EXPERIMENTAL STUDY OF THE BROYDEN CLASS UPDATING METHOD FOR SOLVING NON-LINEAR UNCONSTRAINED OPTIMIZATION PROBLEMS

B. I. Oruh,[1] B. O. Osu[2] and M. K. Ahamefule[3]
[1, 2, 3]*(Department of Mathematics, Michael Okpara University of Agriculture Umudike, Nigeria)*

Abstract
The main purpose of this paper is to seek for a suitable combination of the most successful updates-the DFP and the BFGS-and use the combined updates in solving Non-Linear optimization problems. This combined updates belongs to what is referred to in literature as the Broyden updates which is family of quasi Newton methods that depend on a real valued parameter. Its Hessian approximation update formula is $H^{\phi} = (1-\phi)H^{DFP} + \phi H^{BPGS}$ *where* H^{BFGS} *stands for the update obtained by the Broyden Fletcher Goldfarb Shanno (BFGS) method,* H^{DFP} *for the update of the Davidon Fletcher Powell (DFP) method and* $\emptyset \in \mathbb{R}$. *The simulation results shows that as* $\emptyset \to 0$ *both the average time for execution and number of iterations reduces. The value* $\emptyset = 0.3$ *is thus recommended as a suitable value for the parameter* $\emptyset$ *in the convex combination of the BFGS and DFP to enhance the convergence of the quasi-Newton method.*

Keywords: *Broyden class updating technique, DFP, BFGS, convex combination.*

1. Introduction

The basic unconstrained optimization problem can be expressed as

$$\text{minimize } f(x), \qquad x \in \mathrm{R}^n \tag{1.1}$$

where $\mathbb{R}^n$ is an n-dimensional Euclidean space and $f: \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable function on $\mathbb{R}^n$ (i.e. f is smooth).

The solution to (1.1) can be obtained by analytical procedures or by numerical approach. For numerical approach to obtaining the solution, many methods abound which many authors (Nocedal and Yuan, 1998, Li *et al*, 2015, Gertz, 2004, Dennis (Jr.) and Schnabel, 1983, Burdakov *et al*, 2017) have done some researches on.

Amongst the methods of solution are steepest (gradient) descent method, Newton method and quasi-Newton method. Gradient descent method is a way to find a local minimum of a function. The way it works is that, we start with an initial guess of the solution and we take the gradient of the function at that point. We step the solution in the negative direction of the gradient and we repeat the process. The algorithm will eventually converge where the gradient is zero (which corresponds to a local minimum). Its brother, the gradient ascent, finds the local maximum nearer the current solution by stepping it towards the positive direction of the gradient. They are both first-order algorithms because they take only the first derivative of the function. Newton's method for solving unconstrained problems stems from the Newton's method for solving systems of nonlinear equations. Newton's method uses curvature information to take a more direct route, that is, can identify better search directions than can be obtained via the gradient method. Quasi-Newton methods are arguably the most effective methods for finding a minimizer of a smooth nonlinear function when second derivatives are either unavailable or too expensive to calculate. Quasi-Newton methods build up second-derivative information by estimating the curvature along a sequence of search directions. Each curvature estimate is installed in an approximate Hessian by applying a rank-one or rank-two update.

In the quasi-Newton context, the availability of an explicit basis for the gradient subspace makes it possible to represent the approximate curvature in terms of a reduced approximate Hessian matrix with order at most $k + 1$. Quasi-Newton algorithms that explicitly calculate a reduced Hessian have been proposed by Fenelon (1981) and Nazareth (1986), who also considered modified Newton methods in the same context. Siegel (1992) proposed methods that work with a reduced inverse approximate Hessian. In practical terms, the reduced-Hessian formulation can require significantly less work per iteration when k is small relative to n. This property can be exploited by forcing iterates to linger on a manifold while the objective function is minimized to greater accuracy. While iterates linger, the search direction is calculated from a system that is generally smaller than the reduced Hessian. In many practical situations, convergence occurs before the dimension of the lingering subspace reaches n, resulting in substantial savings in computing time.

