Mongolian Mathematical Journal
26 (2025) 1–19
DOI - https://doi.org/10.5564/mmj.v26i1.5002

A family of the best iterative methods for systems of nonlinear equations

Tugal Zhanlav, Khuder Otgondorj*, Khangai Enkhbayar

Abstract. In this paper, we develop an iterative method with scalar and vector coefficients, exhibiting convergence orders ($4 \leq \rho \leq 8$) for solving nonlinear systems and further extend it to m-step formulations. All of these methods require only a single inversion of the Jacobian matrix per iteration. We define concepts such as **best iterative methods**, which require a minimum total cost, allowing us to classify both new and existing methods in terms of their effectiveness. The computational efficiency of the proposed techniques is discussed and compared with existing methods. Moreover, the basin of attraction method is studied for nonlinear systems to validate our findings and identify the most effective methods, while a dynamical analysis confirms the scheme's superior stability and extensive convergence regions. Finally, numerical experiments confirm and validate the theoretical results and demonstrate their effectiveness.

1. Introduction

At present, many m-step ($m = 1, 2, \dots$) iterative methods are widely used for solving systems of nonlinear equations of the form

$$F(x) = 0, \quad x = (x_1, x_2, \cdots, x_n)^T \in \mathbb{R}^n. \tag{1.1}$$

These methods differ in their order of convergence and efficiency index, which is defined as

$$EI = \rho^{\frac{1}{C}}, \tag{1.2}$$

where ρ denotes the order of convergence, and C is the total computational cost, given by

$$C(l, \tilde{m}) = d + op = \frac{l}{3}n^3 + \tilde{m}n^2 + O(n),\ l = 0, 1, 2, \dots, \tilde{m} = m + 1, m + 2, \dots. \tag{1.3}$$

From (1.1) and (1.2), it is clear that using iterations with a high efficiency index is very desirable, i.e., those with the smallest possible total cost. Another important characteristic of iterative methods is the optimality property, which was introduced by Cordero et al. in [1]:

Conjecture 1. *The order of convergence of any iterative method, without memory for solving nonlinear systems cannot exceed $2^{k_1+k_2-1}$, where k_1 is the number for evaluation of the Jacobian matrix and k_2 is the number of evaluations of the nonlinear*

2025 *Mathematics Subject Classification:* 65H05,65D05.

Key Words and Phrases: Nonlinear systems, Higher-order iterations, Order of convergence, Best iterative methods.

Received: 2025.11.15; *Accepted:* 2025.12.15; *Published:* 2025.12.16.

function per iteration, and $k_1 \leq k_2$. When a scheme reaches this upper bound, we say it is optimal.

Of course, the optimal schemes have a higher efficiency index. But the optimality property was given only for scheme of order $2^k, k = 1, 2, \ldots$ whereas there are many iterative methods with $\rho \neq 2^k$ used in practice. In connection with this, a question arises as to how to choose good iterations among the class of ρ-order iterative methods. In our opinion, the very good sorting is efficiency index. We introduce the following definition

Definition 1. *The iterative methods from the class of m-step iterations with convergence order of ρ are called the **best one** if it requires minimal total cost*

$$\min_{l,\tilde{m}} C(l, \tilde{m}) = C(1, m+1) = \frac{1}{3}n^3 + (m+1)n^2 + O(n). \tag{1.4}$$

*They are called **super-efficient** if $l = 0$ in (1.3), that is, if $C(0, \tilde{m}) = O(n^2)$.*

To date, it is not clear whether super-effective methods exist or not. If such a method is found, it will be a revolutionary discovery in numerical analysis. The best schemes are differ from each others only $O(n)$ term and require only one inverse of Jacobian matrix and solve m-linear systems with the same matrix. It should be pointed out that another classification of iterations was given in [12] as maximum efficiency or MAXP. We consider the following m-step iterative methods:

$$\begin{aligned}
&\psi_1^k = x_k - F'(x_k)^{-1}F(x_k),\\
&\psi_2^k = \psi_1^k - H_1 F'(x_k)^{-1}F(\psi_1^k),\\
&\psi_3^k = \psi_2^k - H_2 F'(x_k)^{-1}F(\psi_2^k),\\
&\ldots\\
&\psi_i^k = \psi_{i-1}^k - H_{i-1} F'(x_k)^{-1}F(\psi_{i-1}^k),\ x_{k+1} = \psi_i^k,\ i = 2, 3 \ldots, m
\end{aligned} \tag{1.5}$$

where H_{i-1}, $i = 2, 3, \ldots m$ are parameter coefficients to be determined properly. Using Definition 1 we can state the following theorem.

Theorem 1. *The m-step iterations (1.5) are not best one, if at least one of the parameters H_i is matrix.*

Proof. From (1.5) it is clear that the method requires solving m-linear systems with the same matrix $F'(x_k)$ and, in addition, at least one matrix–vector multiplication (n^2 operations). Thus, the total cost of iterations $\frac{1}{3}n^3 + (m+s)n^2 + O(n)$, $s \geq 2$ and hence the iteration (1.5) are not best one. □

According to Theorem 1, the best scheme may be exist only when H_i are vector or scalar coefficients. Of course, the converse is true for scheme (1.5) with vector or scalar coefficients. The aim of this paper is to construct best schemes with vector or scalar coefficients in $\mathbb{R}^n$ involving point-wise multiplication and division of vectors. First, we consider two- and three-step iterations

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k), \\ x_{k+1} &= y_k - \bar{\tau}_k F'(x_k)^{-1}F(y_k), \end{aligned} \tag{1.6}$$

and

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k), \\ z_k &= y_k - \bar{\tau}_k F'(x_k)^{-1}F(y_k), \\ x_{k+1} &= z_k - \alpha_k F'(x_k)^{-1}F(z_k). \end{aligned} \tag{1.7}$$

In [3, 4] it was proven that the order of convergence of iteration (1.6) is equal to p if $\bar{\tau}_k$ satisfies

$$\bar{\tau}_k = \begin{cases} I + O(h), \ p = 3, & (1.8a) \\ I + 2\eta_k + O(h^2), \ p = 4, & (1.8b) \\ I + 2\eta_k + 3d_k + 5\eta_k^2 + O(h^3), \ p = 5. & (1.8c) \end{cases}$$

Here,

$$\eta_k = \frac{1}{2}F'(x_k)^{-1}F''(x_k)\xi_k, \ d_k = -\frac{1}{6}F'(x_k)^{-1}F'''(x_k)\xi_k^2, \quad \xi_k = F'(x_k)^{-1}F(x_k). \tag{1.9}$$

The order of convergence of three-step iterations (1.7) equal to ρ if $\bar{\tau}_k$ satisfies (1.8) and α_k satisfies condition

$$\alpha_k = \begin{cases} I + O(h), \ \rho = p + 1, & (1.10a) \\ I + 2\eta_k + O(h^2), \ \rho = p + 2, & (1.10b) \\ I + 2\eta_k + 3d_k + 6\eta_k^2 + O(h^3), \ \rho = p + 3. & (1.10c) \end{cases}$$

Here, p is the order of iteration z_k, i.e. $F(z_k) = O(h^p)$.

2. Transition to scheme with vector coefficients

In order to construct a scheme with vector and scalar coefficients, we need to write the coefficients $\bar{\tau}_k$ and α_k satisfying (1.8) and (1.10) in vector form. First, in [6] the conditions (1.8b) and (1.10b) were written as follows

$$\bar{\tau}_k = \frac{\mathbf{1} + a\Theta_k + b\Theta_k^2}{\mathbf{1} + (a-2)\Theta_k + d\Theta_k^2}, \qquad \alpha_k = \frac{\mathbf{1} + \tilde{a}\Theta_k + \tilde{b}\Theta_k^2}{\mathbf{1} + (\tilde{a}-2)\Theta_k + \tilde{d}\Theta_k^2}, \qquad a, b, d, \tilde{a}, \tilde{b}, \tilde{d} \in \mathbb{R}, \tag{2.1}$$

$$\Theta_k = \frac{F(y_k)}{F(x_k)}. \tag{2.2}$$

Let $a = \tilde{a} = 2$. Then (2.1) leads to

$$\bar{\tau}_k = \frac{\mathbf{1} + 2\Theta_k + b\Theta_k^2}{\mathbf{1} + d\Theta_k^2}, \tag{2.3}$$

and

$$\alpha_k = \frac{\mathbf{1} + 2\Theta_k + \tilde{b}\Theta_k^2}{\mathbf{1} + \tilde{d}\Theta_k^2}. \tag{2.4}$$

