Soft Computing (2026) 30:4765–4782
https://doi.org/10.1007/s00500-025-10673-1

OPTIMIZATION

A third-order-convergence improved Newton method and its numerical performance

Saheya Barintag[1,2,3] · Juan Hu[1] · Zhanlav Tugal[3,4] · Jein-Shan Chen[5]

Accepted: 19 May 2025 / Published online: 9 July 2026

Abstract
This study focuses on an improved Newton-type iterative scheme for solving systems of nonlinear equations, which is based on a rational approximation with linear numerator and denominator (RALND) model and proposed in (Saheya et al SpringerPlus 5:1–13, 2016). We prove and verify the third-order convergence of this algorithm through theoretical analysis and numerical computations. Numerical experiments and application instances are conducted to demonstrate its effectiveness. The performance profile compares this method with traditional Newton's method and another third-order convergent method, which shows that this method outperforms the traditional Newton's method in terms of the number of iterations, and its computational efficiency is comparable to that of the third-order method. The analysis of the basin of attraction maps compares the INM method with the traditional Newton's method and another third-order convergent method. The results demonstrate that the INM method offers superior characteristics in terms of the contour of the convergence region and the complexity of the boundary compared to the other methods.

Keywords Nonlinear system of equations · Third-order convergence · Improved Newton method

Mathematics Subject Classification 65H05 · 65B99

1 Introduction

In fields of engineering and electrical engineering calculations, the following nonlinear root-finding problem is commonly encountered:

$$F(x) = 0, \tag{1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$. More specifically, it is a nonlinear equations system, which is denoted as

$$F(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix} = 0$$

with $f_i : \mathbb{R}^n \to \mathbb{R}$. There are plenty of iterative schemes for solving systems of nonlinear equations, widely used in science, physics, biology, and engineering, see (Dai and Jazar 2012; Fan 2000; Leung 2013; Marinca et al. 2008) and references therein. In existing methods, improved Newton methods with third-order convergence is widely used in

✉ Jein-Shan Chen
jschen@math.ntnu.edu.tw

Saheya Barintag
saheya@imnu.edu.cn

Juan Hu
juanhu20224012020@163.com

Zhanlav Tugal
tzhanlav@yahoo.com

1 College of Mathematics Science, Inner Mongolia Normal University, 010022 Hohhot, Inner Mongolia, PR China

2 Laboratory of Infinite-dimensional Hamiltonian System and Its Algorithm Application, 010022 Hohhot, Inner Mongolia, PR China

3 Center for Applied Mathematics, 010022 Hohhot, Inner Mongolia, PR China

4 Institute of Mathematics and Digital Technology, Mongolian Academy of Sciences, 13330 Ulaanbaatar, Mongolia

5 Department of Mathematics, National Taiwan Normal University, 116059 Taipei, Taiwan

high-order algorithms due to its fast convergence rate when solving systems of nonlinear equations, and has achieved significant results in various fields. For example, fixed-point algorithms, information maximization problems, Functional Magnetic Resonance Imaging (FMRI) data processing, and task component separation (Weng and Wang 2009). For readers' convenience, we name a few instances as below. First, the algorithm described in Weng et al. (2009) analyzed and applied brain functional magnetic resonance imaging data under resting-state FMRI conditions. In addition, the paper (Zhang 2016) proposes a common fault recovery method for optimizing cloud platforms. The third-order convergent Newton-type algorithm was also employed for power flow calculation. In particular, the results demonstrated that under the same convergence accuracy, the number of iterations is fewer than Newton's method, although the computational complexity is slightly higher (Sun et al. 2009). In 2013, a class of third-order methods was employed for the VSC-HVDC power flow calculation method (Wei 2013). In 2022, Zheng et al. implemented an improved Newton method in the fields of power flow calculation, integrated energy system, and multi-energy power flow calculation for electric and gas energy systems in Zheng et al. 2022. For more relevant literature on applications, please refer to (Jin and Wang 2015; Jin 2019; Liu et al. 2024; Liu et al. 2013; Niazkar 2021; Qureshi 2022).

The third-order improved Newton methods mentioned in the above applications have relatively improved convergence speed and order compared to the traditional Newton method (Kollerstrom 1992). Nonetheless, when the iterative formula includes second-order derivatives, the computational cost and algorithm complexity increases as observed in Halley's method (Gander 1985), hyper-Halley's method, and Chebyshev-Halley's family of iterative methods (Gutierrez et al. 1997; Varona 2002).

In contrast, the iterative algorithm in (Laadhari et al. 2025; Weerakoon et al. 2000) reduces computational complexity by avoiding the need for second-order derivative calculations. Weerakoon and Fernando (2000) approximated the definite integral using the trapezoidal rule instead of the rectangular rule. Inspired by the research of Weerakoon and Fernando, Homeier (2005) proposed a new method using the inverse function, which replaces the arithmetic mean with the harmonic mean under specific parameter conditions. In fact, some other scholars have also made improvements to Newton's method by eliminating the second-order derivatives. For example, Fronting et al. (2003; 2004) considered a method whose iterative formula includes a function value and two Jacobian matrices. Darvishi and Barati (Darvishi and Barati2007)considered a method that uses two function values and one Jacobian matrix for each iteration. Ababneh (2012) improved Newton's method based on the inverse harmonic mean. Saheya (2016) proposed an improved Newton-type method (INM) without second-order derivatives, based on a new rational approximation model (RALND) and a rank-1 correction to the Jacobian matrix.

Only the second-order convergence of INM algorithm is proved in reference Saheya et al. (2016). However, inspired by the modification of the Jacobian matrix of the INM algorithm and the analysis techniques for the convergence order of the iterative method used by Zhanlav et al. (2021; 2013), the INM algorithm is likely to have a higher convergence order or certain improvements in performance and stability. Building upon this conceptual foundation, we employ Zhanlav's analytical framework to rigorously establish the third-order convergence characteristics of the INM algorithm through construction and analysis of key parameters. Additionally, sufficient numerical experiments are performed to validate the algorithm's efficacy and confirm its convergence order. To more comprehensively evaluate the algorithm, introduce more test questions, and continue to increase the number of test questions by disturbing the initial value several times, and then use the performance profile index and it's curve graphs to compare the solving efficiency of related algorithms. Furthermore, by analyzing the Basin of Attraction maps, the scope and boundary features of the attraction domain are explored to preliminarily assess the algorithm's stability.

The main contributions are:

- The third-order local convergence of the INM method is proven through parameter analysis and error estimation.
- Performance Profile curves are plotted based on initial value perturbations, systematically evaluating the algorithm's effectiveness and sensitivity to initial values.
- Through basin of attraction analysis and comparison, the INM method demonstrates better characteristics in terms of the shape of the convergence region and the complexity of the boundary.
- The solution efficiency of the INM method is compared with other methods based on standard test functions and eight practical application problems.

Finally, the structure of this paper is as follows: Sect. 2 introduces the iterative algorithms; Sect. 3 proves the third-order convergence for scalar and nonlinear systems of equations; Sect. 4 presents numerical experiments to confirm the convergence and compares the method with other known methods; and Sect. 5 concludes the paper.

2 An improved Newton-type algorithm

In this section, we roughly review the improved Newton-type algorithm proposed in Saheya et al. (2016), which is the main focus of this paper. The rational approximation model (RALND), $R:\mathbb{R}^n \to \mathbb{R}^n$, was initially introduced in (Yunkang and Saheya 2014) with a linear numerator and denominator, which is defined by

$$R_i(x) = a_0 + \frac{(a_k)^{\mathrm{T}}\left(x - x^k\right)}{1 + b_k^{\mathrm{T}}\left(x - x^k\right)} \tag{2}$$

where $a_k, b_k \in \mathbb{R}^n$ are the undetermined coefficient vectors and $x^k \in \mathbb{R}^n$ represents the current point.
Let

$$\begin{aligned} &R_i\left(x^k\right) = f_i\left(x^k\right), \\ &\nabla R_i\left(x^k\right) = \nabla f_i\left(x^k\right), \\ &\nabla R_i\left(x^{k-1}\right) = \nabla f_i\left(x^{k-1}\right), \end{aligned} \tag{3}$$

and consider the following interpolation conditions:

$$\begin{aligned} m_0 =& \nabla f_i\left(x^{k-1}\right)^{\mathrm{T}}\left(x^{k-1} - x^k\right), \\ m_1 =& \nabla f_i\left(x^k\right)^{\mathrm{T}}\left(x^{k-1} - x^k\right). \end{aligned} \tag{4}$$

Consequently, the RALND function is constructed as

$$\begin{aligned} R_i(x) =& f_i\left(x^k\right) \\ &+ \frac{\nabla f_i\left(x^k\right)^{\mathrm{T}}\left(x - x^k\right)}{1 + \frac{1}{m_0}\left(\sqrt{\frac{m_0}{m_1}}\nabla f_i\left(x^k\right)^{\mathrm{T}} - \nabla f_i\left(x^{k-1}\right)^{\mathrm{T}}\right)\left(x - x^k\right)}, \end{aligned} \tag{5}$$

where $x^k, x^{k-1} \in \mathbb{R}^n$ denote two adjacent iteration points. It is noted that when this function is used, the number of iterations will be reduced. However, because b_k is a variable in each equation, nonlinear equations are quite complex.

From discussions in Saheya et al. (2016), b_k is chosen to reduce complexity while preserving convergence properties. In particular, we observe

$$F(x^k + s) \approx R(x^k + s) = F(x^k) + \frac{J_k s}{1 + b_k^{\mathrm{T}} s} = 0, \tag{6}$$

where $s_k = x^{k+1} - x^k$ and $R(x) = (R_1(x), R_2(x), \ldots, R_n(x))^{\mathrm{T}}$, and where J_k is the Jacobian matrix of $F(x)$ at x^k. When $b_k = 0$, the expression (6) can be simplified as

$$x^{k+1} = x^k - {J_k}^{-1} F(x^k), \tag{7}$$

When $b_k \neq 0$, we achieve a new iterative formula as

$$\left(J_k + F_k b_k^{\mathrm{T}}\right) s_k = -F_k. \tag{8}$$

Where J_k is assumed to be nonsingular, the matrix obtained by applying a rank-one correction to it is also nonsingular. Therefore, the matrix $F_k b_k^{\mathrm{T}}$ is nonsingular. Assuming that matrix $J_k + F_k b_k^{\mathrm{T}}$ is invertible, then we have

$$x^{k+1} = x^k - \left(J_k + F_k b_k^{T}\right)^{-1} F_k. \tag{9}$$

Davidon et al. (1980) proposed a generalization from quadratic approximation to conic approximation. The conic model is described by

$$c_k(x) = f\left(x^k\right) + \frac{\nabla f\left(x^k\right)^{\mathrm{T}} s}{1 + b_k^{\mathrm{T}} s} + \frac{s^{\mathrm{T}} B_k s}{2\left(1 + b_k^{\mathrm{T}} s\right)^2}, \tag{10}$$

where $f(x) \approx c_k(x)$, b_k is a parameter, and B_k is the second order Fréchet derivatives of F at x^k. Note that the first two terms in (10) are (6). Therefore, we can use the parameter b_k of the conic model to define b_k in (9). In other words, $b_k$offers a variety of possibilities, as discussed in (Deng and Li 1995; Gourgeon et al. 1985; Sheng 1995).

