International Journal of Engineering, Science and Mathematics
Vol. 8 Issue 1, January 2019,
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

STUDY OF CONJUGATE GRADIENT METHOD FOR SYSTEM OF LINEAR EQUATIONS USING PRECONDITIONING

Asha C. S. [1], Patil Shankar Shrimant [2], Lakshmi B. N. [3]
[1]Assistant Professor., [3]Research Scholar, P. G. Dept. of Mathematics and Research Centre in Applied Mathematics, M E S College of Arts, Commerce and Science, 15th cross, Malleshwaram, Bangalore-560003.
[2]Lecturer, Sahakari Vidya Kendra Polytechnic, C.A. No. 03, 41st cross, East End B Main Road, Jayanagar 9^{th} Block, Bangalore - 560069.
Email ID: csasharukmini@gmail.com, shripatil4753@gmail.com, lakshmibn95@gmail.com

Abstract:
System of linear equations are solved either by direct methods which yield exact solution or by iterative methods which gives an approximation solution. In computational mathematics, iterative proccess plays a vital role that uses an initial guess to determine the approximation solution of any given problem. [7, 9, 10]There are several methods to solve system of linear and nonlinear equations, such as iterative methods, approximation methods, elimination methods and interpolation methods[13, 16, 17]. In this paper we use Conjugate Gradient Method to solve system of linear equations, as this is one of the most popular and well known iterative techniques for solving sparse symmetric positive definite systems of linear equations[24]. In particular, we discuss about pure conjugate gradient method and preconditioned conjugate gradient method. Numerical examples are shown for each method and comparison of these method is done based on the number of iterations and faster convergence. Thus we observe that the preconditioned conjugate gradient method is more preferred when the system is large because it gives less number of iterations and converge faster.

Keywords: Linear, Search directions, Orhogonality, Convergence, Preconditioner.
Mathematics Subject Classification: **65-XX**

1 *Introduction*

Linear system is a collection of two or more linear equations involving the same set of variables. We have come across some of the iterative methods[1, 5] namely Gauss Elimination, Gauss Jacobi, Gauss-Seidel, Relaxation methods, etc. The Conjugate Gradient Method is used to an $n \times n$ positive definite linear system. Generally this method is employed as an iterative approximation method for solving very large systems where it is not practically possible to solve with a direct method. The method of conjugate gradient was developed by E-Stiefel and M. R. Hestences (1952)[1, 4]. Shortly thereafter, they jointly published what is considered the seminal reference on conjugate gradient[18]. Fletcher et. al. generalised this method for nonlinear problems in 1964[22, 24] based on the work by Davidon[20, 24] and Powell et. al.[21]. Daniel analysed about the convergence of nonlinear conjugate gradient with inexact line searches[19, 24]. Gilbert et. al. discussed about the choice of β for nonlinear conjugate gradient[23, 24]. Peter et. al. have considered different types of sparse matrices. For these matrix types, the counter-movement of the applicability of discretization schemes and the possibilities of performance tuning is shown[25, 24]. O. Osadcha et. al. have done comparison of steepest descent method with Conjugate gradient method and implemented to solve systems of linear equations. They have observed that steepest descent method is faster method, because it solve equations in less amount of time. [26, 24]Conjugate gradient method is slower, but more productive, because, it converges after less iterations. So, they have reported that one method can be used, when we want to find solution very fast and another can be converge to maximum in less iteration[26, 24]. This method earlier developed as direct method, but later on became popular for its properties as an iterative method, especially in the field of using preconditioning techniques[1, 2, 13, 5]. This method solves the given system of linear equations by finding n conjugate vectors, and then computing the coeffcients. The conjugate gradient methods exploit the conjugacy concept by using gradient information[3]. In this method, the search direction is established as a linear combination of all the previous search directions and newly determined gradient. This method is stable and quadratically convergent and is used extensively. Rao (1979) found it to be the best general-purpose unconstrained optimization technique that use derivatives[11, 12]. In general nonlinear cases, the number of iterations is not fixed, but still the method converges very rapidly. Another advantage of the conjugate gradient method is that we go downhill not along a line, but on various planes. In this case, we can overcome small local minima of the misfit functional and go faster directly to its global minimum. This methods begins with choosing search directions in such a way that the sequence of approximation coverges rapidly to the solution[1]. Residual vector takes the form as mentioned in steepest descent method but that can not be used for linear system as it is slow convergence. As an alternative approach a set of vectors uses A-orthogonality condition. These A-ortogonal vectors associated with positive definite matrix is linearly dependent[1, 2]. In this method we find, for the quadratic function the negative residual is equal to the steepest descent or negative gradient direction. Hence the name of the method is the conjugate gradient method. Some times pure Conjugate Gradient method simply does not converge as fast as we would like[27]. This slow convergence is may be because the system is ill-conditioned. An ill-conditioned system is a system with a high condition number κ. The condition number for a system like ours can be expressed as $\kappa = \frac{\lambda_{max}}{\lambda_{min}}$[27]. It can be shown that for the conjugate gradient method, the number of iterations required to reach convergence is proportional to the condition number. Therefore we use method of preconditiong to obtain faster convergence. In this method, we use the matrix A is positive definite and symmetric. We obtain more effective calculations and good results in only about $\sqrt{n}$ iterations, if the matrix has been preconditioned. Therefore preconditioned conjugate gradient method is preffered over other iterative methods. [6, 8]The method has one of the major advantage that to solve large-scale problems ($5{,}00{,}000\ variables$) with lower storage as compared to some other existing methods. This method is especially used when the problems arise in the solution of boundary value problem. Thus the conjugate gradient method is superior than any other elimination methods.

