AIMS Mathematics, 11(8): 26273–26300.
DOI: 10.3934/math.20261054
Received: 05 June 2026
Revised: 10 August 2026
Accepted: 13 August 2026
Published: 24 August 2026

https://www.aimspress.com/journal/Math

Research article

A higher-order iterative scheme and its integration with the Grey Wolf Optimization algorithm for solving complex nonlinear systems

Aanchal Chandel[1], **Sonia Bhalla**[1], **Gurjeet Singh**[2], **Alicia Cordero**[3] **and Juan R. Torregrosa**[3,*]

[1] Department of Mathematics, Chandigarh University, Gharuan, Mohali, 140413, Punjab, India
[2] Department of Mathematics, RIMT University, Mandi, Gobindgarh, 147301, Punjab, India
[3] Department of Mathematics, Universitat Politècnica de València, 46022, Valencia, Spain

* **Correspondence:** Email: jrtorre@mat.upv.es.

Abstract: This study proposes a novel iterative scheme for an effective solution of systems of nonlinear equations (SoNE) that improves the order of convergence of the classical methods. Along with the proposed deterministic scheme, a hybrid computational framework is produced by combining the proposed iterative technique with the Grey Wolf Optimization (GWO) algorithm. The goal of integration is to combine the iterative method's strong local search ability with the metaheuristic optimization technique's global search ability. This will make the process more robust and less sensitive to initial guesses. To determine how reliable and efficient the proposed iterative scheme is, we look at its theoretical convergence order. Additionally, a comprehensive statistical analysis is tested to assess the efficacy of the proposed methods regarding accuracy, convergence rate, and stability. Numerical experiments are conducted on various nonlinear problems to validate the efficacy of the methodologies. To demonstrate their practical utility, the proposed methods are employed on real-world chemical engineering models characterized by nonlinear dependence among process variables. The computational results show that the proposed methods produce accurate numerical solutions with a better convergence order.

Keywords: Newton-type methods; metaheuristic techniques; convergence analysis; nonlinear system
Mathematics Subject Classification: 65H05, 65G99

1. Introduction

Determining the solutions of systems of nonlinear equations (SoNE) is one of the hardest and crucial problems in mathematics, engineering, economics, and applied sciences. A nonlinear system is important because it can be used to model many real-world phenomena across many fields, such as neuropsychology, combustion processes, chemical equilibrium analysis, and the generation and

distribution of electric power [1]. These practical issues are typically expressed as a SoNE in the following manner:

$$G(u) = 0, \tag{1.1}$$

where $G : \Omega \subseteq \mathbb{R}^m \to \mathbb{R}^m$, $G(u) = (g_1(u), g_2(u), g_3(u), \ldots, g_m(u))^T$ such that the coordinate functions g_i, $i = 1, 2, \ldots, m$, are nonlinear in variables $u = (u_1, u_2, u_3, \ldots, u_m)^T$. In numerous cases, obtaining the precise solution of (1.1) via analytical methods is either not feasible or highly complex. As a result, researchers have used different iterative methods to obtain the solutions of nonlinear system. The Newton–Raphson method (NM) [2] is the simplest and most widely used of these methods. It can be written as follows:

$$u^{(n+1)} = u^{(n)} - (G'(u^{(n)}))^{-1} \cdot G(u^{(n)}), \tag{1.2}$$

where $G'(u^{(n)})$ is the Jacobian matrix of function G evaluated at $u^{(n)}$.

Newton's method has quadratic convergence, but it also has some disadvantages. For example, it is sensitive to the initial approximation, and it may not work when $G'(u)$ is almost singular. To rectify these deficiencies and enhance convergence order, numerous researchers have proposed for higher-order iterative techniques. Halley's [3, 4], Householder's [5], Ostrowski's [6], Traub's [7], and Jarratt's [8] methods are some of the most important ones that help solve nonlinear problems faster. Over the decades, several higher-order techniques have been constructed which are capable of rapidly solving SoNE. Examples of these types of methods include [7, 9–11], which are multistep iterative methods, predictor–corrector schemes, and other modified Newton-type algorithms that are meant to speed up convergence.

The SoNE represents many real-life applications with high complexity. These problems may exhibit nondifferentiable or discontinuous functions, and potentially fall into polynomial time hard or nondeterministic classes [12]. As a result, the solution of these systems often requires high memory resources and considerable computational cost. Applying traditional iterative techniques to such complex problems can be computationally expensive, prompting the development of metaheuristic algorithms as alternative solution strategies [13]. For the past couple of decades, researchers designed a variety of metaheuristic algorithms.

In the last few decades, researchers have achieved great progress in developing many metaheuristic algorithms. Metaheuristic techniques are computational techniques that are inspired by various natural principles; human behavior; hunting and foraging behavior of animals, birds, insects, mammals etc. Some examples of metaheuristic techniques are the following: particle swarm optimization (PSO) [14], Harris hawk optimization (HHO) [15], the Butterfly Optimization algorithm (BOA) [16], and the Walrus Optimization algorithm (WOA) [17]. Usually, those algorithms work by three main phases: initialization, exploration, and exploitation. The initial phase generates an initial population that forms the basis of the search process. Exploration helps to discover the different fields of the search space, and exploitation relies on refining candidate solutions to achieve the optimal solution. To increase the performance of such optimization methods and obtain more accurate solutions, several modifications and improvements have been proposed. One common strategy is the integration of two algorithms to develop a combined approach that preserves the advantages of each method while reducing their individual limitations. Examples of such population-based integrated approaches include the artificial bee colony–based combined algorithm [18], PSO integrated with the Genetic Algorithm [19], and several other variants [20, 21]. Although such combined algorithms generally improve the accuracy of

solutions for nonlinear systems, but several issues such as entrapment in local minima and premature convergence may still occur. The entrapment in local minima refers to the tendency of an algorithm to get stuck at a suboptimal solution, and premature convergence occurs when the algorithm converges too early without sufficient exploration of the search space, potentially missing the global optimum solution of the problem.

To address these limitations, Karr et al. in 1998 [22] first introduced an integrated framework that combined the Genetic Algorithm (GA) with NM, where the GA was used to generate suitable initial approximations for NM. Later, Lou et al. in 2008 [23] proposed a combined strategy incorporating the quasi-Newton method with the Chaos Optimization algorithm to solve complex nonlinear equations. More recently, Sihwail et al. in 2021 [24] introduced a combined method known as NHHO by integrating NM with the Harris Hawk Optimization algorithm to solve general SoNE. In 2023, Solaiman et al., [25] constructed a modified combined technique that integrates NM with the Sperm Swarm Optimization algorithm, yielding better solutions in fewer iterations. Subsequently, Zafar et al. in 2024, [26] proposed a Jarratt–PSO integrated method to address power flow analysis problems. Recently, Chandel et al. [27] in 2025 proposed a hybrid method using the Walrus Optimization algorithm.

Motivation: In order to clarify the gap that the proposed method fills, let us remark that, usually in iterative methods, the initial guess plays a significant role in convergence; even high-order methods may diverge if a proper initial guess is not available. Moreover, classical iterative methods depend on local search techniques, which may not guarantee global search capabilities starting from poor initial guesses. On the contrary, some of the metaheuristic methods may possess strong global search capabilities; however, they may face problems of slow convergence and low accuracy near the optimal solution. Our aim is to avoid these problems hybridizing iterative methods and some of the metaheuristic methods. In this manuscript, a new fourth-order iterative scheme has been developed and integrated with the Grey Wolf Optimization algorithm. The proposed framework is expected to reduce initial guess requirements and increase the reliability of the method along with accuracy; thus, problems such as divergence, premature convergence, and entrapment into local optima can be avoided in a more efficient and novel way.

The summary of this paper is as follows: The background and motivation for solving SoNE, along with a brief overview of existing approaches in the literature, is described in Section 1. The construction of the proposed fourth–order iterative technique and its integration with the Grey Wolf Optimization algorithm is discussed in Section 2. Section 3 reports the experimental work carried out to evaluate the efficiency of the proposed methods on several benchmark nonlinear systems. It also provides the convergence and statistical analysis of the developed iterative scheme to check its theoretical convergence properties. Finally, Section 4 shows the concluding remarks and summarizes the main conclusions of the study.

2. Proposed algorithms

In this section, the proposed iterative and integrated Grey Wolf Optimization algorithm are discussed.

2.1. Fourth-order iterative method (FM)

In this section, we develop a two-step iterative method for solving system of nonlinear equations and also establish its fourth-order convergence. The newly constructed two-step iterative scheme is defined as:

$$\begin{cases} v^{(n)} = u^{(n)} - \frac{2}{3}\left(G'(u^{(n)})\right)^{-1} G(u^{(n)}), \\ u^{(n+1)} = v^{(n)} + \frac{1}{24}\left[I - 9\left((G'(v^{(n)}))^{-1}G'(u^{(n)})\right)^2\right](G'(u^{(n)}))^{-1}G(u^{(n)}), \end{cases} \tag{2.1}$$

where I is the identity matrix of size $m \times m$, and $G'(u^{(n)})$ and $G'(v^{(n)})$ are the Jacobian matrix evaluated at $u^{(n)}$ and $v^{(n)}$, respectively.

To analyze the convergence of the proposed method, we need some tools and procedures introduced in [28]. Let $G : \Omega \subseteq \mathbb{R}^m \to \mathbb{R}^m$ be sufficiently differentiable in an open neighborhood Ω. The lth derivative of G at $k \in \mathbb{R}^m$, $l \geq 1$, is the l-linear function $G^{[l]}(k) : \mathbb{R}^m \times \cdots \times \mathbb{R}^m \to \mathbb{R}^m$ such that $G^{[l]}(k)(w_1, \ldots, w_l) \in \mathbb{R}^m$. It is easy to observe that

1. $G^{[l]}(k)(w_1, \ldots, w_{l-1}, \cdot) \in \mathcal{L}(\mathbb{R}^m)$.
2. $G^{[l]}(k)(w_{\sigma(1)}, \ldots, w_{\sigma(l)}) = G^{[l]}(k)(w_1, \ldots, w_l)$, for all permutations σ of $\{1, 2, \ldots, l\}$.

From the above properties, we can use the following notation:

(a) $G^{[l]}(k)(w_1, \ldots, w_l) = G^{[l]}(k)w_1, \ldots, w_l$.