Moreover, by using the following conditions to update b_k,

$$R(x^{k-1}) = F(x^{k-1}). \tag{11}$$

Then, combining (6) and (11) yields

$$F_{k-1} = F_k - \frac{J_k s_{k-1}}{1 + b_k^{\mathrm{T}} s_{k-1}}. \tag{12}$$

For notational convenience, we denote

$$\beta_k := 1 - b_k^{\mathrm{T}} s_{k-1} \quad \text{and} \quad y_{k-1} := F_k - F_{k-1}. \tag{13}$$

This together with (12) gives

$$\beta_k y_{k-1} = J_k s_{k-1} \quad \text{and} \quad \beta_k = \frac{y_{k-1}^{\mathrm{T}} J_k s_{k-1}}{y_{k-1}^{\mathrm{T}} y_{k-1}}. \tag{14}$$

Hence, we obtain

$$b_k = \frac{(1 - \beta_k)\, a_k}{a_k^{\mathrm{T}} s_{k-1}}. \tag{15}$$

for any $a_k \in \mathbb{R}^n$ with $a_k^{\mathrm{T}} s_{k-1} \neq 0$. If we plug in $a_k = s_{k-1}$, then

$$b_k^{\mathrm{T}} = \frac{(1-\beta_k)\, s_{k-1}^{\mathrm{T}}}{s_{k-1}^{\mathrm{T}} s_{k-1}} = \frac{y_{k-1}^{\mathrm{T}} \left(y_{k-1} - J_k s_{k-1}\right) s_{k-1}^{\mathrm{T}}}{\left(y_{k-1}^{\mathrm{T}} y_{k-1}\right)\left(s_{k-1}^{\mathrm{T}} s_{k-1}\right)}, \tag{16}$$

Now, combining (14) and (16) leads to

$$F_k b_k^{\mathrm{T}} = \frac{y_{k-1}^{\mathrm{T}} \left(y_{k-1} - J_k s_{k-1}\right)}{y_{k-1}^{\mathrm{T}} y_{k-1}} \frac{F_k s_{k-1}^{\mathrm{T}}}{s_{k-1}^{\mathrm{T}} s_{k-1}}. \tag{17}$$

According to (17), we outline the algorithmic scheme for improved Newton's method (Saheya et al. 2016).

2.1 Algorithm 1: Improved Newton's Method (INM)

Let x_0 be a given initial approximation to the solution of $F(x) = 0$, where J_0 is nonsingular, and choose a tolerance $\varepsilon_0 > 0$. The steps for obtaining x^{k+1}, $k = 0, 1, 2, \ldots$, are as follows:

Step 1: Initialization. Set $b_0 = 0$, $\varepsilon = \varepsilon_0$, and $k = 0$.
Step 2: Stopping Criterion. If $\|F(x^k)\| \leq \varepsilon$, terminate the algorithm and output x^k as the solution.
Step 3: Iteration Update. Compute $F(x^k) b_k^{\mathrm{T}}$ and x^{k+1} as follows:

$$F(x^k) b_k^{\mathrm{T}} = \frac{y_{k-1}^{\mathrm{T}} (y_{k-1} - J_k s_{k-1})}{y_{k-1}^{\mathrm{T}} y_{k-1}} \cdot \frac{F(x^k) s_{k-1}^{\mathrm{T}}}{s_{k-1}^{\mathrm{T}} s_{k-1}},$$
$$x^{k+1} = x^k - \left(J_k + F(x^k) b^{\mathrm{T}}(x^k)\right)^{-1} F(x^k).$$

Step 4: Update Iteration Index. Increment k by 1, i.e., $k = k + 1$, and return to Step 2.

2.2 Notation

1. $J_k = F'(x^k)$: the Jacobian matrix of $F(x)$ at x^k,
2. $y_{k-1} = F(x^k) - F(x^{k-1})$,
3. $s_{k-1} = x^k - x^{k-1}$.

For the special case of scalar setting in the above algorithm, Step 3 is:

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k) + f(x^k) b_k}, \tag{18}$$

where

$$b_k = \frac{f(x^k) - f(x^{k-1}) - f'(x^k)(x^k - x^{k-1})}{(f(x^k) - f(x^{k-1}))(x^k - x^{k-1})}. \tag{19}$$

3 Analysis of Convergence

In light of the improved algorithm, the proof of third-order convergence is divided into two parts: the scalar setting equation and the nonlinear system of equations. Before proceeding with the convergence analysis, we first recall some background concepts.

Definition 1 (Zhanlav et al. 2020) Let $\left\{x^k\right\}_{k\geq 0}$ be a sequence in $\mathbb{R}^n$ which converges to x^*. Then, the convergence is called order $p (p > 1)$ if

$$\frac{\left\|F\left(x^{k+1}\right)\right\|}{\left\|F\left(x^k\right)\right\|^p} = O(1)$$

or

$$F\left(x^{k+1}\right) = O\left(h^p\right), \quad h = F\left(x^k\right), \quad \text{and} \quad h^p = (\overbrace{h, h, \ldots, h}^{p}).$$

Definition 2 (Zhanlav et al. 2021) Let $f(x)$ be a real function with simple root $x^* \in I$ and let $\{x^n\}$ be a sequence of real numbers that converges towards x^*. The order of convergence p is given by

$$\left|x^{n+1} - x^*\right| \leq M \left|x^n - x^*\right|^p, \quad p \in \mathbb{R}^+, \quad 0 < M < \infty. \tag{20}$$

It is easy to show that the condition (20) is equivalent to

$$\left|f\left(x^{n+1}\right)\right| \leq C \left|f\left(x^n\right)\right|^p, \quad \text{for some } 0 < C < \infty,$$

provided that $f'(x) \neq 0$ around the small neighborhood of x^*.

Definition 3 (Argyros et al. 2024; Chang et al. 2025) Let the iterative function $\varphi(x)$ be used to solve the equation

$$x = \varphi(x),$$

and suppose that x^* is a solution of this equation. If there exists a $\delta > 0$ such that the neighborhood of x^*,

$$U(x^*, \delta) = \{x \in X : \|x - x^*\| < \delta\},$$

satisfies the following property: when the initial value $x_0 \in U(x^*, \delta)$, the sequence $\{x_n\}$ generated by the iterative formula

$$x_{n+1} = \varphi(x_n)$$

converges to x^*, i.e.,

$$\lim_{n\to\infty} x_n = x^*,$$

then the method is said to have local convergence.
For subsequent analysis, we need the Kronecker Delta (Kozen 2007). Let $e_1 = \langle 1, 0, 0\rangle$, $e_2 = \langle 0, 1, 0\rangle$, $e_3 = \langle 0, 0, 1\rangle$ it is known that

$$\delta_{ij} = e_i \otimes e_j = \begin{cases} 1, i = j \\ 0, i \neq j \end{cases}, \quad i, j \in \{1, 2, 3\}, \quad \boldsymbol{I} = \delta_{ij} \boldsymbol{e}_i \otimes \boldsymbol{e}_j.$$

Then, we have first order tensor: $\boldsymbol{u} = u_i \boldsymbol{e}_i$; second order tensor: $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_{ij} \boldsymbol{e}_i \otimes \boldsymbol{e}_j$; and third order tensor: $\boldsymbol{\eta} = \eta_{ijk} \boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k$.

Lemma 1 *If* $A = \dfrac{s_{k-1} a_k^{\mathrm{T}}}{a_k^{\mathrm{T}} s_{k-1}}$, *then A is an idempotent matrix.*

Proof According to the contraction and union rules of tensors, we see that

$$\begin{aligned} A^2 &= \left(\frac{s_{k-1} a_k^{\mathrm{T}}}{a_k^{\mathrm{T}} s_{k-1}}\right)^2 = \left(\frac{s_{k-1} \otimes a_k^{\mathrm{T}}}{a_k^{\mathrm{T}} s_{k-1}}\right)^2 \\ &= \left(\frac{s_{k-1} \otimes a_k^{\mathrm{T}}}{a_k^{\mathrm{T}} s_{k-1}}\right)\left(\frac{s_{k-1} \otimes a_k^{\mathrm{T}}}{a_k^{\mathrm{T}} s_{k-1}}\right) = \frac{s_{k-1} \otimes a_k^{\mathrm{T}}}{a_k^{\mathrm{T}} s_{k-1}} \\ &= A. \end{aligned}$$

This completes the proof. □

Lemma 2 *If A is an idempotent matrix, then* $[(A + I)A]^{-1} = A - I$.