2 Methodology

2.1 Theorem

[1, 2] *The vector x^* is a solution to the positive definite linear system $Ax = b$ if and only if x^* produces the minimal value of $g(x) = \langle x, Ax\rangle - 2\langle x, b\rangle$.* (1)

Proof: Let x and $v \neq 0$ be fixed vectors and t be a real number variable, then equation (1) becomes,

$$g(x+tv) = \langle x+tv, Ax+tAv\rangle - 2\langle x+tv, b\rangle,$$

$$g(x+tv) = g(x) - 2t\langle v, b-Ax\rangle + t^2\langle v, Av\rangle. \quad (2)$$

We define the quadratic function h in t by keeping x and v fix, we get

$$h(t) = g(x+tv). \quad (3)$$

$$h^{'}(t) = -2\langle v, b-Ax\rangle + 2t\langle v, Av\rangle, \quad (4)$$

h assumes a minimal value when $h^{'}(t) = 0$, because of its t^2 coefficient, $\langle v, Av\rangle$ is positive.

$h^{'}(t)$ minimum occurs when

$$\hat{t} = \frac{\langle v, b-Ax\rangle}{\langle v, Av\rangle}, \quad (5)$$

from equation (2),

$$h(\hat{t}) = g(x+\hat{t}v),$$

$$h(\hat{t}) = g(x) - \frac{\langle v, b-Ax\rangle^2}{\langle v, Av\rangle}. \quad (6)$$

Thus, for any vector $v \neq 0$, we have $g(x+\hat{t}v) < g(x)$ unless $\langle v, b-Ax\rangle = 0$, in which case

$$g(x) = g(x+\hat{t}v). \quad (7)$$

This is the basic result required to prove theorem.

Suppose x^* satisfies $Ax^* = b$, then $\langle v, b-Ax^*\rangle = 0$, for any vector v and $g(x)$ can not be made any smaller than $g(x^*)$. Thus, x^* minimizes g.

On the other hand, suppose that x^* is a vector that minimizes g. Then for any vector v, we have

$$g(x^*+\hat{t}v) \geq g(x^*). \quad (8)$$

Thus, $\langle v, b-Ax^*\rangle = 0$. This implies that $b - Ax^* = 0$,

$$Ax^* = b. \quad (9)$$

2.2 Conjugate Gradient Method

We choose x, an approximate solution to $Ax^* = b$ and $v \neq 0$ which gives a search direction from x to new improved approximation.

Let,

$$r = b - Ax, \quad (10)$$

be the residual vector associated with x and we know that,

$$t = \frac{\langle v, b-Ax\rangle}{\langle v, Av\rangle},$$

$$t = \frac{\langle v, r\rangle}{\langle v, Av\rangle}. \quad (11)$$

If $r \neq 0$ and if v and r are not orthogonal then $x + tv$ gives a smaller value for g than $g(x^*)$ and is closer to x^* than x.

