An International Journal
computers & mathematics
with applications

PERGAMON Computers and Mathematics with Applications 38 (1999) 197–206
www.elsevier.nl/locate/camwa

Comparison of Krylov Subspace Methods with Preconditioning Techniques for Solving Boundary Value Problems

S. SUNDAR AND B. K. BHAGAVAN
Department of Mathematics, Indian Institute of Technology
Kharagpur-721 302, India
slnt@maths.iitkgp.ernet.in

(Received December1998; accepted January 1999)

Abstract—In this paper, we made an attempt to establish the usefulness of Lanczos solver with preconditioning technique over the preconditioned Conjugate Gradient (CG) solvers. We have presented here a detail comparative study with respect to convergence, speed as well as CPU-time, by considering appropriate boundary value problems. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords—Krylov subspace, Preconditioned conjugate gradient method, Lanczos algorithm, Preconditioned Lanczos algorithm, Symmetric systems.

1. INTRODUCTION

Systems of linear equations arise from the numerical modeling of problems in many branches of science and engineering. For example, the discretization of boundary value problems by finite difference or finite element methods gives rise to huge sparse systems of linear equations. The solution of a linear system of equations with a symmetric coefficient matrix is a most common and usually very important task in scientific computation. The iterative methods for solving linear systems possess certain desirable advantages over direct methods.

One of the most successful and widely accepted iterative solver is the Krylov subspace methods, such as Conjugate Gradients (CG) method [1,2], and Lanczos algorithm [3–5]. All these Krylov subspace methods can obtain exact solutions at most in N steps, when round of error become zero [1] and produce the same approximate solution at each step in exact arithmetic [6], although the approaches they follow are computationally a little different. The strong effect of roundoff error on actual implementation does not prevent convergence but merely delays it [4]. It is well known that a satisfactory accuracy is often achieved for values of m far less than N (order of matrix A) [5].

A common feature of all these methods is that the approximate solution X^m belongs to the affine space $X^0 + K^m$, where K^m is the Krylov subspace of dimension m, $K^m = (\mathbf{r}^0, A\mathbf{r}^0, \dots, A^{m-1}\mathbf{r}^0)$, and $\mathbf{r}^0 = \mathbf{b} - AX^0$ is the initial residual vector. The principal idea here is to make

One of the authors, Bhagavan, would like to thank CSIR, New Delhi for financial support under the Grant No. 9/81/(265)/95-EMR-I.

PII: S0898-1221(99)00298-9

the residual vector $\mathbf{r}^m$ orthogonal to another Krylov subspace L (called left subspace), usually different from K^m [2,7,8].

All the Krylov subspace methods slow down the convergence speed, whenever the system becomes ill conditioned. So the Krylov subspace methods, together with suitable powerful preconditioning techniques, accelerate the speed of convergence, which is a vital part in such high-performance computing. Several authors discussed various preconditioning techniques, viz., the incomplete Cholesky Factorization (ILU) [9,10] polynomial preconditioning [11,12], and so on.

In this paper, we made an attempt to establish the usefulness of Lanczos solver with preconditioning technique over the preconditioned conjugate gradient (CG) solvers. We have presented here a detail comparative study with respect to convergence speed as well as CPU-time, by considering appropriate boundary value problems.

The outline of the paper is as follows. The next section deals with the Preconditioned Conjugate Gradient (PCG) solver for solving symmetric linear systems. The ensuing section describes the Lanczos symmetric solver with preconditioning Lanczos solver. We present the results of our comparative study in the final section.

2. PRECONDITIONED CONJUGATE GRADIENT (PCG) SOLVER FOR SYMMETRIC LINEAR SYSTEM

The method of conjugate gradient originally developed by Hestenes and Stiefel [4] is used for solving the linear system of equations

$$\mathbf{A}x = \mathbf{b}, \tag{1}$$

particularly matrix $\mathbf{A} \in R^{n \times n}$ is a sparse Symmetric Positive Definite (SPD) coefficient matrix.

THEORETICAL BACKGROUND. Let $\mathbf{x}^0$ as initial approximate solution,

$$\begin{aligned} \mathbf{r}^0 &= \mathbf{b} - \mathbf{A}x^0 \\ &\text{for the } m^{\text{th}} \text{ iteration } X^m \in X^0 + K^m \\ K^m &= \text{ Span } \left(\mathbf{r}^0, \mathbf{r}^1, \dots, \mathbf{r}^{m-1}\right) \\ &= \text{Span } \left(\mathbf{r}^0, A\mathbf{r}^0, \dots, A^{m-1}\mathbf{r}^0\right). \end{aligned}$$