Proof The derivation is straightforward by using the properties of idempotent matrices. □

Lemma 3 *Let* $F(x) : D \subset \mathbb{R}^n \to \mathbb{R}^n$ *be a sufficiently Fréchet differentiable function in a convex set* $D \subset \mathbb{R}^n$ *containing a zero* x^*. *Suppose that* $F'(x)$ *is continuous and nonsingular at* x^*. *For an initial approximation sufficiently close to* x^*, *the sequence* $\{x_k\}_{k\geq 0}, x_0 \in D$ *is given by the following iteration process:*

$$x_{k+1} = x_k - \tau_k F'(x_k)^{-1} F(x_k), \tag{21}$$

where τ_k *is some* $n \times n$ *matrix to be determined properly. The iteration scheme (21) has order three if and only if the parameter matrix* τ_k *is given by the following formula:*

$$\tau_k = I + \theta_k + O(h^2), \tag{22}$$

where I is the identity matrix, and

$$\theta_k = \frac{1}{2} F'(x^k)^{-1} F''(x^k) F'(x^k)^{-1} F(x^k), \tag{23}$$

Proof Please refer to (Zhanlav et al. 2020). □

Lemma 4 *Assume*

$$\|P_k\| < 1,$$

where P_k *is a matrix associated with the iteration process, I is the identity matrix, and* $\|\cdot\|$ *denotes the operator norm (also known as the spectral norm or 2-norm). Then, we have*

$$(I - P_k)^{-1} = \sum_{j=0}^{\infty} P_k^j = I + P_k + O\left(P_k^2\right).$$

Proof This is a consequence of (Clason 2020), which is called the Neumann series. □

Lemma 5 *Let* $F : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ *be a function that is sufficiently differentiable on the domain D. The qth derivative of F at a point* $u \in \mathbb{R}^n$, *where* $q \geq 1$, *is defined as the q -linear function*

$$F^{(q)}(u) : \underbrace{\mathbb{R}^n \times \cdots \times \mathbb{R}^n}_{qtimes} \to \mathbb{R}^n,$$

such that $F^{(q)}(u)(v_1, \ldots, v_q) \in \mathbb{R}^n$. *This function maps q vectors* $v_1, \ldots, v_q \in \mathbb{R}^n$ *to a vector in* $\mathbb{R}^n$, *satisfying the linearity property in each of its arguments. It is easy to observe that*

1. $F^{(q)}(u)(v_1, \ldots, v_{q-1}, \cdot) \in \mathcal{L}(\mathbb{R}^n)$,
2. $F^{(q)}(u)\left(v_{\sigma(1)}, \ldots, v_{\sigma(q)}\right) = F^{(q)}(u)(v_1, \ldots, v_q)$ *for all permutations* σ *of* $\{1, 2, \ldots, q\}$.

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

(a) $F^{(q)}(u)(v_1, \ldots, v_q) = F^{(q)}(u) v_1 \ldots v_q$,
(b) $F^{(q)}(u) v^{q-1} F^{(p)} v^p = F^{(q)}(u) F^{(p)}(u) v^{q+p-1}$.

Proof Please refer to (Cordero et al. 2010).

Remark 1 From the 5, we know that the second derivative $F''(x_k)$ is a bilinear function. Therefore, it can be represented using the inner product of vectors. For any vectors u and v, we have

$$F''(x_k)(u,v) = F''(x_k)uv.$$

Keep simplifying the following expression:

$$-\frac{1}{2}\frac{y_{k-1}^{\mathrm{T}}(F''(x_k)F'(x_k)^{-1}F'(x_k)s_{k-1}s_{k-1}+O(s_{k-1}^3))}{y_{k-1}^{\mathrm{T}}y_{k-1}}.$$

Using the fact that $F'(x_k)^{-1}F'(x_k) = I$ leads to

$$F''(x_k)F'(x_k)^{-1}F'(x_k) = F''(x_k)I = F''(x_k).$$

Therefore, the above expression is simplified as

$$-\frac{1}{2}\frac{y_{k-1}^{\mathrm{T}}(F''(x_k)s_{k-1}s_{k-1}+O(s_{k-1}^3))}{y_{k-1}^{\mathrm{T}}y_{k-1}}.$$

This is equivalent to

$$-\frac{1}{2}\frac{y_{k-1}^{\mathrm{T}}(F''(x_k)s_{k-1}^2+O(s_{k-1}^3))}{y_{k-1}^{\mathrm{T}}y_{k-1}},$$

because in this context, $s_{k-1}s_{k-1}$ and s_{k-1}^2 represent the same second derivative acting on the vectors. Finally, we achieve

$$\begin{aligned}&-\frac{1}{2}\frac{y_{k-1}^{\mathrm{T}}(F''(x_k)s_{k-1}^2+O(s_{k-1}^3))}{y_{k-1}^{\mathrm{T}}y_{k-1}}\\ =&-\frac{1}{2}\frac{y_{k-1}^{\mathrm{T}}(F''(x_k)F'(x_k)^{-1}F'(x_k)s_{k-1}s_{k-1}+O(s_{k-1}^3))}{y_{k-1}^{\mathrm{T}}y_{k-1}}.\end{aligned}$$

Remark 2 When $q=2$, $F^{(2)}(u)$ is a bilinear function, meaning for any vectors v_1 and v_2, there holds

$$F^{(2)}(u)(v_1,v_2) = F^{(2)}(u)(v_2,v_1).$$

Thus, for the given expressions, it can be verified that

$$\begin{aligned}&F^{(2)}(x_k)\left(F'(x_k)^{-1}y_{k-1},s_{k-1}\right)\\ =&F^{(2)}(x_k)\left(s_{k-1},F'(x_k)^{-1}y_{k-1}\right).\end{aligned}$$

This says that

$$\begin{aligned}&y_{k-1}^{\mathrm{T}}(F^{(2)}(x_k)F'(x_k)^{-1}y_{k-1}s_{k-1})\\ =&y_{k-1}^{\mathrm{T}}(F^{(2)}(x_k)F'(x_k)^{-1}s_{k-1}y_{k-1}),\end{aligned}$$

which further implies the following expression holds according to condition (a):

$$\begin{aligned}&-\frac{1}{2}\frac{y_{k-1}^{\mathrm{T}}(F''(x_k)F'(x_k)^{-1}y_{k-1}s_{k-1})}{y_{k-1}^{\mathrm{T}}y_{k-1}}+O(h^2)\\ =&-\frac{1}{2}\frac{y_{k-1}^{\mathrm{T}}(F''(x_k)F'(x_k)^{-1}s_{k-1}y_{k-1})}{y_{k-1}^{\mathrm{T}}y_{k-1}}+O(h^2).\end{aligned}$$

3.1 Scalar equation setting

Let the nonlinear function $f:\mathbb{R}\to\mathbb{R}$ be continuously differentiable on $D\subset\mathbb{R}$. As mentioned in Saheya et al. (2016), we approximate the nonlinear function $f(x)$ near x^k as follows:

$$f(x)\approx f\left(x^k\right)+\frac{f'\left(x^k\right)\left(x-x^k\right)}{1+b_k\left(x-x^k\right)}. \tag{24}$$

In order to solve $f(x)=0$, by using (24), we have

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)+f(x^k)b_k}. \tag{25}$$

Then, applying the interpolation method given in Saheya et al. (2016) to determine the parameter b_k, it yields

$$b_k = \frac{f(x^k)-f(x^{k-1})-f'(x^k)(x^k-x^{k-1})}{(f(x^k)-f(x^{k-1}))(x^k-x^{k-1})}. \tag{26}$$

Plugging (26) into (25), we obtain the simplified form of (25),

$$x^{k+1} = x^k - \frac{f(x^k)}{\frac{f'(x^k)f(x^{k-1})}{f(x^{k-1})-f(x^k)}+\frac{f(x^k)}{x^k-x^{k-1}}}. \tag{27}$$

Applying the Taylor expansion of $f(x^{k-1})$ at point x^k. Then, b_k given by (26) can be rewritten as (28). To see this, we first verify

$$
\begin{aligned}
b_k =& \frac{f(x_k) - f(x_{k-1}) - f'(x_k)(x_k - x_{k-1})}{(f(x_k) - f(x_{k-1}))(x_k - x_{k-1})} \\
=& \frac{1}{x_k - x_{k-1}} \\
& - \frac{f'(x_k)}{f(x_k) - (f(x_k) + f'(x_k)(x_{k-1} - x_k) + \frac{1}{2}f''(x_k)(x_{k-1} - x_k)^2 + O(h^3)} \\
=& \frac{1}{(x_k - x_{k-1})} \\
& - \frac{1}{(x_k - x_{k-1})[1 - \frac{f''(x_k)}{2f'(x_k)}(x_k - x_{k-1}) + O(h^2)]} \\
=& \frac{1}{(x_k - x_{k-1})} \\
& - \frac{1}{(x_k - x_{k-1})[1 - [\frac{f''(x_k)}{2f'(x_k)}(x_k - x_{k-1}) + O(h^2)]]} \\
=& \frac{1}{(x_k - x_{k-1})} \\
& - \frac{1}{(x_k - x_{k-1})}[1 + \frac{f''(x_k)}{2f'(x_k)}(x_k - x_{k-1}) + O(h^2)] \\
=& - \frac{f''(x_k)}{2f'(x_k)} + O(h).
\end{aligned}
$$

Assuming the conditions specified in Lemma 4 are satisfied, the fourth equality in the above process uses Lemma 4, resulting in the content of the fifth equality. Hence, (25) can be recast as

$$
x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k) - \frac{f''(x^k)}{2f'(x^k)}f(x^k)} + O(h^3). \tag{28}
$$

Now, for scalar setting, the convergence of Algorithm 1 is stated in Theorem 1 as below.

Theorem 1 *Assume that the function $f : D \subset \mathbb{R} \to \mathbb{R}$ is sufficiently smooth and has a simple zero $x^* \in D$. Suppose that the initial approximation x_0 is sufficiently close to x^* and the parameter b_k satisfies condition (26). Then,the iterative methods in Step 3 have third-order convergence.*

Proof In Zhanlav et al. (2017), it was shown that the two-step iteration method (21) is a third-order method if and only if the iteration parameter τ_k satisfies

$$
\tau_k = 1 + \theta_k + O(\theta_k^2), \quad \theta_k = \frac{f(y_k)}{f(x^k)} = \frac{O(f(x^k)^2)}{f(x^k)} = O(h). \tag{29}
$$

By Taylor's expansion of θ_k at point x^k and using (29) yields

$$
\begin{aligned}
\theta_k =& \frac{f(x^k) + f'(x^k)(y_k - x^k) + \frac{1}{2}f''(x^k)(y_k - x^k)^2 + O(h^3)}{f(x^k)}, \\
=& \frac{1}{2}\frac{f''(x^k)}{f'(x^k)^2}f(x^k) + O(h^2).
\end{aligned}
$$

In view of this, we rewrite (28) as

$$
x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}(1 + \frac{f''(x^k)}{2f'(x^k)^2}f(x^k) + O(h^2)). \tag{30}
$$

Then, combining (21) and (30) leads to

$$
\tau_k = 1 + \frac{1}{2}\frac{f''(x^k)}{f'(x^k)^2}f(x^k) + O(h^2) = 1 + \theta_k + O(h^2),
$$

which indicates that (28) is with the third-order convergence. Therefore, Algorithm 1 exhibits the same convergence rate. □

