GLOBAL JOURNAL OF MATHEMATICAL SCIENCES VOL 8, NO. 2, 2009: 95-105

 ISSN 1596-6208

PRECONDITIONING THE MODIFIED CONJUGATE GRADIENT METHOD

D. E. A. OMOROGBE AND A. A. OSAGIEDE

(Received 4, February 2009; Revision Accepted 10, June 2009)

ABSTRACT

In this paper, the convergence analysis of the conventional conjugate Gradient method was reviewed. And the convergence analysis of the modified conjugate Gradient method was analysed with our extension on preconditioning the algorithm. Convergence of the algorithm is a function of the condition number of $M^{-1}A$. Again, this work broadens our understanding that the modified CGM yields the exacts result after n-iterations, and further proves that the CGM algorithm is quicker if there are duplicated eigenvalues. Given infinite floating point precision, the number of iterations required to compute an exact solution is at most the number of distinct eigenvalues. It was discovered that the modified CGM algorithm converges more quickly when eigenvalues are clustered together than when they are irregularly distributed between a given interval. The effectiveness of a preconditioner is determined by the condition number of the matrix and occasionally by its clustering of eigenvalues. For large scale application, CGM should always be used with a pre-conditioner to improve convergence.

KEYWORDS: Convergence, Conjugate Gradient, eigenvalue, preconditioning.

INTRODUCTION

Optimization theory is aimed at solving problem under investigation with a high degree of precision and under a highly restrictive operation time so as to minimize computing cost. It is necessary to choose a computational scheme that can meet these requirements (Otunta and Ibiejugba, 1991). The desire to construct a suitable and implementable algorithm has motivated the research investigation contained in this work. In this paper we seek to improve the convergence rate of the Modified Conjugate Gradient Method by preconditioning the algorithm.

The Conventional Conjugate Gradient method (CGM)

The conventional conjugate method (CGM) was originally developed by Hestenes and Stiefel (1952) as a method of solution for linear systems. Fletcher and Reeves (1964) built the necessary underlying theory for a successful application of the method to quadratic functional and developed its convergence properties.

To this end we defined quadratic functional as:

$$f(x) = f_0 + < a, x >_H + \frac{1}{2} < x, Ax >_H \quad (1)$$

Where A is an n x n symmetric positive definite operator on the Hilbert space H, and a is vector in

H. The steps in CGM algorithm are describe as follows (see Omolehin et al, 2006).

D. E. A. Omorogbe, Institute of Education, Ekehuan Campus, University of Benin, Benin City, Nigeria.

A. A. Osagiede, Department of Mathematics, Faculty of Physical Sciences, University of Benin, Benin City, Nigeria.

Algorithm 1

Step 1: The first element $X_0 \in H$ of the sequence is guessed, while the remaining members of the sequence are computed with the aid of step 2 to 4.

Step 2: $P_0 = -g_0 = -(a + Ax_0)$ (2)

where P_0 is the descent direction, g_o is the gradient of $f(x)\ and\ x = x_0$

Step 3: $x_{i+1} = x_i + \alpha_i P_i \qquad \alpha = \dfrac{< g_i, g_i >_H}{< P_i, AP_i >_H}$ (3)

α is the step length.

$$g_{i+1} = g_i + \alpha_i AP_i \quad (4)$$

$$P_{i+1} = g_{i+1} + \beta_i P_i \quad (5)$$

$$\beta_i = \frac{< g_{i+1}, g_{i+1} >_H}{< g_i, g_i >_H} \quad (6)$$

Step 4: If $g_i = 0$, for some í, terminate the sequence, else set í = í+1

We state the following theorem because it will give an understanding to the analysis of the convergence rate of the conventional conjugate gradient method (see Omolehin et al, 2006).

Theorem 1 (statement only): The convergence rate of GM algorithm for quadratic functional remains stable if □ = $^m/_M$ where m and M are the smallest and largest eigen values of the control operator A respectively. (See proof in Omolehin et al, 2006)

Convergence Rate of Conventional CGM Algorithm

To fully understand this work it will be necessary to show the convergence rate of the conventional CGM Algorithm (See Omolehin et al, 2006). Recall the quadratic functional

$$f(x) = f_0 + < a, x >_H + \frac{1}{2} < x, Ax >_H$$

where f_0 is constant, H is a Hilbert space, x is a n x n dimensional vector in H, a positive definite constant matrix operator.