(b) $G^{[l]}(k)w^{l-1}G^{[p]}(k)w^p = G^{[l]}(k)G^{[p]}(k)w^{p+l-1}$.

On the other side, for $(\xi + h) \in \mathbb{R}^m$ lying in a neighborhood of a solution ξ of $G(u) = 0$, we can apply Taylor's series expansion, and assuming that the Jacobian matrix $G'(\xi)$ is nonsingular, we have

$$G(\xi + h) = G'(\xi)\left[h + \sum_{l=2}^{p-1} C_l h^l + O(h^p)\right], \tag{2.2}$$

where $C_l = 1/l!(G'(\xi))^{-1}G^{[l]}(\xi)$ for $l \geq 2$. It is clear that $C_l h^l \in \mathbb{R}^m$, as $G^{[l]}(\xi) \in \mathcal{L}(\mathbb{R}^m \times \cdots \times \mathbb{R}^m, \mathbb{R}^m)$, and $(G'(\xi))^{-1} \in \mathcal{L}(\mathbb{R}^m)$. In addition, G' can be expressed as

$$G'(\xi + h) = G'(\xi)\left[I + \sum_{l=2}^{p-2} lC_l h^{l-1} + O(h^{p-1})\right], \tag{2.3}$$

where I is the identity matrix of size $m \times m$. Therefore, $lC_l h^{l-1} \in \mathcal{L}(\mathbb{R}^m)$. From Eq (2.3), we have

$$(G'(\xi + h))^{-1} = \left(I + B_2 h + B_3 h^2 + B_4 h^3 + B_5 h^4 + \cdots + O(h^p)\right)(G'(\xi))^{-1}, \tag{2.4}$$

where

$$\begin{aligned} B_2 &= -2C_2, \\ B_3 &= 4C_2^2 - 3C_3, \\ B_4 &= -4C_4 + 6C_2C_3 + 6C_3C_2 - 8C_2^3, \\ &\vdots \end{aligned}$$

obtained from $(G'(\xi + h))^{-1}G'(\xi + h) = I$.

Let $e^{(n)} = u^{(n)} - \xi$ be the error at the nth iteration. The equation $e^{(n+1)} = M(e^{(n)})^p + O((e^{(n)})^{p+1})$, where M is a p-linear function $M \in \mathcal{L}(\mathbb{R}^m \times \cdots \times \mathbb{R}^m, \mathbb{R}^m)$, is called the error equation, with p its convergence order.

Theorem 1. *Let $G : \Omega \subseteq \mathbb{R}^m \to \mathbb{R}^m$ be a function that is sufficiently differentiable in an open convex neighborhood Ω containing a zero ξ of G, and assume that $G'(u)$ is nonsingular at $u = \xi$ and the initial guess $u^{(0)}$ is sufficiently close to ξ. Then, the iterative scheme defined by* (2.1) *exhibits fourth-order convergence with the error equation*

$$e^{(n+1)} = \frac{1}{9}(13C_2^3 - 10C_2C_3 + C_3C_2 + C_4)(e^{(n)})^4 + O((e^{(n)})^5), \tag{2.5}$$

where the following notation is used:

- $C_j = \left(\frac{1}{j!}\right)(G'(\xi))^{-1}G^{[j]}(\xi), \quad j \geq 2\,,$
- $u^{(n)} - \xi = e^{(n)}$.

Also, in the proof, A_j and B_j, $j \geq 2$ denote the coefficients of the Taylor expansion of $G'(v^{(n)})^{-1}$ and $G'(u^{(n)})^{-1}$, respectively.

Proof. For calculating the error equation of first step of Eq (2.1), $v^{(n)} = u^{(n)} - (2/3)\left(G'(u^{(n)})\right)^{-1} G(u^{(n)})$, we perform the Taylor series of $G(u^{(n)})$ and $G'(u^{(n)})$ around ξ as follows:

$$G(u^{(n)}) = G'(\xi)\Big[e^{(n)} + C_2(e^{(n)})^2 + C_3(e^{(n)})^3 + C_4(e^{(n)})^4 + O\left(\left(e^{(n)}\right)^5\right)\Big], \tag{2.6}$$

$$G'(u^{(n)}) = G'(\xi)\Big[I + 2C_2e^{(n)} + 3C_3(e^{(n)})^2 + 4C_4(e^{(n)})^3 + 5C_5(e^{(n)})^4 + O\left(\left(e^{(n)}\right)^5\right)\Big]. \tag{2.7}$$

Assuming that the Jacobian matrix $G'(\xi)$ is nonsingular, we derive the Taylor series expansion of $(G'(u_n))^{-1}$ as follows:

$$\left(G'\left(u^{(n)}\right)\right)^{-1} = \left[I + B_2e^{(n)} + B_3\left(e^{(n)}\right)^2 + B_4\left(e^{(n)}\right)^3 + O\left(\left(e^{(n)}\right)^4\right)\right](G'(\xi))^{-1}\,, \tag{2.8}$$

where $B_2 = -2C_2$, $B_3 = 4C_2^2 - 3C_3$, and $B_4 = 6C_3C_2 + 6C_2C_3 - 4C_4 - 8C_2^3$, have been determined from the following:

$$\left(G'(u^{(n)})\right)^{-1} G'(u^{(n)}) = I.$$

Therefore,

$$\begin{aligned}\left(G'(u^{(n)})\right)^{-1} = \Big[&I - 2C_2e^{(n)} + (4C_2^2 - 3C_3)(e^{(n)})^2 \\ &+ (6C_3C_2 + 6C_2C_3 - 4C_4 - 8C_2^3)(e^{(n)})^3 + O((e^{(n)})^4)\Big](G'(\xi))^{-1}\,.\end{aligned} \tag{2.9}$$

By using (2.6) and (2.9), one can obtain

$$\begin{aligned}\left(G'(u^{(n)})\right)^{-1} G(u^{(n)}) = &e^{(n)} - C_2(e^{(n)})^2 + 2(C_2^2 - C_3)(e^{(n)})^3 \\ &+ (3C_3C_2 + 4C_2C_3 - 3C_4 - 4C_2^3)(e^{(n)})^4 + O((e^{(n)})^5).\end{aligned} \tag{2.10}$$

The first step of proposed scheme (2.1) can be written as

$$v^{(n)} - \xi = u^{(n)} - \xi - \frac{2}{3}\left(G'(u^{(n)})\right)^{-1} G(u^{(n)}). \tag{2.11}$$

In view of (2.10), Eq (2.11) yields

$$v^{(n)} - \xi = \frac{e^{(n)}}{3} + \frac{2}{3}C_2(e^{(n)})^2 - \frac{4}{3}(C_2^2 - C_3)(e^{(n)})^3 + \left(\frac{8C_2^3}{3} - \frac{8C_2C_3}{3} - 2C_3C_2 + 2C_4\right)(e^{(n)})^4 + O((e^{(n)})^5). \tag{2.12}$$

The Taylor's expansion of $G'(v^{(n)})$ around ξ can be evaluated similarly to Eq (2.7):

$$\begin{aligned} G'(v^{(n)}) &= G'(\xi)\Big[I + 2C_2(v^{(n)} - \xi) + 3C_3(v^{(n)} - \xi)^2 + 4C_4(v^{(n)} - \xi)^3 + O\left(\left(v^{(n)} - \xi\right)^4\right)\Big] \\ &= G'(\xi)\Big[I + \frac{2C_2}{3}e^{(n)} + \left(\frac{4}{3}C_2^2 - \frac{C_3}{3}\right)(e^{(n)})^2 + \left(\frac{8}{3}C_2^3 - \frac{8}{3}C_2C_3 + \frac{4}{3}C_3C_2 \right. \\ &\quad \left. + \frac{4}{27}C_4\right)(e^{(n)})^3\Big] + O((e^{(n)})^4). \end{aligned} \tag{2.13}$$

Assuming that the Jacobian matrix $G'(\xi)$ is nonsingular, we derive the Taylor series expansion of $\left(G'(v^{(n)})\right)^{-1}$ as follows:

$$\left(G'\left(v^{(n)}\right)\right)^{-1} = \left[I + A_2e^{(n)} + A_3\left(e^{(n)}\right)^2 + A_4\left(e^{(n)}\right)^3 + O\left(\left(e^{(n)}\right)^4\right)\right](G'(\xi))^{-1}, \tag{2.14}$$

where $A_2 = (-2/3)C_2$, $A_3 = (8/9)C_2^2 - (1/3)C_3$, and $A_4 = (1/27)(64C_2^3 - 60C_2C_3 - 24C_3C_2 - C_4)$.

By using these values, Eq (2.14) will become

$$\begin{aligned} \left(G'(v^{(n)})\right)^{-1} &= \Big[I - \frac{2C_2}{3}e^{(n)} + \left(\frac{8C_2^2}{9} - \frac{C_3}{3}\right)(e^{(n)})^2 + \frac{1}{27}(64C_2^3 - 60C_2C_3 - 24C_3C_2 - C_4)(e^{(n)})^3 + O((e^{(n)})^4)\Big] \\ &\quad (G'(\xi))^{-1}. \end{aligned} \tag{2.15}$$

Now, using Eqs (2.7) and (2.15), we get

$$\begin{aligned} \left(G'(v^{(n)})\right)^{-1} G'(u^{(n)}) &= I + \frac{4C_2e^{(n)}}{3} - \left(\frac{20C_2^2}{9} - \frac{8C_3}{3}\right)(e^{(n)})^2 \\ &\quad + \left(\frac{16}{27}C_2^3 - \frac{38}{9}C_2C_3 - \frac{14}{9}C_3C_2 + \frac{104}{27}C_4\right)(e^{(n)})^3 \\ &\quad + \left(\frac{160}{81}C_2^4 - \frac{124}{81}C_2^2C_3 - \frac{332}{81}C_2C_3C_2 - \frac{104}{81}C_3C_2^2 \right. \\ &\quad \left. - \frac{32}{9}C_3^2 + \frac{416}{81}C_2C_4 + \frac{16}{81}C_4C_2 + \frac{400}{81}C_5\right)(e^{(n)})^4 + O((e^{(n)})^5). \end{aligned} \tag{2.16}$$

