



Research article**A single parameter fourth-order Jarrat-type iterative method for solving nonlinear systems****Jia Yu and Xiaofeng Wang***

School of Mathematical Sciences, Bohai University, Jinzhou 121000, Liaoning, China

* **Correspondence:** Email: xiaofengwang@bhu.edu.cn.

Abstract: In this paper, a Jarrat-type iterative method for solving nonlinear systems is proposed through the weight function method. It is proved that the convergence order of the iterative method reaches fourth order in Banach space. The stability of this fourth-order iterative method was analyzed using real dynamics theory. The number of fixed points and critical points was solved using real dynamics tools, and the dynamic plane was drawn. Through the dynamic plane, it can be observed that the stability of the iterative method is best when the parameters are $0 \leq \beta < \frac{423}{8}$. Next, we compare the efficiency of the proposed method with two fourth-order iterative methods. The results show that when $\beta = \frac{9}{8}$, our method shows better efficiency in terms of calculation time, calculation error, and calculation efficiency. In addition, we also use this method to successfully solve the Hammerstein-type integral equation, boundary value problem, heat conduction problem in partial differential equation, and other nonlinear equation systems. The experimental results are consistent with the theoretical analysis, which further confirms the accuracy of the method.

Keywords: Jarrat-type iterative method; nonlinear systems; stability analysis**Mathematics Subject Classification:** 65B99, 65H05

1. Introduction

In the field of engineering and scientific computing, solving systems has become a core task. Iterative algorithms have become the preferred method for solving such problems due to their efficiency and flexibility in estimating solutions to nonlinear systems. The iterative method solves the nonlinear problem by gradually approaching the estimated value of the iterative adjustment solution until it converges to the exact solution or meets the predetermined error tolerance.

The Jarrat-type iterative method has been optimized and improved based on the Newton iteration method, and is an effective method for solving nonlinear systems [1]. The classic Jarrat-type iterative

method takes the form of (1.1):

$$\begin{cases} \tilde{h}^{(k)} = v^{(k)} - \frac{2}{3} [\Gamma'(v^{(k)})]^{-1} \Gamma(v^{(k)}), \\ v^{(k+1)} = S [\Gamma'(v^{(k)})]^{-1} \Gamma(v^{(k)}), \end{cases} \quad (1.1)$$

where Γ is a function defined on R^n , $v \in R^n$ and $S = [6\Gamma'(\tilde{h}^{(k)}) - 2\Gamma'(v^{(k)})]^{-1} [3\Gamma'(\tilde{h}^{(k)}) + \Gamma'(v^{(k)})]$, comparing the convergence order of Newton's iteration method and the classical Jarrat-type iteration method, it is evident that the Jarrat-type iteration method has better convergence.

In order to further improve the convergence speed and order of the Jarrat-type iteration method, and expand their scope of use, Cordero et al. [2], denoted by S1, proposed a two-step fourth-order Jarrat-type iterative method:

$$\begin{cases} \tilde{h}^{(k)} = v^{(k)} - \frac{2}{3} [\Gamma'(v^{(k)})]^{-1} [\Gamma(v^{(k)})], \\ v^{(k+1)} = v^{(k+1)} - \frac{1}{2} \left[-I + \frac{9}{4} [\Gamma'(\tilde{h}^{(k)})]^{-1} [\Gamma(v^{(k)})] + \frac{3}{4} [\Gamma'(v^{(k)})]^{-1} [\Gamma(\tilde{h}^{(k)})] \right] [\Gamma'(v^{(k)})]^{-1} [\Gamma(v^{(k)})], \end{cases} \quad (1.2)$$

Francisco et al. [3] combined the Newton method with the Traub method and proposed two new methods for solving nonlinear systems, which were named S2 and S3, respectively:

$$\begin{cases} \tilde{h}^{(k)} = v^{(k)} - [\Gamma'(v^{(k)})]^{-1} [\Gamma(v^{(k)})], \\ v^{(k+1)} = v^{(k)} - [I - 2\sigma_k]^{-1} [1 - \sigma_k] [\Gamma'(v^{(k)})]^{-1} [\Gamma(v^{(k)})], \end{cases} \quad (1.3)$$

and

$$\begin{cases} \tilde{h}^{(k)} = v^{(k)} - [\Gamma'(v^{(k)})]^{-1} [\Gamma(v^{(k)})], \\ v^{(k+1)} = v^{(k)} - [-2\sigma_k^2 + I]^{-1} [\sigma_k + I] [\Gamma'(v^{(k)})]^{-1} [\Gamma(v^{(k)})], \end{cases} \quad (1.4)$$

where $\sigma_k = [\Gamma'(v^{(k)})]^{-1} (\Gamma'(v^{(k)}) - [\tilde{h}^{(k)}, v^{(k)}; F])$.

Inspired by the above methods, this article proposes a fourth-order iterative algorithm based on weight functions. Compared to traditional Jarrat-type iterative methods, this new algorithm exhibits superior convergence performance in solving nonlinear system problems.

This article proposes a single-parameter Jarrat-type iterative method for solving nonlinear systems. In Section 2, the convergence of iterative methods has been rigorously proven within the specific mathematical framework of Banach spaces. Based on the proven convergence conclusions of the iterative method, the stability of the iterative method was analyzed in depth in Section 3 of this article. At the same time, the number of fixed points and critical points was accurately calculated through corresponding mathematical operations. In Section 4, the dynamic plane under different parameters is drawn. The stability of the iterative method under different parameters is analyzed by using the real dynamics method. Determine the parameter values with better stability by comparing different attractive basins. In Section 5, the efficiency of the iterative method designed in this paper is compared with two iterative methods of the same order. In Section 6, four numerical experiments were conducted on the iterative method, and the obtained numerical results further proved that the parameter values of the iterative method have better stability under $0 \leq \beta < \frac{423}{8}$. Finally, in Section 7 of the article, a summary was made.

2. Convergence analysis of Jarrat-type iterative method

According to [4], a new method for solving nonlinear systems has been proposed:

$$\begin{cases} \hbar^{(k)} = \nu^{(k)} - \frac{2}{3} [\Gamma'(\nu^{(k)})]^{-1} [\Gamma(\nu^{(k)})], \\ \nu^{(k+1)} = \nu^{(k)} - G(\lambda) [\Gamma'(\nu^{(k)})]^{-1} [\Gamma(\nu^{(k)})], \end{cases} \quad (2.1)$$

where $\lambda = [\Gamma'(\hbar^{(k)})]^{-1} \Gamma'(\nu^{(k)})$. And

$$G(\lambda) = \left(\frac{5}{8} - \beta\right)I + \beta\lambda + \frac{\beta}{3}\lambda^{-1} + \left(\frac{3}{8} - \frac{\beta}{3}\right)\lambda^2, \quad (2.2)$$

where $k = 0, 1, 2, \dots$, and $G(\lambda)$ is a weight function. In particular, when parameter $\beta = \frac{9}{8}$, the iterative method will be as shown in (1.2). When parameter $\beta = 0$, the iterative method is in the form of S4:

$$\begin{cases} \hbar^{(k)} = \nu^{(k)} - \frac{2}{3} [\Gamma'(\nu^{(k)})]^{-1} [\Gamma(\nu^{(k)})], \\ \nu^{(k+1)} = \nu^{(k)} - \left(\frac{5}{8} + \frac{3}{8}\lambda^2\right) [\Gamma'(\nu^{(k)})]^{-1} [\Gamma(\nu^{(k)})], \end{cases} \quad (2.3)$$

and iterative method (2.1) has a fourth order of convergence, which is proved by the following theorem.

