Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 3, 2026

AN EFFICIENT ITERATIVE FRAMEWORK FOR UNCONSTRAINED NONLINEAR OPTIMIZATION

Tazkia Anwar[1], Nudrat Aamir[2], Farah Jaffar[*3]

[1]*Department of Mathematics, Shaheed Benazir Bhutto Women University Peshawar, Pakistan*
[2] *Sarhad University of Science and Information Technology, Islamabad*
[*3]*Department of Basic Science and Humanities, CECOS University of IT and Emerging sciences, Peshawar, Pakistan*

DOI:https://doi.org/10.5281/zenodo.22809716

Keywords
Multi-step quasi newton method, Hessian, Unconstrained, Three-step accumulative technique, BFGS method

Article History
Received: 30 January 2026
Accepted: 15 March 2026
Published: 31 March 2026

Corresponding Author: *
Farah Jaffar

Abstract
In multi-step Quasi-Newton methods, to update Hessian at each step, interpolatory polynomial is used to derived information from more than one previous steps. To calculate interpolating points Ford and Moghrabi developed some techniques, Fixed and Accumulative method. In this paper, we use three step Accumulative technique with new scheming to update Hessian. The numerical results indicates successive performance with comparison to existing BFGS method.

INTRODUCTION:

Unconstrained optimization plays a critical role in modern AI, and researchers have made extensive contributions to its development. Moreover, AI techniques now have the potential to enhance every stage of the optimization process enabling more accurate models and unlocking innovative solutions to complex problems. In this paper, we are concern to solve nonlinear high dimensional optimization problem. Such as

$$minf(x) \tag{1}$$

Where $f(x)$ is continuous and twice differentiable. There are numerous iterative methods presents in literature to solve such optimization problems [1]. Among all these methods, the class of quasi Newton's methods considered best in case of large dimensional problems. These methods are variants of Newton's method. In newton method, we are to compute double derivative at each step. Which is become very hard to compute in case of high dimensions [9]. Where in quasi-Newton's methods double derivative (Hessian) $\nabla^2 f(x)$ is updated through suitable approximation [2]. BFGS formula is most popular and commonly used technique of this class [3]. Because of favorable numerical experience and fast theoretical Convergence [6-8]. In this method, the approximated Hessian matrix B_{i+1} is updated by the following formula,

$$B_{i+1} = B_i + \frac{y_i y_i^T}{y_i^T s_i} + \frac{B_i s_i (B_i s_i^T)}{s_i^T B_i s_i}. \tag{2}$$

and the inverse of Hessian is approximated by,

$$H^{-1} = \left(\mathrm{I} - \frac{y_i s_i^T}{y_i^T s_i}\right)^T H_i \left(\mathrm{I} - \frac{y_i s_i^T}{y_i^T s_i}\right) + \frac{s_i s_i^T}{y_i^T s_i}. \quad (3)$$

Where $s_i = x_{i+1} - x_i$ and $y_i = g_{i+1} - g_i$ are vectors. And $g_i = \nabla f(x_i)$ the gradient vector of f. The updated matrix B_{i+1} is need to satisfy the secant equation,

$$B_{i+1} s_i = y_i. \quad (4)$$

Furthermore, Ford and Moghrabi [4] introduced multi-step Quasi-Newton methods. Where they showed, how to update Hessian at each iteration by employing interpolating polynomials to derive accurate information from more than one previous steps. The updated Hessian approximation B_{i+1} in multi-step methods is acquire to satisfy the relation,

$$B_{i+1} r_i = w_i. \quad (5)$$

Where r_i and w_i are multi-step vectors, calculated through interpolating polynomial.

Multi-Step Methods

Recent advancements in multi-step methods, specifically two-step [12] and three-step [13] approaches, have demonstrated remarkable improvements over the conventional single-step Quasi-Newton method by employing diverse strategic frameworks.

In multi-step methods, the vectors r_i and w_i are calculated from available multi iterative points by defining a path X to be polynomial $x(\tau)$ of degree m.

