IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 15, Issue 4 Ser. I (Jul – Aug 2019), PP 42-47
www.iosrjournals.org

Quasi- Newton Method for Solving Non- Linear Optimization Problems of Convergence Functional

Adamu Wakili
Department of Mathematical Sciences Faculty of Science Federal University Lokoja, Nigeria.

Abstract: *Quasi-Newton Method is used to solve no-linear optimization problems as an extension of Newton Method for multi-variables. This method is also used to solve second order derivatives of the o f the objective functions in the form of Hessian Matrix. Quasi-Newton Method is the most effective method for finding minimizers of a smooth non-linear function when second derivatives are either unavailable or too difficult to compute. The algorithm method is used on the functions to generate solutions and theses solutions are found to be convergent after certain number of iterations.*
The new point is then obtained by the sum of the previous point and the result is multiplied by the step length and the search direction. This process is continued until convergent is reached and this method is easy to handle . The Qusi-Newton Method uses less time for computations and less number of iterations.
Keywords: *Quasi, optimization, minimization, Matrix, symmetric, Genetic Algorithm, Particle Swarm and positive definite.*

Date of Submission: 25-06-2019 Date of acceptance: 12-07-2019

I. Introduction

Most Optimization problems are solved using Newton and quasi Newton Methods, the Newton Method was discovered by Sir Isaac Newton in 1664, who used the method to find the root of the function of a single variable, and later in 1690 the method was extended to solve multivariable optimization problems by deriving Taylor Series expansion. This method was also used in solving second partial derivatives of the objective function in the form of a matrix (Hessian Matrix). When solving the optimization problem, the evaluation of the Hessian Matrix and the gradient of the objective function at each step of the iteration and due to this difficulty with Newton Method in solving complicated parameters of the objective function [8].

Quasi Newton Method was derived by [1, 4] for the solution of Non-Linear Optimization Problems. In Quasi Newton Method the Hessian Matrix of the second derivative of the function to be minimized does not need to be computed [2].

Quasi Newton Method is arguably the most effective method for finding a minimizer of a smooth non-linear function when second derivatives are either unavailable or too difficult to calculate.

Quasi Newton Method builds up second derivative information by estimating the curvature along a sequence of smooth direction [6]. Quasi Newton Method is one of the most successful update of Broyden-Fletcher-Goldforb –Shanno (BFGS) formula, which is a member of the wider Broyden Class of rank-two updates. Quasi Newton Method requires approximately large number of iterations and functions evaluations on some problems [7].

II. Literature Review

In recent years works have been carried out in the usage and the convergence of Newton Method. This method has been modified by many researchers such as [12, 9, 10, 11 and 3] who re-derived the Newton Method by using the rectangular rule t0 compute the integral.

However, in [12] and [5] modified the Method for the solution of multivariable case and having a convergent of the order 3. According to [2], Quasi Newton Method for multi objective optimization converges super linearly to the solution of the origin problem, if all functions involved are twice continuously differentiable and strongly convex. As numerical performance of the Quasi Newton Method for multi objective optimization problems with strong convex objective function can be efficiently solved. The Method is quite robust with respect to the number of variables and the starting point [14 and 9].

[13] and [5] provided the self scaling Quasi Newton Method for large scale unconstraint optimization, [1] and [11] developed a new direction of modification of the Quasi Newton Method.

According to [8] there are plenty of variants like the simplified Newton Method, Quasi Newton Method and Global Newton Method. Only very few of them exhibit quadratic convergence property. [5] presented a modified Quasi Newton Method for structure of unconstraint optimization. Quasi Newton Method is useful for unconstraint optimization problems, non linear equation and non linear least square problems since it is a Line Search Method which needs a line search procedure after determining a search line rule to choose a step size along the search direction [11]. Some of the widely known researchers of Quasi Newton Method for large scale unconstraint optimization family include Broyden's Method, the SRI formula and the DFP Method [4, 3, 1]. There are four main approaches to non smooth convex optimization problem: Quasi Newton Method, Bundle Method, Stochastic Dual Method and Smooth Optimization. The Non Smooth Optimization Method try to find a descent quasi Newton direction at every iteration and invoke a line search function along that direction [14].

