Journal of the Nigerian Association of Mathematical Physics
***Volume* 12 (May, 2008), 359 - 368**
© J. of NAMP

Convergence analysis of a modified conjugate gradient method

[1]Dickson E. A. Omorogbe and [2]A. A.Osagiede
[1]Institute of Education, University of Benin,
Benin City, Nigeria.
[2]Department of Mathematics
University of Benin, Benin City, Nigeria.

Abstract

In this paper, the convergence analysis of the modified conjugate gradient method was thoroughly examined. The pick perfect polynomial and the energy norms were employed in the analysis. The convergence of the modified C G M after i iterations depends on how close a polynomial P_i of degree i can be to zero on each eigenvector given the constraint $P_i(0)$-1. The modified C G M algorithm finds the polynomial that minimizes the expression, but convergence is as good as the convergence of the least eigenvector. Also convergence of the algorithm is a function of the condition number. Again, this paper reinforces our understanding that the modified C G M yields the exact result after n-iterations, and further proves that the C G M 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. Another important finding from this work is that the modified C G M algorithm converges more quickly when eigenvalues are clustered together than when there are irregularly distributed between a given interval. It is clear that the modified C G M converges greater than the modified steepest descent method (S D M) of our earlier work, as well as the conventional C G M, in many cases. Comparing S D M with the C G M algorithm, it was concluded that both algorithm have different time complexities for n- dimensional problems. It is obvious that these results are new and clear departure from the analysis of modified steepest descent method; since the modified CGM is an improvement on the earlier mentioned method.

***Keywords*:** convergence, conjugate gradient, eigenvalue.

1.0 Introduction

Optimization is a scientific approach to obtaining the best decision in any set of circumstances. Hence it is the act of obtaining the best result under any given circumstances. The main objective of optimization is to solve a 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 this computational requirement. The desire to construct a suitable and highly implementable algorithm has motivated the research investigations contained in this work. As a result this paper seek an improvement on our earlier work on the modified steepest descent method.

2.0 Conventional conjugate gradient method (CGM)

The conventional conjugate method (CGM) was originally developed by Hestenes and Stiefel (1952 [3]) as a method of solution for linear systems. Fletcher and Reeves (1964 [1]) built the necessary

[1]e-mail: dickomorogbe@yahoo.com

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:

$$f(x) = f_0 + < a, x >_H + \tfrac{1}{2} < x, Ax >_H \tag{2.1}$$

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

2.1 Algorithm

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

$$P_0 = -g_0 = -(a + Ax_0) \tag{2.2}$$

where P_0 is the descent direction, g_0 is the gradient of $f(x)$ and $x = x_0$

Step 3:

$$x_{i+1} = x_i + \alpha_i p_i, \alpha = \frac{< g_i, g_i >_H}{< P_i}, AP_i >_H \tag{2.3}$$

α is the step length.

$$g_{i+1} = g_{ii} + \alpha_i AP_i \tag{2.4}$$

$$P_{i+1} = g_{i+1} + \beta_i P_i \tag{2.5}$$

$$\beta_i = \frac{< g_{i+1}, g_{i+i} >_H}{< g_i, g_i >_H} \tag{2.6}$$

Step 4:

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

We state the following theorem because it will give an understanding to the analysis of the convergence rate of the conventional conjugate gradient method by Ibiejugba et al (1999 [4]).

***Theorem* 2.1** (statement only)

The convergence rate of GM algorithm for quadratic functional remains stable if $\lambda = m/M$ *where m and M are the smallest and largest eigen values of the control operator A respectively*. See proof in Omolehin et al (2006 [5])

3.0 Convergence rate of conventional CGM. algorithm

To fully understand this work it will be necessary to show the convergence rate of the conventional C G M Algorithm by Ibiejugba (1999 [4]). Recall the quadratic functional

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

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* 3.1**

The law of convergence of the C G M algorithm is given as

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

where *m* and *M* are the smallest and largest eigen values of *A* respectively.

