American Journal of Applied Mathematics
2026, Vol. 14, No. 5, pp. 307–330
https://doi.org/10.11648/j.ajam.20261405.15

Research Article

A Modified Algorithm for Broyden Family Using Natural Cubic Spline Interpolation Polynomial

Tarek Abou-El-Enien[1], Kamal El-Dib[2], Soher Mohamed[2, *]

[1]Operations Research and Decision Support Department, Cairo University, Giza, Egypt
[2]Mathematics Department, Fayoum University, Fayoum, Egypt

Abstract

The approximation of the objective function's second-derivatives matrix underlies the Broyden family (BF) of unconstrained optimization methods, and richer gradient information generally yields a more accurate approximation. This paper proposes a new optimization technique that replaces the traditional two-point, secant-based linear model of the gradient with a Natural Cubic Spline Interpolation Polynomial (NCSIP), constructed using either three points (M=2) or four points (M=3). The proposed method was implemented in MATLAB and tested against the traditional Broyden family method (M=1) on a set of standard unconstrained test problems across the range $\Phi \in [0, 1]$. The traditional method (M=1) recorded a total of 11564 iterations and 14867 function/gradient evaluations, while the four-point NCSIP model (M=3) achieved a clear efficiency improvement, with totals of 11063 iterations and 14082 function/gradient evaluations; the improvement achieved by the three-point model (M=2) was comparatively modest (11627 iterations and 14750 function/gradient evaluations). The best performance of the M=3 model was observed at higher values of the parameter Φ (near Φ=1), where it clearly outperformed the traditional method. The proposed method was also compared against the related Newton Divided Difference Interpolation (NDDI) method, using its corresponding three-point (M=4) and four-point (M=5) variants; the results showed a marginal numerical advantage of NCSIP over NDDI in total function/gradient evaluations when using four points (14082 vs. 14187). Taken together, these findings suggest that the number of gradient evaluations exploited, rather than the specific interpolation scheme, is the primary driver of efficiency gains. It should be noted that the algorithm's convergence properties are inferred from its algebraic reduction to the classical secant-based Broyden equation near the minimum, rather than established through a formal convergence proof.

Keywords

Unconstrained Optimization Methods, Broyden Family, Quasi-Newton Methods, Natural Cubic Spline Interpolation, Newton Divided Difference Interpolation

1. Introduction

Recent studies have developed various improvements to quasi-Newton methods from different perspectives. A. Hassan and A. R. Ayoob [1] proposed a new quasi-Newton equation based on cubic interpolation for unconstrained optimization. I. A. R. Moghrabi and B. A. Hassan [2] proposed an efficient limited-memory multi-step quasi-Newton method, with emphasis

*Correspondence: Soher Mohamed (sma37@fayoum.edu.eg)

Received: 23 August 2026; **Accepted:** 5 September 2026; **Published:** 24 September 2026

on reducing memory requirements and improving computational efficiency for large-scale optimization. Nosrati and Amini [3] proposed a new diagonal quasi-Newton method based on the weak secant equation and established its global convergence using the Armijo line search. K. K. Lai et al. [4] proposed a modified q-BFGS method using a new q-quasi-Newton equation and the q-Armijo–Wolfe line search, preserving global convergence without requiring a convexity assumption. N. Vater and Borzì [5] investigated a quasi-Newton method for systems of nonlinear underdetermined equations and established a semi-local convergence result for a damped method. B. A. Hassan and I. A. R. Moghrabi [6] proposed a modified secant relation based on a quadratic model to improve the approximation of the second-order curvature of the objective function, using both gradient and function values and establishing global convergence without a convexity assumption. Most recently, W. Zhang et al. [7] constructed a modified BFGS quasi-Newton method with a new secant equation and strong Wolfe line-search conditions, and derived local and superlinear convergence results. Taken together, these studies [1-7] demonstrate several approaches to improving quasi-Newton methods, including the construction of new secant or quasi-Newton equations, the use of interpolation or additional function information, limited-memory strategies, and the development of convergence results.

The problem we are thinking about is the following: we want to find a local minimum $X^* \in R^n$ of a differentiable function $F: R \rightarrow R$. It is assumed that for all X, the gradient g: $g: R^n \rightarrow R^n$ can be computed [8-10]

Unconstrained optimization problems such as Minimize f(X) can be solved by several methods; they can be broadly categorized depending upon whether derivative information f(X) is used or not. Search methods that rely only on function evaluations, like Nelder and Mead's simplex search, are best suited for nonlinear or discontinuous situations. Gradient methods are generally effective when the function to be reduced is continuous in its first derivatives [8-11]. Higher order methods, such as Newton's method, are only suitable provided the second order information is quickly and cheaply calculated because second order calculation by numerical differentiation is computationally expensive. Gradient methods use information about the slope of the function to select a search direction where the minimum is believed to be determined. When using gradient information, the quasi-Newton methods are the most effective. Newton-type algorithms employ a line search to find the local minimum after several iterations, while quasi-Newton algorithms calculate the Hessian matrix directly. This requires a substantial amount of computer time to compute the Hessian matrix. By constructing the curvature knowledge from the observed behavior of the objective function and its gradient and then using an appropriate updating formula for the Hessian matrix, quasi-Newton algorithms overcome this problem. Many update formulas for the Hessian matrix have been proposed. Among these, a very interesting family of updates is the Broyden family. These iterative algorithms approximate the Newton method without computing the second derivative [8, 9, 12] The sequence of iterations is defined by

$$X_{i+1} = X_i - \beta_i H_i^{-1} g_i \tag{1}$$

where β_i is a positive scalar selected to approximately minimize F along the search direction [8-10, 13]:

$$D_i = H_i^{-1} g_i \tag{2}$$

The matrix H (or its inverse B) is constructed to approximate the Hessian matrix of the objective function F (or its inverse), with the aim of mimicking Newton's method without the computational cost of directly evaluating the Hessian and thereby achieving a high rate of convergence [14-18].

Every iteration results in a new approximation by updating the matrix H:

$$H_{i+1}^{\phi} = H_i + \frac{z_i z_i^T}{z_i^T s_i} - \frac{H_i s_i s_i^T H_i}{s_i^T H_i s_i} + \phi_i u_i u_i^T \tag{3}$$

Where $\phi = \in [0.1]$ is a scalar parameter and

$$u_i = (s_i^T H_i s_i)^{1/2} \left(\frac{z_i}{z_i^T s_i} - \frac{H_i s_i}{s_i^T H_i s_i} \right) \tag{4}$$

The relationship (3) satisfies the secant (or quasi-Newton) equation [19, 20]:

$$H_{i+1} s_j = z_i \tag{5}$$

Where

$$s_i = X_{i+1} - X_i \tag{6}$$

$$z_i = g(X_{i+1}) - g(X_i) \tag{7}$$

The exact Hessian at X_{i+1} satisfies the Newton equation, which can be thought of as an approximation of the secant equation [21]:

$$G(X_{i+1})s_i = \beta_i \frac{dg(\beta)}{d\beta}|_{\beta=\beta_i} \tag{8}$$

where G represents the exact Hessian. Equation (5) is obtained as an approximation to (8) if the derivative is approximated by the backward difference $\frac{g_{i+1}-g_i}{\beta_i}$, This explains why the term secant equation is used. Any equation that is a reasonable approximation to (8) is a quasi-Newton equation, according to Ford and Saadallah [14]. There have been suggestions by a number of scholars for other approximations to (8) [18, 21-28]. This can be viewed as a way of developing different approximations to the derivative $\frac{dg}{d\beta}$, which are based on different theories about either F or g.

In this paper, we examine a modified minimization algorithm based on an alternative nonlinear model for approximating the Newton equation and the Broyden family formula. A comparison between the proposed Natural Cubic Spline interpolation (NCSIP) model and the classical Newton-based approach is also presented. It is worth noting that another established technique for approximating the gradient and function values in numerical optimization is the Newton Divided Difference Interpolation (NDDI) method [18, 29], which similarly builds on multiple gradient evaluations obtained during the line search. The present study differs from NDDI mainly in the interpolation mechanism: rather than a single global interpolating polynomial, we construct a natural cubic spline (NCSIP) through the same set of points. Since the spline is piecewise-defined and enforces continuity of the second derivative across segments, it may be less susceptible to the oscillatory behavior (Runge's phenomenon) that a single high-degree polynomial can sometimes exhibit as the number of interpolation points grows, particularly in the four-point model. A direct numerical comparison between NCSIP and NDDI to evaluate whether this structural difference translates into a measurable performance advantage.