These expressions represent the vector variants of conditions (1.8b) and (1.10b), respectively. We now proceed to transform the condition (1.8c) and (1.10c). Using Taylor expansion of the smooth enough differentiable function $F(y_k)$ one can easily see that [3]

$$\Theta_k = \eta_k + d_k + O(h^3),\ d_k = O(h^2) \tag{2.5}$$

$$d_k = \frac{F(y_k) - F(w_k)}{2F(x_k)} + \mathbf{1} + O(h^3),\ w_k = x_k + F'(x_k)^{-1}F(x_k), \tag{2.6}$$

$$\eta_k = \frac{F(y_k) + F(w_k)}{2F(x_k)} - \mathbf{1} + O(h^3).$$

Then, the condition (1.8c) can be rewritten in vector form as bellow:

$$\bar{\tau}_k = \mathbf{1} + 2\eta_k + 5\eta_k^2 + 3d_k. \tag{2.7}$$

Analogously, the condition (1.10c) can be expressed in vector form as follows:

$$\alpha_k = \mathbf{1} + 2\eta_k + 6\eta_k^2 + 3d_k. \tag{2.8}$$

Thus, the conditions (1.8) and (1.10) are equivalent to

$$\bar{\tau}_k = \begin{cases} \mathbf{1} + O(h),\ p = 3, & (2.9a) \\ \dfrac{\mathbf{1} + 2\Theta_k + b\Theta_k^2}{\mathbf{1} + d\Theta_k^2} + O(h^2),\ p = 4, & (2.9b) \\ 2\cdot\mathbf{1} + \dfrac{1}{2}\Big(5\Theta_k - \dfrac{F(w_k)}{F(x_k)}\Big) + 5\Theta_k^2 + O(h^3),\ p = 5. & (2.9c) \end{cases}$$

and

$$\alpha_k = \begin{cases} \mathbf{1} + O(h),\ \rho = p + 1, & (2.10a) \\ \dfrac{\mathbf{1} + 2\Theta_k + \tilde{b}\Theta_k^2}{\mathbf{1} + \tilde{d}\Theta_k^2} + O(h^2),\ \rho = p + 2, & (2.10b) \\ 2\cdot\mathbf{1} + \dfrac{1}{2}\Big(5\Theta_k - \dfrac{F(w_k)}{F(x_k)}\Big) + 6\Theta_k^2 + O(h^3),\ \rho = p + 3, & (2.10c) \end{cases}$$

respectively. The combination of (2.9) and (2.10) gives the following best iterative methods with vector coefficients: Fourth-order two step method BI2.11$_4$

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k), \\ x_{k+1} &= y_k - \frac{1}{1 + d\Theta_k^2}F'(x_k)^{-1}\ [(1 + b\Theta_k^2)F(y_k) + 2\Theta_k^2F(x_k)]. \end{aligned} \tag{2.11}$$

Fifth-order two step method BI2.12$_5$

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k), \\ x_{k+1} &= y_k - F'(x_k)^{-1}\ [(2\cdot\mathbf{1} + 5\Theta_k^2)F(y_k) + \frac{1}{2}\Big(5\Theta_k^2 - \frac{F(w_k)F(y_k)}{F(x_k)^2}\Big)F(x_k)]. \end{aligned} \tag{2.12}$$

Furthermore, the fourth-order two-step method (2.11) is denoted by BI2.11$_4$. Similarly, all other considered methods are referred to by their corresponding short names. Sixth-order three step method BI2.13$_6$

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k), \\ z_k &= y_k - \frac{1}{1 + d\Theta_k^2}F'(x_k)^{-1}\ [(1 + b\Theta_k^2)F(y_k) + 2\Theta_k^2F(x_k)], \\ x_{k+1} &= z_k - \frac{1}{1 + \tilde{d}\Theta_k^2}F'(x_k)^{-1}\ [(1 + \tilde{b}\Theta_k^2)F(z_k) + 2\Theta_k\bar{w}_kF(x_k)],\ \bar{w}_k = \frac{F(z_k)}{F(x_k)}, \end{aligned} \tag{2.13}$$

and

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k),\\ z_k &= y_k - F'(x_k)^{-1}\ [(2\cdot\mathbf{1} + 5\Theta_k^2)F(y_k) + \frac{1}{2}\Big(5\Theta_k^2 - \frac{F(w_k)F(y_k)}{F(x_k)^2}\Big)F(x_k)],\\ x_{k+1} &= z_k - F'(x_k)^{-1}F(z_k), \end{aligned} \tag{2.14}$$

and

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k),\\ z_k &= y_k - F'(x_k)^{-1}F(y_k),\\ x_{k+1} &= z_k - F'(x_k)^{-1}[(2\cdot\mathbf{1} + 6\Theta_k^2)F(z_k)\\ &\quad + \frac{1}{2}\Big(5\frac{F(y_k)F(z_k)}{F(x_k)^2} - \frac{F(w_k)F(z_k)}{F(x_k)^2}\Big)F(x_k)]. \end{aligned} \tag{2.15}$$

Let us denote the sixth-order three-step methods (2.14) and (2.15) by $BI2.14_6$ and $BI2.15_6$, respectively. Seventh-order three step method $BI2.16_7$

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k),\\ z_k &= y_k - \frac{1}{1+d\Theta_k^2}F'(x_k)^{-1}\ [(\mathbf{1} + b\Theta_k^2)F(y_k) + 2\Theta_k^2F(x_k)],\\ x_{k+1} &= z_k - F'(x_k)^{-1}\ [(2\cdot\mathbf{1} + 6\Theta_k^2)F(z_k) + \frac{1}{2}\Big(5\Theta_k\bar{w}_k - \frac{F(w_k)F(z_k)}{F(x_k)^2}\Big)\ F(x_k)], \end{aligned} \tag{2.16}$$

and $BI2.17_7$

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k),\\ z_k &= y_k - F'(x_k)^{-1}\ [(2\cdot\mathbf{1} + 5\Theta_k^2)F(y_k) + \frac{1}{2}\Big(5\Theta_k^2 - \frac{F(w_k)F(y_k)}{F(x_k)^2}\Big)F(x_k)],\\ x_{k+1} &= z_k - \frac{1}{1+\tilde{d}\Theta_k^2}F'(x_k)^{-1}\ [(1+\tilde{b}\Theta_k^2)F(z_k) + 2\frac{F(y_k)F(z_k)}{F(x_k)^2}F(x_k)]. \end{aligned} \tag{2.17}$$

Eighth-order three step method $BI2.18_8$

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k),\\ z_k &= y_k - F'(x_k)^{-1}\ [(2\cdot\mathbf{1} + 5\Theta_k^2)F(y_k) + \frac{1}{2}\Big(5\Theta_k^2 - \frac{F(w_k)F(y_k)}{F(x_k)^2}\Big)F(x_k)],\\ x_{k+1} &= z_k - F'(x_k)^{-1}[(2\cdot\mathbf{1} + 6\Theta_k^2)F(z_k)\\ &\quad + \frac{1}{2}\Big(5\frac{F(y_k)F(z_k)}{F(x_k)^2} - \frac{F(w_k)F(z_k)}{F(x_k)^2}\Big)F(x_k)]. \end{aligned} \tag{2.18}$$

For corrections, we pointed out that the scheme (2.18) is a realization variant of scheme (36) in [3]. From (2.11)-(2.18) we see that all these schemes belong to the best one. For completeness and comparison, we present here another best schemes of seventh and eighth order obtained in [5]

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k),\\ z_k &= y_k - F'(x_k)^{-1}((\mathbf{1} + \beta\Theta_k^2)F(y_k) + 2\Theta_k^2F(x_k)),\\ x_{k+1} &= z_k - F'(x_k)^{-1}\Big(\mathbf{1} + 2\Theta_k + (1+\beta)\Theta_k^2 + \frac{F(z_k)}{F(y_k)}\Big)F(z_k). \end{aligned} \tag{2.19}$$

and

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k), \\ z_k &= y_k - F'(x_k)^{-1}((\mathbf{1} + \beta\Theta_k^2)F(y_k) + 2\Theta_k^2 F(x_k)), \\ x_{k+1} &= z_k - F'(x_k)^{-1}\Big(\mathbf{1} + 2\Theta_k + (1+\beta)\Theta_k^2 \\ &\quad + (2\beta + \gamma - 4)\Theta_k^3 + (\mathbf{1} + 4\Theta_k)\frac{F(z_k)}{F(y_k)}\Big)F(z_k). \end{aligned} \tag{2.20}$$

Further, methods (2.19) and (2.20) are denoted by BS2.19$_7$ and BS2.20$_8$, respectively. It is worth noting that, according to Conjecture 1 and Definition 1, the iteration (2.20) is the best and optimal eighth-order scheme. However, in some cases, the best scheme may not be optimal. For example, we present the eighth-order scheme constructed in [3] (see Theorem 10 therein). It has the same form (1.7) with $\bar{\tau}_k$, α_k given by (2.7) and (2.8). This best eighth-order scheme is not optimal, because of $k_1 = 1$, $k_2 = 4$ and is maximally efficient [12], because of $k_2 = 3$.