Proof:

Define $$E(x) = \tfrac{1}{2} < (x - x^*),\ A(x - x^*) >_H \tag{3.1}$$

Therefore $$E(x) = \tfrac{1}{2} < (x - x^*),\ A(x - x^*) >_H = \tfrac{1}{2} < x + A^{-1}a,\ A(x + A^{-1}a) >_H \tag{3.2}$$

$$= \tfrac{1}{2} < x + A^{-1}a,\ Ax + AA^{-1}a >_H \tag{3.3}$$

$$= \tfrac{1}{2} < x + A^{-1}a,\ Ax + a >_H \tag{3.4}$$

$$= + \pi\ A^{-1}a, Ax >_H + \tfrac{1}{2}\ \pi\ A^{-1}a, a >_H \tag{3.5}$$

$$= \tfrac{1}{2} < x + Ax >_H + \tfrac{1}{2} < x,\ a >_H$$

$$= + \frac{1}{2}\ \pi\ A^{-1}a, Ax >_H + \tfrac{1}{2}\ \pi - x^*, a >_H \tag{3.6}$$

$$= \frac{1}{2}\ \pi\ x, Ax\ \phi_H + \frac{1}{2} < x, a\ \phi_H + \frac{1}{2}\ \pi\ x, Ax^*\ \phi_H + \frac{1}{2}\ \pi\ A^{-1}a, Ax >_H \tag{3.7}$$

$$E(x) = f(x) - f_0 + \tfrac{1}{2} < x^*,\ Ax^* > = f(x) - f_{(0)} \tag{3.8}$$

Therefore, $E(x)$ is $f(x)$ plus a constant term, hence the convergence of $E(x)$ is considered instead of that of $f(x)$ as from now.

Recall that

$$E(x) = \tfrac{1}{2} < x + A^{-1}a,\ Ax + a >_H = \tfrac{1}{2} < A^{-1}(Ax + a), Ax + a >_H$$

$$= \tfrac{1}{2} < A^{-1}g(x),\ g(x) >_H \tag{3.9}$$

Hence $$E(x_I) - E(x_{i+1}) = \tfrac{1}{2} < x_i - x^*, A(x_i - x^*) >_H - \tfrac{1}{2} < x_{i+1} - x^*,\ A(x_{i+1-x^*}) \tag{3.10}$$

But $X_{i+1} = x_i + \alpha_i P_i$. Therefore,

$$E(x_I) - E(x_{i+1}) = \tfrac{1}{2} < x_i - x^*, A(x_i - x^*) >_H - \tfrac{1}{2} < x_i + x_i + x_i p_i - x^*,\ A(x_i + \alpha_i p_i - x^*) >_H \tag{3.11}$$

$$= \tfrac{1}{2} < x_i + x_i p_i - x^*,\ A(x_i + \alpha_i p_i - x^*) >_H$$

$$= \tfrac{1}{2} < x_i - x^*, A(x_i - x^*) >_H - \tfrac{1}{2} < x_i - x^*,\ A(x_i - x^*) >_H$$

$$- \tfrac{1}{2}\alpha_i < p_i,\ A(x_i + \alpha_i p_i - x^*) >_H - \tfrac{1}{2}\alpha_i < x_i - x^*, Ap_i >_H \tag{3.12}$$

$$= -\underline{\alpha_i} < p_i,\ A(x_i - x^*) >_H - 1/2\,\alpha_i < x_i - x_i, Ap_i >_H - 1/2\,\alpha_i < p_i,\ \alpha_0 p_0 >_H \tag{3.13}$$

$$= -\alpha_i < p_i,\ Ax_i + a >_H - \tfrac{1}{2}\alpha_i^2 < p_i,\ Ap_i >_H - \tfrac{1}{2}\alpha_i^2 < p_i,\ Ap_i >_H$$