2. Natural Cubic Spline Interpolation Polynomial Approach

The gradient is evaluated only twice in the standard Broyden class. If further evaluations of $g(X(\beta_i))$ are possible, a better approximation can be expected. Suppose g has been evaluated at several points $\left(X((\beta_i))\right)_{i=0}^{n}$, where n is a positive integer.

Based on equation (1), it is natural to consider a linear combination of the available values of g to approximate g(β). As a function of β, we consider the Natural Cubic Spline Interpolation polynomial (NCSIP) $g(\beta) = g(X_i + \beta\ D_i)$:

$$g(\beta) = g(X_i + \beta\ D_i) = \sum_{i=1}^{n} I_i S_i(\beta) \tag{9}$$

$$\mathrm{S_i}(\beta) = A_i M_{i-1} + B_i M_i + \frac{(\beta_i-\beta)y_{i-1}}{h_i} + \frac{(\beta-\beta_{i-1})y_i}{h_i} \tag{10}$$

where

$$A_{i=}\left(\frac{(\beta_i-\beta)^3}{6h_i} - \frac{h_i(\beta_i-\beta)}{6}\right) \tag{11}$$

$$\mathrm{B_{i=}}\left(\frac{(\beta-\beta_{i-1})^3}{6h_i} - \frac{h_i(\beta-\beta_{i-1})}{6}\right) \tag{12}$$

$$I_i = \begin{cases} 1 & \beta_{i-1} \le \beta \le \beta_i \\ 0 & otherwise \end{cases} \quad i = 1.2 \dots . n \tag{13}$$

$$\frac{h_i}{6} M_{i-1} + \left(\frac{h_i+h_{i+1}}{3}\right) M_i + \frac{h_{i+1}}{6} M_{i+1} = \frac{(y_{i+1}-y_i)}{h_{i+1}} + \frac{(y_i-y_{i-1})}{h_i} \tag{14}$$

Where

For $i = 1.2 \dots . n$

forms a system of $(n\ -1)$ linear equations in $(n+1)$ unknowns $\mathrm{M_0.M_1.\dots.M_n}$

$$\emptyset''(\beta_0) = S''_1(\beta_0) = M_0 = 0$$

$$\emptyset''(\beta_n) = S''_n(\beta_n) = M_0 = 0$$

Then

$$\frac{dg(\beta)}{d\beta} = \sum_{i=1}^{n} I_i S'_i(\beta) \tag{15}$$

where

$$S'_i(\beta) = -\frac{(\beta_i-\beta)^2}{2h_i} M_{i-1} + \frac{(\beta-\beta_{i-1})^2}{2h_i}\ M_i - \frac{(M_i-M_{i-1})}{6} h_i\ + \frac{(y_i-y_{i-1})}{h_i} \tag{16}$$

$$I_i = \begin{cases} 1 & \beta \in [\beta_{i-1}.\beta_i] \\ 0 & otherwise \end{cases} \quad i = 1.2 \dots . n \tag{17}$$

The approximation to the right hand side of Newton equation (8) can be written in the following form:

$$\omega = \beta_i \frac{dg(\beta)}{d\beta}|_{\beta=\beta_i} = \beta_n \sum_{i=1}^{n} I_i S'_i(\beta) \tag{18}$$

We shall know g_0 and g_1, if the first step in the search direction results in a satisfactory point. This is a good property, and the new model reduction to a linear model (i.e., we reduced to use the standard Broyden family). The reason is that the behavior of the objective function near the minimum is supposed to be like a quadratic function, which might be reached in a single step (often in one iteration, i.e., β = 1). Thus, the new algorithm has all the properties of the "secant-based" Broyden family of methods near the minimum.

The "Newton equation" can now be approximated as follows:

$$H_{i+1} s_i = \omega_i \tag{19}$$

The update of the Broyden family update method can be easily adjusted to take the new estimate of the vector ω into consideration by replacing the vector ω with the vector z. Thus, we can have

$$B_{i+1}^{\phi} = B_i + \frac{s_i s_i^T}{s_i^T \omega_i} - \frac{B_i \omega_i \omega_i^T B_i}{\omega_i^T B_i \omega_i} + \phi_i \nu_i \nu_i^T \tag{20}$$

Where

$$\nu_i = (\omega_i^T B_i \omega_i)^{1/2} \left(\frac{s_i}{s_i^T \omega_i} - \frac{B_i \omega_i}{\omega_i^T B_i \omega_i} \right) \tag{21}$$

Formulae B_{i+1}^{ϕ} maintains positive definiteness for any $\phi > \bar{\phi}$, where $\bar{\phi}$ is value which causes B_{i+1}^{ϕ} to be singular and is given by:

$$\bar{\phi} = \frac{1}{\left(1 - \frac{(\omega_i^T B_i \omega_i)(s_i^T H s_i)}{(s_i^T \omega_i)^2} \right)} \tag{22}$$

As long as the stability condition $s^T \omega > 0$ is satisfied, the Broyden family's update formulas (in which z is substituted with ω) ensure positive definite matrix H_{i+1}.

Different methods are derived according to the number of points obtained during the line search. The Newton equation (also known as the secant equation) is approximated using ω=z if we have only two points are obtained (i.e., n=1).

2.1. Three Points Model

In this model, we employ three points in NCSIP to approximate $g(\beta)$. (15), (16), (17) and (18), we have

$$\omega = \beta_2 \sum_{i=0}^{2} g_i B'_i(\beta_2) \tag{23}$$

$$B'_0(\beta_2) = \frac{\beta_2 - \beta_1}{C_0(\beta_2)} \tag{24}$$

$$B'_1(\beta_2) = \frac{-1}{C_1(\beta_2)} - \frac{\beta_2 - \beta_1}{C_0(\beta_2)} - \frac{1}{C_2(\beta_2)} \tag{25}$$

$$B'_2(\beta_2) = \frac{1}{C_1(\beta_2)} + \frac{1}{C_2(\beta_2)} \tag{26}$$

Where

$$C_0(\beta_2) = 2(\beta_1 - \beta_0)(\beta_2 - \beta_0) \tag{27}$$

$$C_1(\beta_2) = 2(\beta_2 - \beta_0) \tag{28}$$

$$C_2(\beta_2) = (\beta_2 - \beta_1) \tag{29}$$

The stability criterion *condition* $s^T \omega > 0$ is checked in each iteration; if it is not satisfied, ω=z is used for that iteration.

In selecting which points to use, factors such as the need to ensure that n is reasonably small so that the calculation of ω is not excessively computationally intensive, the preference for points on both sides of β_n [30] and the need to ensure that the points are sufficiently separated. More specifically, there was never a relative difference of less 10^{-4} between any two points used. If a pair of points was received that did not meet this criterion, one of the pairs was rejected. This meant that fewer than the required number of points were used to obtain ω on occasion.

2.2. Four Points Model

In this model, we employ four points in NCSIP to approximate $g(\beta)$. (15), (16), (17) and (18), we have

$$\omega = \beta_3 \sum_{i=0}^{3} g_i B'_i(\beta_3) \tag{30}$$

$$B'_0(\beta_3) = \frac{-(\beta_2 - \beta_1)(\beta_3 - \beta_2)}{C_0(\beta_3)} \tag{31}$$

$$B'_1(\beta_3) = \frac{(\beta_3 - \beta_2)}{C_1(\beta_3)} + \frac{(\beta_2 - \beta_1)(\beta_3 - \beta_2)}{C_0(\beta_3)} + \frac{(\beta_2 - \beta_0)(\beta_3 - \beta_2)}{C_2(\beta_3)} \tag{32}$$

$$B'_2(\beta_3) = \frac{-(\beta_3 - \beta_2)}{C_1(\beta_3)} - \frac{(\beta_2 - \beta_0)}{C_4(\beta_3)} - \frac{(\beta_2 - \beta_0)(\beta_3 - \beta_2)}{C_2(\beta_3)} \tag{33}$$

$$B'_3(\beta_3) = \frac{(\beta_2 - \beta_0)}{C_3(\beta_3)} \tag{34}$$

Where

$$C_0(\beta_3) = 4(\beta_2 - \beta_0)(\beta_3 - \beta_1)(\beta_1 - \beta_0) - (\beta_2 - \beta_1)^2(\beta_1 - \beta_0) \tag{35}$$

$$C_1(\beta_3) = 4(\beta_2 - \beta_0)(\beta_3 - \beta_1) - (\beta_2 - \beta_1)^2 \tag{36}$$

$$C_2(\beta_3) = 2(\beta_2 - \beta_0)(\beta_2 - \beta_1)(\beta_3 - \beta_1) - (\beta_2 - \beta_1)^3 \tag{37}$$

$$C_3(\beta_3) = 2(\beta_2 - \beta_0)\ (\beta_3 - \beta_1) - (\beta_3 - \beta_1)^2 \tag{38}$$

$$C_4(\beta_3) = 2(\beta_2 - \beta_0)(\beta_3 - \beta_1) - (\beta_2 - \beta_1)^2 \tag{39}$$

In each iteration, we check for the stability condition $s^T \omega > 0$, if it is not satisfied, we use $\omega = z$ for that iteration.

