



# Newton-Preconditioned Krylov Subspace Solvers for System of Nonlinear Equations: A Numerical Experiment

S. SUNDAR\*, B. K. BHAGAVAN AND S. PRASAD

Department of Mathematics, IIT  
Kharagpur-721 302, India

*(Received February 2000; accepted March 2000)*

**Abstract**—In this paper, we present a numerical comparative study of the Newton-preconditioned Lanczos algorithms and Newton-preconditioned CG-like methods, with respect to convergence speed and CPU-time, by considering appropriate test problems. © 2000 Elsevier Science Ltd. All rights reserved.

**Keywords**—Krylov subspace, Preconditioned CG-like methods, Preconditioned Lanczos algorithm, System of nonlinear equations.

## 1. INTRODUCTION

Nonlinear systems of equations often 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 a huge sparse system of nonlinear equations. The Newton method is well known for solving nonlinear system of equations. At each Newton step of the Newton method, it requires us to solve a linear system of equations, where the system matrix is a Jacobian matrix and it converges rapidly for any sufficiently good initial guess. However, solving a system of linear equations at each Newton step becomes expensive if the number of unknowns are large and may not be justified when the iterative solution  $\mathbf{X}^{(new)}$  is far from a solution.

Chronopoulos [1] explained that the Newton method coupled with direct linear system solvers is an efficient way to solve the nonlinear systems when the dimension of the Jacobian matrix is small. When the Jacobian becomes large and sparse, some kind of iterative methods may be used. The Newton-iterative method uses a linear iterative method such as SOR [2] to approximate the solution of linear system in each Newton step.

The sequence of major iterates  $\mathbf{X}^{(new)}$  (outer) generated by such a method depends upon the particular linear iterative method chosen and the criteria used to stop the minor (inner) iteration. Therefore, the Krylov subspace methods (called a parameter free iterative method) play an important role in performing the convergence speed of the Newton-iterative method.

A nonlinear conjugate method has been introduced and analyzed by Daniel [3]. Fletcher

---

\*First author's email address: [slnt@maths.iitkgp.ernet.in](mailto:slnt@maths.iitkgp.ernet.in).

and Reeves [4] have obtained a nonlinear conjugate method which converges if the Jacobian is symmetric and uniformly positive definite. The number of Newton steps is decreased significantly when using a generalized CGS method as a linear solver for the Newton correction equation and it was so successful as a linear solver in the Newton scheme (see the work of Fokkema *et al.* [5]). The Lanczos algorithm has also been extended for solving a nonlinear system of equations (see [6]). Hybrid Krylov methods for a nonlinear system of equations were given by Brown and Saad [7].

In this paper, we have focused our attention on making a comparative numerical study of the Newton method with the preconditioned Krylov subspace methods. Our results reveal that the convergence speed of the Newton method with preconditioned CG-like methods is faster than the Newton method with a preconditioned Lanczos algorithm for any arbitrary initial vector.

The layout of the paper is as follows. The next section gives a brief account of the Newton method for a nonlinear system. The ensuing section describes the Krylov space methods for solving symmetric and nonsymmetric linear systems. The numerical experiments for sample problems are given in Section 4 and the conclusions are drawn in the final section.

## 2. NEWTON METHOD FOR NONLINEAR SYSTEMS OF EQUATIONS

Consider a nonlinear system of equations

$$\mathbf{F}(\bar{\mathbf{X}}) = 0, \quad (1)$$

where  $\mathbf{F} : R^n \rightarrow R^n$  is a nonlinear mapping with the following properties.

- (1) There exists an  $\bar{\mathbf{X}} \in R^n$  with  $\mathbf{F}(\bar{\mathbf{X}}) = 0$ .
- (2)  $\mathbf{F}$  is continuously differentiable in a neighborhood of  $\bar{\mathbf{X}}$ .
- (3)  $\mathbf{F}'(\bar{\mathbf{X}})$  (Jacobian of  $\mathbf{F}$ ) is nonsingular.

### Newton's Algorithm

STEP 1. Start with an initial guess  $\bar{\mathbf{X}}^0 = \bar{\mathbf{X}}^{(\text{old})}$ .

