Journal of the Nigerian Mathematical Society
Vol. 36, Issue 2, pp. 355-367, 2017

A FOURTH-ORDER NONLINEAR CONJUGATE GRADIENT METHOD AND TRUST REGIONS IN UNCONSTRAINED OPTIMIZATION

E. NWAEZE[1] AND L. O. OMENYI

ABSTRACT. This paper presents a fourth-order nonlinear conjugate gradient method and trust regions in unconstrained optimization. The method has been designed to solve unconstrained optimization problems with high accuracy. It is based on a nonlinear polynomial approximation of the objective function. The idea is to approximate the minimizing function byTaylor series expansion using fourth-order terms. The algorithms are presented by steps and some properties of the gradients are proved, using classical results. Also, the convergence analysis has been proved under classical and known assumptions. The main algorithm generates adequate trust region radius iteratively and has a global convergence property. Numerical results are presented and compared to some existing results by Dolan Moore's performance profile. The analysis of results confirms that this method is accurate, since the computed results are very close to the exact solutions.

Keywords and phrases: Fourth-order conjugate gradient method, trust region, unconstrained optimization, objective function, nonlinear polynomial approximation, large scale optimization.
2010 Mathematical Subject Classification: 90C30, 65K05, 49M37

1. INTRODUCTION

The unconstrained minimization of a smooth function (f) in many variables remains an important problem in optimization theory. Various fields of science and engineering seek to solve this class of problems, in real life applications. The general approach is to find the zeros of the function gradient since the local minima occur at stationary points. In order to achieve fast global convergence, we develop and present a fourth-order nonlinear conjugate gradient method with trust region technique for solving unconstrained minimization problems. Consider the unconstrained optimization

Received by the editors December 09, 2016; Revised: June 25, 2017; Accepted: June 26, 2017
www.nigerianmathematicalsociety.org
[1]Corresponding author

problem

$$\min_{x\in\Re^n} f(x) \tag{1}$$

where$f$is a differentiable function. The process of minimizing a non-quadratic objective function through the conjugate gradients is called a nonlinear conjugate gradient method [1, 2]. Various types of conjugate gradient method have been used to solve unconstrained minimization problems [3]. Usually, a function F is constructed to approximate f. Many scholars have published their findings on this method [4, 5, 6]. If the objective function is not quadratic or the inexact line searches are used, some of the conjugate gradient methods fail to converge globally [7, 8]. In order to solve problem (1) effectively, we designed an algorithm that reduces the high storage and computation cost of some calculated matrices [9]. Some conventional algorithms on nonlinear conjugate gradient method are available [10, 11, 12, 13, 14]. Every conjugate gradient method is an iterative scheme of the form

$$x_{k+1} = x_k \ + \ \alpha_k d_k, \quad k = 0,\ 1,\ 2,\ ... \tag{2}$$

wherex_0 is an initial point, $\alpha_k$is a step size and the search direction is

$$d_k = -g_k \ if \ k = 0; \ \ d_k = -g_k + \beta_{k-1} d_{k-1} \ if \ k \geq 1 \tag{3}$$

$g_k = \nabla f(x_k)$and$\beta_k$specifies the choice of conjugate gradient method [15]. It can take any of the following forms.
$\beta^{FR} = \frac{||g_k||^2}{||g_{k-1}||^2}$ [4], $\beta^{PRP} = \frac{g_k^T(g_k - g_{k-1})}{||g_{k-1}||^2}$[16], $\beta^{HS} = \frac{g_k^T(g_k - g_{k-1})}{d_{k-1}^T(g_k - g_{k-1})}$[17, 18], $\beta^{CD} = -\frac{||g_k||^2}{d_{k-1}^T g_{k-1}}$[14], $\beta^{DY} = \frac{||g_k||^2}{d_{k-1}^T(g_k - g_{k-1})}$ [12] and $\beta^{LS} = -\frac{g_k^T(g_k - g_{k-1})}{d_{k-1}^T g_{k-1}}$ [19].
Many of these conjugate gradient methods use a line search technique or trust region method [20]. Others use line search and trust region approach [21]. Trust region methods seek to solve problem (1) within a region or disc where it is hoped that the solution resides. Yuan and Stoer [22] studied the conjugate gradient method on a subspace and obtained a new variant of the method. Motivated by their study, we construct a fourth-order nonlinear conjugate gradient method (FONCGM) and trust regions in unconstrained optimization. Section two contains the fourth-order nonlinear conjugate gradient method and trust region technique. Sections three and four discuss the convergence analysis and some

numerical experiments, respectively. Section five presents a discussion on the numerical results while section six ends this work with a conclusion.

