In following Tables, We show the iteration number without using preconditioner by "no pre". The computations have been done on a P.C. with Corw 2 Pue 2.0 Ghz and 1024 MB RAM. We consider hyperbolic differential equation

$$u_{tt} = u_{xx} + u_x + u_t,$$

subject to appropriate initial and Dirichlet boundary conditions (1-2,1-3), where

$$u(x,t) = \exp(2x + 3t).$$

We discretized equation (5.1) by using forth-order approximation. The resulting coefficient matrix is block tri-diagonal and the diagonal elements are tri-diagonal matrices. We show the iteration count of different Krylov subspace methods in combination various preconditioning in Tables 2-6. When mesh size h is finer, we encounter break down by using direct preconditioners while BLAGE preconditioners work quite well. In Figures 1-5, comparison of convergence behavior are shown. Also, In Fig. 6, 7, for sample we show distribution of eigenvalues ADI and BLAGE preconditioners that M_1, M_2 are left and right preconditioners respectively. It is obvious that the distribution of eigenvalue BLAGE preconditioning is regular than ADI.

Table 2 Number of iterations with GMRES method

N	no pre	Jacobi	SOR	SSOR	ADI	BLAGE
9	74	42	27	22	13	28
19	147	131	72	53	33	63
29	251	231	121	93	43	120
39	350	335	170	144	48	183
49	452	430	221	204	132	246
59	564	530	273	318	220	294
69	665	630	488	Nun	310	356
79	770	724	Nun	Nun	405	455

DOI https://doi.org/2024-JICS-22597 | Generated on 2025-04-20 18:19:54
OPEN ACCESS

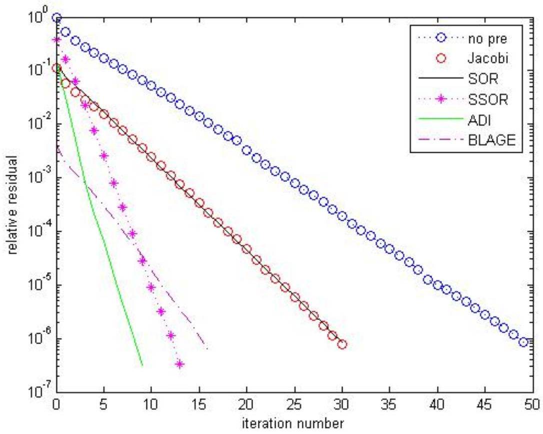

Fig.1 Comparison of convergent behavior of GMRES method

Table 3 Number of iterations with CGS method

N	no pre	Jacobi	SOR	SSOR	ADI	BLAGE
9	45	34	22	14	12	22
19	162	155	82	57	52	79
29	370	368	212	190	90	199
39	621	878	417	637	114	369
49	909	1509	1082	Nun	1329	548
59	1232	Nun	Nun	Nun	Nun	912
69	1493	Nun	Nun	Nun	Nun	1102
79	2003	Nun	Nun	Nun	Nun	1430

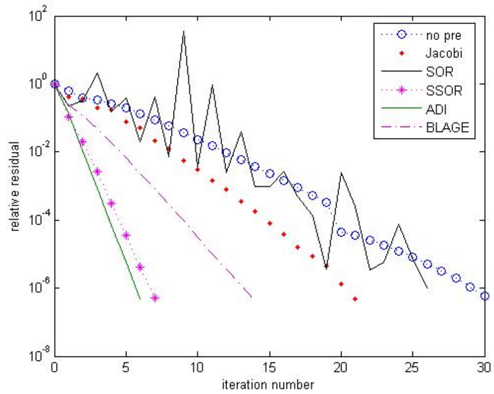

Fig.2 Comparison of convergent behavior of CGS method

Table 4 Number of iterations with QMR method

N	no pre	Jacobi	SOR	SSOR	ADI	BLAGE
9	76	54	39	23	17	39
19	154	143	102	60	45	81
29	322	322	207	150	92	179
39	553	605	407	409	93	309
49	863	1232	1228	Nun	595	519
59	1113	Nun	Nun	Nun	Nun	781
69	1456	Nun	Nun	Nun	Nun	1173
79	1930	Nun	Nun	Nun	Nun	1356

DOI https://doi.org/2024-JICS-22597 | Generated on 2025-04-20 18:19:54
OPEN ACCESS

