Check for updates

RESEARCH ARTICLE

Update Quasi-Newton Algorithm for Training ANN

[version 1; peer review: awaiting peer review]

Farah Ghazi[1], Luma Tawfiq[1], Zainab Kareem[2]

[1]Mathematics, University of Baghdad, Baghdad, Iraq
[2]Ministry of Education, Directorate General of Education, KARKH II, Baghdad, Iraq

v1 **First published:** 16 Jan 2026, **15**:71
https://doi.org/10.12688/f1000research.172826.1

Latest published: 16 Jan 2026, **15**:71
https://doi.org/10.12688/f1000research.172826.1

Abstract

The proposed design of neural network in this article is based on new accurate approach for training by unconstrained optimization, especially update quasi-Newton methods are perhaps the most popular general-purpose algorithms. A limited memory BFGS algorithm is presented for solving large-scale symmetric nonlinear equations, where a line search technique without derivative information is used. On each iteration, the updated approximations of Hessian matrix satisfy the quasi-Newton form, which traditionally served as the basis for quasi-Newton methods. On the basis of the quadratic model used in this article, we add a new update of quasi-Newton form. One innovative features of this form's is its ability to estimate the energy function's or performance function with high order precision with second-order curvature while employ the given function value data and gradient. The global convergence of the proposed algorithm is established under some suitable conditions. Under some hypothesis the approach is established to be globally convergent. The updated approaches can be numerical and more efficient than the existing comparable traditional methods, as illustrated by numerical trials. Numerical results show that the given method is competitive to those of the normal BFGS methods. We show that solving a partial differential equation can be formulated as a multi-objective optimization problem, and use this formulation to propose several modifications to existing methods. Also the proposed algorithm is used to approximate the optimal scaling parameter, which can be used to eliminate the need to optimize this parameter. Our proposed update is tested on a variety of partial differential equations and compared to existing methods. These partial differential equations include the fourth order three dimensions nonlinear equation, which we solve in up to four dimensions, the convection-diffusion equation, all of which show that our proposed update lead to enhanced accuracy.

Open Peer Review

Approval Status *AWAITING PEER REVIEW*

Any reports and responses or comments on the article can be found at the end of the article.

Keywords
Robust quasi-Newton methods, Convergence analysis, Numerical experiments, ANNs. unconstrained optimization.

This article is included in the Fallujah Multidisciplinary Science and Innovation gateway.

Corresponding author: Luma Tawfiq (luma.n.m@ihcoedu.uobaghdad.edu.iq)

Author roles: Ghazi F: Data Curation, Resources, Software, Validation; **Tawfiq L**: Conceptualization, Formal Analysis, Funding Acquisition, Investigation, Methodology, Supervision, Writing – Review & Editing; **Kareem Z**: Methodology, Project Administration, Visualization, Writing – Original Draft Preparation

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

How to cite this article: Ghazi F, Tawfiq L and Kareem Z. **Update Quasi-Newton Algorithm for Training ANN [version 1; peer review: awaiting peer review]** F1000Research 2026, **15**:71 https://doi.org/10.12688/f1000research.172826.1

First published: 16 Jan 2026, **15**:71 https://doi.org/10.12688/f1000research.172826.1

1. Introduction

In recent years, some authors have used neural networks (ANNs) as an important technique to solve many real-world problems because of their specifications. Some authors have used ANNs to solve different types of differential equations, such that[1,3] first proposed the concept of solving differential equations using ANNs by formulating a trial solution of the differential equation. The authors tested the applicability and accuracy of their developed method not only for differential equations but also for systems of coupled differential equations. Furthermore, the authors compared their results with those obtained using other numerical methods and reported that the developed ANN was superior in terms of memory requirements and accuracy.[4–6] For this reason, the authors aimed to develop this technique to obtain the best results. One of these developments is the training rules, particularly the quasi-Newton method, because it is a second-order convergence. Many authors such[7–12] have proposed modifications for the training algorithm. Others such[13–20] suggest some rules for the speed of convergence. Several attempts have been made to solve different types of differential equations by using feed forward neural networks. In,[21] reported a hybrid method was reported that combines optimization techniques with neural networks to solve high-order differential equations.

The quasi-Newton method is the most useful method for minimizing a smooth n variable function.

$$\text{minimize}\, f(x), \mathrm{x} \in R^n \tag{1}$$

where $f: R^n \to R^1$ is continuously differentiable.[22] In contrast to utilizing the real value of the Hessian or its inverse, in the proposed update, we use a symmetric positive definite estimate of the Hessian (H) or its inverse (*inv* H). The following is the form:

$$x_{k+1} = x_k + \alpha_k d_k,\ d_k = -\frac{g_k}{H_k} = -g_k H_k^{-1} \tag{2}$$

If H is not an invertible matrix, then the pseudoinverse of H.