3. The Algorithm

In this section, the modified algorithm for the Broyden family method is given.

Step 1. Initialize $i = 0, B = I, \beta_0 = 0$, X_0, $\emptyset_0$, F.

Step 2. If $i \geq 10000$ (or any suitable number) go to Step 12.

Step 3. Compute g_i and $\| g_i \|$.

Step 4. If $\| g_i \| \leq 10^{-5}$ then X_i is optimal, go to Step 12.

Step 5. Use eq^n (2) to compute the search direction $D_j = B_j g_j$, given X_i, B_i, g_i.

Step 6. Use the cubic interpolation method [8, 9] compute the step length β_i.

Step 7. Use eq^n (1) to compute X_{i+1}.

Step 8. Use eq^n (6) to compute s_i.

Step 9.

1) If you employ the traditional Broyden family, use equation (7) to compute z.
2) If you employ the three points model, use equations (23) to (29) to compute ω.
3) If you employ the four points model, use equations (30) to (39) to compute ω.

Step 10. Use eq^n (21) to compute v_i and eq^n (20) to compute $B^{\emptyset}_{i+1}$.

Step 11. Set $i = i + 1$, return to Step 2.

Step 12. Stop.

4. Numerical Tests and Discussion of Results

For the validation of the applicability of the presented method, numerical tests of a few unconstrained optimization problems using the well-known programming tool for solving problems of mathematical programming, MATLAB, have been carried out. An M-file is constructed for the above algorithm. For the purpose of comparison of the suggested method with the traditional one, the problems considered in [31] are solved using the traditional Broyden family methods. The programming tool has been created for the implementation of the modifications of the above algorithm; the values of g are kept in the process of the line search for the computation of ω.

In an attempt to reduce the possibility that unusual results may be achieved for a given problem due to an unusually lucky or unlucky choice of initial values, the solution was provided for each problem with various initial values. Moreover, the solution of each problem was provided for various values of Φ, where $0 \leq \Phi \leq 1$.

"N" is the number of variables, "FG" is the number of function and gradient evaluations required for convergence, and "k" is the number of iterations for the tables. For step (9-a), M = 1, For step (9-b), M = 2 and for step (9-c) of this proposed method, M = 3.

The asterisk represents the approach yielding the best results for the test scenarios, as defined by the number of function/gradient evaluations, with ties determined by iterations.

Therefore, from the results presented in Tables 1-10, it can be concluded that the numerical results have shown the proposed method to be efficient and effective compared to the traditional method. It can be generally concluded that the new method is efficient and effective, which has resulted in a decrease in the number of function and gradient evaluation steps. Although the performance of the new method may vary slightly in individual cases, the overall performance of the new method has been found to be efficient and effective compared to the traditional method.

Moreover, the performance results presented in Table 11 have also confirmed the efficiency and effectiveness of the new method compared to the traditional method. To statistically substantiate the observed efficiency gains, a Wilcoxon signed-rank test was applied to the FG values across all tested Φ values (Table 11). The four-point NCSIP model (M=3) showed a statistically significant improvement over both the traditional Broyden method (M=1: p = 0.0371) and the three-point NCSIP model (M=2: p = 0.0488), while the difference between the three-point model (M=2) and the traditional method (M=1) was not statistically significant (p = 0.5566). These results indicate that the four-point interpolation is necessary to achieve a statistically robust improvement in efficiency, whereas the three-point model alone does not yield a significant gain.

Table 1. *Result for Φ=0.1.*

Function Name	N	Initial Point	Φ=0.1					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
	2	(-1.2, 1)	85*	109*	86	113	92	111
Rosenbrock		(5, 5)	78*	92*	113	140	174	206
		(10, 10)	115*	141	182	216	183	138*

Function Name	N	Initial Point	Φ=0.1					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
	2	(0, 0)	13	16	12*	16*	12*	16*
Beale		(1, -1)	14	21	12*	17*	14	21
		(0.1, -2)	59	75	35*	49*	51	73
	8	(90,…, 90)	178	194	72*	86*	130	143
Trigonometric		(0.1,…, 0.1)	42	46	32*	35*	37	40
		(30,…, 30)	76*	89*	109	121	82	94
	4	(-1.2, 1, -1.2, 1)	287	320	230*	254*	231	258
wood		(3, 3, 3, 3)	288	317	306	329	257*	292*
		(3, -1, 0, 1)	306	330	147*	161*	156	177
	4	(-1.2, 1, -1.2, 1)	141*	161*	211	241	177	198
Extended Rosenbrock		(0, 0, 0, 0)	86	100	16*	31*	53	63
		(3, 0, 3, 0)	275	316	294	322	137*	153*
	8	(3, -1, 0, 1, 3, -1, 0, 1)	153	177	136	154	130*	151*
Extended Powell		(10, 10, 10, 10, 0, 0, 0, 0)	100	119	91*	113*	91*	115
		(6. .... 6)	146	177	136	154	130*	151*
Total			2442	2800	2220	2552	2137*	2400*

Table 2. *Result for Φ=0.2.*

Function Name	N	Initial Point	Φ=0.2					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
	2	(-1.2, 1)	30*	40*	34	48	32	49
Rosenbrock		(5, 5)	60	77	67	94	45*	66*
		(10, 10)	66	97	80	121	54*	76*
	2	(0, 0)	14*	17*	24	27	24	27
Beale		(1, -1)	14	21	12*	17*	14	21
		(0.1, -2)	56	72	31*	38*	50	65
	8	(90,…, 90)	177	198	167*	186*	209	232
Trigonometric		(0.1,…, 0.1)	28*	33*	31	34	31	34
		(30,…, 30)	63*	76*	89	101	171	187
	4	(-1.2, 1, -1.2, 1)	197	229	148*	175*	155	181
wood		(3, 3, 3, 3)	164*	200	194	227	172	193*
		(3, -1, 0, 1)	204	227	93	107*	90*	114

Function Name	N	Initial Point	Φ=0.2					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
	4	(-1.2, 1, -1.2, 1)	102	117	126	143	77*	95*
Extended Rosenbrock		(0, 0, 0, 0)	59	70	39*	50*	46	58
		(3, 0, 3, 0)	129	152	183	222	85*	102*
	8	(3, -1, 0, 1, 3, -1, 0, 1)	123	149	112*	133*	134	155
Extended Powell		(10, 10, 10, 10, 0, 0, 0, 0)	67*	84*	101	120	71	94
		(6. .... 6)	129	149	156	179	115*	135*
Total			1682	2008	1687	2022	1575*	1884*

***Table 3**. Result for Φ=0.3.*

Function Name	N	Initial Point	Φ=0.3					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
	2	(-1.2, 1)	31*	45*	34	47	70	91
Rosenbrock		(5, 5)	51*	69*	66	88	53	74
		(10, 10)	66	89	80	115	60*	87*
	2	(0, 0)	15*	18*	21	24	21	24
Beale		(1, -1)	13	20	11*	16*	13	20
		(0.1, -2)	33	48	31	39	24*	35*
	8	(90,…, 90)	90*	104*	108	123	106	120
Trigonometric		(0.1,…, 0.1)	26*	30*	28	31	29	32
		(30,…, 30)	62*	77*	104	125	79	93
	4	(-1.2, 1, -1.2, 1)	124	160	105*	127*	109	130
wood		(3, 3, 3, 3)	124	155	156	184	108*	139*
		(3, -1, 0, 1)	92	110	72	90	67*	91*
	4	(-1.2, 1, -1.2, 1)	74*	89*	74*	93	82	99
Extended Rosenbrock		(0, 0, 0, 0)	50	61	29*	42*	36	45
		(3, 0, 3, 0)	87	108	106	133	70*	86*
	8	(3, -1, 0, 1, 3, -1, 0, 1)	60*	81*	91	112	82	101
Extended Powell		(10, 10, 10, 10, 0, 0, 0, 0)	54*	71*	66	84	61	79
		(6. .... 6)	37*	55*	139	158	84	103
Total			1089*	1390*	1321	1631	1154	1449

