ISSN: 1392-6292/ eISSN: 1648-3510

VILNIUS TECH
Vilnius Gediminas Technical University

MATHEMATICAL MODELLING and ANALYSIS

2025
Volume 30
Issue 2
Pages 254–276
https://doi.org/10.3846/mma.2025.21097

Development and analysis of an efficient Jacobian-free method for systems of nonlinear equations

Janak Raj Sharma[a] ✉, Harmandeep Singh[b] and Sunil Kumar[c]

[a] *Department of Mathematics, Sant Longowal Institute of Engineering and Technology, Longowal-148106, India*

[b] *Department of Mathematics, Statistics and Physics, Punjab Agricultural University, Ludhiana-141004, India*

[c] *Department of Mathematics, University Centre for Research and Development, Chandigarh University, Mohali-140413, India*

Article History:

- received March 4, 2024
- revised July 20, 2024
- accepted October 4, 2024

Abstract. A multi-step derivative-free iterative technique is developed by extending the well-known Traub-Steffensen iteration for solving the systems of nonlinear equations. Keeping in mind the computational aspects, the general idea to construct the scheme is to utilize the single inverse operator per iteration. In fact, these type of techniques are hardly found in literature. Under the standard assumption, the proposed technique is found to possess the fifth order of convergence. In order to demonstrate the computational complexity, the efficiency index is computed and further compared with the efficiency of existing methods of similar nature. The complexity analysis suggests that the developed method is computationally more efficient than their existing counterparts. Furthermore, the performance of method is examined numerically through locating the solutions to a variety of systems of nonlinear equations. Numerical results regarding accuracy, convergence behavior and elapsed CPU time confirm the efficient behavior of the proposed technique.

Keywords: systems of nonlinear equations; Traub-Steffensen method; computational efficiency; convergence analysis.

AMS Subject Classification: 65H10; 47J25; 41A25.

✉ Corresponding author. E-mail: jrshira@yahoo.co.in

1 Introduction

Approximating the solutions in numerical form for the nonlinear equations is the most investigated topic in the field of numerical analysis. Most of the mathematical models or equations, which describe the physical or real-world phenomena, are inherently nonlinear in nature. In general, the closed or analytical form of solution of these equations is not feasible to obtain. The iterative methods, on the other hand, provide the solution in numerical form with the desired precision. For solving the uni-variate equation, $g(t) = 0$, where $g : D \subseteq \mathbb{R} \to \mathbb{R}$ is a nonlinear function, the quadratically convergent Newton method [17] is the most basic approach, which is described as follows:

$$t^{(k+1)} = t^{(k)} - \frac{g(t^{(k)})}{g'(t^{(k)})}, \quad k = 0, 1, 2, \dots,$$

where $t^{(0)}$ is the initial approximation in sufficient proximity of the solution t^*. In the cases where the evaluation of derivative is not feasible, the Traub-Steffensen technique [3, 17] is applicable wherein the derivative $g'(t)$ at any $t \in D$ is to be approximated by the divided difference $[u, t; g] = \frac{g(u)-g(t)}{u-t}$, where $u = t + \beta g(t)$, $\beta \in \mathbb{R}\backslash\{0\}$. Therefore, the Traub-Steffensen scheme is represented as

$$t^{(k+1)} = t^{(k)} - \frac{g(t^{(k)})}{[u^{(k)}, t^{(k)}; g]}. \tag{1.1}$$

In particular for $\beta = 1$, the scheme (1.1) reduces to Steffensen method [1,17].

In order to generalize the iterative schemes to multidimensional case $G : D \subseteq \mathbb{R}^m \to \mathbb{R}^m$, where $G(t) = (g_1(t), \dots, g_m(t))^T$, $t = (t_1, \dots, t_m)^T$ and $g_i : D \to \mathbb{R}$ are nonlinear functions, the generalized Newton scheme [16, 19] is represented as,

$$t^{(k+1)} = t^{(k)} - G'(t^{(k)})^{-1} G(t^{(k)}), \tag{1.2}$$

where $G' : D \to \mathcal{L}(\mathbb{R}^m)$ is a linear operator on $\mathbb{R}^m$, particularly known as the Jacobian matrix [12,16]. Furthermore, the generalized Traub-Steffensen [10,11] scheme is represented as

$$t^{(k+1)} = t^{(k)} - [u^{(k)}, t^{(k)}; G]^{-1} G(t^{(k)}), \tag{1.3}$$

where $u^{(k)} = t^{(k)} + \beta G(t^{(k)})$. Here, $[\cdot, \cdot; G] : D \times D \to \mathcal{L}(\mathbb{R}^m)$ is the generalized divided difference operator [1,12] defined as $[u, v; G](u-v) = G(u)-G(v)$, where $u, v \in \mathbb{R}^m$. However, for the computational purposes, the divided difference operator $[t, y; G]$ can be considered as the approximation of Jacobian matrix $G'(t)$, by defining the divided difference matrix (see [10, 17]) with elements as given below,

$$[t, y\,; G]_{ij} = \frac{g_i(t_1, ..., t_j, y_{j+1}, ..., y_m) - g_i(t_1, ..., t_{j-1}, y_j, ..., y_m)}{t_j - y_j}, \;\; 1 \leqslant i, j \leqslant m,$$

where $t = (t_1, ..., t_m)^T$ and $y = (y_1, ..., y_m)^T$.

Due to the fact that scheme (1.3) preserves quadratic convergence without the need of any derivatives, it significantly improves upon Newton's technique (1.2). In contrast to the case of higher-order methods with derivative evaluations, the construction of derivative-free schemes is a challenging task, in particular for solving nonlinear systems. As already pointed out that the Jacobian matrix is approximated by the divided difference matrix, it might be possible that the derivative-free schemes are not as precise as the schemes requiring derivative evaluation. Thus, it is imperative to design a Jacobian-free scheme in such a manner that it turns out to be efficient and must hold superior convergence properties.