This suggest the following method:

Let $x^{(0)}$ be an initial approximation to x^* and let $v^{(1)} \neq 0$ be an initial search direction. For $k = 1,2,3, \ldots$, we compute

$$t_k = \frac{\langle v^{(k)}, b - Ax^{(k-1)} \rangle}{\langle v^{(k)}, Av^{(k)} \rangle}, \qquad (12)$$

$$x^{(k)} = x^{(k-1)} + t_k v^{(k)}. \qquad (13)$$

Choose a new search direction $v^{(k+1)}$. The main object is to make this selection so that the sequence of approximations $\{x^{(k)}\}$ converges rapidly to x^*.

2.3 Method of choosing search directions:

We have, g as a function of the components of $x = (x_1, x_2, \ldots, x_n)^t$.
Then,

$$g(x_1, x_2, \ldots, x_n) = \langle x, Ax \rangle - 2\langle x, b \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j - 2\sum_{i=1}^{n} x_i b_i. \qquad (14)$$

Taking partial derivatives with respect to the component variables x_k gives,

$$\frac{\partial g}{\partial x_k}(x) = 2\sum_{i=1}^{n} a_{ki} x_i - 2b_k, \qquad (15)$$

which is the k^{th} component of the vector $2(Ax - b)$.
The gradient of g is,

$$\nabla g(x) = \left(\frac{\partial g}{\partial x_1}(x), \frac{\partial g}{\partial x_2}(x), \ldots, \frac{\partial g}{\partial x_n}(x) \right)^t = 2(Ax - b) = -2r. \qquad (16)$$

We know that, the direction of greatest decrease in the value of $g(x)$ is the direction given by $-\nabla g(x)$, in the direction of the residual r. The method chooses,

$$v^{(k+1)} = r^{(k)} = b - Ax^{(k)}, \qquad (17)$$

is the **steepest descent** method [1]. Here we should observe that in the above method we need to compute the set of search directions before applying the conjugate gradient method. But it is difficult to find such search directions when we find the solution for large-scale system. This is one major drawback of this method. Thus we approach alternative method to find set of search directions.

3 Alternative Method

An alternative approach uses a set of non zero direction vectors $\{v^{(1)}, \ldots, v^{(n)}\}$ that satisfy

$$\langle v^{(i)}, Av^{(j)} \rangle = 0, \quad if \;\; i \neq j, \qquad (18)$$

this condition is called an ***A-orthogonality condition*** and the set of vectors $\{v^{(1)}, \ldots, v^{(n)}\}$ is said to be ***A-orthogonal***. With these set of A-orthogonal vectors[3], we can say that the positive definite matrix A is linearly independent. The set of search directions gives

$$t = \frac{\langle v^{(k)}, b - Ax^{(k-1)} \rangle}{\langle v^{(k)}, Av^{(k)} \rangle},$$

$$t = \frac{\langle v^{(k)}, r^{(k-1)} \rangle}{\langle v^{(k)}, Av^{(k)} \rangle}, \qquad (19)$$

and

$$x^{(k)} = x^{(k-1)} + t_k v^{(k)}. \qquad (20)$$

Now, we prove the theorem which gives convergence in at most n-step with choice of search directions. Then it produces the exact solution as a direct method, assuming that the arithmetic is exact.

3.1 Convergence

3.1.1 Theorem

Let $\{v^{(1)},\dots,v^{(n)}\}$ be an A-orthogonal set of nonzero vectors associated with the positive definite matrix A, and let $x^{(0)}$ be arbitrary.

Define,

$$t_k = \frac{\langle v^{(k)}, b - Ax^{(k-1)}\rangle}{\langle v^{(k)}, Av^{(k)}\rangle}, \tag{21}$$

and

$$x^{(k)} = x^{(k-1)} + t_k v^{(k)}. \tag{22}$$

for $k = 1,2,\dots,n$. Then, assuming exact arithmetic, $Ax^{(n)} = b$.

Proof:

Since, for each $k = 1,2,\dots,n$,

$$x^{(k)} = x^{(k-1)} + t_k v^{(k)}, \tag{23}$$

we have,

$$Ax^{(n)} = Ax^{(n-1)} + t_n Av^{(n)},$$
$$Ax^{(n)} = Ax^{(0)} + t_1 Av^{(1)} + t_2 Av^{(2)} + \cdots + t_n Av^{(n)}. \tag{24}$$

Subtracting b from equation (24) , we get

$$Ax^{(n)} - b = Ax^{(0)} - b + t_1 Av^{(1)} + t_2 Av^{(2)} + \cdots + t_n Av^{(n)}.$$