PRECONDITIONING TECHNIQUES. The convergence of the iterative methods highly depends on the eigenvalue distribution of the matrix $\mathbf{A}$ [13]. A criterion for the width of the spectrum is the Euclidean condition number, that is, for spd matrices is $\kappa = \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2 = \lambda_{\max}(\mathbf{A})/\lambda_{\min}(\mathbf{A})$. With $\gamma = (\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$, the distance to the exact solution x^* in the i^{th} iteration is bounded by $\|x^i - x^*\|_2 \leq 2\sqrt{\kappa}\gamma^i \|x^0 - x^*\|_2$. The right-hand side increases with growing condition number. Hence, lower condition number usually accelerate the speed of convergence. The most efficient preconditioning is direct matrix inversion, but this is not a practical choice. An alternative procedure such as incomplete Cholesky Factorization [9,10] or polynomial preconditioning [9,11,12] are used as an approximation for matrix inversion. Here we have to apply the regular CG method to the transformed system,

$$\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}, \tag{2}$$

where $\tilde{\mathbf{A}} = \mathbf{SAS}^{\mathsf{T}}$, $\tilde{\mathbf{x}} = \mathbf{S}^{-\mathsf{T}}\mathbf{x}$, and $\tilde{\mathbf{b}} = \mathbf{Sb}$. Here $\mathbf{S}$ is a nonsingular matrix chosen and the preconditioner $\mathbf{M} = (\mathbf{S}^{\mathsf{T}}\mathbf{S})^{-1}$. The good choice of $\mathbf{M}$ implies that $\tilde{\mathbf{A}}$ of the transformed system has a smaller condition number than matrix $\mathbf{A}$. $\tilde{\mathbf{A}}$ satisfies $\mathbf{S}^{\mathsf{T}}\tilde{\mathbf{A}}\mathbf{S}^{-\mathsf{T}} = \mathbf{S}^{\mathsf{T}}\mathbf{SA} = \mathbf{M}^{-1}\mathbf{A}$, therefore, the matrix $\mathbf{M}^{-1}\mathbf{A}$ is similar to $\tilde{\mathbf{A}}$ so that condition number of $\tilde{\mathbf{A}}$ is the ratio of the maximum and minimum eigenvalues of $\mathbf{M}^{-1}\mathbf{A}$.

ALGORITHM 2.1. Preconditioned Conjugate Gradient Solver (PCG)

STEP 1. Choose an arbitrary $\mathbf{x}^0$,

$$\text{set } \mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0, \tag{3}$$

$$\text{solve } M\tilde{\mathbf{r}}^0 = \mathbf{r}^0, \tag{4}$$

$$\mathbf{p}^0 = \tilde{\mathbf{r}}^0. \tag{5}$$

STEP 2. Iterate for $i = 0, 1, 2, \dots$ until convergence

$$\alpha^i = \frac{(\tilde{\mathbf{r}}^i, \mathbf{r}^i)}{(\mathbf{p}^i, \mathbf{A}\mathbf{p}^i)}, \tag{6}$$

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \alpha^i \mathbf{p}^i, \tag{7}$$

$$\mathbf{r}^{i+1} = \mathbf{r}^i - \alpha^i \mathbf{A} p^i, \tag{8}$$

$$\text{solve } \mathbf{M}\tilde{\mathbf{r}}^{i+1} = \mathbf{r}^{i+1}. \tag{9}$$

TERMINATION. If $(\mathbf{r}^{i+1}, \mathbf{r}^{i+1}) < \epsilon$, stop

$$\beta^i = \frac{(\tilde{\mathbf{r}}^{i+1}, \mathbf{r}^{i+1})}{(\tilde{\mathbf{r}}^i, \mathbf{r}^i)}, \tag{10}$$

$$\mathbf{p}^{i+1} = \tilde{\mathbf{r}}^{i+1} + \beta^i \mathbf{p}^i. \tag{11}$$

3. LANCZOS SOLVER FOR SYMMETRIC LINEAR SYSTEMS

The Lanczos algorithm has proved to be a powerful solution method for solving linear systems of equations [8]. The Lanczos algorithm in exact arithmetic is described as the process of constructing a set of orthonormal vectors $(\mathbf{q}^1, \mathbf{q}^2, \dots, \mathbf{q}^i)$ by applying Gram-Schmidt orthogonalization to the set of vectors $(\mathbf{r}^0, A\mathbf{r}^0, \dots, A^{i-1}\mathbf{r}^0)$ which form the Krylov subspace. Because of special properties of the orthonormal vectors [8], it is possible to orthogonalize $A\mathbf{q}^i$ against $\mathbf{q}^{i-1}$ and q^i to obtain the next orthogonal vector

$$\mathbf{r}^{i+1} = \beta^{i+1}\mathbf{q}^{1+1} = A\mathbf{q}^i - \alpha^i \mathbf{q}^i - \beta^i \mathbf{q}^{i-1}, \tag{12}$$

without using the vectors $(\mathbf{q}^1, \mathbf{q}^2, \dots, \mathbf{q}^i)$ in 12, where $\alpha^i = (\mathbf{q}^i)^{\mathsf{T}} A\mathbf{q}^i$ and $\beta^i = \|\mathbf{r}^i\|$. This special choice of the vectors for the subspace has the property that the projection of a matrix A onto this subspace is a tridiagonal matrix T^i,