Theorem 2: The law of convergence of the CGM algorithm is given as

$$E(x_n) = \left\{ \frac{1 - m/M}{1 + m/M} \right\}^{2n} E(x_0)$$

This establishes the convergence rate of the conventional CGM algorithm. (see proof in Omolehin et al, 2006).

The Modified Conjugate Gradient Method

In our previous work, Omorogbe and Osagiede (2008a) on the general convergence of the steepest descent method, the number of matrix-vector products per iteration can be reduced to one by using a recurrence to find the residual:

$$
\begin{aligned}
r_{i+1} &= -Ae_{i+1} \\
&= -A(e_i + \alpha_i d_i) \\
&= r_i - \alpha_i A d_i
\end{aligned} \tag{7}
$$

Here, the conjugate gradient is simply the method of conjugate direction where the search direction are constructed by conjugation of the residuals (i.e by setting $\mu_i = r_i$). The residual worked for steepest descent in our previous work Omorogbe and Osagiede (2008a), and it will even worked better for the conjugate gradient method. It has the property that it's orthogonal to the search direction

i.e $d_i r_j = 0$, í<j (by A- orthogonal of d-vectors) (8)

So, it's guaranteed always to produce a new, linearly independent search direction unless the residual is zero, in which case the problem is feasible.
As we shall see, there is an even better reason to choose, the residual.
Let us consider the implication of this choice, because the search vectors are built from the residuals and the subspace span $\{r_0, r_i, \ldots r_{i-1}\}$ is equal to D_i. As each residual is orthogonal to the previous search directions, it is also orthogonal to the previous residuals

$$r_i^T r_j = 0, \quad i \neq j \tag{9}$$

Interestingly, Eq(7) shows that each new residual r_i is just a linear combination of the previous residual and Ad_{i-1}, recalling that $d_{i-1} \in D_i$, this fact implies that each new subspace D_{i+1} is formed from the union of the previous subspace D_i and the subspace Ad_i. Hence,

$D_i = span\{d_0, Ad_0, A^2 d_0, \ldots, A^{i-1} d_0\}$
$= span\{r_0, Ar_0, A^2 r_0, \ldots, A^{i-1} r_0\}$

According to Shewchuk (1994), this supspace is called krylov subspace created by repeatedly applying a matrix to a vector. It has a fascinating property; because Ad_i is included in D_{i+1}, the fact that the preceding residual r_{i+1} is orthogonal to D_{i+1} by using Gram-Schmidt conjugation r_{i+1} is already A-orthogonal to all previous directions except d_i. The process of generating the set of A-orthogonal search directions $\{d_i\}$ is called conjugate Gram-Schmidt process (Gilbert and Nocedal, 1992). In the context of this paper, It follows that the Gram-Schmidt constant are:

$\beta_{ij} = - r_i^T Ad_j / d_j Ad_j$

Simplifying this expression and taking inner product of r_i and eq (7)

$r_i^T r_{j+1} = r_i^T r_j - \alpha_j r_i^T Ad_j$

$\alpha_j r_i^T Ad_j = r_i^T r_i - r_j^T r_{j+1}$

$$
r_i^T Ad_j = \begin{cases} \dfrac{r_i^T r_i}{\alpha_i} & i = j \\ \dfrac{-r_i^T r_i}{\alpha_{i-1}} & i \neq j \\ 0 & otherwise \end{cases} \qquad \text{by equation (9)}
$$

$\therefore \beta_{ij} = \begin{cases} \frac{1}{\alpha_{i-1}} \quad \frac{r_i^T r_i}{d_i^T{}_{-1} A d_{i-1}} & i \neq j+1 \quad \text{(using Gram Schmidt conjugation)} \\ 0 & \text{otherwise} \end{cases}$