Now, we take inner product on both the sides with the vector $v^{(k)}$ and using the properties of inner products and since A is symmetric to get,

$$\langle Ax^{(n)} - b, v^{(k)}\rangle = \langle Ax^{(0)} - b, v^{(k)}\rangle + t_1\langle Av^{(1)}, v^{(k)}\rangle + \cdots + t_n\langle Av^{(n)}, v^{(k)}\rangle,$$
$$= \langle Ax^{(0)} - b, v^{(k)}\rangle + t_1\langle v^{(1)}, Av^{(k)}\rangle + \cdots + t_n\langle v^{(n)}, Av^{(k)}\rangle. \tag{25}$$

By A-orthogonality property, for each k,

$$\langle Ax^{(n)} - b, v^{(k)}\rangle = \langle Ax^{(0)} - b, v^{(k)}\rangle + t_k\langle v^{(k)}, Av^{(k)}\rangle. \tag{26}$$

As,

$$t_k\langle v^{(k)}, Av^{(k)}\rangle = \langle v^{(k)}, b - Ax^{(k-1)}\rangle,$$
$$= \langle v^{(k)}, b - Ax^{(0)}\rangle + \langle v^{(k)}, Ax^{(0)} - Ax^{(1)}\rangle$$
$$+ \cdots + \langle v^{(k)}, Ax^{(k-2)} - Ax^{(k-1)}\rangle. \tag{27}$$

for any i,

$$x^{(i)} = x^{(i-1)} + t_i v^{(i)}, \tag{28}$$

and

$$Ax^{(i)} = Ax^{(i-1)} + t_i Av^{(i)},$$
$$Ax^{(i-1)} - Ax^{(i)} = -t_i Av^{(i)}.$$

Thus,

$$t_k\langle v^{(k)}, Av^{(k)}\rangle = \langle v^{(k)}, b - Ax^{(0)}\rangle - t_1\langle v^{(k)}, Av^{(1)}\rangle - \cdots - t_{k-1}\langle v^{(k)}, Av^{(k-1)}\rangle. \tag{29}$$

Because of the A-orthogonality, $\langle v^{(k)}, Av^{(i)}\rangle = 0$, for $i \neq k$,

$$\langle v^{(k)}, Av^{(k)}\rangle t_k = \langle v^{(k)}, b - Ax^{(0)}\rangle. \tag{30}$$

From equation (26),

$$\langle Ax^{(n)} - b, v^{(k)}\rangle = \langle Ax^{(0)} - b, v^{(k)}\rangle + \langle v^{(k)}, b - Ax^{(0)}\rangle = 0. \tag{31}$$

Hence, the vector $Ax^{(n)} - b$ is orthogonal to the A-orthogonal set of vectors $\{v^{(1)},\dots,v^{(n)}\}$.

From this, it follows that

$$Ax^{(n)} - b = 0,$$
$$Ax^{(n)} = b. \tag{32}$$

3.2 Orthogonality of residual vector and direction vectors

3.2.1 Theorem

The residual vector $r^{(k)}$, where $k = 1,2,\dots,n$, for a conjugate direction method, satisfy the equations

$$\langle r^{(k)}, v^{(j)}\rangle = 0, \quad foreach \quad j = 1,2,\dots,k. \tag{33}$$

Proof:

To construct the direction vectors $\{v^{(1)}, v^{(2)}, \dots\}$ and the approximations $\{x^{(1)}, x^{(2)}, \dots\}$. We take an initial approximation $x^{(0)}$ and we use the steepest descent direction

$$r^{(0)} = b - Ax^{(0)}, \tag{34}$$

as the first search direction $v^{(1)}$.

Assume that the conjugate search directions $v^{(1)}, v^{(2)}, \dots, v^{(k-1)}$ and the approximations $x^{(1)}, x^{(2)}, \dots, x^{(k-1)}$ which is computed with

$$x^{(k-1)} = x^{(k-2)} + t_{k-1}v^{(k-1)}. \tag{35}$$

A-orthogonality condition is,

$\langle v^{(i)}, Av^{(j)}\rangle = 0 \quad and \quad \langle r^{(i)}, r^{(j)}\rangle = 0, \quad for\ i \neq j$. If $x^{(k-1)}$ is the solution to $Ax = b$ then we get vectors.

If $r^{(k-1)} = b - Ax^{(k-1)} \neq 0$ and we know that, $\langle r^{(k-1)}, v^{(i)}\rangle = 0\ foreach\ i = 1,2,\dots,k-1$.