Higher-order methods are very important in many scientific applications which need high precision in their computations [9, 13]. For example, He and Ding [6] demonstrated the use of arbitrary precision computations to improve the results obtained in climate simulations, whereas Zhang and Huang [20] used high-precision calculations to solve interpolation problems in astronomy. The precision required necessitates a large number of iterations if one uses a low-order method. Results of the numerical experiments in [6] and [20] show that the high-order methods associated with multiprecision floating point arithmetic are very useful, as they yield a clear reduction in iterations.

We present some of the higher-order derivative-free methods that are available in the literature. For convenience, let us denote these methods as $\mathrm{M}_{p,i}$, where p is the convergence order, and i is the counting index for the method having p-th order of convergence. Extending the Traub-Steffensen method to a two-step method involving the additional inverse operators, Wang-Zhang [18] developed the following fourth-order method:

$$\begin{aligned} y^{(k)} &= t^{(k)} - [u^{(k)}, t^{(k)}; G]^{-1} G(t^{(k)}), \\ t^{(k+1)} &= M_{4,1}(t^{(k)}, y^{(k)}) = y^{(k)} - [y^{(k)}, t^{(k)}; G]^{-1} \big([y^{(k)}, t^{(k)}; G] - [y^{(k)}, u^{(k)}; G] \\ &\qquad + [u^{(k)}, t^{(k)}; G]\big) [y^{(k)}, t^{(k)}; G]^{-1} G(y^{(k)}), \end{aligned}$$

where $u^{(k)} = t^{(k)} + \beta G(t^{(k)})$ and $\beta \in \mathbb{R}\backslash\{0\}$. In terms of computational cost, the method $\mathrm{M}_{4,1}$ requires three evaluations each of function G and divided difference operator $[\cdot, \cdot; G]$, and two of inverse operator $[\cdot, \cdot; G]^{-1}$.

In the quest to generate higher-order methods, Grau et al. [5] developed the two-step fourth-order and three-step sixth-order schemes as given below,

$$\begin{aligned} y^{(k)} &= t^{(k)} - [u^{(k)}, v^{(k)}; G]^{-1} G(t^{(k)}), \\ z^{(k)} &= M_{4,2}(t^{(k)}, y^{(k)}) = y^{(k)} - \big(2[y^{(k)}, t^{(k)}; G] - [u^{(k)}, v^{(k)}; G]\big)^{-1} G(y^{(k)}), \\ t^{(k+1)} &= M_{6,1}(t^{(k)}, y^{(k)}, z^{(k)}) = z^{(k)} - \big(2[y^{(k)}, t^{(k)}; G] - [u^{(k)}, v^{(k)}; G]\big)^{-1} G(z^{(k)}), \end{aligned} \tag{1.4}$$

where $u^{(k)} = t^{(k)} + G(t^{(k)})$ and $v^{(k)} = t^{(k)} - G(t^{(k)})$. Note that, the schemes $\mathrm{M}_{4,2}$ and $\mathrm{M}_{6,1}$ have been developed in [5] by generalizing the fourth-order and sixth-order schemes, respectively, that are presented earlier by Grau and Díaz [4] for uni-variate case. It is clear that the scheme (1.4) utilizes evaluations of five functions and two evaluations each of divided difference and inverse operator.

In similar fashion, Sharma and Arora [14] constructed the fourth-order as well as sixth-order schemes as

$$
\begin{aligned}
y^{(k)} &= t^{(k)} - [u^{(k)}, v^{(k)}; G]^{-1} G(t^{(k)}), \\
z^{(k)} &= M_{4,3}(t^{(k)}, y^{(k)}) = y^{(k)} - \big(3I - 2[u^{(k)}, v^{(k)}; G]^{-1}[y^{(k)}, t^{(k)}; G]\big) \\
&\quad \times [u^{(k)}, v^{(k)}; G]^{-1} G(y^{(k)}), \\
t^{(k+1)} &= M_{6,2}(t^{(k)}, y^{(k)}, z^{(k)}) = z^{(k)} - \big(3I - 2[u^{(k)}, v^{(k)}; G]^{-1}[y^{(k)}, t^{(k)}; G]\big) \\
&\quad \times [u^{(k)}, v^{(k)}; G]^{-1} G(z^{(k)}),
\end{aligned} \tag{1.5}
$$

where $u^{(k)} = t^{(k)} + \beta G(t^{(k)})$ and $v^{(k)} = t^{(k)} - \beta G(t^{(k)})$. Notice that the methods $\mathrm{M}_{4,3}$ and $\mathrm{M}_{6,2}$ only use single inverse operator as compared to the two inverse operators in the methods $\mathrm{M}_{4,2}$ and $\mathrm{M}_{6,1}$.

Singh [15] proposed a fifth-order method which is given as,

$$
\begin{aligned}
y^{(k)} &= t^{(k)} - [u^{(k)}, t^{(k)}; G]^{-1} G(t^{(k)}), \\
z^{(k)} &= y^{(k)} - [w^{(k)}, y^{(k)}; G]^{-1} G(y^{(k)}), \\
t^{(k+1)} &= z^{(k)} - (2[w^{(k)}, y^{(k)}; G]^{-1} - [u^{(k)}, t^{(k)}; G]^{-1}) G(z^{(k)}),
\end{aligned} \tag{1.6}
$$

where $u^{(k)} = t^{(k)} + G(t^{(k)})$ and $w^{(k)} = y^{(k)} + G(y^{(k)})$. This iterative scheme utilizes five functions and two each of divided difference and inverse operator. Note that both, the sixth (1.4) and the fifth (1.6) order methods, use same number of operators, i.e., two divided differences and two inversions. The sixth order method (1.4), however, is computationally more efficient due to its higher order. On the other hand, the method (1.5) although uses only one inverse operator, it is still not as efficient due to using two divided differences.