2. 4^{th}ORDER NONLINEAR CONJUGATE GRADIENT METHOD

The fourth-order nonlinear conjugate gradient method is based on the Taylor series representation of$f$ by another function F.This representation is expected to be a better approximation of$f$ than the conventional approach. The following is the representation of Fat the point x_k.

$$F(x) = f(x_k) \ + \ df(x_k) \ + \ \frac{1}{2\,!}d^2f(x_k) \ + \frac{1}{3\,!}d^3f(x_k) \ + \frac{1}{4\,!}d^4f(x_k), \quad (4)$$

where$d^{\,n}f(x_k) = \sum_{i1}^{N}\sum_{i2}^{N}...\sum_{in}^{N} h_{i_1}h_{i_2}...h_{i_n}\frac{\partial^n f(x_k)}{\partial x_{i_1}\partial x_{i_2}...\partial x_{i_N}}$,$x, \ x_k, \ h_{i_j} \in \Re^N; \ h_{i_j} = x - x_k, \ \ 2 \leq n \leq 4.$

Using a vector $h = x - x_k$and $A_i = \nabla^i f(x_k)$, in equation (1), we have

$$F(x) = f(x_k) + h^T A_1 + \frac{1}{2\,!}h^T A_2 h \ + \frac{1}{3\,!}h^T\left(h^T A_3 h\right) + \frac{1}{4\,!}h^T\left(h^T A_4 h\right)h \quad (5)$$

$$= f(x_k) \ + \ h^T A_1 + \frac{1}{2\,!}h^T\left[A_2 \ + \ \left\{\frac{2}{3\,!}A_3 \ + \ \frac{2}{4\,!}A_4 h\right\}h\right]h.$$

Using tensor notations presented in [23, 24], we have

$$\left[\frac{2}{3\,!}A_3 \ + \ \frac{2}{4\,!}A_4 h\right]h = \left[\sum_{j=3}^{4}\frac{2}{j\,!}A_j\prod_{p=1}^{j-3}h^{Z_p}\right]h \quad (6)$$

$$= \left[\sum_{j=3}^{4}\frac{2}{j\,!}\sum_{m=0}^{j-3}(-1)^m\left(\begin{array}{c} j-3 \\ m \end{array}\right)g\left(x_k \ + \{(j-3) \ - m\}\,h\right)\right]^T h$$

$$= \frac{1}{12}\ \left[g(x) \ + \ 3g(x_k)\right]^T h$$

where$g(x)$ denotes the gradient of f, at point x, $\prod_{p=1}^{2}h^{Z_p} = h^T h$, $\left(\begin{array}{c} n \\ m \end{array}\right) = \frac{n\,!}{(n-m)\,!\,m\,!}$,$Z_p = T^{\frac{1}{2}[1\,+\,(-1)^p]}$ and T denotes transpose. It follows that

$$F(x) = f(x_k) \ + \ h^T A_1 \ + \ \frac{1}{2}h^T H(x)h \quad (7)$$

where$H(x) = A_2 \ + \ \frac{1}{12}\left[g(x) \ + \ 3g(x_k)\right]^T h.$

Similarly,

$$\nabla F(x) = A_1 \ + \ A_2 h^T \ + \ \frac{1}{2}h^T A_3 h \ + \ \frac{1}{3\,!}h^T(h^T A_4 h) \quad (8)$$

$$\nabla F(x_{k+1}) = A_1 \ + \ \frac{1}{6}\left[g(3x_{k+1} - 2x_k) \ + \ 3g(x_{k+1}) \ - \ 4g(x_k)\right] \quad (9)$$

$$\nabla F(x_k) = A_1 = g(x_k). \quad (10)$$

$$\nabla F(x_{k+1}) = \nabla F(x_k) \; + \; \frac{1}{6}\left[g(3x_{k+1} - 2x_k) \; + \; 3g(x_{k+1}) \; - \; 4g(x_k)\right]$$

$$= \frac{1}{6}\left[g(3x_{k+1} - 2x_k) \; + \; 3g(x_{k+1}) \; + \; 2g(x_k)\right]. \tag{11}$$

Using Dai and Yuan [12] $\beta_k$and$G_{k+1} = \nabla F(x_{k+1})$, we derive a new algorithm in which the directions of search, D_0, D_1, ..., D_k are H conjugate. With a given x_0, $\alpha_k$is computed such that

$$F(x_k \; + \; \alpha_k D_k) < F(x_k), \tag{12}$$

$$D_k = -G_k \; if \; k = 0; \;\; D_k = -G_k + \beta_{k-1} K_{k-1} \; if \; k \geq 1$$

and$x_{k+1} = x_k \; + \; \alpha_k D_k, \; k = 0, \; 1, \; ...$
The algorithm is summarized below.