Wolfe conditions are used to determine the step length (α_k) and search direction (d_k), as follows:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k \tag{3}$$

$$d_k^T g(x_k + \alpha_k d_k) \geq \sigma\, d_k^T g_k \tag{4}$$

where $0 < \delta < \sigma < 1$was typically used. For more details, refer to.[23] The parameter α_k is computed using a line - search in the following form:

$$\alpha_k = -g_k^T d_k / d_k^T Q d_k \tag{5}$$

For more details, please refer to.[24] Its direction is computed by solving:

$$B_k d_k + g_k = 0 \tag{6}$$

For each iteration, B_k is the updated Hessian estimate. The Broyden Fletcher Goldfarb-Shanno (BFGS) approach, proposed by Broyden, Fletcher, Goldfarb, and Shanno, is now one of the most effective training methods. Using the following formula, matrix B_{k+1} in the BFGS technique can be updated:

$$B_{k+1}^{BFGS} = B_k - \frac{B_k s_k s_k^T B_k{}^T}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k} \tag{7}$$

Let H_k be the inverse of B_k. Undoubtedly, the suggested update in (8) is publicly known as

$$H_{k+1}^{BFGS} = H_k - \frac{H_k y_k s_k^T + s_k y_k^T H_k}{s_k^T y_k} + \left[1 + \frac{y_k^T H_k y_k}{s_k^T y_k}\right] \frac{s_k s_k^T}{s_k^T y_k} \tag{8}$$

See[25,26] for further details. For the update process, we let:

$$B_{k+1} s_k = y_k \tag{9}$$

where $s_k = x_{k+1} - x_k = \alpha_k d_k$ and $y_k = g_{k+1} - g_k$ (see[27]). The numerical experiment showed that the BFGS technique outperformed all the other training approaches. Convex minimization using the update approach has been extensively investigated; for example, see.[1,2,28] To demonstrate that the update approach using the Wolfe line search may not succeed for non-convex functions, Dai created an example with six cycling points.[29] Many improvements have been suggested, including changes in the regular BFGS technique, and a modified BFGS algorithm (MBFGS) has been devised to improve and speed the global convergence of the BFGS method.[30,31] They demonstrated that the approach converged worldwide for nonconvex optimization problems. To determine whether a novel quasi-Newton methodology has global convergence and outperforms the BFGS method in terms of computation, see.[32,33] In practice, the modified BFGS technique is typically preferred to efficiently compute matrix H (or H^{-1}) using a symmetric positive definite matrix. While the standard method employs only gradient values, the modified approach uses both. Without making any convexity assumptions about the goal function, global convergence was demonstrated.[34]

2. Derivation of suggested update

A new additional update was derived using a quadratic model of the goal function. Consequently, the quadratic model of the objective function is given as

$$f_{k+1} = f_k + s_k^T g_k + \frac{1}{2} s_k^T Q(x_k) s_k \tag{10}$$

where $Q(x_k)$ is the Hessian matrix. The first derivative of the above equation can be written as:

$$\nabla f_{k+1} = g_k + Q(x_k) s_k \tag{11}$$

Thus, the curvature information in Eq. (10) can be approximated by

$$s_k^T Q(x_k) s_k = \frac{2}{3}\left(f_k - f_{k+1}\right) + \frac{2}{3} s_k^T Q(x_k) s_k \tag{12}$$

Because the updated B_{k+1} is supposed to approximate the $Q(x_k)$, it is reasonable to have

$$s_k^T B_{k+1} s_k = \frac{2}{3}\left(f_k - f_{k+1}\right) + \frac{2}{3} s_k^T Q(x_k) s_k \tag{13}$$

Using (11) in (13), we obtain:

$$s_k^T B_{k+1} s_k = \frac{2}{3} s_k^T y_k + \frac{2}{3}\left(f_k - f_{k+1}\right) \tag{14}$$

The new quasi-Newton (QN-) equation is given by:

$$s_k^T \tilde{y_k} = \frac{2}{3} s_k^T y_k + \frac{2}{3}\left(f_k - f_{k+1}\right) \tag{15}$$