Clearly, most of the β_{ij} term have disappeared. It is no longer necessary to store old search vectors to ensure the A-orthogonality of new search vectors. This major advance is what makes the modified conjugate gradient as important an algorithm as it is because both the iteration are reduced from $0(n^2)$ to 0(m), where mn is the number of zero entries of A (Gilbert and Nocedal, 1992). Henceforth, we shall use the abbreviation
$\beta_i = \beta_{i, i-1}$ *simplifying further.*

$$\beta_i = \frac{r_i^T r_i}{d^T_{i-1} r_{i-1}}$$

$$= \frac{r_i^T r_i}{r^T_{i-1} r_{i-1}}$$

Putting everything together, the modified conjugate gradient algorithm is given below

Algorithm 2

1. Start with $x=x_0$, otherwise set $x_0=0$

2. $d_0 = r_0 = b - Ax_0$

where. $\alpha_i = \frac{r_i^T r_i}{d^T_i A d_i}$

3. $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i d_i$

4. $r_{i+1} = r_i - \alpha_i A d_i$

where $\beta_{i+1} = \frac{r_{i+1}^T r_{i+1}}{r i^T r i}$ (*)

5. $d_{i+1} = r_{i+1} + \beta_{i+1} d_i$

6. When the algorithm reaches the minimum point, the residual becomes zero, and if (*) is evaluated on iteration later, a division by zero will result. Then, STOP.

The above algorithm of the modified CGM is clearly an improvement on the modified steepest descent method as well as algorithm 1 of the conventional conjugate method.

The performance of the modified conjugate gradient method is demonstrated in Fig 1.

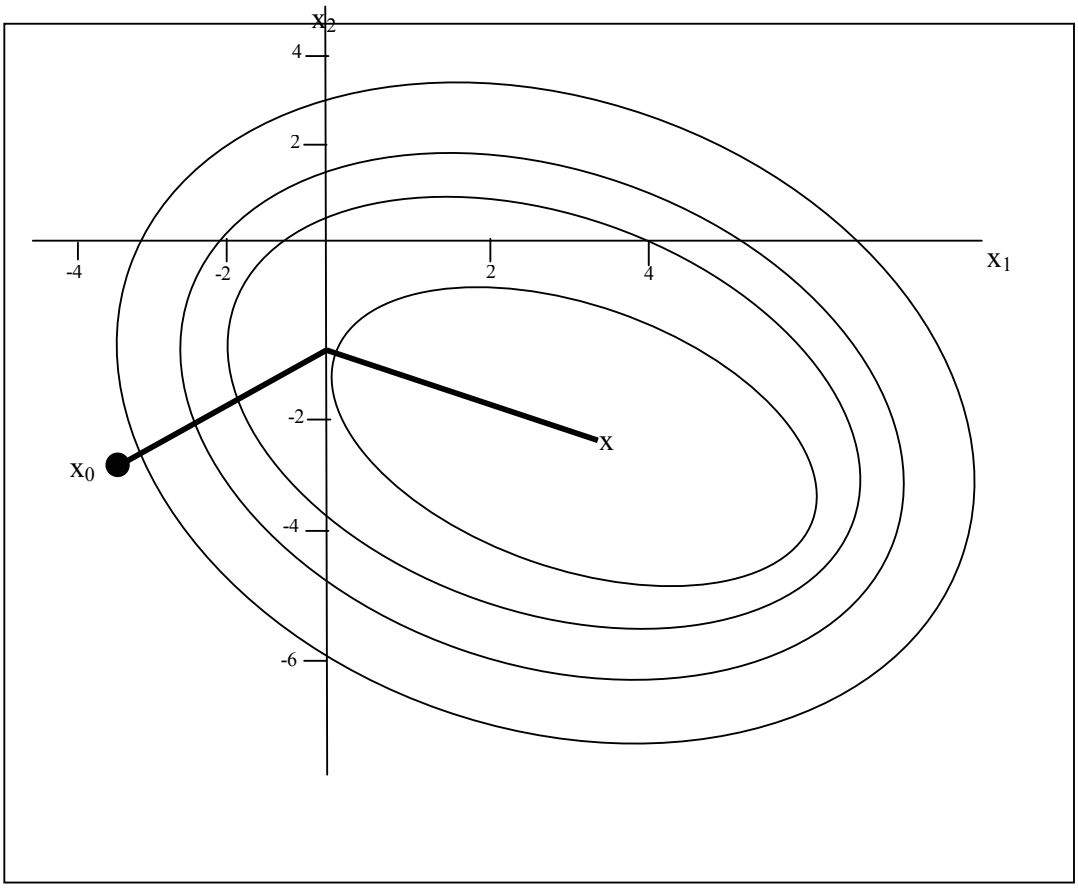


Fig 1: The modified conjugate gradient method.