$$T^i = \left(Q^i\right)^{\mathsf{T}} AQ^i, \tag{13}$$

$$T^i = \begin{bmatrix} \alpha^1 & \beta^2 & & & \\ \beta^2 & \alpha^2 & \beta^3 & & \\ & \beta^3 & \alpha^3 & \ddots & \\ & & \ddots & \ddots & \\ & & & & \beta^i \\ & & & \beta^i \alpha^i & \end{bmatrix}, \tag{14}$$

where $Q^i = [\mathbf{q}^1, \mathbf{q}^2, \dots, \mathbf{q}^i]$. Now by applying the orthogonal similarity transformation, the original system (1), becomes

$$(Q^n)^{\mathsf{T}} AQ^n (Q^n)^{\mathsf{T}} x = (Q^n)^{\mathsf{T}} b. \tag{15}$$

And the solution $\mathbf{x}$ can be achieved by transforming the solution $\mathbf{y}$ on the Krylov space back to the original space. From (15) by taking $T^n = (Q^n)^{\mathsf{T}} AQ^n$, we get

$$T^n \mathbf{y} = (Q^n)^{\mathsf{T}} \mathbf{b}, \tag{16}$$

$$\mathbf{x} = Q^n \mathbf{y}. \tag{17}$$

Introducing the decomposition for tridiagonal matrix T

$$T^i = L^i D^i \left(L^i\right)^{\top}. \tag{18}$$

We define

$$\mathbf{z}^i = \left(L^i\right)^{\top} \mathbf{y}^i = \left(L^i\right)^{\top} \left(Q^i\right)^{\top} \mathbf{x}^i \tag{19}$$

$$\text{and} \qquad L^i \left(C^i\right)^{\top} = \left(Q^i\right)^{\top}. \tag{20}$$

We obtain an iteration procedure, where the storage of the large $(i \times n)$ matrix $\mathbf{Q}^i$ is not required by

$$L^i D^i \mathbf{z}^i = \left(Q^i\right)^{\top} \mathbf{b}, \tag{21}$$

$$\mathbf{x}^i = C^i \mathbf{z}^i. \tag{22}$$

By applying equations (20) and (21), we have

$$\begin{bmatrix} 1 & & & & \\ \delta^2 & 1 & & & \\ & \delta^3 & 1 & & \\ & \ddots & & & \\ & & & 1 & \\ & & & \delta^i & 1 \end{bmatrix} \begin{bmatrix} d^1 & & & & \\ & d^2 & & & \\ & & d^3 & & \\ & & & \ddots & \\ & & & & d^i \end{bmatrix} \begin{bmatrix} \zeta^1 \\ \zeta^2 \\ \zeta^3 \\ \vdots \\ \zeta^i \end{bmatrix} \begin{bmatrix} (q^1)^{\top} \\ (q^2)^{\top} \\ (q^3)^{\top} \\ \vdots \\ (q^i)^{\top} \end{bmatrix} b. \tag{23}$$

$$\begin{bmatrix} 1 & & & & \\ \delta^2 & 1 & & & \\ & \delta^3 & 1 & & \\ & \ddots & & & \\ & & & 1 & \\ & & & \delta^i & 1 \end{bmatrix} \begin{bmatrix} (c^1)^{\top} \\ (c^2)^{\top} \\ (c^3)^{\top} \\ \vdots \\ (c^i)^{\top} \end{bmatrix} = \begin{bmatrix} (q^1)^{\top} \\ (q^2)^{\top} \\ (q^3)^{\top} \\ \vdots \\ (q^i)^{\top} \end{bmatrix} \tag{24}$$

reveals by simple transformation

$$\zeta^i = \frac{\left(q^i\right)^{\top} b - \delta^i \zeta^{i-1} d^{i-1}}{d^i}, \tag{25}$$

$$\mathbf{c}^i = \mathbf{q}^i - \delta^i \mathbf{c}^{i-1}. \tag{26}$$

From (22), $\mathbf{x}^i$ can be rewritten as

$$\mathbf{x}^i = \mathbf{C}^{i-1} \mathbf{z}^{i-1} + \zeta^i \mathbf{c}^i = \mathbf{x}^{i-1} + \zeta^i \mathbf{c}^i. \tag{27}$$

3.1. Lanczos Algorithm

The Lanczos algorithm for a symmetric $(n \times n)$ matrix A can be stated as follows.

STEP 1. Choose a starting vector $\mathbf{x}^0$, then

$$\mathbf{r}^1 = \mathbf{b} - A\mathbf{x}^0 \text{ with } \mathbf{r}^1 \neq 0, \qquad \mathbf{q}^0 = 0, \qquad \mathbf{c}^0 = 0,$$
$$d^0 = 0, \qquad \delta^1 = 0,$$
$$\beta^1 = \left\|\mathbf{r}^1\right\|_2,$$
$$\mathbf{q}^1 = \frac{\mathbf{r}^1}{\beta^1},$$

for $i = 1, 2, 3, \dots$.