Table 4. *Result for Φ=0.4.*

Function Name	N	Initial Point	Φ=0.4					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
	2	(-1.2, 1)	33	44*	32*	45	36	52
Rosenbrock		(5, 5)	47	64*	43*	66	58	78
		(10, 10)	69	96	78	123	61*	85*
	2	(0, 0)	15*	18*	23	26	23	26
Beale		(1, -1)	13	20	12*	17*	13	20
		(0.1, -2)	28	41	28	38	22*	32*
	8	(90,…, 90)	72	86	58*	72*	114	134
Trigonometric		(0.1,…, 0.1)	25*	28*	26	29	26	29
		(30,…, 30)	63	78	89	103	52*	67*
	4	(-1.2, 1, -1.2, 1)	88*	120	94	119*	90	122
wood		(3, 3, 3, 3)	102*	132*	120	153	114	144
		(3, -1, 0, 1)	85	104	69	84	65	90
	4	(-1.2, 1, -1.2, 1)	69*	85*	72	90	78	97
Extended Rosenbrock		(0, 0, 0, 0)	45	60	30*	43*	39	48
		(3, 0, 3, 0)	96	116	90	111	61*	79*
	8	(3, -1, 0, 1, 3, -1, 0, 1)	66	88	60*	81*	67	84
Extended Powell		(10, 10, 10, 10, 0, 0, 0, 0)	53*	69*	72	90	68	85
		(6. .... 6)	98	122	128	147	81*	102*
Total			1067*	1371*	1124	1437	1068	1374

Table 5. *Result for Φ=0.5.*

Function Name	N	Initial Point	Φ=0.5					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
	2	(-1.2, 1)	33*	47*	39	62	43	58
Rosenbrock		(5, 5)	46	64	35*	53*	42	61
		(10, 10)	60*	86*	71	110	66	96
	2	(0, 0)	18*	21*	19	22	19	22
Beale		(1, -1)	12	19	11*	16*	12	19
		(0.1, -2)	20*	33*	29	39	28	36
	8	(90,…, 90)	114	136	136	163	106*	132*
Trigonometric		(0.1,…, 0.1)	23*	26*	24	27	24	27

Function Name	N	Initial Point	Φ=0.5					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
		(30,…, 30)	57	73	70	84	50*	67*
	4	(-1.2, 1, -1.2, 1)	82	117	78*	101*	88	119
wood		(3, 3, 3, 3)	101	131	114	147	89*	114*
		(3, -1, 0, 1)	76	105	60*	78*	62	86
	4	(-1.2, 1, -1.2, 1)	60*	76*	65	80	60*	78
Extended Rosenbrock		(0, 0, 0, 0)	44	60	28*	41*	35	46
		(3, 0, 3, 0)	94	118	79	100	54*	72*
	8	(3, -1, 0, 1, 3, -1, 0, 1)	86	109	75	91*	71*	92
Extended Powell		(10, 10, 10, 10, 0, 0, 0, 0)	61*	77*	68	85	64	81
		(6. .... 6)	65*	87*	104	124	73	93
Total			1052	1385	1105	1423	986*	1299*

Table 6. *Result for Φ*=0.6.

Function Name	N	Initial Point	Φ=0.6					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
	2	(-1.2, 1)	32*	46*	37	52	32*	46*
Rosenbrock		(5, 5)	42	59	34*	54*	39	59
		(10, 10)	61	87*	59*	92	65	92
	2	(0, 0)	18	21	16*	20*	16*	20*
Beale		(1, -1)	13	20	12*	17*	13	20
		(0.1, -2)	19	33	16*	22*	22	32
	8	(90,…, 90)	80*	102*	108	129	95	116
Trigonometric		(0.1,…, 0.1)	22*	25*	22*	25*	22*	25*
		(30,…, 30)	57*	72*	62	76	82	99
	4	(-1.2, 1, -1.2, 1)	81	112	79	105	72*	99*
wood		(3, 3, 3, 3)	92	125	100	131	84*	116*
		(3, -1, 0, 1)	74	101	61	77*	56*	80
	4	(-1.2, 1, -1.2, 1)	56	76	60	78	53*	71*
Extended Rosenbrock		(0, 0, 0, 0)	39	51	27*	39*	34	47
		(3, 0, 3, 0)	62	87	59	79	41*	60*
	8	(3, -1, 0, 1, 3, -1, 0, 1)	44*	66*	50	67	67	89
Extended Powell		(10, 10, 10, 10, 0, 0, 0, 0)	45	61	71	89	32*	49*

Function Name	N	Initial Point	Φ=0.6					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
		(6. .... 6)	88	113	65	88	61*	82*
Total			925	1257	938	1240	886*	1202*

Table 7. Result for Φ=0.7.

Function Name	N	Initial Point	Φ=0.7					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
	2	(-1.2, 1)	32*	47*	34	47*	33	47*
Rosenbrock		(5, 5)	47	63	32*	49*	42	62
		(10, 10)	60	86*	57*	86*	65	90
	2	(0, 0)	12*	16*	12*	17	12*	17
Beale		(1, -1)	13	21	11*	16*	13	21
		(0.1, -2)	19	33	15*	22*	26	32
	8	(90,…, 90)	91	115	66*	87*	82	102
Trigonometric		(0.1,…, 0.1)	22	25	21*	24*	21*	24*
		(30,…, 30)	51*	66*	77	94	77	92
	4	(-1.2, 1, -1.2, 1)	78	110	73	99	69*	93*
wood		(3, 3, 3, 3)	85	109*	87	115	81*	111
		(3, -1, 0, 1)	75	102	58	77	53*	80*
	4	(-1.2, 1, -1.2, 1)	53	74	53	71	50*	69*
Extended Rosenbrock		(0, 0, 0, 0)	36	51	24*	36*	31	41
		(3, 0, 3, 0)	53	79	58	80	44*	68*
	8	(3, -1, 0, 1, 3, -1, 0, 1)	42*	65*	69	88	50	75
Extended Powell		(10, 10, 10, 10, 0, 0, 0, 0)	48	64	63	80	31*	48*
		(6. .... 6)	80	102	69*	92*	82	103
Total			897	1228	879	1180	862*	1175*

Table 8. *Result for Φ=0.8.*

Function Name	N	Initial Point	Φ=0.8					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
Rosenbrock	2	(-1.2, 1)	33*	48*	35	48	35	51
		(5, 5)	46	63	34	50	41*	62*
		(10, 10)	61	90	50*	80*	57	81
Beale	2	(0, 0)	12	16	13	18	10*	15*
		(1, -1)	12	20	11*	16*	12	20
		(0.1, -2)	19	31	17*	23*	20	28
Trigonometric	8	(90,…, 90)	63*	79*	72	94	80	106
		(0.1,…, 0.1)	20	24	20	23	19*	22*
		(30,…, 30)	49*	64*	68	84	61	75
wood	4	(-1.2, 1, -1.2, 1)	75	116	74	99	67*	95*
		(3, 3, 3, 3)	84	112	84	109*	83*	112
		(3, -1, 0, 1)	66	93	57	76	48*	72*
Extended Rosenbrock	4	(-1.2, 1, -1.2, 1)	55	74	44*	62*	61	77
		(0, 0, 0, 0)	32	50	21*	32*	34	46
		(3, 0, 3, 0)	52	79	52	73	47*	64*
Extended Powell	8	(3, -1, 0, 1, 3, -1, 0, 1)	52*	71*	59	78	55	77
		(10, 10, 10, 10, 0, 0, 0, 0)	42*	58*	55	72	46	65
		(6. .... 6)	62	85	53	75	50*	70*
Total			835	1173	819*	1112*	826	1138

Table 9. *Result for Φ=0.9.*

Function Name	N	Initial Point	Φ=0.9					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
Rosenbrock	2	(-1.2, 1)	31*	44*	34	47	35	49
		(5, 5)	42	62	30*	48	41	36*
		(10, 10)	54*	81*	62	95	57	85
Beale	2	(0, 0)	11	15	13	18	9*	14*
		(1, -1)	13	21	12*	17*	13	21
		(0.1, -2)	21	31	17*	23*	17	27
Trigonometric	8	(90,…, 90)	66	85	62*	78*	65	81
		(0.1,…, 0.1)	20	24	19*	22*	19*	22*

Function Name	N	Initial Point	Φ=0.9					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
		(30,…, 30)	48*	63*	56	71	58	72
wood	4	(-1.2, 1, -1.2, 1)	70*	107	70*	102	73	101
		(3, 3, 3, 3)	83	116	82	113	77*	106*
		(3, -1, 0, 1)	71	101	50*	70*	50*	74
Extended Rosenbrock	4	(-1.2, 1, -1.2, 1)	49	69	46*	63*	52	71
		(0, 0, 0, 0)	32	45	23*	36*	31	41
		(3, 0, 3, 0)	51	71	43*	69	44	65*
Extended Powell	8	(3, -1, 0, 1, 3, -1, 0, 1)	36*	57*	57	79	51	75
		(10, 10, 10, 10, 0, 0, 0, 0)	40*	56*	52	69	44	61
		(6. …. 6)	56*	79*	57	82	82	103
Total			794	1127	785*	1102*	818	1104

Table 10. *Result for Φ*=1.