Convergence Analysis of The Modified Conjugate Method.

Normally CGM is complete after n-Iterations. However in practice, accumulated floating point roundoff error courses the residual to gradually lose accuracy, and cancellation error causes the search vectors to lose A- orthogonality. This convergence analysis is important because the modified CGM algorithm is used for large class of problems that is not feasible to run even in n-iterations. The analysis is done using picking perfect polynomials (Omorogbe and Osagiede, 2008b).

Pick perfect polynomials

We have seen that, each step of the modified CGM algorithm, the value e_i is chosen from e_0 **+** D_i, where

$D_i = Span \{r_0, Ar_0, A^2r_0, \ldots\ldots A^{i-1}r_0\}$

$= Span \{Ae_{(0)}, A^2e_{(0)}, A^3e_0, \ldots\ldots, A^ie_0\}$

Using Krylov subspaces, for a fixed í, the error term has the form

$$e_i = \left[I + \sum_{j=1}^{i} \psi_i A_j \right] e_0$$

The coefficient α_j are related to the value β_i and β_{i-1}, but the precise relationship is that CGM algorithm closes the α_j coefficients that minimize $\|e_i\|_A$.

The expression in parentheses above can be expressed as a polynomial. Let $P_i(\square)$ be a polynomial of degree í, P_i can take either a scalar or a matrix as its argument, and will evaluate to the same; i.e

If $P_2(\square) = 2\square^2 + 1$, *then* $P_2(A) = 2A^2+1$. This feasible notation comes in handy such that

$$P_i(A)v - P_i(\square)v = 0$$

Then, we can express the error term as

$$e_i = P_i(A)e_0$$

If we require that $P_i(0) = 1$ the modified CGM chooses this polynomials when it chooses the $\square_j$ coefficients. Let's examine the effect of applying this polynomial to e_0

As in the analysis of the steepest descent in our earlier work Omorogbe and Osagiede (2008b), this expresses e_0 as a linear combination of orthogonal unit eigen vectors

$$e_0 = \sum_{j=1}^{n} \xi_j V_j$$

and we find that

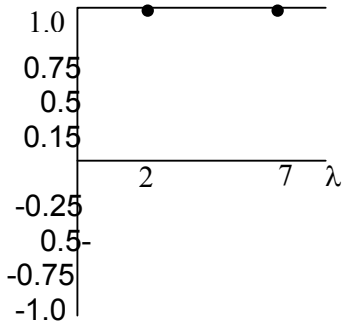
$$e_i = \square \xi_j P_i(\square_j) V_j$$

$$Ae_i = \square \xi_j P_i(\square_j)\, \square_j V_j \qquad \text{Implies}$$

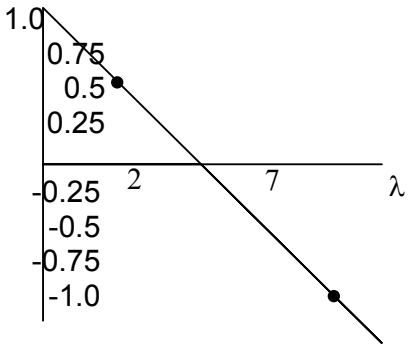
$$\|e_i\|^2_A - \square \xi_j \square_j^2 (P_i(\square_j))^2 \square_j$$

The performance of the modified CGM is illustrated in Figure 2 (a-c)