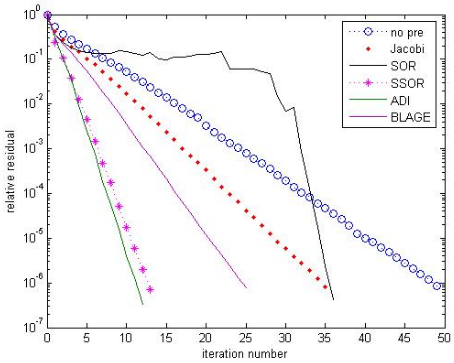

Fig.3Comparison of convergent behavior of QMR method

Table 5 Number of iterations with BiCG method

N	no pre	Jacobi	SOR	SSOR	ADI	BLAGE
9	78	55	38	23	18	39
19	154	148	103	61	44	82
29	344	323	209	151	87	180
39	553	608	486	405	91	309
49	854	1249	2932	Nun	651	515
59	1126	Nun	Nun	Nun	Nun	803
69	1495	Nun	Nun	Nun	Nun	1333
79	1922	Nun	Nun	Nun	Nun	1504

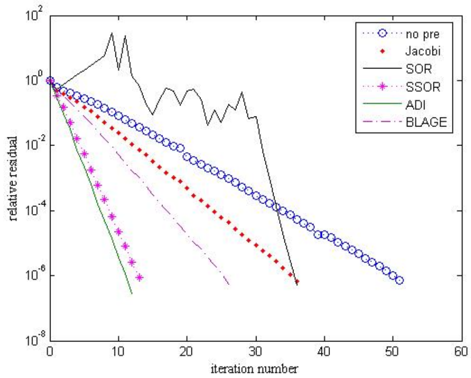

Fig.4 Comparison of convergent behavior of BiCG method

Table 5 Number of iterations with BiCGSTAB method

N	no pre	Jacobi	SOR	SSOR	ADI	BLAGE
9	58	40	22	13	10	23
19	369	291	96	67	54	83
29	676	615	247	209	94	249
39	1067	1265	584	917	111	499
49	1438	2400	1554	Nun	816	718
59	2169	Nun	Nun	Nun	Nun	1381
69	2932	Nun	Nun	Nun	Nun	1702
79	3503	Nun	Nun	Nun	Nun	2132

DOI https://doi.org/2024-JICS-22597 | Generated on 2025-04-20 18:19:54

OPEN ACCESS

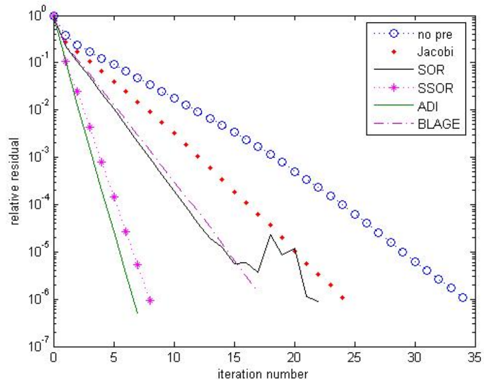

Fig.5 Comparison of convergent behavior of BiCGSTAB method

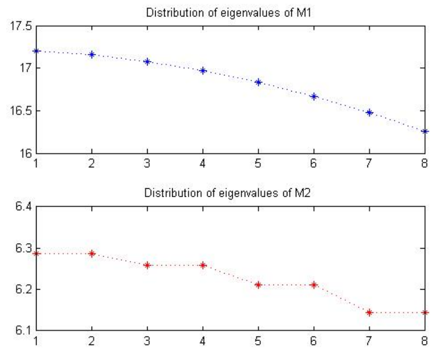

Fig.6 Distribution of eigenvalues in ADI Preconditioner

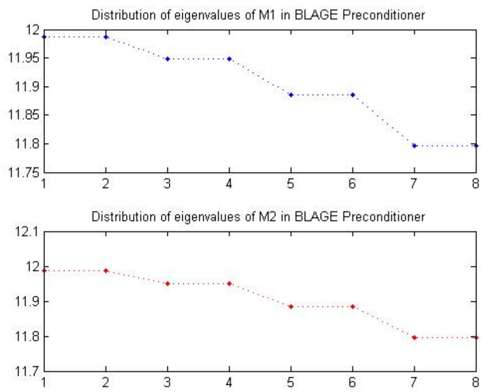

Fig.7 Distribution of eigenvalues in BLAGE Preconditioner

We see that we obtain less iteration number with using ADI and SSOR preconditioners but SSOR preconditioner needs more computing time than other preconditioners. Also, we saw that using ADI and BLAGE preconditioner we save in computing time. It is seen that when the condition number is high, the ADI and SSOR preconditioner do not work very well but in well-conditioned problems the iteration number of the BLAGE, SSOR preconditioners is less and the iteration number of the Jacobi and SOR preconditioners is more. We found when mesh size is finer, the QMR, BiCG, CGS and BiCGSTAB methods in composition preconditioners don't work very well but with using GMRES method in composition preconditioners, we get less iteration number than other preconditioned krylov subspace methods. Also, preconditioned GMRES method has regular convergence behavior.