We use $r^{(k-1)}$ to find $v^{(k)}$ by taking,

$$v^{(k)} = r^{(k-1)} + s_{k-1}v^{(k-1)}. \tag{36}$$

We want to choose s_{k-1} such that,

$$\langle v^{(k-1)}, Av^{(k)}\rangle = 0, \tag{37}$$

by equation (36),

$$Av^{(k)} = Ar^{(k-1)} + s_{k-1}Av^{(k-1)},$$

and

$$\langle v^{(k-1)}, Av^{(k)}\rangle = \langle v^{(k-1)}, Ar^{(k-1)}\rangle + s_{k-1}\langle v^{(k-1)}, Av^{(k-1)}\rangle, \tag{38}$$

When

$$s_{k-1} = -\frac{\langle v^{(k-1)}, Ar^{(k-1)}\rangle}{\langle v^{(k-1)}, Av^{(k-1)}\rangle}, \tag{39}$$

then

$$\langle v^{(k-1)}, Av^{(k)}\rangle = 0.$$

We can also show that, $\langle v^{(k)}, Av^{(i)}\rangle = 0$, for each $i = 1,2,\dots,k-2$ with the choice of s_{k-1}. Thus $v^{(1)}, v^{(2)}, \dots, v^{(k-1)}$ is an A-orthogonal set.

We have, $v^{(k)}$ (chosen), now we compute

$$\begin{aligned} t_k &= \frac{\langle v^{(k)}, r^{(k-1)}\rangle}{\langle v^{(k)}, Av^{(k)}\rangle}, \\ &= \frac{\langle r^{(k-1)} + s_{k-1}v^{(k-1)}, r^{(k-1)}\rangle}{\langle v^{(k)}, Av^{(k)}\rangle}, \\ t_k &= \frac{\langle r^{(k-1)}, r^{(k-1)}\rangle}{\langle v^{(k)}, Av^{(k)}\rangle} + s_{k-1}\frac{\langle v^{(k-1)}, r^{(k-1)}\rangle}{\langle v^{(k)}, Av^{(k)}\rangle}, \end{aligned} \tag{40}$$

we know that, $\langle v^{(k-1)}, r^{(k-1)}\rangle = 0$,

equation (40) becomes,

$$t_k = \frac{\langle r^{(k-1)}, r^{(k-1)}\rangle}{\langle v^{(k)}, Av^{(k)}\rangle}. \tag{41}$$

Thus,

$$x^{(k)} = x^{(k-1)} + t_k v^{(k)}. \tag{42}$$

We have to compute $r^{(k)}$, multiply by A and subtract b from equation (42), we get

$$Ax^{(k)} - b = Ax^{(k-1)} - b + t_k A v^{(k)},$$
$$r^{(k)} = r^{(k-1)} - t_k A v^{(k)}. \quad (43)$$

equation (43) gives,

$$\langle r^{(k)}, r^{(k)} \rangle = \langle r^{(k-1)}, r^{(k)} \rangle - t_k \langle A v^{(k)}, r^{(k)} \rangle = -t_k \langle r^{(k)}, A v^{(k)} \rangle.$$

and also from equation (??),

$$\langle r^{(k-1)}, r^{(k-1)} \rangle = t_k \langle v^{(k)}, A v^{(k)} \rangle.$$

Thus, we get

$$s_k = -\frac{\langle v^{(k)}, A r^{(k)} \rangle}{\langle v^{(k)}, A v^{(k)} \rangle},$$
$$s_k = \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle r^{(k-1)}, r^{(k-1)} \rangle}. \quad (44)$$

We get the values,

$$r^{(0)} = b - Ax^{(0)}; \quad v^{(1)} = r^{(0)}; \;\; for \;\; k = 1,2,\dots,n, \quad (45)$$

$$t_k = \frac{\langle r^{(k-1)}, \langle r^{(k-1)} \rangle}{\langle v^{(k)}, A v^{(k)} \rangle}, \quad (46)$$

$$x^{(k)} = x^{(k-1)} + t_k v^{(k)}, \quad (47)$$

$$r^{(k)} = r^{(k-1)} - t_k A v^{(k)}, \quad (48)$$

$$s_k = \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle r^{(k-1)}, r^{(k-1)} \rangle}, \quad (49)$$

$$v^{(k+1)} = r^{(k)} + s_k v^{(k)}. \quad (50)$$

Example:

$$410x_1 - x_2 = 9,$$
$$-x_1 + 10x_2 - 2x_3 = 7, \quad (51)$$
$$-2x_2 + 10x_3 = 6.$$