2.1 Algorithm 1: Fourth-order nonlinear conjugate gradient method (FONCGM)

Step 1: Select s_0, $x_0 \in \Re^N$, $N \geq 2$ and $\varepsilon > 0$(a small number: 0.000001). Set

$D_0 = -G_0 = -\nabla F(x_0)$,$H(x_0)$, $s_0 = 0$(zero vector) and $k = 0$.

Step 2: If $||G_k|| \leq \varepsilon$, stop. Take $s_k$as an estimate of the
exact solution of problem (1). Otherwise go to step 3.

Step 3: Compute α_k such that $F(s_k \; + \; \alpha_k D_k) < F(s_k)$[25]

Step 4: Compute

$$s_{k+1} = s_k \; + \; \alpha_k D_k, G_{k+1} = \frac{1}{6}\left[g(3s_{k+1} - 2s_k) \; + \; 3g(s_{k+1}) \; + \; 2g(s_k)\right].$$

$\beta_k = \frac{||G_{k+1}||^2}{D_k^T(G_{k+1} - G_k)}$, $D_k^T(G_{k+1} - G_k) \neq 0$. If $D_k^T(G_{k+1} - G_k) = 0$, $\beta_k = 0$.$D_{k+1} = -G_k \; + \beta_k D_k$.

Step 5: Set $k = k + 1$. Go to step 2.

Employing Tang and Yuan [20] results $\alpha_k$may be computed to satisfy Wolfe's conditions. Next, we construct a sub problem for a known trust region.

2.2 Construction of a sub problem for anew trust region

The basic trust region method is characterized by a constrained minimization problem and
inexact line search procedure[18]. An approximate model function (Q) is usually defined within a trust region where it is hoped that the solution of the unconstrained minimization problem resides. At the current iteration x_k, the basic trust region problem is

$$\min_{s \in \Re^N} Q_k(s) = f(x_k) + g_k^T s + \frac{1}{2} s^T B_k s \tag{13}$$

$$s.\, t \quad ||s|| \; \leq \Delta_k,$$

where $g_k = g(x_k)$,$B_k = \nabla^2 f(x_k)$ and $\Delta_k$is the prescribed trust region radius. The constraint condition restricts us to points on or inside the disc

whose radius is Δ_k.Inside the disc, we solve the unconstrained problem

$$\min_{s\in\Re^N} Q_k(s) = f(x_k) + g_k^T s + \frac{1}{2}s^T B_k s. \tag{14}$$

On the boundary of the disc we solve the inequality problem

$$\min_{s\in\Re^N} Q_k(s) = f(x_k) + g_k^T s + \frac{1}{2}s^T B_k s \tag{15}$$

$$s.\,t \quad ||s|| \leq \Delta_k,$$

Using $H(s)$:
$H(s) = A_2 + \frac{1}{12}\left[g(s) + 3g(x_k)\right](s - x_k)^T$in place of B_k,$A_2 = \nabla^2 f(x_k)$ and $A_1 = \nabla f(x_k)$ equation (13) becomes:

$$\min_{s\in\Re^N} Q_k(s) = f(x_k) + G_k^T s + \frac{1}{2}s^T H(s) s \tag{16}$$

$$s.\,t \quad ||s|| \leq \Delta_k,$$

Various methods have been used to solve the trust region subproblem (13) approximately for an iterates$s_k$through inexact line search procedures. Some of these methods fail to find the correct value ofs_k. Thus, we restate the basic trust region algorithm as follows.

2.3 Algorithm 2: The trust region method

Initialization: Set $k = 0,\ \Delta_0 > 0$. Choose $x_0,\ nv \in (0,\ 1),\ ns \in (0,\ nv),\ y_2 \geq 1$ and $y_1 \in (0,\ 1)$.
$nv = 0.0001,\ ns = 0.25,\ y_1 = 0.5,\ and\ y_2 = 2$are typical values.
Until convergence, repeat the following.

(1) Construct a trust region subproblem: equation (16).
(2) Use algorithm (1) to solve the trust region subproblem approximately for s_k. Define

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{Q_k(0) - Q_k(s_k)}.$$

(3) If $\rho_k > nv$ ("a successful step"), set $x_{k+1} = x_k + s_k$,
(4) Else, (" an unsuccessful step"), set $x_{k+1} = x_k$.
(5) If $\rho_k < ns$, set $\Delta_{k+1} = y_1\Delta_k$,
(6) Else, set $\Delta_{k+1} = y_2\Delta_k$.
(7) Increase k by 1.

3. CONVERGENCE ANALYSIS

We assume that the objective function satisfies the following conditions and theorem [3].

