



## OPEN Modified iterative double step length model for solving nonlinear systems with application to motion control

Abubakar Sani Halilu<sup>1,2</sup>, Mohamad Afendee Mohamed<sup>1</sup>, Mohammed A. Saleh<sup>3</sup>, Kabiru Ahmed<sup>4</sup>, Abdulgader Z. Almaymuni<sup>3</sup>✉, Arunava Majumder<sup>5</sup>, Sulaiman M. Ibrahim<sup>6,7</sup> & Badr Almutairi<sup>8</sup>

The double-step-length procedure presents an inventive approach by incorporating two corrections during each iteration, which improves the robustness of the iterative process. The idea that if one correction fails, the other might still help guide the system in the right direction seems likely to reduce the chance of getting divergence, a common challenge in nonlinear solvers. This article presents a more effective approach to solving systems of nonlinear equations by combining the double-step length iterative method with the Picard-Mann hybrid technique. This combination improves performance. We streamline the process by employing a positive scalar to approximate the Jacobian matrix in Newton's method, thereby eliminating the need to calculate derivatives. An inexact line search technique was used to determine the values of the two-step lengths. The proposed method converges globally under mild conditions. The numerical tests highlight its efficiency, proving that it is more effective than existing double-direction and length methods. Furthermore, we have effectively implemented this technique in motion control applications, showcasing its practicality and efficacy.

**Keywords** Jacobian matrix, Acceleration parameter, Double direction approach, Double step size, Picard–Mann process, Motion tracking, Optimization problems

### Abbreviations

NSE	Nonlinear systems of equations
JM	Jacobian matrix
NM	Newton's method
QNM	Quasi-Newton method
DDM	Double direction method
DSM	Double-step length method
PMHP	Picard–Mann hybrid iterative process

Nonlinear systems of equations (NSE) play a vital role in a broad range of domains, including physics, engineering, economics, and applied mathematics, especially in solving systems of nonlinear differential equations<sup>1–3</sup>. These systems arise in scenarios where the relationship between variables cannot be accurately described using linear equations, making them more complex to solve<sup>4–6</sup>. Researchers are actively working to develop efficient and reliable iterative methods for finding approximate solutions to these equations. A typical nonlinear system of equations can be expressed in the following form:

<sup>1</sup>Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, 22200 Besut, Terengganu, Malaysia.

<sup>2</sup>Department of Mathematics, Sule Lamido University, Kafin Hausa, Nigeria. <sup>3</sup>Department of Cybersecurity, College of Computer, Qassim University, Buraydah, Saudi Arabia. <sup>4</sup>Department of Mathematical Sciences, Bayero University, Kano, Nigeria. <sup>5</sup>Department of Mathematics, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, Chennai 603203, India. <sup>6</sup>College of Applied and Health Sciences, A'Sharqiyah University, PO Box 42, 400 Ibra, Sultanate of Oman. <sup>7</sup>Faculty of Education and Arts, Sohar University, 311 Sohar, Sultanate of Oman. <sup>8</sup>College of Computer and Information Sciences, Majmaah University, Almajmaah, Saudi Arabia. ✉email: almaymuni@qu.edu.sa

$$\mathfrak{F}(s) = 0, \quad (1)$$

where  $\mathfrak{F} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a nonlinear mapping.

These problems can be solved using a variety of iterative strategies. These include methods that do not depend on derivatives<sup>7–13</sup>, as well as Newton's method (NM)<sup>14</sup> and quasi-Newton methods (QNM)<sup>15,16</sup>. Notably, Newton's method is one of the most effective approaches because it is straightforward to implement and converges rapidly. However, it does require the calculation and storage of the Jacobian matrix (JM) at each iteration. These techniques employ a recursive formula to generate a sequence of points.

$$s_{k+1} = s_k + \alpha_k d_k, \quad k = 0, 1, \dots \quad (2)$$

where  $\alpha_k$  is the step size,  $d_k$  is the search direction, and the current and the previous iterations are  $s_{k+1}$  and  $s_k$ , respectively<sup>17,18</sup>. The direction  $d_k$  of the NM is determined by solving:

$$\mathfrak{F}(s_k) + \mathfrak{F}'(s_k)d_k = 0.$$

Here,  $\mathfrak{F}'(s_k)$  is the Jacobian matrix of  $\mathfrak{F}(s_k)$  at  $s_k$ . However, one limitation of Newton's scheme is that it requires the derivative of JM to be computed at each iteration, which may not always be available or may not be precisely determined. In such cases, NM cannot be applied directly.

To solve this problem, researchers created QNM. These methods improve traditional optimisation techniques by replacing the JM or its inverse with a new approximation that updates during each step. This change makes the optimisation process more efficient. In QNM, the search direction is determined by:

$$d_k = -B_k^{-1} \mathfrak{F}(s_k).$$

The problem (1) may be derived from an unconstrained optimisation problem<sup>19</sup>. Let  $h$  be defined as a norm function represented by the equation:

$$h(s) = \frac{1}{2} \|\mathfrak{F}(s)\|^2. \quad (3)$$

Consequently, the nonlinear equations problem (1) can be equivalently expressed as the following global optimisation problem:

$$\min h(s), \quad s \in \mathbb{R}^n, \quad h : \mathbb{R}^n \rightarrow \mathbb{R}. \quad (4)$$

The NM and QNM are among the classical and efficient algorithms for solving optimization problems. However, their major limitation lies in the requirement for significant matrix storage due to the calculation or estimation of the Jacobian at each iteration step. This drawback makes Newton-type algorithms less suitable for large-scale problems. To address this issue, numerous studies have developed matrix-free methods. One effective matrix-free algorithm is the Double Direction Method (DDM)<sup>20</sup>, which generates sequences of iteration points via:

$$s_{k+1} = s_k + \alpha_k a_k + \alpha_k^2 b_k, \quad (5)$$

where  $a_k$  and  $b_k$  define the search directions. The main motivation behind the DDM algorithm is that it incorporates two corrections in the formula described by (5). If one of the correction schemes fails during the iterative process, the other serves as a backup mechanism to ensure convergence and stability of the system.

Recently, Petrović and Stanimirović<sup>20</sup> introduced a new concept to the unconstrained optimization known as the accelerated double direction approach. This pioneering approach incorporates an acceleration parameter  $\delta_k > 0$  to formulate the Hessian matrix approximation, which is expressed as follows:

$$\nabla^2 h(s_k) \approx \delta_k I,$$

with  $I$  denoting the identity matrix and  $\nabla^2 h(s_k)$  representing the Hessian of  $h(s_k)$ . To improve the robustness of this algorithm, Petrović further refine the formula in<sup>21</sup> by introducing a double-step length method (DSM) as follows:

$$s_{k+1} = s_k + \alpha_k a_k + \beta_k b_k. \quad (6)$$

where  $\alpha_k$  and  $\beta_k$  denotes the two independently chosen step lengths, allowing for a more adaptive and flexible approach to convergence. Preliminary results in<sup>21</sup> show that this modified algorithm demonstrates superior performance compared to the initial DDM, attaining a reduction in the iteration count required in addition to decrease in processor time.

To further improve the convergence and robustness of the algorithm, Petrović et al.<sup>22</sup> consider hybridizing the DDM with the Picard-Mann Hybrid Iterative Process (PMHP), presented by Khan<sup>23</sup>. This hybridization utilizes the strengths of both schemes, which results in improved performance when addressing different benchmark

problems. Incorporating the PMHP improves the complexity of the scheme, leading to improvements in both numerical efficiency and convergence of the scheme.

**Definition 1.1** The PMHP is characterized by three relations:

$$s_0 = s \in \Omega, \quad (7)$$

$$s_{k+1} = T(y_k), \quad (8)$$

$$y_k = (1 - \eta_k)s_k + \eta_k T(s_k), \quad k \geq 0, \quad (9)$$

with  $T : \Omega \rightarrow \Omega$  denoting the mapping on a nonempty convex subset  $\Omega$  of a normed space  $\mathbb{E}$ . The sequences  $y_k$  and  $s_k$  are follow from the iterative procedure defined in (9) and (8), and  $\{\eta_k\}_{k \geq 0}$  is a sequence of positive values within the interval  $(0, 1)$ .

Building on the hybridization rule introduced in<sup>23,24</sup> proposed a double-step length scheme equipped with a convergence procedure. This innovative approach was designed to refine the DSM described in<sup>21</sup>, thereby overcoming its earlier limitations.

The current literature regarding derivative-free DDMs for solving NSE is notably scarce. This identified gap in research prompted Halilu and Waziri<sup>9</sup> to develop a derivative-free methodology that employs a DDM to effectively address the equation presented in (1), as articulated in equation (5). Subsequently, in<sup>25</sup>, they developed a modified version of the algorithm in<sup>9</sup>. Their study establishes that the proposed method exhibits global convergence. The new scheme proposed by<sup>25</sup> lays foundation for reliable solutions to nonlinear equations without requiring derivatives. Building upon the existing framework, Abdullahi et al.<sup>11</sup> developed another modification DDM by revising the original formula in<sup>9</sup> and incorporating a conjugate gradient scheme tailored for symmetric nonlinear equations. Their scheme not only retain the derivative-free property of the formula but also facilitated global convergence, which was achieved using the modified line search approach proposed by Li and Fukushima<sup>19</sup>. In another attempt, Halilu and Waziri<sup>26</sup> further modified the DDM discussed in (5) and established global convergence of the new scheme under some mild assumptions. Extensive computational experiments demonstrated the efficiency and robustness of the method. In their insightful studies presented in<sup>12</sup> and<sup>27</sup>, the authors built upon the foundational concepts in equation (6) by introducing the DSM. This innovative method enhances the efficacy of the DDM described in<sup>9</sup>. Their results are particularly encouraging, indicating that the DSM achieves significantly faster convergence than the previous method. These findings not only highlight the potential of their approach to optimize numerical algorithms but also pave the way for further advancements in this area of research.