STEP 2. Computation of $\mathbf{q}^i$, α^i, β^{i+1},

$$\alpha^i = \left(\mathbf{q}^i\right)^\top A\mathbf{q}^i, \tag{28}$$

$$\mathbf{r}^{i+1} = \beta^{i+1}\mathbf{q}^{i+1} = \mathbf{A}\mathbf{q}^i - \alpha^i\mathbf{q}^i - \beta^i\mathbf{q}^{i-1}, \tag{29}$$

$$\beta^{i+1} = \left\|\mathbf{r}^{i+1}\right\|, \tag{30}$$

$$\mathbf{q}^{i+1} = \frac{\mathbf{r}^{i+1}}{\beta^{i+1}}. \tag{31}$$

STEP 3. $LDL^\top$ factorization of T^j,

$$d^i = \alpha^i - \left(\delta^i\right)^2 d^{i-1}, \tag{32}$$

$$\delta^{i+1} = \frac{\beta^{i+1}}{d^i}. \tag{33}$$

STEP 4. The new estimation of $\mathbf{x}$,

$$\zeta^i = -\frac{\delta^i\zeta^{i-1}d^{i-1}}{d^i} \left(\text{with } \zeta^1 = \frac{\beta^1}{d^1}\right), \tag{34}$$

$$\mathbf{c}^i = \mathbf{q}^i - \delta^i\mathbf{c}^{i-1}, \tag{35}$$

$$\mathbf{x}^i = \mathbf{x}^{i-1} + \zeta^i\mathbf{c}^i. \tag{36}$$

STEP 5. Termination Criterion

$$\frac{\left\|\hat{r}^i\right\|}{\left\|r^1\right\|} = \frac{\left|\beta^{i+1}\zeta^i\right|}{\left|\beta^1\right|} < \epsilon. \tag{37}$$

THE PRECONDITIONED LANCZOS SOLVER. The convergence speed of the Lanczos algorithm is improved by using a suitable preconditioning to the original system (1). Therefore, the original system (1) is transformed into transformed system (2) with $\tilde{\mathbf{A}} = \mathbf{W}^{-1}\mathbf{A}\mathbf{W}^{-\top}$, $\tilde{\mathbf{x}} = \mathbf{W}^\top\mathbf{x}, \tilde{\mathbf{b}} = \mathbf{W}^{-1}\mathbf{b}$. Therefore, $\tilde{\mathbf{A}}$ has the same eigenvalues as $\mathbf{R}^{-1}\mathbf{A}$, where $\mathbf{R} = \mathbf{W}\mathbf{W}^\top$. Now the preconditioned Lanczos algorithm is given as follows.

3.2. Preconditioned Lanczos Algorithm

STEP 1. Choose $\mathbf{x}^0$, then

$$\mathbf{r}^1 = \mathbf{b} - \mathbf{A}\mathbf{x}^0, \qquad \mathbf{q}^0 = 0, \qquad \tilde{\mathbf{c}}^0 = 0,$$
$$\tilde{d}^0 = 0, \qquad \tilde{\delta}^1 = 0,$$
$$\tilde{\beta}^1 = \|\tilde{\mathbf{r}}^1\| = \sqrt{(\mathbf{r}^1)^\top}\mathbf{R}^{-1}\mathbf{r}^1 \text{ with } \mathbf{q}^1 = \frac{\mathbf{r}^1}{\tilde{\beta}^1},$$

for $i = 1, 2, 3, \ldots$.

STEP 2.

$$\tilde{\mathbf{u}}^i = R^{-1}\mathbf{q}^i, \tag{38}$$

$$\tilde{\alpha}^i = \left(\tilde{\mathbf{u}}^i\right)^\top A\tilde{\mathbf{u}}^i, \tag{39}$$

$$\mathbf{r}^{i+1} = \mathbf{A}\tilde{\mathbf{u}}^i - \tilde{\alpha}^i\mathbf{q}^i - \tilde{\beta}^i\mathbf{q}^{i-1}, \tag{40}$$

$$\tilde{\beta}^{i+1} = \sqrt{\left(\mathbf{r}^{i+1}\right)^\top \mathbf{R}^{-1}\mathbf{r}^{i+1}}, \tag{41}$$

$$\mathbf{q}^{i+1} = \frac{\mathbf{r}^{i+1}}{\tilde{\beta}^{i+1}}. \tag{42}$$

STEP 3.

$$\tilde{d}^i = \tilde{\alpha}^i - \left(\tilde{\delta}^i\right)^2 \tilde{d}^{i-1}, \tag{43}$$

$$\tilde{\delta}^{i+1} = \frac{\tilde{\beta}^{i+1}}{\tilde{d}^i}. \tag{44}$$

STEP 4.