Assumptions [3]:

i. f is bounded below in $\Re^N$and is four times continuously differentiable in a neighborhood Z of the level set

$$L = \left\{x \in \Re^N : \ f(x) \leq f(x_0)\right\}.$$

ii. The gradient, $g(x)$, is Lipschitz continuous in Z,namely, there exists a constant $Lc > 0$such that

$$||\nabla f(x) - \nabla f(y)|| \leq Lc||x - y||, \ \ x, y \in Z.$$

iii. The extended hessian matrix $H(x)$ is positive definite.

Lemma:

(i). Suppose that x_0 is a starting point for which the above assumptions are satisfied. Consider any method of the form (2), where D_k, a vector, is the descent direction and $\alpha_k$satisfies the standard Wolfe conditions [18], then

$$\sum_{k\geq 0} \frac{\left(G_k^T D_k\right)^2}{||D_k||^2} < \infty$$

(ii). Suppose that x_0 is a starting point for which the above assumptions are satisfied. Let
$\{x_k, \ k = 1, \ 2, \ ...\}$ be generated by algorithm (1). Then, the algorithm either terminates at a stationary point or converges in the sense that

$$\lim_{k\to\infty} \inf ||G(x_k)|| = 0$$

Theorem: Suppose that f in (1) is continuously differentiable and bounded below and the norm of $H(x)$ is bounded, the iteration $\{x_k\}$ generated by the algorithm (2) satisfies $x_k \to x*$ as $k \to \infty$ and the matrix $H(x)$ of f is positive definite. Let ε_k be the relative error in algorithm (1). If $\varepsilon_k \to 0$, then $\{x_k\}$ converges linearly:

$$\lim_{k\to\infty} \frac{||x_{k+1} - x*||}{||x_k - x*||} = 0.$$

Proof (Lemma (i)):

Dai and Yuan proved this lemma: $\sum_{k\geq 0} \frac{\left(g_k^T d_k\right)^2}{||d_k||^2} < \infty$. A Similar proof is presented for algorithm (1), since the search directions $d_k$and D_k have similar definitions. This is obvious on using the assumptions of lemma (i), and $k$in place of $k + 1$in equation (11):

$$\frac{\left(G_k^T D_k\right)^2}{||D_k||^2} = \frac{\frac{1}{36}\left\{\left[g(3x_k - 2x_{k-1}) + 3g(x_k) + 2g(x_{k-1})\right]^T D_k\right\}^2}{||D_k||^2}$$

$$= \frac{\frac{1}{36}\left\{\left[g(3x_k - 2x_{k-1}) + 3g(x_k) + 2g(x_{k-1})\right]^T d_k\right\}^2}{||d_k||^2},\ d_k = D_k$$

$$= \frac{\frac{1}{36}\left\{\left(g(3x_k - 2x_{k-1})^T d_k\right) + 3\left(g(x_k)^T d_k\right)\right\}^2}{||d_k||^2},\ g(x_{k-1})^T d_k = 0$$

$$\leq \frac{\frac{1}{36}\left\{\left(g(3x_k - 2x_{k-1})^T d_k\right)^2 + 9\left(g(x_k)^T d_k\right)^2 + 6\left(g(3x_k - 2x_{k-1})^T d_k\right)^2 + 6\left(g(x_k)^T d_k\right)^2\right\}}{||d_k||^2}$$

$$= \frac{1}{36}\left\{\frac{\left(g(3x_k - 2x_{k-1})^T d_k\right)^2}{||d_k||^2} + 9\frac{\left(g(x_k)^T d_k\right)^2}{||d_k||^2} + 6\frac{\left(g(3x_k - 2x_{k-1})^T d_k\right)^2}{||d_k||^2} + 6\frac{\left(g(x_k)^T d_k\right)^2}{||d_k||^2}\right\}$$

$$\sum_{k\geq 0}\frac{\left(G_k^T D_k\right)^2}{||D_k||^2} \leq \frac{1}{36}\left\{\sum_{k\geq 0}\frac{\left(g(3x_k - 2x_{k-1})^T d_k\right)^2}{||d_k||^2} + 9\sum_{k\geq 0}\frac{\left(g(x_k)^T d_k\right)^2}{||d_k||^2} + \right.$$

$$\left. 6\sum_{k\geq 0}\frac{\left(g(3x_k - 2x_{k-1})^T d_k\right)^2}{||d_k||^2} + 6\sum_{k\geq 0}\frac{\left(g(x_k)^T d_k\right)^2}{||d_k||^2}\right\} < \infty$$

Thus,
$\sum_{k\geq 0}\frac{\left(G_k^T D_k\right)^2}{||D_k||^2} < \infty$as required.