$$= -\alpha_i < p_i g_i >_H - \tfrac{1}{2}\alpha_i < g_i, g_i > \tag{3.14}$$

Since $$\alpha_i = \frac{< g_i\, g_i >_H}{< p_i,\ Ap_i >_H} \tag{3.15}$$

$$= -\alpha_i < -g_i \beta_i -, P_{i-1}, >_H - \tfrac{1}{2}\alpha_i < g_i, g_i >_H$$

$$= -\alpha_i < g_i, g_i > - \tfrac{1}{2} < g_i >_H \text{, (since } < p_i - g_i >_{H=0} \text{ it orthogonality of } p_i \text{ and } g_i \text{)}$$

$$= \alpha_i < g_i, g_i >_H - \tfrac{1}{2}\alpha_i < g_i, g_i >_H = \tfrac{1}{2}\alpha_i < g_i, g_i >_H = \frac{\tfrac{1}{2}\alpha_i < g_i, g_i >_H^2}{< p_i, Ap_i >_{H+\_}}$$

because

$$\alpha_i = \frac{< g_i, g_i >_H}{< p_i, Ap_i} \tag{3.16}$$

Hence,
$$E(x_i) - E(x_{i+1}) = \frac{< g_i, g_i >_H \; E(x_i)}{< p_i, \; Ap_i >_H \; g_i A^{-1} g_i \; \phi_H} \tag{3.17}$$

Using the fact that $g_i = \beta_{i-1} p_{i-1} - p_i$, we have

$$< g_i, Ag_i >_H = < \beta_{i-1} p_{i-1}, \; A(\beta_{i-1} p_{i-1} - p_i) \; \phi_H = \beta_{i-1} < p_{i-1}, \; Ap_{i-1} >_H + < p_i, Ap_i >_H$$
$$\geq < p_{i-1}, \; Ap_{i-1} >_H, \sin ce < p_{i-1}, Ap_{i-1} >_H \geq 0 \tag{3.18}$$

(due to the positive definiteness of operator *A*), $< g_i, \; Ag_i >_H \geq < p_i, \; Ap_i >_H$. Therefore

$$E(x_i) - E(x_{i+1}) \geq \frac{< g_i, g_i >^2_H \; E(x_i)}{< g, Ag_i >_H < g_i, A^{-1} g_i >_H} \tag{3.19}$$

But for a bounded self adjoint operator in a Hilbert space *H*, Kantorovich established the following inequality

$$\frac{< x, x >_H^2}{< x, \; Ax >_H < x, A^{-1} >_H} \geq \frac{4mM}{(m+M)^2} \tag{3.20}$$

where *m* and *M* are respectively the greatest lower and least upper bounds of the spectrum of operator *A*. Using Kantorovich's inequality we obtain

$$E(x_{i+1}) \leq \left\{ \frac{1 - m/M}{1 + m/M} \right\}^2 E(x_0) \tag{3.21}$$

This establishes the convergence rate of the conventional C G M algorithm in this case A is a matrix operator, where *m* and *M* are the smallest and greatest eigen values of A respectively

4.0 The modified conjugate gradient method

In our previous work 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:

$$r_{i+1} = -Ae_{i+1} = -A(e_i + \alpha_i d_i) = r_i - \alpha_i A d_i \tag{4.1}$$

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 (2008 [7]), 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.

$$d_i r_j = 0, \; i < j \text{ (by A- orthogonal of d-vectors)} \tag{4.2}$$

So, it's guaranteed always to produce a new, linearly independent search direction unless the residual is zero, in which case the problem is always solved. 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 $\{r_0, r_i \text{K}\ r_{i-1}\}$ is equal to D_i. As each residual is orthogonal to the previous search directions, it is also orthogonal to the previous residuals

$$r_i^T r_j = 0 \quad i \neq j \tag{4.3}$$

Interestingly, Equation (4.1) 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**,**

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

According to Shewchuk (1994 [[8]), 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. 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 equation (4.1)