The modern optimization methods also called Non traditional optimization methods have emerged as powerful aJnd popular methods for solving complex Engineering Optimization Problems in the recent years. These methods include Genetic Algorithms, simulation Annealing, Particle Swarm Optimization and Ant Colony Optimization. The Genetic Algorithms are computerised Search and Optimization Algorithms based on the mechanism of the natural genetic and natural selection 6, 7 and 9].

III. Methodology

The necessary condition for f(x) to have an optimum at x^* is that $f^1(x^*) = 0$.

Quasi Newton Method is for minimization of the multi-variable functions. For this, consider the quadratic approximation of the function $f(x)$ at $x = x^*$, using Taylor's Series expansion.

$$f(x) = f(x_i) + \nabla f_i (x - x_i) + \frac{1}{2}(x - x_i)^2 J_i, \quad i = 1, 2, 3, \ldots \quad (1)$$

where $|J_i| = |J|$ at x_i is the matrix of second partial derivative (Hessian Matrix) of f evaluated at the point x. By setting the partial derivative of (1) equal to zero, and then find the minimum of $f(x)$ and solving

$$\frac{\partial f(x)}{\partial x_i} = 0, \; i = 1, 2, 3....n \quad (2)$$

$$\nabla f = \nabla f_i + J_i (x - x_i) = 0, \qquad i = 1, 2, 3, ...n \quad (3)$$

If J_i is non singular, then $|J_i \neq 0|$ and (3) can be solved by obtaining improved approximation of $x = x_{i+1}$ giving

$$x_{i+1} = x_i - (J_i)^{-1} \nabla f_i, \quad i = 1, 2, 3, \; ....n \quad (4)$$

Since higher order terms have been neglected in (3) and (4), it is useful in iteration of the optimum solution of x^*.

The sequence of points $x_1, x_2, x_3, \; ,,, x_{i+1}$ can be shown to converge to the actual solution x^* from any initial point, x_0 sufficiently close to the solution x^*, provided that (J_i) is non singular. It can be seen that the second partial derivatives of the objective function (in the form of the matrix (j_i)) and hence is a second order method. It could be noted that the method requires the following ;

(i) Storing nxn matrix (J_i)

(ii) The inverse of (J_i) at each step

(iii) The evaluation of $(J_i)^{-1} \nabla f_i$ at each step.

These features make the method impracticable for problems involving a complicated objective functions with large number of variables (Odunayo, 2014).

The iterative process used of the form

$$X_{i+1} = X_i - (J_i)^{-1} \nabla f(x_i) \quad (5)$$

where the Hessian Matrix $[J_i]$ is composed of the second partial derivatives of the f and varies with the decision vector X_i for non quadratic objective function, f. The basic idea behind the Quasi Newton or Variable Matrix Method is to approximate either $[J_i]$ by another $[A_i]$ or $[J_i]^{-1}$ by another matrix $[B_i]$, using only the first partial derivatives of f.

If $[J_i]^{-1}$ is approximated by $[B_i]$ of (5) can be expressed as

$$X_{i+1} = X_i - \lambda[B_i]\nabla f(x_i) \quad (6)$$

where λ can be considered as the optimal step length along the direction of

$$S_i = -[B_i]\lambda f(x_i) \quad (7)$$

It can be seen that the steepest descent direction method can be obtained as a special case of (7) by setting $[B_i] = [I]$.

The computation of $[B_i]$ in order to implement (6), an approximate inverse of the Hessian Matrix, $[B_i] = [A_i]^{-1}$, is to be computed. For this, first expand the gradient of f about an arbitrary reference point, X_0, using Taylor's Series as:

$$\nabla f(x) = \nabla f(x_0) + [J_i](X - X_0) \quad (8)$$

If we pick two points X_i *and* X_{i+1} and use $[A_i]$ to approximate $[J_0]$, (7) can be written as

$$\nabla f(X_{i+1}) = \nabla f(X_i) + [A_i](X_{i+1} - X_i) \quad (9)$$

$$\nabla f(X_i) = \nabla f(X_i) + [A_i](X_i - X_i) \quad (10)$$

Subtracting (9) from (10), to get

$$[A_i]d_i = g_i \quad (11)$$

where $d_i = X_{i+1} - X_i$ (12)

$$g_i = \nabla f_{i+1} - \nabla f_i \quad (13)$$

where the solution of (11) can be written as :