Generalizing these vectors, let us suppose we have two current iterates x_k, x_{k+1} and $m-1$ most recent iterative points with their corresponding gradient values are available to us, such as, $x_{k-m+1}, x_{k-m+2}, \dots, x_{k-1}$. Thus, in total we have $m+1$ iterates. The interploting polynomial $x(\tau)$ can be define as,

$$x(\tau_k) = x_{i-m+k+1} \quad \text{for } k = 0, 1, \dots, m. \quad (6)$$

This polynomial depends on values of $[\tau_k]_{k=0}^m$. Initially, it is taken to be

$$\tau_k = k - m + 1 \quad \text{for } k = 0, 1, \dots, m. \quad (7)$$

The interpolating polynomial and its gradient can be approximated by Langrangian form of polynomial, i-e

$$x(\tau) = \sum_{k=0}^{m} L_k(\tau) x_{i-m+k+1} \quad (8)$$

$$g\big(x(\tau)\big) = \sum_{k=0}^{m} L_k(\tau) g(x_{i-m+k+1}) \quad (9)$$

Where $L_k(\tau)$ presents Langrangian form of polynomial

By differentiating (7) and (8), the required relation (4) can be satisfied as

$$r_i \triangleq \dot{x}(\tau) = \sum_{k=0}^{m} L_k'(\tau_m) x_{i-m+k+1} \quad (10)$$

$$w_i \triangleq g'\big(x(\tau)\big) \approx \sum_{k=0}^{m} L_k'(\tau_m) g(x_{i-m+k+1}) \quad (11)$$

Where $L_k'(\tau_m)$ can be explicitly define as

$$L_k'(\tau_m) = (\tau_k - \tau_m)^{-1} \prod_{\substack{j=0 \\ j \neq k}}^{m-1} \frac{(\tau_m - \tau_j)}{(\tau_k - \tau_j)} \quad \text{for } k \neq m.$$

And

$$L_m'(\tau_m) = \textstyle\sum_{j=0}^{m-1} (\tau_m - \tau_j)$$

Ford and Moghrabi showed [5] that r_i and w_i can be written in terms of most recent step vectors $\{s_{i-j}\}_{j=0}^{m-1}$ and $\{y_{i-j}\}_{j=0}^{m-1}$.

$$r_i = \sum_{j=0}^{m-1} s_{i-j} \sum_{k=m-j}^{m} L_k'(\tau_m)$$

$$w_i = \sum_{j=0}^{m-1} y_{i-j} \sum_{k=m-j}^{m} L_k'(\tau_m)$$

They have tested this algorithm for two and three steps. In this paper, we are using three steps technique.

In order to choose suitable values for $[\tau_j]_{j=0}^m$ Ford and Moghrabi [5] defined different approaches. Accumulative and Fixed point technique. In this paper, our focus is upon Accumulative technique. Where the distance between two iterates is define by θ_m with the any appropriate positive-definite matrix (M say)

$$\theta_m(y_1, y_2) = \{(y_1 - y_2) M (y_1 - y_2)\}^{1/2}$$

They have used different choices of M such as, $M = I, M = B_i$ and $M = B_{i+1}$. In our procedure , we used $M = B_i$.

Accumulative Technique

In this approach, one interpolating point is fixed as "0" and other points can be calculated through formula given bellow, accumulatively.

$$\tau_m = 0,$$
$$\tau_i = \tau_{i+1} - \theta_M(x_{i-m+j+2}, x_{i-m+j+1}) \quad \text{for } j = 0.1, \dots, m-1.$$

Proposed Framwork

We are considering an accumulative point method in three step so we will fix one of the step. Suppose we are taking $\tau_3 = 0$ at iterate x_{i+1} . Now previously researcher's Ford and Moghrabi [5] followed the following pattern i-e, When we are at x_{i+1} , we apply three step accumulative method to find B_{i+1}. At x_i we apply single step BFGS method to find B_i. Similarly at x_{i-1}, we use simple BFGS method to get the value of B_{i-1} by using B_{i-2}, which is approximation to original matrix and is usually taken as identity.