By using (2.16), one can obtain the following:

$$\begin{aligned} I - 9\left(\left(G'(v^{(n)})\right)^{-1} G'(u^{(n)})\right)^2 &= -8I - 24C_2e^{(n)} + 24\left(C_2^2 - 2C_3\right)(e^{(n)})^2 \\ &+ \left(\frac{400}{3}C_2^3 + 44C_2C_3 + 100C_3C_2 - \frac{208}{3}C_4\right)(e^{(n)})^3 \\ &+ \Big(- \frac{448}{9}C_2^4 + \frac{1360}{9}C_2^2C_3 + \frac{920}{9}C_2C_3C_2 + \frac{728}{9}C_3C_2^2 + 80C_3^2 \\ &- \frac{1040}{9}C_2C_4 - \frac{80}{3}C_4C_2 - \frac{800}{9}C_5\Big)(e^{(n)})^4 + O((e^{(n)})^5). \end{aligned} \tag{2.17}$$

Substituting Eqs (2.10) and (2.17) into the second substep of scheme (2.1), we obtain the error equation as follows:

$$e^{(n+1)} = u^{(n+1)} - \xi = \frac{1}{9}(13C_2^3 - 10C_2C_3 + C_3C_2 + C_4)(e^{(n)})^4 + O((e^{(n)})^5), \tag{2.18}$$

which confirms the fourth-order convergence of the vectorial iterative scheme. □

2.2. Integrated Grey Wolf Optimization algorithm (IGWOA)

The integrated method combining the proposed fourth–order numerical method (2.1) with the Grey Wolf Optimization (GWO) algorithm is employed to solve SoNE and real–world applications. The Grey Wolf Optimization algorithm is defined as follows.

2.2.1. Grey Wolf Optimization algorithm (GWO)

In 2014, Seyedali Mirjalili et al. [29] introduced the GWO algorithm. It is a metaheuristic algorithm which is based on the hunting strategy and leader hierarchy of grey wolves in the form of population-based metaheuristics algorithm. Grey wolves like living in groups, and it is a habit of these wolves to have a pack of five to twelve members. All members of the group abide by a highly strict social dominance hierarchy. The exploration stage in GWO involves searching prey, and the exploitation stage involves surrounding the prey. Attacking and hunting plans are shown in Eqs (2.19) and (2.20), respectively:

$$\begin{aligned} D &= |C \cdot u_p^{(n)} - u^{(n)}|, \\ u^{(n+1)} &= u_p^{(n)} - A \cdot D, \end{aligned} \tag{2.19}$$

where,

$$\begin{aligned} A = 2a\, d_1 - a, C = 2d_2 \\ D_p &= |C_1 \cdot u_p - u|, \\ D_q &= |C_1 \cdot u_q - u|, \\ D_r &= |C_1 \cdot u_r - u|, \\ u_1 &= u_p - A_1 \cdot D_p, \\ u_2 &= u_q - A_2 \cdot D_q, \\ u_3 &= u_r - A_3 \cdot D_r. \end{aligned} \tag{2.20}$$

Here, D represents the distance vector between the grey wolf and the prey; p, q, and r represent the top three candidate solutions identified across the entire search domain; and n denotes the current iteration number. The position of the prey is given by the vector $u^{(n)}$, and C and A are coefficient vectors. The position vector of a grey wolf is denoted by u. The components of a decrease linearly from 2 to 0, where the random vectors d_1 and d_2 belong to the interval [0, 1]. All the operations described in (2.19) and (2.20) are pointwise.

Finally, the solution in GWO is updated as

$$u_{\mathrm{GWO}}^{(n+1)} = \frac{u_1 + u_2 + u_3}{3}. \tag{2.21}$$

The procedure above is continued until the condition $n \leq N$, where N is the maximum number of iterations, is satisfied. Ultimately, one gets the optimal solution and its fitness when the grey wolf completes its hunt by catching the prey.

By incorporating the fast local exploitation capability of the iterative scheme with the global exploration capability of GWO, the proposed approach efficiently searches complex and high–dimensional solution spaces while reducing the possibility of being trapped in local optima. The algorithm for the proposed Integrated Grey Wolf Optimization algorithm (IGWOA) method is described as follows.

The stepwise algorithm for the constructed IGWOA algorithm is presented in Algorithm 1.

The procedure begins with the selection of parameters such as the population size (T), maximum number of iterations (N), and the upper and lower initial intervals. An initial population is generated randomly, and the corresponding objective function values $\|G(u^{(n)})\|_2$ are evaluated. In this framework, grey wolves act as the search agents, and their positions are updated during the exploration phase and the exploitation phase using Eqs (2.19)–(2.21). This updating process is repeated for all grey wolves, resulting in their new positions obtained through the GWO mechanism. After each iteration, the proposed fourth–order iterative technique is applied as a refinement step using Eq (2.1). The fitness value corresponding to the refined solution obtained from the iterative method is then evaluated and compared with the fitness value produced by the GWO algorithm. The updated value having better fitness is selected for the next iteration. The procedure continues until the stopping criterion of the proposed scheme $n \leq N$ is satisfied. The final optimal solution obtained through IGWOA represents the best candidate solution.

Algorithm 1 : Integrated Grey Wolf Optimization Algorithm (IGWOA)

BEGIN

Assign all the parameters of the GWO.
Set the population size of grey wolves T and the maximum number of iterations N.
Initialize the positions of the grey wolves $u_i^{(0)}$, $(i = 1, 2, \ldots, T)$.
Evaluate the fitness of each wolf using $\left\|G(u_i^{(0)})\right\|_2$.
Identify the best GWO candidate and assign it as $u_{\mathrm{GWO}}^{(0)}$.
Initialize the parameters a, A, and C.
for $n = 0, 1, \ldots, N-1$ **do**
  Update the GWO parameters a, A, and C.
  for $i = 1, 2, \ldots, T$ **do**
    Calculate the updated positions of the grey wolves $u_{1,i}^{(n+1)}$, $u_{2,i}^{(n+1)}$, and $u_{3,i}^{(n+1)}$ using (2.19) and (2.20).
    Generate the GWO candidate $u_{i,\mathrm{GWO}}^{(n+1)}$ using (2.21).
    Evaluate the fitness $\left\|G(u_{i,\mathrm{GWO}}^{(n+1)})\right\|_2$.
  end for
  Identify the best candidate among the updated GWO population and denote it by $u_{\mathrm{GWO}}^{(n+1)}$.
  Apply the fourth-order iterative method (2.1) to $u_{\mathrm{GWO}}^{(n+1)}$ to obtain the locally refined candidate $u_{\mathrm{LR}}^{(n+1)}$.
  Calculate the fitness values

$$f_{\mathrm{GWO}} = \left\|G(u_{\mathrm{GWO}}^{(n+1)})\right\|_2, \qquad f_{\mathrm{LR}} = \left\|G(u_{\mathrm{LR}}^{(n+1)})\right\|_2.$$

  if $f_{\mathrm{LR}} < f_{\mathrm{GWO}}$ **then**
    Accept the locally refined candidate:

$$u_{\mathrm{acc}}^{(n+1)} = u_{\mathrm{LR}}^{(n+1)}.$$

  else
    Retain the GWO candidate:

$$u_{\mathrm{acc}}^{(n+1)} = u_{\mathrm{GWO}}^{(n+1)}.$$

  end if
  Update the best solution using $u_{\mathrm{acc}}^{(n+1)}$.
  Update the population for the next GWO iteration using the accepted candidate.
end for
Output: Best accepted solution and its fitness value.

END

3. Experimental setup and results

In this section, the performance of the proposed methods, namely FM and IGWOA, is evaluated on different SoNE. These problems are widely used in the literature, making them suitable for comparative analysis. The experimental conclusions obtained by the proposed methods are compared with known

iterative techniques such as the Cordero et al. method (CM) [30], the Kansal et al. method (KM) [31], and Jarratt's scheme (JM) [8].

Jarratt's method [8] is a major variation of Newton's method. It is a two-step iterative scheme along with fourth-order convergence. The scheme is described as follows:

$$\begin{cases} v^{(n)} & = u^{(n)} - \frac{2}{3}G'\left(u^{(n)}\right)^{-1} G\left(u^{(n)}\right), \\ u^{(n+1)} & = u^{(n)} - \left[I - \frac{3}{2}\left(3G'\left(v^{(n)}\right) - G'\left(u^{(n)}\right)\right)^{-1}\left(G'\left(v^{(n)}\right) - G'\left(u^{(n)}\right)\right)\right] \\ & \qquad \dot{G}'\left(u^{(n)}\right)^{-1} G\left(u^{(n)}\right), \end{cases} \tag{3.1}$$

where $G'(u^{(n)})$ and $G'(v^{(n)})$ are described as the Jacobian matrix evaluated at $u^{(n)}$ and $v^{(n)}$, respectively.

In 2021, Kansal et al. [31] proposed an improvement to the Chebyshev–Halley–type methods fourth–order by introducing two points. The iterative formula is described by (3.2):

$$\begin{cases} v^{(n)} = u^{(n)} - \dfrac{2}{3}\,(G'(u^{(n)}))^{-1}G(u^{(n)}), \\ u^{(n+1)} = u^{(n)} - \eta(u^{(n)})^{-1}\mu(u^{(n)})(G'(u^{(n)}))^{-1}G(u^{(n)}), \end{cases} \tag{3.2}$$

where

$$\begin{aligned} \mu(u^{(n)}) &= A_1 I + 3A_2\Omega(u^{(n)}) + 9A_3\Omega(u^{(n)})^2 - 27A_4\Omega(u^{(n)})^3, \\ \eta(u^{(n)}) &= 2A_5 I - 3A_6\Omega(u^{(n)}) + 9A_7\Omega(u^{(n)})^2 - 27A_8\Omega(u^{(n)})^3, \\ \Omega(u^{(n)}) &= (G'(u^{(n)}))^{-1}G'(v^{(n)}), \end{aligned}$$

and the coefficients are given by

$$\begin{aligned} A_1 &= 54b^3 - 135b^2 + 102b - 23, \\ A_2 &= -54b^3 + 135b^2 - 112b + 29, \\ A_3 &= 18b^3 - 45b^2 + 38b - 11, \\ A_4 &= (b-1)^2(2b-1), \\ A_5 &= 27b^3 - 54b^2 + 33b - 4, \\ A_6 &= 27b^3 - 54b^2 + 35b - 6, \\ A_7 &= 9b^3 - 18b^2 + 11b - 2, \\ A_8 &= b(b-1)^2. \end{aligned}$$

Here, b is a parameter that can vary freely, $G'(u^{(n)})$ is the Jacobian matrix of function G at point $u^{(n)}$, and I is the identity matrix of order m.