$$d_i = [B_i]g_i \quad (14)$$

Where $[B_i] = [A_i]^{-1}$ denotes an approximation to the inverse of the Hessian Matrix, $[J_i]^{-1}$. It can be seen that (14) represents a system of n-equations in n^2 unknown elements of the Matrix $[B_i]$. Thus for $n \geq 1$ the choice of $[B_i]$ is not unique and one would like to choose $[B_i]$ that is closest to $[J_0]$, in some sense. Many techniques have been suggested in the computation of $[B_i]$ as well as the iterative process of $[B_{i+1}]$. It should be noted that $[B_i]$ is symmetric and positive definite. Since this matrix is symmetric and positive definite then the following modifications can be made

$$[B_{i+1}] = [B_i] + [\lambda B_i] \text{ and } [\lambda B_i] = CZZ^T \quad \textbf{(15)}$$

Where constants C and the n-component vector, Z are to be determined form above equations. Therefore these yield $[B_{i+1}] = [B_i] + CZZ^T$.

By letting (7) satisfies the Quasi Newton condition $d_i = [B_i]g_i$ thus

$$d_i = ([B_i] + CZZ^T)g_i$$

$d_i = [B_i]g_i + CZ(Z^T g_i)$ **since** $Z^T g_i$ **is a scalar, the** $CZ = \dfrac{d_i - [B_i]g_i}{Z^T g_i}$

Thus a simple choice for Z and C could be $Z = d_i - [B_i]g_i$ and $C = \dfrac{1}{Z^T g_i}$

This leads to the unique rank 1 and the modified formula for $[B_{i+1}]$ is

$$[B_{i+1}] = [B_i] + [\lambda B_i] \equiv [B_i] + \frac{(d_i - [B_i]g_i)(d_i - [B_i]g_i)^T}{(d_i - [B_i]g_i)^T g_i}$$

This formula has been attributed to Broyden Algorthm.

Modification of the symmetric rank

Step 1:

$i = 0$; *select* X_0, *tolerance* $\in$ *and a real symmetric positive definite matrix* $[J_0] = [B_0]$

Step 2: *If* $g_i \leq \in$, *Stop, else set* $d_i = -[B_i]g_i$

Step 3: $Compute \quad \alpha_i = \arg\min \ f(X_i + \alpha d_i) X_{i+1} = X_i + \alpha d_i$

Step 4: $Compute \quad \lambda X_i = \alpha_i d_i$

$$\nabla_i = g_{i+1} + g_i$$

$$[B_{i+1}] = \frac{[B_i] + (\lambda X_i - [B_i]\lambda g_i)(\lambda X_i - [B_i]\lambda g_i)^T}{(\lambda X_i - [B_i]\lambda g_i)^T \lambda g_i}$$

Step 5: $Set \ i = i+1$; $Goto \ step \ 2$

For conjugate direction algorithm , use rank 2 as follows:

$\lambda[B_i] = C_1 Z_1 Z_1^T + C_2 Z_2 Z_2^T$ where $C_1 \quad and \quad C_2$ and the n-component vectors are to be determined. Therefore this result follows;

$[B_{i+1}] = [B_i] + C_1 Z_1 Z_1^T + C_2 Z_2 Z_2^T$ and also letting (15) satisfies the Quasi Newton condition

$d_i = [B_i] g_i + C_1 Z_1 (Z_1^T g_i) + C_2 Z_2 (Z_2^T g_i)$ where $Z_1 g_i \quad and \quad Z_2 g_i$ can be identified as scalars.

Although, the vectors $Z_1 \quad and \quad Z_2$ are not unique, the following choice can be made to satisfy them:

$Z_i = d_i$ and $Z_2 = [B_i] g_i$

$C_1 = \dfrac{1}{Z_1^T g_i}$ and $C_2 = \dfrac{-1}{Z_2^T g_i}$ this can be expressed as

$$[B_{i+1}] = [B_i] + [\lambda B_i] \equiv [B_i] + \frac{d_i d_i^T}{d_i^T g_i} - \frac{([B_i] g_i)([B_i] g_i)^T}{([B_i])^T g_i}$$

Since $X_{i+1} = X_i + \lambda S_i$ where S_i $is \ the \ direction$ $d_i = X_{i+1} - X_i$ can be written as

$d_i = \lambda S_i$ which can be expressed as $[B_{i+1}] = [B_i] + \dfrac{[\lambda S_i S_i^T]}{S_i^T g_i} - \dfrac{[B_i] g_i g^T [B_i]}{g_i^T [B_i] g_i}$

Algorithm for the Quasi Newton Method

1 Start with an initial point x_0 and a nxn positive definite symmetric matrix $[B_i]$ to approximate the inverse of the Hessian Matrix of f. Usually, the matrix is taken as the identity matrix [I].
Set the iteration number i=1.