3. Transition to schemes with scalar coefficients

Using transition rules [2] and replacing the point-wise multiplication by dotted product we

$$\begin{aligned} \frac{F(z_k)F(y_k)}{F(x_k)^2} &\Longleftrightarrow \alpha_k = \frac{F(z_k)^T F(y_k)}{\|F(x_k)\|^2}, \\ \frac{F(y_k)F(w_k)}{F(x_k)^2} &\Longleftrightarrow u_k = \frac{F(w_k)^T F(y_k)}{\|F(x_k)\|^2}, \\ \frac{F(w_k)F(z_k)}{F(x_k)^2} &\Longleftrightarrow \psi_k = \frac{F(w_k)^T F(z_k)}{\|F(x_k)\|^2}, \\ \Theta_k^2 &\Longleftrightarrow v_k = \frac{\|F(y_k)\|^2}{\|F(x_k)\|^2}, \ \mathbf{1} \Longleftrightarrow 1, \end{aligned}$$

in schemes (2.11)-(2.18) we obtain corresponding schemes with scalar coefficients: Fourth-order and fifth-order two step method BS3.1$_4$

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k), \\ x_{k+1} &= y_k - \frac{1}{1 + dv_k}F'(x_k)^{-1}\ [(1 + bv_k)F(y_k) + 2v_kF(x_k)], \end{aligned} \tag{3.1}$$

and BS3.2$_5$

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k), \\ x_{k+1} &= y_k - F'(x_k)^{-1}\ [(2 + 5v_k)F(y_k) + \frac{1}{2}(5v_k - u_k)F(x_k)]. \end{aligned} \tag{3.2}$$

Sixth-order three step method BS3.3$_6$

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k), \\ z_k &= y_k - \frac{1}{1 + dv_k}F'(x_k)^{-1}\ [(1 + bv_k)F(y_k) + 2v_kF(x_k)], \\ x_{k+1} &= z_k - \frac{1}{1 + \tilde{d}v_k}F'(x_k)^{-1}\ [(1 + \tilde{b}v_k)F(z_k) + 2\alpha_kF(x_k)], \end{aligned} \tag{3.3}$$

and

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k), \\ z_k &= y_k - F'(x_k)^{-1}\,[(2+5v_k)F(y_k) + \frac{1}{2}(5v_k - u_k)F(x_k)], \\ x_{k+1} &= z_k - F'(x_k)^{-1}F(z_k), \end{aligned} \tag{3.4}$$

and

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k), \\ z_k &= y_k - F'(x_k)^{-1}F(y_k), \\ x_{k+1} &= z_k - F'(x_k)^{-1}\,[(2+6v_k)F(z_k) + \frac{1}{2}(5\alpha_k - \psi_k)F(x_k)]. \end{aligned} \tag{3.5}$$

Furthermore, denote the sixth-order three-step methods (3.4) and (3.5) by BS3.4_6 and BS3.5_6, respectively. Seventh-order three step method BS3.6_7

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k), \\ z_k &= y_k - \frac{1}{1+dv_k}F'(x_k)^{-1}\,[(1+bv_k)F(y_k) + 2v_kF(x_k)], \\ x_{k+1} &= z_k - F'(x_k)^{-1}\,[(2+6v_k)F(z_k) + \frac{1}{2}(5\alpha_k - \psi_k)F(x_k)], \end{aligned} \tag{3.6}$$

and BS3.7_7

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k), \\ z_k &= y_k - F'(x_k)^{-1}\,[(2+5v_k)F(y_k) + \frac{1}{2}(5v_k - u_k)F(x_k)], \\ x_{k+1} &= z_k - \frac{1}{1+\tilde{d}v_k}F'(x_k)^{-1}\,[(1+\tilde{b}v_k)F(z_k) + 2\alpha_kF(x_k)]. \end{aligned} \tag{3.7}$$

Eighth-order three step method BS3.8_8

$$\begin{aligned} y_k &= x_k - F'(x_k)^{-1}F(x_k), \\ z_k &= y_k - F'(x_k)^{-1}\,[(2+5v_k)F(y_k) + \frac{1}{2}(5v_k - u_k)F(x_k)], \\ x_{k+1} &= z_k - F'(x_k)^{-1}\,[(2+6v_k)F(z_k) + \frac{1}{2}(5\alpha_k - \psi_k)F(x_k)]. \end{aligned} \tag{3.8}$$

It should be pointed out that, in [2], iterative methods with scalar coefficients were first proposed. But they require two inverses $F'(x_k)^{-1}$ and $F'(y_k)^{-1}$. Hence, they do not belong to the best one.

4. The best m-step iterations

By properly, choosing parameter H_i one can obtain different m-step iterative methods with different convergence order. For example, we choose H_i in (1.5) as:

$$H_1 = H_2 = \cdots = H_{m-1} = \frac{\mathbf{1} + 2\Theta_k + b\Theta_k^2}{\mathbf{1} + d\Theta_k^2},\ \Theta_k = \frac{F(y_k)}{F(x_k)} = \frac{F(\psi_1^k)}{F(x_k)},\ b, d \in \mathbb{R}.$$

Then iteration (1.5) reads as:

$$\begin{aligned}
&\psi_1^k = x_k - F'(x_k)^{-1}F(x_k),\\
&\psi_i^k = \psi_{i-1}^k - \frac{1}{1+d\Theta_k^2}F'(x_k)^{-1}\,[(1+b\Theta_k^2)F(\psi_{i-1}^k) + 2\frac{F(\psi_1^k)F(\psi_{i-1}^k)}{F(x_k)^2}F(x_k)],\\
&\qquad\qquad\qquad\qquad i = 2,3\dots,m,\\
&x_{k+1} = \psi_m^k.
\end{aligned} \tag{4.1}$$

We denote method (4.1) by BS4.1. From (2.9b) and (2.10b) it is clear that the order of local convergence of (4.1) is equal to $\rho = 2m$. The iterations (4.1) require only one inverse matrix $F'(x_k)^{-1}$ and to solve m linear system with the same matrix $F'(x_k)$. So the total cost of iterations (4.1) is equal to

$$C = \frac{1}{3}n^3 + (m+1)n^2 + O(n).$$

Thus, according to Definition 1, the iterations (4.1) belongs to class of the best schemes with order $\rho = 2m$. The version iterations (4.1) with scalar coefficient is BS4.2:

$$\begin{aligned}
&\psi_1^k = x_k - F'(x_k)^{-1}F(x_k),\\
&\psi_i^k = \psi_{i-1}^k - \frac{1}{1+dv_k}F'(x_k)^{-1}\,[(1+bv_k)F(\psi_{i-1}^k) + 2\alpha_{i-1,k}F(x_k)],\ i = 2,3,\dots,m,\\
&x_{k+1} = \psi_m^k,
\end{aligned} \tag{4.2}$$

where

$$v_k = \frac{\|F(\psi_1^k)\|^2}{\|F(x_k)\|^2},\ \alpha_{i-1,k} = \frac{F(\psi_{i-1}^k)^T F(\psi_1^k)}{\|F(x_k)\|^2},\ i = 2,3,\dots,m.$$

Of course, the iterations (4.2), like the iterations (4.1), also belong to the class of best schemes with convergence order $\rho = 2m$. Analogously, we consider the following $m-$step iterations

$$\begin{aligned}
&\psi_1^k = x_k - F'(x_k)^{-1}F(x_k),\\
&\psi_i^k = \psi_{i-1}^k - F'(x_k)^{-1}\,[2\cdot\mathbf{1} + \sigma\Theta_k^2 + \frac{1}{2}(5\Theta_k - \gamma_k)]F(\psi_{i-1}^k),\ i = 2,3\dots,m\\
&x_{k+1} = \psi_m^k,
\end{aligned} \tag{4.3}$$

where

$$\Theta_k = \frac{F(\psi_1^k)}{F(x_k)},\ \gamma_k = \frac{F(w_k)}{F(x_k)},\ w_k = x_k + F'(x_k)^{-1}F(x_k) = 2x_k - \psi_1^k \tag{4.4}$$

and

$$\sigma_k = \begin{cases} 5, & \text{when } i = 2\\ 6, & \text{when } i \geq 3. \end{cases} \tag{4.5}$$