More recently, Siegel (1994) proposed the conjugate-direction scaling algorithm, which is a quasi-Newton method based on a conjugate-direction factorization of the inverse approximate Hessian. Although no explicit reduced Hessian is updated, the method maintains a basis for the expanding subspaces and allows iterates to linger on a manifold. The method also has the benefit of finite termination on a quadratic function (Leonard, 1995). More importantly, Siegel's method includes a feature that can considerably enhance the benefits of lingering. Siegel notes that the search direction is the sum of two vectors: one with the scale of the estimated derivatives and the other with the scale of the initial approximate Hessian. Siegel suggests rescaling the second vector using newly computed approximate curvature. Algorithms that combine lingering and rescaling have the potential for giving significant improvements over conventional quasi-Newton methods.

In recent years, a class of iterative processes for solving non-linear equation or minimization problem has been considered frequently and applied to many practical problems. Its members are variously called quasi-Newton methods, variable metric methods, modification methods, or update methods. Most of the studies about these methods were framed in a minimization setting. In this setting, Powell (1970 and 1978) proved a global convergence theorem for one of the oldest known member of this class, the Davidon Fletcher Powell (DFP) method. Only recently, Dennis and More (1977), in continuation of earlier work by Dennis (1970 and 1971) developed highly interesting local convergence results for a class of update methods without assuming the minimization setting or requiring specific relaxation factors.

One of the most successful updates is the Broyden- Fletcher- Goldfarb- Shanno (BFGS) formula(Wen etal,2014), which is a member of the wider Broyden class of rank-two updates. Despite the success of these methods on a wide range of problems, it is well known that conventional quasi-Newton methods can require a disproportionately large number of iterations and function evaluations on some problems. This inefficiency may be caused by a poor choice of initial approximate Hessian or may result from the search directions being poorly defined when the Hessian is ill-conditioned (Philip and Michael, 2001).

The main purpose of this paper is to seek for a suitable combination of the most successful updates-the DFP and the BFGS-and use the combined updates in solving NL optimization problems. This combined updates belongs to what is referred to in literature as the Broyden updates, which is family of quasi Newton methods that depend on a real valued parameter. Its Hessian approximation update formula is $H^{\phi} = (1-\phi)H^{DFP} + \phi H^{BPGS}$, where H^{BFGS} stands for the update obtained by the Broyden Fletcher Goldfarb Shanno (BFGS) method, H^{DFP} for the update of the Davidon Fletcher Powell (DFP) method and $\emptyset \in \mathbb{R}$. In this case, all members of the Broyden class satisfy the well known secant equation, $H_{k+1}q_i = P_i$ central to many quasi Newton method. However, a suitable value for the parameter $\emptyset$ that will enhance the convergence of the quasi-Newton method is not known in literature to have been determined. In this paper, we first determine a suitable $\emptyset$ for the convex combination of DFP and BFGS.

This work was carried out with the aid of the following test functions: Freudenstein and Roth, Beales and Woods. The functions are represented mathematically as follows.

Freudenstein and Roth function

$$F(x) = (-13 + x_1 + ((5 - x_2) * x_2 - 2) * x_2)^2 + (-29 + x_1 + ((x_2 + 1) * x_2 - 14) * x_2)^2.$$

Beales function

$$F(x) = (1.5 - x_1 * (1 - x_2)^2 + (2.25 - x_1 * (1 - x_2^2))^2 + (2.625 - x_1 * (1 - x_2^3))^2.$$

$$x^0 = (1,1) \;;\; x^* = (3,0.5)$$

Woods function

$$F(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10(x_2 + x_4 - 2)^2 + 0.1(x_2 - x_4)^2.$$

$$x^0 = (-3,-1,-3-1); \;\; x^* = (1,1,1,1)$$

2. Methodology

In this section the algorithms for DFP, BFGS and their convex combination of DFP and BFGS for the minimization problem are presented.