Proof (Lemma (ii)):
Dai and Yuan also proved this lemma for his algorithm algorithm. He proved that $\lim_{k\to\infty}\inf||g(x_k)|| = 0$. It follows that $\lim_{k\to\infty}\inf||g(3x_k - 2x_{k-1})|| = 0$and $\lim_{k\to\infty}\inf||g(x_{k-1})|| = 0$ at points of convergence of his algorithm. Using the assumptions of lemma (ii), equation (10) ($G_k = G(x_k) = \nabla F(x_k) = g(x_k)$) and Dai's proof, it follows that $\lim_{k\to\infty}\inf||G_k|| = \lim_{k\to\infty}||g_k|| = 0$. Alternatively, using k in place of $k+1$ in equation (11) gives

$$\lim_{k\to\infty}\inf||G_k|| = \lim_{k\to\infty}\inf\frac{1}{6}||\left\{g(3x_k - 2x_{k-1}) + 3g(x_k) + 2g(x_{k-1})\right\}||$$

$$\leq \lim_{k\to\infty}\inf\frac{1}{6}||g(3x_k - 2x_{k-1})|| + \lim_{k\to\infty}\inf\frac{1}{6}||3g(x_k)||$$

$$+ \lim_{k\to\infty}\inf\frac{1}{6}||2g(x_{k-1})|| = 0.$$

$\lim_{k\to\infty}\inf||G_k|| \leq 0$ and $||G_k|| \geq 0$ implies that $\lim_{k\to\infty}\inf||G_k|| = 0$.

Proof of theorem: The proof is available in many literatures. Noting that $G_k = G(x_k) = \nabla F(x_k) = g(x_k)$, the proof is same since the assumptions on algorithm (1) meet the requirements of this theorem. Using $M \geq m \geq 0$, $\varepsilon_k = x_k - x*$ and the results from NMC [26] we have

$$||x_{k+1} - x*||^2 = \left(x_{k+1} - x*\right)^T\left(x_{k+1} - x*\right) = \left(x_{k+1} - x*\right)^T\left(x_k + \alpha_k d_k - x*\right)$$

$$\leq \frac{M}{n}\left(\frac{1-R}{1+R}\right)^{2(k+1)} ||x_k - x*||^2 \leq \frac{M}{n}\left(\frac{1-R}{1+R}\right)^{2(k+1)} ||x_0 - x*||^2; \;\; R = \frac{m}{M}.$$

$||x_{k+1} - x*|| \leq \sqrt{\frac{M}{n}}\left(\frac{1-R}{1+R}\right)^{(k+1)} ||x_k - x*||.$ $\lim_{k\to\infty} \frac{||x_{k+1}-x*||}{||x_k-x*||} = 0$, since $\lim_{k\to\infty} \sqrt{\frac{M}{n}}\left(\frac{1-R}{1+R}\right)^{(k+1)} = 0.$

Similar proof is available in Steihaug [27] or Yuan [13].

4. NUMERICAL EXPERIMENTS

FONCGM solves equation (1) where $f(x)$ is given in the following test problems from [15].

Problem 1. A simple function:

$$f(x) = x_1^3 - 3x_1 + x_2^2 - 2x_2; \quad [x_0] = [2 \;\; 3].$$

Problem 2. Penalty function I (problem (1) in [15]) $f(x) = \sum_{i=1}^{N} 10^{-5}(x_i - 1)^2 + \left[\left(\sum_{i=1}^{N} x_i^2\right) - \frac{1}{4}\right]^2, \quad [x_0]_i = i.$

Problem 3. Variable dimensioned function (problem (2) in [15])

$$f(x) = \sum_{i=1}^{N}(x_i - 1)^2 + \left[\sum_{i=1}^{N} i(x_i - 1)\right]^2 + \left[\sum_{i=1}^{N} i(x_i - 1)\right]^4, \quad [x_0]_i = 1 - \frac{i}{N}.$$

Problem 4. Trigonometric function (problem (3) in [15])

$$f(x) = \sum_{i=1}^{N}\left[N - \sum_{j=1}^{N}\cos(x_j) + i(1 - \cos(x_i)) - \sin(x_i)\right]^2, \quad [x_0]_i = 1/N.$$

Problem 5. A penalty function (problem (4) in [15])

$$f(x) = 1 \; + \; \sum_{i=1}^{N} x_i + 10^3\left(1 - \sum_{i=1}^{N} 1/x_i\right)^2 + 10^3\left(1 - \sum_{i=1}^{N} i/x_i\right)^2, \quad [x_0]_i = 1.$$

Problem 6. Extended Rosenbrock function (problem (5) in [15])

$$f(x) = \sum_{i=1}^{N}\left[100\left(x_{2i} - x_{2i-1}^2\right)^2 + (1 - x_{2i-1})^2\right], \quad [x_0]_{2i-1} = -1.2, \; [x_0]_{2i} = 1.$$

Problem 7. Linear function-rank 1 (problem (8) in [15] with new initial values)

$$f(x) = \sum_{i=1}^{m}\left[i\left(\sum_{j=1}^{N} j x_j\right) - 1\right]^2, \; m \geq N, \;\; [x_0]_i = \frac{1}{i}.$$

We set $\Delta_0 = ||x_k|| \; + \; 0.0001, \;\; nv = 0.0001, \; ns = 0.25, \; y_1 = 0.5, \; and \; y_2 = 2.$

A MATLAB program computed the following results, based on algorithm (2). The stopping criterion is $||g(x_k)|| < 0.000001$.