Construction of multi-step higher-order methods using additional evaluations of functions or inverse operators significantly increases the computational cost of iterative procedure. Indeed, the methods involving single inverse operators can be considered the best choice among Newton-type or Traub-Steffensen-type methods. The reason being that the numerical operational cost per iteration would be minimal in solving the system of equations using LU matrix decomposition process. Additionally, it is more challenging to generate the higher-order derivative-free methods that efficiently solve the systems of nonlinear equations. In this regard, the primary purpose of this paper is to design a multi-step derivative-free method that utilizes single inverse operator per iteration. Indeed, it would lead to the development of a computationally efficient technique. Let us note that the iterative procedures can be analyzed for their computational efficiency through computing the efficiency index [5], which is based on some parameters that are specifically introduced to determine the computational cost per iteration.

Keeping in mind the above discussion, here we propose a derivative-free method with fifth-order of convergence for solving nonlinear equations. Novelty of the proposed method is that it requires only a single evaluation of divided difference and inverse operator. The derivative-free iterative schemes with such characteristics are hardly found in the literature.

We summarize the rest of the paper. The derivative-free iterative scheme is presented as well as analyzed in the Section 2. Computational efficiency of the developed method is established and further compared with the existing methods in Section 3. Furthermore, in Section 4, the numerical performance is analyzed by solving the various systems of nonlinear equations. Finally, the concluding remarks are presented in Section 5.

2 Development of method

The prime focus here is to develop a multi-step generalized Traub-Steffensen-type method for solving systems of nonlinear equations. To achieve this, we initially develop a multi-step derivative-free scheme for solving only uni-variate functions. This development will be further generalized for the case of multi-dimensional systems in view of the methodology discussed in previous section. In particular for solving the scalar equation $g(t) = 0$, consider the scheme:

$$\begin{aligned} w^{(k)} &= t^{(k)} - \frac{g(t^{(k)})}{[u^{(k)}, v^{(k)}; g]}, \ y^{(k)} = w^{(k)} - \frac{g(w^{(k)})}{[u^{(k)}, v^{(k)}; g]}, \\ z^{(k)} &= w^{(k)} + \frac{g(w^{(k)})}{[u^{(k)}, v^{(k)}; g]}, \\ t^{(k+1)} &= w^{(k)} - \frac{1}{[u^{(k)}, v^{(k)}; g]}\big(a_1 g(w^{(k)}) + a_2 g(y^{(k)}) + a_3 g(z^{(k)})\big), \end{aligned} \tag{2.1}$$

where $u^{(k)} = t^{(k)} + \beta g(t^{(k)})$, $v^{(k)} = t^{(k)} - \beta g(t^{(k)})$, $\beta = \mathbb{R}\backslash\{0\}$ and a_1, a_2, a_3 are some parameters to be determined. Now, the convergence of presented method is analyzed in the following theorem:

Theorem 1. *Let $g : D \subset \mathbb{R} \to \mathbb{R}$ be a sufficiently differentiable function with respect to its zero t^* in neighborhood D. The series of iterates produced by method (2.1) for $t^{(0)} \in D$ converges to t^* with order of convergence five if the initial approximation, $t^{(0)}$, is close enough to t^*, provided $a_1 = -3$, $a_2 = 3$ and $a_3 = 2$.*

Proof. Let $\sigma^{(k)} = t^{(k)} - t^*$, $\sigma_u^{(k)} = u^{(k)} - t^*$, $\sigma_v^{(k)} = v^{(k)} - t^*$, $\sigma_w^{(k)} = w^{(k)} - t^*$, $\sigma_y^{(k)} = y^{(k)} - t^*$ and $\sigma_z^{(k)} = z^{(k)} - t^*$ be the errors in the k-th iteration of method (2.1). Then, on account of the fact that $g(t^*) = 0$ and $g'(t^*) \neq 0$, the Taylor series expansions of $g(t^{(k)})$, $g(u^{(k)})$, $g(v^{(k)})$, $g(w^{(k)})$, $g(y^{(k)})$ and $g(z^{(k)})$ about t^* gives, respectively

$$g(t^{(k)}) = g'(t^*)\big[\sigma^{(k)} + A_2\sigma^{(k)^2} + A_3\sigma^{(k)^3} + A_4\sigma^{(k)^4} + A_5\sigma^{(k)^5} + \ldots\big], \tag{2.2}$$

$$g(u^{(k)}) = g'(t^*)\big[\sigma_u^{(k)} + A_2\sigma_u^{(k)^2} + A_3\sigma_u^{(k)^3} + A_4\sigma_u^{(k)^4} + A_5\sigma_u^{(k)^5} + \ldots\big], \tag{2.3}$$

$$g(v^{(k)}) = g'(t^*)\big[\sigma_v^{(k)} + A_2\sigma_v^{(k)^2} + A_3\sigma_v^{(k)^3} + A_4\sigma_v^{(k)^4} + A_5\sigma_v^{(k)^5} + \ldots\big], \tag{2.4}$$

$$g(w^{(k)}) = g'(t^*)\big[\sigma_w^{(k)} + A_2\sigma_w^{(k)^2} + A_3\sigma_w^{(k)^3} + A_4\sigma_w^{(k)^4} + A_5\sigma_w^{(k)^5} + \ldots\big], \tag{2.5}$$

$$g(y^{(k)}) = g'(t^*)\big[\sigma_y^{(k)} + A_2\sigma_y^{(k)^2} + A_3\sigma_y^{(k)^3} + A_4\sigma_y^{(k)^4} + A_5\sigma_y^{(k)^5} + \ldots\big], \tag{2.6}$$

$$g(z^{(k)}) = g'(t^*)\big[\sigma_z^{(k)} + A_2\sigma_z^{(k)^2} + A_3\sigma_z^{(k)^3} + A_4\sigma_z^{(k)^4} + A_5\sigma_z^{(k)^5} + \ldots\big], \tag{2.7}$$

where $A_i = (1/i!)g^{(i)}(t^*)/g'(t^*),\ i = 2, 3, 4, \ldots$.