Motivated by the concepts presented by Petrović in<sup>24</sup>, this study aims to develop an advanced hybrid DSM, as outlined in<sup>12</sup>, for addressing the NSE using the PMHP proposed in<sup>23</sup>. The objective is to enhance both the efficiency and accuracy of identifying solutions to Problem (1).

The contributions from this study are as follows:

- This paper introduces a new iterative algorithm for systems of nonlinear equations.
- To develop a derivative-free approach, the Jacobian matrix is approximated using the acceleration parameter.
- We propose a new algorithm that ensures global convergence.
- The correction parameter is derived using Picard-Mann iterative schemes to enhance the numerical performance of the proposed method.
- The proposed method is successfully applied to address issues related to motion control.

The organisation of this paper is outlined as follows: The subsequent section will detail the derivation of the proposed approach. A comprehensive convergence analysis of the suggested algorithm will be presented in “Convergence analysis”. “Numerical experiments” will illustrate various numerical tests and motion control applications of the proposed method. Finally, the article will be concluded in “Conclusion”, while the list of abbreviations is provided in “Abbreviations” section. In this paper, we refer to  $\mathbb{R}^n$  as the  $n$ -dimensional real space, with the notation  $\|\cdot\|$  representing the Euclidean norm.

## Preliminaries and algorithms

In this section, we present the motivation behind and the algorithm for the proposed method. First, we consider the accelerated DSM for the unconstrained optimisation problem defined in (4). Petrović<sup>21</sup> finds the minimiser of (4) using the DSM. By approximating the Hessian matrix through the acceleration parameter and employing a second-order Taylor expansion, the author derives the following acceleration parameter:

$$\delta_{k+1}^{ADSS} = 2 \frac{\mathfrak{h}(s_{k+1}) - \mathfrak{h}(s_k) + (\alpha_k \delta_k^{-1} + \beta_k) \|\nabla \mathfrak{h}(s_k)\|^2}{(\alpha_k \delta_k^{-1} + \beta_k)^2 \|\nabla \mathfrak{h}(s_k)\|^2}.$$

To enhance the convergence properties and the numerical results of the method in<sup>21</sup>, Petrović in<sup>24</sup> hybridised it with the PMHP by Khan<sup>23</sup>, resulting in the following acceleration parameter.

$$\delta_{k+1}^{HADSS} = 2 \frac{\mathfrak{h}(s_{k+1}) - \mathfrak{h}(s_k) + \eta_k (\alpha_k \delta_k^{-1} + \beta_k) \|\nabla \mathfrak{h}(s_k)\|^2}{\eta_k^2 (\alpha_k \delta_k^{-1} + \beta_k)^2 \|\nabla \mathfrak{h}(s_k)\|^2}, \quad \eta_k \in (0, 1).$$

The enhancement not only improves the algorithm’s efficiency but also ensures greater accuracy in reaching solutions, thereby advancing the overall effectiveness of the method in practical applications. This improvement is made possible by the addition of the correction parameter  $\eta_k$  in the scheme.

Now, let’s refer to DSM in<sup>12</sup>. The approach created a derivative-free method for solving (1) via

$$\mathfrak{F}'(s_k) \approx \delta_k I, \tag{10}$$

where  $\delta_k > 0$ . The procedure in<sup>12</sup> generate the iterates of  $\{s_k\}$  such that  $s_{k+1} = s_k + x_k$ , where  $x_k = (\alpha_k + \beta_k)a_k$  and the direction  $a_k$  is given as

$$a_k = -\delta_k^{-1}\mathfrak{F}(s_k). \tag{11}$$

The acceleration parameter is derived using a first-order Taylor expansion as follows:

$$\delta_{k+1} = \frac{v_k^T v_k}{(\alpha_k + \beta_k)v_k^T a_k}, \tag{12}$$

where  $v_k = \mathfrak{F}(s_{k+1}) - \mathfrak{F}(s_k)$ . The method presented in<sup>12</sup> converged globally. However, its numerical performance is unsatisfactory when  $\delta_k$  approaches or equals 1. This constraint leads us to suggest a hybrid technique that produces better numerical results. Consequently, inspired by the approach in<sup>24</sup>, we are interested in hybridising it with the PMHP. Therefore, we define the mapping  $T$  in Definition 1.1 as

$$T(y_k) = y_k - (\alpha_k + \beta_k)\delta_k^{-1}\mathfrak{F}(s_k). \tag{13}$$

Therefore, from (8), (9), and (13), we have the following three relations:

$$s_0 = s \in \Omega, \tag{14}$$

$$s_{k+1} = y_k - (\alpha_k + \beta_k)\delta_k^{-1}\mathfrak{F}(s_k), \tag{14}$$

$$y_k = (1 - \eta_k)s_k + \eta_k(s_k - (\alpha_k + \beta_k)\delta_k^{-1}\mathfrak{F}(s_k)), \quad k \geq 0. \tag{15}$$

where  $\eta_k \in (0, 1)$  is a correction parameter. Consequently, we have the following lemma.

**Lemma 2.1** Let the accelerated double-direction scheme be generated by the three term iterative process in (14) and (15), then the proposed scheme is given by

$$s_{k+1} = s_k - \rho_k(\alpha_k + \beta_k)\delta_k^{-1}\mathfrak{F}(s_k), \quad \rho_k \in (1, 2). \tag{16}$$

**Proof** By substituting (15) into (14), we have

$$\begin{aligned} s_{k+1} &= (1 - \eta_k)s_k + \eta_k(s_k - (\alpha_k + \beta_k)\delta_k^{-1}\mathfrak{F}(s_k) - (\alpha_k + \beta_k)\delta_k^{-1}\mathfrak{F}(s_k)) \\ &= s_k - \eta_k s_k + \eta_k s_k - \eta_k(\alpha_k + \beta_k)\delta_k^{-1}\mathfrak{F}(s_k) - (\alpha_k + \beta_k)\delta_k^{-1}\mathfrak{F}(s_k) \\ &= s_k - (\eta_k + 1)(\alpha_k + \beta_k)\delta_k^{-1}\mathfrak{F}(s_k). \end{aligned}$$

Taking  $(\eta_k + 1) = \rho_k \in (1, 2)$ , we have (16). □

Now, we are finding the updated value of the acceleration parameter by employing the first-order Taylor expansion of the proposed iteration (16).

$$\mathfrak{F}(s_{k+1}) \approx \mathfrak{F}(s_k) + \mathfrak{F}'(\xi)(s_{k+1} - s_k),$$

where  $\xi \in [s_k, s_{k+1}]$  can be expressed as  $\xi = s_k + \varrho(s_{k+1} - s_k)$ ,  $0 \leq \varrho \leq 1$ . By selecting  $\varrho = 1$ , we have

$$\mathfrak{F}(s_{k+1}) \approx \mathfrak{F}(s_k) - \mathfrak{F}'(s_{k+1})\rho_k(\alpha_k + \beta_k)\delta_k^{-1}\mathfrak{F}(s_k). \tag{17}$$

From (10) and (17), it not difficult to verify

$$v_k \approx \delta_{k+1}x_k, \tag{18}$$

$$x_k = s_{k+1} - s_k.$$

By multiplying both sides of Eq. (18) by the transpose of  $v_k^T$ , the proposed search direction and acceleration parameter are respectively defined as:

$$d_k = -\rho_k\delta_k^{-1}\mathfrak{F}(s_k), \tag{19}$$

$$\delta_{k+1} = -\frac{\delta_k\rho_k^{-1}\|v_k\|^2}{(\alpha_k + \beta_k)\rho_k v_k^T \mathfrak{F}(s_k)}. \tag{20}$$

**Remark 2.2** In this paper, we adopt a consistent parameterisation by setting  $\rho_k = \rho$  for all  $k \geq 0$ . The parameter  $\rho$  is set to be in the interval  $(1, 2)$ . This selection is based on its influence on the algorithm’s convergence properties.