The iteration (4.3) is based on the choices (2.9c) and (2.10c) and its convergence order is equal to $\rho = 3m - 1$. From (4.3) easily proceed to its version with scalar coefficients and it has a form:

$$\begin{aligned}
&\psi_1^k = x_k - F'(x_k)^{-1}F(x_k),\\
&\psi_i^k = \psi_{i-1}^k - F'(x_k)^{-1}\,[(2\cdot\mathbf{1} + \sigma v_k)F(\psi_{i-1}^k) + \alpha_{i-1,k}F(x_k)],\ i = 2,3,\dots,m,\\
&x_{k+1} = \psi_m^k,
\end{aligned} \tag{4.6}$$

where

$$v_k = \frac{\|F(\psi_1^k)\|^2}{\|F(x_k)\|^2}, \ \alpha_{i-1,k} = \frac{1}{2}\frac{F(\psi_{i-1}^k)^T(5F(\psi_1^k) - F(w_k))}{\|F(x_k)\|^2}, \ i = 2, 3, \ldots, m. \tag{4.7}$$

Of course, the convergence order ($\rho = 3m - 1$) is maintained for (4.6). Note that in [9, 10] were considered similar to (4.3) multi-step (m) iteration with convergence order of $2m + 1$ and $2m + \rho$ respectively. The iteration (4.3) for $m = 2$ and $m = 4$ was considered in [3].

5. Computational efficiency

Evaluating the computational efficiency of higher-order iterative methods is an important task, as the performance of an algorithm depends on many parameters. We employ the efficiency index as defined in [14, 15], $CI = \rho^{\frac{1}{C}}$, where ρ denotes the order of convergence of the iterative method and C represents the computational cost per iteration. We will investigate the computational efficiency of the proposed methods and compare them with the methods in [7], [10], [11], [12] and [13].

The components contributing to the total computational cost are as follows. In any iterative method, computing F requires the evaluation of n scalar functions, whereas the computation of a new derivative F' involves n^2 scalar function evaluations. Recall that solving r linear systems with the same coefficient matrix by LU factorization requires approximately $\frac{1}{3}n^3 + rn^2 + \frac{1}{3}n$ where n denotes the dimension of the system. Thus, the computational cost rises by n^2 for each additional system with the same coefficient matrix.

As shown in Figure 1 and presented in Tables 1, the proposed $BI2.13_6$ and $BI3.3_6$ schemes exhibit higher computational efficiency and better overall performance compared to other existing fourth-, fifth-, and sixth-order methods. Moreover, the method ST8S [13], along with the $BI2.18_8$ and $BS3.8_8$ methods, achieves higher computational efficiency and overall performance than other existing seventh- and eighth-order methods.

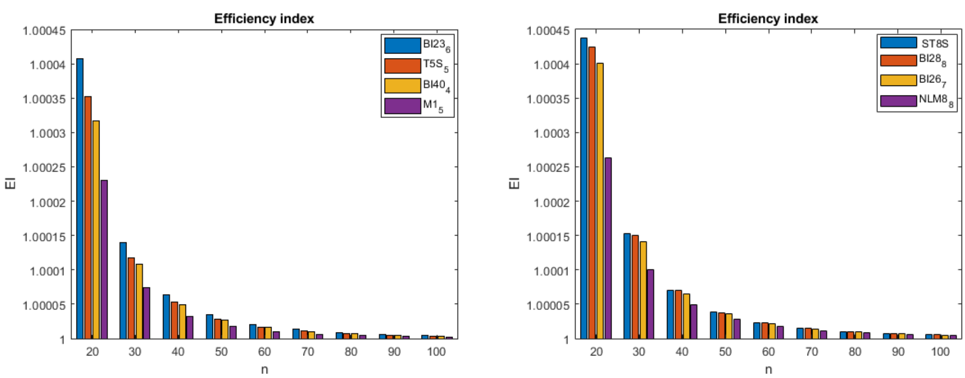


Figure 1: Efficiency indices for different sizes of the system.

Table 1: Comparison of computational efficiency

№	methods	ρ	C_i	CI
1	$BI2.11_4$, $BS3.1_4$	4	$C_1 = \frac{1}{3}n^3 + 3n^2 + \frac{11}{3}n$	$4^{1/C_1}$
2	$BI2.12_5$, $BI3.2_5$	5	$C_2 = \frac{1}{3}n^3 + 4n^2 + \frac{20}{3}n$	$5^{1/C_2}$
3	$BI2.13_6$, $BI3.3_6$	6	$C_3 = \frac{1}{3}n^3 + 4n^2 + \frac{22}{3}n$	$6^{1/C_3}$
4	$BI2.14_6$, $BS3.4_6$	6	$C_4 = \frac{1}{3}n^3 + 5n^2 + \frac{23}{3}n$	$6^{1/C_4}$
5	$BI2.15_6$, $BI3.5_6$	6	$C_5 = \frac{1}{3}n^3 + 5n^2 + \frac{29}{3}n$	$6^{1/C_5}$
6	$BI2.16_7$, $BS3.6_7$	7	$C_6 = \frac{1}{3}n^3 + 5n^2 + \frac{29}{3}n$	$7^{1/C_6}$
7	$BI2.17_7$, $BS3.7_7$	7	$C_7 = \frac{1}{3}n^3 + 5n^2 + \frac{29}{3}n$	$7^{1/C_7}$
8	$BI2.18_8$, $BS3.8_8$	8	$C_8 = \frac{1}{3}n^3 + 5n^2 + \frac{35}{3}n$	$8^{1/C_8}$
9	$M1_5$ [7]	5	$C_9 = \frac{2}{3}n^3 + 4n^2 + \frac{10}{3}n$	$5^{1/C_9}$
10	$NLM5_5$ [11]	5	$C_{10} = \frac{1}{3}n^3 + 8n^2 + \frac{5}{3}n$	$5^{1/C_{10}}$
11	$T5S_5$ [12]	5	$C_{11} = \frac{1}{3}n^3 + 4n^2 + \frac{17}{3}n$	$5^{1/C_{11}}$
12	ST8S [13]	8	$C_{13} = \frac{1}{3}n^3 + 5n^2 + \frac{14}{3}n$	$8^{1/C_{13}}$
13	$ZMO1_8$ [10]	8	$C_{14} = \frac{2}{3}n^3 + 7n^2 + \frac{17}{3}n$	$8^{1/C_{14}}$
14	$ZMO2_8$ [10]	8	$C_{15} = \frac{2}{3}n^3 + 6n^2 + \frac{11}{3}n$	$8^{1/C_{15}}$
15	$NLM8_8$ [11]	8	$C_{15} = \frac{1}{3}n^3 + 13n^2 + \frac{8}{3}n$	$8^{1/C_{16}}$

6. Numerical experiments

We demonstrate the efficiency and confirm the theoretical results of the proposed methods, which include the fourth-order methods $BI2.11_4$ and $BS3.1_4$; the fifth-order methods $BI2.12_5$ and $BS3.2_5$; the sixth-order methods $BI2.13_6$, $BI2.14_6$, $BI2.15_6$, $BS3.3_6$, $BS3.4_6$, and $BS3.5_6$; the seventh-order methods $BI2.16_7$, $BI2.17_7$, $BS2.19_7$, $BS3.6_7$, and $BS3.7_7$; the eighth-order methods $BI2.18_8$, $BS2.20_8$, and $BS3.8_8$; and finally, the m-step methods BS4.1 and BS4.2. Furthermore, we compare our methods with several established iterative methods: the fourth- and sixth-order methods $M2_4$ and $M2_6$ by J. L. Hueso et al. (2015) [9]; the fifth- and eighth-order methods $NLM5_5$ and $NLM8_8$ by J. R. Sharma et al. (2017) [11]; the eighth-order methods $ZMO1_8$ and $ZMO2_8$ proposed in 2023 [10]; the fifth- and sixth-order methods $T5S_5$ and $EH6S_6$ by A. Cordero et al. (2025) [12]; the m-step eighth-order method ST8S ($m = 4$) by A. Cordero et al. (2025) [13]; and the fifth-order method $M1_5$ by H. Singh et al. (2023) [8].