STEP 2. Set

$$\mathbf{A} = \mathbf{J}(\bar{\mathbf{X}}^{(\text{old})}), \quad (2)$$

$$\bar{\mathbf{b}} = -\mathbf{F}(\bar{\mathbf{X}}^{(\text{old})}). \quad (3)$$

STEP 3. Solve the system

$$\mathbf{A}\bar{\mathbf{y}} = \bar{\mathbf{b}}. \quad (4)$$

STEP 4.

$$\bar{\mathbf{X}}^{(\text{new})} = \bar{\mathbf{X}}^{(\text{old})} + \bar{\mathbf{y}}. \quad (5)$$

STEP 5. Check

$$\|\bar{\mathbf{X}}^{(\text{new})} - \bar{\mathbf{X}}^{(\text{old})}\| \text{ minimum.} \quad (6)$$

STEP 6. If not goto Step 2 with

$$\bar{\mathbf{X}}^{(\text{old})} = \bar{\mathbf{X}}^{(\text{new})}. \quad (7)$$

Computing the solution of (4) using a direct method at every Newton method can be expensive if the number of unknowns are large and may not be justified when  $\mathbf{X}^{(\text{new})}$  is far from  $\bar{\mathbf{X}}$ . Therefore, it is reasonable to use an iterative method to solve (4) only approximately. Step 3, often consisting of the Jacobian, is only required for performing Jacobian times vector operations. The explicit computation of the Jacobian requires additional sparse storage and computation time. Efficient methods to compute directly sparse Jacobian have been proposed by Griewank [8].

In our implementation, the Jacobian times vector operation is approximated the following divided difference [7,9]:

$$F'(\mathbf{x}^0)\mathbf{y} \equiv \frac{F(\mathbf{x}^0 + \epsilon\mathbf{y}) - F(\mathbf{x}^0)}{\epsilon}. \quad (8)$$

### 3. KRYLOV SUBSPACE METHODS

If  $\mathbf{b}$  is an arbitrary nonzero vector, a Krylov subspace of dimension  $m$  is defined by  $\mathbf{K}^m = \mathbf{K}^m(\mathbf{A}, \mathbf{b}) = \text{Span}(\mathbf{b}, \mathbf{A}\mathbf{b}, \mathbf{A}^2\mathbf{b}, \dots, \mathbf{A}^{m-1}\mathbf{b})$ .

The Krylov subspace methods are mainly based on projection process, either orthogonal or oblique, onto Krylov subspace. The general projection method for solving the linear system  $\mathbf{A}\bar{\mathbf{y}} = \bar{\mathbf{b}}$  is a method which seeks an approximate solution  $\mathbf{x}^m$  from an affine space  $\mathbf{x}^0 + \mathbf{K}^m$ , by imposing a Petro-Galerkin condition  $\bar{\mathbf{b}} - \mathbf{A}^m \perp \mathbf{L}^m$ . Here  $\mathbf{K}^m$  is the Krylov subspace of dimension  $m$  and  $\mathbf{K}^m = \text{Span}(\mathbf{r}^0, \mathbf{A}\mathbf{r}^0, \mathbf{A}^2\mathbf{r}^0, \dots, \mathbf{A}^{m-1}\mathbf{r}^0)$ , where  $\mathbf{r}^0 = \bar{\mathbf{b}} - \mathbf{A}\mathbf{x}^0$  is a initial residual vector. The principal idea here is to make the residual vector  $\mathbf{r}^m$  orthogonal to another Krylov subspace  $\mathbf{L}^m$  (called left subspace), usually different from  $\mathbf{K}^m$  [10].

The symmetric Lanczos process is based on an orthogonal projection process onto the Krylov subspace  $\mathbf{K}^m$ , whereas the nonsymmetric Lanczos process is based on oblique projection process. In the oblique projection process, the right space  $\mathbf{K}^m$  is a Krylov subspace  $\mathbf{K}^m = \text{Span}(\mathbf{v}^1, \mathbf{A}\mathbf{v}^1, \dots, \mathbf{A}^{m-1}\mathbf{v}^1)$ , where  $\mathbf{v}^1$  is a starting vector, while  $\mathbf{L}$  is Krylov subspace associated with  $\mathbf{L}^m = \text{Span}(\mathbf{w}^1, \mathbf{A}^\top \mathbf{w}^1, \dots, (\mathbf{A}^\top)^{m-1}\mathbf{w}^1)$ .