The proposed method algorithm is specified as follows:

- 
- Input:** Given  $s_0, \delta_0 = 1, \epsilon = 10^{-5}, \omega_1, \omega_2 > 0, r, q \in (0, 1), \rho \in (1, 2)$ . Set  $k = 0$ .  
**S1:** Compute  $\mathfrak{F}(s_k)$ .  
**S2:** If  $\|\mathfrak{F}(s_k)\| \leq \epsilon$  then stop; otherwise, continue to **S3**.  
**S3:** Determine the search direction  $d_k = -\rho\delta_k^{-1}\mathfrak{F}(s_k)$ .  
**S4:** Establish  $s_{k+1} = s_k + (\alpha_k + \beta_k)d_k$ , with  $\alpha_k = q^{m_k}$  and  $\beta_k = r^{m_k}$  with  $m_k$  denotes the smallest nonnegative integer  $m$  such that

$$\mathfrak{h}(s_k + (\alpha_k + \beta_k)d_k) - \mathfrak{h}(s_k) \leq -\omega_1\|(\alpha_k + \beta_k)\mathfrak{F}(s_k)\|^2 - \omega_2\|(\alpha_k + \beta_k)d_k\|^2 + \chi_k\mathfrak{h}(s_k). \tag{21}$$

Suppose that  $\{\chi_k\}$  is a positive sequence such that

$$\sum_{k=0}^{\infty} \chi_k < \chi < \infty. \tag{22}$$

- S5:** Compute  $\mathfrak{F}(s_{k+1})$ .  
**S6:** Determine  $\delta_{k+1} = -\frac{\delta_k\|v_k\|^2}{(\alpha_k + \beta_k)\rho v_k^T \mathfrak{F}(s_k)}$ .  
**S7:** Consider  $k = k + 1$  and go to **S2**.
- 

**Algorithm 1.** Modified double step size approach (MIDSL).

---

### Convergence analysis

We present how the proposed Algorithm 1 (MIDSL) converges globally in this section. To begin, let’s define the level set.

$$\Omega = \{s \mid \|\mathfrak{F}(s)\| \leq \|\mathfrak{F}(s_0)\|\}. \tag{23}$$

**Assumption 3.1** The following assumptions have been stated:

1. There exists  $s^* \in \mathbb{R}^n$  such that  $\mathfrak{F}(s^*) = 0$ .
2. The function  $\mathfrak{F}(s)$  is smooth in some neighborhood say  $Q$  of  $s^*$  containing  $\Omega$ .
3. The Jacobian of  $\mathfrak{F}(s)$  is bounded symmetric and positive definite on  $Q$ , i.e., there exist positive constants  $H_1 > H_2 > 0$  such that

$$\|\mathfrak{F}'(s)\| \leq H_1, \quad \forall s \in Q, \tag{24}$$

and

$$H_2\|d\|^2 \leq d^T \mathfrak{F}'(s)d, \quad \forall s \in Q, d \in \mathbb{R}^n. \tag{25}$$

**Remark 3.2** We make the following remark:

Assumption 3.1 implies that there exist constants  $H_1 > H_2 > 0$  such that

$$H_2\|d\| \leq \|\mathfrak{F}'(s)d\| \leq H_1\|d\|, \quad \forall s \in Q, d \in \mathbb{R}^n. \tag{26}$$

$$H_2\|s - v\| \leq \|\mathfrak{F}(s) - \mathfrak{F}(v)\| \leq H_1\|s - v\| \quad \forall s, v \in Q. \tag{27}$$

Since  $\rho^{-1}\delta_k I$  approximates  $\mathfrak{F}'(s_k)$  along  $x_k$ , the following assumption can be made.

**Assumption 3.3**  $\rho^{-1}\delta_k I$  is a good approximation to  $\mathfrak{F}'(s_k)$ , i.e.,

$$\|(\mathfrak{F}'(s_k) - \rho^{-1}\delta_k I)d_k\| \leq \epsilon\|\mathfrak{F}(s_k)\|, \tag{28}$$

where,  $\epsilon \in (0, 1)$  is a small quantity.

*Lemma 3.4* Let  $\{s_k\}$  be produced by the MIDS algorithm, supposing Assumption 3.3 is true. Then  $d_k$  is a sufficient descent direction for  $\mathfrak{h}(s_k)$  at  $x_k$  i.e.,

$$\nabla\mathfrak{h}(s_k)^T d_k < c\|\mathfrak{F}(s_k)\|^2, \quad c > 0. \tag{29}$$

**Proof** From (3), (19), and (28), we have

$$\begin{aligned} \nabla\mathfrak{h}(s_k)^T d_k &= \mathfrak{F}_k^T \mathfrak{F}'(s_k) d_k \\ &= \mathfrak{F}(s_k)^T [(\mathfrak{F}'(s_k) - \rho^{-1}\delta_k I)d_k - \mathfrak{F}(s_k)] \\ &= \mathfrak{F}(s_k)^T (\mathfrak{F}'(s_k) - \rho^{-1}\delta_k I)d_k - \|\mathfrak{F}(s_k)\|^2. \end{aligned} \tag{30}$$

Applying the Cauchy–Schwarz inequality, we have,

$$\begin{aligned} \nabla\mathfrak{h}(s_k)^T d_k &\leq \|\mathfrak{F}(s_k)\| \|(\mathfrak{F}'(s_k) - \rho^{-1}\delta_k I)d_k\| - \|\mathfrak{F}(s_k)\|^2 \\ &\leq -(1 - \epsilon)\|\mathfrak{F}(s_k)\|^2. \end{aligned} \tag{31}$$

Since  $\epsilon \in (0, 1)$ , taking  $c = 1 - \epsilon$ , Lemma 3.4 is true.

We can conclude from Lemma 3.4 that the norm function  $\mathfrak{h}(s_k)$  is a descent along  $d_k$ , which means that  $\|\mathfrak{F}(s_{k+1})\| \leq \|\mathfrak{F}(s_k)\|$  is true.  $\square$

*Lemma 3.5* Let  $\{s_k\}$  be produced by the MIDS algorithm, supposing Assumption 3.3 is true. Then  $\{s_k\} \subset \Omega$ .

**Proof** From Lemma 3.4, we have  $\|\mathfrak{F}(s_{k+1})\| \leq \|\mathfrak{F}(s_k)\|$ . Furthermore, for all  $k$ ,

$$\|\mathfrak{F}(s_{k+1})\| \leq \|\mathfrak{F}(s_k)\| \leq \|\mathfrak{F}(s_{k-1})\| \leq \dots \leq \|\mathfrak{F}(s_0)\|.$$

This means that  $\{s_k\} \subset \Omega$ .  $\square$

*Lemma 3.6* Let  $\{s_k\}$  be produced by the MIDS algorithm, supposing Assumption 3.3 is true. Then there exists a constant  $H_2 > 0$  such that for all  $k$ ,

$$v_k^T x_k \geq H_2 \|x_k\|^2, \tag{32}$$

where  $x_k = (\alpha_k + \beta_k)d_k$ .

**Proof** From mean-value theorem and (25),

$$v_k^T x_k = x_k^T (\mathfrak{F}(s_{k+1}) - \mathfrak{F}(s_k)) = x_k^T F'(\xi)x_k \geq H_2 \|x_k\|^2.$$

Where  $\xi = s_k + \kappa(s_{k+1} - s_k)$ ,  $\kappa \in (0, 1)$ .

As shown in the related Lemma 3.6 and (27), the following inequality is true.

$$\frac{v_k^T x_k}{\|x_k\|^2} \geq H_2, \quad \frac{\|v_k\|^2}{v_k^T x_k} \leq \frac{H_1^2}{H_2}. \tag{33}$$

$\square$

*Lemma 3.7* Let  $\{s_k\}$  be produced by the MIDS algorithm, supposing Assumption 3.3 is true. Then we have

$$\lim_{k \rightarrow \infty} \|(\alpha_k + \beta_k)d_k\| = 0, \tag{34}$$

and

$$\lim_{k \rightarrow \infty} \|(\alpha_k + \beta_k)\mathfrak{F}(s_k)\| = 0. \tag{35}$$

**Proof** From (21) for all  $k > 0$

$$\begin{aligned} \omega_2 \|(\alpha_k + \beta_k)d_k\|^2 &\leq \omega_1 \|(\alpha_k + \beta_k)\mathfrak{F}(s_k)\|^2 + \omega_2 \|(\alpha_k + \beta_k)d_k\|^2 \\ &\leq \|\mathfrak{F}(s_k)\|^2 - \|\mathfrak{F}(s_{k+1})\|^2 + \chi_k \|\mathfrak{F}(s_k)\|^2. \end{aligned} \tag{36}$$

By summing the above inequality, we have

$$\begin{aligned}
 \omega_2 \sum_{i=0}^k \|(\alpha_i + \beta_i)d_i\|^2 &\leq \sum_{i=0}^k (\|\mathfrak{F}(s_i)\|^2 - \|\mathfrak{F}(s_{i+1})\|^2) + \sum_{i=0}^k \chi_i \|\mathfrak{F}(s_i)\|^2, \\
 &= \|\mathfrak{F}(s_0)\|^2 - \|\mathfrak{F}(s_{k+1})\|^2 + \sum_{i=0}^k \chi_i \|\mathfrak{F}(s_i)\|^2, \\
 &\leq \|\mathfrak{F}(s_0)\|^2 + \|\mathfrak{F}(s_0)\|^2 \sum_{i=0}^k \chi_i, \\
 &\leq \|\mathfrak{F}(s_0)\|^2 + \|\mathfrak{F}(s_0)\|^2 \sum_{i=0}^{\infty} \chi_i.
 \end{aligned}
 \tag{37}$$

Based on the level set and the fact that the sequence  $\{\chi_k\}$  meets the criterion in (22), it follows that the series  $\sum_{i=0}^{\infty} \|(\alpha_i + \beta_i)d_i\|^2$  converges. This leads to the conclusion in (34). By applying the same reasoning as above, but now considering  $\omega_1 \|(\alpha_k + \beta_k)\mathfrak{F}_k\|^2$  on the left side, we arrive at (35).  $\square$