Theorem 2.1. Let $\Gamma : I \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a sufficiently differentiable function in the open convex set I . Let us consider that $\nu^* \in I$ is a root of $\Gamma(\nu) = 0$ and the initial estimate $\nu^{(0)}$ is sufficiently close to ν^* , then the family (2.1) is fourth-order convergent, and the error equation is as follows:

$$\kappa^{(k+1)} = \frac{1}{81} \left((117 + 64\beta) C_2^3 - 81C_2C_3 + 9C_4 \right) (\kappa^{(k)})^4 + O\left((\kappa^{(k)})^5\right), \quad (2.4)$$

where $C_w = \frac{1}{w!} [\Gamma'(\nu^*)]^{-1} \Gamma^{(w)}(\nu^*)$, $w \geq 2$, and $\kappa^{(k)} = \nu^{(k)} - \nu^*$ is the error in k -th iteration.

Proof. The Taylor expansion of $\Gamma(\nu^{(k)})$ and $\Gamma'(\nu^{(k)})$ at point ν^* is as follows:

$$\Gamma(\nu^{(k)}) = \Gamma'(\nu^*) \left[\kappa^{(k)} + C_2(\kappa^{(k)})^2 + C_3(\kappa^{(k)})^3 + C_4(\kappa^{(k)})^4 + C_5(\kappa^{(k)})^5 \right] + O\left((\kappa^{(k)})^6\right), \quad (2.5)$$

and its derivative is expressed as

$$\Gamma'(\nu^{(k)}) = \Gamma'(\nu^*) \left[I + 2C_2\kappa^{(k)} + 3C_3(\kappa^{(k)})^2 + 4C_4(\kappa^{(k)})^3 + 5C_5(\kappa^{(k)})^4 \right] + O\left((\kappa^{(k)})^5\right). \quad (2.6)$$

From the equality $[\Gamma'(\nu^{(k)})]^{-1} \Gamma'(\nu^{(k)}) = I$, $[\Gamma'(\nu^{(k)})]^{-1}$ can be calculated:

$$[\Gamma'(\nu^{(k)})]^{-1} = [\Gamma'(\nu^*)]^{-1} \left(I + L_1\kappa^{(k)} + L_2(\kappa^{(k)})^2 + L_3(\kappa^{(k)})^3 + L_4(\kappa^{(k)})^4 \right) + O\left((\kappa^{(k)})^5\right), \quad (2.7)$$

where

$$\begin{aligned} L_1 &= -2C_2, \quad L_2 = 4C_2^2 - 3C_3, \quad L_3 = -8C_2^3 + 6C_2C_3 + 6C_3C_2 - 4C_4, \\ L_4 &= 16C_2^4 + 9C_3^2 + 8C_2C_4 + 8C_4C_2 - 12C_2^2C_3 - 12C_2C_3C_2 - 12C_3C_2^2 - 5C_5. \end{aligned}$$

Using (2.4) and (2.6), we obtain

$$\hbar^{(k)} - \nu^* = \frac{1}{3}\kappa^{(k)} + \frac{2}{3}C_2(\kappa^{(k)})^2 - \frac{4}{3}(C_2^2 - C_3)(\kappa^{(k)})^3 + \frac{2}{3}(4C_2^3 - 7C_2C_3 + 3C_4)(\kappa^{(k)})^4 + O((\kappa^{(k)})^5). \quad (2.8)$$

Thus,

$$\Gamma(\hbar^{(k)}) = \Gamma'(\nu^*) \left(\frac{1}{3}\kappa^{(k)} + \frac{7}{9}C_2(\kappa^{(k)})^2 - \frac{46}{27}C_3(\kappa^{(k)})^3 + \left(\frac{20}{9}C_2^3 - \frac{32}{9}C_2C_3 + \frac{163}{81}C_4 \right) (\kappa^{(k)})^4 \right) + O((\kappa^{(k)})^5). \quad (2.9)$$

The first derivative of $\Gamma(\hbar^{(k)})$ can be found as

$$\Gamma'(\hbar^{(k)}) = \Gamma'(\nu^*) \left(I + \frac{2}{3}C_2\kappa^{(k)} + \left(\frac{4}{3}C_2^2 + \frac{1}{3}C_3 \right) (\kappa^{(k)})^2 + 2 \left(-\frac{4}{3}C_2^3 + 2C_2C_3 - \frac{2}{27}C_4 \right) (\kappa^{(k)})^3 \right) + O((\kappa^{(k)})^4). \quad (2.10)$$

From the equality $[\Gamma'(\hbar^{(k)})]^{-1} \Gamma'(\hbar^{(k)}) = I$, the inverse of $\Gamma'(\hbar^{(k)})$ can be calculated:

$$[\Gamma'(\hbar^{(k)})]^{-1} = [\Gamma'(\nu^*)]^{-1} \left(I + \Lambda_1 \kappa^{(k)} + \Lambda_2 (\kappa^{(k)})^2 + \Lambda_3 (\kappa^{(k)})^3 \right) + O((\kappa^{(k)})^4), \quad (2.11)$$

where

$$\Lambda_1 = -\frac{2}{3}C_2, \quad \Lambda_2 = -\frac{8}{9}C_2^2 - \frac{1}{3}C_3, \quad \Lambda_3 = -\frac{88}{27}C_2^3 + \frac{34}{9}C_2C_3 + \frac{4}{3}C_2^2 + \frac{1}{3}C_3 - \frac{4}{27}C_4.$$

According to (2.8) and (2.9), we obtain

$$\begin{aligned} \lambda &= [\Gamma'(\hbar^{(k)})]^{-1} \Gamma'(\nu^{(k)}) \\ &= I + \frac{4}{3}C_2\kappa^{(k)} + \frac{2}{3} \left(\frac{26}{3}C_2^2 - 4C_3 \right) (\kappa^{(k)})^2 + \frac{8}{3} \left(\frac{8}{9}C_2^3 - \frac{7}{3}C_2C_3 + \frac{13}{9}C_4 \right) (\kappa^{(k)})^3 + O((\kappa^{(k)})^4). \end{aligned} \quad (2.12)$$

Thus,

$$G(\lambda) = I + C_2\kappa^{(k)} + \left(\left(\frac{40}{9}\beta - 6 \right) C_2^2 + 2C_3 \right) (\kappa^{(k)})^2 + e(\beta) (\kappa^{(k)})^3 + O((\kappa^{(k)})^4), \quad (2.13)$$

where $e(\beta) = 2 \left(\left(-\frac{10}{27}\beta - \frac{1}{6} \right) C_2^3 - \frac{3}{4}C_2C_3 - \frac{3}{12}C_4 \right)$.