2.1 Davidon Fletcher Powell Method

One of the most schemes for constructing the inverse Hessian, was originally proposed by Davidon and later developed by Fletcher and Powell.

For a quadratic objective, it simultaneously generates the directions of the conjugate gradient method while constructing the inverse Hessian. At each step, the inverse Hessian is updated by the sum of two symmetric rank one matrices, and this scheme is therefore often referred to as a rank two correction procedure. The method is also often referred to as the variable matric method, the name originally suggested by Davidon.

Algorithm 2.1 DFP (Nocedal and Yuan (1998))

Step 1: Given starting point x_0, convergence tolerance ε >0, inverse Hessian approximation H_0; $K \to 0$,

Step 2: Compute search direction $d_k = -H_k g_k or - H_k \nabla f(x_k)$

Step 3: Calculate the step length $\propto_k$ that is

$\propto_k$ = arg min $f(x_k + \propto d_k)$ with respect to $\propto \geq 0$

To obtain $x_{k+1} = x_k + \propto_k d_k$

Step 4: Compute the difference $q_k = g_{k+1} - g_k and \; p_k = \propto_k d_k$

Step 5: Update H_k by the formula to obtain H_{k+1}

That is;

$$H_{k+1} = H_k + \frac{P_k P_k^T}{P_k^T q_k} - \frac{H_k q_k q_k^T H_k}{q_k^t H_k q_k}. \quad (2.1)$$

Step 6: Set $k = k+1$ and go to step 2.

This shows that the DFP algorithm is a quasi-Newton method, in the sense that when applied to quadratic problems, we have $H_{k+1} q_i = P_{i,} \quad 0 \leq i \leq k.$

The DFP algorithm is superior to the rank one algorithm in that it preserves the positive definiteness of H_k. However, it turns out that in the case of larger non quadratic problems the algorithm has the tendency of sometimes getting "Stuck". We discuss an algorithm that alleviates this problem

2.2 Broyden Fletcher Goldfarb Shannon

An alternative update formula was suggested independently by Broyden, Fletcher, Goldfarb and Shannon in 1970. The Broyden Fletcher Goldfarb Shannon (BFGS) algorithm is an iterative method for solving unconstrained nonlinear optimization problems. BFGS method approximate Newton's method, a class of hill climbing optimization techniques seeks a stationary point of a (preferably twice continuously differentiable) function for such problems, a necessary condition for optimality is that the gradient be zero.

Algorithm 2. 2 BFGS (Adewale and Oruh, 2013)

Step 1 Given starting point x_o, convergence tolerance $\varepsilon > 0$, inverse Hessian approximation $B_0, k \to 0$ while $// \nabla f(x_k) // > \varepsilon$

Step 2: compute search direction $d_k = -B_k g_k \; or - B_k \nabla f(x_k)$

Step 3: Calculate the step length α_k, that is set $x_{k+1} = \alpha_k d_k$, where α_k is step length d_k is search direction.

Step 4: Compute the difference $q_k = g_{k+1} - g_k and \; p_k = \propto_k d_k$

Step 5: Update B_k by the formula to obtain B_{k+1} that is

$$B_{k+1} = B_k + \frac{q_k q_k^T}{q_k^T P_k} - \frac{B_k P_k P_k^T B_k}{P_k^T B_k S_k} \quad (2.2)$$

Step 6: Set $k = k+1$ and go to step 2.