**Lemma 3.8** *Let  $\{s_k\}$  be produced by the MIDS algorithm, supposing Assumption 3.3 is true. Then there exists a constant  $M_1 > 0$  such that for all  $k > 0$ ,*

$$\|d_k\| \leq M_1.
 \tag{38}$$

**Proof** From (19), (20), and (27), we have

$$\begin{aligned}
 \|d_k\| &= \left\| -\rho \frac{(\alpha_{k-1} + \beta_{k-1})v_{k-1}^T d_{k-1} \mathfrak{F}(s_k)}{v_{k-1}^T v_{k-1}} \right\| \\
 &= \left\| -\rho \frac{v_{k-1}^T x_{k-1} \mathfrak{F}(s_k)}{\|v_{k-1}\|^2} \right\| \\
 &\leq \frac{\rho \|\mathfrak{F}(s_k)\| \|x_{k-1}\| \|v_{k-1}\|}{H_2^2 \|x_{k-1}\|^2} \\
 &\leq \frac{\rho \|\mathfrak{F}(s_k)\| H_1 \|x_{k-1}\|}{H_2^2 \|x_{k-1}\|} \\
 &\leq \frac{\rho \|\mathfrak{F}(s_k)\| H_1}{H_2^2} \\
 &\leq \frac{\rho \|\mathfrak{F}(s_0)\| H_1}{H_2^2}.
 \end{aligned}
 \tag{39}$$

Taking  $M_1 = \frac{\rho \|\mathfrak{F}(s_0)\| H_1}{H_2^2}$ , we have (38).  $\square$

**Theorem 3.9** *Let  $\{s_k\}$  be produced by the MIDS algorithm, supposing Assumption 3.3 is true. Assume further for all  $k > 0$ ,*

$$(\alpha_k + \beta_k) \geq \lambda \frac{|\mathfrak{F}(s_k)^T d_k|}{\|d_k\|^2},
 \tag{40}$$

where  $\lambda$  is some positive constant. Then

$$\lim_{k \rightarrow \infty} \|\mathfrak{F}(s_k)\| = 0.
 \tag{41}$$

**Proof** From Lemma 3.8 we have (38). Also, from (34) and the boundedness of  $\{\|d_k\|\}$ , we have

$$\lim_{k \rightarrow \infty} (\alpha_k + \beta_k) \|d_k\|^2 = 0,
 \tag{42}$$

from (40) and (42) we have

$$\lim_{k \rightarrow \infty} |\mathfrak{F}(s_k)^T d_k| = 0.
 \tag{43}$$

Also, from (19) we have,

$$\mathfrak{F}(s_k)^T d_k = -\rho \delta_k^{-1} \|\mathfrak{F}(s_k)\|^2, \tag{44}$$

$$\begin{aligned} \|\mathfrak{F}(s_k)\|^2 &= |-\mathfrak{F}(s_k)^T d_k \rho^{-1} \delta_k| \\ &\leq \rho^{-1} |\delta_k| |\mathfrak{F}_k^T d_k|. \end{aligned} \tag{45}$$

Since

$$\delta_k^{-1} = \frac{(\alpha_k + \beta_k) v_k^T d_k}{v_{k-1}^T v_{k-1}} = \frac{v_{k-1}^T x_{k-1}}{\|v_{k-1}\|^2} \geq \frac{h \|x_{k-1}\|^2}{\|v_{k-1}\|^2} \geq \frac{H_2 \|x_{k-1}\|^2}{H_1^2 \|x_{k-1}\|^2} = \frac{H_2}{H_1^2}.$$

Then,

$$|\delta_k^{-1}| \geq \frac{H_2}{H_1^2}.$$

Therefore from (45) we have,

$$\|\mathfrak{F}(s_k)\|^2 \leq |\mathfrak{F}(s_k)^T d_k| \left( \frac{H_1^2}{\rho H_2} \right). \tag{46}$$

Consequently,

$$0 \leq \|\mathfrak{F}(s_k)\|^2 \leq |\mathfrak{F}(s_k)^T d_k| \left( \frac{H_1^2}{\rho H_2} \right) \rightarrow 0. \tag{47}$$

Hence,

$$\lim_{k \rightarrow \infty} \|\mathfrak{F}(s_k)\| = 0. \tag{48}$$

The proof is complete.  $\square$

### Numerical experiments

In this section, the study discusses the computational efficiency of the methods on a system of nonlinear equations with application to motion control problems, illustrating the efficiency and relevance of the proposed method real-world situations.

#### Experiments on system of nonlinear equations

The robustness and efficiency of the new (MIDSL) scheme was assessed by comparing with other existing algorithms including:

1. A double step length method (IDS) presented in<sup>12</sup>.
2. A modified double-direction method (MDFDD) is presented in<sup>25</sup>.

The computer codes used in this study were developed using Matlab version 9.4.0 (R2018a) and executed on a PC that features an 8 GB of RAM and 1.80 GHz CPU processor. All three methods run under similar line search method, as detailed in equation (21). Throughout the experiments, the parameters for the IDS and MDFDD algorithms were set according to the settings outlined in<sup>12</sup> and<sup>25</sup>, respectively. For the MIDSL method,

the following parameter values were used:  $\omega_1 = \omega_2 = 10^{-4}$ ,  $\rho = 1.2$ ,  $q = 0.2$ ,  $r = 0.3$  and  $\chi_k = \frac{1}{(k+1)^2}$ .

The program execution terminates if the iteration count exceeds 1000 or if  $\|\mathfrak{F}(s_k)\| \leq 10^{-5}$ . The symbol ‘-’ denotes a failure when the number of iterations exceeds 1000 without discovering a point of  $s_k$  that meets the established stopping criterion. To demonstrate the comprehensive numerical experiments conducted with the three methods, we applied these techniques to six benchmark test problems, using various initial points and dimensions ranging from 1000 to 100,000.

The initial guesses presented in Table 1 were utilized in the experiments:

and the subsequent test problems were selected based on the SNE to illustrate the performance of the proposed algorithm.

#### Problem 1<sup>7</sup>

$$\begin{aligned} \mathfrak{F}_1(s) &= s_1 - e^{\cos\left(\frac{s_1+s_2}{s_{i-1}+s_i+s_{i+1}}\right)}, \\ \mathfrak{F}_i(s) &= s_i - e^{\cos\left(\frac{n+1}{s_{n-1}+s_n}\right)}, \\ \mathfrak{F}_n(s) &= s_n - e^{\cos\left(\frac{n+1}{n+1}\right)}, \quad i = 2, 3, \dots, n-1. \end{aligned}$$

#### Problem 2<sup>12</sup>

$$\mathfrak{F}_i(s) = (1 - s_i^2) + s_i(1 + s_i s_{n-2} s_{n-1} s_n) - 2, \quad i = 1, 2, \dots, n.$$

#### Problem 3<sup>9</sup>

$$\mathfrak{F}_i(s) = s_i - 3s_i \left( \frac{\sin s_i}{3} - \frac{33}{50} \right) + 2, \quad i = 1, 2, \dots, n.$$

INITIAL POINTS (IP)	VALUES
$s(1)$	$\left(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2}\right)^T$
$s(2)$	$\left(\frac{1}{5}, \frac{1}{5}, \dots, \frac{1}{5}\right)^T$
$s(3)$	$\left(\frac{3}{2}, \frac{3}{2}, \dots, \frac{3}{2}\right)^T$
$s(4)$	$\left(\frac{2}{5}, \frac{2}{5}, \dots, \frac{2}{5}\right)^T$
$s(5)$	$\left(0, \frac{1}{2}, \frac{2}{3}, \dots, 1 - \frac{1}{n}\right)^T$
$s(6)$	$\left(\frac{1}{4}, -\frac{1}{4}, \dots, \frac{(-1)^n}{4}\right)^T$

**Table 1.** Initial points.

**Problem 4**<sup>27</sup>

$$\tilde{\mathfrak{F}}_i(s) = s_i - \left(1 - \frac{c}{2n} \sum_{j=1}^n \frac{\mu_i s_j}{\mu_i + \mu_j}\right)^{-1}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n \text{ with } c \in [0, 1) \text{ and}$$

$$\mu = \frac{i-0.5}{n}. \text{ (In our experiment we take } c=0.1\text{).}$$

**Problem 5**<sup>25</sup>

$$\tilde{\mathfrak{F}}_i(s) = 2s_i - \sin |s_i|, \quad i = 1, 2, \dots, n.$$

**Problem 6**<sup>25</sup>

$$\begin{aligned} \tilde{\mathfrak{F}}_1(s) &= (s_1^2 + s_2^2)s_1 - 1, \\ \tilde{\mathfrak{F}}_i(s) &= (s_{i-1}^2 + 2s_i^2 + s_{i+1}^2)s_i - 1, \\ \tilde{\mathfrak{F}}_n(s) &= (s_{n-1}^2 + s_n^2)s_n, \quad i = 2, 3, \dots, n-1. \end{aligned}$$

Detailed results of the numerical experiment are reported in Tables 2, 3 and 4. In evaluating these methods, we focused on three key metrics: the number of iterations (Iter), function evaluations (Fun), CPU time in seconds (Time(s)) required for each approach to attain a solution, and the residual norm  $\|\tilde{\mathfrak{F}}(s_k)\|$ . Below, we now analyse these metrics and their significance.