Now, by using Equations (2.2)–(2.7) in the last step of (2.1), we obtain the error equation as

$$\sigma^{(k+1)} = \sigma_w^{(k)} - a_1\sigma_w^{(k)} - a_2\sigma_y^{(k)} - a_3\sigma_z^{(k)} - (a_1\sigma_w^{(k)^2} + a_2\sigma_y^{(k)^2} + a_3\sigma_z^{(k)^2})A_2 + 2\sigma^{(k)}A_2 \times (a_1\sigma_w^{(k)} + a_2\sigma_y^{(k)} + a_3\sigma_z^{(k)} + (a_1\sigma_w^{(k)^2} + a_2\sigma_y^{(k)^2} + a_3\sigma_z^{(k)^2})A_2) - \sigma^{(k)^2}(a_1\sigma_w^{(k)} + a_2\sigma_y^{(k)} + a_3\sigma_z^{(k)} + (a_1\sigma_w^{(k)^2} + a_2\sigma_y^{(k)^2} + a_3\sigma_z^{(k)^2})A_2)(-A_3 \times (3+(\beta g'(t^*))^2)+4A_2^2) - 2\sigma^{(k)^3}(a_1\sigma_w^{(k)} + a_2\sigma_y^{(k)} + a_3\sigma_z^{(k)} + (a_1\sigma_w^{(k)^2} + a_2\sigma_y^{(k)^2} + a_3\sigma_z^{(k)^2})A_2)(-4A_2^3 - 2A_4(1+(\beta g'(t^*))^2) + A_2A_3(6+(\beta g'(t^*))^2)) + \ldots. \tag{2.8}$$

Substitution of (2.2)–(2.4) in the first step of (2.1) yields

$$\sigma_w^{(k)} = A_2\sigma^{(k)^2} + \big(-2A_2^2 + A_3(2+(\beta g'(t^*))^2)\big)\sigma^{(k)^3} + \big(4A_2^3 - A_2A_3(7 + (\beta g'(t^*))^2) + A_4(3+4(\beta g'(t^*))^2)\big)\sigma^{(k)^4} + \big(-8A_2^4 - 2A_2A_4(5 + 2(\beta g'(t^*))^2) + A_2^2A_3(20+3(\beta g'(t^*))^2) - A_3^2(6+3(\beta g'(t^*))^2 + (\beta g'(t^*))^4) + A_5(4+10(\beta g'(t^*))^2 + (\beta g'(t^*))^4)\big)\sigma^{(k)^5} + O(\sigma^{(k)^6}). \tag{2.9}$$

Then, using (2.2)–(2.5) and (2.9) in the second and third steps of (2.1), it is obtained that

$$\sigma_y^{(k)} = 2A_2^2\sigma^{(k)^3} + \big(-4A_2^3 + 2A_2A_3(\beta g'(t^*))^2 + A_4(3+4(\beta g'(t^*))^2)\big)\sigma^{(k)^4} + \big(8A_2^4 - 6A_2A_4 + 2A_3^2(\beta g'(t^*))^2 - A_2^2A_3(6+5(\beta g'(t^*))^2) + A_5(4 + 10(\beta g'(t^*))^2 + (\beta g'(t^*))^4)\big)\sigma^{(k)^5} + O(\sigma^{(k)^6}), \tag{2.10}$$

$$\sigma_z^{(k)} = 2A_2\sigma^{(k)^2} + \big(-6A_2^2 + 2A_3(2+(\beta g'(t^*))^2)\big)\sigma^{(k)^3} + \big(12A_2^3 - 2A_2A_3(7 + 2(\beta g'(t^*))^2) + A_4(3+4(\beta g'(t^*))^2)\big)\sigma^{(k)^4} + \big(-24A_2^4 - 2A_2A_4(7 + 4(\beta g'(t^*))^2) + A_2^2A_3(46+11(\beta g'(t^*))^2) - 2A_3^2(6+4(\beta g'(t^*))^2 + (\beta g'(t^*))^4) + A_5(4+10(\beta g'(t^*))^2 + (\beta g'(t^*))^4)\big)\sigma^{(k)^5} + O(\sigma^{(k)^6}). \tag{2.11}$$

Finally, using (2.9)–(2.11) in (2.8) and upon simplification, we obtain

$$\sigma^{(k+1)} = -(a_1+2a_3-1)A_2\sigma^{(k)^2} + \big(2(2a_1-a_2+5a_3-1)A_2^2 - (a_1+2a_3-1) \times A_3(2+(\beta g'(t^*))^2)\big)\sigma^{(k)^3} + \big((4-13a_1+13a_2-41a_3)A_2^3 - (a_1+2a_3-1) \times A_4(3+4(\beta g'(t^*))^2) + A_2A_3(35a_3 - 7a_2 - 7 - (\beta g'(t^*))^2 - 3a_2(\beta g'(t^*))^2 + 11a_3(\beta g'(t^*))^2 + 2a_1(7+2(\beta g'(t^*))^2))\big)\sigma^{(k)^4} + \big(2(19a_1 - 28a_2 + 76a_3 - 4)A_2^4 - (a_1+2a_3-1)A_5(4+10(\beta g'(t^*))^2+(\beta g'(t^*))^4) + 2A_2A_4(25a_3 - 5 - 2(\beta g'(t^*))^2 + 22a_3(\beta g'(t^*))^2 + 2a_1(5+4(\beta g'(t^*))^2) - a_2(5+6(\beta g'(t^*))^2))$$

$$+ A_2^2A_3(20 - 200a_3 + 3(\beta g'(t^*))^2 - 54a_3(\beta g'(t^*))^2 + 4a_2(16 + 5(\beta g'(t^*))^2)$$
$$- a_1(64 + 15(\beta g'(t^*))^2)) + A_3^2(-6 + 30a_3 - 3(\beta g'(t^*))^2 + 21a_3(\beta g'(t^*))^2$$
$$- (\beta g'(t^*))^4 + 5a_3(\beta g'(t^*))^4 + 2a_1(6 + 4(\beta g'(t^*))^2 + (\beta g'(t^*))^4)$$
$$- a_2(6 + 5(\beta g'(t^*))^2 + (\beta g'(t^*))^4))\sigma^{(k)^5} + O(\sigma^{(k)^6}). \quad (2.12)$$