Now, in our proposed technique we follow following steps.

I. At x_{i-2} we have B_{i-2} (approximation to identity matrix).

II. When we are at x_{i-1}, we have values of B_{i-2}, s_{i-2}, y_{i-2} and x_{i-2}. So, by using these values we will find B_{i-1} with the help of single step BFGS method.

III. Now at iterate x_i, we have s_{i-2}, s_{i-1}, y_{i-2}, y_{i-1}, B_{i-2}, B_{i-1}, x_{i-2} , x_{i-1}. Here we will use two step accumulative method to update B_i .

IV. We are at x_{i+1} and also we know the values B_i, B_{i-2}, B_{i-1}, s_i, s_{i-1}, y_i, y_{i-2}, y_{i-1}, x_i x_{i-2}, x_{i-1}. Now to update B_{i+1} we will use three step accumulative technique.

Taking $B_{i-2} = \mathrm{I}$, the Hessian matrix B_{i-1} at the next iterate will be updated by the following formula.

$$B_{i-1} = B_{i-2} + \frac{y_{i-1}y_{i-1}^T}{y_{i-1}^T s_{i-1}} + \frac{B_{i-1}s_{i-1}(B_{i-1}s_{i-1}^T)}{s_{i-1}^T B_{i-1}s_{i-1}}.$$

Now at the third iterate x_i, we will use two step accumulative method. Therefore, we are to calculate here values of τ_2, τ_1 and δ. Such as

$$\tau_2 = \tau_3 - \theta_m(x_{i+1}, x_i) = -((x_{i+1} - x_i)^T B_i (x_{i+1} - x_i))^{\frac{1}{2}} = -\left(s_i^T B_i s_i\right)^{\frac{1}{2}} \quad (12)$$

$$\tau_1 = \tau_2 - \theta_m(x_i, x_{i-1}) = \tau_2 - ((x_i - x_{i-1})^T B_i (x_i - x_{i-1}))^{\frac{1}{2}} = \tau_2 - \left(s_{i-1}^T B_i s_{i-1}\right)^{\frac{1}{2}} \quad (13)$$

The matrix vector calculation in (10) and (11) is very expensive to compute. To circumvent this difficulty, let consider the search direction,

$$p_i = -B_i^{-1} g(x_i)$$

Where

$$x_{i+1} = x_i + \alpha_i p_i = x_i - \alpha_i B_i^{-1} g(x_i)$$

For some scalar $\alpha_i > 0$. This gives

$$x_{i+1} - x_i = -\alpha_i B_i^{-1} g(x_i)$$

Taking $s_i = x_{i+1} - x_i$. We have,

$$s_i = -\alpha_i B_i^{-1} g(x_i)$$
$$B_i s_i = -\alpha_i g(x_i)$$

This value $B_i s_i$ after substituting in (12). We obtain,

$$\tau_2 = -\left\{-\alpha_i s_i^T g(x_i)\right\}^{1/2}$$

The expression $s_{i-1}^T B_i s_{i-1}$ in (13) is still an expensive computation. To handle this situation Ford and Moghrabi [5] provide an approximation by replacing $i+1$ by i. i-e,

$$B_i s_{i-1} = y_{i-1} \quad (14)$$

Hence we can obtain the approximation

$$s_{i-1}^T B_i s_{i-1} = s_{i-1}^T y_{i-1}$$

Thus,

$$\tau_1 = -\left\{-\alpha_i s_i^T g(x_i)\right\}^{\frac{1}{2}} - \left(s_{i-1}^T y_{i-1}\right)^{1/2}$$

where,

$$\delta = \frac{(\tau_3 - \tau_2)}{(\tau_2 - \tau_1)}$$

Then,

$$r_i = s_i - \left\{\frac{\delta^2}{2\delta + 1}\right\} s_{i-1}$$
$$w_i = y_i - \left\{\frac{\delta^2}{2\delta + 1}\right\} y_{i-1}$$