3.2 The nonlinear system of equations

In the general vector setting, Step 3 can be expressed as

$$x^{k+1} = x^k - \left(I + F'(x^k)^{-1}F(x^k)b_k^T\right)^{-1} F'(x^k)^{-1}F(x^k). \quad (31)$$

According to Lemma 4, we have

$$\tau_k = (I + F'(x^k)^{-1}F(x^k)b_k^T)^{-1} = I - \underbrace{F'(x^k)^{-1}F(x^k)b_k^T} + O(h^2). \quad (32)$$

Then, the iterative formula considered in Step 3 becomes

$$x^{k+1} = x^k - (I - F'(x^k)^{-1}F(x^k)b_k^T)F'(x^k)^{-1}F(x^k). \quad (33)$$

By using Taylor expansion of $F(x^{k-1})$ at point x^k, y_{k-1} can be represented as

$$\begin{aligned} y_{k-1} &= F(x^k) - F(x^{k-1}); \\ &= F(x^k) - (F(x^k) + F'(x^k)(x^{k-1} - x^k) \\ &\quad + \frac{1}{2}F''(x^k)(x^{k-1} - x^k)^2 + O((x^{k-1} - x^k)^3). \end{aligned}$$

$$\begin{aligned} (1-\beta_k) &= 1 - \frac{y_{k-1}^{\mathrm{T}} J_k s_{k-1}}{y_{k-1}^{\mathrm{T}} y_{k-1}} \\ &= \frac{y_{k-1}^{\mathrm{T}}(y_{k-1} - J_k s_{k-1})}{y_{k-1}^{\mathrm{T}} y_{k-1}} \\ &= -\frac{1}{2}\frac{y_{k-1}^{\mathrm{T}}(F''(x_k)s_{k-1}^2 + O(s_{k-1}^3))}{y_{k-1}^{\mathrm{T}} y_{k-1}} \\ &= -\frac{1}{2}\frac{y_{k-1}^{\mathrm{T}}(F''(x_k)F'(x_k)^{-1}F'(x_k)s_{k-1}s_{k-1} + O(s_{k-1}^3))}{y_{k-1}^{\mathrm{T}} y_{k-1}} \\ &= -\frac{1}{2}\frac{y_{k-1}^{\mathrm{T}}(F''(x_k)F'(x_k)^{-1}(y_{k-1} + O(s_{k-1}^2))s_{k-1} + O(s_{k-1}^3))}{y_{k-1}^{\mathrm{T}} y_{k-1}} \\ &= -\frac{1}{2}\frac{y_{k-1}^{\mathrm{T}}(F''(x_k)F'(x_k)^{-1}y_{k-1}s_{k-1})}{y_{k-1}^{\mathrm{T}} y_{k-1}} + O(h^2) \\ &= -\frac{1}{2}\frac{y_{k-1}^{\mathrm{T}}(F''(x_k)F'(x_k)^{-1}s_{k-1}y_{k-1})}{y_{k-1}^{\mathrm{T}} y_{k-1}} + O(h^2). \end{aligned} \quad (37)$$

Thus, we obtain three expressions for y_{k-1} as follows:

$$y_{k-1} = O(s_{k-1}); \quad (34)$$

$$y_{k-1} = F'(x^k)s_{k-1} + O(s_{k-1}^2); \quad (35)$$

$$y_{k-1} = F'(x^k)s_{k-1} - \frac{1}{2}F''(x^k)s_{k-1}^2 + O(s_{k-1}^3). \quad (36)$$

With the above discussions, we present the convergence for the vector setting in Theorem 2.

Theorem 2 *Assume $F(x) : D \subset \mathbb{R}^n \to \mathbb{R}^n$ is a sufficiently Fréchet differentiable function defined on a convex set $D \subset \mathbb{R}^n$ containing a zero x^* of F(x). Suppose $F'(x)$ is continuous and nonsingular at x^*. For an initial approximation $x_0 \in D$ sufficiently close to x^*, the sequence $\{x^k\}_{k\geq 0}$ obtained via (33) exhibits third-order convergence if and only if the parameter matrix $F(x^k)b_k^{\mathrm{T}}$ satisfies $F(x^k)b_k^{\mathrm{T}} = -\frac{1}{2}F''(x^k)F'(x^k)^{-1}F(x^k) + O(h^2)$.*

Proof According to (13), it is known that $1 - \beta_k = b_k^{\mathrm{T}} s_{k-1}$. Considering (12), (14), and (15), we have

$$\beta_k y_{k-1} = J_k s_{k-1}, \quad \beta_k = \frac{y_{k-1}^{\mathrm{T}} J_k s_{k-1}}{y_{k-1}^{\mathrm{T}} y_{k-1}}, \quad b_k = \frac{(1-\beta_k)\, a_k}{a_k^{\mathrm{T}} s_{k-1}},$$

for any $a_k \in \mathbb{R}^n$ with $a_k^{\mathrm{T}} s_{k-1} \neq 0$.

Then, applying Eq. (14), Lemma 5, Remark 1, and Remark 2, $(1 - \beta_k)$ can be simplified directly as

For the calculation of the remainder term, Eq. (34) is utilized. The second equality is derived using Eq. (36), leading to the third equality. Similarly, the fourth equality is obtained by applying Eq. (35), resulting in the fifth equality. From (37), we obtain

$$(1-\beta_k)I = -\frac{1}{2}F''(x^k)F'(x^k)^{-1}s_{k-1} + O(h^2). \quad (38)$$

If we denote

$$c_k^{\mathrm{T}} := -\frac{1}{2}F''(x^k)F'(x^k)^{-1} + O(h), \quad (39)$$

then we have

$$c_k^{\mathrm{T}} s_{k-1} I = -\frac{1}{2} F''(x^k) F'(x^k)^{-1} s_{k-1} + O(h^2), \tag{40}$$

which is equivalent to

$$(1-\beta_k) I = c_k^{\mathrm{T}} s_{k-1} I. \tag{41}$$

Next, using (15) and (41) yields

$$\begin{aligned} I b_k^{\mathrm{T}} &= \frac{(1-\beta_k) I a_k^{\mathrm{T}}}{a_k^{\mathrm{T}} s_{k-1}} \\ &= \frac{c_k^{\mathrm{T}} s_{k-1} I a_k^{\mathrm{T}}}{a_k^{\mathrm{T}} s_{k-1}} \\ &= -\frac{1}{2} \frac{F''(x_k) F'(x_k)^{-1} s_{k-1} a_k^{\mathrm{T}}}{a_{k-1}^{\mathrm{T}} s_{k-1}} + O(h). \end{aligned} \tag{42}$$

For notational convenience, let $B = I b_k^{\mathrm{T}}$ and $C = -\frac{1}{2} F''(x^k) F'(x^k)^{-1} + O(h)$. Equation (42) can be expressed as $B = CA$. According to Lemma 1, we have $BA = CA^2 = CA$ and $BA^2 + BA = CA^2 + CA$. After simplifying $BA^2 + BA = CA^2 + CA$, we conclude that

$$B(A+I)A = C(A+I)A.$$

By applying Lemma 2 to both sides of the equation and simultaneously inverting $(A+I)A$, we obtain

$$B = C \quad \text{and} \quad b_k^{\mathrm{T}} = c_k^{\mathrm{T}}.$$

Then, it follows that

$$F(x^k) b_k^{\mathrm{T}} = F(x^k) c_k^{\mathrm{T}} = -\frac{1}{2} F''(x^k) F'(x^k)^{-1} F(x^k) + O(h^2). \tag{43}$$

Using (43) in (32), we achieve

$$\tau_k = I + \frac{1}{2} F'(x^k)^{-1} F''(x^k) F'(x^k)^{-1} F(x^k) + O(h^2). \tag{44}$$

To sum up, (44) satisfies the condition in Lemma 3 for any a_k; and hence (33) converges in the third-order. Especially, the iterative equation for Step 3 also exhibits third-order convergence. □

4 Numerical experiments

This section illustrates the effectiveness of the algorithm through numerical experiments. We compare the INM method with classical Newton's method (NM) and a third-order Newton-type method (also see THNM (Darvishi and Barati 2007), THNM1 (Huang and Ma 2013), THNM2 (Frontini et al. 2004), NM2 (Rehman et al. 2021)) in terms of iteration times, CPU time, error, and convergence order. The r represents the number of iterations, whereas the CPU represents the CPU time when the algorithm terminates. "Err"denotes the error, while ρ denotes the convergence order. The convergence order is calculated using the following formula (Frontini et al. 2004):

$$\rho = \frac{\log\left(\left|x^{k+1} - a\right| / \left|x^k - a\right|\right)}{\log\left(\left|x^k - a\right| / \left|x^{k-1} - a\right|\right)}.$$

In our study, we utilize 10 test functions and 8 computational problems that have practical applications, which encompass two issues related to the numerical solutions of partial differential equations. All numerical computations in this study were performed on a MacBook Pro (15-inch, 2019) equipped with a 2.4 GHz 8-core Intel Core i9 processor and 32 GB of 2400 MHz DDR4 memory, running macOS. The calculations were conducted using Mathematica 12.3.0.