From the above equation, the different gradients can be written as

$$B_{k+1} s_k = \tilde{y_k}, \tilde{y_k} = \frac{2}{3} y_k + \frac{2/3\left(f_k - f_{k+1}\right)}{s_k^T u_k} u_k \tag{16}$$

where u_k is a vector such that $s_k^T u_k \neq 0$. The BFGS update is modified based on the revised quasi-Newton equation. Alternatively, the vector u_k choices in Equation (16) can be expressed as:

(i) For $u_k = y_k$, Equation (16) becomes:

$$\tilde{y_k} = \frac{2}{3} y_k + \frac{2/3\left(f_k - f_{k+1}\right)}{s_k^T y_k} y_k.$$

(ii) For $u_k = g_k$, Equation (16) becomes:

$$\tilde{y_k} = \frac{2}{3} y_k + \frac{2/3\left(f_k - f_{k+1}\right)}{s_k^T g_k} g_k.$$

(iii) For $u_k = g_{k+1}$, Equation (16) becomes:

$$\tilde{y_k} = \frac{2}{3} y_k + \frac{2/3\left(f_k - f_{k+1}\right)}{s_k^T g_{k+`1}} g_{k+`1}.$$

From the above explanation of the results, we can write the algorithm as follows:

Stage 1: Let $x_0 \in R^n$, $k = 0$ and $H_0 = I$

Stage 2: If $\|g_k\| = 0$, stop.

Stage 3: Evaluate $d_k = -H_k g_k$.

Stage 4: Determine the optimal learning rate (step - size) by α_k using Eqs. (4) & (5).

Stage 5: Let $x_{k+1} = x_k + \alpha_k d_k$. Update H_{k+1} by using Equations (9) and (16) if $s_k^T \tilde{y_k} > 0$; otherwise, leave $H_{k+1} = H_k$.

Stage 6: Take $k = k + 1$, and then go to Stage 2.

The following theorem illustrates the theoretical benefits of the new quasi-Newton Equation (16). To ensure that the matrix B_{k+1} is positive definite, we need only prove that $s_k^T \tilde{y_k} > 0$ holds.

Theorem 1. Let matrix sequence B_{k+1} be generated using Equation (6). Thus, the sequence B_{k+1} is positive-definite.

Proof. From the different gradient definitions, we have:

$$s_k^T \tilde{y_k} = \frac{2}{3} y_k + \frac{2}{3}\left(f_k - f_{k+1}\right) \tag{17}$$

By applying Wolfe's condition to the previous equation, we obtain:

$$s_k^T \tilde{y_k} \geq \frac{2}{3}\left(s_k^T y_k - \delta g_k^T s_k\right) \tag{18}$$

Because $s_k^T y_k > 0$ and $-\delta g_k^T s_k > 0$, Eq. (18), we obtain

$$s_k^T \tilde{y_k} \geq 0 \tag{19}$$

Therefore, B_{k+1} is positive -definite.

3. Convergent analysis

We provide a global convergence of innovative approaches under circumstances that are comparatively understated.

1. The level was set to $L_0 = \{x \in R^n : f(x) \leq f(x_0)\}$ be convex.

2. Because the gradient satisfies the Lipschitz continuity, there is a positive constant called $L > 0$:

$$(\nabla f(\bar{x}) - \nabla f(x^+)) \leq L\|\bar{x} - x^+\|, \forall \bar{x}, x^+ \in L_0. \tag{20}$$

The series $\{x_k\}$ generated by a new algorithm is evident in S because $\{f_k\}$ is a decreasing series, and there is a constant f^* that results in

$$\lim_{k \to \infty} f_k = f^* \tag{21}$$

3. Let Q be a matrix from the 2^{nd} derivatives of the f. Then, there exist constants R and r, such that:

$$r\|z\|^2 \le z^T Qz \le R\|z\|^2 \tag{22}$$

for all $z \in R^n$, for more details see.[12–14]

Theorem 2. If $\{x_k\}$ is generated using the proposed algorithm. Then we have:

$$r\|s_k\|^2 \le s_k^T \tilde{y}_k \le R\|s_k\|^2. \tag{23}$$

and

$$\|\tilde{y}_k\| \le (L+R)\|s_k\|. \tag{24}$$

Proof: By different gradient definitions $\tilde{y}_k$ and combining Equations (10) with (16), we obtain:

$$s_k^T \tilde{y}_k = s_k^T Q(x_k) s_k = \frac{2}{3} s_k^T y_k + \frac{2}{3}(f_k - f_{k+1}) = 2(f_{k+1} - f_k) - 2s_k^T g_k. \tag{25}$$