2.3 Linear Combination of Dfp And Bfgs

Both the DFP and the BFGS update have symmetric rank two correction that are constructed from the vectors P_k and $H_k q_k$. This leads to collection of updates, known as the Broyden family defined by

$$H^{\phi} = (1-\phi)H^{DFP} + \phi H^{BFGS}$$

Where ϕ is a parameter that may take any real value

Clearly ϕ = 0 and ϕ =1 yield the DFP and BFGS updates respectively. The Broyden class is a family of updated specified by the formula

$$H_{k+1}^{\phi} = H_k + \frac{P_k P_k^T}{P_k^T q_k} - \frac{H_k q_k q_k^T H_k}{q_k^T H_k q_k} + \phi V_k V_k^T = H_{k+1}^{DFP} + \phi V_k V_k^T$$

Where

$$V_k = \left(q_k^T H_k q_k\right)^{1/21}\left(\frac{P_k}{P_k^T q_k} - \frac{H_k q_k}{q_k^T H_k q_k}\right)$$

The Broyden method is a quasi-Newton method in which at each iteration, a member of the Broyden family is used as the updating formula. The parameter ϕ is allowed to vary from one iteration to another. A sequence ϕ_1, ϕ_2, ……, of parameter values is Broyden method. A pure Broyden method is one that uses a constant ϕ. Since H^{DFP} and H^{BFGS} satisfy the fundamental relation that is $H_{k+1} q_i = p_i$ for updates, this relation is also satisfied by all members of the Broyden family. Therefore many properties that were found to hold for the DFP method will also hold any Broyden methods.

Algorithm 2.3 Convex Combination of DFP and BFGS

Step 1 Given starting point x_o, convergence tolerance $\varepsilon > 0$, inverse Hessian approximation $B_0, k \to 0$ while $// \nabla f(x_k) // > \varepsilon$

Step 2: compute search direction $d_k = -B_k g_k \; or - B_k \; \nabla f(x_k)$

Step 3: Calculate the step length α_k, that is set $x_{k+1} = \alpha_k d_k$, where α_k is step length d_k is search direction.

Step 4: Compute the difference $q_k = g_{k+1} - g_k \, and \;\; p_k = \propto_k d_k$

Step 5: Update B_k by the formula to obtain B_{k+1} that is

$$H^{\phi} = (1-\phi)H^{DFP} + \phi H^{BFGS}$$

Step 6: Set $k = k+1$ and go to step 2.

2.4 Material and Method

The algorithms for DFP, BFGS and the convex combination of DFP and BFGS for minimization problem would be presented. Simulations of the DFP, BFGS and their convex combinations at points 0.1 to 0.9 would be carried out in MATLAB environment for the following three commonly test functions: Freudenstein and Roth function, Beales function and Woods function. The performance of the algorithm would be based on the average time and number of iterations needed to reach the minimum value of the functions. Algorithm with least average time and least number of iterations would be considered most efficient. A tolerance of $\varepsilon = 10^{-5}$ would be set as the stopping criteria. Graphs would be plotted in both excel and MATLAB.

3.0 Numerical Examples

This section presents the results obtained in the simulations of the Davidon Fletcher Powell (DFP), Broyden Fletcher Goldfarb Shannon (BFGS) and their convex combinations at points 0.1 to 0.9 in MATLAB environment for a set of test problems from CUTE collections established by (Bongartz *et al,* 2003).

The numerical results produced by implementing the algorithm to the test functions are presented in the Table 4.1-4.5. The efficiency of the algorithm is based on the average time and number of iterations needed to reach the minimum value of the functions. Algorithm with least average time and least number of iterations is considered most efficient. A tolerance of
$\varepsilon = 10^{-5}$ is set as the stopping criteria.

3.1 Matlab Simulation Result

The following shows the result of the simulations for the tested functions in MATLAB environment.

Quadratic function

$$F(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2, \; x^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}; \; x^* = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

Result generated using DFP (Quadratic Function)

n	*a*	f0	X(1,1)	X(2,1)	(norm)b
0	0.0557	74.0000	0	0	50.9902
1.0000	0.4998	1.5797	1.8941	2.1169	2.5172
2.0000	0.4998	1.5797	1.0000	3.0000	0.0000
Average time = 1.041307					

Result generated using (BFGS)

n	*a*	f0	X(1,1)	X(2,1)	(norm)b
0	0.0557	74.0000	0	0	50.9902
1.0000	0.4998	1.5797	1.8941	2.1169	2.5172
2.0000	0.4998	1.5797	1.0000	3.0000	0
Average time = 1.020159.					