$$\tilde{\zeta}^i = -\frac{\tilde{\delta}^i \tilde{\zeta}^{i-1} \tilde{d}^{i-1}}{\tilde{d}^i} \left(\text{with } \tilde{\zeta}^1 = \frac{\tilde{\beta}^1}{\tilde{d}^1}\right), \tag{45}$$

$$\tilde{\mathbf{c}}^i = R^{-1}\mathbf{q}^i - \tilde{\delta}^i \tilde{\mathbf{c}}^{i-1}, \tag{46}$$

$$\mathbf{x}^i = \mathbf{x}^{i-1} + \tilde{\zeta}^i \tilde{\mathbf{c}}^i. \tag{47}$$

STEP 5. Termination Criterion

$$\frac{\left\|\tilde{r}^i\right\|}{\left\|r^1\right\|} = \frac{\left|\tilde{\zeta}^i\right| \cdots \left\|r^{i+1}\right\|}{\left\|r^1\right\|} < \epsilon.$$

Note $\tilde{\alpha}^i$, $\tilde{\beta}^i$ are elements of transformed tridiagonal matrix $(\tilde{T}^i)$.

4. NUMERICAL EXPERIMENTS

The numerical experiments described in this section have been performed on a DEC-ALPHA 3000/600 OSF system. The single precision has been used throughout.

We have compared the Lanczos solver (Algorithm 3.1) with the conjugate gradient (CG) solver and Lanczos preconditioned solver (Algorithm 3.2) with preconditioned conjugate gradient solver (Algorithm 2.1) by considering the following model problems.

PROBLEM 1. Here we have considered the boundary value problem $\nabla^2\mathbf{u}(x,y) = 0$ defined in $0 \le x \le 1$, $0 \le y \le 1$, with boundary conditions are $\mathbf{u}(x,y) = 0$ on three sides and $\mathbf{u}(x,y) = 1$ on the remaining side. A five-point finite difference approximation with uniform grids is applied.

Figure 1 compares the convergence of the Lanczos algorithm and the conjugate gradient (CG) method with Incomplete Cholesky Preconditioning (ILU) and Polynomial Preconditioning

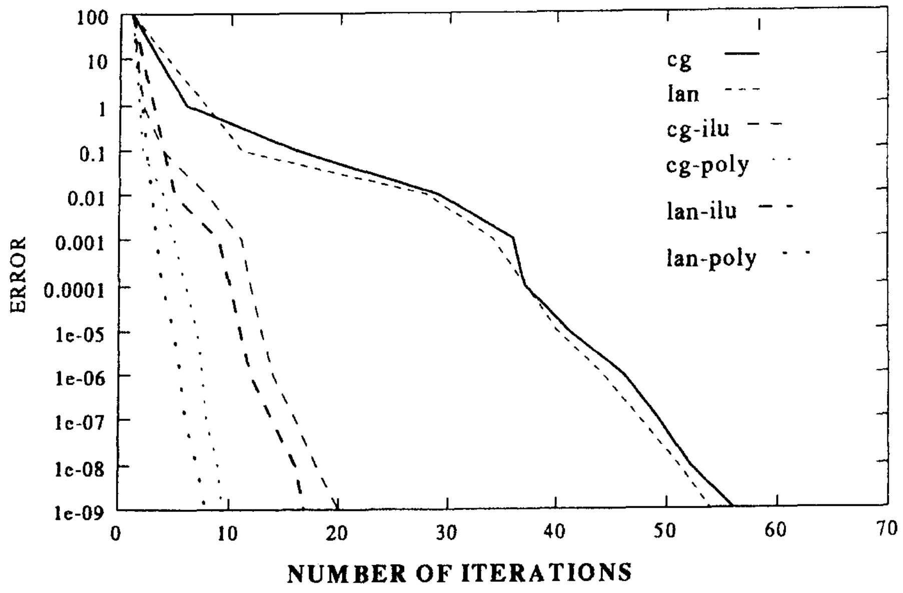

Figure 1. Convergence history of Lanczos and CG with preconditioning of 25 × 25 uniform grid.