Utilizing the mean value theorem and Taylor series, we obtain:

$$f_{k+1} = f_k + s_k^T g_k + \frac{1}{2} s_k^T Q(\eta_k) s_k \tag{26}$$

where $\xi \in (0,1)$ and $\eta_k = x_k + \xi(x_{k+1} - x_k)$. As such by Eqs. (25) and (26), as follows:

$$s_k^T \tilde{y}_k = 2\left(s_k^T g_k + \frac{1}{2} s_k^T Q(\eta_k) s_k\right) - 2s_k^T g_k = 2s_k^T g_k + s_k^T Q(\eta_k) s_k - 2s_k^T g_k = s_k^T Q(\eta_k) s_k \tag{27}$$

Meeting Assumption 3, it is simple to surmise:

$$r\|s_k\|^2 \le s_k^T \tilde{y}_k \le R\|s_k\|^2 \tag{28}$$

Then, we obtain different gradient definitions of $\tilde{y}_k$ by direct calculations:

$$\|\tilde{y}_k\| = \left\|\frac{2}{3} y_k + \frac{[2/3(f_k - f_{k+1})]}{s_k^T u_k} u_k\right\| \le \frac{2}{3}\|y_k\| + \frac{|[s_k^T Q(\eta_k) s_k - 2/3(s_k^T y_k)]|}{\|\delta_k\|\|u_k\|}\|u_k\| \le \frac{4}{3}\|y_k\| + \frac{|[s_k^T Q(\eta_k) s_k]|}{\|s_k\|} \le 4/3L\|s_k\| + R\|s_k\| \le (4/3L+R)\|s_k\| \tag{29}$$

The proof is finished.

Theorem 3. If the constants $a_1 > 0$ and $a_2 > 0$ exist, then the following inequality holds:

$$s_k^T B_k s_2 \ge a_2\|s_k\|^2, \text{ and } \|B_k s_k\| \le a_1\|s_k\| \tag{30}$$

for any k. The sequence $\{x_k\}$ is obtained using the new algorithm, and we obtain:

$$\lim_{k\to\infty} inf\,\|g_k\| = 0. \tag{31}$$

Proof: The proof is straightforward, similar to the proof of Theorem 3 in.[6]

In this study, we prove a global convergence theorem for non-convex problems and suggest a cautious updating strategy that is comparable to that mentioned previously. We state a Powell-related lemma for motivational purposes.[15]

Lemma 1. A smooth function f that is limited below can be treated using the BFGS technique if a constant $M > 0$ exists, which makes the inequality:

$$\|\tilde{y_k}\|^2 / s_k^T \tilde{y}_k \leq M \tag{32}$$

then:

$$\lim_{k\to\infty} inf\, \|g_k\| = 0. \tag{33}$$

Theorem 4. If these Assumptions hold, $\{x_k\}$ is generated by the new algorithm. Then Eq. (32) holds.

Proof: If Eq.(33) fails to hold, then there exists a constant $\varepsilon > 0$, such that:

$$\|g_k\| \geq \varepsilon. \tag{34}$$

Therefore, a constant $r > 0$ exists, such that:

$$r\|s_k\|^2 \leq s_k^T \tilde{y}_k. \tag{35}$$

So, combining Eqs. (29) and (35) imply that:

$$\|\tilde{y_k}\|^2 / s_k^T \tilde{y}_k \leq M. \tag{36}$$

The proof is finished.

4. Numerical experiments

In this section, we present a numerical comparison of QN -techniques and suggest modifications for solving 4th order nonlinear partial differential equations.

Example 1: Consider the nonlinear 4th order has the form;

$$\begin{gathered} u_{xt} - u_{xxxy} - 2u_{xx}u_y - 4u_x u_{xy} = 0; \\ u(x,y,0) = \frac{1}{2} sech^2\left(\frac{1}{2}(x+y)\right) \text{ and,} \\ exact\ solution\ u(x,y,z,t) = tanh\left(\frac{1}{2}(x+y-t)\right) \end{gathered}$$

The results of solving the above equation at different times t are presented in Table 1. The neural solution for this equation is shown in Figure 1.

We stopped utilizing the algorithms by employing Himmeblau's law[18]:

If $|f(x_k)| > 10^{-5}$, then $\frac{|f(x_k) - f(x_{k+1})|}{|f(x_k)|} = 1$. Otherwise, $|f(x_k) - f(x_{k+1})| = 1$. For every problem, if $\|g_k\| < \varepsilon$ is used, the program is terminated.