All of our numerical computations were performed in programming system MATLAB R2024a using multi-precision arithmetic with 1500 digits for Intel(R) i5-13700, 2.10GHz, 32GB RAM. In all experiments, the iterative process was terminated when the following stopping criterion was satisfied:

$$\|x_{k-1} - x_k\| \leq 10^{\varepsilon}, \quad \varepsilon = -100.$$

The computational order of convergence of the method at iteration k is defined as:

$$\rho_k = \frac{\ln(\|x_{k+1} - x_k\|/\|x_{k-1} - x_k\|)}{\ln(\|x_{k-1} - x_k\|/\|x_{k-2} - x_{k-1}\|)}.$$

For the numerical experiments, the following examples are considered and solved:

Example 1. *Let us consider the following system of 13 equations:*

$$\sum_{j=1,j\neq i}^{n} x_j - e^{-x_i} = 0,\ i = 1, 2, \ldots, n$$

with $n = 13$. *The initial vector is* $x_0 = (1.5, 1.5, \ldots, 1.5)^T$ *and exact solution is* $\alpha = (0.077146207613064638, 0.077146207613064638, \ldots, 0.077146207613064638)^T$. *The numerical results can be seen in Table 2.*

Example 2. *Consider the following system of equations:*

$$\tan^{-1}(x_i) + 1 - 2 \sum_{j=1,j\neq i}^{n} x_j = 0,\ i = 1, 2, \ldots, n$$

with $n = 100$. *The exact solution is* $(0.0588, 0.0588, \ldots, 0.0588)^T$ *and* $x_0 = (1/10, 1/10, \cdots, 1/10)^T$ *is the initial vector. The numerical results can be seen in Table 3.*

Example 3. *Next, we consider the following system of nonlinear equations with trigonometric functions:*

$$\begin{cases} x_i \sin(x_{i+1}) - 1 = 0,\ i = 1, 2, \ldots, n-1 \\ x_n \sin(x_1) - 1 = 0 \end{cases}$$

with $n = 500$. *The exact solution is* $(1.11415714, 1.11415714, \ldots, 1.11415714)^T$ *and* $x_0 = (1.3, 1.3, \cdots, 1.3)^T$ *is the initial vector. The numerical results can be seen in Table 4.*

Example 4. *Consider the system of equations:*

$$\begin{cases} x_i^2 x_{i+1} - 1 = 0,\ i = 1, 2, \ldots, n-1 \\ x_n^2 x_1 - 1 = 0 \end{cases}$$

The exact solution is $(1, 1, \ldots, 1)^T$ *and the initial vector is* $x_0 = (5/4, 5/4, \cdots, 5/4)^T$. *The numerical results can be seen in Table 5.*

The computational results in Table 2 show that the methods with the lowest CPU times were the fourth-order BI2.11_4 (0.556 s), the fifth-order BI2.12_5 (0.525 s), the sixth-order EH6S_6 (0.574 s), the seventh-order BS2.19_7 (0.603 s), and the eighth-order BS2.20_8 (0.607 s), respectively.

From the results presented in Table 3, it is evident that the methods with the lowest CPU times were the fourth-order BI2.11_4 (19.604 s), the fifth-order BI2.12_5 (20.235 s), the sixth-order EH6S_6 (21.722 s), the seventh-order BS2.19_7 (21.848 s), and the eighth-order BS2.20_8 (21.806 s).

Next, Table 4 indicates that the methods exhibiting the lowest CPU times were the fourth-order BI2.11_4 (20.189 s), the fifth-order BS3.2_5 (18.891 s), the sixth-order BI2.13_6 (21.413 s), the seventh-order BS2.19_7 (22.020 s), and the eighth-order BS2.20_8 (22.260 s).

Table 2: Comparison of numerical results for Example 1

Method	Iter	$\|x_{k+1} - x_k\|$	ACOC	CPU Time
$BI2.11_4$ $(b = 0, d = -4)$	5	2.0631e-290	4.00	**0.556**
$BS3.1_4$ $(b = 0, d = -4)$	5	2.0631e-290	4.00	0.568
BS4.1 $(m = 2, b = 0, d = -4)$	5	2.0632e-290	4.00	0.691
BS4.2 $(m = 2, b = 0, d = -4)$	5	2.0632e-290	4.00	0.695
$M2_4$ [9]	5	2.7184e-331	4.00	1.122
$BI2.12_5$	4	1.4668e-117	5.00	**0.525**
$M1_5$ [8]	4	1.5371e-145	5.00	0.527
$BS3.2_5$	4	1.4668e-117	5.00	0.559
$T5S_5$ [12]	4	1.5614e-149	5.00	0.613
EH1 $(m = 2)$ [17]	4	7.7874e-129	5.00	0.631
NLM_5 [11]	4	4.9555e-189	5.00	0.883
$EH6S_6$ [12]	4	2.7802e-242	6.00	**0.574**
$BI2.13_6$ $(b = \bar{b} = 0, d = \bar{d} = -4)$	4	2.9284e-237	6.00	0.614
$BI2.15_6$	4	7.3256e-218	6.00	0.695
$BI2.14_6$	4	1.3738e-215	6.00	0.698
$BS3.4_6$	4	1.3738e-215	6.00	0.716
$BS3.3_6$ $(b = \bar{b} = 0, d = \bar{d} = -4)$	4	2.9284e-237	6.00	0.721
$BS3.5_6$	4	7.3256e-218	6.00	0.721
BS4.1 $(m = 3, b = 0, d = -4)$	4	1.457e-231	6.00	0.716
BS4.2$(m = 3, b = 0, d = -4)$	4	1.457e-231	6.00	0.729
$M2_6$ [9]	4	8.7325e-239	6.00	1.176
$BS2.19_7$ $(\beta = 2)$	4	1.7937e-363	7.00	**0.603**
$BI2.17_7$ $(\bar{b} = 0, \bar{d} = -4)$	4	4.6341e-335	7.00	0.711
$BS3.6_7$ $(b = 0, d = -4)$	4	2.3128e-328	7.00	0.711
$BS3.7_7$ $(\bar{b} = 0, \bar{d} = -4)$	4	4.6341e-335	7.00	0.728
$BI2.16_7$ $(b = 0, d = -4)$	4	2.3128e-328	7.00	0.762
EH1 $(m = 3)$ [17]	4	7.6065e-377	7.00	0.986
$BS2.20_8$ $(\beta = 2, \gamma = 0)$	4	1.3082e-537	8.00	**0.607**
BS4.3$(m = 3)$	4	3.7624e-447	8.00	0.655
BS4.6 $m = 3$	4	3.7624e-447	8.00	0.674
$BI2.18_8$	4	3.7624e-447	8.00	0.696
ST8S [13]	4	4.9853e-562	8.00	0.699
$BS3.8_8$	4	3.7624e-447	8.00	0.729
BS4.1 $(m = 4, b = 0, d = -4)$	4	3.9742e-536	8.00	0.852
BS4.2 $(m = 4, b = 0, d = -4)$	4	3.9742e-536	8.00	0.876
NLM_8 [11]	4	2.6110e-729	8.00	1.238
$ZMO2_8$ [10]	4	3.3748e-613	8.00	1.282
$ZMO1_8$ [10]	4	2.0847e-437	8.00	1.727

Table 5 further shows that the methods achieving the lowest CPU times were the fourth-order $BS3.1_4$ (84.635 s), the fifth-order $BS3.2_5$ (96.457 s), the sixth-order $BI2.13_6$ (88.963 s), the seventh-order $BS2.19_7$ (88.277 s), and the eighth-order $BS2.20_8$ (88.538 s).

Finally, based on the results presented in Tables 1–5 for Examples 1–5, we can conclude that the fourth-order $BS3.1_4$, fifth-order $BS3.2_5$, sixth-order $BI2.13_6$ and $EH6S_6$, seventh-order $BS2.19_7$, and eighth-order $BS2.20_8$ methods require less computational time than the other methods.

Note that The iteration $EH6S_6$ presented in [13] is a special case of $BI2.13_6$ with parameters $(b = d = 0, \bar{b} = \bar{d} = 0)$.