Substituting the above equations into the family (2.1), the error equation can be obtained:

$$\kappa^{(k+1)} = \frac{1}{81} \left((117 + 64\beta) C_2^3 - 81C_2C_3 + 9C_4 \right) (\kappa^{(k)})^4 + O((\kappa^{(k)})^5). \quad (2.14)$$

3. Stability analysis

In this section of the research, the focus is on exploring the iterative method family (2.1), with the core focus on whether its stability is related to the value of parameter β . To deeply analyze this key issue, a targeted analysis method was specifically selected, which is to systematically and meticulously analyze the stability of high-order iterative methods in real dynamic planes using dimensional quadratic polynomials. It is worth emphasizing that when conducting stability analysis for the iterative method

family (2.1), we strictly follow the rigorous multidimensional dynamic analysis method, which is a scientific and standardized path. Under the analytical framework constructed by this method, the determination of convergence is carried out by referring to the preliminary estimation process detailed in academic works [5–7], in order to ensure the logical rigor and reliability of the entire analysis process, and provide solid theoretical support for stability-related research in the iterative method family (2.1).

3.1. The basic definition of real dynamics analysis

Applying the n -dimensional rational vector polynomial $p(v)$, $p : \mathbb{R}^n \rightarrow \mathbb{R}^n$, and the iterative method (2.1), the rational function E can be obtained: $\mathbb{R}^n \rightarrow \mathbb{R}^n$, a orbit point is defined as $v^{(0)} \in \mathbb{R}^n$, defining a set of $\mathbb{R} : \{v^{(0)}, E(v^{(0)}), E^2(v^{(0)}), \dots, E^n(v^{(0)})\}$.

It is evident that distinct progressive dynamics are exhibited when conducting a dynamical analysis of initial points within the n -dimensional Banach space. These behaviors can be systematically categorized as follows: A point \bar{v} is a fixed point of E if $E(\bar{v}) = \bar{v}$. Particularly, if \bar{v} is the solution of the equation $E(\bar{v}) \neq \bar{v}$, \bar{v} is a strange fixed point. Furthermore, a point $\bar{v} \in \mathbb{R}^n$ satisfies $E^k(\bar{v}) = \bar{v}$, then $E^i(\bar{v}) \neq \bar{v}$ ($i = 1, 2, \dots, k-1$) is called an i -periodic point; see [8–10].

Theorem 3.1. Consider the function $E : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Let us presuppose that \bar{v} is an i -periodic point that considers the function $E : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Denote the eigenvalues of the Jacobian matrix $E'(\bar{v})$ by $\xi_1, \xi_2, \dots, \xi_n$, then:

- (1) \bar{v} is superattracting if $|\xi_i| = 0$, for all $i = 1, 2, \dots, n$.
- (2) \bar{v} is an attracting if $|\xi_i| < 1$, for all $i = 1, 2, \dots, n$.
- (3) \bar{v} is neutral if $|\xi_i| = 1$, for all $i = 1, 2, \dots, n$.
- (4) \bar{v} is repelling if $|\xi_i| > 1$, for all $i = 1, 2, \dots, n$.
- (5) \bar{v} is a hyperbolic point or saddle point if $|\xi_i| \neq 1$, for all $i = 1, 2, \dots, n$.
- (6) \bar{v} is a superattractive point if $|\xi_i| = 0$, for all $i = 1, 2, \dots, n$.

Allow the basin that draws the fixed point \bar{v} to be the collection of points whose track tends towards \bar{v} , denoted by $A(\bar{v})$,

$$A(\bar{v}) = \{v^{(0)} \in \mathbb{R}^n : V^n(v^{(0)}) \rightarrow \bar{v}, n \rightarrow \infty\}. \quad (3.1)$$

Finally, critical points satisfy $\frac{\partial r_i(v)}{\partial v_i} = 0$, $v \in \mathbb{R}^n$ of V , for all $i = 1, 2, \dots, n$. According to (6) in Theorem 3.1: It can be concluded that the superattractive fixed points also satisfy the definition of critical points.

3.2. Stability analysis of fixed points

This part mainly relates the family of iterative methods (2.1) to polynomial systems. From the perspective of two-dimensional space, the graphical visualization of the basins of attraction on the real dynamic plane is analyzed. And generalize the dynamical properties of two-dimensional polynomials to n -dimensional polynomials. The system under scrutiny within the range of stability analysis is delineated by the ensuing polynomial expression:

$$p_j(v) = v_j^2 - 1, \quad i = 1, 2, \dots, n. \quad (3.2)$$

By employing the family (2.1) to a quadratic polynomial $p(v)$, one derives vector rational operators with the ensuing formulation:

$$V(v, \beta) = \begin{bmatrix} m_1(v, \beta) \\ \vdots \\ m_n(v, \beta) \end{bmatrix}, \quad (3.3)$$

where

$$m_j(v, \beta) = \frac{8\beta(-1 + v_j^2)^4 + 9v_j^2(5 + 31v_j^2 + 91v_j^4 + 17v_j^6)}{144v_j^3(1 + 2v_j^2)^2}, \quad j = 1, 2, \dots, n. \quad (3.4)$$

The stability of the fixed point $X(v, \beta)$ will be summarized in Theorem 3.2 next.

Theorem 3.2. *There will be a combination of $V(v, \beta)$ and the representation method (2.1), with 2^n super-attractive fixed points, whose component is the root of $p(v)$. Let the function $V(v, \beta) = v$, and the real roots obtained are called fixed points. At this point, the fixed points are also the roots of $p(v)$ and the polynomials $l(t) = 8\beta(-1 + t^2)^3 - 9t^2(5 + 20t^2 + 47t^4)$ that depend on β :*

(1) *If $\beta < 0$, there are two real roots, $l_1(\beta)$ and $l_2(\beta)$.*

(2) *If $0 \leq \beta < \frac{423}{8}$, there is no real root here.*

(3) *If $\beta \geq \frac{423}{8}$, there are two real roots, $l_3(\beta)$ and $l_4(\beta)$.*

Proof. The solution of the rational function $V(v, \beta) = v$ is the value of a fixed point:

$$\frac{(-1 + v_j^2) \left(8\beta(-1 + v_j^2)^3 - 9v_j^2(5 + 20v_j^2 + 47v_j^4) \right)}{144v_j^3(1 + 2v_j^2)^2} = 0, \quad j = 1, 2, \dots, n. \quad (3.5)$$

It is actually solving for $(-1 + v_j^2) \left(8\beta(-1 + v_j^2)^3 - 9v_j^2(5 + 20v_j^2 + 47v_j^4) \right)$. Apparently, $v_j = \pm 1$ is a fixed-point polynomial of $p_j(v) = v_j^2 - 1$. Meanwhile, when $t \neq 0$, there are six roots on the $l(t)$ component, but only $l_1(\beta)$ and $l_2(\beta)$ are real roots.