The number of iterations represents the cycles the algorithm undergoes to find a solution. Typically, a lower number of iterations indicates a more efficient algorithm. The tables show that the MIDSL method requires fewer iterations than both the IDS and MDFDD methods, demonstrating its effectiveness in solving the problem. Function evaluations denote the frequency with which the algorithm computes the value of the objective function during its operation. While the function evaluations for the proposed methods are considerably higher than those for the IDS method, the proposed methods still achieve a solution more rapidly than IDS. CPU time refers to the actual duration required by the algorithm to compute a solution, influenced by the complexity of each iteration and the overall number of iterations. In practical terms, a shorter CPU time signifies a faster and more efficient method. Our findings indicate that the MIDSL consistently outperforms the other methods in terms of CPU time, often arriving at a solution more quickly than both IDS and MDFDD. Moreover, the accuracy of the solution is evaluated through the residual norm, which indicates how well the computed solution meets the equations of the original problem. The tables reveal that all three methods achieve a sufficiently low residual norm, ensuring their effectiveness in solving problem (1). Upon analysing the results, it becomes evident that all three methods seek to address the issue defined by equation (1). Nevertheless, the proposed method not only converges to the answer more rapidly but also requires significantly fewer iterations and less CPU processing time than those used for comparison. In addition, the MIDSL method demonstrates its superiority by successfully solving all the problems that the MDFDD method was unable to tackle. A notable example is found in Problem 4, where the MDFDD method failed to arrive at a solution, while the proposed method delivered effective results. This highlights the reliability and efficiency of the proposed method in addressing the nonlinear problems.

Leveraging the performance profile established by Dolan and Moré<sup>28</sup>, we have generated Figures 1, 2 and 3 to provide a detailed overview of the performance and efficiency of three distinct methods. Each figure conveys essential information about the fraction  $P(\tau)$  of problems for which a particular method operates within a factor  $\tau$  of the optimal execution time. From Figs. 1 and 3, it is clear that the curves representing the MIDSL method outperform those of both the IDS method and the MDFDD method. In most cases, the MIDSL method delivers superior performance. Nevertheless, Fig. 2 reveals instances where the IDS method significantly surpasses the MIDSL method for a specific set of problems, highlighting that while the MIDSL method is generally more effective, there are particular scenarios in which it may not be the optimal choice. Ultimately, these observations suggest that the proposed MIDSL method excels in efficiency, evidenced by a reduction in both the number of iterations required and the CPU time consumed (measured in seconds). The graphical representations in the figures demonstrate that our method is adept at solving large-scale nonlinear systems of equations, establishing its value in practical applications.

### Application of the proposed method in motion control models

In this section, the problems arising from motion control involving three-joint planar robotic manipulators are considered. Since (1) is equivalent to the global optimisation problem (4), we can apply the MIDSL algorithm

Problem	Dim	IP	MIDSL				IDS				MDFDD			
			Iter	Fun	Time(s)	$\ \mathfrak{F}(s_k)\ $	Iter	Fun	Time(s)	$\ \mathfrak{F}(s_k)\ $	Iter	Fun	Time(s)	$\ \mathfrak{F}(s_k)\ $
P1	1,000	s(1)	7	1	0.0039	5.62E-06	20	1	0.0243	8.84E-06	47	92	0.0315	8.28E-06
		s(2)	7	1	0.0034	4.12E-06	21	1	0.0089	5.13E-06	47	92	0.0224	9.41E-06
		s(3)	7	1	0.0038	6.54E-06	19	1	0.0032	7.97E-06	45	88	0.0118	7.88E-06
		s(4)	7	1	0.0029	4.62E-06	20	1	0.0046	9.30E-06	47	92	0.0235	8.66E-06
		s(5)	7	1	0.0029	6.62E-06	20	1	0.0046	7.97E-06	47	92	0.0091	9.45E-06
		s(6)	7	1	0.0058	6.82E-06	21	1	0.0038	6.04E-06	48	94	0.0119	8.43E-06
	10,000	s(1)	7	1	0.0040	2.09E-06	22	1	0.0054	5.25E-06	58	113	0.0140	7.71E-06
		s(2)	7	1	0.0042	2.38E-06	22	1	0.0037	5.97E-06	58	113	0.0335	8.76E-06
		s(3)	7	1	0.0033	1.15E-06	21	1	0.0062	5.76E-06	55	107	0.0130	9.66E-06
		s(4)	7	1	0.0033	2.19E-06	22	1	0.0056	5.49E-06	58	113	0.0136	8.06E-06
		s(5)	7	1	0.0034	1.67E-06	21	1	0.0056	8.40E-06	57	111	0.0218	8.12E-06
		s(6)	7	1	0.0040	2.80E-06	22	1	0.0060	7.03E-06	59	115	0.0118	7.84E-06
	100,000	s(1)	7	1	0.0075	7.01E-06	23	1	0.0139	8.36E-06	90	149	0.0463	8.60E-06
		s(2)	7	1	0.0075	7.96E-06	23	1	0.0114	9.49E-06	90	149	0.0586	9.76E-06
		s(3)	7	1	0.0067	3.85E-06	22	1	0.0131	9.18E-06	88	145	0.0463	8.18E-06
		s(4)	7	1	0.0063	7.33E-06	23	1	0.0118	8.74E-06	90	149	0.0557	8.99E-06
		s(5)	7	1	0.0065	5.46E-06	23	1	0.0185	6.51E-06	89	147	0.0418	8.81E-06
		s(6)	7	1	0.0064	9.38E-06	24	1	0.0152	5.59E-06	91	151	0.0426	8.75E-06
P2	1,000	s(1)	9	1	0.0093	1.27E-06	18	2	0.0126	6.45E-06	8	17	0.0053	4.99E-07
		s(2)	5	2	0.0030	1.65E-06	19	2	0.0043	8.30E-06	9	19	0.0065	2.82E-06
		s(3)	10	4	0.0037	1.63E-06	16	2	0.0042	9.39E-06	11	23	0.0041	4.66E-07
		s(4)	8	1	0.0036	2.98E-06	17	2	0.0040	6.44E-06	8	17	0.0077	2.89E-06
		s(5)	14	3	0.0042	8.35E-06	23	1	0.0046	6.90E-06	40	62	0.0101	9.13E-06
		s(6)	6	2	0.0032	4.36E-06	19	2	0.0057	5.68E-06	11	23	0.0036	7.06E-07
	10,000	s(1)	9	1	0.0039	4.01E-06	20	2	0.0033	5.10E-06	8	17	0.0038	1.58E-06
		s(2)	5	2	0.0026	5.21E-06	21	2	0.0048	6.56E-06	9	19	0.0041	8.91E-06
		s(3)	10	4	0.0045	5.15E-06	18	2	0.0044	7.42E-06	11	23	0.0043	1.47E-06
		s(4)	8	1	0.0035	9.44E-06	19	2	0.0045	5.09E-06	8	17	0.0028	9.14E-06
		s(5)	13	2	0.0046	4.84E-06	23	1	0.0050	9.47E-06	69	112	0.0111	8.33E-06
		s(6)	7	2	0.0042	1.38E-06	20	2	0.0048	8.98E-06	11	23	0.0034	2.23E-06
	100,000	s(1)	10	1	0.0061	1.27E-06	21	2	0.0085	8.07E-06	8	17	0.0063	4.99E-06
		s(2)	6	2	0.0052	1.65E-06	23	2	0.0060	5.19E-06	10	21	0.0073	1.13E-06
		s(3)	11	4	0.0061	1.63E-06	20	2	0.0109	5.87E-06	11	23	0.0054	4.66E-06
		s(4)	9	1	0.0051	2.98E-06	20	2	0.0085	8.05E-06	9	19	0.0060	1.16E-06
		s(5)	14	5	0.0080	2.38E-06	22	1	0.0057	7.23E-06	70	114	0.0197	8.80E-06
		s(6)	7	2	0.0050	4.36E-06	22	2	0.0120	7.10E-06	11	23	0.0088	7.06E-06

**Table 2.** Numerical experiments for problems 1 & 2 with respect to MIDSL, IDS & MDFDD methods.

to solve this problem. The kinematics model provides a detailed framework for understanding the orientation and position of each component within a robotic system. This model represents the dynamics of the various components and their interactions, facilitating a comprehensive analysis of the robot’s overall functionality and performance. The model can be described as follows.

$$\varphi(\theta) = \begin{bmatrix} a_1 + a_2 + a_3 \\ b_1 + b_2 + b_3 \end{bmatrix}, \tag{49}$$

where the kinematics are mapped using the function  $\varphi(\cdot)$ ,  $a_1 = l_1 \cos(\theta_1)$ ,  $a_2 = l_2 \cos(\theta_1 + \theta_2)$ ,  $a_3 = l_3 \cos(\theta_1 + \theta_2 + \theta_3)$ ,  $b_1 = l_1 \sin(\theta_1)$ ,  $b_2 = l_2 \sin(\theta_1 + \theta_2)$ , and  $b_3 = l_3 \sin(\theta_1 + \theta_2 + \theta_3)$ . Additionally, the  $j^{th}$  rod’s lengths are  $l_1, l_2$  and  $l_3$ . The vectors  $\varphi(\theta) \in \mathbb{R}^3$  and  $\theta \in \mathbb{R}^3$  correspond to the end effector position vector and joint angle vector, respectively. Now, at any time interval  $t_k \in [0, t_f]$ , the minimisation issue may be addressed as

$$\min_{\theta \in \mathbb{R}^3} \frac{1}{2} \|\varphi(\theta) - u_{t_k}\|^2, \tag{50}$$

where  $t_f$  is the end of task duration and  $u_{t_k} \in \mathbb{R}^2$  denotes the vector of the desired path at time  $t_k$ . According to<sup>29</sup>, the rod lengths are chosen to be equal to 1, and a Lissajous curve is tracked to control the end effector as