Table 3: Comparison of numerical results for Example 2

Method	Iter	$\|x_{k+1} - x_k\|$	ACOC	CPU Time
$BI2.11_4$ $(b = 0, d = -4)$	4	6.7672e-202	4.00	**19.604**
$BS3.1_4$ $(b = 0, d = -4)$	4	6.7672e-202	4.00	19.632
BS4.2 $(m = 2, b = 0, d = -4)$	4	6.7672e-202	4.00	20.311
BS4.1 $(m = 2, b = 0, d = -4)$	4	6.7672e-202	4.00	20.313
$M2_4$ [9]	4	6.5715e-165	4.00	110.642
$BI2.12_5$	4	5.3470e-345	5.00	**20.235**
$BS3.2_5$	4	5.3470e-345	5.00	20.306
$M1_5$ [8]	4	1.3359e-432	5.00	20.999
$T5S_5$ [12]	4	1.6668e-435	5.00	29.141
EH1 $(m = 2)$ [17]	4	4.9113e-422	5.00	29.451
NLM_5 [11]	4	1.0574e-440	5.00	102.721
$EH6S_6$ [12]	3	7.9900e-101	6.08	**21.722**
$BI2.13_6$ $(b = \bar{b} = 0, d = \bar{d} = -4)$	3	8.0036e-101	6.08	21.756
BS4.1 $(m = 3, b = 0, d = -4)$	3	8.0172e-101	6.08	22.326
$BS3.5_6$	3	9.0480e-109	6.22	22.359
$BI2.15_6$	3	9.0480e-109	6.22	22.366
$BS3.3_6$ $(b = \bar{b} = 0, d = \bar{d} = -4)$	3	8.0172e-101	6.08	22.376
$BS3.4_6$	3	8.9547e-109	6.22	22.415
BS4.2$(m = 3, b = 0, d = -4)$	3	8.0172e-101	6.08	22.427
$BI2.14_6$	3	8.9547e-109	6.22	22.439
$M2_6$ [9]	4	6.5872e-541	6.00	122.878
$BS2.19_7$ $(\beta = 2)$	3	1.5726e-147	7.12	**21.848**
$BS3.7_7$ $(\bar{b} = 0, \bar{d} = -4)$	3	6.4648e-127	7.15	22.328
$BI2.17_7$ $(\bar{b} = 0, \bar{d} = -4)$	3	6.4520e-127	7.15	22.500
$BS3.6_7$ $(b = 0, d = -4)$	3	6.5420e-127	7.15	23.773
$BI2.16_7$ $(b = 0, d = -4)$	3	6.5420e-127	7.15	28.871
EH1 $(m = 3)$ [17]	3	4.9523e-179	7.20	36.646
$BS2.20_8$ $(\beta = 2, \gamma = 0)$	3	1.2925e-172	8.06	**21.806**
BS4.3 $(m = 3)$	3	1.0760e-155	8.21	22.158
BS4.6 $m = 3$	3	1.0760e-155	8.21	22.271
$BS3.8_8$	3	1.0760e-155	8.21	22.411
$BI2.18_8$	3	1.0760e-155	8.21	22.452
ST8S [13]	3	1.6088e-172	8.06	28.816
BS4.2 $(m = 4, b = 0, d = -4)$	3	1.6198e-172	8.06	29.353
BS4.1 $(m = 4, b = 0, d = -4)$	3	1.6198e-172	8.06	29.376
NLM_8 [11]	3	1.3899e-208	8.15	105.124
$ZMO1_8$ [10]	3	2.0820e-182	8.18	107.902
$ZMO2_8$ [10]	3	1.3951e-208	8.15	150.057

7. Basins of Attraction of Methods

The basins of attraction of a method provide crucial insights into its stability and dynamical behavior. In order to investigate the basins of attraction associated with the proposed method, we consider the following nonlinear system [16]:

$$f_2(x_1, x_2) = \begin{cases} p_1(x) = x_1x_2 + x_1 - x_2 - 1, \\ p_2(x) = x_1x_2 - x_1 + x_2 - 1, \end{cases} \tag{7.1}$$

whose roots are $(1, 1)$, and $(-1, -1)$. The two-dimensional quadratic polynomial system has all of its roots contained within the square domain $[-2, 2] \times [-2, 2]$. To generate the basins of attraction associated with the solutions of the considered system of nonlinear equations, we defined a square domain $[-2, 2] \times [-2, 2]$ consisting of 401×401 grid points that encompass all the solution of the example. Each iterative method was applied starting from every point within this domain. A distinct color was assigned to each

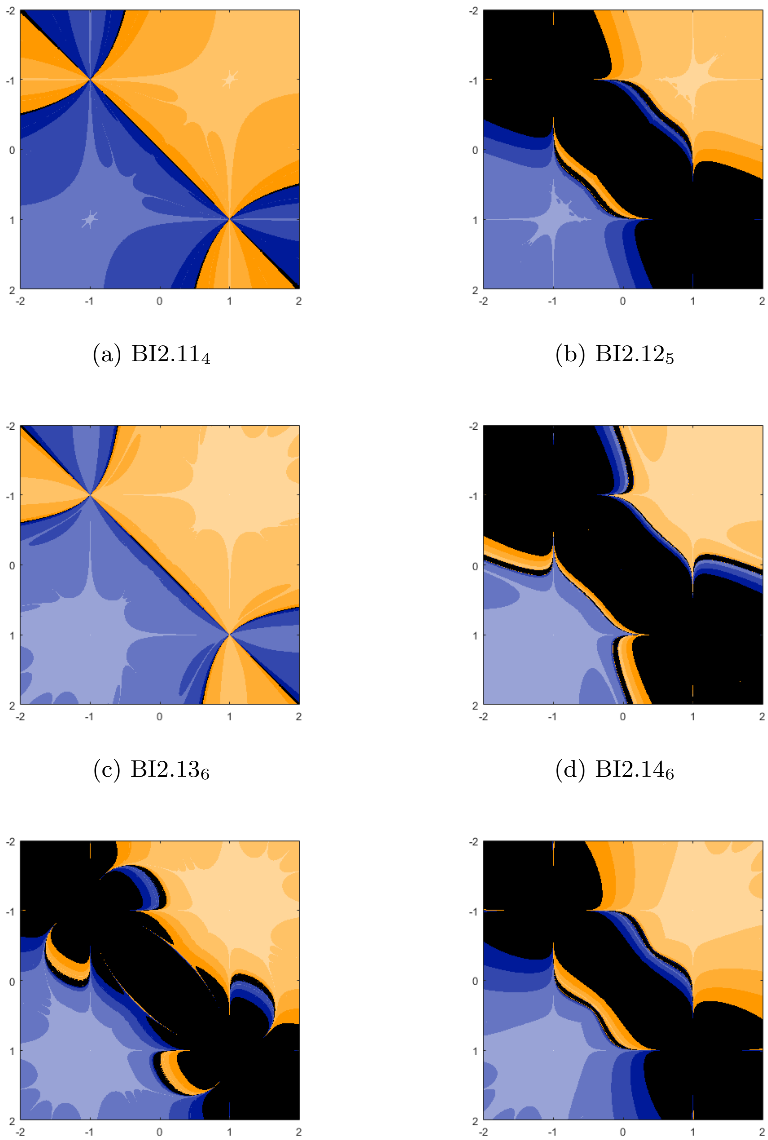
Table 4: Comparison of numerical results for Example 3