Since the highest degree term of polynomial $l(t)$ is sixth, let $l_1(\beta), l_2(\beta), \dots, l_6(\beta)$ be the six real roots of $l(t)$ as the root of polynomial $l(t) = 0$, where $l_i(\beta)$, $i = 1, 2, \dots, 6$ as the six real roots of $l(t)$ is related to parameter β . The equilibrium of the fixed point $V(v, \beta)$ is ascertained by the sum of the integral eigenvalues of the corresponding Jacobian matrix at the fixed point. Owing to the intrinsic properties of the polynomial system, the eigenvalues align with the coordinate functions of the rational operators. The equilibrium state of a fixed point $V(v, \beta)$ is determined by the sum of the integral eigenvalues of the corresponding Jacobian matrix at that fixed point. Given the inherent properties of polynomial systems, eigenvalues are obtained when the coordinate functions of rational operators exhibit relative homogeneity.

$$\text{Eig}_j(l_j(\beta), \dots, l_j(\beta)) = u_j(\beta) \frac{8\beta(u_j(\beta))^3 - 9(l_j(\beta))^2 \alpha_j(\beta)}{144(l_j(\beta))^3 (h_j(\beta))^2}, \quad (3.6)$$

where $u_j(\beta) = -1 + l_j(\beta)^2$, $\alpha_j(\lambda) = 5 + 20(l_j(\beta))^2 + 47(l_j(\beta))^4$ and $h_j(\beta) = 1 + 2l_j(\beta)^2$. In real space, consider the absolute values of fixed point eigenvalues. Therefore, fixed points with ± 1 components exhibit superattractive fixed points. \square

3.3. Stability analysis of critical point

In this section, if we want to achieve optimal stability in the iterative method of dynamic analysis, we must determine the most favorable value for the parameter β . Therefore, the first-order derivative of the rational function $V(v, \beta)$ can be obtained to analyze the sum critical point of the Jacobian matrix $V'(v, \beta)$. The free critical point is the root that satisfies both $V'(v, \beta) = 0$ and $p(v) \neq 0$ simultaneously.

Theorem 3.3. *The free critical point (cr_1, cr_2, cr_3) constitutes a solution to the equation $V'(v, \beta) = 0$, but it does not satisfy the polynomial equation $p(v) = 0$, that is:*

(1) *If $-\infty < \beta < 0$, the expressions $cr_1 = -\sqrt{r^*}$ and $cr_2 = \sqrt{r^*}$ delineate the distinct components of the free critical points. Here, r^* signifies the positive root of the polynomial*

$$k(\sigma) = 24\beta + (45 + 152\beta)\sigma + (306 + 16\beta)\sigma^2. \quad (3.7)$$

(2) *If $\beta = 0$, there is a free critical point with $cr_3 = 0$.*

(3) *If $\beta > 0$, there dose not exist free critical points.*

Proof. If $V'(v, \beta) = 0$, two components can be multiplied, and the free critical point is the component other than $p(v)$, as follows:

$$\frac{\partial m_j(v_j, \beta)}{\partial v_j} = (p_j(v))^3 \frac{9v_j^2(5 + 34v_j^2) + 8\beta(3 + 19v_j^2 + 2v_j^4)}{144v_j^4(1 + 2v_j^2)^3}. \quad (3.8)$$

Therefore, it is obvious that the free critical point is the root of $24\beta + (45 + 152\beta)\sigma + (306 + 16\beta)\sigma^2$, when the values of the roots are real numbers, they are the components of the critical point. Let $t = \sigma^2$, and the free critical point can be defined by solving $k(t)$. \square

4. Stable and unstable dynamic plane under parameter β

In order to conduct in-depth visualization analysis of the dynamic behavior of iterative algorithms, this study adopted the method of drawing the attractive basin map of the method family under different parameter values β . By careful observation on the dynamic plane, we can evaluate the stability characteristics of specific members in the iterative algorithm family. This method not only reveals the sensitivity of the algorithm to parameter changes, but also provides an intuitive basis for identifying parameter configurations with high stability. In Figure 1, we show the initial iteration set of family (2.1), which is defined by the X_1 -axis and the X_2 -axis in a two-dimensional coordinate system. On this coordinate plane, each specific point (x_1, x_2) represents a guess value during the initial iteration process. These points constitute the geometric representation of the initial iteration set, and their distribution on the plane provides us with an intuitive understanding of the starting point of the iteration process. Each point corresponds to a potential starting position in the iterative algorithm, from which the iterative process will evolve according to the defined mapping rules. By systematically analyzing these initial guess points, we can further explore the dynamic behavior and convergence characteristics of each member in the family (2.1). Within the framework of dynamic plane analysis, it is worth noting that the set of standard mappings encompasses all points whose trajectories tend towards attractive fixed points. Specifically, the dynamic evolution of these points will ultimately

stabilize at a fixed point in the system, which is a key indicator for system stability analysis. Determine the existence of singular fixed points and the correlation between the free critical point and the value of parameter β through Theorems 3.2 and 3.3. To assess the stability across various members of the family (2.1), distinct parameter values are selected. We use graphical visualization tools to construct a 400×400 grid to display the dynamic plane, where each point on the grid is an initial estimated set $(x_1, x_2) \in [-2, 2] \times [-2, 2]$.

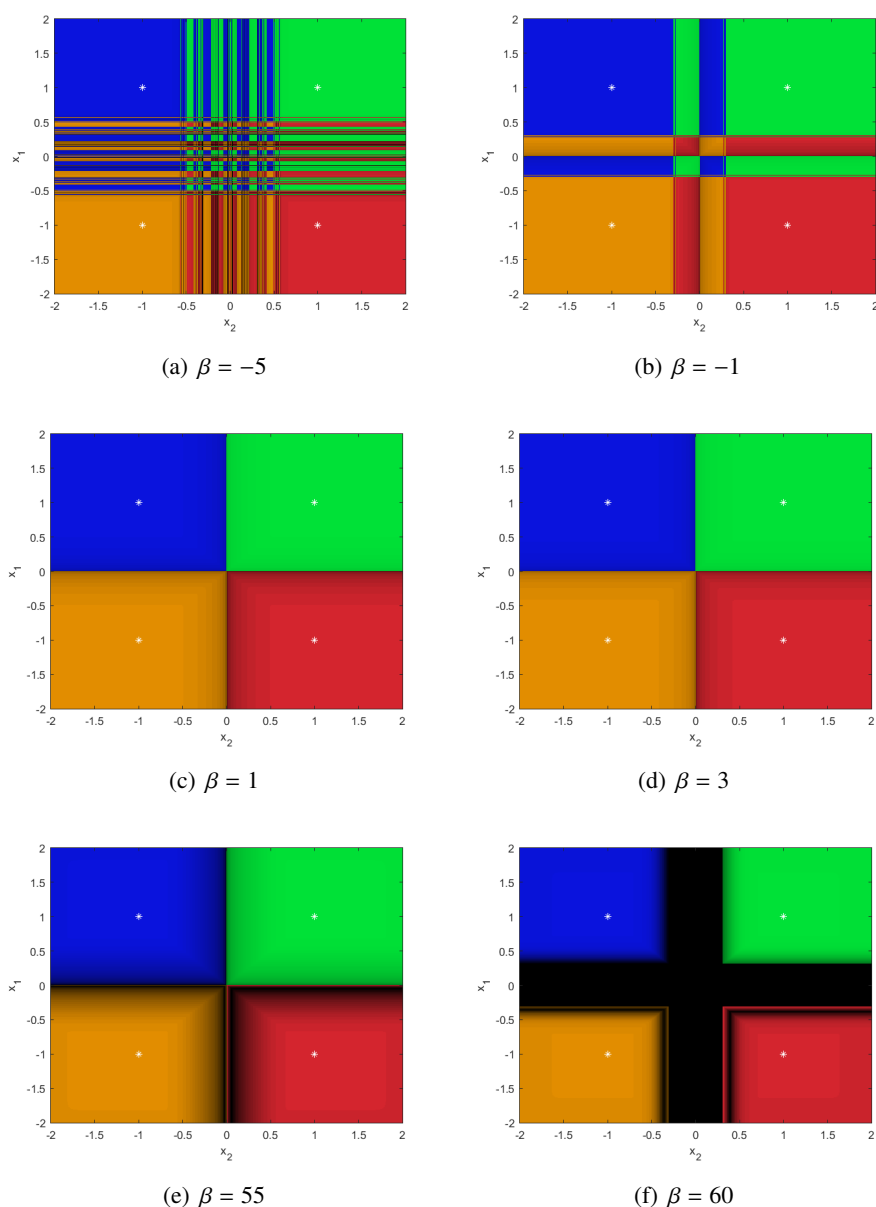


Figure 1. Dynamical planes for different values of β .