It would be enough to equate the coefficients of $\sigma^{(k)^2}$, $\sigma^{(k)^3}$ and $\sigma^{(k)^4}$ to zero in order to find the parameters a_1, a_2 and a_3. As a result, we have

$$a_1+2a_3-1 = 0, \; 2a_1-a_2+5a_3-1 = 0, \; -13a_1 + 13a_2 - 41a_3 + 4 \quad = 0. \quad (2.13)$$

Solving the Equations (2.13), we get $a_1 = -3$, $a_2 = 3$ and $a_3 = 2$. Substituting these values in (2.11), the error equation (2.12) becomes,

$$\sigma^{(k+1)} = \big(14A_2^4 + 4A_2^2A_3\big)\sigma^{(k)^5} + O(\sigma^{(k)^6}). \quad (2.14)$$

The proof is completed and error equation (2.14) demonstrates that the local order of convergence is five for the method (2.1). □

2.1 Generalized method

When an iterative method achieves high computing speed at low computational cost, it is considered computationally efficient in numerical analysis. As covered in the introduction, the evaluation of functions, divided difference operators, and inverse operators are the primary computationally expensive components of derivative-free methods. The evaluation of an inverse operator stands out as a significant barrier in the development of an effective iterative scheme due to its high computing cost. It is therefore important to use as few of these inversions as possible while developing a numerical method.

Let us take the problem of solving $G(t) = 0$ by an iterative method that is based on the construction of scheme (2.1). Now, write the corresponding formula $a_1 = -3$, $a_2 = 3$ and $a_3 = 2$ for the system of equations as

$$\begin{aligned} w^{(k)} &= t^{(k)} - [u^{(k)}, v^{(k)}; G]^{-1}G(t^{(k)}), \\ y^{(k)} &= w^{(k)} - [u^{(k)}, v^{(k)}; G]^{-1}G(w^{(k)}), \\ z^{(k)} &= w^{(k)} + [u^{(k)}, v^{(k)}; G]^{-1}G(w^{(k)}), \\ t^{(k+1)} &= w^{(k)} - [u^{(k)}, v^{(k)}; G]^{-1}(-3G(w^{(k)}) + 3G(y^{(k)}) + 2G(z^{(k)})), \end{aligned} \quad (2.15)$$

where $u^{(k)} = t^{(k)} + \beta G(t^{(k)})$, $v^{(k)} = t^{(k)} - \beta G(t^{(k)})$, and $[u^{(k)}, v^{(k)}; G]$ is the divided difference operator that is already defined in previous section. The iterative scheme (2.15) defines a four-step family of derivative-free methods. It is easy to see that this scheme utilizes six functions and single evaluation each of divided difference operator and inverse operator per iteration.

In order to examine convergence properties of the scheme (2.15), the following approximation of divided difference operators is considered (see [5, 7]):

$$[t + h, t; G] = \int_0^1 G'(t + xh)dx, \qquad \forall\, t, h \in \mathbb{R}^m.$$

Expanding $G'(t+xh)$ using Taylor series at the point t and then integrating,

$$[t+h,t;G]=\int_0^1 G'(t+xh)\,dx = G'(t)+\frac{1}{2}G''(t)h+\frac{1}{6}G'''(t)h^2+\dots, \quad (2.16)$$

where $h^i=(h,\overset{i-times}{\cdots},h)\in\mathbb{R}^m\times\overset{i-times}{\cdots}\times\mathbb{R}^m$. Let us note that, for each $k\in\mathbb{N}$ and $t\in\mathbb{R}^m$, $G^{(k)}(t):\mathbb{R}^m\times\overset{k-times}{\cdots}\times\mathbb{R}^m\to\mathbb{R}^m$ is a k-linear function [12] such that $G^{(k)}(t)h^k\in\mathbb{R}^m$ and $G^{(k)}(t)h^{k-1}\in\mathcal{L}(\mathbb{R}^m)$. In view of this, the operations on the right hand side of Equation (2.16) are well-defined.
Let $\sigma^{(k)}=t^{(k)}-t^*$ be the error in the k-th iteration of method (2.15). Assuming that $\Gamma=G'(t^*)^{-1}$ exists, then developing $G(t^{(k)})$ and its derivatives in a neighborhood of t^*, we have

$$G(t^{(k)})=G'(t^*)\big(\sigma^{(k)}+A_2{\sigma^{(k)}}^2+A_3{\sigma^{(k)}}^3+A_4{\sigma^{(k)}}^4+O({\sigma^{(k)}}^5)\big), \quad (2.17)$$

$$G'(t^{(k)})=G'(t^*)\big(I+2A_2\sigma^{(k)}+3A_3{\sigma^{(k)}}^2+4A_4{\sigma^{(k)}}^3+O({\sigma^{(k)}}^4)\big), \quad (2.18)$$

$$G''(t^{(k)})=G'(t^*)\big(2A_2+6A_3\sigma^{(k)}+12A_4{\sigma^{(k)}}^2+O({\sigma^{(k)}}^3)\big), \quad (2.19)$$

$$G'''(t^{(k)})=G'(t^*)\big(6A_3+24A_4\sigma^{(k)}+O({\sigma^{(k)}}^2)\big), \quad (2.20)$$

where I is an identity operator, $A_i=\frac{1}{i!}\Gamma G^{(i)}(t^*)$, $i=2,3,\dots$ and ${\sigma^{(k)}}^i=(\sigma^{(k)},\overset{i-times}{\cdots},\sigma^{(k)})$, $\sigma^{(k)}\in\mathbb{R}^m$.