Solution:The system (51) can be written in the form of coefficient matrix,

$$A = [10 \; -1 \; 0 \; -1 \; 10 \; -2 \; 0 \; -2 \; 10]. \quad (52)$$

$$r^{(0)} = b - Ax^{(0)},$$
$$= (9\;7\;6)^t - [10 \; -1 \; 0 \; -1 \; 10 \; -2 \; 0 \; -2 \; 10][0\;0\;0] = b = (9,7,6)^t. \quad (53)$$

$$v^{(1)} = r^{(0)} = (9,7,6)^t. \quad (54)$$

$$t_1 = \frac{\langle r^{(0)}, r^{(0)} \rangle}{\langle v^{(1)}, A v^{(1)} \rangle} = \frac{166}{1366} = 0.121522694. \quad (55)$$

$$x^{(1)} = x^{(0)} + t_1 v^{(1)},$$
$$= (0,0,0)^t + (0.121522694)(9,7,6)^t,$$
$$x^{(1)} = (1.093704246, 0.850658858, 0.729136164)^t. \quad (56)$$

Proceeding in this way, we get

$$x^{(2)} = (0.99931295, 0.964273445, 0.778426657\,)^t. \quad (57)$$
$$x^{(3)} = (0.99578954, 0.957894655, 0.791578911\,)^t. \quad (58)$$

This is the required solution for the given linear system.

We extend the conjugate gradient method by including preconditioning. We use this preconditioning only when the matrix is ill-conditioned [5]. Normally, Conjugate gradient method is highly susceptible to rounding of error if the matrix is ill-conditioned. In preconditioning, the conjugate gradient method is not applied directly to the given matrix A, but to another positive definite matrix which has a smaller condition number [5]. We have to choose this preconditioning in such a way that once the solution to this new system is found it will easy to obtain the solution to the original system. To maintain the positive definiteness of the resulting, we need to multiply on both the sides by a non-singular matrix. and this matrix defined as C^{-1}.

4 Method of Preconditioning[1, 2, 24]

Consider,

$$\tilde{A} = C^{-1}A(C^{-1})^t, \quad (59)$$

with $\tilde{A}$ has a lower condition number than A. For the simplification, we use the matrix notation $C^{-t} \equiv (C^{-1})^t$.

Now, we will consider the conjugate applied to $\tilde{A}$.

Consider the linear system

$$\tilde{A}\tilde{x} = \tilde{b}, \quad (60)$$

where, $\tilde{x} = C^t x$ and $\tilde{b} = C^{-1}b$, then

$$\tilde{A}\tilde{x} = (C^{-1}AC^{-t})(C^t x) = C^{-1}Ax.$$

Thus, we could solve $\tilde{A}\tilde{x} = \tilde{b}$ for $\tilde{x}$ and then obtain x by multiplying by C^{-t}. Since,

$$\tilde{x}^{(k)} = C^t x^{(k)},$$

we have,

$$\tilde{r}^{(k)} = \tilde{b} - \tilde{A}\tilde{x}^{(k)} = C^{-1}r^{(k)}. \quad (61)$$

Let,

$$\tilde{v}^{(k)} = C^t v^{(k)},$$

and

$$w^{(k)} = C^{-1}r^{(k)},$$

then

$$\tilde{s}_k = \frac{\langle \tilde{r}^{(k)}, \tilde{r}^{(k)} \rangle}{\langle \tilde{r}^{(k-1)}, \tilde{r}^{(k-1)} \rangle} = \frac{\langle w^{(k)}, w^{(k)} \rangle}{\langle w^{(k-1)}, w^{(k-1)} \rangle}. \quad (62)$$

Also,

$$\tilde{t}_k = \frac{\langle \tilde{r}^{(k-1)}, \tilde{r}^{(k-1)} \rangle}{\langle \tilde{v}^{(k)}, \tilde{A}\tilde{v}^{(k)} \rangle} = \frac{\langle w^{(k-1)}, w^{(k-1)} \rangle}{\langle C^t v^{(k)}, C^{-1}Av^{(k)} \rangle}, \quad (63)$$

now,

$$\langle C^t v^{(k)}, C^{-1}Av^{(k)} \rangle = [C^t v^{(k)}]^t C^{-1}Av^{(k)} = \langle v^{(k)}, Av^{(k)} \rangle, \quad (64)$$

then we get,

$$\tilde{t}_k = \frac{\langle w^{(k-1)}, w^{(k-1)} \rangle}{\langle v^{(k)}, Av^{(k)} \rangle}. \quad (65)$$