The Hessian matrix B_i can be update by the formula,

$$B_i = B_{i-1} + \frac{w_{i-1}w_{i-1}^T}{w_{i-1}^T r_{i-1}} + \frac{B_{i-1}r_{i-1}(B_{i-1}r_{i-1}^T)}{r_{i-1}^T B_{i-1} r_{i-1}}$$

This approximation is acquire to satisfy the relation,

$$B_i r_i = w_i$$

As at fourth iterate x_{i+1}, we will perform three step accumulative approach. To follow algorithm we are to compute the following values,

$$\tau_2 = \tau_3 - \theta_m(x_{i+1}, x_i)$$
$$= -((x_{i+1} - x_i)^T B_i (x_{i+1} - x_i))^{\frac{1}{2}}$$
$$= -\left(s_i^T B_i s_i\right)^{\frac{1}{2}}$$
$$\tau_1 = \tau_2 - \theta_m(x_i, x_{i-1})$$
$$= \tau_2 - ((x_i - x_{i-1})^T B_i (x_i - x_{i-1}))^{\frac{1}{2}}$$
$$= \tau_2 - \left(s_{i-1}^T B_i s_{i-1}\right)^{\frac{1}{2}}$$
$$\tau_0 = \tau_1 - \theta_m(x_{i-1}, x_{i-2})$$
$$= \tau_1 - ((x_{i-1} - x_{i-2})^T B_i (x_{i-1} - x_{i-2}))^{\frac{1}{2}}$$
$$= \tau_1 - \left(s_{i-2}^T B_i s_{i-2}\right)^{\frac{1}{2}}$$

Here the computation of τ_0, can be found by using (14) i-e,

$$B_i s_{i-2} = y_{i-2}$$
$$s_{i-2}^T B_i s_{i-2} = s_{i-2}^T y_{i-2}$$

Thus,

$$\tau_1 = -\left\{-\alpha_i s_i^T g(x_i)\right\}^{\frac{1}{2}} - \left(s_{i-1}^T y_{i-1}\right)^{\frac{1}{2}} - \left(s_{i-2}^T y_{i-2}\right)^{1/2}$$

Where δ_1 and δ_2 are defined as,

$$\delta_1 = \frac{(\tau_3 - \tau_1)}{(\tau_1 - \tau_0)}, \quad \delta_2 = \frac{(\tau_3 - \tau_2)}{(\tau_2 - \tau_0)}$$

The three step quasi newton vectors are given as,

$$r_i = s_i + \left\{\frac{-\delta_1^2(\delta_2 + 1)^3}{(\delta_1 - \delta_2)(3\delta_1\delta_2 + \delta_1 + \delta_2)}\right\} s_{i-1} + \left\{\frac{(\delta_1\delta_2)^2}{(3\delta_1\delta_2 + \delta_1 + \delta_2)}\right\} s_{i-2}$$

$$w_i = y_i + \left\{\frac{-\delta_1^2(\delta_2 + 1)^3}{(\delta_1 - \delta_2)(3\delta_1\delta_2 + \delta_1 + \delta_2)}\right\} y_{i-1} + \left\{\frac{(\delta_1\delta_2)^2}{(3\delta_1\delta_2 + \delta_1 + \delta_2)}\right\} y_{i-2}$$

Now the new updated matrix B_{i+1} can be derived by formula,

$$B_{i+1} = B_i + \frac{w_i w_i^T}{w_i^T r_i} + \frac{B_i r_i (B_i r_i^T)}{r_i^T B_i r_i},$$

which is required to satisfy the relation,

$$B_{i+1} r_i = w_i.$$

As we are working on accumulative method, therefore, the notation "A" will be used for presentation of this basic method. In this paper, the new scheming accumulative approach will be presented by $A_3^{[1,2A]}$. Here the value in subscript shows three step method and the superscript values indicates new scheming approach i-e (the number 1 presents single step update and 2A indicates two-step accumulative technique that we have used to calculate interpolating points at third iterative point.)