In order to achieve intuitive visualization in the dynamic plane, we define specific convergence criteria: When the orbit of a point satisfies the condition $\|(x_1, x_2) - (\pm 1, \pm 1)\| < 10^{-3}$, or when the point has not converged to the root of the polynomial $p(v)$ after 50 iterations, we mark the point with

a white asterisk. This standard ensures that we can identify points that fail to achieve the expected convergence behavior. Furthermore, when the trajectory of a point converges to different roots of a polynomial $p(v)$, these points will be represented in blue, green, orange, and red on the plane to distinguish their corresponding roots. Points that fail to converge to any specific root are represented in black, which helps us to conduct detailed analysis and interpretation of dynamic behavior, as described in references [11–14]. Through this color coding, we can clearly identify points with different convergence properties, thereby gaining a deeper understanding of the structure and behavior of dynamic systems.

Next, the attractive basin in the dynamic plane is shown in Figure 1. Figures 1(a) and 1(b) respectively represent the dynamic planes of parameters $\beta = -5$ and $\beta = -1$, indicating that the convergence domains in these two figures are irregular. The convergence effect is not as good as shown in Figures 1(c) and 1(d). Figure 1(c) corresponds to the dynamic plane with parameter $\beta = 1$, and Figure 1(d) corresponds to the dynamic plane with parameter $\beta = 3$, both of which do not have black convergence regions. The last two figures, Figures 1(e) and 1(f), correspond to the dynamic planes of parameter $\beta = 55$ and parameter $\beta = 60$, respectively. There are obvious black areas of non-convergence in these two planes, especially in the $(x_1, x_2) \in [-0.5, 0.5] \times [-0.5, 0.5]$ area in Figure 1(f). Therefore, according to Figure 1, when the parameters $\beta < 0$ and $\beta \geq \frac{423}{8}$, there are other points of attraction and non-convergence regions, showing poor stability. Finally, when the parameter $0 \leq \beta < \frac{423}{8}$ is within the range, there is no non-convergence region and no other attraction point in the dynamic plane, and the iterative method has the best stability.

5. Computational efficiency index

According to the calculation cost and function evaluation times, in order to select a more effective method, we compare the efficiency of several fourth-order iterative methods proposed in this paper. We can use the efficiency index $I = R^{\frac{1}{k}}$ defined by Cordero [15], where p is the order of the method and k is the number of new function evaluations required for each iteration of the method. The computational cost of solving linear equations depends on its size. Since the proposed methods can be used to solve large-scale equations, we compare the performance of these methods with the computational efficiency index introduced in [16], that is, $CT = R^{\frac{1}{k+op}}$, where OP is the number of operations per iteration (product sum quotient) and the term CL denotes the computational cost. Next, in Table 1, the calculation cost of each operation of n -dimensional linear equations is explained.

Table 1. Comparison of efficiency index of iterative method S1–S4.

Method	R	k	op	CL	CT
S1	4	$2n^2 + n$	$\frac{2}{3}n^3 + 4n^2 - \frac{2}{3}n$	$\frac{2}{3}n^3 + 6n^2 + \frac{1}{3}n$	$4^{\frac{1}{\frac{2}{3}n^3 + 6n^2 + \frac{1}{3}n}}$
S2	4	$\frac{3}{2}n^2 + \frac{n}{2}$	$\frac{2}{3}n^3 + 6n^2 - \frac{2}{3}n$	$\frac{2}{3}n^3 + \frac{15}{2}n^2 - \frac{1}{6}n$	$4^{\frac{1}{\frac{2}{3}n^3 + \frac{15}{2}n^2 - \frac{1}{6}n}}$
S3	4	$\frac{3}{2}n^2 + \frac{n}{2}$	$\frac{2}{3}n^3 + 8n^2 - \frac{2}{3}n$	$\frac{2}{3}n^3 + \frac{19}{2}n^2 - \frac{1}{6}n$	$4^{\frac{1}{\frac{2}{3}n^3 + \frac{19}{2}n^2 - \frac{1}{6}n}}$
S4	4	$2n^2 + n$	$\frac{2}{3}n^3 + 4n^2 - \frac{2}{3}n$	$\frac{2}{3}n^3 + 6n^2 + \frac{1}{3}n$	$4^{\frac{1}{\frac{2}{3}n^3 + 6n^2 + \frac{1}{3}n}}$

Figure 2 shows the efficiency index of iterative methods S1–S4 in Table 1 when n takes different values. It is obvious from Figure 2 that the efficiency of iterative methods S1 and S4 is significantly

higher than that of other iterative methods of the same order. Although the efficiency of these four iterative methods decreases with the increase of the n value, the efficiency of iterative methods S1 and S4 is still higher than that of the other two iterative methods.

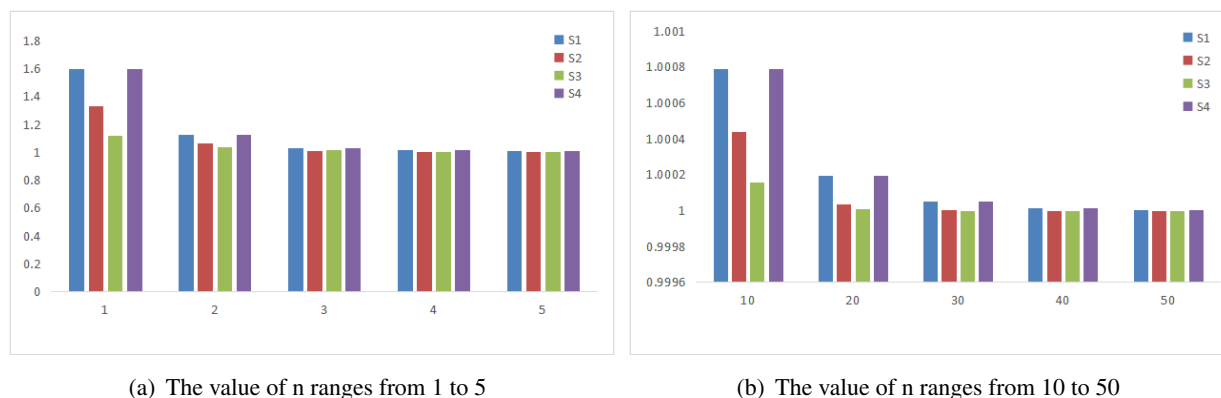


Figure 2. According to Table 1, methods S1–S4 are compared for efficiency at different n values.