With the help of above expressions, we can now analyze the convergence behavior of scheme (2.15) through the following theorem:

Theorem 2. *Let $G:D\subset\mathbb{R}^m\to\mathbb{R}^m$ be a sufficiently differentiable function with its zero t^* in neighborhood D. When the initial approximation $t^{(0)}$ is close enough to t^*, the method* (2.15) *for $t^{(0)}\in D$ produces a sequence of iterates that converges to t^* with order of convergence five.*

Proof. Let $\sigma^{(k)}=t^{(k)}-t^*$, $\sigma_u^{(k)}=u^{(k)}-t^*$, $\sigma_v^{(k)}=v^{(k)}-t^*$, $\sigma_w^{(k)}=w^{(k)}-t^*$, $\sigma_y^{(k)}=y^{(k)}-t^*$ and $\sigma_z^{(k)}=z^{(k)}-t^*$ be the errors in the k-th iteration of method (2.15). Using the expression of (2.17), it is obtained that

$$\sigma_u^{(k)}=\sigma^{(k)}+B\big(\sigma^{(k)}+A_2{\sigma^{(k)}}^2+A_3{\sigma^{(k)}}^3+A_4{\sigma^{(k)}}^4+A_5{\sigma^{(k)}}^5+O({\sigma^{(k)}}^6)\big), \quad (2.21)$$

$$\sigma_v^{(k)}=\sigma^{(k)}-B\big(\sigma^{(k)}+A_2{\sigma^{(k)}}^2+A_3{\sigma^{(k)}}^3+A_4{\sigma^{(k)}}^4+A_5{\sigma^{(k)}}^5+O({\sigma^{(k)}}^6)\big), \quad (2.22)$$

where $B=\beta G'(t^*)$. Expanding $G(u^{(k)})$ and $G(v^{(k)})$ as Taylor series using Equations (2.21) and (2.22),

$$G(u^{(k)})=G'(t^*)\big[(I+B)\sigma^{(k)}+A_2(I+3B+B^2){\sigma^{(k)}}^2+(A_3(I+4B+3B^2+B^3)$$
$$+2B(I+B)A_2^2){\sigma^{(k)}}^3+(A_4(I+5B+6B^2+4B^3+B^4)+2B(I+B)A_2A_3$$

$$+ 3B(I+B)^2A_3A_2 + B^2A_2^3)\sigma^{(k)^4} + (A_5(I+6B+10B^2+10B^3+5B^4$$
$$+ B^5)+2B(I+B)A_2A_4+3B(I+B)^2A_3^2 + 4B(I+B)^3A_4A_2 + B^2A_2^2A_3$$
$$+ B^2A_2A_3A_2 + 3B^2(I+B)A_3A_2^2)\sigma^{(k)^5} + O(\sigma^{(k)^6})\big], \tag{2.23}$$

$$G(v^{(k)}) = G'(t^*)\big[(I-B)\sigma^{(k)}+A_2(I-3B+B^2)\sigma^{(k)^2}+(A_3(I-4B+3B^2-B^3)$$
$$+ 2B(B-I)A_2^2)\sigma^{(k)^3}+(A_4(I-5B+6B^2-4B^3+B^4) + 2B(B-I)A_2A_3$$
$$- 3B(B-I)^2A_3A_2 + B^2A_2^3)\sigma^{(k)^4} + (A_5(I-6B+10B^2+10B^3+5B^4$$
$$- B^5)+2B(B-I)A_2A_4-3B(B-I)^2A_3^2 + 4B(B-I)^3A_4A_2 + B^2A_2^2A_3$$
$$+ B^2A_2A_3A_2 - 3B^2(B-I)A_3A_2^2)\sigma^{(k)^5} + O(\sigma^{(k)^6})\big]. \tag{2.24}$$

With the help of Equations (2.23) and (2.24), we obtain that

$$[u^{(k)}, v^{(k)}; G]$$
$$= G'(t^*)\big[I + 2A_2\sigma^{(k)} + A_3(3+B^2)\sigma^{(k)^2} + (4A_4 + 4A_4B^2 + 3A_2A_3$$
$$+ 3A_2(-A3 + A_2^2) + 4A_3A_2 + 3B^2A_3A_2 - (4A_3 + A_3B^2 + 2A_2^2)A_2$$
$$- A2^3)\sigma^{(k)^3} + (5A_5 + 10A_5B^2 + A_5B^4 + 3A_2A_4 + 3A_2(-A_4 + A_2A_3$$
$$+ A_3A_2 - A_2^3) + 4A_3^2 + 3B^2A_3^2 + 5A_4A_2 + 12B^2A_4A_2 + (4A_3 + A_3B^2$$
$$+ 2A_2^2)(-A_3 + A_2^2) - (A_4(5I + 4B^2) + 2A_2A_3 + 3(I + B^2)A_3A_2)A_2$$
$$- A_2^2A_3 - A_2A_3A_2 - A_3A_2^2 + 3B^2A_3A_2^2 + A_2^4)\sigma^{(k)^4} + O(\sigma^{(k)^5})\big], \tag{2.25}$$