Quasi-Newton approaches perform better when an appropriate quasi-Newton equation is employed. The performance of the new update with $u_k = g_{k+1}$ was the best of the three methods, whereas the normal performance of the new update with $u_k = y_k$ and $u_k = g_k$ was somewhat better than that of the BFGS technique. As a result, among the QN -procedures for unconstrained problems, the new update with $u_k = g_{k+1}$ is the most well -organized.

Example 2: Consider the nonlinear 4th order has the form:

$$\begin{gathered} u_{tt} - u_{xx} - u_{xxxx} - u_{yy} - u_{zz} - 3\left(u^2\right)_{xx} = 0 \\ u(x,y,z,0) = \frac{1}{2}\mathrm{sech}^2\left(\frac{1}{2}(x+y+z)\right), u_t = \tanh\left(\frac{1}{2}(x+y+z)\right)\mathrm{sech}^2\left(\frac{1}{2}(x+y+z)\right) \end{gathered}$$

Table 1. **The results of suggested algorithm for different values of time t.**

X = y	ti exact	Suggested update				
		t = 0.001	t = 0.01	t = 0.05	t = 0.25	t = 0. 5
0	-0.000499999958333338	-0.000048659724380	-0.00499995832713615	-0.025004418506876	-0.124353001771672	-0.244918662401479
0.1	0.0991729368500791	0.099174522493650	0.0947152247011525	0.074859690643595	-0.0249947929685649	-0.148885033624227
0.2	0.196894751347250	0.196894751347288	0.192565398608004	0.173235732159165	0.0748596906873580	-0.0499583749589804
0.3	0.290854977351376	0.290854977351250	0.286730291373398	0.268271182008229	0.173235157834554	0.0499583749579298
0.4	0.379521061607639	0.379521061607816	0.375662661174346	0.358357398344881	0.268271160988048	0.148885033623492
0.5	0.461723842547565	0.461723842547454	0.458175852175461	0.442230453940485	0.358357335349861	0.244918662402002
0.6	0.536693682582613	0.536693686709420	0.533482128457157	0.519021833904887	0.442230290513323	0.336375352939167
0.7	0.604050311415608	0.604050311415511	0.601184473121516	0.588259256403465	0.519021833898177	0.421898609908564
0.8	0.663757149868171	0.663757149868364	0.661232203097477	0.649827607630977	0.588259256398005	0.500520211189160
0.9	0.716054324313046	0.716054380560282	0.713854553039899	0.703905603862037	0.649827419353020	0.571668985813867
1	0.761384088809508	0.761384088809337	0.759486275064505	0.750893283626045	0.703905603936521	0.635140845030389