6. Numerical experiments

In this section of the study, we adopted the iterative method (2.1) to solve three nonlinear systems. Through this method, we conducted a detailed analysis of the computational accuracy and convergence of the iterative process under different parameter values β . This not only verifies the optimal range of parameters for the stability of the iterative method in the fourth part of the paper, but also better evaluates the performance of the iterative method (2.1). The variables $iter$, $\|v^{(k+1)} - v^{(k)}\|$, $time$, and $\|\Gamma(v^{(k+1)})\|$ in Tables 1–3 represent the number of iterations, the error value, computing time, and the residual error related to the final function value, respectively. Using the approximate computational order of convergence (ACOC) described in [17], the convergence rate of the iterative method is numerically determined to be

$$\rho \approx \frac{\ln\left(\frac{\|v^{(k+1)} - v^{(k)}\|}{\|v^{(k)} - v^{(k-1)}\|}\right)}{\ln\left(\frac{\|v^{(k)} - v^{(k-1)}\|}{\|v^{(k-1)} - v^{(k-2)}\|}\right)}, \quad k = 2, 3, \dots \quad (6.1)$$

Every numerical calculation follows the termination criteria $\|v^{(k+1)} - v^{(k)}\|$ or $\|\Gamma(v^{(k+1)})\| < 10^{-100}$.

Example 6.1. Hammerstein-type integral equation and demonstrate the application of theoretical results. The Hammerstein-type nonlinear integral equation is a specific version of the Urysohn-type integral equation. The following is the form of the Hammerstein equation:

$$v(t) = 1 + \frac{1}{3} \int_0^1 \Theta(t, \omega) u(\omega)^3 d\omega, \quad (6.2)$$

where $v \in \mathbb{C}(0, 1)$, $t, \omega \in [0, 1]$, with the kernel Θ as

$$\Theta(t, \omega) = \begin{cases} (1 - t)\omega, & \text{if } \omega \leq t, \\ (1 - \omega)t, & \text{if } t < \omega. \end{cases} \quad (6.3)$$

By using discretization techniques, Eq (5.1) is transformed into a nonlinear equation form for easy solution. Subsequently, this study used the Gauss-Legendre numerical integration method to approximate the integration in Eq (5.1).

$$\int_0^1 \iota(\omega) d\omega \approx \sum_{i=1}^7 \Psi_i \iota(\omega_i)^3. \quad (6.4)$$

In this study, variables ω_i and Ψ_i were designated as nodes and weights of Gaussian Legendre polynomials, respectively. By constructing a nonlinear system, we represent the approximate value of v_i and estimate Eq (5.2) accordingly. Specifically, for $i = 1, 2, \dots, 7$, we use $v(\omega_i)$ to approximate v_i , where v_i is approximated by

$$v_i = 1 + \frac{1}{3} \sum_{j=1}^7 \xi_{ij} v_j^3, \quad (6.5)$$

where

$$\xi_{ij} = \begin{cases} \Psi_j \omega_j (1 - \omega_i), & \text{if } i \geq j, \\ \Psi_j \omega_i (1 - \omega_j), & \text{if } i < j. \end{cases} \quad (6.6)$$

Rewriting the system in one method is

$$\begin{aligned} \Phi(v) &= v - 1 - \frac{1}{3} K h_v, \quad h_v = (v_1^3, v_2^3, \dots, v_7^3)^T, \\ \Phi'(v) &= I - KM(v), \quad M(v) = \text{diag}(v_1^2, v_2^2, \dots, v_7^2). \end{aligned} \quad (6.7)$$

According to (2.1), we will use Φ' to solve the corresponding nonlinear system. This process involves calculating the derivatives of nonlinear operators and applying them to the solution process of nonlinear systems. By taking $v_0 = (1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5)^T$, we obtain numerical results in Table 2.

Table 2. Numerical results for Example 6.1.

<i>Method</i>	<i>iter</i>	$\ v^{(k+1)} - v^{(k)}\ $	$\ \Gamma(v^{(k+1)})\ $	<i>ACOC</i>	<i>time</i>
<i>M4</i> ₋₅	5	8.024e-279	1.382e-1116	4.00015	0.312
<i>M4</i> ₋₁	5	4.449e-259	1.217e-1036	4.00018	0.921
<i>M4</i> ₁	5	2.365e-235	1.442e-941	4.00018	0.362
<i>M4</i> ₃	5	2.980e-221	4.822e-885	4.00017	0.328
<i>M4</i> ₇	5	3.169e-203	9.208e-813	4.00017	0.593
<i>M4</i> ₅₅	6	2.850e-558	2.990e-2232	4.00001	0.812
<i>M4</i> ₆₀	6	3.212e-546	5.220e-2184	4.00001	0.812
<i>S1</i>	5	2.738e-234	2.641e-937	4.00018	0.265
<i>S2</i>	6	7.374e-693	5.592e-2771	4.00000	0.453
<i>S3</i>	6	3.658e-392	8.622e-1179	4.00000	0.968
<i>S4</i>	5	3.017e-245	3.196e-981	4.00000	0.390

Example 6.2. Boundary-value problem:

$$-\ell''(\vartheta) = z(\vartheta)^2, \vartheta \in [0, 1], \quad (6.8)$$

$$\ell(0) = 0, \ell(1) = 1. \quad (6.9)$$

This study uses the finite difference method to numerically discretize the first and second derivatives in the discussed problem. Through this method, we convert continuous derivative operations into discrete algebraic expressions, making numerical calculations and solutions easier.

$$\ell''_k = \frac{\ell_{k+1} - 2\ell_k + \ell_{k-1}}{m^2}, \quad k = 1, 2, 3, \dots, n-1, \quad (6.10)$$

and

$$\ell'_k = \frac{\ell_{k+1} - \ell_{k-1}}{2m}, \quad k = 1, 2, 3, \dots, n-1. \quad (6.11)$$

In order to split the interval $[0, 1]$ into n smaller intervals, we first define the endpoints as $0 = \vartheta_0 < \vartheta_1 < \dots < \vartheta_{n-1} < \vartheta_n = 1$. This splitting is regular, meaning that the length of each subinterval is equal, which is $\Delta_{\vartheta_k} = \frac{1}{n}$, for all t . On the basis of this rule splitting, we can discover the following nonlinear systems:

$$\ell_{t-1} - 2\ell_t + \ell_{t+1} - m^2 u_k^2 = 0, \quad k = 1, 2, 3, \dots, n-1. \quad (6.12)$$

In practical applications, we may need to choose appropriate numerical methods and computational strategies based on the nature and requirements of specific problems to ensure that the solving process of nonlinear systems is both efficient and accurate. The solution $(0.01073, 0.02147, 0.03220, \dots, 0.9928)^T$ is built from the initial value $\vartheta^{(0)} = (1.5, 1.5, \dots, 1.5)^T$ for $n = 90$. Table 3 displays the numerical results.

Table 3. Numerical results for Example 6.2.