Cordero et al. [30] in 2025 proposed a fourth-order iterative scheme for solving systems of nonlinear equations (using Euclidean norms) defined as

$$\begin{cases} v^{(n)} = u^{(n)} - (G'(u^{(n)}))^{-1} \cdot G(u^{(n)}) \\ u^{(n+1)} = v^{(n)} - (G'(u^{(n)}))^{-1}\left(\rho^{(n)}G(v^{(n)}) + q^{(n)}G(u^{(n)})\right), \quad n = 0, 1, \ldots \end{cases} \tag{3.3}$$

where

$$(\lambda, \psi) \in \mathbb{R}^2, \quad \mu^{(n)} = \frac{\left\|G(v^{(n)})\right\|^2}{\left\|G(u^{(n)})\right\|^2}, \quad \rho^{(n)} = \frac{1 + \psi\mu^{(n)}}{1 + \lambda\mu^{(n)}}, \quad q^{(n)} = \frac{2\mu^{(n)}}{1 + \lambda\mu^{(n)}}.$$

The comparison between the iterative methods is done on the basis of residual error $\|G(u^{(n)})\|_2$, error estimation $\|u^{(n)} - u^{(n-1)}\|_2$, order of convergence, number of iterations, and computational time. On the other hand, the integrated IGWOA method is compared with modern optimization and hybrid algorithms including GWO [29], the Tuna Swarm Optimization algorithm (TSO) [32], WOA [17], the Hybrid Equilibrium Slime Mould Optimization algorithm (ESMA) [20], the Hybrid Adaptive Grey Wolf Optimization algorithm (HGWO) [21], the Jarratt–Butterfly Optimization algorithm (JBOA) [33], and the Newton–Harris Hawk Optimization algorithm (NHHO) [24]. The comparison is carried out based on key performance indicators such as functional error (fitness) $\left(\|G(u^{(n)})\|_2\right)$, number of iterations, and computational (CPU) time. Tables 1 and 2 represent the general parametric values, which are same for all the methods, and the values which are fixed for the optimization methods.

Remark: Throughout the Tables 3–14, the meaning of $ke - y$ means $k \times 10^{-y}$. All computations were terminated after a prescribed maximum number of iterations ($n \leq N$). No residual–based convergence tolerance is applied. All numerical experiments were performed in MATLAB using Variable Precision Arithmetic (VPA) with 600 decimal digits. The use of high-precision arithmetic minimizes round-off errors, thereby enabling an accurate assessment of the theoretical convergence order of the iterative methods. To ensure a fair comparison, all competing algorithms were executed using the same precision level, and the reported CPU times include the computational cost associated with multiprecision arithmetic.

Table 1. Uniform experimental setup and parameter selection across all algorithms.

Parameter	**Value**
Population size (T)	20
Total iterations optimization (N)	100
Total iterations iterative (N)	10
CPU time	Seconds
Fitness	$\|G(u^{(n)})\|_2$
Error estimation	$\|u^{(n)} - u^{(n-1)}\|_2$
Order of convergence	ρ
Digits	600
Software	MATLAB 2026a Windows 11 system; Processor: Intel i5, 8 GB RAM

Table 2. Tuning parameters used for competing techniques.

Method	Different parameters	Assigned values
IGWOA	a	Between 2 to 0
	r_1, r_2	Random numbers between [0, 1]
GWO	a	Between 2 to 0
	r_1, r_2	[0, 1]
TSO	a	0.7
	$rand$	Random number between 0 and 1
	b	Between 0 and 1
	z	0.05
	TF	Random value 1 or −1
WOA	$rand$	[0, 1]
	I	$I \in \{1, 2\}$
	SW	Represents the best solution
ESMA	q	0.2
	a_1	2
	a_2	1
	z	0.6
	GP	0.5
	V	1
HGWO	a	Random number between [0, 2]
	ω	[0.3, 0.9]
	θ	0.5
	$V_{rand}(t)$	[−20, 20]
JBOA	Sensory modality	0.01
	p (probability parameter switch)	0.8
	Power exponent	0.1
NHHO	β	1.5
	E_0	Random number between [−1, 1]
CM	ψ and λ	0.9 and 0.8
KM	b	1

Example 1. *The first test case considered is the Modified Strictly Convex Function II [34]. Due to its strictly convex nature, the problem possesses a unique solution, making it an appropriate benchmark for testing the accuracy and stability of the proposed methods. The mathematical formulation of the system is given as follows:*

$$g_i(u) = \left(\frac{i}{m+1}\right) e^{u_i} - 1, \quad i = 1, 2, 3, 4, 5, \quad \text{and } m = 5. \tag{3.4}$$

The results for this problem are determined in the initial guess $u^{(0)} = (1 - 1/m, 1 - 2/m, \ldots, 0)^T$*, and for the optimization method, the initial interval is taken as* $[0, 5]$.

The obtained approximate root for the considered problem is $(1.791759469228055\ldots, 1.098612288668110\ldots, 0.693147180559945\ldots, 0.405465108108164\ldots,$

$0.182321556793955\ldots)^T$. From Table 3, it is concluded that the proposed method (FM) demonstrates competitive convergence behavior among the iterative schemes. Specifically, FM achieves a rapid reduction in both the error norm and the residual norm regarding the known methods CM, KM, and JM. Moreover, FM attains high accuracy with the least computational time, confirming its efficiency and stability among the iterative methods. Although all methods exhibit fourth-order convergence, FM shows an advantage in terms of both convergence speed and computational cost. In addition, Table 4 shows that the proposed IGWOA achieves better results than the competing methods in terms of solution accuracy, achieving a low fitness value of $4.0783e-56$ within only three iterations. In contrast, other algorithms such as GWO, TSO, HGWO, and ESMA exhibit low accuracy with relatively high fitness values and larger iteration counts. Even the hybrid methods such as JBOA and NHHO, although yielding better fitness values, require more iterations and higher computational time compared to IGWOA. Overall, the results demonstrate that the proposed approaches FM and IGWOA are efficient and accurate.

Table 3. Comparative analysis of the proposed method and iterative techniques for Example 1.

Method	n	$\|u^{(n)}-u^{(n-1)}\|_2$	$\|G(u^{(n)})\|_2$	ρ	CPU time (sec)
FM	3	$3.2959e-05$	$3.2959e-05$	4.000	3.663846
	4	$1.2019e-20$	$1.1013e-21$		
	5	$2.1258e-78$	$2.1258e-80$		
CM	3	$2.2823e-01$	$1.5191e-01$	4.000	4.176201
	4	$8.1286e-05$	$5.2395e-05$		
	5	$1.6071e-22$	$1.0361e-23$		
KM	3	$1.2016e-02$	$1.2088e-03$	4.000	6.813619
	4	$9.6113e-10$	$9.6113e-10$		
	5	$3.9508e-38$	$3.9508e-38$		
JM	3	$1.2016e-02$	$1.2088e-02$	4.000	8.219551
	4	$9.6113e-10$	$9.6113e-10$		
	5	$3.9508e-38$	$1.2799e-38$		

Table 4. Comparative analysis of the proposed method and optimization techniques for Example 1.

Method	$\|G(u^{(n)})\|_2$	Iter	CPU time (sec)
IGWOA	$4.0783e-56$	3	0.314663
GWO	2.389716	97	1.314488
TSO	2.389451	95	1.002316
WOA	0.909264	10	1.548684
HGWO	2.389457	73	0.846469
ESMA	2.389457	73	0.994671
JBOA	$1.4794e-31$	43	0.731146
NHHO	$6.4762e-26$	10	0.954662

Example 2. *The second test problem considered in this study is the Tridiagonal Exponential Function [34], which is a commonly used benchmark problem for evaluating numerical methods for nonlinear systems. This problem contains exponential and hyperbolic cosine terms arranged in a tridiagonal structure, making it computationally challenging for many iterative algorithms. For this function, the value of* $m = 30$ *is considered, which makes a system of* 30×30 *equations given as:*

$$\begin{cases} g_1(u) = u_1 - e^{\cos(h(u_1+u_2))}, \\ g_i(u) = u_i - e^{\cos(h(u_{i-1}+u_i+u_{i+1}))}, \quad i = 2, 3, \ldots, m-1, \\ g_m(u) = u_m - e^{\cos(h(u_{m-1}+u_m))}, \end{cases} \tag{3.5}$$

$$h = \frac{1}{m+1}.$$

The initial guess taken for the considered benchmark problem is $(0,\ 1/m,\ 2/m,\ \ldots,\ (m-1)/m)^T$*, and for the integrated approach, the initial interval is* $[0, 5]$.

The obtained root to this problem is $(2.678794257174148\ldots, 2.630997946823661\ldots, 2.632001991493735\ldots, 2.631980954422541\ldots, \ldots, 2.632001991493735\ldots, 2.630997946823661\ldots, 2.678794257174148\ldots)^T$. From Table 5, FM achieves high accuracy in error and residual norms, along with the lowest computational time among the compared iterative methods. On the other hand, Table 6 highlights the comparison with optimization-based algorithms, where IGWOA shows better results. The method attains a low fitness value of $4.8939e-55$ within only three iterations and the least CPU time of 0.231161 seconds. In contrast, other algorithms such as GWO, TSO, HGWO, and ESMA exhibit much larger fitness values and require higher iteration counts. Even relatively competitive methods such as JBOA and NHHO, though achieving lower fitness values, still demand more iterations and higher computational time compared to IGWOA.

Table 5. Comparative analysis of the proposed method and iterative techniques for Example 2.