Table 4.1: Average time for the test functions

	Test Functions		
Initial Point	**Freudenstein and Roth**	**Beales**	**Woods**
DFP	4.7319	7.1991	11.3351
BFGS	3.9507	5.2600	9.2369
LC (0.1)	4.9542	6.4844	11.3368
LC (0.2)	4.5686	6.5703	11.8795
LC (0.3)	4.9542	7.1135	12.0581
LC (0.4)	4.5686	7.6623	12.9340
LC (0.5)	4.7517	7.9661	14.9944
LC (0.6)	5.5238	8.5648	14.9130
LC (0.7)	5.9350	9.0692	15.6826
LC (0.8)	6.6464	10.6435	19.2377
LC (0.9)	7.8094	12.4352	24.6682

Table 4.5: Number of iterations for the test functions

	Test Functions		
Initial Point	**Freudenstein and Roth**	**Beales**	**Woods**
DFP	5	8	39
BFGS	5	8	39
LC (0.1)	6	8	40
LC (0.5)	6	12	57
LC (0.6)	9	13	57
LC (0.7)	11	14	68
LC (0.8)	13	18	90
LC (0.9)	17	20	118

3.2 Graphical Analysis of Results

The following shows the graphical analysis of the simulation results.

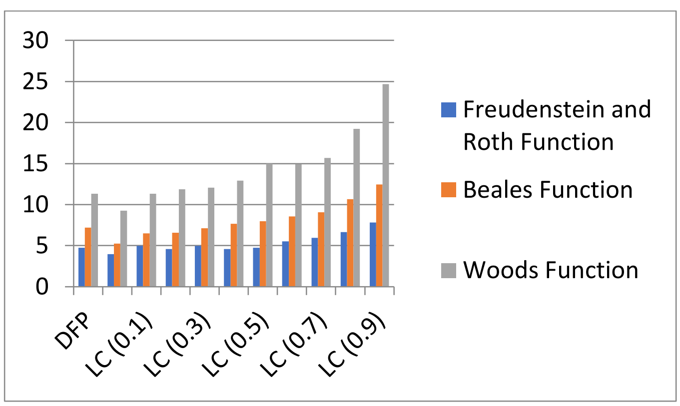

Figure 4.1a: Excel Graph of Average Time of Test Functions

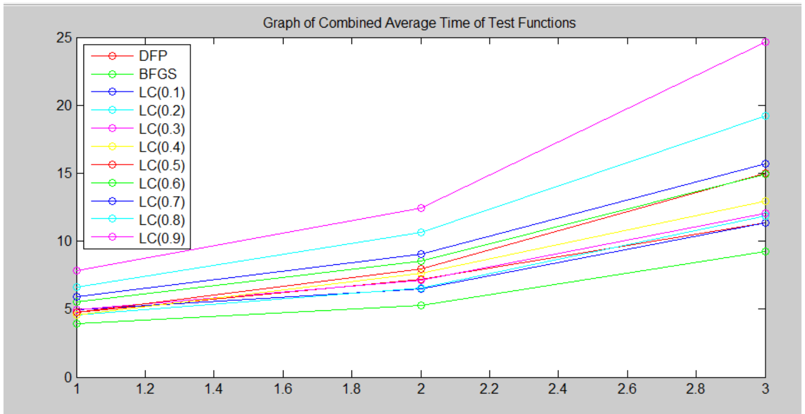

Figure 4.1b: MATLAB Plot of Average Time of Test Functions

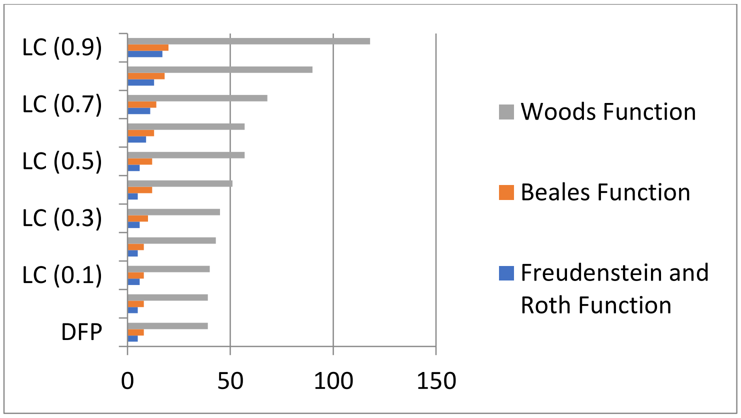

Figure 4.2a: Excel Graph of Number of Iterations of Test Functions

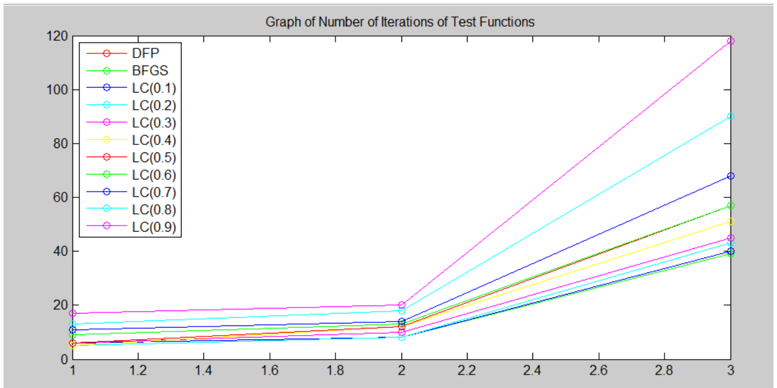

Figure 4.2b: MATLAB Plot of Number of Iterations of Test Functions

3.3 Discussions of Results

Items 1, 2 and 3 in the horizontal axes of the MATLAB plots in Figure 4.1b and 4.2b represent the test functions Freudenstein and Roth, Beales and Woods respectively.

It is evident from Table 4.4, Figures 4.1a and 4.1b that BFGS algorithm gives the best performance in terms of the average time it took the test functions to generate result using the initial guesses of (0.5, -2), (1, 1) and (-3, -1, -3, -1) for Freudenstein and Roth, Beales and Woods functions respectively. It is the fastest of all the algorithms with average time of 3.9507, 5.2600 and 9.2369 for the functions accordingly.