### 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.

#### Experiment 1

We have considered the following system of  $n$  equations in  $n$  unknowns:  $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}(\mathbf{x}) = \mathbf{0}$ , where  $\mathbf{x}$  is the vector with components  $x_1$  through  $x_n$ ,  $\mathbf{A}$  is an  $n \times n$  tridiagonal matrix:

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & & -1 & 2 \\ & & & & -1 & 2 \end{bmatrix},$$

and  $\mathbf{b}$  is a vector whose components depend on  $\mathbf{x}$ :

$$\mathbf{b}(\mathbf{x}) = \begin{bmatrix} e^{-x_1} \\ e^{-x_2} \\ e^{-x_3} \\ \vdots \\ e^{-x_{n-1}} \\ e^{-x_n} \end{bmatrix}.$$

The Jacobian of this system is a tridiagonal matrix

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} 2 + e^{-x_1} & -1 & & & & \\ -1 & 2 + e^{-x_2} & -1 & & & \\ & -1 & 2 + e^{-x_3} & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & & -1 & 2 + e^{-x_n} \\ & & & & -1 & 2 + e^{-x_n} \end{bmatrix},$$

which is symmetric whatever the value of  $\mathbf{x}$ . Here, we find the solution of the given system by employing the Newton's method with the CG, preconditioned CG-ICF, Lanczos and preconditioned Lanczos-ICF methods used for solving the intermediate linear system. The comparative study is made for the case when  $n = 30$  and for the initial guess  $\mathbf{x}^0 = (1, 1, \dots, 1)^\top$ .

The details of the execution (CPU) time as well as the number of Newton (outer) and linear (inner) iterations are summarized in Table 1. Here it can be observed that the total execution (CPU) time is the same for all different cases considered. The results clearly show that the Newton-CG method without preconditioning takes more number of Newton (outer) iterations as compared to the Newton-preconditioned CG method with incomplete Cholesky factorization (ICF) preconditioning. Although the Newton-Lanczos method with and without preconditioning takes the same number of Newton (outer) iterations, but when the comparison is made along with the total number of linear (inner) iterates, we find that the Newton-preconditioned Lanczos method with incomplete Cholesky preconditioning is faster than the Newton-Lanczos method without preconditioning. We also infer from Table 1 that the Newton-preconditioned CG method with ICF preconditioning is faster than the Newton-preconditioned Lanczos method with ICF preconditioning.

Table 1. Comparative results (outer, inner, CPU-time) for Experiment 1 using different Krylov subspace solvers.

Outer Error	Newton-CG			Newton-CG-ICF			Newton-LAN			Newton-LAN-ICF		
$1.0 \times 10^{-01}$	5	99	0.02	5	5	0.02	5	120	0.02	5	19	0.02
$1.0 \times 10^{-02}$	7	140	0.02	6	6	0.02	6	151	0.02	6	24	0.02
$1.0 \times 10^{-03}$	8	161	0.02	6	6	0.02	6	151	0.02	6	24	0.02
$1.0 \times 10^{-04}$	9	182	0.02	7	7	0.02	7	167	0.02	7	27	0.02
$1.0 \times 10^{-05}$	10	203	0.02	7	7	0.02	7	167	0.02	7	27	0.02
$1.0 \times 10^{-06}$	12	245	0.02	7	7	0.02	7	167	0.02	7	27	0.02

## Experiment 2

We consider the following system of  $n$  nonlinear equations in  $n$  unknowns [8]:

$$\mathbf{F}(x) = 0,$$

where

$$\begin{aligned} f_1(x) &= (3 - 5x_1)x_1 + 1 - 2x_2, \\ f_i(x) &= (3 - 5x_i)x_i + 1 - x_{i-1} - 2x_{i+1}, \quad i = 2, 3, \dots, (n-1), \end{aligned}$$

and

$$f_n = (3 - 5x_n)x_n + 1 - x_{n-1}.$$

The Jacobian of the above system of equations is given by

$$J(x) = \begin{bmatrix} 3 - 10x_1 & -2 & & & & \\ -1 & 3 - 10x_2 & -2 & & & \\ & -1 & 3 - 10x_3 & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & & -1 & \\ & & & & -1 & 3 - 10x_n \end{bmatrix},$$

which can never be symmetric, whatever the value of  $x$ . As the Jacobian is nonsymmetric, we solve the given system using the Newton method with the CGS (conjugate gradient squared method), preconditioned CGS-ILU, and the Lanczos biorthogonalization methods used for solving the intermediate linear system. We solve the problem for the case when  $n = 30$  and for the initial

Table 2. Comparative results (outer, inner, CPU-time) for Experiment 2 using different Krylov subspace solvers.