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

$$r_i^T Ad_j = \begin{cases} \dfrac{r_i^T r_i}{\alpha_i}, i = j \\ \dfrac{-r_i^T r_i}{\alpha_{i-1}}, i \neq j \\ \\ 0,\ otherwise \end{cases}$$

by equation (4.3)

$$\therefore \qquad \beta_{ij} = \begin{cases} \dfrac{1}{\alpha_{i+1}} \dfrac{r_i^T r_i}{d_i^T\ Ad_{i-1}}, i \geq j+1 \\ 0,\ otherwise \end{cases}$$

(using Gram Schmidt conjugation)

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*. Henceforth, we shall use the abbreviation $\beta_i = \beta_{i,i-1}$ simplifying further. $\beta_i = \dfrac{r_i^T r_i}{d_{i-1}^T r_{i-1}} = \dfrac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$. Putting everything together, the modified conjugate gradient algorithm is given below

4.1 Algorithm

1. $d_0 = r_0 = b - Ax_0$, where. $\alpha_i = \dfrac{r_i^T r_i}{d_i^T A d_i}$

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

3. $r_{i+1} = r_i - \alpha_i \ Ad_i$, where $\beta_{i+1} = \dfrac{r_{i+1} T r_{i+1}}{r_i^T r_i}$ (*)

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

4.2 Remark

Starting: if you have a rough estimate of the value of x, use it as the starting value x_0, otherwise set $x_0 = 0$.

Stopping: when the CGM algorithm reaches the minimum point, the residual becomes zero, and if (*) in the algorithm 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. The performance of the modified conjugate gradient method is demonstrated in Figure 4.1. below.

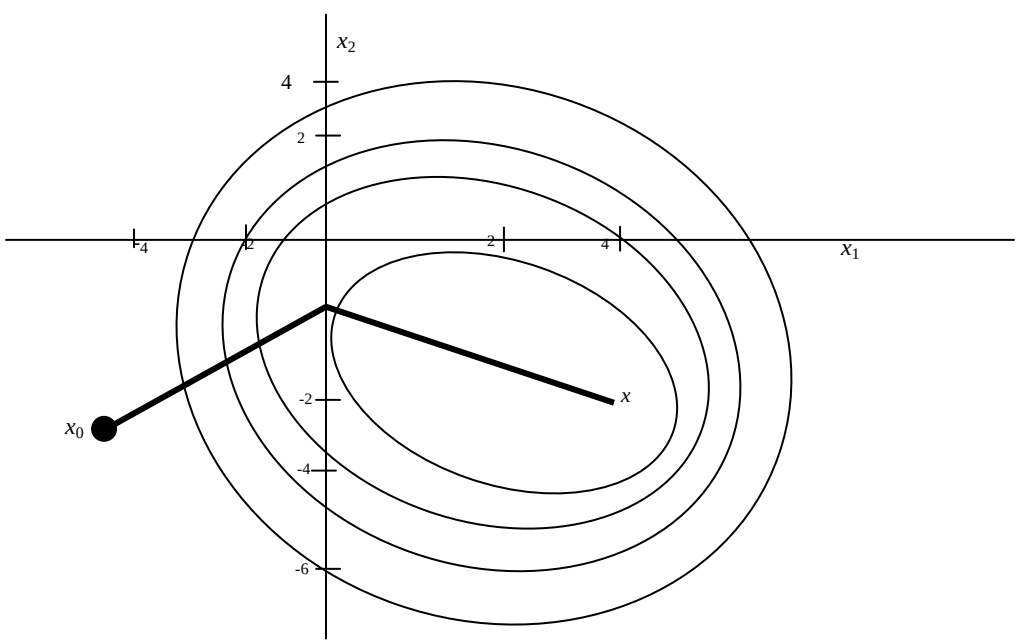


Figure 4.1: The modified conjugate gradient method.