Function Name	N	Initial Point	Φ=1					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
Rosenbrock	2	(-1.2, 1)	34	48	34	47	28*	41*
		(5, 5)	43	62	31*	46*	38	59
		(10, 10)	60	90	50*	75*	62	91
Beale	2	(0, 0)	12	16	13	18	10*	15*
		(1, -1)	13	21	11*	16*	13	21
		(0.1, -2)	21	31	16*	23*	23	32
Trigonometric	8	(90,…, 90)	50	68	39*	54*	50	63
		(0.1,…, 0.1)	20	23	19*	22*	19*	22*
		(30,…, 30)	45*	60*	69	87	54	72
wood	4	(-1.2, 1, -1.2, 1)	72	109	67	93	61*	88*
		(3, 3, 3, 3)	80	115	79	111	77*	105*
		(3, -1, 0, 1)	71	108	49	68*	48*	73
Extended Rosenbrock	4	(-1.2, 1, -1.2, 1)	45*	63*	50	70	50	70
		(0, 0, 0, 0)	34	51	21*	34*	28	39
		(3, 0, 3, 0)	48	73	46	73	44*	62*
Extended Powell	8	(3, -1, 0, 1, 3, -1, 0, 1)	44*	64*	53	74	50	73
		(10, 10, 10, 10, 0, 0, 0, 0)	42	58	37*	54*	43	59

Function Name	N	Initial Point	Φ=1					
			M=1		M=2		M=3	
			K	FG	K	FG	K	FG
		(6. … . 6)	47*	68*	65	86	53	72
Total			781	1128	749*	1051*	751	1057

Table 11. *Numerical results for different values of Φ.*

	M=1		M=2		M=3	
Φ	**K**	**FG**	**K**	**FG**	**K**	**FG**
0.1	2442	2800	2220	2552	2137*	2400*
0.2	1682	2008	1687	2022	1575*	1884*
0.3	1089*	1390*	1321	1631	1154	1449
0.4	1067*	1371*	1124	1437	1068	1374
0.5	1052	1385	1105	1423	986*	1299*
0.6	925	1257	938	1240	886*	1202*
0.7	897	1228	879	1180	862*	1175*
0.8	835	1173	819*	1112*	826	1138
0.9	794	1127	785*	1102*	818	1104
1	781	1128	749*	1051*	751	1057
Total	11564	14867	11627	14750	11063*	14082*

In this section, the proposed NCSIP method is compared with the NDDIP method. In the comparison below, $\Phi \in [0.1]$. The notation N represents the number of variables, the notation FG represents the number of function and gradient evaluations, and the notation k represents the number of iterations Additionally, "M=2" and "M=3" represent the three and four points of the NCSI method, while "M=4" and "M=5" represent the three and four points of the NDDIP method [18, 29]. The notation * represents the best result for FG, and in the case of a tie, the result is determined by the notation k.

The numerical results clearly indicate that the proposed method performs best when M=2 or M=3. For both of these two-way methods, the proposed method requires fewer function and gradient evaluations for the majority of test problems, which clearly indicates a higher level of efficiency and robustness in Tables 12-21.

Although both methods perform competitively, a careful analysis of the overall results indicates that M=3 performs slightly better. In fact, the cumulative results presented in Table 22 clearly indicate that the total number of function and gradient evaluations is minimized when M=3, which clearly indicates its superior efficiency.

Table 12. *Result for Φ=0.1.*

Function Name	N	Initial Point	Φ=0.1							
			M=4		M=5		M=2		M=3	
			K	FG	K	FG	K	FG	K	FG
Rosenbrock	2	(-1.2, 1)	50*	70*	90	109	86	113	92	111
		(5, 5)	115	150	162	194	113*	140*	174	206
		(10, 10)	60*	105*	183	238	182	216	183	138
Beale	2	(0, 0)	12*	16*	12*	16*	12*	16*	12*	16*
		(1, -1)	12*	17*	14	21	12*	17*	14	21
		(0.1, -2)	29*	41*	102	118	35	49	51	73
Trigonometric	8	(90,…, 90)	72*	86*	110	123	72*	86*	130	143
		(0.1,…, 0.1)	32*	35*	37	40	32*	35*	37	40
		(30,…, 30)	109	121	111	123	109	121	82*	94*
wood	4	(-1.2, 1, -1.2, 1)	231	266	208*	235*	230	254	231	258
		(3, 3, 3, 3)	325	362	249*	282*	306	329	257	292
		(3, -1, 0, 1)	147*	161*	156	177	147*	161*	156	177
Extended Rosenbrock	4	(-1.2, 1, -1.2, 1)	148	164	147*	160*	211	241	177	198
		(0, 0, 0, 0)	14*	25*	67	77	16	31	53	63
		(3, 0, 3, 0)	216	247	123*	137*	294	322	137	153
Extended Powell	8	(3, -1, 0, 1, 3, -1, 0, 1)	119	142	96*	117*	136	154	130	151
		(10, 10, 10, 10, 0, 0, 0, 0)	91*	113*	91*	115	91*	113*	91*	115
		(6. .... 6)	202	229	176	195	136	154	130*	151*
Total			1984*	2350*	2134	2477	2220	2552	2137	2400

Table 13. *Result for Φ=0.2.*

Function Name	N	Initial Point	Φ=0.2							
			M=4		M=5		M=2		M=3	
			K	FG	K	FG	K	FG	K	FG
Rosenbrock	2	(-1.2, 1)	34	48*	32*	49	34	48*	32*	49
		(5, 5)	43*	63*	46	67	67	94	45	66
		(10, 10)	83	126	52*	74*	80	121	54	76
Beale	2	(0, 0)	24*	27*	24*	27*	24*	27*	24*	27*
		(1, -1)	12*	17*	14	21	12*	17*	14	21
		(0.1, -2)	31*	38*	50	65	31*	38*	50	65
Trigonometric	8	(90,…, 90)	167*	186*	199	222	167*	186*	209	232
		(0.1,…, 0.1)	31*	34*	31*	34*	31*	34*	31*	34*

Function Name	N	Initial Point	Φ=0.2 M=4 K	FG	M=5 K	FG	M=2 K	FG	M=3 K	FG
		(30,…, 30)	89*	101*	92	104	89*	101*	171	187
wood	4	(-1.2, 1, -1.2, 1)	176	209	159	186	148*	175*	155	181
		(3, 3, 3, 3)	216	250	170*	190*	194	227	172	193
		(3, -1, 0, 1)	93	107	90*	114*	93	107	90*	114*
Extended Rosenbrock	4	(-1.2, 1, -1.2, 1)	142	158	133	161	126	143	77*	95*
		(0, 0, 0, 0)	33*	45*	49	59	39	50	46	58
		(3, 0, 3, 0)	154	177	86	101*	183	222	85*	102
Extended Powell	8	(3, -1, 0, 1, 3, -1, 0, 1)	113	132	73*	94*	112	133	134	155
		(10, 10, 10, 10, 0, 0, 0, 0)	101	120	71*	94*	101	120	71*	94*
		(6. .... 6)	112	131	93*	111*	156	179	115	135
Total			1654	1969	1464*	1773*	1687	2022	1575	1884

Table 14. *Result for Φ=*0.3.