(a) $P_0(\lambda)$



(b) $P_1(\lambda)$



(c) $P_2(\square)$

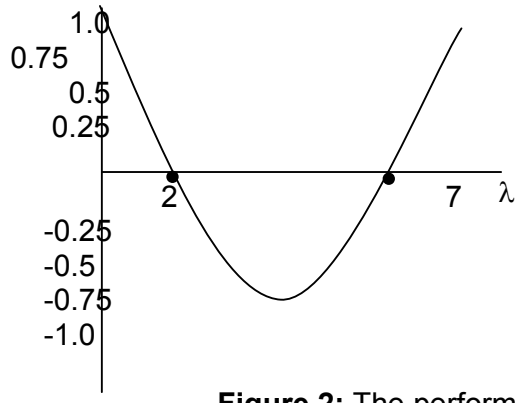


Figure 2: The performance of the modified CGM algorithm

From Figure 2 above, the convergence of the modified CGM after i-iterations depends on how close a polynomial P_i of degree í can be to zero on each eigenvalue, given the constraint that $P_i(0) = 1$. The CGM algorithm finds the polynomial that minimizes this expression, but convergence is only as good as the convergence of the least eigenvectors.
Letting E(A) be the set of eigenvalues of A. we have

$$\| e_i \|^2 A \square \; Min \max_{P_i, \square \epsilon E(A)} \{ Pí (\square) \}^2 \square \xi_j^2 \square_j$$

$$= Min \max_{P_i \square \square E(A)} \{ P_i(\square) \}^2 \| eo \|_A^2 \qquad (10)$$

Fig2 illustrated, for several values of i, the p_i that minimizes this expression from our illustration with eigen values 2 and 7. There is only one polynomial of degree zero that satisfied P_0 (0) =1, and that is $P_0(\square)$ =1, graphed into Fig2 (a). The optimal polynomial of degree one is $P_1(\square)$=1-2x/9 as graphed in Fig 2 (b). Note that $P_1(2)$ = 5/9 and $P_1(7)$ = - 5/9, and so the energy norm of the error term after one iteration of the CGM is no greater than 5/9 it initial value. Figure 2 (c) shows that, after two iterations, Equations (*) evaluates to zero. This is because the polynomial of degree two can be fit to the three points $P_2(0) = 1$, $P_2(2) = 0$ and $P_2(7) = 0$. In general, a polynomial of degree n can fit n+ 1 point, and thereby accommodate n separate eigen values.

The foregoing discussion reinforces our understanding that the modified CGM yields the exact result after n iterations; and further proves that the modified CGM is quicker if there are duplicated eigen values, given infinite floating-point precision, the number of iterations required to compute an exact solution is at most the number of distinct eigenvalues. We also find that modified CGM converges more quickly when eigenvalues are clustered together than when they are irregularly distributed between $\square_{min}$ and $\square_{max}$, because it is easier for the algorithm to choose a polynomial that makes equation (10) small.

Chebyshev Polynomials.

A useful approach is to minimize equation (10) over the range $[\square_{min}, \square_{max}]$ rather than at a finite number of points. The polynomials that accomplish this are based on Chebyshev polynomials (Gilbert and Nocedal, 1992).

The Chebyshev polynomial of degree i is $T_i(\square) = 1/2 [(\square + \sqrt{\square^2 - 1})^i + (\square - \sqrt{\square^2 - 1})^i]$ The Chebyshev polynomials have the property that $|T_i(\square)| \le 1$ on the domain $\square \; \square$ [-1,1] and further that $T_i(\square)$ is maximum on the domain $\square \; \square$ [-1,1] among all such polynomials (Gilbert and Nocedal, 1992). Equation (10) is minimized by choosing

$$Pi(\lambda) = \frac{T_i\left[\frac{\lambda_{\max} + \lambda_{\min} - 2\lambda}{\lambda_{\max} - \lambda_{\min}}\right]}{T_i\left[\frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}}\right]}$$

This polynomial has the oscillating properties of Chebyshev polynomials with the domain $\square_{min} \le \square \le \square_{max}$. The denominator enforces our requirement that $P_i(0) = 1$. The numerator has a maximum value on the interval between $\square_{min}$ and $\square_{max}$ so, from equation (10) we have,

$$\| e_i \|_A < T_i\left[\frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}}\right]^{-1} \| e_0 \|_A$$

$$= T_i \left[\frac{K+1}{K-1} \right]^{-1} \| e_0 \|_A$$

$$= 2 \left[\left[\frac{\sqrt{k}+1}{\sqrt{k}-1} \right]^i + \left[\frac{\sqrt{k}-1}{\sqrt{k}+1} \right]^i \right]^{-1} \| e_0 \|_A \quad (11)$$