Method	n	$\|u^{(n)} - u^{(n-1)}\|_2$	$\|G(u^{(n)})\|_2$	ρ	CPU time (sec)
FM	3	$2.1880e-04$	$2.3207e-04$	4.000	4.034256
	4	$3.5575e-23$	$3.7821e-23$		
	5	$3.2969e-98$	$3.5057e-98$		
CM	3	$4.9763e-03$	$5.2480e-03$	4.000	5.453406
	4	$4.4627e-13$	$4.5679e-13$		
	5	$2.1624e-43$	$2.2381e-43$		
KM	3	$6.5557e-04$	$6.8173e-04$	4.000	9.780043
	4	$1.5928e-17$	$1.6186e-18$		
	5	$1.3625e-70$	$1.3625e-71$		
JM	3	$6.8827e-03$	$6.7180e-04$	4.000	9.076345
	4	$1.7414e-15$	$1.7347e-15$		
	5	$6.1589e-63$	$6.0875e-64$		

Table 6. Comparative analysis of the proposed method and optimization techniques for Example 2.

Method	$\|G(u^{(n)})\|_2$	Iter	CPU time (sec)
IGWOA	$4.8939e-55$	3	0.231161
GWO	14.662388	96	1.968412
WOA	0.617989	44	1.301189
TSO	13.462360	43	1.003146
HGWO	30.493743	45	0.945668
ESMA	13.58656	54	0.763346
JBOA	$2.5599e-30$	18	0.593468
NHHO	$1.1833e-30$	15	0.681314

Example 3. *To evaluate the performance and robustness of the proposed methods, a logarithmic test problem [34] is considered. This problem is commonly used as a benchmark due to its nonlinear nature and the presence of logarithmic terms, which introduce additional computational complexity. The problem statement is as follows:*

$$g_i(u) = \log(u_i + 1) - \frac{u_i}{m}, \quad i = 1, 2, \ldots, m, \ m = 40. \tag{3.6}$$

The initial guess taken for the considered benchmark problem is $(0.1, 0.1, 0.1, \ldots, 0.1, 0.1, 0.1)^T$*, and for the integrated approach or optimization methods, the initial interval is* $[-1, 1]$.

The approximate root for the considered benchmark problem is the zero vector, within machine precision. In addition, FM exhibits a rapid reduction in functional error in less CPU time and fewer iterations. From Table 8, it is clear that the IGWOA delivers a good performance among the optimization methods. It achieves a low fitness value of $4.7219e-69$ within only four iterations and a minimal CPU time of 0.299130 seconds. In contrast, traditional algorithms such as GWO, TSO, and ESMA require a higher number of iterations and produce comparatively larger fitness values. Even though methods such as JBOA and NHHO yield relatively better fitness values, they still lag behind IGWOA in terms of both computational cost and convergence speed.

Table 7. Comparative analysis of the proposed method and iterative techniques for Example 3.

Method	n	$\|\|u^{(n)} - u^{(n-1)}\|\|_2$	$\|\|G(u^{(n)})\|\|_2$	ρ	CPU time (sec)
FM	3	$8.0436e-13$	$7.8425e-13$	4.000	3.470064
	4	$5.9069e-51$	$5.7587e-51$		
	5	$5.2694e-203$	$5.1377e-203$		
CM	3	$7.2948e-10$	$7.1124e-10$	4.000	4.006745
	4	$5.1997e-29$	$5.0697e-29$		
	5	$1.6367e-86$	$1.5958e-86$		
KM	3	$1.4521e-03$	$1.4155e-03$	4.000	8.563950
	4	$2.3413e-15$	$2.2828e-15$		
	5	$9.2486e-62$	$9.0174e-62$		
JM	3	$1.4155e-03$	$1.4520e-03$	4.000	7.999453
	4	$2.2828e-15$	$2.3413e-15$		
	5	$9.0174e-62$	$9.2486e-62$		

Table 8. Comparative analysis of the proposed method and optimization techniques for Example 3.

Method	$\|\|G(u^{(n)})\|\|_2$	Iter	CPU time (sec)
IGWOA	$4.7219e-69$	4	0.299130
GWO	$2.5533e-05$	99	2.819697
WOA	$1.2000e-14$	19	2.009773
TSO	$5.0500e-13$	63	1.897344
ESMA	$7.3145e-15$	62	2.311454
HGWO	$2.9616e-09$	21	1.999364
JBOA	$9.8500e-32$	46	0.993461
NHHO	$2.9507e-23$	16	0.963468

Example 4. *To further assess the performance of the proposed numerical method, the Chandrasekhar H-equation [34] is considered as a test problem. The Chandrasekhar H-equation plays a fundamental role in the radiative transfer theory, particularly in modeling the scattering of radiation in planetary atmospheres and stellar atmospheres. It is used to describe the reflection and transmission of radiation in a semi-infinite scattering medium. This equation is widely used as a benchmark problem for testing numerical algorithms designed to solve large-scale nonlinear systems. The nonlinear system is given by*

$$g_i(u) = u_i - \left(1 - \frac{c}{2m}\sum_{j=1}^{m}\frac{\delta_i u_j}{\delta_i + \delta_j}\right)^{-1}, \quad i = 1, 2, \ldots, m, \tag{3.7}$$

where

$$c = 0.9, \qquad \delta_i = \frac{i - 0.5}{m}, \quad i = 1, 2, \ldots, m. \tag{3.8}$$

Here, $m = 10$ represents the dimension of the nonlinear system, and u_i are the unknown variables. The parameters δ_i arise from the discretization of the integral form of the Chandrasekhar equation. This problem is challenging because each equation depends on all variables u_j, resulting in a fully coupled nonlinear system. This type of coupling results in increased computational complexities, especially for larger values of n, which makes this a good benchmark problem in determining the efficiency, robustness, and convergence properties of iterative numerical solution techniques. The Chandrasekhar H-equation has important applications in radiative transfer theory, particularly in modeling radiation scattering in stellar and planetary atmospheres. The initial guess used for this problem is $((m-1)/m, (m-2)/m, \ldots, 0)^T$, and the initial interval for the optimization method is $[-10, 10]$.

The obtained root of the above problem is (1.096735816834478 . . . , 1.233484021722359 . . . , 1.342362913031034 . . . , 1.435646349049370 . . . , 1.517868486486611 . . . , 1.591492086882111 . . . , 1.658105751392686 . . . , 1.718837251229592 . . . , 1.774536373739270 . . . , 1.825869482591646 . . .)^T.
The numerical outcomes reported in Tables 9 and 10 show the computational efficiency and robustness of the proposed methods. Moreover, FM requires only 3.475209 seconds for obtaining the optimal solution. Similarly, Table 10 shows that the IGWOA algorithm yields better results than the other optimization techniques. It achieves a low fitness value of $9.4955e-140$ in just four iterations with a CPU time of 0.314600 seconds. In contrast, other algorithms such as GWO, WOA, and TSO require nearly 98–99 iterations and yield fitness values of the order $1e{-}3$ to $1e{-}4$. Even comparatively efficient methods such as JBOA and NHHO produce fitness values of $9.8607e{-}32$ and $2.4652e{-}31$, respectively, but still fall short of the accuracy achieved by IGWOA. From this, it is concluded that the proposed methods are efficient in solving this nonlinear system.

Table 9. Comparative analysis of the proposed method and iterative techniques for Example 4.

Method	n	$\|\|u^{(n)} - u^{(n-1)}\|\|_2$	$\|\|G(u^{(n)})\|\|_2$	ρ	CPU time (sec)
FM	2	$1.5452e-08$	$9.0008e-08$	4.000	3.475209
	3	$8.2895e-31$	$4.8344e-31$		
	4	$6.8323e-124$	$3.9846e-124$		
CM	2	$6.4101e-08$	$1.2945e-08$	4.000	5.453428
	3	$1.9811e-23$	$3.0482e-23$		
	4	$5.1436e-70$	$5.6612e-70$		
KM	2	$3.8034e-07$	$2.2229e-07$	4.000	9.534268
	3	$2.5109e-29$	$1.4737e-29$		
	4	$9.0124e-108$	$9.0124e-108$		
JM	2	$2.2229e-07$	$3.8034e-07$	4.000	9.006347
	3	$1.4737e-29$	$2.5109e-29$		
	4	$2.7825e-118$	$4.7407e-118$		

Table 10. Comparative analysis of the proposed method and optimization techniques for Example 4.

Method	$\|G(u^{(n)})\|_2$	Iter	CPU time (sec)
IGWOA	$9.4955e-140$	4	0.314600
GWO	$2.6364e-03$	98	2.314662
WOA	$5.5351e-04$	99	2.00131
TSO	$2.9394e-04$	98	1.981364
ESMA	$1.1325e-01$	97	0.196718
HGWO	$3.4200e-02$	99	0.346119
JBOA	$9.8607e-32$	80	0.993168
NHHO	$2.4652e-31$	10	0.813161

Example 5. ***Vapor-liquid equilibrium: Wilson's equation*** *Vapor–liquid equilibrium (VLE) analysis [35] is a fundamental aspect of chemical engineering, particularly in processes involving phase separation. In practical systems, mixtures rarely behave ideally due to molecular interactions between components. Hence, in order to effectively model this phenomenon, thermodynamic relations are required, which take into account these deviations in behavior. One such relation, which has found wide applicability in determining activity coefficients in nonideal mixtures, is Wilson's equation. In this paper, the problem of VLE has been effectively modeled using a system of ten coupled nonlinear equations with ten unknowns, which include temperature, saturation pressures, composition in the vapor phase, and activity coefficients. Due to the presence of exponential and logarithmic terms, this system is highly nonlinear in nature. Wilson's equation is significant in this regard, as this equation can effectively determine activity coefficients, which are a measure of deviations in real mixtures from ideal mixtures. Such a model is significant in petrochemical, pharmaceutical, and chemical industries, where exact control over mixtures is required. The vapor–liquid equilibrium problem based on Wilson's model can be expressed as*

$$
\begin{aligned}
g_1(u) &= \ln(P_1^{sat}) - 17 + \frac{3600}{T-54} = 0,\\
g_2(u) &= \ln(P_2^{sat}) - 16.5 + \frac{3850}{T-47} = 0,\\
g_3(u) &= k_1\gamma_1 P_1^{sat} - y_1 P = 0,\\
g_4(u) &= k_2\gamma_2 P_2^{sat} - y_2 P = 0,\\
g_5(u) &= 1 - y_1 - y_2 = 0,\\
g_6(u) &= \ln(\gamma_1) + \ln(k_1 + k_2\Lambda_{12}) - k_2 W = 0,\\
g_7(u) &= \ln(\gamma_2) + \ln(k_1\Lambda_{21} + k_2) + k_1 W = 0,\\
g_8(u) &= W - \frac{\Lambda_{12}}{k_1 + k_2\Lambda_{12}} + \frac{\Lambda_{21}}{k_1\Lambda_{21} + k_2} = 0,\\
g_9(u) &= \Lambda_{12} - \frac{V_2}{V_1}\exp\left(\frac{-a_{12}}{RT}\right) = 0,\\
g_{10}(u) &= \Lambda_{21} - \frac{V_1}{V_2}\exp\left(\frac{-a_{21}}{RT}\right) = 0.
\end{aligned}
\tag{3.9}
$$