Problem	Dim	IP	MIDSL				IDS				MDFDD			
			Iter	Fun	Time(s)	$\ \mathfrak{F}(s_k)\ $	Iter	Fun	Time(s)	$\ \mathfrak{F}(s_k)\ $	Iter	Fun	Time(s)	$\ \mathfrak{F}(s_k)\ $
P3	1,000	s(1)	7	2	0.0091	4.38E-06	21	1	0.0119	5.77E-06	6	13	0.0026	1.05E-06
		s(2)	7	2	0.0032	1.54E-06	21	1	0.0043	5.84E-06	5	11	0.0037	9.36E-06
		s(3)	8	2	0.0029	6.53E-06	19	1	0.0041	9.67E-06	7	15	0.0079	5.55E-07
		s(4)	7	2	0.0028	3.44E-06	21	1	0.0084	5.92E-06	6	13	0.0032	7.61E-07
		s(5)	7	2	0.0036	2.32E-06	20	1	0.0033	9.58E-06	6	13	0.0027	2.72E-06
		s(6)	6	2	0.0039	1.78E-06	20	1	0.0086	6.83E-06	5	11	0.0030	1.26E-06
	10,000	s(1)	8	2	0.0031	1.39E-06	22	1	0.0199	9.12E-06	6	13	0.0075	3.32E-06
		s(2)	7	2	0.0038	4.87E-06	22	1	0.0053	9.23E-06	6	13	0.0028	1.20E-06
		s(3)	9	2	0.0042	2.07E-06	21	1	0.0071	7.65E-06	7	15	0.0068	1.75E-06
		s(4)	8	2	0.0040	1.09E-06	22	1	0.0035	9.36E-06	6	13	0.0084	2.41E-06
		s(5)	8	2	0.0035	1.01E-06	22	1	0.0047	6.55E-06	7	15	0.0036	5.21E-07
		s(6)	6	2	0.0028	5.63E-06	22	1	0.0049	5.40E-06	5	11	0.0026	3.99E-06
	100,000	s(1)	8	2	0.0057	4.38E-06	24	1	0.0068	7.21E-06	7	15	0.0061	4.27E-07
		s(2)	8	2	0.0069	1.54E-06	24	1	0.0093	7.30E-06	6	13	0.0108	3.80E-06
		s(3)	9	2	0.0058	6.53E-06	23	1	0.0068	6.05E-06	7	15	0.0111	5.55E-06
		s(4)	8	2	0.0050	3.44E-06	24	1	0.0110	7.40E-06	6	13	0.0049	7.61E-06
		s(5)	8	2	0.0058	4.94E-06	24	1	0.0102	5.00E-06	7	15	0.0058	1.77E-06
		s(6)	7	2	0.0054	1.78E-06	23	1	0.0060	8.54E-06	6	13	0.0049	5.12E-07
P4	1,000	s(1)	6	1	0.0089	2.04E-06	16	1	0.0172	9.85E-06	21	40	0.0070	7.08E-06
		s(2)	6	1	0.0029	3.82E-06	14	1	0.0068	5.91E-06	17	35	0.0077	8.59E-06
		s(3)	6	1	0.0028	2.78E-06	18	1	0.0047	8.01E-06	46	62	0.0107	9.37E-06
		s(4)	6	1	0.0035	6.85E-06	16	1	0.0043	7.39E-06	19	37	0.0063	5.87E-06
		s(5)	9	1	0.0037	9.17E-06	19	1	0.0077	8.19E-06	60	102	0.0271	9.76E-06
		s(6)	7	1	0.0037	1.07E-06	15	1	0.0050	6.28E-06	16	32	0.0038	9.46E-06
	10,000	s(1)	7	1	0.0041	2.99E-06	18	1	0.0098	8.52E-06	-	-	-	-
		s(2)	6	1	0.0037	6.28E-06	17	1	0.0049	6.81E-06	835	848	0.1936	6.48E-06
		s(3)	7	1	0.0039	3.25E-06	20	1	0.0053	6.39E-06	-	-	-	-
		s(4)	4	1	0.0029	6.52E-06	18	1	0.0050	6.81E-06	-	-	-	-
		s(5)	9	1	0.0039	3.68E-06	21	1	0.0052	7.56E-06	-	-	-	-
		s(6)	6	1	0.0036	6.18E-06	17	1	0.0048	8.52E-06	-	-	-	-
	100,000	s(1)	7	1	0.0056	5.69E-06	20	1	0.0101	6.74E-06	-	-	-	-
		s(2)	7	1	0.0071	9.67E-06	19	1	0.0066	5.39E-06	-	-	-	-
		s(3)	8	1	0.0063	4.56E-06	22	1	0.0669	5.06E-06	-	-	-	-
		s(4)	7	1	0.0059	3.78E-06	20	1	0.0302	5.39E-06	-	-	-	-
		s(5)	7	1	0.0059	7.50E-06	22	1	0.0092	7.50E-06	-	-	-	-
		s(6)	7	1	0.0052	9.34E-06	19	1	0.0069	6.74E-06	-	-	-	-

**Table 3.** Numerical experiments for problems 3 & 4 with respect to MIDSL, IDS & MDFDD methods.

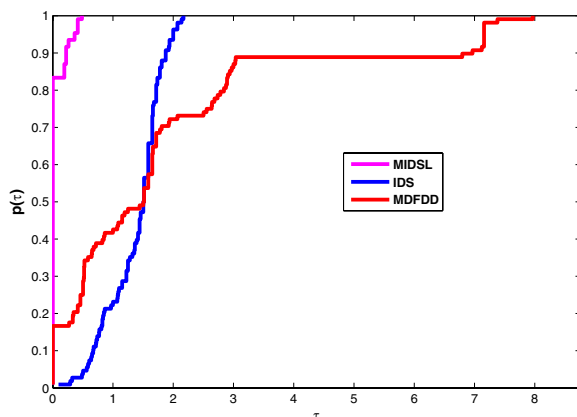
$$u_{t_k} = \begin{bmatrix} \frac{3}{2} + \frac{2}{5} \sin\left(\frac{\pi t_k}{5}\right) \\ \frac{\sqrt{3}}{2} + \frac{2}{5} \sin\left(\frac{\pi t_k}{5} + \frac{\pi}{3}\right) \end{bmatrix}. \tag{51}$$

We set the following parameters for this experiment:  $t_f = 10$  seconds,  $\omega_1 = \omega_2 = 10^{-4}$ ,  $\rho = 1.2$ ,  $q = 0.2$ ,  $r = 0.3$ , and the starting point  $\theta_0 = [\theta_1, \theta_2, \theta_3] = [0, \frac{\pi}{3}, \frac{\pi}{2}]^T$ . For fairness of comparison, the parameters of IDS and MDFDD are chosen precisely as specified in<sup>12</sup> and<sup>25</sup>, respectively. Furthermore, the work duration  $[0, 10]$  is divided into 200 equal portions.

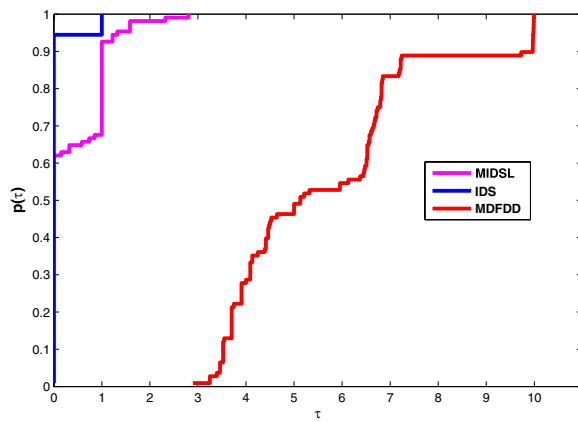
Figures 4, 5, 6 and 7 provide a comprehensive visualisation of the robot trajectories generated by the MIDSL algorithm. The patterns observed in these figures indicate that the MIDSL algorithm not only meets the given task objectives but does so with high efficiency. Specifically, Figures 6 and 7 highlight the precision of three algorithms by showing that the errors incurred in MIDSL and MDFDD are both approximately  $10^{-5}$ , whereas one in IDS algorithm is around  $10^{-1}$ . This minimal error margin emphasises the reliability and effectiveness of our proposed method in achieving accurate results in robotic trajectory planning. Overall, these findings support the practical applicability of the MIDSL algorithm in real-world problems.