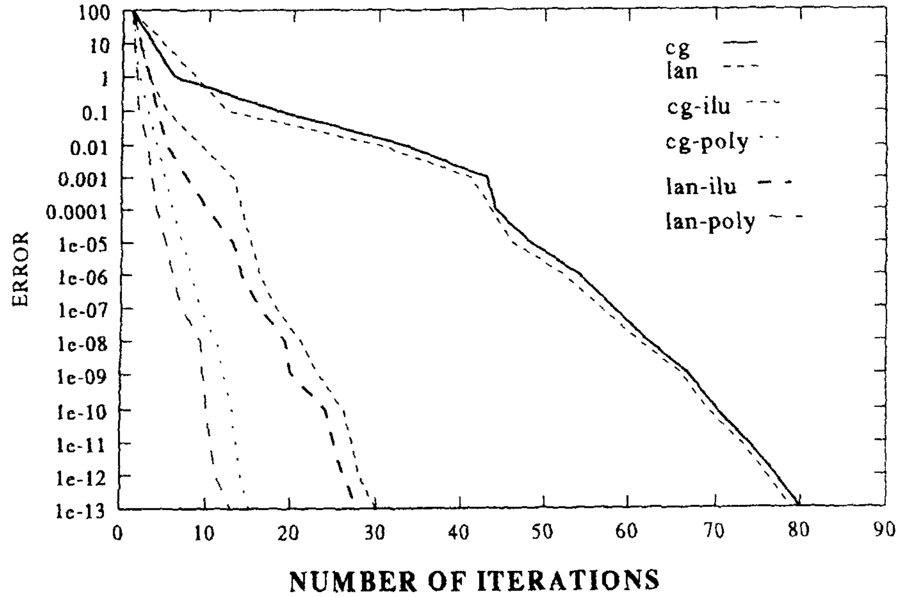

Figure 2. Convergence history of Lanczos and CG with preconditioning of 30 × 30 uniform grid.

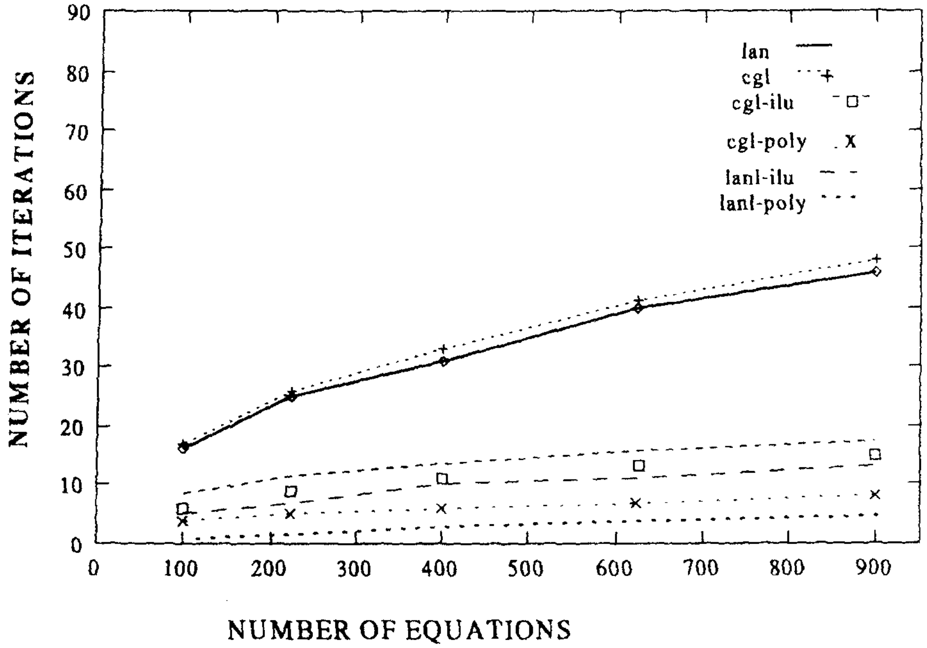

Figure 3. Number of iterations versus number of equations for Lanczos and CG with preconditioning.

(PP(5)) of 25×25 uniform grids. It is seen that the speed of convergence are equivalent, when zero initial guess have been taken. But the convergence speed of the preconditioned Lanczos algorithm is faster than the preconditioned conjugate gradient solver.

The convergence history of using the Lanczos algorithm and the CG method with preconditioning is summarized in Figure 2. The results show that the iteration number reduced when incomplete Choleski Factorization or polynomial precondition is applied. For example, at an error level of (1×10^{-6}), a four-fold reduction is obtained for both Lanczos-PP (5) and CG-PP (5)(i.e., with a fifth-order polynomial), and a three-fold reduction for both Lanczos-ILU and CG-ILU(i.e., with the ILU-preconditioning) of 30×30 uniform grids. The Lanczos process and