The problem is based on a binary mixture and includes ten unknown variables defined by $u = (P_1^{sat}, P_2^{sat}, T, W, y_1, y_2, \Lambda_{12}, \Lambda_{21}, \gamma_1, \gamma_2)$. *The variables are represented by the following notations:* P_1^{sat} *and* P_2^{sat} *denote the saturation pressures determined by the two components, respectively. The temperature of the system is represented by the variable* T. *The variables* y_1 *and* y_2 *are the mole fractions of the two components in the vapor phase. The activity coefficients* γ_1 *and* γ_2 *are also considered to take into account any nonideal solution behavior. The variables* Λ_{12} *and* Λ_{21} *are the interaction parameters of Wilson's equation and are considered to take into account the interactions between molecules of different components. The variable* W *is considered an auxiliary variable to simplify the process of formulating the problem equations.*

The parameters for the problem are also provided and are as follows: The liquid phase is considered to have a composition of components 1 and 2 in the ratio of $k_1 = 0.85$ *and* $k_2 = 0.15$, *respectively. The total pressure of the system is considered to be* $P = 100$ *kPa. The energy interaction parameters for the components are* $a_{12} = 440$ *cal/mol and* $a_{21} = 1250$ *cal/mol, respectively. The molar volumes of the components is* $V_1 = 77$ cm^3/mol *and* $V_2 = 18$ cm^3/mol, *respectively. The value of the universal gas constant is* $R = 1.987$. *The solution of the nonlinear equilibrium problem is important and plays a significant role in practical engineering problems. The solution of such a problem is important for the design of a distillation column and helps engineers make better decisions regarding the design and operation of such units. The solution of the problem also helps engineers design such units efficiently and to maintain a high level of energy efficiency, which is important for the overall operation of such industries and helps in saving costs. The solution of such a problem is also important for ensuring the quality of the end product in chemical and pharmaceutical industries.*

The initial values taken for the considered application are $(90, 50, 350, 0.5, 0.6, 0.5, 1, 1, 1, 1)^T$, *and the initial interval for the optimization methods is* $[-500, 500]$.

The values of the unknowns are obtained as $(P_1^{sat}, P_2^{sat}, T, W, y_1, y_2, \Lambda_{12}, \Lambda_{21}, \gamma_1, \gamma_2)$, and they are (99.076726223908508, 34.706084739028263, 344.2264913894762, −0.794810512055846, 0.860754222025224, 0.139245777974776, 0.122857617866317, 0.687900967331513, 1.022088702874058, 2.674762058235259)^T. The saturation pressures of the two pure components are obtained as $P_1^{sat} = 99.076726223908508$ and $P_2^{sat} = 34.706084739028263$, indicating that the first component is considerably more volatile than the second at the equilibrium temperature because it possesses a substantially higher vapor pressure. The equilibrium temperature is found to be $T = 344.2264913894762$ K, which corresponds to approximately 71.08° C, representing the temperature at which both phases coexist under the specified equilibrium conditions. The auxiliary variable is computed as $W = -0.794810512055846$. Although this quantity does not possess a direct physical interpretation, its finite value confirms that the transformed nonlinear equations are simultaneously satisfied, and this contributes to improving the numerical stability and convergence of the solution procedure. The vapor-phase mole fractions are obtained as $y_1 = 0.860754222025224$ and $y_2 = 0.139245777974776$, demonstrating that approximately 86.07% of the vapor phase consists of component 1, whereas only 13.92% corresponds to component 2. This distribution is consistent with the significantly higher saturation pressure of component 1, which causes it to preferentially partition into the vapor phase. The Wilson interaction parameters are estimated as $\Lambda_{12} = 0.122857617866317$, and $\Lambda_{21} = 0.68790096733151$. Because these values are unequal, they indicate asymmetric molecular interactions between the two components, implying that the influence of component 1 on component 2 differs from the influence of component 2 on component 1. Such asymmetry is characteristic of

real liquid mixtures and reflects differences in molecular sizes and interaction energies. The activity coefficients are calculated as $\gamma_1 = 1.022088702874058$, and $\gamma_2 = 2.674762058235259$. The value of γ_1 being very close to unity, indicates that component 1 exhibits behavior that is nearly ideal in the liquid phase, with only minor deviations from Raoult's law. In contrast, the significantly larger value of γ_2 reveals substantial positive deviation from ideality for component 2, suggesting weaker attractive interactions between unlike molecules than between like molecules. Consequently, component 2 possesses a greater escaping tendency from the liquid phase than predicted by an ideal solution model. Overall, the computed parameter values describe a binary mixture in which component 1 is the more volatile constituent and behaves nearly ideally, whereas component 2 exhibits pronounced nonideal behavior due to stronger thermodynamic interaction effects, validating the necessity of incorporating Wilson's activity coefficient model in the vapor–liquid equilibrium formulation.

Tables 11 and 12 depict the performance of the constructed methods. From Table 11, FM achieves high precision with $\|u^{(n)} - u^{(n-1)}\|_2 \approx 3.0177e{-}114$ and $\|G(u^{(n)})\|_2 \approx 3.0302e{-}116$, along with a lower CPU time of 3.113468 seconds. From Table 12, IGWOA attains a significantly lower fitness value of $4.8809e{-}94$ in only four iterations with a CPU time of 0.531316 seconds, whereas other methods such as GWO, WOA, TSO, HGWO, and ESMA yield fitness values of order $e{+}3$, which means the traditional optimization methods show the divergent results for the considered problem. Even JBOA and NHHO produce comparatively lower fitness values of $6.3573e{-}30$ and $3.1593e{-}30$, respectively, but less than IGWOA.

Overall, the proposed methods provide faster convergence and higher accuracy with reduced computational cost compared to existing approaches. The ability to produce stable and consistent results ensures dependable performance in engineering and scientific computations. Furthermore, the exact estimation of the parameters of the model highlights the applicability of the proposed methods in handling complex systems, such as thermodynamic and chemical equilibrium problems.

Remark: From the above results, it is concluded that IGWOA incorporates a deterministic fourth-order refinement stage, which naturally provides an advantage over pure metaheuristic algorithms such as GWO, WOA, and TSO. It is important to note that the purpose of these comparisons is not to claim equivalence in methodological scope, but to benchmark IGWOA against widely used standalone optimizers and to demonstrate the incremental benefit of hybridizing metaheuristics with deterministic refinement. Metaheuristic algorithms are widely used for nonlinear optimization problems, and comparing IGWOA against them establishes how much improvement is gained by incorporating the refinement. The deterministic refinement is not intended to replace metaheuristics but to complement them.

Table 11. Comparative analysis of the proposed method and iterative techniques for Example 5.

Method	n	$\|\|u^{(n)} - u^{(n-1)}\|\|_2$	$\|\|G(u^{(n)})\|\|_2$	ρ	CPU time (sec)
FM	3	$2.4884e-05$	$1.6915e-07$	4.000	3.113468
	4	$4.3025e-27$	$3.9622e-29$		
	5	$3.0177e-114$	$3.0302e-116$		
CM	3	$2.0302e-03$	$1.2540e-05$	4.000	5.563983
	4	$5.0734e-14$	$2.3690e-16$		
	5	$1.4599e-45$	$1.0681e-47$		
KM	3	$2.2398e-05$	$1.5405e-07$	4.000	7.546384
	4	$1.9382e-27$	$1.8149e-29$		
	5	$1.5353e-106$	$1.8333e-107$		
JM	3	$2.2398e-03$	$1.5405e-04$	4.000	8.600735
	4	$1.9382e-14$	$1.8149e-18$		
	5	$8.4943e-60$	$8.5949e-71$		

Table 12. Comparative analysis of the proposed method and optimization techniques for Example 5.

Method	$\|\|G(u^{(n)})\|\|_2$	Iter	CPU time (sec)
IGWOA	$4.8809e-94$	4	0.531316
GWO	$1.7501e+03$	15	2.811467
WOA	$1.7445e+03$	8	1.031146
TSO	$1.1279e+03$	5	1.643167
HGWO	$1.6578e+03$	3	2.007346
ESMA	$1.1293e+03$	12	2.370146
JBOA	$6.3573e-30$	49	1.003164
NHHO	$3.1593e-30$	8	1.981643

3.1. Stability and consistency

With respect to the iterative methods, the results obtained in each run are the same because the method works based on an initial guess and results in the same output in each run. However, considering that many optimization techniques include randomness, the results derived in one run might not be optimal. The results derived in multiple runs help in assessing the robustness and stability of the algorithm by comparing the results in terms of mean, standard deviation, and the best results. This gives a precise idea about the efficiency of the algorithm. For assessing the performance of the proposed algorithms in terms of stability and efficiency, two statistical measures, mean and standard deviation, can be used. The mean value represents the stability of an algorithm, which works based on the ability to provide consistent results in 30 runs. An optimization method can be considered consistent if there is minimal variation in fitness values. The standard deviation measures the results derived, which help in understanding the stability of the algorithm. If the standard deviation is low,

it means that the algorithm works more stably. From Table 13, it can be observed that the proposed algorithms provide better average fitness values compared to the conventional optimization algorithms. Table 14 explains the standard deviation values, which indicate that the proposed algorithms provide results with low standard deviation values. This ensures the stability of the proposed optimization techniques compared to the existing techniques.

Table 13. Average mean of the fitness values in 30 runs for Examples 1–5.