Problem	Dim	IP	MIDSL				IDS				MDFDD			
			Iter	Fun	Time(s)	$\ \mathfrak{F}(s_k)\ $	Iter	Fun	Time(s)	$\ \mathfrak{F}(s_k)\ $	Iter	Fun	Time(s)	$\ \mathfrak{F}(s_k)\ $
P5	1,000	s(1)	6	1	0.0060	1.37E-06	18	1	0.0243	6.23E-06	44	89	0.0275	8.74E-06
		s(2)	5	1	0.0030	6.34E-06	16	1	0.0037	9.71E-06	41	83	0.0058	8.17E-06
		s(3)	8	1	0.0033	4.19E-06	19	1	0.0097	8.59E-06	47	95	0.0105	9.44E-06
		s(4)	6	1	0.0034	1.21E-06	17	1	0.0073	9.87E-06	43	87	0.0101	9.30E-06
		s(5)	7	1	0.0035	1.50E-06	18	1	0.0028	9.64E-06	45	91	0.0155	9.56E-06
		s(6)	7	1	0.0031	9.27E-06	17	1	0.0037	9.44E-06	10	21	0.0149	7.09E-06
	10,000	s(1)	6	1	0.0030	4.34E-06	19	1	0.0043	9.85E-06	48	97	0.0112	9.22E-06
		s(2)	6	1	0.0046	2.00E-06	18	1	0.0042	7.68E-06	45	91	0.0065	8.62E-06
		s(3)	9	1	0.0057	1.33E-06	21	1	0.0031	6.79E-06	51	103	0.0080	9.96E-06
		s(4)	6	1	0.0042	3.81E-06	19	1	0.0042	7.81E-06	47	95	0.0161	9.81E-06
		s(5)	8	1	0.0036	2.19E-06	20	1	0.0043	9.64E-06	50	101	0.0071	9.44E-06
		s(6)	8	1	0.0038	2.93E-06	19	1	0.0030	7.47E-06	11	23	0.0075	6.28E-06
	100,000	s(1)	7	1	0.0040	1.37E-06	21	1	0.0079	7.79E-06	52	105	0.0188	9.73E-06
		s(2)	6	1	0.0043	6.34E-06	20	1	0.0075	6.07E-06	49	99	0.0296	9.09E-06
		s(3)	9	1	0.0047	4.19E-06	23	1	0.0092	5.37E-06	56	113	0.0419	7.99E-06
		s(4)	7	1	0.0036	1.21E-06	21	1	0.0177	6.17E-06	52	105	0.1486	7.87E-06
		s(5)	8	1	0.0037	8.28E-06	22	1	0.0107	7.92E-06	55	111	0.0251	7.82E-06
		s(6)	8	1	0.0049	9.27E-06	21	1	0.0085	5.90E-06	12	25	0.0131	5.56E-06
P6	1,000	s(1)	24	5	0.0103	7.39E-06	42	3	0.0131	9.16E-06	33	64	0.0179	7.68E-06
		s(2)	24	7	0.0059	9.19E-06	34	3	0.0074	9.74E-06	34	66	0.0060	7.54E-06
		s(3)	24	7	0.0055	8.78E-06	40	3	0.0048	9.11E-06	34	67	0.0056	9.26E-06
		s(4)	23	6	0.0054	9.91E-06	38	2	0.0069	9.26E-06	33	64	0.0065	9.68E-06
		s(5)	25	5	0.0060	7.93E-06	35	4	0.0066	8.80E-06	36	70	0.0071	7.47E-06
		s(6)	25	4	0.0058	6.46E-06	38	2	0.0062	9.08E-06	36	70	0.0278	8.68E-06
	10,000	s(1)	23	6	0.0064	8.97E-06	41	4	0.0096	9.14E-06	33	64	0.0107	7.48E-06
		s(2)	24	6	0.0056	7.25E-06	36	3	0.0086	8.79E-06	34	66	0.0145	7.44E-06
		s(3)	27	9	0.0077	9.73E-06	43	5	0.0100	9.36E-06	34	67	0.0126	9.08E-06
		s(4)	23	5	0.0046	9.02E-06	45	2	0.0095	7.36E-06	33	64	0.0105	9.45E-06
		s(5)	27	7	0.0051	4.43E-06	33	1	0.0078	8.83E-06	36	70	0.0115	7.41E-06
		s(6)	23	5	0.0051	8.87E-06	34	4	0.0054	7.03E-06	36	70	0.0252	7.79E-06
	100,000	s(1)	24	6	0.0113	7.02E-06	41	3	0.0132	9.68E-06	33	64	0.0148	6.61E-06
		s(2)	24	6	0.0126	3.26E-06	30	3	0.0085	7.16E-06	34	66	0.0166	7.44E-06
		s(3)	27	10	0.0117	8.22E-06	46	9	0.0152	8.87E-06	34	67	0.0161	9.04E-06
		s(4)	26	6	0.0129	6.14E-06	46	3	0.0312	6.43E-06	33	64	0.0126	9.46E-06
		s(5)	29	9	0.0137	6.18E-06	31	3	0.0110	8.53E-06	35	68	0.0135	9.72E-06
		s(6)	27	6	0.0110	5.94E-06	42	6	0.0141	9.11E-06	36	70	0.0164	7.80E-06

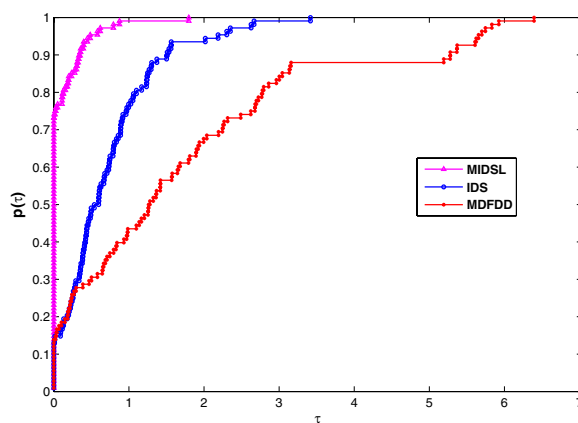
**Table 4.** Numerical experiments for problems 5 & 6 with respect to MIDSL, IDS & MDFDD methods.



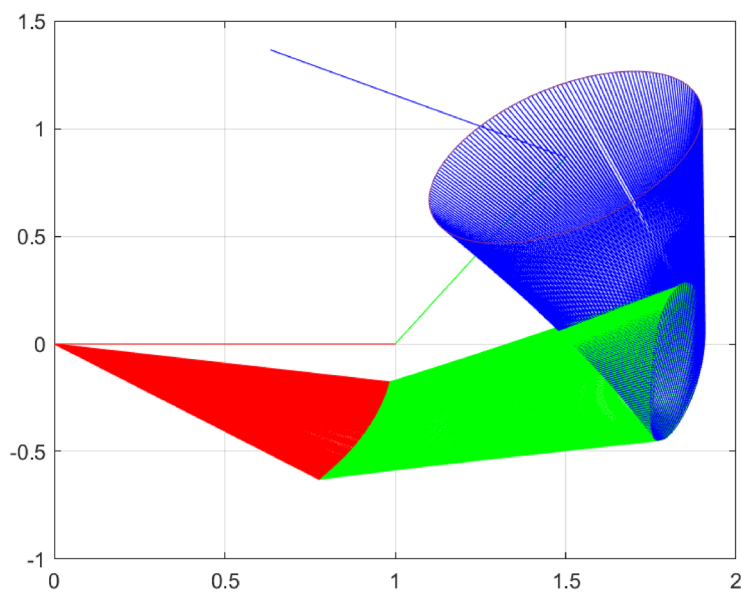
**Fig. 1.** Performance profile based on the number of iterations within a factor  $\tau$  of the best time.



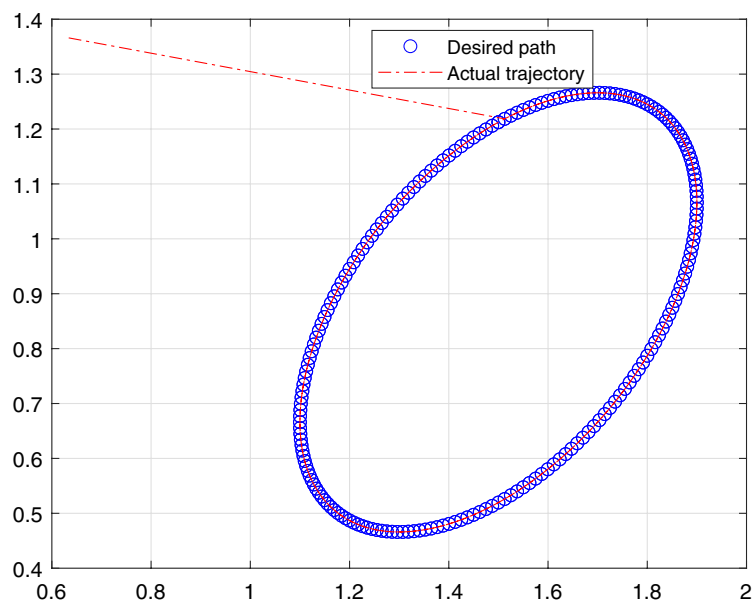
**Fig. 2.** Performance profile based on the function evaluation within a factor  $\tau$  of the best time.



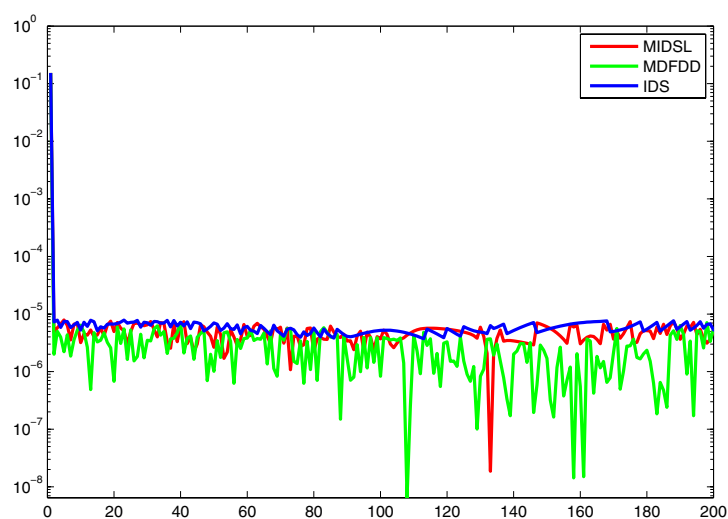
**Fig. 3.** Performance profile with respect to CPU time (in seconds) within a factor  $\tau$  of the best time.