Also, Table 4.4 and Figure 4.1b show that the average time of the test functions increase as the linear constant increases for the Linear Combination (LC) algorithm. Moreover, LC (0.9) gives the greatest average time of 7.8094, 12.4352 and 24.6682 for Freudenstein and Roth, Beales and Woods functions respectively as depicted by Figure 4.1b.

Considering the number of iterations, it is clear from Table 4.5, Figures 4.2a and 4.2b that both DFP and BFGS give the same and best results. The two methods produce the least number of iterations. 5, 8 and 39 iterations were generated by DFP and BFGS for the three test functions; Freudenstein and Roth, Beales and Woods functions respectively as shown in Figures 4.2a and 4.2b. The result of the number of iterations for the LC algorithm is similar to what obtains for average time with LC (0.9) having the highest number of iterations of 17, 20 and 118 for Freudenstein and Roth, Beales and Woods functions respectively as shown in Table 4.5, Figures 4.2a and 4.2b.

The results as illustrated in the Excel graphs and MATLAB plots show that Freudestein and Roth function performs better than Beales function. Woods function gives the least performance of the three tested functions.

4.1 Conclusion

In this study we have attempted to determine the best method for solving unconstrained minimization problem with the best result. We considered only three commonly tested functions namely, Freudenstein and Roth, Beales and Woods functions. The result of the simulations in MATLAB environment and the graphical analysis revealed that BFGS formula gives the best performance as it gives the same convergence rate but better average time than DFP in all the test functions. The average time and the number of iterations of the test functions increase as the linear constant increases for the Linear Combination (LC) algorithm. It can therefore be concluded that BFGS is superior to DFP and the LC algorithm at any linear constant for solving unconstrained minimization problem.