which ultimately leads to the following,

$$[u^{(k)}, v^{(k)}; G]^{-1}$$
$$= \big[I - 2A_2\sigma^{(k)} - ((3I + B^2)A_3 - 4A_2^2)\sigma^{(k)^2} - (4(A_4 + A_4B^2 + 2A_2^3)$$
$$- (7I + 3B^2)A_2A_3 + (B^2 - 5)A_3A_2)\sigma^{(k)^3} + (5A_5 + 10A_5B^2$$
$$+ A_5B^4 - 2(5I + 6B^2)A_2A_4 - (9I + 4B^2 + B^4)A_3^2 - 6A_4A_2$$
$$+ 4B^2A_4A_2 + 15A_2^2A_3 + 7B^2A_2^2A_3 + 11A_2A_3A_2 - 3B^2A_2A_3A_2$$
$$+ 10A_3A_2^2 + B^2A_3A_2^2 - 16A_2^4)\sigma^{(k)^4} + O(\sigma^{(k)^5})\big]\Gamma, \tag{2.26}$$

where $\Gamma = G'(t^*)^{-1}$. The first step of (2.15), using (2.17) and (2.26), yields

$$\sigma_w^{(k)}=A_2\sigma^{(k)^2}+(A_3(2I+B^2)-2A_2^2)\sigma^{(k)^3}+(3A_4 + 4A_4B^2-(5I+3B^2)A_2A_3$$
$$+ 2(B^2-I)A_3A_2 + 4A_2^3)\sigma^{(k)^4}+(A_5(4I+10B^2+B^4)-4(2I + 3B^2)A_2A_4$$
$$- (6I + 3B^2 + B^4)A_3^2 + (-2I + 8B^2)A_4A_2 + (11I + 7B^2)A_2^2A_3 + (4I$$
$$- 6B^2)A_2A_3A_2 + (5I + 2B^2)A_3A_2^2 - 8A_2^4)\sigma^{(k)^5} + O(\sigma^{(k)^6}). \tag{2.27}$$

Now, developing the series expansion of $G(w^{(k)})$ as

$$G(w^{(k)}) = G'(t^*)\big[A_2\sigma^{(k)^2}+(A_3(2I+B^2)-2A_2^2)\sigma^{(k)^3}+(3A_4+4A_4B^2-(5I+3B^2)A_2A_3+2(B^2-I)A_3A_2+5A_2^3)\sigma^{(k)^4}+(A_5(4I+10B^2+B^4)-4(2I+3B^2)A_2A_4-(6I+3B^2+B^4)A_3^2+(-2I+8B^2)A_4A_2+(13I+8B^2)A_2^2A_3+(6I-5B^2)A_2A_3A_2+(5I+2B^2)A_3A_2^2-12A_2^4)\sigma^{(k)^5}+O(\sigma^{(k)^6})\big]. \quad (2.28)$$

Then, using (2.26)–(2.28) in second and third steps of (2.15), we have

$$\sigma_y^{(k)}=2A_2^2\sigma^{(k)^3}+\big(2A_2(A_3(2I+B^2)-2A_2^2)+(3A_3+A_3B^2-4A_2^2)A2-A_2^3\big)\sigma^{(k)^4}+\big(2A_2(3A_4+4A_4B^2-(5I+3B^2)A_2A_3+2(B^2-I)A_3A_2+5A_2^3)+(3A_3+A_3B^2-4A_2^2)(2A_3+A_3B^2-2A_2^2)+((B^2-5I)A_3A_2-(7+3B^2)A_2A_3+4(A_4+A_4B^2+2A_2^3))A_2-(2I+B^2)A_2^2A_3-(2I+B^2)A_2A_3A_2+4A_2^4\big)\sigma^{(k)^5}+O(\sigma^{(k)^6}), \quad (2.29)$$

$$\sigma_z^{(k)} = 2A_2\sigma^{(k)^2}+2\big(A_3(2I+B^2)-3A_2^2\big)\sigma^{(k)^3}+\big(6A_4+8A_4B^2-2(5I+3B^2)A_2A_3-2A_2(2A_3+A_3B^2-2A_2^2)-4A_3A_2+4B^2A_3A_2-(3A_3+A_3B^2-4A_2^2)A_2+9A_2^3\big)\sigma^{(k)^4}+\big(2A_5(4I+10B^2+B^4)-8(2I+3B^2)A_2A_4-2A_2((3+4B^2)A_4-(5I+3B^2)A_2A_3+2(B^2-I)A_3A_2+5A_2^3)-12A_3^2-6B^2A_3^2-2B^4A_3^2+4(4B^2-I)A_4A_2-(3A_3+A_3B^2-4A_2^2)(2A_3+A_3B^2-2A_2^2)-((B^2-5I)A_3A_2-(7+3B^2)A_2A_3+4(A_4+A_4B^2+2A_2^3))A_2+3(8I+5B^2)A_2^2A_3+(10I-11B^2)A_2A_3A_2+2(5I+2B^2)A_3A_2^2-20A_2^4\big)\sigma^{(k)^5}+O(\sigma^{(k)^6}). \quad (2.30)$$

Now, the Taylor expansions of $G(y^{(k)})$ and $G(z^{(k)})$ are given as

$$G(y^{(k)}) = G'(t^*)\big[2A_2^2\sigma^{(k)^3}+(2(2I+B^2)A_2A_3+(3I+B^2)A_3A_2-9A_2^3)\sigma^{(k)^4}+((6I+8B^2)A_2A_4+(6I+5B^2+B^4)A_3^2+4(I+B^2)A_4A_2-(20I+11B^2)\times A_2^2A_3-13A_2A_3A_2-(11I+B^2)A_3A_2^2+30A_2^4)\sigma^{(k)^5}+O(\sigma^{(k)^6})\big], \quad (2.31)$$

$$G(z^{(k)}) = G'(t^*)\big[2A_2\sigma^{(k)^2}+2(A_3(2I+B^2)-3A_2^2)\sigma^{(k)^3}+(21A_2^3-2(7I+4B^2)A_2A_3+(6+8B^2)A_4+(3d^2k^2-7I)A_3A_2)\sigma^{(k)^4}+(2A_5(4I+10B^2+B^4)-70A_2^4-2(11I+16B^2)A_2A_4-(18I+11B^2+3B^4)A_3^2+4(3B^2-2I)A_4A_2+(50I+29B^2)A_2^2A_3+(29I-8B^2)A_2A_3A_2+(21I+5B^2)A_3A_2^2)\sigma^{(k)^5}+O(\sigma^{(k)^6})\big]. \quad (2.32)$$

Finally, by the use of Equations (2.23)–(2.32) in the last step of (2.15), we present the error equation as

$$\sigma^{(k+1)} = 2\big((3I+B^2)A_3A_2^2-(I+B^2)A_2A_3A_2+7A_2^4\big)\sigma^{(k)^5}+O(\sigma^{(k)^6}).$$

This completes the proof. □

For the further reference in this study, let us denote the new method (2.15) by $M_{5,1}$.