**Fig. 4.** Manipulator trajectories.



**Fig. 5.** End effector trajectory and desired path.



**Fig. 6.** Tracking errors on the horizontal x-axis.

## Conclusion

This study constructed a hybridized step size algorithm for solving systems of nonlinear equations, utilizing the Picard-Mann hybrid iterative scheme as articulated in<sup>23</sup>. The proposed scheme was derived by incorporating a correction parameter that improve convergence. An important feature of the proposed algorithm is its derivative-free scheme, which enhance the efficiency of the algorithm for large-scale problems. To evaluate the efficiency of the new scheme, the study conducted extensive numerical experiment on a set of large-scale test problems. Preliminary findings show that the proposed algorithm outperformed other existing algorithms by consistently requiring fewer iteration count and less CPU time. The study further extended the new scheme to solve motion control model with three degrees of freedom, demonstrating its practical applicability. Findings from the application problem demonstrate the method's effectiveness in real-world situations. Future studies aim to explore the construction of a double-step size algorithm that does not rely on the Lipschitz assumption. This approach if successful will further improve the efficiency and robustness of the method.

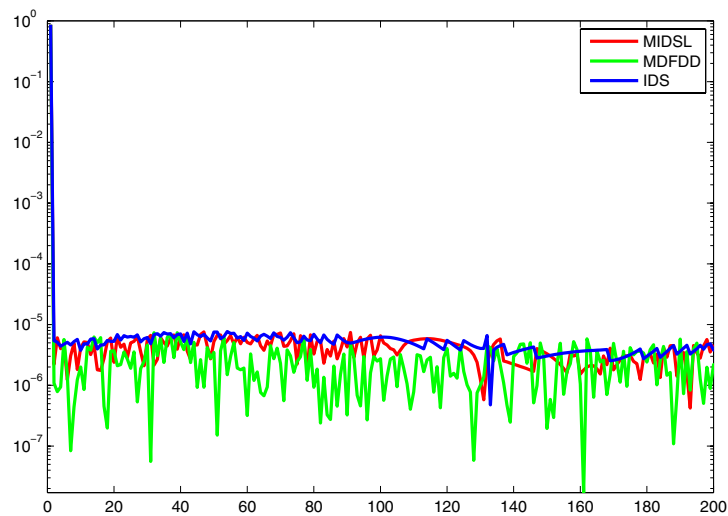


Fig. 7. Tracking errors on the vertical y-axis.

## Data availability

All data generated or analyzed during this study are contained within the manuscript.

Received: 3 August 2025; Accepted: 14 October 2025

Published online: 19 November 2025

## References

- Sinha, V. K. & Maroju, P. New development of variational iteration method using quasilinearization method for solving nonlinear problems. *Mathematics* **11**(4), 935. <https://doi.org/10.3390/math11040935> (2023).
- Sinha, V. K. & Maroju, P. A Numerical Approach to Variational Iteration Method for System of Nonlinear Ordinary Differential Equations. <https://doi.org/10.1063/5.0224795> (2024).
- Sinha, V. K. & Maroju, P. Numerical algorithm for solving real-life application problems of Lane–Emden type equation. *J. Comput. Sci.* **75**. <https://doi.org/10.1016/j.jocs.2023.102185> (2024).
- Sambas, A. et al. A new hyperjerk system with a half line equilibrium: Multistability, period doubling reversals, antimotonicity, electronic circuit, FPGA design, and an application to image encryption. *IEEE Access* **12**, 9177–9194 (2024).
- Sambas, A., Vaidyanathan, S., Mamat, M., Ws, M. S. & Prastio, R. P. Design, analysis of the Genesio–Tesi chaotic system and its electronic experimental implementation. *Int. J. Control Theory Appl.* **9**(1), 141–149 (2016).
- Sambas, A. et al. A 3-D multi-stable system with a peanut-shaped equilibrium curve: Circuit design, FPGA realization, and an application to image encryption. *IEEE Access* **8**, 137116–137132 (2020).
- Halilu, A. S. & Waziri, M. Y. A transformed double step length method for solving large-scale systems of nonlinear equations. *J. Numer. Math. Stoch.* **9**, 20–32 (2017).
- Awwal, A. M. et al. Derivative-free method based on DFP updating formula for solving convex constrained nonlinear monotone equations and application. *AIMS Math.* **6**(8), 8792–8814 (2021).
- Halilu, A. S. & Waziri, M. Y. An improved derivative-free method via double direction approach for solving systems of nonlinear equations. *J. Ramanujan Math. Soc.* **33**, 75–89 (2018).
- Halilu, A. S. & Waziri, M. Y. Enhanced matrix-free method via double step length approach for solving systems of nonlinear equations. *Int. J. Appl. Math. Res.* **6**, 147–156 (2017).
- Abdullahi, H., Halilu, A. S. & Waziri, M. Y. A modified conjugate gradient method via a double direction approach for solving large-scale symmetric nonlinear systems. *J. Num. Math. Stoch.* **10**(1), 32–44 (2018).
- Halilu, A. S. & Waziri, M. Y. Inexact double step length method for solving systems of nonlinear equations. *Stat. Optim. Inf. Comput.* **8**, 165–174 (2020).
- Halilu, A. S., Majumder, A., Waziri, M. Y., Ahmed, K. & Awwal, A. M. Motion control of the two joint planar robotic manipulators through accelerated Dai–Liao method for solving system of nonlinear equations. *Eng. Comput.* <https://doi.org/10.1108/EC-06-2021-0317> (2022).
- Dennis J.E. & Schnabel R.B. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. (Prentice Hall, 1983).
- Waziri, M. Y., Leong, W. J. & Hassan, M. A. Jacobian-free diagonal Newton’s method for solving nonlinear systems with singular Jacobian. *Malays. J. Math. Sci.* **5**, 241–255 (2011).
- Yuan, G. & Lu, X. A new backtracking inexact BFGS method for symmetric nonlinear equations. *Comput. Math. Appl.* **55**, 116–129 (2008).
- Malik, M., Mamat, M., Abas, S. S. & Sulaiman, I. M. A new spectral conjugate gradient method with descent condition and global convergence property for unconstrained optimization. *J. Math. Comput. Sci.* **10**(5), 2053–2069 (2020).
- Mohammed, I. S. et al. A modified nonlinear conjugate gradient method for unconstrained optimization. *Appl. Math. Sci.* **9**(54), 2671–2682 (2015).
- Li, D. & Fukushima, M. A global and superlinear convergent Gauss–Newton based BFGS method for symmetric nonlinear equation. *SIAM J. Numer. Anal.* **37**, 152–172 (1999).
- Petrovic, M. J. & Stanimirovic, P. S. Accelerated double direction method for solving unconstrained optimization problems. *Math. Probl. Eng.* **2014**, 1–8 (2014).
- Petrovic, M. J. An accelerated double step size model in unconstrained optimization. *Appl. Math. Comput.* **250**, 309–319 (2015).
- Petrović, M.J., Stanimirović, P.S., Kontrec, N., & Mladenović, J. Hybrid modification of accelerated double direction method. *Math. Probl. Eng.* **8**, 1523267. <https://doi.org/10.1155/2018/1523267> (2018).
- Safeer, H. K. A Picard–Mann hybrid iterative process. *Fixed Point Theory Appl.* **69**, 10 (2013).

24. Petrović, M. J. Hybridization rule applied on accelerated double step size optimization scheme. *Filomat* **33**(3), 655–665 (2019).
25. Kiri, A. I., Waziri, M. Y. & Halilu, A. S. Modification of the double direction approach for solving systems of nonlinear equations with application to Chandrasekhar's integral equation. *Iran. J. Numer. Anal. Optim.* **12**(2), 426–448 (2022).
26. Halilu, A. S. & Waziri, M. Y. Solving systems of nonlinear equations using improved double direction method. *J. Nigerian Math. Soc.* **32**(2), 287–301 (2020).
27. Halilu, A. S., Majumder, A., Waziri, M. Y. & Abdullahi, H. Double direction and step length method for solving system of nonlinear equations. *Eur. J. Mol. Clin. Med.* **7**(7), 3899–3913 (2020).
28. Dolan, E. & Moré, J. Benchmarking optimization software with performance profiles. *J. Math. Prog.* **91**, 201–213 (2002).
29. Zhang, Y. et al. General four-step discrete-time zeroing and derivative dynamics applied to time-varying nonlinear optimization. *J. Comput. Appl. Math.* **347**, 314–329 (2019).

## Acknowledgements

The Researchers would like to thank the Deanship of Graduate Studies and Scientific Research at Qassim University for financial support (QU-APC-2025).

## Author contributions

Conceptualization; A.S.H. and K.A.; Methodology, A.S.H.; and K.A.; Software, M.A.S.; S.M.I.; and B.A.; Validation, M.A.M, A.M.; and A.Z.A.; Formal analysis, S.M.I.; B.A.; and A.Z.A.; Investigation, A.S.H.; K.A.; and A.M.; Resources, M.A.S.; A.Z.A.; and B.A.; Data curation, A.S.H.; K.A.; and S.M.I.; Writing—original draft preparation, A.S.H.; M.A.M.; and K.A.; Writing—review and editing, A.M.; S.M.I.; M.A.S.; A.Z.A.; Visualization, B.A.; S.M.I.; and A.S.H.; Supervision, A.M; and M.A.M.; Funding acquisition, M.A.S.; A.Z.A.; and B.A. The final version of the manuscript was read by all the authors and agreed to be published in its current form.

## Declarations

### Competing interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to A.Z.A.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025