Numerical Results

In this section we will discuss the numerical performance of our proposed method in comparison with single step BFGS method. We tested the proposed algorithm on two nonlinear optimization problems [10, 11], with four different starting points. Both problems are tested for variable dimensions $33 \leq n \leq 176$. Thus, in total we have 13 functions. Furthermore, these dimensions are divided in two sets, hard $(33 \leq n \leq 60)$ and very hard $(61 \leq n \leq 176)$. All the algorithms are implemented in MATLAB (R2009b).

The names of test problems with dimensions are listed in Table 1. In Table 2, we have presented the numerical performance of proposed method for set of hard dimensions in terms of iteration, function evaluation and time spend by each method. Similarly, Table 3 showed the numerical results for very hard set of dimensions.

Table 1: Test problems

Problems	Dimensions
Discrete ODE I function [11]	33, 44, 66, 88, 110, 132, 176
Discrete ODE II function [10]	42, 58, 78, 96, 114, 160

Table 2: Hard dimensions

Method	Iterations	Function evluations	Time (sec)
BFGS	2431	2451	0.5389
$A_3^{[1,2A]}$	1942	1966	0.4272

Table 3: Very hard dimensions

Method	Iterations	Function evluations	Time (sec)
BFGS	14075	14144	9.3839
$A_3^{[1,2A]}$	10953	11032	4.7434

Table 4: Combined dimensions

Method	Iterations	Function evluations	Time (sec)
BFGS	16506	16595	9.9228
$A_3^{[1,2A]}$	12895	12998	5.1706

Investigating, the numerical results of both methods, we can easily observe that, the new method winnig in all grounds in comparison to single step BFGS method for all dimensions. It computationally saves time and also produce less number of iterations and function evaluations than BFGS method.

Conclusion

We have presented multi-step BFGS method with new interpolating scheme. In this method, to update Hessian matrix, we use three step accumulative technique with different approach. The numerical results indicates, that the new approach is efficient and robust in function evaluations, number of iteration and time consumption .

References

[1] S. Wright, and J. Nocedal, Numerical optimization. Springer Science, 1999.
[2] P. E. Gill, W. Murray and M.H. Wright, Practical Optimization, Academic London, 1981.
[3] YH. Dai, Convergence properties of the BFGS algoritm, SIAM Journal on Optimization, vol. 13, no. 3, pp. 693-701, 2002.
[4] J. A. Ford and I.A. Moghrabi, Multi-step quasi-Newton methods for optimization, J. Comput. Appl. Math., 50 (1994) 305-323.
[5] J. A. Ford and I. A. Moghrabi, Alternative parameter choices for multistep quasi-newton methods, Optim. Methods Softw., 2 (1993), 357-370.
[6] D. H. Li and M. Fukushima, On the global convergence of the bfgs method for nonconvex unconstrained optimization problems, SIAM J. Optim., 11(2001), 1054-1064.
[7] Y. H. Dai, Convergence properties of the bfgs algoritm, SIAM J. Optim., 13 (2002), pp. 693-701.
[8] D. C. Sorensen, The q-superlinear convergence of a collinear scaling algorithm for unconstrained optimization, SIAM J. Numer. Anal., 17 (1980), 84-114.
[9] F. Modarres, M. A. Hassan, and W. J. Leong, Multi-steps symmetric rank-one update for unconstrained optimization, World Applied Sci. J., 7(2009), 610-615. [10] G. Alefeld and L. Platz□oder, A quadratically convergent krawczyk-like algorithm, SIAM J. Numer. Anal., 20 (1983), 210-219.
[11] W. Burmeister, Inversionsfreie verfahren zur l□osung nichtlinearer operatorgleichungen, Math. Mech., 52 (1972), 101-110.

[12] F. jaffar, W. K. Mashwani, S. M. Al-marzouki, N. Aamir, and M. Abiad, "Self decisive algorithm for unconstrained optimization problems as in biomedical image analysis," *Front. Comput. Neurosci.*, vol. 16, 2022.

[13] F. Jaffar, S. Ibrahim, R. Laila, Noureen, N. Aamir, and T. Hassan, "An innovative approach of engineering optimization," *Spectrum Emerg. Sci.*, vol. 3, pp. 461–467, 11 2025.