The second addend inside the square brackets converges to zero as í increases, so it is common to express the convergence of CGM with the weaker inequality

$$\| e_i \|_A < 2 \left[\frac{\sqrt{k}-1}{\sqrt{k}+1} \right]^i \| e_0 \|_A \quad (12)$$

The first step of the modified CGM is identical to a step on the steepest descent method. Setting í = 1 in equation (11), we obtain the convergence result for the steepest descent method of our earlier work (Omorogbe and Osagiede, 2008b):

$$i.e. \| e_i \|_A < \left[\frac{k-1}{k+1} \right]^i \| e_0 \| \quad (13)$$

This is just the polynomial case illustrated in Figure 2(b). However in practice CGM usually converges faster than equation (12) would suggest, because of good eigenvalue distribution or good starting points. Comparing equation (12) of the modified CGM and equation (13) of the modified steepest descent method, it is clear that the convergence of the modified CGM is much quicker than that of modified steepest descent method as well as the conventional CGM algorithm. But it is not necessarily true that every iteration of CGM enjoys faster convergence, for example, the first equation of CGM is an iteration of steepest descent the factor 2 in equation (12) allows CGM a little slow for these poor iterations.

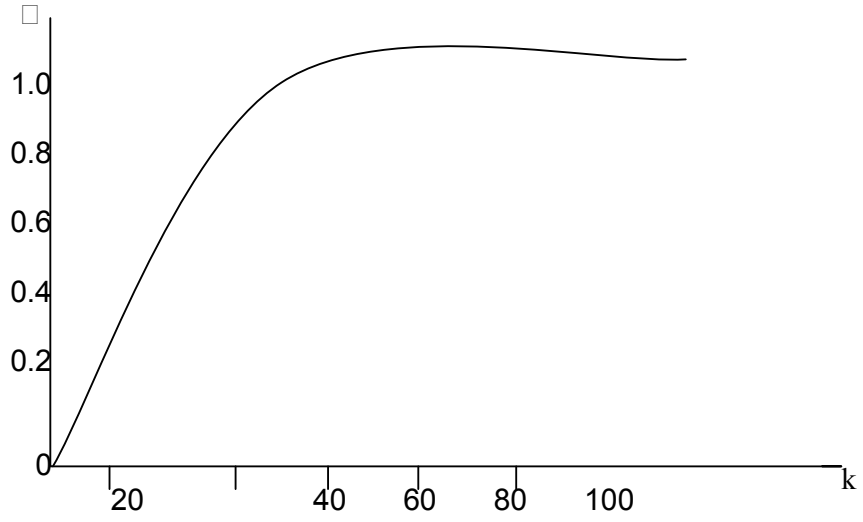


Fig. 3: Illustration of convergence of the modified CGM as a function of condition number.

Preconditioning the Modified Conjugate Gradient Method

Preconditioning is a technique for improving the condition number of a matrix. (Gilbert and Nocedal, 1992), Suppose that M is a symmetric matrix that approximates A. but is easier to invert.

We can solve

$$AX = b$$

indirectly by solving

$$M^{-1} AX = M^{-1}b \qquad (14)$$

If $k(M^{-1}A)$ □ $k(A)$, or if the eigenvalues of $M^{-1}A$ are better clustered than those of A, we can iteratively solve equation (14) more quickly that the original problem. $M^{-1}A$ is not generally symmetric nor definite, even if M and A are.

We can prevent this difficulty, because for every symmetric, positive – definite M there is matrix E that has the property that $EE^T = M$, (the matrix E can be obtained by Cholesky factorization). The matrices $M^{-1} A$ and $E^{-1} AE^{-T}$ have the same eigenvalues. This is true because if V is an eigenvector of $M^{-1}A$ with eigenvalue □, then E^TV is an eigenvector of $E^{-1} AE^{-T}$ with eigenvalue □: $(E^{-1}AE^{-T}) (E^TV) – (E^TE^{-T}) E^{-1} AV – E^TM^{-1}AV - □E^TV$.