2 Compute the gradient of the function, λf_i at the point X_i, and set $S_i = -[B_i]\lambda f_i$

3 Find the optimal step length , λ^* in the direction S_i and set $X_{i+1} = X_i + \lambda^* S_i$

4 Test the new point X_{i+1} for optimality. If X_{i+1} is optimal, terminate the iterative process , otherwise go to step 5 .

5 Update the Matrix $[B_i]$ using the formula $[B_{i+1}] = [B_i] + [M_i] + \{N_i]$ where

$$[M_i] = \lambda^* \frac{S_i S_i^T}{S_i^T g_i} \qquad and \qquad [N_i] = -([B_i] g_i ([B_i] g_i)$$

$$g_0 = \lambda f(X_{i+1}) - \lambda f(X_i) \equiv \lambda f_{i+1} - \lambda f_i$$

5 Set the new iteration number as i=i+1 and Go to step 2.

Consider the function $f(x) = \sum_{r=1}^{n-1} (x_r^2 - 1)^2 + (\sum_{r=1}^{n} x_r^2 - 0.25)^2, X_0 = (1,2,3, \ n)^T \quad for \quad n = 10$

$$f(x) = \sum_{r=1}^{n} (x_r - 1)^2 + (\sum_{r=1}^{n} x_r^2 - 0.25)^2, \ X_0 = (1,2,3,4,5,6,7,8,9,10)^T$$

$$f(x) = (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 + \ldots + (x_9 - 1)^2 + (x_1^{\ 2} + x_2^{\ 2} + x_3^{\ 2} + \ldots + x_9^{\ 2} - 0.25)^2$$

Results and Analysis

The following results are obtained from the algorithm.

x	V_1	V_2	V_3	V_4
X_1	0.052195689	0.348256789	0.328933520	0.310096800
X_2	0.103156640	0.355101161	0.341819912	0.325623036
X_3	0.1541236380	0.361945532	0.354706304	0.341149270
X_4	0.2050876130	0.368789905	0.367592696	0.356675499
X_5	0.2560515870	0.375634277	0.380479089	0.372201728
X_6	0.3070155620	0.382478649	0.393365481	0.387727957
X_7	0.3579795360	0.389323021	0.406251873	0.403254185
X_8	0.4089435110	0.396167393	0.419138265	0.418780413
X_9	0.4599074850	0.403011765	0.432024658	0.434306648
X_{10}	0.521956890	0.248214942	0.18973481	0.199011046
f_i	5.727319639	4.687078259	4.671672096	4.665980990

Fig 1

The number of iteration here is 4 and the time taken is 0.02 seconds.

Consider the function $f(x)\sum_{r=1}^{n}(x_r - 1)^4,\ x_0 \quad for\ \ n = 10$

$$f(x) = \sum_{r=1}^{n}(x_r - 1)^4, \quad x_0 = (2,2,2.\ 2,\ 2,\ 2,\ 2,\ 2,2,2,)^T$$

$$f(x) = (x_1 - 1)^4 + (x_2 - 1)^4 + (x_3 - 1)^4 + (x_4 - 1)^4 + (x_5 - 1)^4 + (x_6 - 1)^4 + (x_7 - 1)^4 + (x_8 - 1)^4 +$$
$$(x_9 - 1)^4 + (x_{10} - 1)^4$$