Function Name	N	Initial Point	Φ=0.3 M=4 K	FG	M=5 K	FG	M=2 K	FG	M=3 K	FG
Rosenbrock	2	(-1.2, 1)	34*	47*	70	91	34*	47*	70	91
		(5, 5)	43*	61*	53	74	66	88	53	74
		(10, 10)	62	99	61	87	80	115	60*	87*
Beale	2	(0, 0)	21*	24*	21*	24*	21*	24*	21*	24*
		(1, -1)	11*	16*	13	20	11*	16*	13	20
		(0.1, -2)	31	39	54	67	31	39	24*	35*
Trigonometric	8	(90,…, 90)	108	123	111	125	108	123	106*	120*
		(0.1,…, 0.1)	28*	31*	29	32	28*	31*	29	32
		(30,…, 30)	99	121	98	113	104	125	79*	93*
wood	4	(-1.2, 1, -1.2, 1)	113	139	116	138	105*	127*	109	130
		(3, 3, 3, 3)	146	173	108*	139*	156	184	108*	139*
		(3, -1, 0, 1)	72	90*	67*	91	72	90*	67*	91
Extended Rosenbrock	4	(-1.2, 1, -1.2, 1)	77	96	88	109	74*	93*	82	99
		(0, 0, 0, 0)	39	52	34	44	29*	42*	36	45
		(3, 0, 3, 0)	120	147	71	87	106	133	70*	86*
Extended Powell	8	(3, -1, 0, 1, 3, -1, 0, 1)	60*	83*	93	116	91	112	82	101
		(10, 10, 10, 10, 0, 0, 0, 0)	66	84	61*	79*	66	84	61*	79*

Function Name	N	Initial Point	Φ=0.3 M=4 K	FG	M=5 K	FG	M=2 K	FG	M=3 K	FG
		(6. .... 6)	114	136	113	133	139	158	84*	103*
Total			1244	1561	1261	1569	1321	1631	1154*	1449*

Table 15. *Result for Φ*=0.4.

Function Name	N	Initial Point	Φ=0.4 M=4 K	FG	M=5 K	FG	M=2 K	FG	M=3 K	FG
	2	(-1.2, 1)	32*	45*	36	52	32*	45*	36	52
Rosenbrock		(5, 5)	67	88	58	78	43*	66*	58	78
		(10, 10)	58*	90	61	85*	78	123	61	85*
	2	(0, 0)	23*	26*	23*	26*	23*	26*	23*	26*
Beale		(1, -1)	12*	17*	13	20	12*	17*	13	20
		(0.1, -2)	28	38	31	43	28	38	22*	32*
	8	(90,…, 90)	58*	72*	63	76	58*	72*	114	134
Trigonometric		(0.1,…, 0.1)	26*	29*	26*	29*	26*	29*	26*	29*
		(30,…, 30)	89	103	86	106	89	103	52*	67*
	4	(-1.2, 1, -1.2, 1)	80*	104*	86	107	94	119	90	122
wood		(3, 3, 3, 3)	120	146	114*	144*	120	153	114*	144*
		(3, -1, 0, 1)	69	84*	65*	90	69	84*	65*	90
	4	(-1.2, 1, -1.2, 1)	76	96	86	108	72*	90*	78	97
Extended Rosenbrock		(0, 0, 0, 0)	31	45	40	49	30*	43*	39	48
		(3, 0, 3, 0)	93	118	63	80	90	111	61*	79*
	8	(3, -1, 0, 1, 3, -1, 0, 1)	86	105	84	102	60*	81*	67	84
Extended Powell		(10, 10, 10, 10, 0, 0, 0, 0)	72	90	68*	85*	72	90	68*	85*
		(6. .... 6)	109	132	117	137	128	147	81*	102*
Total			1129	1428	1120	1417	1124	1437	1068*	1374*

Table 16. *Result for Φ=0.5.*

Function Name	N	Initial Point	Φ=0.5 M=4 K	FG	M=5 K	FG	M=2 K	FG	M=3 K	FG
Rosenbrock	2	(-1.2, 1)	39*	62	43	58*	39*	62	43	58*
		(5, 5)	41	60	42	61	35*	53*	42	61
		(10, 10)	60*	91*	68	98	71	110	66	96
Beale	2	(0, 0)	19*	22*	19*	22*	19*	22*	19*	22*
		(1, -1)	11*	16*	12	19	11*	16*	12	19
		(0.1, -2)	29	39	32	41	29	39	28*	36*
Trigonometric	8	(90,…, 90)	141	168	88*	108*	136	163	106	132
		(0.1,…, 0.1)	24*	27*	24*	27*	24*	27*	24*	27*
		(30,…, 30)	70	84	66	80	70	84	50*	67*
wood	4	(-1.2, 1, -1.2, 1)	82	108	86	117	78*	101*	88	119
		(3, 3, 3, 3)	115	143	89*	114*	114	147	89*	114*
		(3, -1, 0, 1)	60*	78*	62	82	60*	78*	62	86
Extended Rosenbrock	4	(-1.2, 1, -1.2, 1)	65	81	56*	75*	65	80	60	78
		(0, 0, 0, 0)	27*	40*	36	47	28	41	35	46
		(3, 0, 3, 0)	63	80	53*	70*	79	100	54	72
Extended Powell	8	(3, -1, 0, 1, 3, -1, 0, 1)	82	101	76	96	75	91*	71*	92
		(10, 10, 10, 10, 0, 0, 0, 0)	68	85	64*	81*	68	85	64*	81*
		(6. ….6)	115	138	86	107	104	124	73*	93*
Total			1111	1423	1002	1303	1105	1423	986*	1299*

Table 17. *Result for Φ=0.6.*

Function Name	N	Initial Point	Φ=0.6 M=4 K	FG	M=5 K	FG	M=2 K	FG	M=3 K	FG
Rosenbrock	2	(-1.2, 1)	37	52	32	46	37	52	32	46
		(5, 5)	39	58	39	59	34	54	39	59
		(10, 10)	64	103	61	86	59	92	65	92
Beale	2	(0, 0)	16	20	16	20	16	20	16	20
		(1, -1)	12	20	13	20	12	17	13	20
		(0.1, -2)	16	22	22	32	16	22	22	32
Trigonometric	8	(90,…, 90)	108	129	87	107	108	129	95	116
		(0.1,…, 0.1)	22	25	22	25	22	25	22	25

Function Name	N	Initial Point	Φ=0.6							
			M=4		M=5		M=2		M=3	
			K	FG	K	FG	K	FG	K	FG
		(30,…, 30)	62	76	72	88	62	76	82	99
	4	(-1.2, 1, -1.2, 1)	77	103	70	97	79	105	72	99
wood		(3, 3, 3, 3)	90	115	85	117	100	131	84	116
		(3, -1, 0, 1)	61	77	56	80	61	77	56	80
	4	(-1.2, 1, -1.2, 1)	60	78	55	73	60	78	53	71
Extended Rosenbrock		(0, 0, 0, 0)	27	39	33	42	27	39	34	47
		(3, 0, 3, 0)	61	81	52	70	59	79	41	60
	8	(3, -1, 0, 1, 3, -1, 0, 1)	45	62	53	76	50	67	67	89
Extended Powell		(10, 10, 10, 10, 0, 0, 0, 0)	71	89	32	49	71	89	32	49
		(6. … . 6)	75	99	80	99	65	88	61	82
Total			943	1248	880*	1186*	938	1240	886	1202

Table 18. *Result for Φ*=0.7.

Function Name	N	Initial Point	Φ=0.7							
			M=4		M=5		M=2		M=3	
			K	FG	K	FG	K	FG	K	FG
	2	(-1.2, 1)	34	47*	33*	47*	34	47*	33*	47*
Rosenbrock		(5, 5)	43	64	42	62	32*	49*	42	62
		(10, 10)	59	95	65	90	57*	86*	65	90
	2	(0, 0)	12	17	12*	17*	12*	17*	12*	17*
Beale		(1, -1)	11*	16*	13	21	11*	16*	13	21
		(0.1, -2)	12*	22*	26	32	15	22	26	32
	8	(90,…, 90)	66*	87*	97	121	66*	87*	82	102
Trigonometric		(0.1,…, 0.1)	21*	24*	21*	24*	21*	24*	21*	24*
		(30,…, 30)	77	94	69*	88*	77	94	77	92
	4	(-1.2, 1, -1.2, 1)	73	100	70	94	73	99	69*	93*
wood		(3, 3, 3, 3)	94	131	81*	111*	87	115	81*	111*
		(3, -1, 0, 1)	58	77*	53*	80	58	77*	53*	80
	4	(-1.2, 1, -1.2, 1)	60	78	54	73	53	71	50*	69*
Extended Rosenbrock		(0, 0, 0, 0)	26	38	34	44	24*	36*	31	41
		(3, 0, 3, 0)	58	80	42*	61*	58	80	44	68
	8	(3, -1, 0, 1, 3, -1, 0, 1)	59	79	55	74	69	88	50*	75*
Extended Powell		(10, 10, 10, 10, 0, 0, 0, 0)	63	80	31	48	63	80	31*	48*

Function Name	N	Initial Point	Φ=0.7							
			M=4		M=5		M=2		M=3	
			K	FG	K	FG	K	FG	K	FG
		(6. …. 6)	86	111	41	62	69	92	82	103
Total			912	1240	839*	1149*	879	1180	862	1175