5.0 Convergence analysis of the modified conjugate method.

Normally CGM is complete after n-Iterations, However in practice, accumulated floating point round off 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

5.1 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

$$D_i = span\,\{r_0,\, Ar_0,\, A^2 r_0, \ldots, A^{i-1} r_0\} = span\,\{Ae_{(0)}, A^2 e_{(0)}, A^3 e_0, \ldots, A^i e_0\}$$

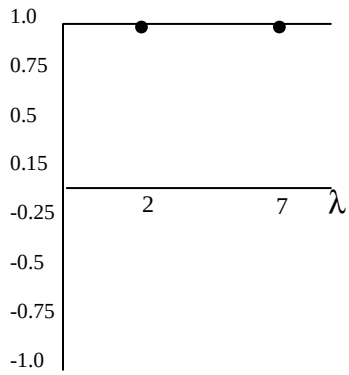
using Krylov subspaces, for a fixed í, the error term has the form $e_i = \left(1 + \sum_{j=1}^{i} \psi_j A^j\right) e_0$. The coefficient ψ_j are related to the value α_i and β_i, but the precise relationship is that C G M algorithm closes the ψ_j coefficients that minimize $\|e_i\|_A$. The expression in parentheses above can be expressed as a polynomial. Let $P_i(\lambda)$ be a polynomial of degree i, P_i can take either a scalar or a matrix as its argument, and will evaluate to the same; that is, if $P_2(\lambda) - 2\lambda^2 + 1$, then $P_2(A) = 2A^2 + 1$.. This feasible notation comes in handy such that $P_i(A)v - P_i(\lambda)v = 0$. 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 ψ_j coefficients. Let's examine the effect of applying this polynomial to e_0 . As in the analysis of the

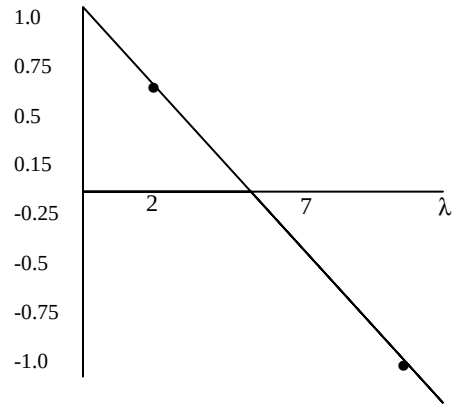
steepest descent in our earlier work Omorogbe and Osagiede (2008), This expresses e_0 as a linear combination of orthogonal unit eigen vectors $e_0 = \sum_{j=1}^{n} \xi_j V_j$ and we find that $e_i = \sum \xi_j P_i(\lambda_j) V_j$,

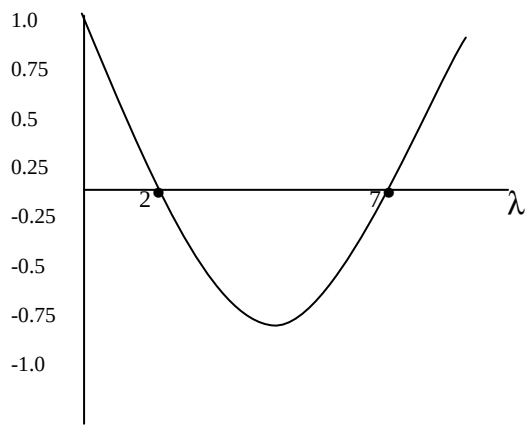
$Ae_i = \sum \xi_j P_i(\lambda_j) V_j$ implies $\|e_i\|_A^2 - \sum \xi_j \lambda_j^2 (P_i(\lambda_j))^2 \lambda_j$. The performance of the modified CGM is illustrated in figure 5.1 (a-c) below:

(a) $P_0(\lambda)$



(b) $P_1(\lambda)$





(c) $\boldsymbol{P_2(\lambda)}$

Figure 5.1: The performance of the modified C G M algorithm

From figure 5.1 above, the convergence of the modified CGM after i-iterations depends on how close a polynomial P_i of degree i 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

$$\|e_i\|^2 A \leq \min_{P_i,\, \lambda \in E(A)} \max \left(P_i(\lambda)\right)^2 \sum \xi_j^2 \lambda_j = \min_{P_i,\, \lambda \in E(A)} \max \left(P_i(\lambda)\right)^2 \|e_0\|_A^2 \qquad (5.1)$$

Figure 5.1 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 $P_0(\lambda)=1$, graphed into Figure 5.1(a). The optimal polynomial of degree one is $P_1(\lambda)-1-2x/9$ as graphed in Figure 5.1b. 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 C G M is no greater than 5/9 it initial value. Figure 5.1 (c) shows that, after two iterations, Equations (*) evaluates to zero. This is because of polynomial of degree two can be fit to 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 points, and thereby accommodate n separate eigen values.

The foregoing discussion reinforces our understanding that the modified C G M yields the exact result after n iterations; and further proves that the modified C G M 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 C G M converges more quickly when eigenvalues are clustered together than when they are irregularly distributed between λ_{min} and λ_{max}, because it is easier for the algorithm to choose a polynomial that makes equation (3.1) small.

5.2 Chebyshev polynomials.

A useful approach is to minimize equation (5.1) over the range [λ_{min}, λ_{max}] rather than at a finite number of points. The polynomials that accomplish this are based on Chebyshev polynomials. The Chebyshev polynomial of degree i is $T_i(\omega)=1/2\left[(\omega+\sqrt{\omega^2}-1)^i+(\omega-\sqrt{\omega^2}-1)^i\right]$. The Chebyshev polynomials have the property that $|T_i(\omega)|\leq 1|$ on the domain $\omega\in[-1,1]$ and further that $T_i(\omega)$ is

maximum on the domain $\omega \in [-1, 1]$ among all such polynomials. Equation (5.1) is minimized by choosing

$$P_i(\lambda) = \frac{T_i\left(\dfrac{\lambda_{\max} + \lambda_{\min} - 2\lambda}{\lambda_{\max} - \lambda_{\min}}\right)}{T_i\left(\dfrac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}}\right)}$$

This polynomial has the oscillating properties of Chebyshev polynomials with the domain $\lambda_{\min} \le \lambda \le \lambda_{\max}$. The denominator enforces our requirement that $P_i(0) = 1$. The numerator has a maximum value on the interval between $\lambda_{\min}$ and $\lambda_{\max}$ so, from equation (5.1) we have,

$$\|e_i\|_A < T_i\left(\frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}}\right)^{-1} \|e_0\|_A = T_i\left(\frac{K+1}{K-1}\right)^{-1} \|e_0\|_A$$

$$= 2\left[\left(\frac{\sqrt{k+1}}{k-1}\right)^i + \left(\frac{\sqrt{k-1}}{k+1}\right)^i\right]^{-1} \|e_0\|_A \qquad (5.2)$$

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

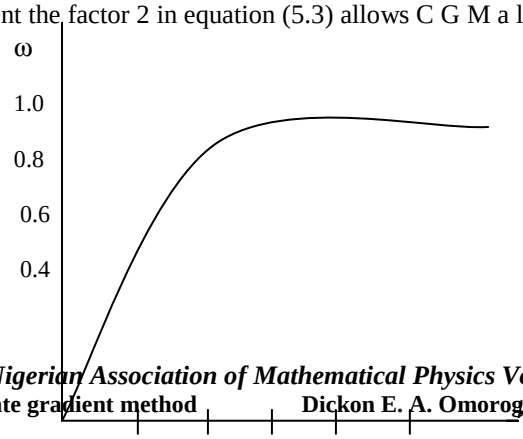
$$\|e_i\|_A < 2\left(\frac{\sqrt{k+1}}{k-1}\right)^i \|e_0\|_A \qquad (5.3)$$