The system $Ax = b$ can be transformed into the problem $E^{-1}\hat{A}E^{-T} X = E^{-1}b$. $\hat{T}hen\ X – E^{T}X$,
is solved first for x_0, then for x_i. Since $E^{-1}AE^T$ is symmetric and positive –definite, x can be found by steepest descent method or conjugate Gradient method (CGM). The system is called the transformed preconditioned conjugate Gradient method (Sluis and Vorst, 1986):

$$\hat{d}_0 = \hat{r}_0 = E^{-1}b – E^{-1}AE^{-1}\hat{x}_o$$

$$\alpha_i = \frac{\hat{r}_i^T r_i}{d_i^T E^{-1} AE^{-T} d_i}$$

$$x_{i+1} = \hat{x}_i + □_i\hat{d}_i$$

$$r_{i+l} = \hat{r}_i – r_i - □_i E^{-1}AE^{-T}\hat{d}_i$$

$$\beta_{i+1} = \frac{\hat{r}_{i+1}^T r_{i+1}}{\hat{r}_i^T \hat{r}_i}$$

$$\hat{d}_{i+1} = \hat{r}_{i+l} + □_{l+1} d_i$$

This method has the undesirable characteristics that E must be computed. However, a few careful variable substitution can eliminate E. Set $r_i = E^{-1} r_i = E^T d_i$, and use the identities

$$X_i = E^T x_i \text{ and } E^{-T} E^{-1} = M^{-1}.$$

We derive the untransformed preconditioned conjugate Gradient method:

$$r_0 – b - Ax_0,$$

$$d_0 = M^{-1} r_o,$$

$$\alpha_i = \frac{r_i^T M^{-1} r_i}{d_i^T Ad_i}$$

$$x_{i+1} = x_i + \alpha_i d_i$$

$$\beta_{i+1} = \frac{r^T M^{-1} r_{i+1}}{r^T M^{-1} r_i}$$

$d_{i+1} M^{-1} r_{i+1} + \square_{i+1} + d_i$

The matrix *E* does not appear in these equations; only M^{-1} is needed. By the same means, it is possible to derive a preconditioned steepest Descent method that does not use *E*.

The effectiveness of a preconditioned *M* is determined by the condition number of $M^{-1} A$, and occasionally by its clustering of eigenvalues. The problem remains of finding a pre conditioner that approximate *A* well enough to improve convergence enough to make up for the cost of computing the product $M^{-1} r_i$ once per iteration. It is not necessary to explicitly compute *M or* M^{-1}; it is only necessary to be able to compute the effect of applying M^{-1} to a vector. Within this constraint there is surprisingly rich supply of possibilities.

Intuitively, pre-conditioning is an attempt to stretch the quadratic form to make it appear more spherical, so that the eigenvalues are close to each other. A perfect pre conditioner is $M = A$; for this pre conditioner, $M^{-1}A$ has a condition number of one, and the quadratic form is perfectly spherical, so solution takes only one iteration. Unfortunately, the preconditioning step is solving the system $MX = b$, but this is not a very expedient pre conditioner.

The simplest pre conditioner is a diagonal matrix whose diagonal entries are identical to those of *A* . the process of applying this pre conditioner is refers to as diagonal preconditioning or Jacobi preconditioning (Gilbert and Nocedal, 1992). This is equivalent to scaling the quadratic form along the coordinate axes. By comparison the perfect pre conditioner $M = A$ scales the quadratic form along its eigenvector axes. A diagonal matrix is trivial to invert, when it is clear that the condition number has improved. This improvement is much more beneficial for systems where n □ 2.

A more elaborate pre conditioner is incomplete Cholesky preconditioning (Gilbert and Nocedal, 1992). Cholesky factorization is a technique for factorizing a matrix *A* into the form LL^T, where *L* is a triangular matrix. Incomplete Cholesky factorization is a variant in which little or no fill is allowed; A is approximated by the product LL^T, where L might be restricted to have the same pattern of nonzero elements as A; other element of L are thrown away. To use LL^T as a pre conditioner, the solution to L L □ = z is computed by back substitution (the inverse of LL^T is never explicitly computed). Unfortunately, incomplete Cholesky preconditioning is not always stable (Gilbert and Nocedal, 1992).