Method	Iter	$\|x_{k+1} - x_k\|$	ACOC	CPU Time
BI2.11$_4$ $(b = 0, d = -4)$	5	1.9413e-293	4.00	**20.189**
BS3.1$_4$ $(b = 0, d = -4)$	5	1.9413e-293	4.00	21.393
BS4.1 $(m = 2, b = 0, d = -4)$	5	1.9413e-293	4.00	26.159
BS4.2 $(m = 2, b = 0, d = -4)$	5	1.9413e-293	4.00	25.630
M2$_4$ [9]	5	8.2249e-289	4.00	3377.724
BS3.2$_5$	4	7.5455e-128	5.00	**18.891**
BI2.12$_5$	4	7.5455e-128	5.00	19.149
T5S$_5$ [12]	4	1.2124e-143	5.00	21.456
EH1 $(m = 2)$ [17]	4	1.6491e-134	5.00	22.173
M1$_5$ [8]	4	3.4628e-144	5.00	22.415
BI2.13$_6$ $(b = \bar{b} = 0, d = \bar{d} = -4)$	4	4.4791e-225	6.00	**21.413**
EH6S$_6$ [12]	4	3.350e-487	6.00	23.325
BI2.14$_6$	4	1.8231e-232	6.00	23.554
BI2.15$_6$	4	2.6152e-227	6.00	23.627
BS3.5$_6$	4	2.6152e-227	6.00	24.963
BS3.4$_6$	4	1.8231e-232	6.00	25.197
BS3.3$_6$ $(b = \bar{b} = 0, d = \bar{d} = -4)$	4	4.4791e-225	6.00	26.152
BS4.2$(m = 3, b = 0, d = -4)$	4	4.4791e-225	6.00	29.858
BS4.1 $(m = 3, b = 0, d = -4)$	4	4.4791e-225	6.00	30.648
M2$_6$ [9]	4	4.6836e-226	6.00	2746.544
BS2.19$_7$ $(\beta = 2)$	4	4.9301e-346	7.00	**22.020**
BS3.6$_7$ $(b = 0, d = -4)$	4	1.5479e-329	7.00	25.295
BS3.7$_7$ $(\bar{b} = 0, \bar{d} = -4)$	4	5.2749e-336	7.00	25.430
BI2.17$_7$ $(\bar{b} = 0, \bar{d} = -4)$	4	5.2749e-336	7.00	25.471
BI2.16$_7$ $(b = 0, d = -4)$	4	1.5479e-329	7.00	25.811
EH1 $(m = 3)$ [17]	4	4.1083e-392	7.00	33.642
BS2.20$_8$ $(\beta = 2, \gamma = 0)$	4	1.4810e-487	8.00	**22.260**
BS4.3 $m = 3$	4	3.5215e-469	8.00	22.638
BS4.6 $(m = 3)$	4	3.5215e-469	8.00	22.732
BI2.18$_8$	4	3.5215e-469	8.00	25.391
BS3.8$_8$	4	3.5215e-469	8.00	25.441
ST8S $(m = 4)$ [13]	4	7.1202e-255	8.00	29.195
BS4.2 $(m = 4, b = 0, d = -4)$	4	4.3616e-514	8.00	39.563
BS4.1 $(m = 4, b = 0, d = -4)$	4	4.3616e-514	8.00	39.710
ZMO2$_8$ [10]	4	4.3986e-570	8.00	146.222
ZMO1$_8$ [10]	4	9.1856e-352	8.00	2815.144
NLM$_8$ [11]	4	6.4267e-554	8.00	5427.432

point according to the root to which the iteration, starting from that point, converged. If a method failed to converge within 40 iterations, the corresponding initial point was classified as divergent and represented in black. The basins of attraction for the eighteen methods are presented in Figures 2–4. From Figures 2–4, it can be observed that the methods BI2.11$_4$ and BI2.13$_6$ exhibit larger basins of attraction compared to the other considered methods, followed closely by BS3.1$_4$ and BS3.3$_4$.

8. Conclusions

We have developed an iterative method with scalar and vector coefficients, exhibiting convergence orders $(4 \leq \rho \leq 8)$ for solving nonlinear systems and have further extended it to m-step formulations. The numerical experiments confirm and validate the theoretical results. Overall, the numerical and graphical analyses demonstrate that the with vector and scalar coefficients schemes achieve superior computational efficiency



(a) BI2.11$_4$

(b) BI2.12$_5$

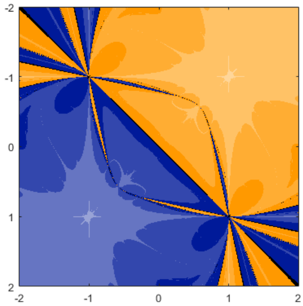
(c) BI2.13$_6$

(d) BI2.14$_6$

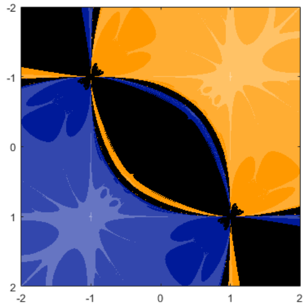
(e) BI2.15$_6$

(f) BI2.16$_7$

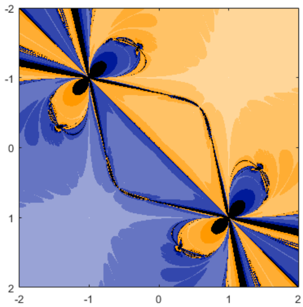
Figure 2: Basins of attraction for the roots of the system $f_1(x_1, x_2) = 0$.



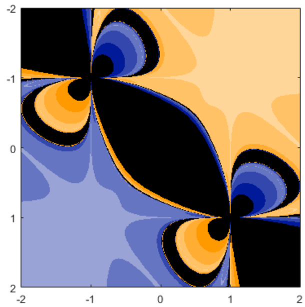
(a) $\mathrm{BS3.1}_4$



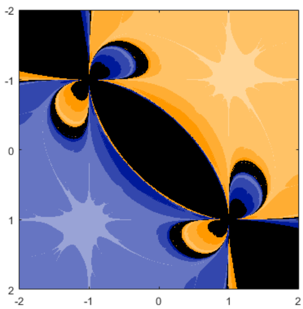
(b) $\mathrm{BI3.2}_5$



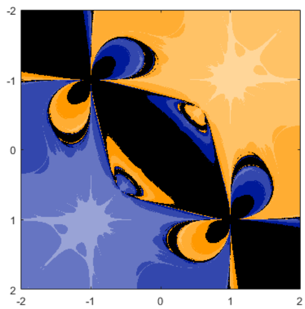
(c) $\mathrm{BI3.3}_6$



(d) $\mathrm{BS3.4}_6$

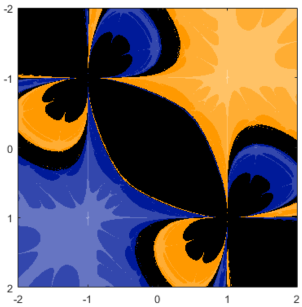


(e) $\mathrm{BI3.5}_6$

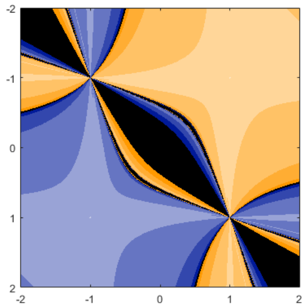


(f) $\mathrm{BS3.6}_7$

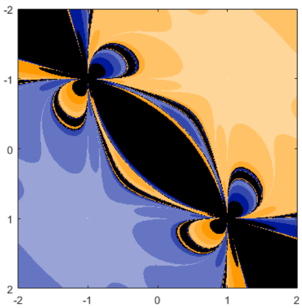
Figure 3: Basins of attraction for the roots of the system $f_1(x_1, x_2) = 0$.



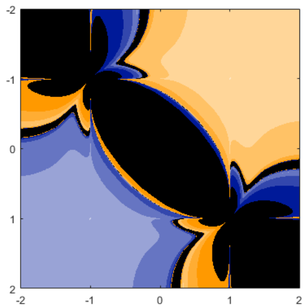
(a) BS3.8_8



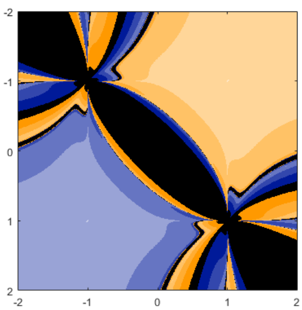
(b) M1_5



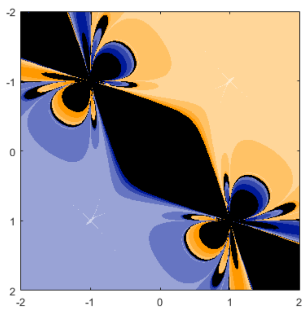
(c) T5S_5



(d) ZMO1_8



(e) ZMO2_8



(f) ST8S

Figure 4: Basins of attraction for the roots of the system $f_1(x_1, x_2) = 0$.