The first step of the modified C G M is identical to a step on the steepest descent method setting í = 1 in equation (5.2), we obtain the convergence result for the steepest descent method of our earlier work

i.e.

$$\|e_i\|_A \le \left(\frac{K-1}{K+1}\right)^i \|e_0\| \qquad (5.4)$$

This is just the polynomial case illustrated in figure 5.1(b). However in practice C G M usually converges faster than equation (5.3) would suggest, because of good eigenvalue distribution or good starting points. Comparing equation (5.3) of the modified C G M and equation (5.4) of the modified steepest descent method, it is clear that the convergence of the modified C G M is much quicker than that of modified steepest descent method as well as the conventional C G M algorithm. But it is not necessarily true that every iteration of C G M enjoys faster convergence for example the first equation of C G M is an iteration of steepest descent the factor 2 in equation (5.3) allows C G M a little slow for these poor iterations.



0.2

0

20 40 60 80 100

Figure 5.2: Illustration of convergence of the modified C G M as a function of condition number.

6.0 Complexity, discussion and conclusion

In this work, the dominating separations during iteration of either the modified steepest descent of previous work or the modified C G M are matrix vector products. In general, matrix-vector multiplication requires 0C(m) operations, where m is the number of non-zero entries in matrix. For many problems, A is sparse and m $m \in 0(n)$

Suppose we wish to perform enough iterations to reduce the norm of the error by a factor of ε; that is $\|e_i\| \leq \in \|e_0\|$. Equation (5.4) can be used to show that maximum number of iterations required to achieve this bound, using steepest descent method is

$$i \leq \left(\tfrac{1}{2} k \ \ln \ (\tfrac{1}{e})\right)$$

Where equation (5.3) suggest that the maximum number of iterations the modified C G M requires is

$$i \leq \left(\frac{1}{2}\sqrt{k}\ln\left(\frac{2}{\in}\right)\right)$$

We conclude that the modified steepest descent method has a time complexity of $0(mk)$, whereas the modified C G M has a time complexity of $0\left(m\sqrt{k}\right)$. Both algorithms have a space complexity of 0(m). Finite difference and finite element approximations of second-order elliptic boundary value problems posed on d-dimensional domain often have $k \in 0(n^{2/d})$. Thus, the modified steepest descent has a time complexity of $0(n^2)$ for two-dimensional problems, versus $0(n^{3/2})$ for the modified C G M; and the modified steepest descent method has a time complexity of $0(n^{5/3})$ for three-dimensional problems, versus $0(n^{4/3})$ for the modified C G M. It however noted that convergence of the modified CGM algorithm converges quicker than the modified steepest descent method as well as the conventional CGM in many cases.

References

[1] Fletcher R and Reeves C. M (1964) "Functional Minimization by Conjugate Gradients" *The computer Journal* 7.p. 149-154.

[2] Hestenes, M. F. (1973), "Iterative Methods for Solving Linear Equations" *Journal of Optimization* Washington, D.C.

[3] 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.* Vol 49, p.409-436.

[4] Ibiejugba, M. A et al (1999) "Foundation Postgraduate Course on Computational Methods and Application in Optimization", *National Mathematical centre, Abuja,* Nigerian.

[5] Omolehin et al (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* P. 407-418.

[6] Omorogbe, Dickson A, Chukwendu, C. R, and Okonta, S. D. (2006) "A Supper Convergence of a Modified Non-Classical Conjugate Gradient Method" *Knowledge Review* Vol. 13, No. 7.p.100-106.

[7] Omorogbe D.E.A and Osagiede (2008) "General Convergence Analysis of a Modified Steepest Descent Method" *Journal of Mathematical Physics* (in press)

[8] Shewchuk, J.R (1994) "An Introduction to Conjugate Gradient Method Without Agonizing pain" Carnegie Mellon University Pittsburgh.