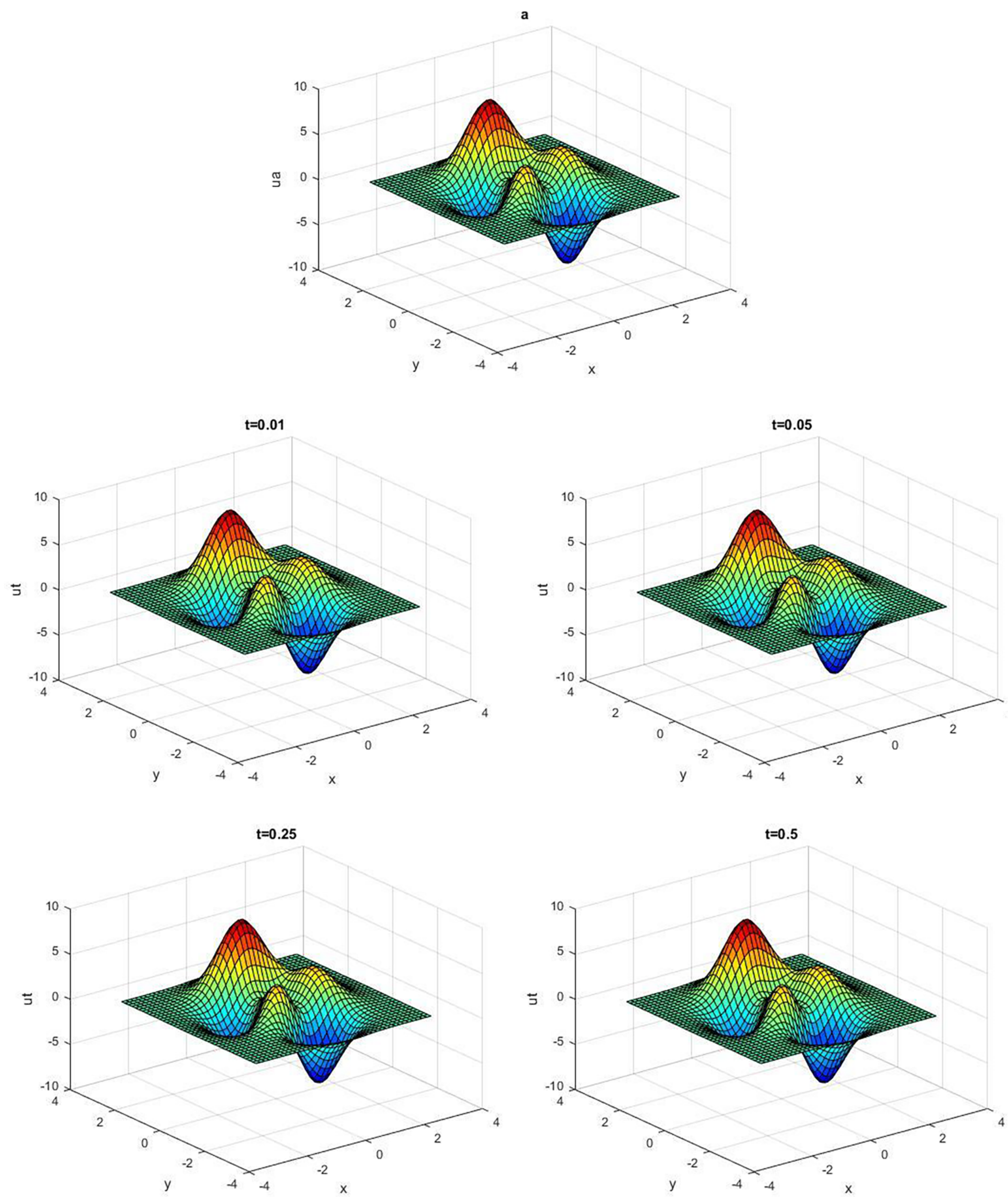

Figure 1. Illustration the results using new algorithm for different time t.

Exact solution:

$$u(x,y,z,t)=\frac{1}{2}\operatorname{sech}^2\left(\frac{1}{2}(x+y+z-2t)\right)$$

The neural solution for this equation is shown in Figure 2 when z = -0.5. The accuracy for epochs and time is presented in Table 2, and Table 3, illustrates the results of the neural solution of the equation.