Table 5: Comparison of numerical results for Example 4

Method	ACOC	CPU Time		
		$n = 500$	$n = 1000$	$n = 1500$
$BS3.1_4$ $(b = 0, d = -4)$	4.00	**19.480**	**49.464**	**84.635**
$BI2.11_4$ $(b = 0, d = -4)$	4.00	20.295	51.968	85.701
BS4.2 $(m = 2, b = 0, d = -4)$	4.00	23.409	58.230	96.906
BS4.1 $(m = 2, b = 0, d = -4)$	4.00	23.666	59.200	96.999
$M2_4$ [9]	4.00	363.249	2744.496	9130.727
$BS3.2_5$	5.00	**23.311**	**57.550**	**96.457**
$BI2.12_5$	5.00	23.989	59.480	97.152
$T5S_5$ [12]	5.00	26.478	62.438	110.843
$M1_5$ [8]	5.00	29.129	68.936	119.868
EH1 $(m = 2)$ [17]	5.00	29.858	70.925	121.422
$BI2.13_6$ $(b = \bar{b} = 0, d = \bar{d} = 0)$	6.00	**20.955**	**55.799**	**88.963**
BS4.2$(m = 3, b = 0, d = -4)$	6.00	23.699	59.244	97.515
BS4.1 $(m = 3, b = 0, d = -4)$	6.00	23.832	60.366	98.072
$BS3.3_6$ $(b = \bar{b} = 0, d = \bar{d} = 0)$	6.00	23.866	60.652	98.132
$EH6S_6$ [12]	6.00	26.655	62.489	113.544
$BS3.5_6$	6.00	29.002	71.754	121.225
$BS3.4_6$	6.00	29.174	71.288	121.230
$BI2.14_6$	6.00	29.253	73.705	121.255
$BI2.15_6$	6.00	29.297	73.386	121.509
$M2_6$ [9]	6.00	279.536	1957.277	6371.464
$BS2.19_7$ $(\beta = 2)$	7.00	**20.549**	**53.137**	**88.277**
$BS3.6_7$ $(b = 0, d = -4)$	7.00	23.403	58.958	97.062
$BI2.17_7$ $(\bar{b} = 0, \bar{d} = -4)$	7.00	24.000	61.470	98.246
$BS3.7_7$ $(\bar{b} = 0, \bar{d} = -4)$	7.00	23.643	59.245	98.857
$BI2.16_7$ $(b = 0, d = -4)$	7.00	25.480	66.750	108.577
EH1 $(m = 3)$ [17]	7.00	35.678	83.599	146.250
$BS2.20_8$ $(\beta = 2, \gamma = 0)$	8.00	**20.574**	**53.801**	**88.538**
$BS3.8_8$	8.00	23.461	58.971	97.136
BS4.3 $(m = 3)$	8.00	23.786	67.330	98.161
BS4.6 $(m = 3)$	8.00	23.741	56.267	98.430
$BI2.18_8$	8.00	24.036	60.804	98.7431
ST8S [13]	8.00	29.236	67.449	117.377
BS4.1 $(m = 4, b = 0, d = -4)$	8.00	28.512	73.794	118.267
BS4.2 $(m = 4, b = 0, d = -4)$	8.00	28.283	59.200	118.308
$ZMO2_8$ [10]	8.00	50.437	151.333	322.643
$ZMO1_8$ [10]	8.00	63.861	187.963	379.125
NLM_8 [11]	8.00	71.104	272.203	689.362

and convergence performance across all tested examples. Specifically, the $BI2.13_6$ and $BI3.3_6$ methods outperform other fourth-, fifth-, and sixth-order schemes, while the ST8S [13], $BI2.18_8$, and $BS3.8_8$ methods show the best performance among the seventh- and eighth-order ones. Furthermore, basin of attraction plots confirm that $BI2.11_4$ and $BI2.13_6$ possess wider convergence regions than the remaining methods. In summary, the $BI2.11_4$, $BI2.13_6$, $BI3.3_6$ schemes consistently demonstrate the lowest computational times, confirming their overall superiority in both efficiency and stability.

Acknowledgement. The authors wish to thank the editor and the anonymous referees for their valuable suggestions and comments, which improved the paper.

References

1. Cordero, A., Rojas-Hiciano, R. V., Torregrosa, J. R., Vassileva, M. P. (2024). A highly efficient class of optimal fourth-order methods for solving nonlinear systems. *Numerical Algorithms*, 95, 1879–1904. https://doi.org/10.1007/s11075-023-01631-9
2. Zhanlav, T., Otgondorj, K. (2024). High efficient iterative methods with scalar parameter coefficients for systems of nonlinear equations. *Journal of Mathematical Sciences*, 279(4), 866–875. https://doi.org/10.1007/s10958-024-07066-4
3. Zhanlav, T., Otgondorj, K. (2024). Development and adaptation of higher-order iterative methods in $\mathbb{R}^n$ with specific rules. *Discrete and Continuous Models and Applied Computational Science*, 32(4), 425–444. https://doi.org/10.22363/2658-4670-2024-32-4-425-444
4. Zhanlav, T., Chun, C., Otgondorj, K., Ulziibayar, V. (2020). High–order iterations for systems of nonlinear equations. *International Journal of Computer Mathematics*, 97, 1704–1724. https://doi.org/10.1080/00207160.2019.1652739
5. Zhanlav, T., Otgondorj, K. (2025). Optimal eighth-order three-step iterative methods for solving systems of nonlinear equations. *Discrete and Continuous Models and Applied Computational Science*, 33(4) 389–403. https://doi.org/10.22363/2658-4670-2025-33-4-389-403
6. Zhanlav, T., Otgondorj, K., Enkhbayar, K. (2025). Extensions of some iterative methods to the multidimensional case. *Interdisciplinary Applications in Engineering Science*. (Manuscript submitted for publication)
7. Singh, H., Sharma, J. R., Kumar, S. (2023). A simple yet efficient two-step fifth-order weighted-Newton method for nonlinear models. *Numerical Algorithms*, 93, 203–225. https://doi.org/10.1007/s11075-022-01412-w
8. Singh, H., Sharma, J. R. (2025). A two-point Newton-like method of optimal fourth-order convergence for systems of nonlinear equations. *Journal of Complexity*, 86, 101907. https://doi.org/10.1016/j.jco.2024.101907
9. Hueso, José L., Martínez, E., Teruel, C. (2015). Convergence, efficiency and dynamics of new fourth and sixth order families of iterative methods for nonlinear systems.*Journal of Computational and Applied Mathematics*, 275, 412–420. https://doi.org/10.1016/j.cam.2014.06.010
10. Zhanlav, T., Mijiddorj, R., Otgondorj, K. (2023). A family of Newton-type methods with seventh and eighth-order of convergence for solving systems of nonlinear equations. *Hacettepe Journal of Mathematics and Statistics*, 54(4), 1006–1021. https://doi.org/10.15672/hujms.1061471
11. Sharma, J. R., Arora, H. (2017). Improved Newton–like methods for solving systems of nonlinear equations. *SeMA Journal*, 74, 147–163 https://doi.org/10.1007/s40324-016-0085-x
12. Cordero, A., Rojas-Hiciano, R. V., Torregrosa, J. R., Vassileva, M. P. (2025). High-level convergence order accelerators of iterative methods for nonlinear problems.*Applied Numerical Mathematics*, 217, 390–411. https://doi.org/10.1016/j.apnum.2025.07.003.
13. Cordero, A., Rojas-Hiciano, R. V., Torregrosa, J. R., Vassileva, M. (2025). High-performance damped Traub-type iterative scheme for nonlinear systems. *Authorea*. http://doi.org/10.22541/au.175568773.38658993/v1.
14. Ostrowski, A. M. (1966). Solutions of equations and systems of equations. *Academic Press*.
15. Traub, J. F. (1982). Iterative methods for the solution of equations. *Chelsea Publishing Company*.
16. Chicharro, F. I., Cordero, A., Garrido, N., Torregrosa, J. R. (2022). On the effect of the multidimensional weight functions on the stability of iterative processes.*Journal of Computational and Applied Mathematics*, 405, 113052. https://doi.org/10.1016/j.cam.2020.113052
17. Erfanifar, R., Hajarian, M. (2024). A new multi-step method for solving nonlinear systems with high efficiency indices. *Numerical Algorithms*, 97, 959–984. https://doi.org/10.1007/s11075-023-01735-2

Tugal Zhanlav
Institute of Mathematics and Digital Technology
Mongolian Academy of Sciences
School of Applied Sciences
Mongolian University of Science and Technology
Ulaanbaatar, Mongolia

tzhanlav@yahoo.com
https://orcid.org/0000-0003-0743-5587

Khuder Otgondorj
School of Applied Sciences
Mongolian University of Science and Technology
Ulaanbaatar, Mongolia

otgondorj@gmail.com
* *Corresponding author*
https://orcid.org/0000-0003-1635-7971

Khangai Enkhbayar
School of Applied Sciences
Mongolian University of Science and Technology
Ulaanbaatar, Mongolia

eegii33@must.edu.mn
https://orcid.org/0000-0002-1259-5502