Outer Error	Newton-CGS			Newton-CGS-ILU			Newton-LAN		
$1.0 \times 10^{-01}$	4	12	0.03	4	4	0.03	4	41	0.03
$1.0 \times 10^{-02}$	4	12	0.03	4	4	0.03	4	41	0.03
$1.0 \times 10^{-03}$	5	15	0.03	5	5	0.03	5	48	0.03
$1.0 \times 10^{-04}$	5	15	0.03	5	5	0.03	5	48	0.03
$1.0 \times 10^{-05}$	6	18	0.03	6	6	0.03	6	55	0.03
$1.0 \times 10^{-06}$	6	18	0.03	6	6	0.03	6	55	0.03

guess  $\mathbf{x}^0 = (-1.2, -1.2, \dots, -1.2)^\top$ . The results obtained by using the various Krylov subspace linear solvers are compared and summarized in Table 2.

The results from Table 2 reveal that the total execution (CPU) time as well as the number of Newton (outer) iterations are the same for all the different cases considered. On comparing the total number of inner iterations required by each of the three different methods, we find that the Newton-preconditioned CGS method with incomplete LU (ILU) decomposition is the fastest among them. It should also be noted that the linear system in the Newton-preconditioned CGS method with ILU preconditioning is solved in a single step for each Newton (outer) iteration.

## 5. CONCLUSIONS

Our numerical experiment on the problem with a symmetric Jacobian revealed that the Newton-preconditioned CG method with incomplete Cholesky preconditioning is faster than the Newton-CG, Newton-Lanczos, and the Newton-preconditioned Lanczos methods. It has also been established that the preconditioned Krylov subspace linear solvers are faster than the ordinary Krylov subspace solvers. From the second problem, we can conclude that the Newton-preconditioned CGS method with ILU preconditioning is a faster method than the Newton-CGS and the Newton-Lanczos biorthogonalization method for nonlinear systems with nonsymmetric Jacobian.

While finding the solution of our problems, we have not considered the polynomial preconditioning technique, even though polynomial preconditioning is considered to be better than the incomplete Cholesky factorization preconditioning. This is because polynomial preconditioning requires  $\lambda_{\max}$  at each Newton iteration (which by itself a huge task), and may drastically reduce the efficiency.

## REFERENCES

1. A.T. Chronopoulos, Nonlinear CG-Like iterative methods, *J. Comput. Appl. Math.* **40**, 73–89 (1992).
2. J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, (1970).
3. J.W. Daniel, Convergence of the Conjugate Gradient method with computationally convenient modifications, *Numer. Math.* **10**, 125–131 (1967).
4. R. Fletcher and C.M. Reeves, Function minimization by conjugate gradients, *Comput. J.* **7**, 149–154 (1964).
5. D.R. Fokkemma, L.G. Gerard and H.A. Van der Vorst, Generalized conjugate gradient squared, *J. Comput. Appl. Math.* **71**, 125–146 (1996).
6. S. Krzyworzcka, Extension of the Lanczos and CGS methods to systems of nonlinear equations, *J. Comp. Appl. Math.* **69**, 181–190 (1996).
7. P.N. Brown and Y. Saad, Hybrid Krylov methods for nonlinear system of equations, *SIAM J. Sci. Statist. Comput.* **11**, 450–481 (1990).

8. A. Griewank, On automatic differentiation, In *Mathematical Programming*, (Edited by M. Iri and K. Tanabe), Kluwer Academic, Dordrecht, (1989).
9. W. Gear and Y. Saad, Iterative solution of linear equations in ODE codes, *SIAM J. Sci. Statist. Comput.* **4**, 583–601 (1983).
10. 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).