6. Conclusions

DOI https://doi.org/2024-JICS-22597 | Generated on 2025-04-20 18:19:54
OPEN ACCESS

Here, we compared the different preconditioners in non-symmetric systems for hyperbolic equation. From Tables and Figures, we see that although all the methods seem to work well with BLAGE preconditioning using GMRES gives the fastest convergence. Also, the computing time of BLAGE preconditioner is less than other preconditioners. So we propose using BLAGE preconditioner because this preconditioner needs to less computing time and have the less iteration numbers than other. We propose using the parallel machines for better comparison of block preconditioners because the BLAGE and ADI preconditioners can be employed in parallel environment where the preconditioning operations can be divided into several subproblems which can be run in parallel [3].

7. References

[1] O. Axelsson, Iterative Solution Methods, Cambridge University Press, New York, 1996.

[2] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine. and H. van der Vorst, Templates for the solution of linear systems: Building blocks for iterative methods, SIAM, 1994.

[3] M. Bhuruth and D. J. Evans, Block alternating group explicit preconditioning (BLAGE) for a class of fourth order difference schemes, Intern. J. Comput. Math., 63, 1997, 121- 136.

[4] M. Bruaset, A Survery of preconditioned iterative methods, Longman Scientific and Technical, UK, 1995.

[5] M. Dehghan and 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, 156-172, 2007.

[6] D. J. Evans and M. Sahimi, The alternating group explicit (AGE) iterative method to solve parabolic and hyperbolic partial differential equations in: Annual Review of Numerical Fluid Mechanics and Heat Transfer 11, C.L. Tien and T.C. Chawla, eds., Hemisphere Pub. Co., USA, 1989, 283-390.

[7] D. J. Evans and W. S. Yousif, The block alternating group explicit method (BLAGE) for the solution of elliptic difference equations, Intern. J. Comput. Math., 22,1987, 177-185.

[8] L. C. Evans, Partial Differential Equations, American Mathematical Society Providence, Rhode Island, 1999.

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

[10] Davod Khojasteh Salkuyeh, On the preconditioning of the block tridiagonal linear system of equations, J. Appl. Math. Comput., 28, 2008, 133–146.

[11] M.H. Koulaei and F. Toutounian, On computing of block ILU preconditioner for block tridiagonal systems, Journal of computational and applied mathematics, 202, 2007, 248-257.

[12] S. Ma. and Y. Saad, Block-ADI preconditioners for solving sparse nonsymmetric linear systems of equations, in: Numerical Linear Algebra, L. Reichel, A. Ruttan and R. S. Varga, editors, Walter de Gruyter and Co., 1993, 165-178.

[13] R.C. Mittal and A.H. Al-Kurdi, An Efficient Method for Constructing an ILU Preconditioner for Solving Large Sparse Nonsymmetric Linear Systems by the GMRES Method, Computers and Mathematics with Applications, 45, 2003, 1757-1772.

[14] R.K. Mohanty and M.K. Jain and Kochurani George , On the use of high order difference methods for the system of one space second order nonlinear hyperbolic equations with variable coefficients, Journal of Computational and Applied Mathematics, 72, 1996, 421-431.

[15] S.M. Molavi-Arabshahi and M. Dehghan, Preconditioned techniques for solving large sparse linear systems arising from the discretization of the elliptic partial differential equations, Applied Mathematics and Computation, 188, 2007, 1371-1388.

[16] D. W. Peaceman and H. H. Rachford, The numerical solution of parabolic and elliptic differential equations, J. Soc. Indust. Appl. Math, 3,1955, 28-41.

[17] G. Pini and G. Zilli, Preconditioned iterative algorithms for large sparse unsymmetric problems}, Numerical methods for partial differential equations, 5,1989, 107-120.

[18] Y. Saad, Iterative Methods for Sparse Linear Systems, Second Edition, PWS Publishing Company, Boston, 2000.

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

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

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

DOI https://doi.org/2024-JICS-22597 | Generated on 2025-04-20 18:19:54

OPEN ACCESS

[22] S. Sundar and B,K. Bhagavan, CGS, Comparison of krylov subspace methods with preconditioning techniques for solving boundary value problems, Computers and mathematics with applications, 38, 1999, 197-206.

[23] R. S. Varga, Matrix Iterative Analysis, Prentice Hall, Englewood Cliffs, New Jersey, 1962.

[24] H.A. Van der Vorst, Iterative Krylov Subspace Methods for Large Linear Systems, Cambridge University Press, Cambridge, 2003.

[25] D. M. Young, Iterative Solution of Large Linear Systems. Academic Press, New York, 1971.

DOI https://doi.org/2024-JICS-22597 | Generated on 2025-04-20 18:19:54

OPEN ACCESS