Table 1. Solution of problem 1.

Iteration	$f(x_k)$	$\|\|g(x_k)\|\|$	Δ_k
1	5	9.848857856263518	3.605651275463989
2	-2.796762459631599	1.688000016092676	7.211302550927979
3	-2.949533386040339	0.799178789121903	14.422605101855957
4	-2.999032137269256	0.108400497711067	28.845210203711915
5	-2.999990393938984	0.010742123370355	57.690420407423829
6	-2.9999999973450	0.0001784861088	115.3808408148477
7	-3.0000000000000	0.0000003430505	230.7616816296953

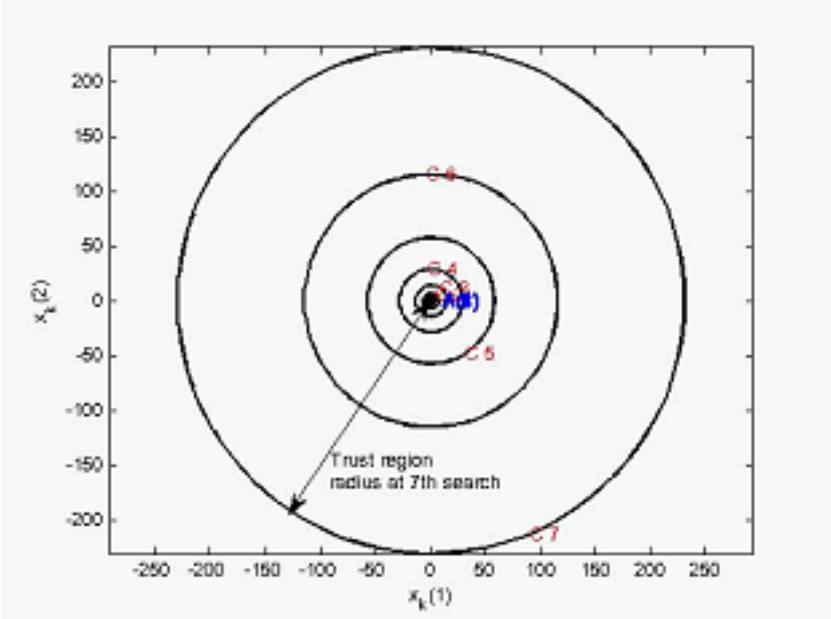

Fig. 1(Problem (1)) : Circles with trust region radius, Δ_k.

Table 2: Solution of problem 6
(**:** $N = 10,000$; $x = (0.999994,\ 0.999989,\ ...,\ 0.999995)$**)**.

Iteration	1	2	3	4	5
f	121000	973.451	972.995	609.422	401.676
$\|g\|$ 2	2.71137E8	741.191	958.236	337920	446339

Iteration	6	7	8	9	10
f	245.384	115.535	18.2926	4.00011	3.89042
$\|g\|$ 2	381378	220755	32324.7	246.181	4.43934

Iteration	11	12	13	14	15
f	3.88515	3.58551	0.00424653	0.00109777	0.00109581
$\|g\|$ 2	7.59182	631.711	6.53287	0.00286512	0.00119897

Iteration	16	17	18	19
f	0.000729101	1.12854E-6	3.7389E-7	3.73282E-7
$\|g\|$ 2	0.505587	0.00156775	8.23191E-7	4.4211E-7

In Table (3), below, we compare the numerical results of problems (2 to 7), Using FONCGM (algorithm (2)) with other results: Zhen-Jun and Guo (ZGM) [27], FR, PRP, CD, DY, HS and LS. **Table 3**. Comparison of numerical results.