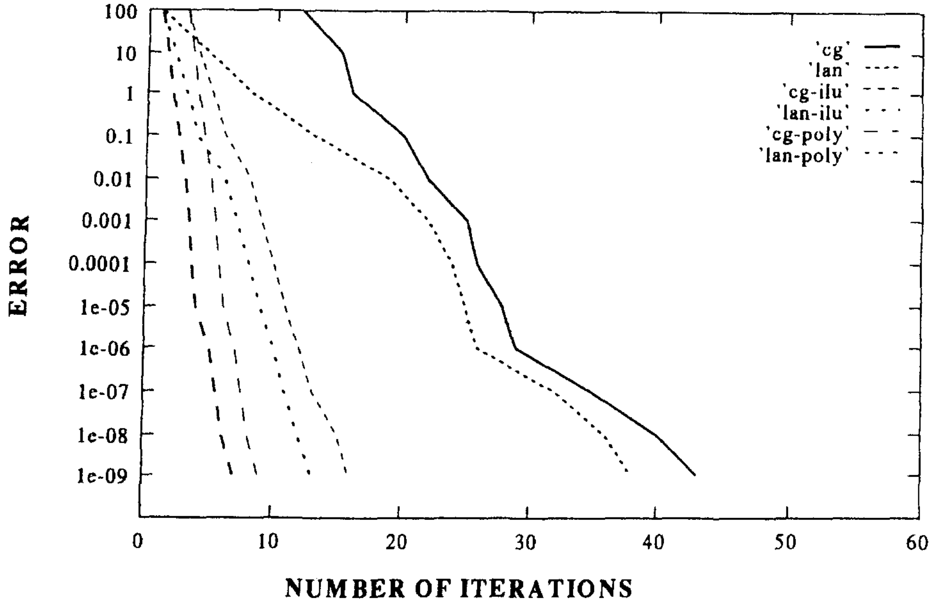

Figure 4. Convergence history of Lanczos and CG with preconditioning.

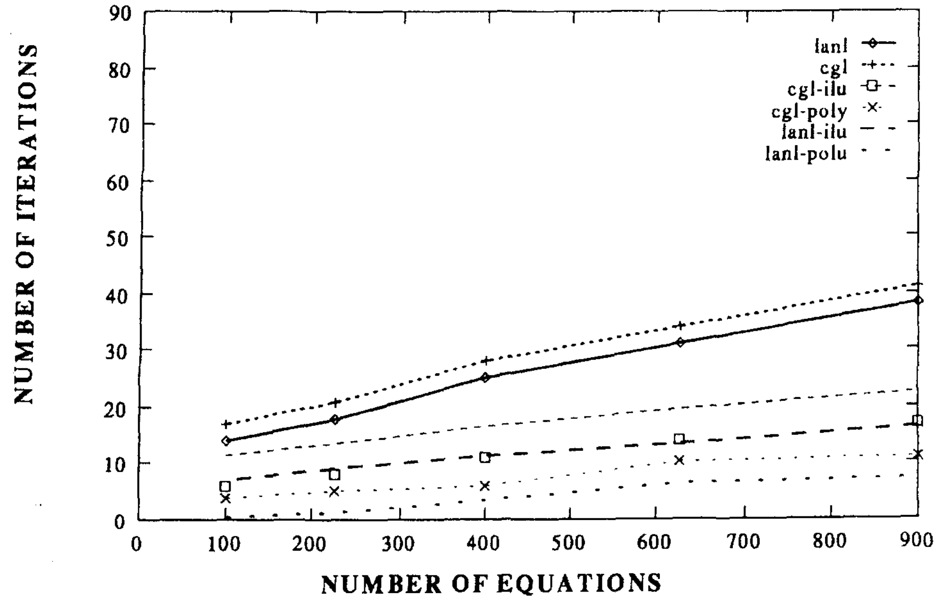

Figure 5. Number of iterations versus number of equations for Lanczos and CG with preconditioning.

CG based computation exhibits a linear relationship between the number of iterations and the number of equations as shown in Figure 3, which indicates the rate of increase in the number of iterations is lower than that in the number of equations. The linear relationship between the number of iterations and the equation numbers proves the Lanczos algorithm and CG method are good competent and stable for large-scale computing. Figure 6 gives the CPU-time for the Lanczos algorithm and CG method with preconditioning solvers for this model problem, and the details of the execution time (CPU-time), as well as number iterations are given in Table 1.

PROBLEM 2. Here we have considered the elliptic differential equation

$$\frac{\partial}{\partial x}\alpha(x,y)\frac{\partial u}{\partial x} + \frac{\partial}{\partial y}\alpha(x,y)\frac{\partial u}{\partial y} = f(x,y),$$

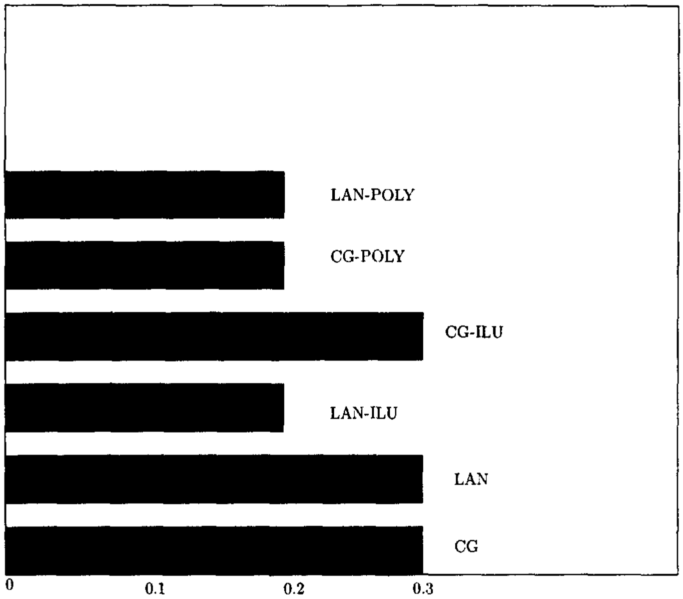

Figure 6. Relative time.