References

Adewale, T. A and Oruh, B. I (2013). "A theoretical and experimental study of the Broyden-Fletcher-Goldfarb-Shano (BFGS) update", *African Journal of Mathematics and Computer Research.* Vol. 6, No. 8, pp. 666-668.

Burdakov, O., Gong, L., Zikrin, S. and Yuan, Y. (2017). On Efficiently Combining Limited-Memory and Trust-region Techniques, *Journal of Mathematical Programming Computation.* 9(1).

Dennis, J. E. (Jr.) and Schnabel, R. B. (1983). Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice-Hall, Inc. Englewood Cliffs, New Jersey.

Dennis, J. E and More, J. J (1977). "Quasi-Newton methods, motivation and theory" *SIAM Rev*. 19: 46-89. DOI: 10.1137/1019005.

Dennis, J. E. (1970). "On the Convergence of Newton-Like Methods"*Numerical Methods for Nonlinear Algebraic Equations*, Gordon and Breach, London, 1970.

Dennis, J. E (1971). "On the Convergence of Broyden's Method for Nonlinear Systems of Equations" *Mathematics of Computation*, Volume 25, No. 115, 1971.

Fenelon, M. C (1981). "Preconditioned Conjugate-Gradient-Type Methods for Large-Scale Unconstrained Optimization", Ph.D. thesis, Department of Operations Research, Stanford University, Stanford, CA.

Gertz, M. (2004). A Quasi-Newton Trust-Region Methods. *Journal of Mathematical Programming*. Springer-Verlag New York, vol. 100(3): 447-470.

Gill, P. E and Leonard, M. W (1997). "Limited-Memory Reduced-Hessian Methods for Unconstrained Optimization", Numerical Analysis Report NA 97-1, University of California, San Diego, CA.

J. Nocedal and Y. Yuan (1998). 'Combining trust region and line search techniques' *Advances in nonlinear programming,* pp. 153-175

Leonard, M. W (1995). "Reduced Hessian Quasi-Newton Methods for Optimization", Ph.D. thesis, Department of Mathematics, University of California, San Diego, CA.

Nazareth, J. L (1986). "The method of successive affine reduction for nonlinear minimization", *Math. Programming,* No. 35, pp. 97-109.

Li, C., Zhou, Q. and Wu, X. (2015). A Non-Monotone Trust-region Method with Non-Monotone Wolfe-Type Line Search Strategy for Unconstrained Optimization. *Journal of Applied Mathematics and Physics,* 3(6): 300-306.

Philip and Michael (2001). "Reduced-Hessian Quasi-Newton Methods for Unconstrained Optimization" *Siam Journal of Optimization*. Society for Industrial and Applied Mathematics Vol. 12, No. 1, pp. 209-237.

Powell, M. J. D (1970). "A Hybrid Method for Nonlinear Equations"*Numerical Methods for Nonlinear Algebraic Equations,* Gordon and Breach, London.

Powell, M. J. D (1987). "Updating conjugate directions by the BFGS formula". *Math. Programming*, Vol. 38, pp. 693-726.

Siegel, D (1992). "Implementing and Modifying Broyden Class Updates for Large Scale Optimization", Report DAMTP/1992/NA12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK.

Siegel, D (1994). "Modifying the BFGS update by a new column scaling technique", *Math. Programming,* No. 66, pp. 45-78.

Strang (2006). *Linear Algebra and Its Applications* (4th ed.). Brooks Cole. ISBN 0-03-010567-6.

Wen, H, Gallivan, K. A and Absil, P. A (2014). "A Broyden Class of Quasi-Newton Methods for Riemannian Optimization" Tech. Report FSU-15-04.