Further,

$$\tilde{x}^{(k)} = \tilde{x}^{(k-1)} + \tilde{t}_k \tilde{v}^{(k)} = x^{(k-1)} + \tilde{t}_k v^{(k)}. \quad (66)$$

Now,

$$\tilde{r}^{(k)} = \tilde{r}^{(k-1)} - \tilde{t}_k \tilde{A}\tilde{v}^{(k)} = r^{(k-1)} - \tilde{t}_k A v^{(k)}. \quad (67)$$

Finally,

$$\tilde{v}^{(k+1)} = \tilde{r}^{(k)} + \tilde{s}_k \tilde{v}^{(k)},$$
$$C^t v^{(k+1)} = C^{-1}r^{(k)} + \tilde{s}_k C^t v^{(k)},$$
$$v^{(k+1)} = C^{-t}C^{-1}r^{(k)} + \tilde{s}_k v^{(k)},$$

$$v^{(k+1)} = C^{-t}w^{(k)} + \tilde{s}_k v^{(k)}. \tag{68}$$

The preconditioned conjugate gradient method depends on using the equations with the order (65), (66), (67), (62) and (68).

Example

$$44x_1 + 3x_2 = 24,$$
$$3x_1 + 4x_2 - x_3 = 30, \tag{69}$$
$$-x_2 + 4x_3 = -24.$$

has the exact solution $x^* = (3,4,-5)^t$. Use the conjugate gradient method with $x^{(0)} = (0,0,0)^t$ and no preconditioning, that is $C = C^{-1} = I$.

Solution: The system (69) can be written in the form of coefficient matrix,

$$A = [4\ 3\ 0\ 3\ 4\ -1\ 0\ -1\ 4\ ]. \tag{70}$$

$$r^{(0)} = b - Ax^{(0)},$$
$$= (24\ 30\ -24\ )^t - [4\ 3\ 0\ 3\ 4\ -1\ 0\ -1\ 4\ ][0\ 0\ 0\ ],$$
$$r^{(0)} = b = (24,30,-24\ )^t. \tag{71}$$

$$w_1 = C^{-1}r^{(0)},$$
$$= [1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ ][24\ 30\ -24\ ],$$
$$w_1 = (24,30,-24\ )^t. \tag{72}$$

$$v^{(1)} = C^{-t}w_1,$$
$$= [1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ ][24\ 30\ -24\ ],$$
$$v^{(1)} = (24,30,-24\ )^t. \tag{73}$$

$$\alpha_1 = \langle w_1, w_1 \rangle = 2052. \tag{74}$$

$$u_1 = Av^{(1)} = [4\ 3\ 0\ 3\ 4\ -1\ 0\ -1\ 4\ ][24\ 30\ -24\ ],$$
$$u_1 = (186,216,-126\ )^t. \tag{75}$$

$$\langle v^{(1)}, u_1 \rangle = (24\ 30\ -24\ )[186\ 216\ -126\ ],$$
$$\langle v^{(1)}, u_1 \rangle = 13968.$$

$$t_1 = \frac{\alpha_1}{\langle v^{(1)}, u_1 \rangle} = \frac{2052}{13968} = 0.146907216. \tag{76}$$

$$x^{(1)} = x^{(0)} + t_1 v^{(1)},$$
$$= (0,0,0\ )^t + (0.146907216)(24,30,-24\ )^t,$$
$$x^{(1)} = (3.525773184,4.40721648,-3.525773184\ )^t. \tag{77}$$

continuing like this, we get

$$x^{(2)} = (2.85801113,4.148971948,-4.954222161\ )^t. \tag{78}$$
$$x^{(3)} = (3.000285002,3.999700826,-5.000091911\ )^t. \tag{79}$$

This is the required solution for the given linear system.

As we use earlier, $C = C^{-1} = I$ as a preconditioning, now we use another preconditioning $D^{-\frac{1}{2}}$ to represent the diagonal matrix whose entries are the reciprocals of the square roots of the diagonal entries of the coefficient matrix. We are using this as preconditioning because the matrix A is positive definite, we expect the eigenvalues of $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ to be close to 1.

From the Table: 1.1, it is clear that approximate solutions are nearly the exact solution, thus rounding error does not significantly effect the results[15].