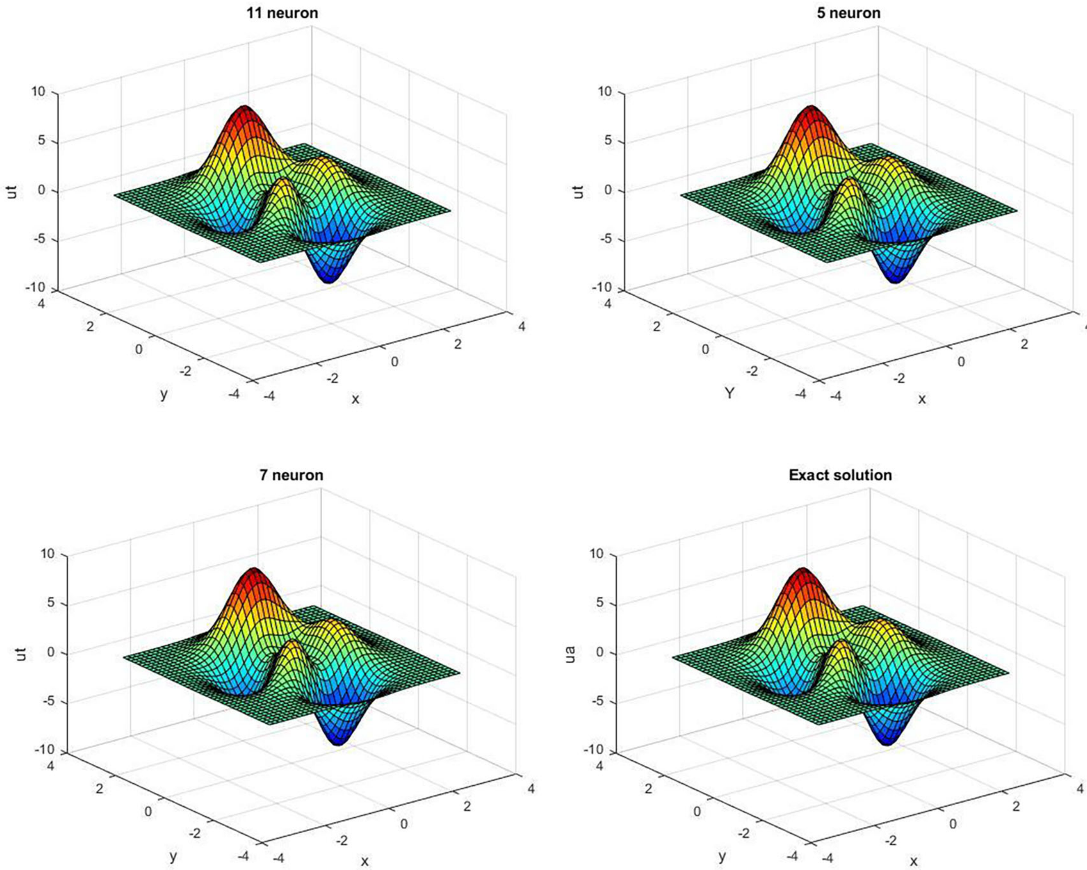

Figure 2. Solution for z = -1/2.

Table 2. Properties of the proposed algorithm for solving Example 1.

Train Function "Trainbfg"	Performance of train	Epoch	Time	Msereg
[t = 0.001]	4.72e-27	818	0:00:02	1.4903e-11
[t = 0.01]	7.27e-23	404	0:00:00	3.0524e-17
[t = 0.05]	9.34e-24	33	0:00:00	7.6100e-12
[t = 0.25]	2.64e-27	909	0:00:01	8.7302e-16
[t = 0. 5]	1.59e-24	593	0:00:01	5.4723e-12

Table 3. MSE and performance for training, validation, and testing for the solution of Example 2.

	LM	Suggested update BFG	SCG	RP
Time	00:00:39	00:00: 8	00:00:44	00:00:12
Best Epoch	1000	810	1000	1000
MSE	2.61912e-12	6.9328543e-17	2.9424106e-07	5.9553091e-06
Best training perf	2.694601e-12	6.694813e-14	2.21545518e-07	5.9894044e-06
Best validation perf	2.334575e-12	7.2694735e-16	1.996644e-07	5.7156087e-06
Best test perf	2.5514463e-12	7.7070942e-15	2.254638e-07	6.0358983e-06

5. Conclusions

In this study, we constructed improved BFGS quasi-Newton updating formulae by using the proposed robust QN -equation. Second-order information from Hessian's Hessian objective function Hessian's is used in this study to develop a novel quasi-Newton equation. Two nonlinear 4^{th} order example are provided to illustrate the accuracy of the suggested update, The results are consistent with the practical results and conform to the results that the suggested update, is globally convergent.

Data availability

No data were included in this study.