Function 1 (Darvishi and Barati 2007)

$$\begin{cases} x_2 x_3 + x_4(x_2 + x_3) = 0, \\ x_1 x_3 + x_4(x_1 + x_3) = 0, \\ x_1 x_2 + x_4(x_1 + x_2) = 0, \\ x_1 x_2 + x_1 x_3 + x_2 x_3 - 1 = 0. \end{cases} \quad \text{with initial value } x_0 = (1.25, 1.25, 1.25, 1.25)^{\mathrm{T}}.$$

Function 2 (Huang and Ma 2013)

$$\begin{cases} e^{x_1^2} - e^{\sqrt{2x_1}} = 0, \\ x_1 - x_2 = 0. \end{cases} \quad \text{with initial value } x_0 = (1.25, 1.25)^{\mathrm{T}}.$$

Function 3 (Darvishi and Barati 2007)

$$\begin{cases} x_1 + 2x_2 - 3 = 0, \\ 2x_1^2 + x_2^2 - 5 = 0. \end{cases} \quad \text{with initial value } x_0 = (1.45, 1.45)^{\mathrm{T}}.$$

Function 4 (Huang and Ma 2013)

$$\begin{cases} x_1 + e^{x_2} - \cos x_2 = 0, \\ 3x_1 - x_2 - \sin x_2 = 0. \end{cases} \quad \text{with initial value } x_0 = (7.45, 7.45)^{\mathrm{T}}.$$

Function 5 (Frontini et al. 2004)

$$\begin{cases} e^{x_1} - 2 - x_2 = 0, \\ \cos x_1 + x_2 - 1 = 0. \end{cases} \quad \text{with initial value } x_0 = (2.48, 2.48)^{\mathrm{T}}.$$

Function 6 (George et al. 2023)

$$\begin{cases} 3{x_1}^2 x_2 - x_2 {x_1}^3 = 0, \\ {x_1}^3 - 3x_1 {x_2}^2 - 1 = 0. \end{cases} \quad \text{with initial value } x_0 = (0.77, 0.77)^{\mathrm{T}}.$$

Function 7 (George et al. 2023)

$$\begin{cases} {x_1}^2 + {x_2}^2 - 4 = 0, \\ 3{x_1}^2 - 7{x_2}^2 - 16 = 0. \end{cases} \quad \text{with initial value } x_0 = (-1.27, -1.27)^{\mathrm{T}}.$$

Function 8 (Solaiman et al. 2023)

$$\begin{cases} x_1 + 1 - e^{x_2} = 0, \\ x_1 + \cos x_2 - 2 = 0. \end{cases} \quad \text{with initial value } x_0 = (-0.81, -0.81)^{\mathrm{T}}.$$

Function 9 (Solaiman et al. 2023)

$$\begin{cases} 2 - e^{x_1} + \frac{1}{tan(x_2)} = 0, \\ \frac{1}{tan({x_1}^2 + {x_2}^2 - 5)} = 0. \end{cases} \quad \text{with initial value } x_0 = (1.02, 1.02)^{\mathrm{T}}.$$

Function 10 (Solaiman et al. 2023)

$$\begin{cases} x_2 + x_3 - e^{-x_1} = 0, \\ x_1 + x_3 - e^{-x_2} = 0, \\ x_1 + x_2 - e^{-x_3} = 0. \end{cases} \quad \text{with initial value } x_0 = (0.3, 0.3, 0.3)^{\mathrm{T}}.$$

Problem 1 (Vivas-Cortez et al. 2023) According to the Casson fluid model, fluid flow in blood vessels exhibits velocity gradients near the wall and the center core of the fluids, which travel as plugs with minimal deformation. The following equation expresses the plug flow of Casson fluids:

$$f(x) = 1 - \frac{16}{7}\sqrt{x} + \frac{4}{3}x - \frac{1}{21}x^4 - G,$$

where $G = 0.4$. The initial value is $x_0 = 0.11$. The method of this paper was applied to the Blood rheology and fractional nonlinear equations model.

Problem 2 (Vivas-Cortez et al. 2023) Consider a mechanical issue with a beam position model. The following nonlinear equations can be solved using the method of this a beam:

$$f(x) = x^4 + 4x^3 - 24x^2 + 16x + 16.$$

The initial value is $x_0 = -0.5$.

Table 1 Results of comparisons for different methods

Equation	NM			INM			THNM		
	r	CPU	Err	r	CPU	Err	r	CPU	Err
Functions 1	5	0.000316	2.09718	4	0.000358	2.94413	3	0.000301	3.41898
Functions 2	2	0.000148	2.00487	2	0.000169	3.23596	2	0.000184	3.01076
Functions 3	4	0.000199	1.99931	3	0.000209	3.18023	3	0.000202	3.0235
Functions 4	11	0.000355	1.99700	9	0.000457	2.70227	8	0.000408	2.97041
Functions 5	6	0.000194	1.99611	4	0.000204	3.03711	4	0.000205	2.91146
Functions 6	6	0.000309	2.01236	7	0.000464	2.78161	6	0.000353	2.96215
Functions 7	4	0.000179	1.99539	3	0.000193	3.03757	3	0.000195	2.81369
Functions 8	3	0.000155	1.072761	3	0.000202	2.98178	2	0.000147	3.14228
Functions 9	5	0.000294	2.00091	4	0.000310	2.84096	4	0.000323	2.74676
Functions 10	2	0.000159	2.00416	2	0.000192	2.91813	2	0.000184	2.73641
Problem 1	2	0.000095	1.99748	2	0.000105	3.36238	2	0.000102	3.00363
Problem 2	3	0.000138	1.99982	2	0.000126	3.04643	2	0.000125	3.01452
Problem 3	3	0.000146	1.99829	2	0.000153	3.06262	2	0.000144	2.99027
Problem 4	3	0.000132	1.99980	2	0.000129	3.04001	2	0.000122	2.98175
Problem 5	3	0.000123	1.99939	2	0.000116	3.23710	2	0.000107	2.96864

Table 2 Results of comparisons for different methods in Problem 6 and 7

Equation	NM			INM			THNM1			THNM2		
	r	CPU	Err	r	CPU	Err	r	CPU	Err	r	CPU	Err
Problem 6a	6	21.6672	6.83933×10^{-10}	10	40.6989	9.58538×10^{-11}	6	45.1121	6.83933×10^{-10}	6	43.7441	6.83933×10^{-10}
Problem 6b	3	10.7784	6.68543×10^{-7}	4	15.6651	2.19989×10^{-11}	3	23.8615	6.68643×10^{-7}	3	22.7267	6.68643×10^{-7}
Problem 7a	1	0.00086	1.73901×10^{-1}	1	0.00700	2.54601×10^{-1}	1	0.00029	1.36224×10^{-1}	1	0.00033	1.35287×10^{-1}
Problem 7b	1	0.00024	3.51606×10^{-1}	1	0.00176	3.97339×10^{-1}	1	0.00032	3.26047×10^{-1}	1	0.00026	3.37816×10^{-1}

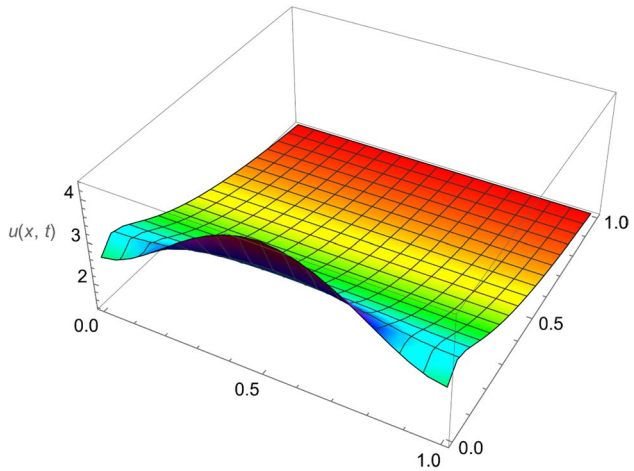

Fig. 1 Approximate solution of Problem 6a

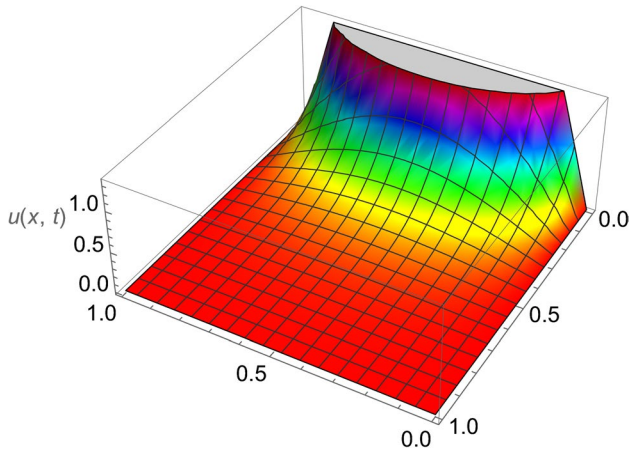

Fig. 2 Approximate solution of Problem 6b

Problem 3 (Solaiman 2021) The Van der Waals equation improves the ideal gas state equation, characterized by considering the size of gas molecules and interactions between molecules. The Van der Waals equation can be rearranged to form the equation:

$$f(V) = PV^3 - n(RT + bP)V^2 + n^2aV - n^3ab.$$

Substituting $P = 40$ atm, $T = 773°\text{C}$, $n = 1.4$, $a = 18$, and $b = 0.1154$, we obtain

$$f(V) = 40V^3 - 95.26535116V^2 + 35.28V - 5.6998368.$$

The initial value is $V_0 = 1.99$.

Problem 4 (Rehman et al. 2021) The fractional conversion rate is the ratio of ammonia generated to the total amount

of nitrogen and hydrogen added to the reaction system. The fractional conversion rate equation is given by:

$$f(A) = -0.186 - \frac{8A^2(A-4)^2}{9(A-2)^3}.$$

This equation is written as:

$$f(A) = A^4 - 7.79075A^3 + 14.744A^2 + 2.511A - 1.674.$$

The initial value is $A_0 = -0.4$.

Problem 5 (Qureshi et al. 2021) In physics, Planck's blackbody radiation law describes the relationship between the emissivity and frequency of electromagnetic radiation emitted from a blackbody at any temperature$T$. The law is expressed as

$$f(x) = e^{-x} + \frac{x}{5} - 1.$$

The initial value is $x_0 = -0.1$.

Problem 6 (Singh et al. 2023) We apply the third-order convergence Newton method to the Fisher, nonlinear partial differential equation. The equation is given by

$$\begin{cases} u_t = u_{xx} + u(1-u), & (x,t) \in [0,1] \times [0,1] \\ u(x,0) = 2\pi \sin(\pi x), \\ u_x(0,t) = 0, \quad u_x(1,t) = 0. \end{cases}$$

The equation is discretized and solved numerically. The initial value is $u^0_{i,j} = 0.2$ for $i = 0, 1, \ldots, N-1$, $j = 1, 2, \ldots, N$ with $N = 21$.