Problem	N	FONCGM	ZGM	PRP	HS	FR	CD	DY	LS
2	10000	69	59	58	67	68	66	67	68
	5000	28	29	37	40	37	37	37	37
3	10000	30	33	fail	Fail	57	69	60	Fail
	5000	24	28	fail	Fail	37	37	37	Fail
4	1000	20	31	41	70	44	45	43	44
	500	17	24	29	41	30	36	30	31
5	10000	30	32	51	Fail	67	48	59	Fail
	5000	20	23	40	37	41	39	40	Fail
6	10000	68	50	fail	66	38	62	41	70
	5000	18	21	fail	27	32	30	26	30
7	10000	30	28	69	Fail	38	37	36	Fail
	5000	17	18	26	27	23	29	28	27

Performance profiles have been introduced by Dolan and More' [28]. The main idea is to show, graphically, the relative performance of various solvers on a given set of problems. That is, the curves are used to compare the efficiency of a set S of solvers on a set P of test problems. $t_{p,s}$ denotes the performance of a solver s (based on the number of iterations, function evaluations, gradient evaluations or time of computation) on the problem p. $r_{p,s}$ denotes the relative performance of a solver s on a problem p and $r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s}:\, s\in S\}}$.Our assumption is that $r_{p,s} \leq w,\ w \in \Re$,for all solvers s on the problems p. $r_{p,s} = w$ if solver s cannot solve problem p. The performance profile of the solver s is the function$y_s : [1,\ w] \to [0,\ 1]$ such that $y_s(t) = \frac{n(\{p\in P: r_{p,s}\leq t\})}{n(P)}$ where $n(.)$ denotes the number of elements of a set. The performance profiles of the methods discussed in this paper are shown below.

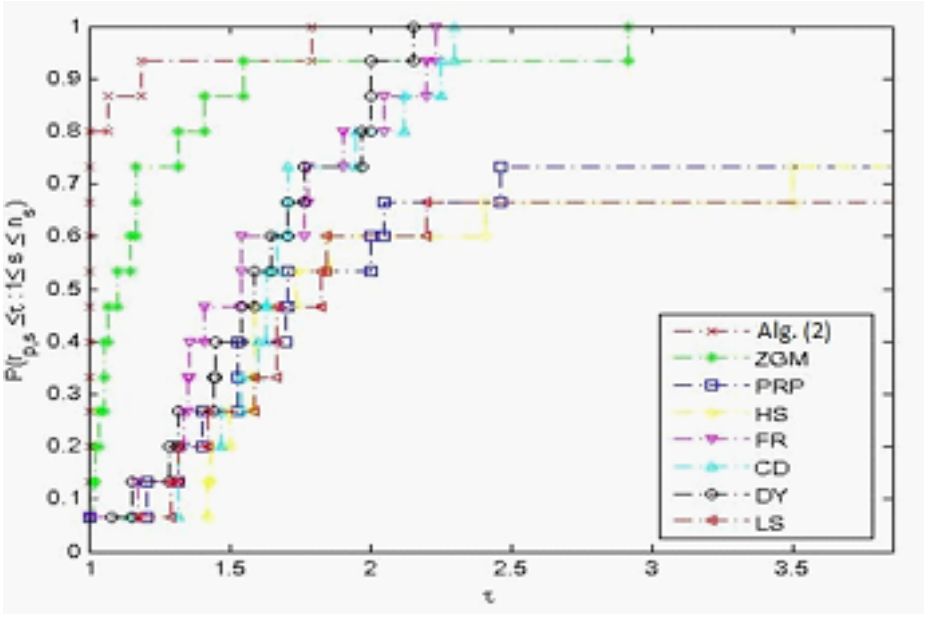

Fig. 2: Performance profiles on the number of iterations.

5. DISCUSSION ON NUMERICAL RESULTS

The solution of problem (1) explained the major steps of the new method. Table (3) contains the numerical results obtained for the new method vis-à-vis some existing methods. These results indicate that the new method compares favorably well with the other methods. The execution time depends on various methods used for evaluating the step lengths and the speed of computer's processing unit. We observed that the new method is relatively faster in some of the iterations recorded for the tested problems. In confirmation, Tables (2, 3) and Figure (1) show that the new method is fast and less costly as the number of function evaluations per computed problem is relatively low. Finally, we saw that the results are very close to the exact solutions.

6. CONCLUSION

In this paper, a new fourth-order nonlinear conjugate gradient method and trust regions in unconstrained optimization are presented. Some of the basic properties of the method have been explored and exploited. Numerical results show that the method is highly efficient and reliable. Further research is underway to develop an algorithm for constrained optimization, using the fourth-order model.

ACKNOWLEDGEMENTS

We acknowledge our colleagues and referees whose comments improved the original version of this paper.

REFERENCES

[1] N. Andrei, *40 conjugate gradient algorithms for unconstrained 0ptimization: A survey on their definition*, ICI Technical report No. 13/08, Research Institute for Informatics, Centre for Advanced Modeling and optimization, Romania, 2008.

[2] S. Sanmtias and E. Vercher,*A generalized conjugate gradient algorithm*, Journal of optimization theory and Applications, **98**, 489 – 502, 1988.

[3] Y.H. Dai and Y. Yuan, *Nonlinear conjugate gradient methods.* Shanghai Science and Technology Press, Shanghai; 187-193, 2000.