Methods	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5
IGWOA	$6.0722e-56$	$2.4282e-55$	$2.4546e-68$	$3.8452e-139$	$1.9868e-94$
GWO	$4.8939e-05$	24.91676	0.00113	0.18323	$1.7623e+03$
TSO	0.03049	13.46236	$1.9698e-07$	0.00118	$1.1300e+03$
WOA	$1.9712e-18$	0.61796	$2.1656e-04$	0.08563	$1.7487e+03$
HGWO	0.15173	34.20593	0.00853	0.08896	$1.1308e+03$
ESMA	0.05354	13.62951	$1.4689e-04$	0.17109	$1.1334e+03$
JBOA	$8.1545e-24$	$6.0978e-26$	$8.4145e-23$	$3.6998e-27$	$3.8656e-29$
NHHO	$6.6256e-31$	$8.3266e-32$	$1.4267e-31$	$8.3200e-32$	$3.3967e-31$

Table 14. Average standard deviation of the fitness values in 30 runs for Examples 1–5.

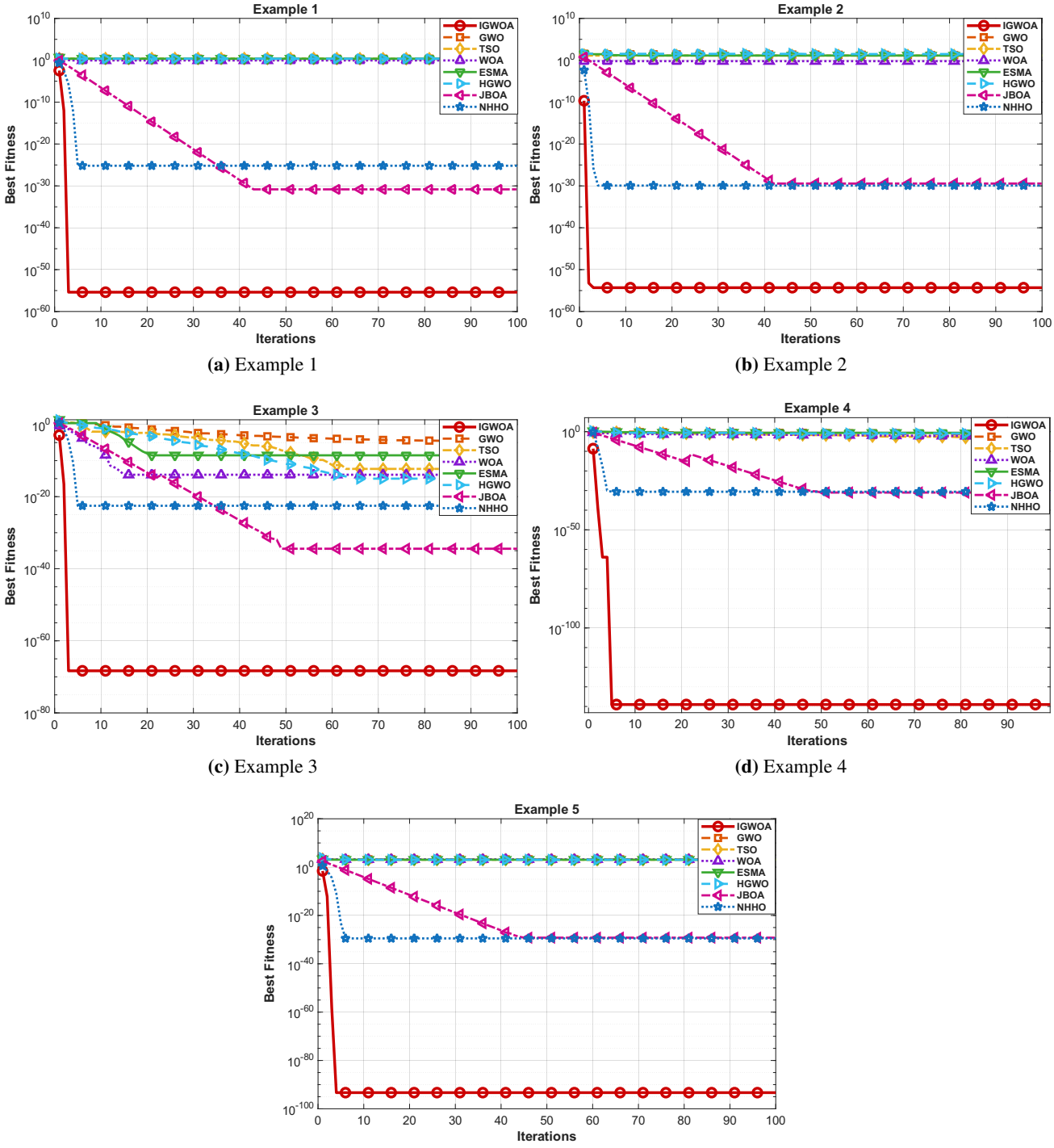
Methods	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5
IGWOA	$7.6418e-56$	$3.3915e-55$	$3.0354e-68$	$5.0417e-139$	$1.7885e-94$
GWO	$1.1019e-05$	0.88903	0.00123	0.04959	43.68645
TSO	0.13581	1.6869	$4.5349e-07$	$9.7851e-04$	$1.1300e+03$
WOA	$3.0596e-18$	0.20552	$8.9812e-04$	0.08191	$1.7487e+03$
HGWO	0.4856	1.11452	0.02153	0.020566	4.30183
ESMA	0.18737	1.23914	$6.4404e-04$	0.03866	$1.1334e+03$
JBOA	$1.5696e-23$	$1.0093e-25$	$3.7964e-22$	$1.2477e-26$	$1.6685e-28$
NHHO	$3.2444e-31$	$1.2591e-31$	$2.3219e-31$	$1.2415e-31$	$7.1236e-31$

The outcomes show that the proposed methods have an effective search strategy that can locate exact answers. It is also capable of avoiding immature convergence and local optimal traps. Additionally, one of the essential factors that is used in the comparison of algorithms is the convergence speed of the method. While comparing algorithms, the convergence speed is a crucial factor. An algorithm's ability to find a solution is indicated by its convergence. The convergence chart shows the accuracy and speed of an algorithm in solving a nonlinear system of equations.

Figure 1 illustrates that the convergence performance of the compared algorithms, namely GWO, TSO, WOA, ESMA, HGWO, JBOA, and NHHO, is significantly enhanced by the proposed method, IGWOA. In comparison to these methods, IGWOA requires fewer iterations to attain the optimal and accurate solution, indicating its faster convergence behavior and computational efficiency.

Furthermore, Figure 2 provides a comparison of fitness values obtained over multiple runs for each algorithm. It can be observed that the fitness values of the existing optimization algorithms exhibit noticeable fluctuations across different runs, whereas the proposed IGWOA method demonstrates

a more stable and consistent performance. This consistency highlights the better performance of IGWOA in producing reliable and uniform solutions across independent runs. Therefore, it can be concluded that the proposed method shows better results than the compared algorithms in terms of both convergence speed and solution stability.

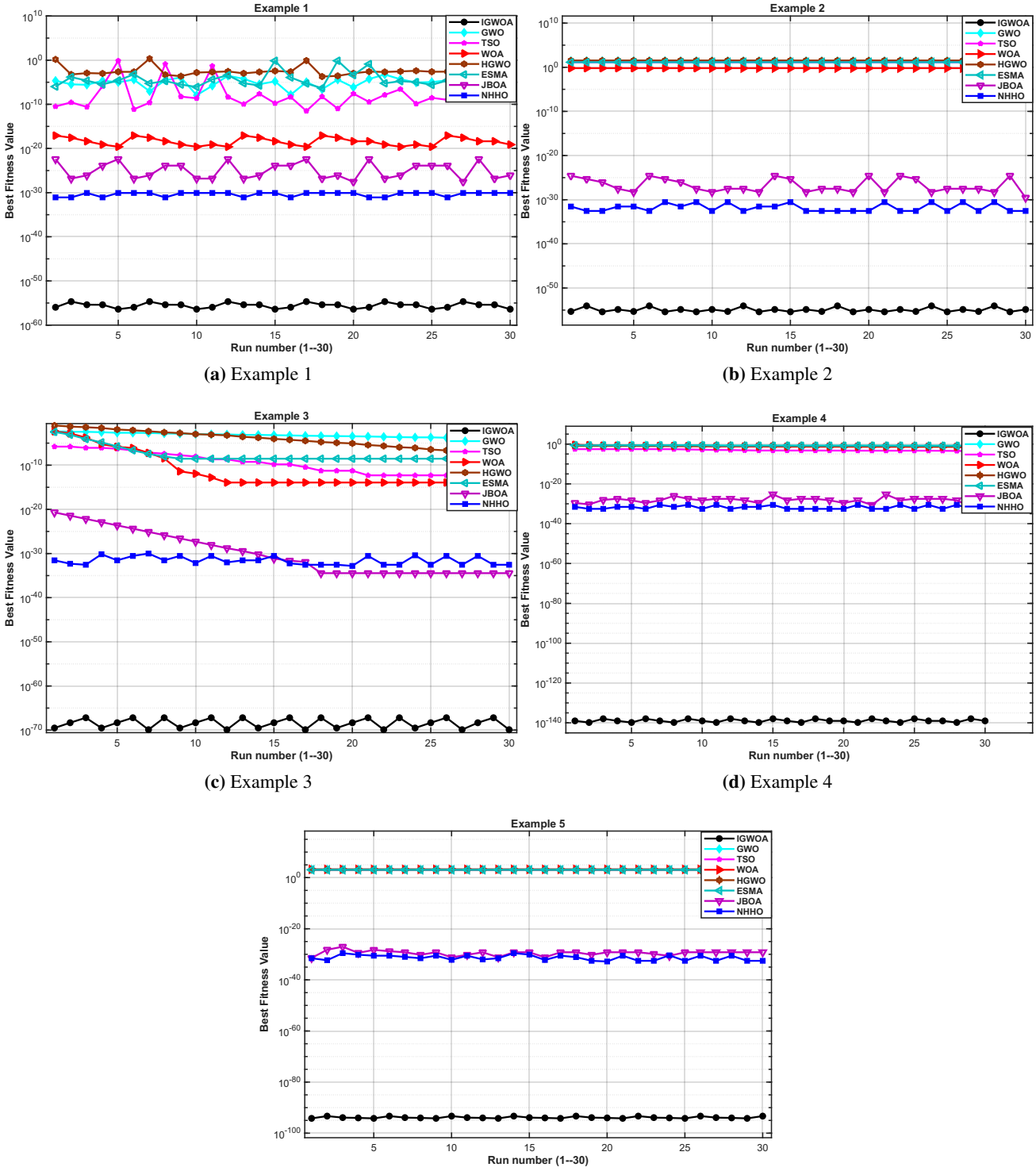


(a) Example 1 **(b)** Example 2

(c) Example 3 **(d)** Example 4

(e) Example 5

Figure 1. Convergence curve of the fitness value versus 100 iterations for Examples 3.1–3.5.