Problem 7 (Capdevila et al. 2023) The heat conduction equation in a nonhomogeneous medium is given by

$$\frac{\partial}{\partial x}\left(k(u)\frac{\partial u}{\partial x}\right) + F(x,t) = c\rho\frac{\partial u}{\partial t},$$

where $F(x,t)$ denotes the heat source density, and k, c, and ρ represent the thermal conductivity, specific heat, and material density, respectively. Assume that c and ρ are constants. Let $c\rho = k_0, \quad \alpha = \frac{k_1}{c\rho}, \quad f = \frac{F}{c\rho}, \quad$ with $\quad k_0 = 1$ and assume that the thermal conductivity depends linearly on temperature, i.e., $k(u) = k_0 + k_1 u$. Then the equation becomes

$$\begin{cases} \alpha u_x^2 + (1+\alpha u)u_{xx} + f = u_t, & (x,t) \in [x_0, x_f] \times [0, T_{\max}], \\ u(-4,t) = 0, \quad u(4,t) = \sin^2(t), \\ u(x,0) = \operatorname{sech}^2(7x). \end{cases}$$

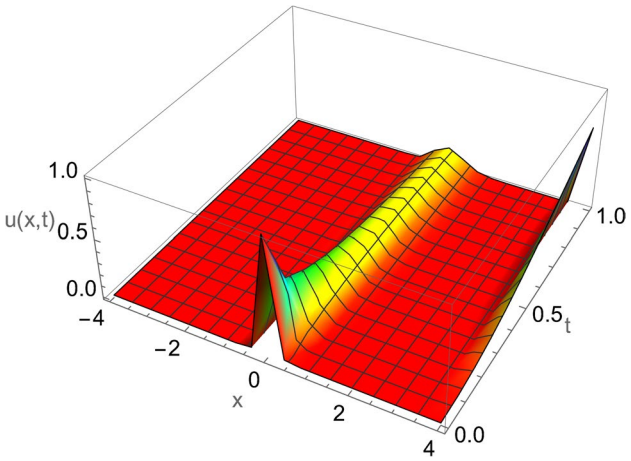

Fig. 3 Approximate solution of Problem 7a

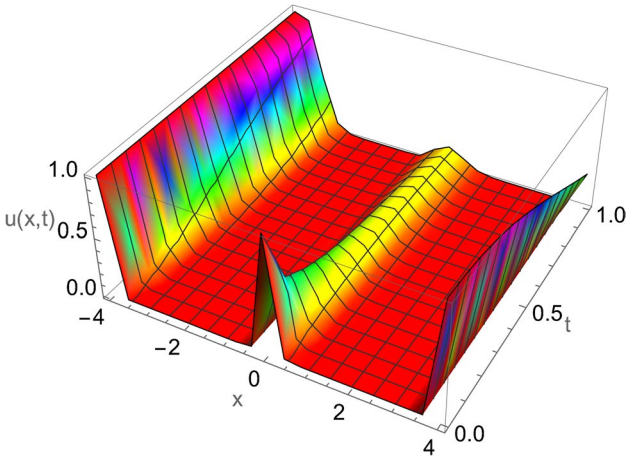

Fig. 4 Approximate solution of Problem 7b

To solve this nonlinear heat equation numerically, we apply an implicit finite difference method. The space-time grid is defined using a spatial step size $h = \frac{x_f - x_0}{nx}$ and time step size $l = \frac{T_{\max}}{nt}$, where *nx* and *nt* are the number of subintervals in the x- and t-directions, respectively. Let $u_{i,k}$ denote the numerical approximation of the solution u at (x_i, t_k). The derivatives are approximated by

$$u_t \approx \frac{u(x,t) - u(x,t-l)}{l}, \quad u_x \approx \frac{u(x+h,t) - u(x,t)}{h},$$
$$u_{xx} \approx \frac{u(x+h,t) - 2u(x,t) + u(x-h,t)}{h^2}.$$

In the numerical experiment, we choose the parameters as $\alpha = 1, T_{\max} = 1, nx = 20, nt = 10, \text{and} f = 0$. The resulting finite difference equation becomes

$$lu_{i-1,k} - \left(h^2 + l\left(2 + \alpha\left(u_{i+1,k} - u_{i-1,k}\right)\right)\right)u_{i,k} - l\alpha u_{i,k}^2 + l\left(u_{i+1,k} + \alpha u_{i+1,k}^2\right) = -h^2 u_{i,k-1}.$$

Table 3 Results of comparisons for different methods in Problem 4 and 8

Equation	NM			INM			NM2			THNM		
	r	CPU	Err	r	CPU	Err	r	CPU	Err	r	CPU	Err
Problem 4	3	0.000132	4.80932×10^{-12}	2	0.000129	1.36386×10^{-7}	3	0.000371	2.09954×10^{-10}	2	0.000122	2.07590×10^{-12}
Problem 8	5	0.001493	1.52724×10^{-8}	7	0.001307	1.77764×10^{-15}	3	0.000256	8.37146×10^{-10}	3	0.000266	1.77636×10^{-15}

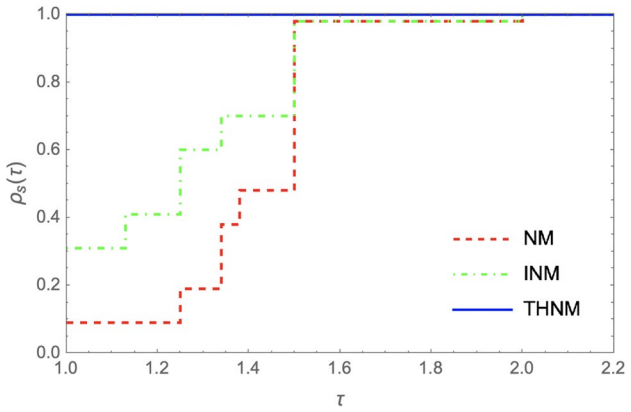

Fig. 5 Performance profile of iteration numbers of NM, INM, THNM

Problem 8 (Rehman et al. 2021) The water flow passing through a region during a selected period of time is influenced by factors such as the slope of the channel. When the channel becomes inclined, Manning's equation applies to the water flow in an open channel under uniform flow conditions:

$$Q = \frac{\sqrt{m}}{n} A R^{2/3},$$

where m is the slope of the channel, A is the cross-sectional area of the channel, R is the hydraulic radius of the channel, and n is the Manning's roughness coefficient. For a rectangular channel with width W and water depth h, it is known that $A = Wh$, and$R = \frac{Wh}{W+2h}$. Thus, the equation becomes:

$$Q = \frac{\sqrt{m}}{n} Wh \left[\frac{Wh}{W+2h}\right]^{2/3}$$

Given the flow rate, it is necessary to determine the depth of water h in the channel. The equation can be rewritten as:

$$f(h) = \frac{\sqrt{m}}{n} Wh \left[\frac{Wh}{W+2h}\right]^{2/3} - Q$$

where $Q = 14.15\ \mathrm{m}^3/\mathrm{s}$, $W = 4.572\ \mathrm{m}$, $n = 0.017$, and $m = 0.0015$. The initial guess is taken to be $h_0 = 8.5\mathrm{m}$.
The numerical experimental results for Function 1 to Function 10 and Problem 1 to Problem 5 are presented in Table 1. The numerical calculation results for "problem 6a"are shown in Table 2, and the approximate solution image for Problem 6 under the initial boundary conditions is depicted in Fig. 1. When some boundary conditions are changed to $u(0,t) = 0$, $u(1,t) = 0$, the image of the approximate solution is shown in Fig. 2, and the numerical calculation results are shown in "problem 6b"in Table 2. In Problem 6, experiments were conducted to analyze the impact of different

boundary and initial conditions. Boundary conditions: The Neumann boundary condition $(u_x(0,t)=0, u_x(1,t)=0)$ represents zero derivative at the boundary, which is suitable for closed systems or symmetric problems with no flux across the boundary. It allows the boundary values to adjust freely based on internal calculations. In contrast, the Dirichlet boundary condition $(u(0,t)=0, u(1,t)=0)$ fixes the boundary values, commonly used in physical systems with known boundary states, such as constant temperature or concentration scenarios. Switching from Neumann to Dirichlet conditions is generally more conducive to maintaining stability and ensuring convergence, as Dirichlet conditions reduce degrees of freedom by fixing boundary values, whereas Neumann conditions rely heavily on derivative accuracy and are more sensitive to numerical errors. Initial conditions: The initial condition determines the system's starting state, directly affecting error propagation and the dynamic behavior of the solution. Unreasonable initial conditions can make the algorithm difficult to converge or result in unstable solutions. To minimize error propagation and improve convergence speed, the initial condition must be compatible with the discretization method. For example, the initial condition $u(x,0)=2\pi\sin(\pi x)$ used in the code is a smooth function, well-suited for discretization with finite difference methods, which significantly enhances convergence and numerical efficiency.

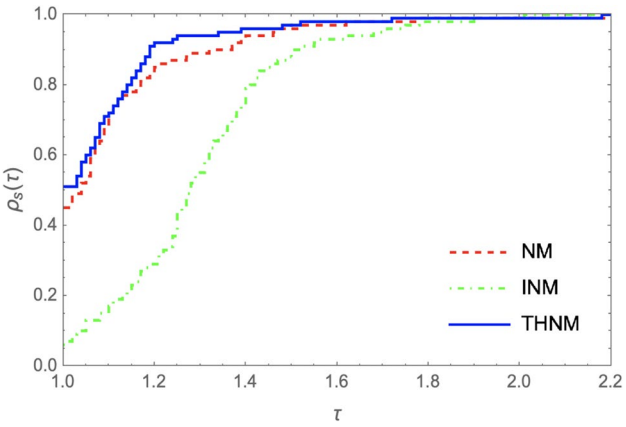

Fig. 6 Performance profile of CPU time of NM, INM, THNM

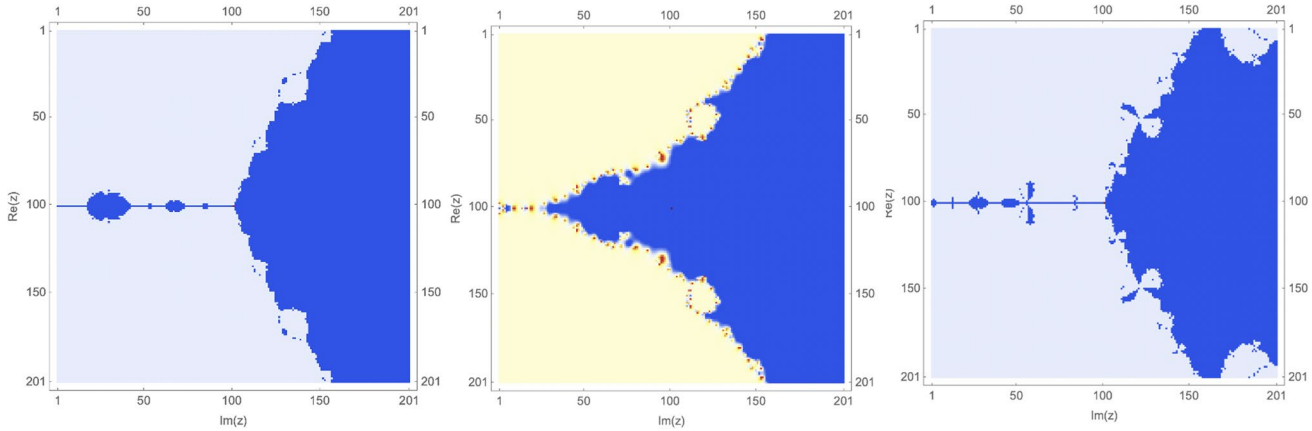

Fig. 7 Basin of attraction map of $f(x)=x^2-\frac{5}{x}+2$ for three iterative methods