<i>Method</i>	<i>iter</i>	$\ v^{(k+1)} - v^{(k)}\ $	$\ \Gamma(v^{(k+1)})\ $	<i>ACOC</i>	<i>time</i>
<i>M4</i> ₋₅	5	1.383e-177	1.006e-713	4.00010	6.203
<i>M4</i> ₋₁	5	1.121e-257	1.133e-1034	4.00000	6.078
<i>M4</i> ₁	5	1.556e-191	1.439e-769	4.00004	5.859
<i>M4</i> ₃	5	2.043e-169	7.285e-681	4.00006	5.937
<i>M4</i> ₇	5	9.120e-146	5.294e-586	4.00008	6.046
<i>M4</i> ₅₅	6	9.410e-01	5.358e-1222	4.00012	6.750
<i>M4</i> ₆₀	6	9.974e-293	5.304e-1173	4.00009	7.218
<i>S1</i>	5	1.103e-189	3.785e-762	4.00018	5.062
<i>S2</i>	5	2.950e-225	8.306e-905	4.00000	5.765
<i>S3</i>	6	2.228e-784	9.919e-01	4.00000	9.781
<i>S4</i>	5	2.981e-168	2.859e-848	4.00000	6.031

Example 6.3. Let us examine the ensuing system of nonlinear equations (excerpted from Grau-Sánchez), which is articulated by the function:

$$G(t) = \sum_{j=1, j \neq i}^3 t_j - e^{-t_i}, \quad 1 \leq i \leq 3, \quad (6.13)$$

we choose $n = 3$ and use $v^{(0)}$ to represent the initial guess. The solution to this problem is $\omega^* \approx (0.5671433, 0.5671433, 0.5671433)^T$, where we select the initial estimates $(0.5, 0.5, 0.5)^T$. The numerical performance of the iterative method under different parameter values is shown in Table 4.

Table 4. Numerical results for Example 6.3.

<i>Method</i>	<i>iter</i>	$\ v^{(k+1)} - v^{(k)}\ $	$\ \Gamma(v^{(k+1)})\ $	<i>ACOC</i>	<i>time</i>
$M4_{-5}$	5	4.918e-192	1.575e-767	4.00000	0.078
$M4_{-1}$	5	6.281e-230	1.261e-919	4.000000	0.093
$M4_1$	5	4.283e-294	4.396e-1177	4.00000	0.062
$M4_3$	5	7.725e-240	3.816e-959	4.00000	0.062
$M4_7$	5	3.167e-195	2.973e-780	4.00000	0.234
$M4_{55}$	5	1.409e-191	1.275e-765	4.00000	0.109
$M4_{60}$	5	1.189e-179	8.728e-718	4.00000	0.046
$S1$	5	1.712e-252	2.351e-1200	4.00000	0.031
$S2$	5	1.335e-215	2.953e-862	4.00000	0.046
$S3$	5	1.013e-264	2.748e-1059	4.00000	0.078
$S4$	5	5.935e-300	2.920e-1010	4.00000	0.045

Example 6.4. Nonlinear PDE problem [18]:

$$\eta_{xx} = \eta_{\varsigma} + \eta_x - \eta^2 + f(x, \varsigma), \quad x \in [0, 1], \quad \varsigma \geq 0, \quad \eta(0, \varsigma) = \eta(1, \varsigma) = 0, \quad (6.14)$$

$$f(\eta, \varsigma) = e^{-\varsigma} \left(-\pi \cos(\pi\eta) - (\pi^2 - 2) \sin(\pi\eta) \right), \quad (6.15)$$

and the initial condition is

$$u(x, 0) = \sin(\pi x). \quad (6.16)$$

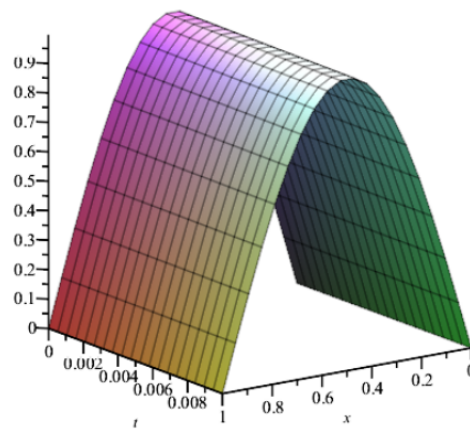
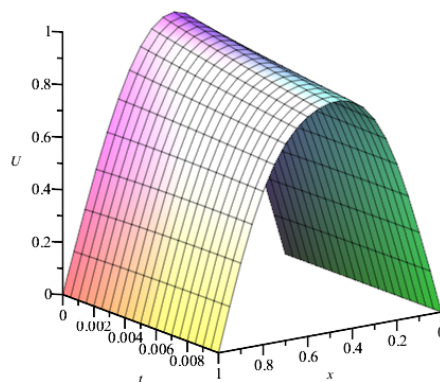
In numerical analysis, this problem is discretized into a nonlinear system via the finite difference. The partial differential equation (PDE) problem under consideration is a heat conduction problem. Specifically, the spatial domain $[0, 1]$ and the temporal domain $[0, T]$ are partitioned into N uniform subintervals, yielding step sizes of $(\varepsilon = \frac{1}{N})$ in the spatial direction and $(\theta = \frac{T}{N})$ in the temporal direction. Let $\eta = \eta(x, \varsigma)$ denote the exact solution of the problem, where $\eta(x, \varsigma)$ is a function of the spatial variable (x) and the temporal variable (ς). Utilizing the finite difference method, we obtain approximations $\eta_{i,j} \approx \eta(x_i, \varsigma_j)$, $\eta_x(x, \varsigma) \approx \frac{\eta(x+\varepsilon, \varsigma) - \eta(x-\varepsilon, \varsigma)}{2\varepsilon}$, $\eta_{\varsigma}(x, \varsigma) \approx \frac{\eta(x, \varsigma) - \eta(x, \varsigma-\theta)}{\theta}$, and $\eta_{xx}(x, \varsigma) \approx \frac{\eta(x+\varepsilon, \varsigma) - 2\eta(x, \varsigma) + \eta(x-\varepsilon, \varsigma)}{\varepsilon^2}$ at discrete points within the domain. This process results in the formation of a nonlinear system, which is presented as follows:

$$(2\theta + \theta\varepsilon)\eta_{i-1,j} + (-4\theta - 2\varepsilon^2)\eta_{i,j} + (2\theta - \theta\varepsilon)\eta_{i+1,j} + 2\theta\varepsilon^2\eta_{i,j}^2 + 2\varepsilon^2\eta_{i,j-1} - 2\theta\varepsilon^2 f(x_i, \varsigma_j) = 0, \quad (6.17)$$

for $i = 1, 2, \dots, N-1$ and $j = 1, 2, \dots, N$, a series of nonlinear systems of size $(N-1) \times (N-1)$ are generated for each fixed j . By varying the values of N and T , the problem is solved using distinct numerical methods. The results are presented in Table 5, where *iter* denotes the number of iterations required for convergence, and *Time* represents the computational CPU time in seconds. The exact solution and the approximate solution of the problem are depicted in Figures 3 and 4, respectively, for the case where $T = 0.01$ and $N = 10$. Additionally, Figure 5 illustrates the absolute error for the problem.

Table 5. Numerical results for Example 6.4.