7. The Modified Conjugate Gradients on the Normal Equations

The modified CGM can be used to solve system where A is not symmetric, not positive – definite, and even not square. A solution to the least squares problem

$$\min_x \left|\left| AX - b \right|\right|^2 \qquad (15)$$

can be found by setting the derivative of the expression (15) to zero:

$$A^T Ax = A^T b. \qquad (16)$$

If A is square and nonsingular, the solution to equation (16) is the solution to Ax = b. If A is not square, and Ax = b is over constrained, that is, has more linear independent equations than variables, then there may or may not be solution to Ax = b, but it is always possible to find a value of x that minimizes expression (15), the sum of squares of the error of each linear equation (Omorogbe and Osagiede, 2008b).

$A^T A$ is symmetric and positive (for any *x,* $x^T A^T Ax = ||Ax|| ^2 \geq 0$). If $Ax = b$ is a constrained, then $A^T A$ is nonsingular, and methods like Steepest Descent and CGM can be used to solve equation (16). The only problem in doing so is that the condition number of $A^T A$ is the square of that of *A*. So convergence is significantly slower (Omorogbe and Osagiede, 2008b).

An important technical point is that the matrix A^TA is never formed explicitly, be cause it is less sparse than A. Instead A^TA is multiplied by d, first we find the product Ad, and then A^TAd. Also, numerical stability is improved if the value d^TA^TAd in (2) of Algorithm 2 is computed by taking the inner product of Ad with itself.

CONCLUSION

The effectiveness of a pre conditioner M is determined by the condition number of M^{-1} A, and occasionally by its clustering of eigenvalues (Omorogbe and Osagiede, 2008b). The problem remains of finding a pre conditioner that approximates A well enough to improve convergence of the CGM to make up for the cost of computing the product $M^{-1}r_i$ once per iteration (Omorogbe and Osagiede, 2008b). It is necessary to explicitly compute M or M^{-1}; it is only necessary to be able to compute the effect of applying M^{-1} to a vector (Omorogbe and Osagiede, 2008a). Within this constraint, there is a surprisingly rich supply of possibilities as earlier discussed (Omorogbe and Osagiede, 2008b). However, it is concluded that for large – scale application, CGM should always be used with a pre conditioner to improve convergence.

REFERENCES

Fletcher, R. and Reeves, C. M., 1964. Functional Minimization by Conjugate Gradients. The Computer Journal 7.p. 149-154.

Gilbert, J.C. and Nocedal., 1992. Global Convergence Properties of Conjugate Gradient Methods for Optimization, SIAM Journal 2, (1): p. 21-42.

Hestenes, M. R. and Stiefel, C.R., 1952. Methods of Conjugate Gradients for Solving Linear System Journal of Res. Of Nat. Bur. Of Standard Section R. 49: 409-436.

Omolehin J.O., Rauf, K., Opawoya, B. and Yahya, W.B., 2006. Jacobian Approach to Optimal Determination of Perturbation Parameter for Gradient Method, International Conference on New Trends in the Mathematical and Computer Sciences with Applications to Real World Problems at Covenant University, Ota, Nigeria. P. 407-418.

Omorogbe, D.E.A and Osagiede, A.A., 2008a. General Convergence Analysis of a Modified Steepest Descent Method. The Nigerian Association of Mathematical Physics Journal 12: 353-358.

Omorogbe D.E.A and Osagiede A.A., 2008b. Convergence Analysis of the Modified Conjugate Gradient method". The Nigerian Association of Mathematical Physics Journal 12, P. 359-368.

Otunta F.O., and Ibiejugba M.A., 1991. On Quadratic Cost Problem for Evolution Equation, Abacus, Journal of Mathematical Association of Nigeria, 2: p. 107-120

Shewchuk, J.R., 1994. An Introduction to Conjugate Gradient Method without Agonizing pain Carnegie Mellon University Pittsburgh.

Van der Sluis A., and Van der Vorst., 1986. The Rule of Convergence of Conjugate Gradients. Numerishe Mathematik 48, (5): p. 543-560.