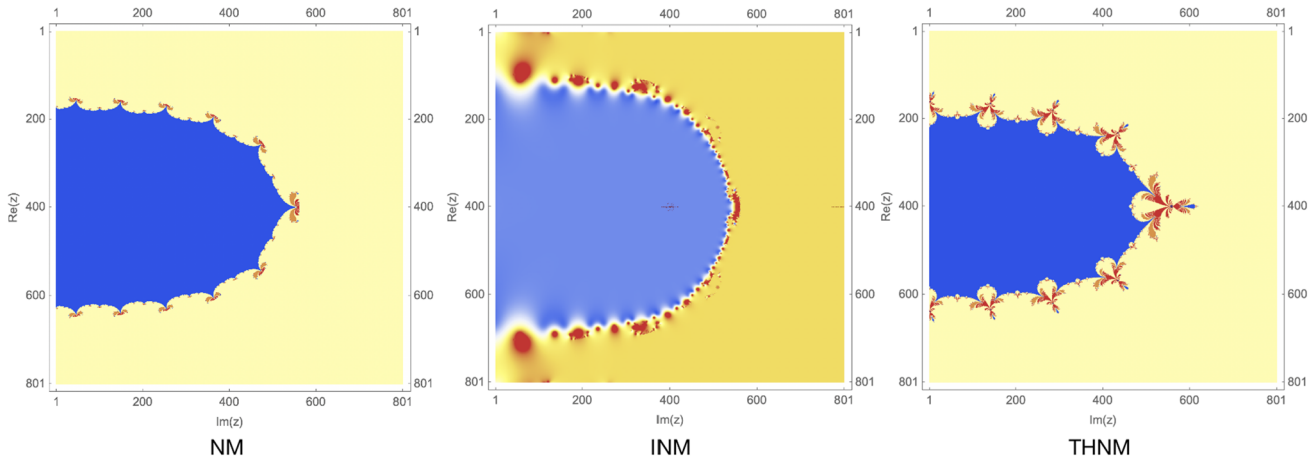

Fig. 8 Basin of attraction map of $f(x)=e^{-x}+\frac{x}{5}-1$ for three iterative methods

The numerical calculation results for "problem 7a"are presented in Table 2, and the approximate solution image for Problem 7 under the initial boundary conditions is shown in Fig. 3. When some boundary conditions are changed to $u(x_0, t) = 1$, $u(x_f, t) = \cos^2(t)$, the image of the approximate solution is shown in Fig. 4, and the numerical calculation results are shown in "problem 7b"in Table 2. Here, the average levels of iteration counts, CPU time, and errors are calculated. The experimental analysis of Problem 7 is similar to that of Problem 6, mainly comparing different boundary conditions and exploring their effects on the convergence and numerical stability of the algorithm.

Based on the numerical results in Tables 1 and 2, the following findings are summarized:

(1) Firstly, the numerical results in both tables confirm the effectiveness of the INM algorithm. Secondly, the high consistency between the numerical results in Table 1 and the theoretical results proves the third-order convergence of the INM algorithm.
(2) According to Table 1, although the number of iterations required by the INM algorithm is less than or equal to that of the NM algorithm and greater than or equal to that of the THNM algorithm, the differences are not significant. Furthermore, due to the involvement of more complex expressions of b_k in the INM algorithm, the CPU time may be relatively long. However, according to Table 1, the algorithm performs well in terms of CPU time, and the CPU time of the INM method is close to that of the NM and THNM methods. The error performance of the INM algorithm falls between the NM and THNM algorithms.
(3) As the complexity of the Problem 6 case increases, the required CPU time and iteration counts also increase. In Table 2 of Problem 6, the CPU time of the INM algorithm is slightly more than that of the NM algorithm, but less than that of the THNM1 and THNM2 algorithms. The iteration count of the INM algorithm is higher than the other algorithms, but its error performance is better. However, when the complexity of the case decreases, as in Problem 7, the CPU time required by the INM algorithm is relatively longer, with error and iteration count similar to other algorithms.

Based on the data in Table 3 and a comparison with the third-order methods mentioned in the literature (Rehman et al. 2021), the following results were obtained: For Problem 4, although INM performs better in terms of iteration count and CPU time, NM has the smallest error and higher efficiency, thus performing the best for this problem. For Problem 8, despite INM having the most iterations, it has the smallest error and the highest accuracy, making it the best performer for this problem. Therefore, overall, INM performs better in terms of accuracy. In order to further comprehensively evaluate the NM, INM, and THNM methods, we adopt the "performance profile"and select the test set from Table 1. Note that the initial point will affect the performance of the iterative formula, mainly because the iterative formula is usually a process of gradually approaching the target value, and the selection of the initial point will determine the starting point and direction of this approximation process. In other words, a suitable initial point can accelerate the convergence of the iteration process to the target value and improve computational efficiency; inappropriate initial points may cause the iteration process to fall into local optima or fail to converge, thereby affecting performance. Therefore, in the numerical experiments, we choose an initial value x_0 that is sufficiently close to x^*. Accordingly, we then randomly generate ten initial values $x_0 + RandomReal[0, 1]$ around the initial point x_0 to compare the three methods with 100 test problems, resulting in s_3 solvers and t_{100} test problems. The performance ratio is defined as

$$R_{p,s} = \frac{F_{p,s}}{\min\{F_{p,s} : s \in S\}},$$

where S denotes the set of solvers, and P represents the set of test problems. The term $F_{p,s}$ refers to the number of iterations or the CPU time used by solver $s \in S$ to solve problem $p \in P$. Assuming for any p, s, the parameter $R_M \geq R_{p,s}$, $R_M = R_{p,s}$ is equivalent to the solver S being able to solve problem P.

To understand an overall assessment for each solver, we define

$$\rho_s(\tau) := \frac{1}{t_p} \text{size}\{p \in P; R_{p,s} \leq \tau\}.$$

This definition is referred to as the iteration performance profile of solver S. Here, $\rho_s(\tau)$ denotes the probability that the performance ratio $R_{p,s}$ of solvers $s \in S$ does not exceed a threshold $\tau \in R$ reflecting the extent to which the algorithm achieves optimal or near-optimal performance.

Figure 5 shows the performance profile of the iteration times of three algorithms for 100 test problems within the range of $\tau \in [1, 2.2]$. Furthermore, Fig. 5 indicates that the numerical performance of the INM solver falls between that of NM and THNM. Figure 6 shows the performance profile of the CPU time of three algorithms for 100 test problems within the range of $\tau \in [1, 2.2]$. Furthermore, Fig. 6 demonstrates that the INM solver exhibits inferior numerical performance in terms of time compared to both NM and THNM.

Performance Profile is used to compare multiple algorithms in terms of efficiency and robustness over a set of test problems, typically based on normalized runtime or error. It intuitively presents the overall performance of different methods, avoiding the limitations of a single metric and providing a more comprehensive algorithm evaluation. In addition to the traditional performance metrics, this study introduces the basin of attraction map as a new analytical tool. This map not only provides a visual representation of the convergence behavior of iterative sequences under different initial conditions, but also effectively addresses complex nonlinear problems. The basin of attraction map allows observation of the convergence regions between different roots and their characteristics, while also helping to identify the variations in convergence speed across different regions. The basin of attraction maps in this study are shown in Figs. 7 and 8.

Different color regions correspond to iterative sequences converging to different roots, forming different "basins". The boundaries between different colors usually represent transition regions between adjacent roots, and initial points on these boundaries may converge to different roots. Complex boundaries reflect the non-uniformity of convergence speed in various regions, which is reflected in some regions converging in fewer iterations while others may require more iterations. In the basin of attraction map, some regions have gradually deepening colors, indicating that the algorithm converges to the root faster within that region. When drawing the basin of attraction map, the size of the contour of the convergent region has different meanings. A larger contour of the convergent region indicates that initial values within that region are more likely to converge to the corresponding root, while a smaller contour indicates that initial values within that region may diverge during iteration or converge to other roots.

In particular, the functions $f(x) = x^2 - \frac{5}{x} + 2$ and $f(x) = e^{-x} + \frac{x}{5} - 1$ are compared using the basin of attraction map for the NM, INM, and THNM methods. Two square regions in the complex plane are considered:

$$D = [-1, 1] \times [-1, 1] \in C \quad \text{and} \quad D = [-4, 4] \times [-4, 4] \in C.$$

Each contains 101×101 and 401×401 grid points, respectively. A color is assigned to each point $z \in D$ for each iterative method. In the corresponding maps for each method, the color represents the magnitude of the error in the iterative method's convergence to the root. Cool colors indicate smaller errors and faster convergence, while warm colors indicate larger errors and slower convergence. Figures 7 and 8 show that the basin of attraction has only one region, indicating that each plot's attractor corresponds to a single unique root. The convergence region of the INM method is larger than that of the NM and THNM methods. A larger basin of attraction implies stronger robustness of the algorithm, while a smaller basin suggests higher sensitivity to the choice of initial guess and weaker robustness. This means that the INM method is more likely to converge to the correct root from a wider range of initial guesses. Additionally, if the boundary does not exhibit complex fractal structures, it indicates that the algorithm is more robust with respect to the initial guess. Repellers, located near the far region of infinity, may appear as a color gradient indicating divergence. This suggests that initial values starting from these regions quickly move away from the convergence behavior and tend to infinity. Figures 7 and 8 show that the boundaries of the basin of attraction present a gradient from red to orange to yellow, suggesting that points at infinity may be repellers. The INM algorithm shows a gradient region in warm colors at the boundary, with some noisy points, which typically indicates higher complexity in the boundary region. However, the overall distinction between the attraction and repulsion parts is clear, suggesting good stability. In summary, the INM method outperforms the NM and THNM methods in terms of computational efficiency.