<i>Method</i>	<i>N</i>	<i>iter</i>	<i>Time</i>
<i>S1</i>	10	2	1.109
<i>S1</i>	20	2	3.140
<i>S1</i>	30	2	11.906
<i>S4</i>	10	2	1.593
<i>S4</i>	20	2	3.454
<i>S4</i>	30	2	14.015

**Figure 3.** The exact solution of S1.**Figure 4.** The approximate solution of S1.

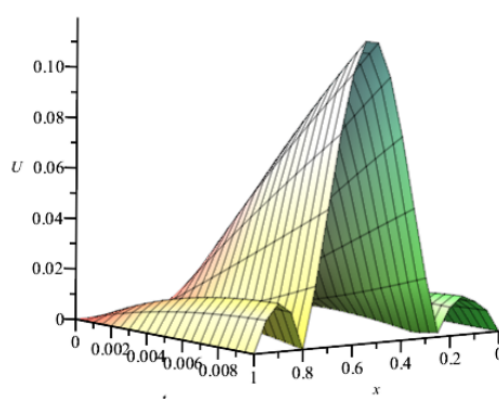


Figure 5. The absolute error of S1.

According to the observations in Tables 1–5, we can conclude that the convergence order of the iterative method (2.1) can reach or approach the fourth-order convergence under different parameter values β . Especially in the range of $0 < \beta < \frac{423}{8}$, the iterative method shows high accuracy. In order to further confirm the optimal performance parameters of the iterative method (2.1), we compared the calculation time of this method and the other two fourth-order iterative methods. The experimental results show that when $\beta = \frac{9}{8}$ is the iterative method S1, the calculation time of the iterative method is the shortest, the error is the smallest, and the calculation efficiency is the best.

7. Conclusions

In this paper, a fourth-order Jarrat-type iterative method for solving nonlinear systems is proposed. The algorithm optimizes the iterative formula (2.1) by introducing a parametric weight function, ensuring that the iterative process can achieve fourth-order convergence regardless of the parameter value. In the stability analysis part, we calculate the fixed point and critical point, and use the critical point to analyze the dynamic plane. In numerical experiments, we solve Hammerstein-type integral equations, boundary value problems, and partial differential equations. The experimental results show that when the parameters meet $0 \leq \beta < \frac{423}{8}$, the iterative method shows excellent stability. In addition, through the comparative experiment with two kinds of fourth-order iterative methods, we found that the iterative method showed obvious advantages in calculation time, cost, and efficiency when the parameter $\beta = \frac{9}{8}$.

Author contributions

Jia Yu: Writing—original draft, Writing—review and editing; Xiaofeng Wang: Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Use of Generative-AI tools declaration

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

Acknowledgments

This research was supported by the Educational Commission Foundation of Liaoning Province of China (No. LJ212410167008) and the Key Project of Bohai University (No. 0522xn078).

Conflict of interest

The authors declare no conflicts of interest.

References

1. I. Bate, K. Senapati, S. George, M. Muniyasamy, G. Chandhini, Jarratt-type methods and their convergence analysis without using Taylor expansion, *Appl. Math. Comput.*, **487** (2025), 129112. <http://doi.org/10.1016/J.AMC.2024.129112>
2. A. Cordero, J. A. Reyes, J. R. Torregrosa, M. P. Vassileva, Stability analysis of a new Fourth-order optimal iterative scheme for nonlinear equations, *Axioms*, **13** (2024), 1–24. <https://doi.org/10.3390/AXIOMS13010034>
3. F. I. Chicharro, A. Cordero, N. Garrido, J. R. Torregrosa, Generalized high-order classes for solving nonlinear systems and their applications, *Mathematics*, **7** (2019), 1–14. <https://doi.org/10.3390/math7121194>
4. A. Cordero, A. Ledesma, J. G. Maimó, J. R. Torregrosa, Design and dynamical behavior of a fourth order family of iterative methods for solving nonlinear equations, *AIMS Math.*, **9** (2024), 8564–8593. <https://doi.org/10.3934/math.2024415>
5. Y. T. Guo, Q. S. Wu, X. F. Wang, An extension of high-order Kou's method for solving nonlinear systems and its stability analysis, *Electron. Res. Arch.*, **33** (2025), 1566–1588. <https://doi.org/10.3934/era.2025074>
6. R. A. Alharbey, M. Kansal, R. Behl, J. A. T. Machado, Efficient three-step class of eighth-order multiple root solvers and their dynamics, *Symmetry*, **11** (2019), 1–30. <https://doi.org/10.3390/sym11070837>
7. R. Thukral, M. S. Petković, A family of three-point methods of optimal order for solving nonlinear equations, *J. Comput. Appl. Math.*, **233** (2010), 2278–2284. <http://doi.org/10.1016/j.cam.2009.10.012>
8. C. Khunpanuk, B. Panyanak, N. Pakkaranang, A new construction and convergence analysis of non-monotonic iterative methods for solving p -demicontractive fixed point problems and variational inequalities involving pseudomonotone mapping, *Mathematics*, **10** (2022), 1–29. <http://doi.org/10.3390/MATH10040623>
9. I. K. Argyros, An inverse-free Newton-Jarratt-type iterative method for solving equations under the gamma condition, *J. Appl. Math. Comput.*, **28** (2008), 15–28. <http://doi.org/10.1007/s12190-008-0065-0>
10. X. F. Wang, N. Shang, Local convergence analysis of a novel derivative-free method with and without memory for solving nonlinear systems, *Int. J. Comput. Math.*, 2025, 1–18. <http://doi.org/10.1080/00207160.2025.2464701>

11. D. Ruan, X. Wang, Y. Wang, Local convergence of seventh-order iterative method under weak conditions and its applications, *Eng. Comput.*, 2025, In press.
12. X. F. Wang, J. Y. Xu, Conformable vector Traub's method for solving nonlinear systems, *Numer. Algor.*, **97** (2024), 1563–1582. <https://doi.org/10.1007/s11075-024-01762-7>
13. X. F. Wang, W. S. Li, Fractal behavior of King's optimal eighth-order iterative method and its numerical application, *Math. Commun.*, **29** (2024), 217–236.
14. D. D. Ruan, X. F. Wang, A high-order Chebyshev-type method for solving nonlinear equations: local convergence and applications, *Electron. Res. Arch.*, **33** (2025), 1398–1413. <https://doi.org/10.3934/era.2025065>
15. A. Cordero, J. L. Hueso, E. Martínez, J. R. Torregrosa, Efficient high-order methods based on golden ratio for nonlinear systems, *Appl. Math. Comput.*, **217** (2011), 4548–4556. <http://doi.org/10.1016/j.amc.2010.11.006>
16. A. Cordero, J. L. Hueso, E. Martínez, J. R. Torregrosa, A modified Newton-Jarratt's composition, *Numer. Algor.*, **55** (2010), 88–99. <https://doi.org/10.1007/s11075-009-9359-z>
17. K. Maatouk, Third order derivative free SPH iterative method for solving nonlinear systems, *Appl. Math. Comput.*, **270** (2015), 557–566. <https://doi.org/10.1016/j.amc.2015.08.083>
18. M. Z. Ullah, S. Serra-Capizzano, F. Ahmad, E. S. Al-Aidarous, Higher order multi-step iterative method for computing the numerical solution of systems of nonlinear equations: application to nonlinear PDEs and ODEs, *Appl. Math. Comput.*, **269** (2015), 972–987. <https://doi.org/10.1016/j.amc.2015.07.096>



AIMS Press

© 2025 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)