(a) Example 1

(b) Example 2

(c) Example 3

(d) Example 4

(e) Example 5

Figure 2. Convergence curve of the fitness value versus 30 runs for Examples 3.1–3.5.

4. Conclusions

In this work, a novel fourth-order iterative method (FM) has been proposed for solving SoNE, along with its integration with the Grey Wolf Optimization algorithm (IGWOA). The developed method exhibits high convergence speed and accuracy, requiring fewer iterations and reduced computational effort. The hybrid approach effectively integrates the local convergence strength of the iterative scheme with the global search capability of the optimization algorithm, resulting in improved performance in terms of fitness accuracy, stability, and consistency. Comparative results demonstrate that the proposed methods improve existing algorithms such as GWO, TSO, WOA, ESMA, HGWO, JBOA, and NHHO, on the analyzed examples. Furthermore, statistical analysis over multiple runs confirms the efficiency and reliability of the proposed method in handling complex nonlinear systems. The results highlight the potential of hybrid optimization–iterative strategies as powerful tools for solving real-world nonlinear problems arising in science and engineering. Moreover, in this work, the initial search intervals used for the optimization methods have been clearly reported, and the sensitivity of the proposed approach to these intervals has been qualitatively discussed. However, we acknowledge that a more detailed quantitative investigation of this factor lies beyond the present scope. Future work will focus on systematically analyzing the influence of initial search intervals to provide deeper insights into the robustness and general applicability of the proposed method.

Author contributions

A. Chandel, S. Bhalla: Conceptualization; J. R. Torregrosa: Methodology; A. Chandel: Software; A. Cordero, J. R. Torregrosa: Validation; G. Singh: Formal analysis; S. Bhalla: Investigation; G. Singh: Resources; A. Chandel: Writing—original draft preparation; A. Cordero, J. R. Torregrosa: Writing—review and editing; S. Bhalla: Supervision. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research is partially supported by the IMU-Simmons Research Fellowship. The authors would like to thank the anonymous reviewers for their useful comments and suggestions.

Conflict of interest

The authors declare no conflict of interests.

Alicia Cordero and Juan R. Torregrosa are the editorial board member for AIMS Mathematics and were not involved in the editorial review or the decision to publish this article.

References

1. C. Grosan, A. Abraham, A new approach for solving nonlinear equations system, *IEEE T. Syst. Man Cy.-Part A,* **38** (2008), 698–714. https://doi.org/10.1109/TSMCA.2008.918599
2. J. F. Epperson, *An introduction to numerical methods and analysis*, New York: John Wiley & Sons, 2013.
3. D. Chen, I. K. Argyros, Q. S. Qian, A note on the Halley method in Banach spaces, *Appl. Math. Comput.,* **58** (1993), 215–224. https://doi.org/10.1016/0096-3003(93)90137-4
4. M. A. Noor, W. A. Khan, A. Hussain, A new modified Halley method without second derivatives for nonlinear equations, *Appl. Math. Comput.,* **189** (2007), 1268–1273. https://doi.org/10.1016/j.amc.2006.12.011
5. A. S. Householder, *The numerical treatment of a single nonlinear equation*, New York: McGraw–Hill, 1970.
6. A. M. Ostrowski, *Solution of equations and system of equations*, New York: Academic Press, 1966.
7. J. F. Traub, *Iterative methods for the solution of equations*, American Mathematical Society, 1982.
8. P. Jarratt, Some fourth-order multipoint iterative methods for solving equations, *Math. Comput.,* **20** (1966), 434–437. https://doi.org/10.1090/S0025-5718-66-99924-8
9. A. Kumar, P. Maroju, R. Behl, D. K. Gupta, S. S. Motsa, A family of higher-order iterations free from the second derivative for nonlinear equations in $\mathbb{R}$, *J. Comput. Appl. Math.,* **330** (2018), 676–694. https://doi.org/10.1016/j.cam.2017.07.005
10. E. Martínez, J. A. Reyes, A. Cordero, J. R. Torregrosa, Stability analysis and local convergence of a new fourth-order optimal Jarratt-type iterative scheme, *Computation,* **13** (2025), 142. https://doi.org/10.3390/computation13060142
11. S. Bhalla, G. Singh, H. Ramos, R. Behl, H. Alshehri, A fourth-order parametric iterative approach for solving system of nonlinear equations, *Computation,* **13** (2025), 241. https://doi.org/10.3390/computation13100241
12. M. Jaberipour, E. Khorram, B. Karimi, Particle swarm algorithm for solving system of nonlinear equations, *Comput. Math. Appl.,* **62** (2011), 566–576. https://doi.org/10.1016/j.camwa.2011.05.031
13. T. Bäck, D. B. Fogel, Z. Michalewicz, Introduction to evolutionary algorithms, *Evol. Comput.,* **1** (2000), 59–63.
14. J. Kennedy, R. Eberhart, *Particle swarm optimization*, In: Proceedings of the IEEE International Conference on Neural Networks, **4** (1995), 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
15. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future Gener. Comput. Syst.,* **97** (2019), 849–872. https://doi.org/10.1016/j.future.2019.02.028
16. S. Arora, S. Singh, Butterfly optimization algorithm: A novel approach for global optimization, *Soft Comput.,* **23** (2019), 715–734. https://doi.org/10.1007/s00500-018-3102-4

17. P. Trojovský, M. Dehghani, A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior, *Sci. Rep.,* **13** (2023), 8775. https://doi.org/10.1038/s41598-023-35863-5

18. A. R. Yildiz, A new hybrid artificial bee colony algorithm for robust optimal design and manufacturing, *Appl. Soft Comput.,* **13** (2013), 2906–2912. https://doi.org/10.1016/j.asoc.2012.04.013

19. H. Garg, A hybrid PSO-GA algorithm for constrained optimization problems, *Appl. Math. Comput.,* **274** (2016), 292–305. https://doi.org/10.1016/j.amc.2015.11.001

20. M. K. Naik, R. Panda, A. Abraham, An entropy minimization based multilevel colour thresholding technique for analysis of breast thermograms using equilibrium slime mould algorithm, *Appl. Soft Comput.,* **113** (2021), 107955. https://doi.org/10.1016/j.asoc.2021.107955

21. M. Yu, J. Xu, W. Liang, Y. Qiu, S. Bao, L. Tang, Improved multi-strategy adaptive Grey Wolf Optimization for practical engineering applications and high-dimensional problem solving, *Artif. Intel. Rev.,* **57** (2024), 277. https://doi.org/10.1007/s10462-024-10821-3

22. C. L. Karr, B. Weck, L. M. Freeman, Solutions to system of nonlinear equations via a genetic algorithm, *Eng. Appl. Artif. Intel.,* **11** (1998), 369–375. https://doi.org/10.1016/S0952-1976(97)00067-5

23. Y. Z. Luo, G. J. Tang, L. N. Zhou, Hybrid approach for solving system of nonlinear equations using chaos optimization and quasi-Newton method, *Appl. Soft Comput.,* **8** (2008), 1068–1073. https://doi.org/10.1016/j.asoc.2007.05.013

24. R. Sihwail, O. S. Solaiman, K. A. Z. Ariffin, M. Alswaitti, I. Hashim, A hybrid approach for solving system of nonlinear equations using Harris hawks optimization and Newton's method, *IEEE Access,* **9** (2021), 95791–95807. https://doi.org/10.1109/ACCESS.2021.3094471

25. O. S. Solaiman, R. Sihwail, H. Shehadeh, I. Hashim, K. Alieyan, Hybrid Newton–sperm swarm optimization algorithm for nonlinear system, *Mathematics,* **11** (2023), 1473. https://doi.org/10.3390/math11061473

26. F. Zafar, A. Cordero, N. Ali, J. R. Torregrosa, A hybrid Jarratt–Particle Swarm Optimization algorithm for power flow analysis, *Authorea*, 2024. https://doi.org/10.22541/au.173094075.54692505/v1

27. A. Chandel, E. Martínez, S. Bhalla, S. Alharbi, R. Behl, A hybrid walrus optimization-based fourth-order method for solving nonlinear problems, *Axioms,* **15** (2025), 6. https://doi.org/10.3390/axioms15010006

28. A. Cordero, J. L. Hueso, E. Martínez, J. R. Torregrosa, A modified Newton-Jarratt's composition, *Numer. Algorithms*, **55** (2010), 87–99. https://doi.org/10.1007/s11075-009-9359-z

29. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.,* **69** (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

30. A. Cordero, R. V. Rojas-Hiciano, J. R. Torregrosa, M. P. Vassileva, Maximally efficient damped composed Newton-type methods to solve nonlinear system of equations, *Appl. Math. Comput.,* **492** (2025), 129231. https://doi.org/10.1016/j.amc.2024.129231

31. M. Kansal, A. Cordero, S. Bhalla, J. R. Torregrosa, New fourth- and sixth-order classes of iterative methods for solving system of nonlinear equations and their stability analysis, *Numer. Algorithms,* **87** (2021), 1017–1060. https://doi.org/10.1007/s11075-020-00997-4

32. L. Xie, T. Han, H. Zhou, Z. R. Zhang, B. Han, A. Tang, Tuna swarm optimization: A novel swarm-based metaheuristic algorithm for global optimization, *Comput. Intel. Neurosci.,* **2021** (2021), 1–22. https://doi.org/10.1155/2021/9210050

33. R. Sihwail, O. S. Solaiman, K. A. Z. Ariffin, New robust hybrid Jarratt–Butterfly optimization algorithm for nonlinear models, *J. King Saud Univ.-Com.,* **34** (2022), 8207–8220. https://doi.org/10.1016/j.jksuci.2022.08.004

34. H. Mohammad, A multivariate spectral hybridization of HS and PRP method for nonlinear system of equations, *arXiv preprint*, 2022. https://doi.org/10.48550/arXiv.2201.02943

35. M. Shacham, N. Brauner, Solving a system of nonlinear algebraic equations you only get error messages—what to do next?, *Chem. Eng. Educ.,* **51** (2017), 75–82.

AIMS Press