References

1. Hassan MA: **Estimate the Effect of Rainwaters in Contaminated Soil by Using Simulink Technique.** *JPCS.* 2018; **1003**: 012057. 1-6.
2. Jamil HJ, Al-Noor TH: **Hookah Smoking with Health Risk Perception of Different Types of Tobacco.** *JPCS.* 2020; **1664**(1): 012127. 1-11.
Publisher Full Text
3. Hussein NA: **Solitary Wave Solution of Zakharov-Kuznetsov Equation.** *AIP Conf. Proc.* 2022; **2398**(1): 060084. 1-6.
4. Ali MH: **Design optimal neural network for solving unsteady state confined aquifer problem.** *Mathematical Modelling of Engineering Problems.* 2023; **10**(2): 565–571.
Publisher Full Text
5. Kareem ZH: **Solving Three-Dimensional Groundwater Recharge Based on Decomposition Method.** *JPCS.* 2020; **1530**: 012068. 1-8.
Publisher Full Text
6. Al-Noor NH, Al-Noor TH: **Estimate the rate of contamination in Baghdad soils by using numerical method.** *JPCS.* 2019; **1294**(3): 032020. 1-10.
7. Khamas AH: **New approach for calculate exponential integral function.** *Iraqi Journal of Science.* 2023; **64**(8): 4034–4042.
8. Abood IN: **Persons' Camp Using Interpolation Method.** *JPCS.* 2018; **1003**: 012055. 1-9.
9. Enadi MO: **New approach for solving three-dimensional space partial differential equation.** *Baghdad Sci. J.* 2019; **16**(3): 786–792.
10. Ghazi FF: **New Approach for Solving Two- dimensional Spaces.** *PDE, JPCS.* 2020; **1530**: 012066. 1-9.
Publisher Full Text
11. Hussein NA: **Efficient approach for solving high-order (2+1) D-differential equation.** *AIP Conf. Proc.* 2022; **2398**: 060018. 1-7.
12. Kareem ZH: **Recent modification of decomposition method for solving wave-like equations.** *Journal of Interdisciplinary Mathematics.* 2023; **26**(5): 809–820.
Publisher Full Text
13. Salih OM: **Design neural network based upon decomposition approach for solving reaction diffusion equation.** *JPCS.* 2019; **1234**(1): 012104.
Publisher Full Text
14. Hussein NA: **New Approach for Solving (1+1)-Dimensional Differential Equation.** *JPCS.* 2020; **1530**: 012098. 1-10.
Publisher Full Text
15. Kareem ZH: **Efficient Modification of Decomposition Method for Solving a System of PDEs.** *Iraqi Journal of Science.* 2021; **62**(9): 3061–3070.
16. Ali MH, Tawfiq LNM: **Novel Neural Network Based on New Modification of BFGS Update Algorithm for Solving Partial Differential Equations.** *Advances in the Theory of Nonlinear Analysis and its Applications.* 2023; **7**(4): 76–88.
17. Hussein NA: **Efficient Approach for Solving (2+1) D- Differential Equations.** *Baghdad Sci. J.* 2023; **20**(1): 166–174.
18. Kareem ZH: **New soliton solution of nonlinear (3+1) D-mKd, (3+1) D-KP, and (3+1)D-gKP equation.** *AIP Conference Proceedings.* 2024; **3036**(1): 040034. 1-12.
19. Khamas AH: **Determine the Effect Hookah Smoking on Health with Different Types of Tobacco by using parallel processing technique.** *JPCS.* 2021; **1818**: 012175. 1-10.
Publisher Full Text
20. Ghazi FF: **Design optimal neural network based on new LM training algorithm for solving 3D-PDEs.** *An International Journal of Optimization and Control: Theories & Applications.* 2024; **14**(3): 249–260.
Publisher Full Text
21. Hussein NA: **Exact solutions for systems of nonlinear (2+1)D-Differential Equations.** *Iraqi Journal of Science.* 2022; **63**(10): 4388–4396.
22. Helal MM: **Double LA transform and their properties for solving partial differential equations.** *AIP Conference Proceedings.* 2023; **2834**(1): 080140. 1-10.
23. Ibrahim AQ: **Efficient method for solving fourth-order PDEs.** *JPCS.* 2021; **1818**: 012166. 1-9.
24. Kareem ZH: **Solving (3+1) D-New Hirota Bilinear Equation Using Tanh Method and New Modification of Extended Tanh Method.** *Advances in the Theory of Nonlinear Analysis and its Applications.* 2023; **7**(4): 114–122.
25. Ali MH: **Efficient design of neural networks for solving third-order partial differential equations.** *JPCS.* 2020; **1530**(1): 012067.
Publisher Full Text
26. Hussein NA: **Exact soliton solution for systems of nonlinear (2+1) D-DEs.** *AIP Conference Proceedings.* 2023; **2834**(1): 080137.
27. Kareem ZH: **New modification of decomposition method for solving high-order strongly nonlinear partial differential equations.** *AIP Conference Proceedings.* 2022; **2398**: 0600363. 1-9.
28. Mohammed MA, Tawfiq LNM: **New accurate method for solving fourth-order two-dimension evolution equations.** *Journal of Interdisciplinary Mathematics.* 2025; **28**(3): 1269–1276.
Publisher Full Text
29. Hussein NA: **New Approach for Solving (2+1)-Dimensional Differential Equation.** *JPCS.* 2021; **1818**(1): 012182.
Publisher Full Text
30. Ghazi FF: **Modeling the Contamination of Soil Adjacent to Mohammed Al-Qassim Highway in Baghdad, Iraqi Journal of.** *Science.* 2020; Vol. **61**(10): pp: 2663–2670. (2020).
Publisher Full Text
31. Thirthar AA, Shah K, Abdeljawad T: **Design an efficient neural network for solving steady state problems.** *J. Math. Comput. Sci.* 2025; **36**(1): 84–98.
Publisher Full Text
32. Ali MH: **Efficient design of neural network based on new BFGS training algorithm for solving ZK-BBM equation.** *AIP Conference Proceedings.* 2024; **3036**(1): 040009.
33. Abdul-Jabbar SA: **Design Efficient Neural Network for Solving 2D-Nonlinear Wave-Like Equations.** *AIP Conference Proceedings.* 2025; **3264**(1): 050055.
34. Ghazi FF: **Solving 5th Order nonlinear 4D-PDEs Using Efficient Design of Neural Network.** *AIP Conference Proceedings.* 2025; **3264**(1): 050042.