[4] R. Fletcher and C. Reeves,*Function Minimization by conjugate gradients.* Computer J., **7**,149-154, 1964.

[5] Z. J. Shi, *Nonlinear conjugate gradient method with exact line search.*Acta Math. Sci. Ser. A Chin. Ed., **24**(6), 675 - 682, 2004.

[6] W.W. Hagger, H. Zhang, *A new conjugate gradient method with guaranteed descent and an efficient line search*, SIAM Journal on Optimization, **16**, 170-192, 2005

[7] W.R. Boland, E.R Kamgnia, and J.S. Kowalik, *A CG optimization method invariant to nonlinear scaling*, Journal of optimization theory and applications, **27**(2), 221 - 230, 1979.

[8] I. Fried, *N-step conjugate gradient minimization scheme for nonquadratic functions,* AIAA Journal, **9**, 2286-2287, 1971.

[9] D Goldfarb, *Variable metric and conjugate-direction methods in unconstrained optimization: Recent developments,* ACM proceedings, Boston, Massachusetts. 155, 1972.

[10] Y.H.Dai, *Conjugate gradient methods with rmijo-type line searches.* Act MathematicaeApp, **18**,123-130, 2002.

[11] Y.H. Dai, J.Y. Han, G.H. Liu, D.F. Sun, H.X. Yin and Y. Yuan, *Convergence properties of nonlinear conjugate gradient methods.* SIAM J. Optim., **10**, 345-358, 2000.

[12] Y.H.Dai and Y. Yuan, *Convergence properties of the conjugate gradient descent method* Adv. Math., **25**(6), 552-562, 1996.

[13] Y. H. Dai and Y. Yuan,*A nonlinear conjugate gradient method with a strong global convergence property.* SIAM J. Optim., **10**, 177-182, 1999.

[14] R. Fletcher, *Practical methods of optimization*, John Wiley & Sons, New York, 19087.

[15] Z.J. Shi and J. Guo, *A new algorithm of nonlinear conjugate gradient method with strong convergence*, Computational and Applied Mathematics, **27**(1) ,93-106, 2008.

[16] E. Polak and G. Ribiere, *Note sur la convergence de directions conjugees.* Rev. Francaise InformatRechercheOpertionelle, 3e Annee, **16**, 35-43, 1969.

[17] B.TPolyak, *The conjugate gradient method in extreme problems.*USSR Comp. Math.and Math. Phys., **9**, 94-112, 1969.

[18] M.R. Hestenes and E.L.Stiefel, *Methods of conjugate gradients for solving linear systems.*J. Res. Nat. Bur. Stds, **49**, 409-436, 1952.

[19] Y. Liu and C. Storey,*Efficient generalized conjugate gradient algorithms.* J. Optim. Theory Appl., **69**, 129-137, 1991.

[20] M. Tang and Y. Yuan, *Using truncated conjugate gradient method in trust-region method with two subprograms and backtracking line search*, Computational and Applied Mathematics, **29**(2) ,89-106, 2010.
[21] J. Nocedal and Y. Yuan, *Combining trust-region and line search techniques. Advances in nonlinear programming*, 153-175, 1998.
[22] Y.Yuan and J. Stoer, *A subspace study on conjugate gradient algorithms.* Z. Angew Math. Mech., **75**, 69-77, 1995.
[23] A.V. Smirnov, *Introduction to tensor calculus*, http://faculty.gg.uwyo.edu, 2004.
[24] E. Nwaeze and O.M. Bamigbola, *A new conjugate gradient method for solving nonlinear unconstrained optimization problems.* International Journal of Advanced Engineering Sciences and technologies., **2**(2), 187 - 193, 2010.
[25] E. Nwaeze, S.U. Isienyi and L. Zhengui, *An augmented cubic line search algorithm for solving high-dimensional nonlinear optimization problems.* Journal of Nigerian Mathematical Society (NMS), **32**, 185 - 191, 2013.
[26] National Mathematical Centre (NMC), Abuja, *Foundation Postgraduate course on computational methods and applications in optimization*, 15 - 17, 1999.
[27] T. Steihaug,*The conjugate gradient method and trust regions in large scale optimization.*SIAM J. Numer. Anal., **20**, 625-637, 1983.
[28] E.D. Dolan and J.J. More', *Benchmarking optimization software with performance profiles*, Mathematical programming, **91**(2), 201-213, 2002.

Department of Maths/Comp. Sc./Stat./Info, Federal University Ndufu-Alike, Ikwo, Nigeria.
E-mail address: nwaezeema@yahoo.com
Department of Maths/Comp. Sc./Stat./Info, Federal University Ndufu-Alike, Ikwo, Nigeria.
E-mail address: omenyilo@yahoo.com