Table 19. *Result for Φ=0.8.*

Function Name	N	Initial Point	Φ=0.8							
			M=4		M=5		M=2		M=3	
			K	FG	K	FG	K	FG	K	FG
	2	(-1.2, 1)	35*	48*	35*	51	35*	48*	35*	51
Rosenbrock		(5, 5)	39	58	41	62	34*	50*	41	62
		(10, 10)	58	86	57	81	50*	80*	57	81
	2	(0, 0)	13	18	10*	15*	13	18	10*	15*
Beale		(1, -1)	11*	16*	12	20	11*	16*	12	20
		(0.1, -2)	17*	23*	20	28	17*	23*	20	28
	8	(90,…, 90)	76	98	86	112	72*	94*	80	106
Trigonometric		(0.1,…, 0.1)	20	23	19*	22*	20	23	19*	22*
		(30,…, 30)	70	89	59*	73*	68	84	61	75
	4	(-1.2, 1, -1.2, 1)	71	103	67*	95*	74	99	67*	95*
wood		(3, 3, 3, 3)	83*	116	83*	112	84*	109	83*	112
		(3, -1, 0, 1)	57	76	48*	72*	57	76	48*	72*
	4	(-1.2, 1, -1.2, 1)	54	71	57	76	44*	62*	61	77
Extended Rosenbrock		(0, 0, 0, 0)	24	37	31	40	21*	32*	34	46
		(3, 0, 3, 0)	51	78	49	70	52	73	47*	64*
	8	(3, -1, 0, 1, 3, -1, 0, 1)	47*	66*	58	79	59	78	55	77
Extended Powell		(10, 10, 10, 10, 0, 0, 0, 0)	55	72	46*	65*	55	72	46*	65*
		(6. …. 6)	71	94	63	84	53	75	50*	70*
Total			852	1172	841	1157	819*	1112*	826	1138

Table 20. *Result for Φ*=0.9.

Function Name	N	Initial Point	Φ=0.9 M=4 K	FG	M=5 K	FG	M=2 K	FG	M=3 K	FG
Rosenbrock	2	(-1.2, 1)	34*	47*	35	49	34*	47*	35	49
		(5, 5)	40	61	41	63	30*	48*	41	36
		(10, 10)	58	92	58	86	62	95	57*	85*
Beale	2	(0, 0)	13	18	9*	14*	13	18	9*	14*
		(1, -1)	12*	17*	13	21	12*	17*	13	21
		(0.1, -2)	17*	23*	23	32	17*	23*	17*	27
Trigonometric	8	(90,…, 90)	62*	78*	63	78*	62*	78*	65	81
		(0.1,…, 0.1)	19*	22*	19*	22*	19*	22*	19*	22*
		(30,…, 30)	56*	71*	58	73	56*	71*	58	72
wood	4	(-1.2, 1, -1.2, 1)	65*	91*	73	101	70	102	73	101
		(3, 3, 3, 3)	83	114	77*	106*	82	113	77*	106*
		(3, -1, 0, 1)	50*	70*	50*	74	50*	70*	50*	74
Extended Rosenbrock	4	(-1.2, 1, -1.2, 1)	53	73	54	74	46*	63*	52	71
		(0, 0, 0, 0)	21*	33*	29	40	23	36	31	41
		(3, 0, 3, 0)	54	75	41*	61*	43	69	44	65
Extended Powell	8	(3, -1, 0, 1, 3, -1, 0, 1)	58	80	52	73	57	79	51*	75*
		(10, 10, 10, 10, 0, 0, 0, 0)	52	69	44*	61*	52	69	44*	61*
		(6. …. 6)	69	95	44*	65*	57	82	82	103
Total			816	1129	783*	1093*	785	1102	818	1104

Table 21. *Result for Φ*=1.

Function Name	N	Initial Point	Φ=1 M=4 K	FG	M=5 K	FG	M=2 K	FG	M=3 K	FG
Rosenbrock	2	(-1.2, 1)	34	47	28*	41*	34	47	28*	41*
		(5, 5)	39	59	38	59	31*	46*	38	59
		(10, 10)	55	85	62	91	50*	75*	62	91
Beale	2	(0, 0)	13	18	10*	15*	13	18	10*	15*
		(1, -1)	11*	16*	13	21	11*	16*	13	21
		(0.1, -2)	16	23	21	35	16*	23*	23	32
Trigonometric	8	(90,…, 90)	39*	54*	50	64	39*	54*	50	63
		(0.1,…, 0.1)	19*	22*	19*	22*	19*	22*	19*	22*

Function Name	N	Initial Point	Φ=1							
			M=4		M=5		M=2		M=3	
			K	FG	K	FG	K	FG	K	FG
		(30,…, 30)	69	87	66	81	69	87	54*	72*
	4	(-1.2, 1, -1.2, 1)	65	93	61*	88*	67	93	61*	88*
wood		(3, 3, 3, 3)	76*	110	78	108	79	111	77	105*
		(3, -1, 0, 1)	49	68*	48*	73	49	68*	48*	73
	4	(-1.2, 1, -1.2, 1)	56	77	51	71	50*	70*	50*	70*
Extended Rosenbrock		(0, 0, 0, 0)	22	35	28	41	21*	34*	28	39
		(3, 0, 3, 0)	49	72	43*	62*	46	73	44	62*
	8	(3, -1, 0, 1, 3, -1, 0, 1)	47	67	34*	54*	53	74	50	73
Extended Powell		(10, 10, 10, 10, 0, 0, 0, 0)	37*	54	42	59	37*	54*	43	59
		(6. …. 6)	78	97	57	78	65	86	53*	72*
Total			774	1084	749*	1063	749*	1051*	751	1057

Table 22. Numerical results for different values of Φ.

	M=4		M=5		M=2		M=3	
Φ	K	FG	K	FG	K	FG	K	FG
0.1	1984*	2350*	2134	2477	2220	2552	2137	2400
0.2	1654	1969	1464	1773	1687	2022	1575*	1884*
0.3	1244	1561	1261	1569	1321	1631	1154*	1449*
0.4	1129	1428	1120	1417	1124	1437	1068*	1374*
0.5	1111	1423	1002	1303	1105	1423	986*	1299*
0.6	943	1248	880*	1186*	938	1240	886	1202
0.7	912	1240	839*	1149*	879	1180	862	1175
0.8	852	1172	841	1157	819*	1112*	826	1138
0.9	816	1129	783	1093	785*	1102*	818	1104
1	774	1084	749*	1063	749*	1051*	751	1057
Total	11419	14604	11073	14187	11627	14750	11063*	14082*

5. Disadvantages of the New Methods

The major drawback of the NCSIP approach is the additional computational time and storage space needed for constructing the spline form and solving the related linear system. However, this additional cost is not unreasonable when compared with the cost of updating and storing other approximation matrices.

6. Conclusions

The paper points out that using more gradient evaluations makes it possible to approximate the second-derivatives matrix more effectively, by applying the "Newton equation." It has been shown that the new algorithm may be expected to reduce to the "secant-based" Broyden family methods when close to the minimum, particularly in situations where only two points are used. As a result, the convergence properties characteristic of the Broyden family methods are carried over to the new algorithm.

The numerical experiments conducted yielded encouraging results. It is worth emphasizing that this concept can be readily applied to any existing algorithm, simply by replacing z with ω in the updating formula.

Future Work

Building on the limitations identified above, several directions are recommended for future research:

1) Large-scale testing. The present study was restricted to small-scale test problems (N = 2, 4, 8). Since quasi-Newton methods are of particular interest for large-scale optimization, future work should test the proposed NCSIP-based algorithm on higher-dimensional problems (e.g., N = 50, 100, 500) to assess whether the observed efficiency gains persist, diminish, or grow as problem size increases, and to evaluate the scalability of the spline construction cost relative to problem dimension.
2) Broader benchmarking. Comparisons in this study were limited to the traditional Broyden family and the NDDI method. Future work should benchmark the proposed algorithm against L-BFGS and other limited-memory quasi-Newton variants, which are widely used in practice, in order to properly contextualize the reported performance gains within the broader landscape of available optimization methods.
3) Reporting computational cost. Only the number of iterations (k) and function/gradient evaluations (FG) were reported in this study. Given the acknowledged additional computational overhead of constructing the spline model and solving the associated linear system, future work should report CPU time and memory usage alongside k and FG, to provide a more complete picture of the trade-off between the modest reduction in FG and the added cost per iteration.