5 Conclusion

In this paper, we have discussed about Conjugate Gradient method using orthogonal vectors, for symmetric positive definite linear system of equations. Also method of Preconditioning has been discussed in detail. Comparison of these method is done based on the number of iterations and faster convergence. Thus we have observed that the preconditioned conjugate gradient method is more preferred when the system is large because it gives less number of iterations and converge faster. A numerical examples for each method are provided with the purpose to illustrate the simplicity of the method, Further numerical results are compared with the results generated with FOSS tool [28]Scilab.

Table: 1.1 Error analysis[14] of Conjugate Gradient Method using preconditioning

No. of iteration	x^*	$x^{(k)}$	Error
3	(3, 4, -5)	(3.000285002,3.999700826, -5.000091911)	(0.000285,0.000299, .000091)

References

[1] R.L. Burden, J.D. Faires, *Numerical analysis*, Cengage Learning, 2011.

[2] K. Bryan, *Conjugate gradient methods*.

[3] B.V. Limaye, *Functional Analysis*, New Age International Publishers, 2017.

[4] M.R. Hestenes , E. Stiefel, *Methods of conjugate gradient for solving linear systems*, J. Res. National Bureau stds. 49(6) (1952) 409-436.

[5] S.S. Sastry, *Introductory methods of numerical analysis*, PHI Learning Private Limited, 2013.

[6] M.K. Jain, S.R.K. Iyengar, R.K. Jain, *Numerical methods for scientific and engineering computation*, New Age International Publishers, 2012.

[7] G. Shankar Rao, *Numerical analysis*, New Age International Publishers, 2010.

[8] Rao V. Dukkipati, *Applied numerical methods using matlab*, New Age International Publishers, 2011.

[9] M.K.Jain, *Numerical solution of differential equations*, New Age International Publishers, 2014.

[10] D. Kincaid and W. Cheney, *Numerical analysis mathematics of scientific computing*, American Mathematical Society, 2010.

[11] R. Schoenberg, *Optimization with the quasi-newton method*, Aptech Systems, Inc. Maple Valley, WA. September 5, 2001.

[12] F. Scheid, *Theory and problems of numerical analysis*, McGraw-Hill Book Co., 1988.

[13] F.B. Hildebrand, *Introduction to numerical analysis*, Dover Publications, Inc., 2008.

[14] E. Balagurusamy, *Numerical methods*, Tata McGraw-Hill publishing comp. Ltd., 2009.

[15] K. Atkinson, *Elementary numerical analysis*, John Wiley & sons, 1994.

[16] V. Rajaraman, *Computer oriented numerical methods*, PHI Private Limited, 2000.

[17] G. Allaire, *Numerical Analysis and optimization*, Oxford Science Publications, 2007.

[18] Magnus R. Hestenes and Eduard Stiefel, *Methods of Conjugate Gradients for Solving Linear Systems*, Journal of Research of the National Bureau of Standards 49 (1952), 409–436.

[19] James W. Daniel, Convergence of the Conjugate Gradient Method with Computationally Convenient Modifications, Numerische Mathematik 10 (1967), 125–131.

[20] W. C. Davidon, *Variable Metric Method for Minimization*, Tech. Report ANL-5990, Argonne National Laboratory, Argonne, Illinois, 1959.

[21] R. Fletcher, M. J. D. Powell, *A Rapidly Convergent Descent Method for Minimization*, Computer Journal 6 (1963), 163–168.

[22] R. Fletcher, C. M. Reeves, *Function Minimization by Conjugate Gradients*, Computer Journal 7(1964), 149–154.

[23] Jean Charles Gilbert, Jorge Nocedal, *Global Convergence Properties of Conjugate Gradient Methods for Optimization*, SIAM Journal on Optimization 2 (1992), no. 1, 21–42.

[24] J. R. Shewchuk, *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*,August 4, 1994.

[25] Peter Gottschling, W. E. Nagel, *An Efficient Parallel Linear Solver with a Cascadic Conjugate Gradient Method:Experience with Reality*, Germany.

[26] O. Osadcha, Z. Marszaek, *Comparison of steepest descent method and conjugate gradient method*, Gliwice, Poland.

[27] R. H. Refsnæs, *A BRIEF INTRODUCTION TO THE CONJUGATE GRADIENT METHOD*,2009.

[28] Er.H. Ramachandran, Dr. A.S. Nair, *Scilab*, S. Chand & Comp. Ltd., 2012.