Table 1. CPU-time/Number of iterations for the corresponding errors of Lanczos and CG with preconditioning.

Methods	Res Norm Norm	CPU-Time	Iterns
LAN	1.0E − 05	0.3	47
	1.0E − 06	0.3	53
CG	1.0E − 05	0.3	48
	1.0E − 06	0.3	54
LAN-ILU	1.0E − 05	0.2	13
	1.0E − 06	0.2	14
CG-ILU	1.0E − 05	0.3	16
	1.0E − 06	0.3	17
LAN-PLY	1.00E − 05	0.2	4
	1.00E − 06	0.2	6
CG-PLY	1.00E − 05	0.3	7
	1.00E − 06	0.3	9

where $\alpha, \beta > 0$ are piecewise continuous functions and $\mathbf{u}$ is defined on the unit square $[0,1] \times [0,1]$. Here we used Dirichlet problem, $\alpha = \beta = 1$. In this case, the linear system $Ax = b$ generated by using five-point discretization of uniform grids, is given by

$$A = \begin{bmatrix} B & -I & & & \\ -I & B & -I & & \\ & -I & B & \ddots & \\ & & \ddots & \ddots & -I \\ & & & -I & B \end{bmatrix}, \quad \text{where } B = \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & -1 & 4 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 4 \end{bmatrix},$$

and matrix I is identity matrix, with a right-hand side consisting of all ones. Here we considered the initial guess is to be arbitrary. Figure 4 indicates the convergence behaviour for the Lanczos algorithm and CG method with ILU and PP (5) preconditioning. The linear relationship between the Lanczos algorithm and CG method along with preconditioning as shown in Figure 5, clearly, indicates that the convergence speed of the Lanczos algorithm is faster than the CG method for any arbitrary initial vector.

Table 2. CPU-time/Number of iterations for the corresponding errors of Lanczos and CG with preconditioning.

Methods	Res Norm	CPU-Time	Iterns
LAN	1.0E – 05	0.3	36
	1.0E – 06	0.3	38
CG	1.0E – 05	0.3	41
	1.0E – 06	0.3	43
LAN-ILU	1.0E – 05	0.2	11
	1.0E – 06	0.2	12
CG-ILU	1.0E – 05	0.3	14
	1.0E – 06	0.3	15
LAN-PLY	1.00E – 05	0.2	4
	1.00E – 06	0.2	6
CG-PLY	1.00E – 05	0.3	6
	1.00E – 06	0.3	8

REFERENCES

1. M.R. Hestenes and E. Stiefel, Method of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standards* **49**, 409–435, (1952).
2. H.W. Lin and L.Der. Chen, Application of the Krylov subspace methods to numerical heat transfer, *Numerical Heat Transfer, A* **30**, 249–270, (1996).
3. C. Lanczos, Solution of systems of linear equations by minimized iterations, *J. Res. Nat. Bur. Standards* **49**, 33–53, (1952).
4. M. Papadrakakis and S. Smerou, A new implementation of the Lanczos method in linear problems, *Int. J. Numer. Methods. Eng.* **29**, 141–159, (1990).
5. B.N. Parlett, A new look at the Lanczos algorithm for solving symmetric systems of linear equations, *Linear Algebra. Appl.* **29**, 323–346, (1980).
6. A.S. Householder, *The Theory of Matrices in Numerical Analysis*, Blaisdell, New York, (1964).
7. G.H. Golub and C.F. Van Loan, *Matrix Computations, Second Edition*, John Hopkins University Press, Baltimore, (1991).
8. Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, MA, (1996).
9. G. Golub and J.M. Ortega, *Scientific Computing, An Introduction with Parallel Computing*, Academic Press, New York, (1993).
10. C.P. Jackson and P.C. Robinson, A numerical study of various algorithms related to the preconditioned conjugate gradient method, *Int. J. Number. Methods. Eng.* **21**, 1315–1338, (1985).
11. O.G. Johnson, C.A. Micchelli and G. Paul, Polynomial preconditioners for conjugate gradient calculations, *SIAM. J. Numer. Anal.* **20**, 362–376, (1983).
12. S. Sundar, B.K. Bhagavan and K.S. Sastri, Comparison of Lanczos and CGS solvers for solving numerical heat transfer problems, *Computers Math. Applic.* **37** (8), 107–117, (1999).
13. A. Basermann, B. Reichel and C. Schelthoff, Preconditioned CG method for sparse matrices on massively parallel machines, *Parallel Computing* **23**, 381–398, (1997).