The following results are obtained from the above function

x	V_1	V_2	V_3	V_4	V_5	V_6	V_7
X_1	1.66666	1.44444	1.29629	1.19753	1.13168	1.08779	1.05852
X_2	1.66666	1.44444	1.29629	1.19753	1.13168	1.08779	1.05852
X_3	1.66666	1.44444	1.29629	1.19753	1.13168	1.08779	1.05852
X_4	1.66666	1.44444	1.29629	1.19753	1.13168	1.08779	1.05852
X_5	1.66666	1.44444	1.29629	1.19753	1.13168	1.08779	1.05852
X_6	1.66666	1.44444	1.29629	1.19753	1.13168	1.08779	1.05852
X_7	1.66666	1.44444	1.29629	1.19753	1.13168	1.08779	1.05852
X_8	1.66666	1.44444	1.29629	1.19753	1.13168	1.08779	1.05852
X_9	1.66666	1.44444	1.29629	1.19753	1.13168	1.08779	1.05852
X_{10}	1.66666	1.44444	1.29629	1.19753	1.13168	1.08779	1.05852
f_1	1.97530	0.39018	0.07707	0.01522	0.00300	0.00059	0.00011

Fig 2

Analysis

From Fig 1, it shows that as values of X_i increases from 0.01 upto 0.52 and across for V_i increases across then the values of the function f_i decreases which mean the function is converging to its limit.

From the Fig 2, it shows that the values of X_i remain constant as that of V_i, also indicating that the function f_i decreases from 1.97530 to 0.00011 meaning that the function is converging to its limit.

Conclusion

The Quassi Method for solving nonlinear optimization problem which was derive using Taylor's series expansion by selecting considering a matrix to approximate the inverse of the Hessien Matrix of the function to be optimized. However, the approximate matrix selected is updated at each iteration. The search direction, S_i at the i^{th} iteration was computed and is multiplied by the step length. The new point is then obtained by the sum of the previous point and the result is multiplied by the step length and search direction. This process is continued until convergence is reached and this method is easy to handle than others. The Quassi Method uses less time and less number of iterations.

Reference

[1]. Deufhard. P. (2010): The History of Newton's Methods. Informatics Techlink, 55-83.
[2]. Fetcher.R. (1987): Practical Methods of Optimization . International Journal of Optimization Theory, 100-170.
[3]. Forsgren. A, Gill. P.E and Wright. M.H (2003): Interior Methods for Non Linear Optimization, SIAM 44(4) , 525-597.
[4]. Hanke. M. (1998): Conjugate Gradient type Method for ill-posed Problems, Pitman Research Notes in Mathematics Series, Vol 327.
[5]. Heyvan . A and Ashraf. G. (2010): A New Structured Quasi-Newton Algorithm using Partial Differentiation on Hessian Matrix. Journal of Computational and Applied Mathematics, 805=811.
[6]. Ogunniran. M.D (2014): A New Quasi-Newton Update of the Newton Iterative Method for Optimization of Non-Linear Multi-Variables Optimization Problems. International Journal of Science and Research, 46-59.
[7]. Optim. S.J. (2001): Reduced-Hessian Quasi-Newton Method for Unconstraint Optimization. Internal Journal of Science and Research, 209-350.
[8]. Peter. D. (2012): A Short Concepts of Newton's Method. Internal Journal of Computational and Applied Mathematics Extra Vol, 25-30.
[9]. Povale.Z. (2014): Quasi-Newton Method for Multi-Objective Optimization. Journal of Computational and Applied Mathematics, 765-777.
[10]. Richard . B, Micheal. B, Tony. C, James. D and June. D (1993): Templates for the System of Linear Systems, Building Blocks for Iterative Methods. SIAM.
[11]. Shi.Z.J.(225): Convergence of Quasi-Newton's Method with New Exact Line Search. Mathematical Analysis and Computation, 120-135.
[12]. Weerakoon. R and Simon. J. (2000); A New Derivation of the Classical Newton's Method using Rectangular Rule to Compute Integrals. Applied Mathematics Lettres 13(8), 87-93.
[13]. Xiaowei.J and Yueting. Y. (2010): A Seft-Scaling Quasi-Newton Method for Large Scale Unconstraint Optimization. Journal of Information and Computational Science, 7:8, 1939-1745.
[14]. Zwei. B. (1998): New Quasi-Newton Methods for Unconstraint Optimization Problems. Journal of Applied Mathematics, 62.

Adamu Wakili. "Quasi- Newton Method for Solving Non- Linear Optimization Problems of Convergence Functional." IOSR Journal of Mathematics (IOSR-JM) 15.4 (2019): 42-47