5 Conclusions

An improved Newton iterative method based on a rational approximation with linear numerator and denominator (RALND) model was presented in Saheya et al. (2016). This paper proves and verifies the third-order convergence of this algorithm through theoretical analysis and numerical computations. Numerical experiments are conducted on ten root-finding examples and eight application instances to demonstrate its effectiveness. The performance profile compares this method with traditional Newton's method and another method with third-order convergence. The experimental results show that this method outperforms traditional Newton's method in terms of the number of iterations, and its computational efficiency is comparable to that of the third-order method. Additionally, the paper offers the basin of attraction maps for the three iterative methods, and the experimental results indicate that the INM method has better characteristics in terms of the contour of the convergence region and the boundary complexity than the other methods. Future research directions may include the stability analysis of the INM algorithm and the design of higher-order algorithms. On the one hand, affine transformations and Möbius transformations can be used to map the iterative formula (as a real function) onto the complex sphere. Such transformations simplify complex problems into more analyzable forms, providing a clearer understanding of the basins of attraction, fixed points, and their stability characteristics. On

the other hand, within the framework of the INM algorithm, higher-order INM algorithms can be designed by analyzing the errors at each step and optimizing the selection of parameter matrices. This approach aims to improve the order of convergence, achieve better convergence performance, and expand the application scope of the algorithm.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00500-025-10673-1.

Acknowledgements The first three authors' work was partially supported by theNational Natural Science Foundation of China (62161044, 11962025), the Natural Science Foundation of Inner Mongolia (2023MS06003, 2022ZD05), Key Laboratory of Infinite-dimensional Hamiltonian System and Its Algorithm Application(IMNU), Ministry of Education (2023KFYB06). The fourth author's work is partially supported by National Scienceand Technology Council, Taiwan.

Author Contributions First draft preparation was done by Saheya and Juan Hu. Editing and reviewing were carried out by Zhanlav and Jein-Shan Chen. Implementation was executed by Saheya and Juan Hu.

Funding Open access funding provided by National Taiwan Normal University

Data availability Please direct to the authors regarding data availability.

Declarations

Conflict of interest Authors do not have any conflicts of interest.

References

Argyros IK, George S, Shakhno S, Regmi S, Havdiak M, Argyros M (2024) Asymptotically Newton-type Methods without Inverses for Solving Equations. Mathematics 12:1069

Ababneh OY (2012) New Newton's method with third-order convergence for solving nonlinear equations. World Acad Sci Eng Technol 61(198):1071–1073

Chang C-W, Qureshi S, Argyros IK, Chicharro FI, Soomro A (2025) A modified two-step optimal iterative method for solving nonlinear models in one and higher dimensions. Math Comput Simul 229:448–467

Capdevila RR, Cordero A, Torregrosa JR (2023) Convergence and dynamical study of a new sixth-order convergence iterative scheme for solving nonlinear systems. AIMS Math 8(6):12751–12777

Clason C (2020) Introduction to Functional Analysis. Compact Textbooks in Mathematics, Springer Nature, Switzerland

Cordero A, Hueso JL, Torregros JR (2010) A modified Newton-Jarratt's composition. Numer Algorithm 55:87–99

Darvishi MT, Barati A (2007) A third-order Newton-type method to solve systems of nonlinear equations. Appl Math Comput 187(2):630–635

Dai L, Jazar RN (2012) Nonlinear approaches in engineering applications. Springer, New York

Deng NY, Li ZF (1995) Some global convergence properties of a conic-variable metric algorithm for minimization with inexact line searches. Optim Methods Softw 5(1):105–122

Davidon WC (1980) Conic approximations and collinear scalings for optimizers. SIAM J Numer Anal 17(2):268–281

Fan E (2000) Extended tanh-function method and its applications to nonlinear equations. Phys Lett A 277(4–5):212–218

Frontini M, Sormani E (2003) Some variant of Newton's method with third-order convergence. Appl Math Comput 140(2–3):419–426

Frontini M, Sormani E (2004) Third-order methods from quadrature formulae for solving systems of nonlinear equations. Appl Math Comput 149(3):771–782

Gander W (1985) On Halley's iteration method. American Math Mon 92(2):131–134

George S, Kunnarath A, Sadananda R, Padikkal J, Argyros IK (2023) Order of convergence, extensions of Newton-Simpson method for solving nonlinear equations and their dynamics. Fract Fract 7(2):163

Gourgeon H, Nocedal J (1985) A conic algorithm for optimization. SIAM J Sci Statistical Comput 6(2):253–267

Gutierrez JM, Hernández MA (1997) A family of Chebyshev-Halley type methods in Banach spaces. Bull Australian Math Soc 55:113–130

Homeier HH (2005) On Newton-type methods with cubic convergence. J Comput Appl Math 176:425–432

Huang N, Ma C-F (2013) A New Third-order Method for Solving Systems of Nonlinear Equations. J Fujian Norm Uni: Nat Sci Edit 1:25–30

Jin H-Y, Wang Z-A (2015) Asymptotic dynamics of the one-dimensional attraction-repulsion Keller-Segel model. Math Methods Appl Sci 38(3):444–457

Jin H-Y, Wang Z-A (2019) Global stabilization of the full attraction-repulsion Keller-Segel system, arXiv preprint arXiv:1905.05990

Kollerstrom N (1992) Thomas Simpson and 'Newton's method of approximation': an enduring myth. British J Hist Sci 25(3):347–354

Kozen D, Timme M (2007) Indefinite summation and the Kronecker delta

Liu C-S, Kuo C-L, Chang C-W (2024) Decomposition–Linearization–Sequential Homotopy Methods for Nonlinear Differential/Integral Equations. Mathematics 12(22):3557

Leung AW (2013) Systems of nonlinear partial differential equations: applications to biology and engineering, vol. 49, Springer Science & Business Media

Laadhari A, Temimi H (2025) Efficient finite element strategy using enhanced high-order and second-derivative-free variants of Newton's method. Appl Math Comput 486

Liu P, Shi J-P, Wang Z-A (2013) Pattern formation of the attraction-repulsion Keller-Segel system. Discret Contin Dyn Syst Ser B 18(10):2597–2625

Marinca V, HeriŞanu N (2008) Application of optimal homotopy asymptotic method for solving nonlinear equations arising in heat transfer. Intern Commun Heat Mass Transf 35(6):710–715
Niazkar M, Yılmaz Türkkan G (2021) Application of third-order schemes to improve the convergence of the Hardy Cross method in pipe network analysis, Adv Math Phys, vol. 2021, pp. 1–12
Qureshi S, Ramos H, Soomro AK (2021) A new nonlinear ninth-order root-finding method with error analysis and basins of attraction. Mathematics 9(16):1996
Qureshi S, Soomro A, Shaikh AA, Hincal E, Gokbulut N (2022) A Novel Multistep Iterative Technique for Models in Medical Sciences with Complex Dynamics, Comput Math Methods Med, vol. July 28
Rehman MA, Naseem A, Abdeljawad T (2021) Some novel sixth-order iteration schemes for computing zeros of nonlinear scalar equations and their applications in engineering. J Funct Spaces 2021:1–11
Saheya B, Chen G-Q, Sui Y-K et al (2016) A new Newton-like method for solving nonlinear equations. SpringerPlus 5:1–13
Sheng S (1995) Interpolation by conic model for unconstrained optimization. Computing 54(1):83–98
Singh H, Sharma JR, Umar SK (2023) A simple yet efficient two-step fifth-order weighted-Newton method for nonlinear models. Numer Algorithms 93(1):203–225
Solaiman OS, Hashim I (2021) Optimal eighth-order solver for nonlinear equations with applications in chemical engineering, Intell Autom Soft Comput, vol. 27, no. 2
Solaiman OS, Sihwail R, Shehadeh H, Hashim I, Alieyan K (2023) Hybrid Newton-Sperm swarm optimization algorithm for nonlinear systems. Mathematics 11(6):1473
Sun Z-Y, Sun Y, Ning W (2009) Power flow algorithm with cubic convergence. Power Syst Prot Control 37(4):5–8
Varona JL (2002) Graphic and numerical comparison between iterative methods. Math Intell 24(1):37–47
Vivas-Cortez M, Ali NZ, Khan AG, Awan MU (2023) Numerical Analysis of New Hybrid Algorithms for Solving Nonlinear Equations. Axioms 12(7):684
Weerakoon S, Fernando T (2000) A variant of Newton's method with accelerated third-order convergence. Appl Math Lett 13:87–93
Wei Y-F, He Q, Sun Y-H, et al. (2013) Improved power flow algorithm for VSC-HVDC system based on high-order Newton-type method, Math Probl Eng, vol. 2013
Weng X-G, Wang H-N (2009) Improved ICA Algorithm and Its Application to fMRI Signals. J South China Univ Technol (Nat Sci Edit) 37(5):27–30
Weng X-G, Wang H-N, Zhang Z-Q et al (2009) Study on Resting Visual Network of Refractive Amblyopia Based on FMRI. Acta Biophys Sinica 25(6):447–452
Yunkang S, Saheya CG (2014) An improvement for the rational approximation RALND at accumulated two-point information. Math Numer Sinica 36(1):51
Zhang Y-D (2016) Research on the key technologies of fault detection for cloud computing system, Master's thesis, Beijing Jiaotong University
Zhanlav T, Chun C, Otgondorj K, Ulziibayar V (2020) High-order iterations for systems of nonlinear equations. Intern J Comput Math 97(8):1704–1724
Zhanlav T, Khongorzul D (2013) Semilocal convergence with R-order three theorems for the Chebyshev method and its modifications, Optim Simul Control, pp. 331–345
Zhanlav T, Otgondorj K (2021) On the Optimal Choice of Parameters in two-point iterative methods for solving nonlinear equations. Comput Math Math Phys 61:29–42
Zhanlav T, Ulziibayar V, Chuluunbaatar O (2017) Necessary and sufficient conditions for the convergence of two-and three-point Newton-type iterations. Comput Math Math Phys 57:1090–1100
Zheng JH, Wu CQ, Xiahou KS et al (2022) A variant of Newton-Raphson method with third-order convergence for energy flow calculation of the integrated electric power and natural gas system. IET Gener Transm Distrib 16(14):2766–2776