Abbreviations

BF	Broyden Family
NCSIP	Natural Cubic Spline Interpolation Polynomial
NDDI	Newton Divided Difference Interpolation

Author Contributions

Tarek Abou-El-Enien: Conceptualization, Formal Analysis, Investigation, Methodology, Software, Supervision, Writing – review & editing

Kamal El-Dib: Supervision, Writing – review & editing

Soher Mohamed: Conceptualization, Formal Analysis, Investigation, Methodology, Software, Validation, Writing – original draft

Data Availability Statement

The data supporting the outcome of this research work has been reported in this manuscript.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Hassan, B. A., Ayoob, A. R. On the New Quasi-Newton Equation for Unconstrained Optimization. In Proceedings of the International Engineering Conference on Sustainable Technology and Development (IEC); 2022. https://doi.org/10.1109/IEC54822.2022.9807584

[2] Moghrabi, I. A. R., Hassan, B. A. An Efficient Limited Memory Multi-Step Quasi-Newton Method. Mathematics. 2024, 12(5), 768. https://doi.org/10.3390/math12050768

[3] Nosrati, M., Amini, K. A New Diagonal Quasi-Newton Algorithm for Unconstrained Optimization Problems. Applications of Mathematics. 2024, 69(4). https://doi.org/10.21136/AM.2024.0045-24

[4] Lai, K. K., Mishra, S. K., Sharma, R., Sharma, M., Ram, B. A Modified q-BFGS Algorithm for Unconstrained Optimization. Mathematics. 2023, 11(6), 1420. https://doi.org/10.3390/math11061420

[5] Vater, N., Borzì A. Convergence of a Quasi-Newton Method for Solving Systems of Nonlinear Underdetermined Equations. Computational Optimization and Applications. 2025, 973-996. https://doi.org/10.1007/s10589-024-00606-3

[6] Hassan, B. A., Moghrabi, I. A. R. A Modified Secant Equation Quasi-Newton Method for Unconstrained Optimization. Journal of Applied Mathematics and Computing. 2023, 451-464. https://doi.org/10.1007/s12190-022-01750-x

[7] Zhang, W., Guo, T., Wu, J., Ruan, Z., Qiu, S. A Modified BFGS Quasi-Newton Method with Wolfe Line Search for Unconstrained Optimization. AIMS Mathematics. 2026, 11(1), 767-784. https://doi.org/10.3934/math.2026033

[8] Fletcher, R. Practical Methods of Optimization. 2nd ed. New York: Wiley; 1987.

[9] Sun, W., Yuan, Y. Optimization Theory and Methods: Nonlinear Programming. USA: Springer; 2006.

[10] Rao, S. S. Engineering Optimization: Theory and Practice. 5th ed. John Wiley & Sons; 2020.

[11] Popkov, A. Y. Gradient Methods for Nonstationary Unconstrained Optimization Problems. Automation and Remote Control. 2005, 66(6), 883-891. https://doi.org/10.1007/s10513-005-0132-z

[12] Fletcher, R. A New Approach to Variable Metric Algorithms. Computer Journal. 1970, 13(3), 317-322. https://doi.org/10.1093/comjnl/13.3.317

[13] Gertz, E. M. A Quasi-Newton Trust-Region Method. Mathematical Programming. 2004, 100(3), 447-470. https://doi.org/10.1007/s10107-004-0511-1

[14] Broyden, C. G. The Convergence of a Class of Double Rank Minimization Algorithms. Journal of the Institute of Mathematics and Its Applications. 1970, 6, 76-90. https://doi.org/10.1093/imamat/6.3.222

[15] Burmen, A., Bratkovic, F., Puhan, J., Fajfar, I., Tuma, T. Extended Global Convergence Framework for Unconstrained Optimization. Acta Mathematica Sinica. 2004, 20(3), 433-440. https://doi.org/10.1007/s10114-004-0340-4

[16] Fletcher, R., Powell, M. J. D. A Rapidly Convergent Descent Method for Minimization. Computer Journal. 1963, 6(2), 163-168. https://doi.org/10.1093/comjnl/6.2.163

[17] Shi, Z. Convergence of Multi-Step Curve Search Method for Unconstrained Optimization. Journal of Numerical Mathematics. 2004, 12(4), 297-309. https://doi.org/10.1163/1569395042571292

[18] Saadallah, A. A., Abou-El-Enien, T. H. M. The Use of Additional Gradient Evaluations in Quasi-Newton Methods. Journal of Advances in Modelling and Analysis A. 1995, 25(1), 41-50.

[19] Dennis, J. E. On Some Methods Based on Broyden's Secant Approximation to the Hessian. In Numerical Methods for Nonlinear Optimization, F. Lootsma, Ed. London: Academic Press; 1977.

[20] Yabe, H., Ogasawara, H., Yashino, M. Local and Superlinear Convergence of Quasi-Newton Methods Based on Modified Secant Conditions. Journal of Computational and Applied Mathematics. 2007, 205(1), 617-632. https://doi.org/10.1016/j.cam.2006.05.018

[21] Ford, J. A., Saadallah, A. A. A Rational Function Model for Unconstrained Optimization. In Numerical Methods, Vol. 50, D. Greenspan, P. Rozsa, Eds.; 1988, pp. 49-64.

[22] Biggs, M. C. Minimization Algorithm Making Use of Non-Quadratic Properties of the Objective Function. Journal of the Institute of Mathematics and Its Applications. 1971, 8(3), 315-327. https://doi.org/10.1093/imamat/8.3.315

[23] Biggs, M. C. A Note on Minimization Algorithms Which Make Use of Nonquadratic Properties of the Objective Function. Journal of the Institute of Mathematics and Its Applications. 1973, 12(3), 337-338. https://doi.org/10.1093/imamat/12.3.337

[24] Ford, J. A., Ghandhari, R.-A. On the Use of Function-Values in Unconstrained Optimization. Journal of Computational and Applied Mathematics. 1989, 28, 187-198. https://doi.org/10.1016/0377-0427(89)90331-2

[25] Ford, J. A., Ghandhari, R.-A. On the Use of Curvature Estimates in Quasi-Newton Methods. Journal of Computational and Applied Mathematics. 1991, 35, 185-196. https://doi.org/10.1016/0377-0427(91)90206-Y

[26] Ford, J. A., Moghrabi, I. A. Multi-Step Quasi-Newton Methods for Optimization. Journal of Computational and Applied Mathematics. 1994, 50, 305-323. https://doi.org/10.1016/0377-0427(94)90309-3

[27] Ford, J. A., Saadallah, A. A. On the Construction of Minimization Methods of Quasi-Newton Type. Journal of Computational and Applied Mathematics. 1987, 20, 239-246. https://doi.org/10.1016/0377-0427(87)90140-3

[28] Saadallah, A. A. A Quasi-Newton Method Using a Nonquadratic Model. Journal of Computational and Applied Mathematics. 1994, 51, 327-338. https://doi.org/10.1016/0377-0427(92)00115-P

[29] Abou-El-Einen, T. H. M. A Modified Algorithm for Broyden Family. Advances in Modelling and Analysis A. 2009, 46(2), 79-101.

[30] Fox, L., Mayers, D. F. Computing Methods for Scientists and Engineers. Oxford: Oxford University Press; 1968.

[31] Shanno, D. F., Phua, K. H. Matrix Conditioning and Nonlinear Optimization. Mathematical Programming. 1978, 14, 149-160. https://doi.org/10.1007/BF01588962

Biography

Tarek Abou-El-Enien Abou-El-Enien is a Professor of Mathematical Optimization & Operations Research- Head of. Operations Research & Decision Support Department - Faculty of Computing & AI-Cairo University- Giza, Egypt. Applied member in International Society on Multiple Criteria Decision Making, Georgia, USA, since 1999.

Kamal El-Dib is a Professor of Mathematics at the Faculty of Science, Fayoum University, Egypt. He has been a Professor since 1986 and has served as Acting Head of the Department of Mathematics since 2011. His research interests include Pure Mathematics, Mathematical Analysis, Mathematical Optimization, and Numerical Mathematics.

Soher Mohamed received her degree in B.Sc. in Mathematics, at the faculty of science, Fayoum University in 2020. She is currently an MSc student. She is employed with the Department of Basic Sciences, Faculty of Computers and Artificial Intelligence, Fayoum University as Lecturer Assistant.

Research Field

Tarek Abou-El-Enien: mathematical optimization, numerical analysis, decision analysis, soft computing, modeling, and multi-level multi-objective Programming field

Kamal El-Dib: qualitative theory of ordinary differential equations, Stability of differential equations, fuzzy mathematics, fuzzy algebra, fuzzy topology, fuzzy set theory, pure Mathematics, mathematical analysis, mathematical optimization, and numerical Mathematics

Soher Mohamed: mathematical optimization, numerical analysis, decision analysis, soft computing and modeling