3 Computational efficiency

An iterative method's computing efficiency is expressed as the efficiency index $E = p^{1/C}$ (see [5]), where p denotes the order of convergence and C represents the computational cost that is determined by

$$C(\mu, m, \eta) = A(m)\mu + P(m, \eta).$$

Here, $P(m, \eta)$ indicates the number of products required each iteration, and $A(m)$ indicates the number of evaluations of scalar functions utilized in the evaluation of G and $[t, y; G]$. A ratio $\mu > 0$ between products and evaluations of scalar functions and a ratio $\eta \geq 1$ between quotients and products are needed in order to represent the value of $C(\mu, m, \eta)$ in terms of products.

In addition, the efficiency index can also be re-formulated as (see [16]),

$$E = \frac{1}{\rho} \frac{\log p}{C}, \tag{3.1}$$

where ρ is a parameter that can be fixed as per convenience. The efficiency index defined by Equation (3.1) is specifically used in this section to assess the computational efficiency of method.

The following are the many assessments and procedures that go into the overall cost of computing. We evaluate m scalar functions $(g_1, g_2, \ldots, g_m)$ when computing G in any iterative function. Similarly, m scalar functions in both $G(t)$ and $G(y)$ are evaluated independently when computing a divided difference $[t, y; G]$. In addition, m^2 quotients from any divided difference must be added. A linear system involving $m(m-1)(2m-1)/6$ products and $m(m-1)/2$ quotients in the LU decomposition and $m(m-1)$ products and m quotients in the resolution of two triangular linear systems must be solved in order to construct an inverse linear operator. When multiplying a matrix by a scalar or by a vector, we need to add m^2 products, and when multiplying a vector by a scalar, we need to add m products.

In order to demonstrate the computational efficiency of new fifth-order method $M_{5,1}$, consider the existing fourth-order, fifth-order and sixth-order methods that are already described in Section 1. The fourth-order methods are denoted as $M_{4,1}$, $M_{4,2}$ and $M_{4,3}$, whereas the sixth-order methods are denoted as $M_{6,1}$ and $M_{6,2}$. Further, denote the existing fifth-order method by $M_{5,2}$ that is expressed in Equation (1.6). Denoting the efficiency index of method $M_{p,i}$ by $E_{p,i}$ and computational cost by $C_{p,i}$, then taking into account the above considerations, we have

$$C_{4,1} = \frac{m}{3}\big(-8 + 2m^2 + 6\eta(1+2m) + 9m(1+\mu)\big) \text{ and } E_{4,1} = \frac{1}{\rho}\frac{\log 4}{C_{4,1}}.$$

$$C_{4,2} = \frac{m}{3}\big(-5+2m^2+3\eta(1+3m)+6\mu+6m(1+\mu)\big) \text{ and } E_{4,2} = \frac{1}{\rho}\frac{\log 4}{C_{4,2}}.$$

$$C_{4,3} = \frac{m}{6}\big(-5+2m^2+15\eta(1+m)+12\mu+3m(7+4\mu)\big) \text{ and } E_{4,3} = \frac{1}{\rho}\frac{\log 4}{C_{4,3}}.$$

$$C_{5,1} = \frac{m}{6}\big(1 + 2m^2 + 3\eta(5+3m) + 30\mu + 3m(5+2\mu)\big) \text{ and } E_{5,1} = \frac{1}{\rho}\frac{\log 5}{C_{5,1}}.$$

$$C_{5,2} = \frac{m}{6}\big(-16+4m^2+18\eta(1+m)+18\mu+6m(3+2\mu)\big) \text{ and } E_{5,2} = \frac{1}{\rho}\frac{\log 5}{C_{5,2}}.$$

$$C_{6,1} = \frac{m}{3}\big(-8+2m^2+3\eta(2+3m)+9\mu+3m(3+2\mu)\big) \text{ and } E_{6,1} = \frac{1}{\rho}\frac{\log 6}{C_{6,1}}.$$

$$C_{6,2} = \frac{m}{6}\big(-5+2m^2+3\eta(9+5m)+18\mu+3m(13+4\mu)\big) \text{ and } E_{6,2} = \frac{1}{\rho}\frac{\log 6}{C_{6,2}}.$$

Consider the ratio

$$R_{p,i;q,j} = \frac{E_{p,i}}{E_{q,j}} = \frac{C_{q,j}\log(p)}{C_{p,i}\log(q)},$$

to compare the efficiency of the iterative processes under consideration, such as $\mathrm{M}_{p,i}$ against $\mathrm{M}_{q,j}$.

It is evident that the iterative technique $\mathrm{M}_{p,i}$ will be more efficient than $\mathrm{M}_{q,j}$ for $R_{p,i;q,j} > 1$. Let us symbolize it mathematically as $\mathrm{M}_{p,i} \gtrdot \mathrm{M}_{q,j}$. Thus, resolving the inequality $R_{p,i;q,j} > 1$ will provide the range of parameters, for which $\mathrm{M}_{p,i} \gtrdot \mathrm{M}_{q,j}$ holds. These results can further be projected geometrically for some special cases of m. In particular, by fixing the value of η, the considered ratio $R_{p,i;q,j}$ can be plotted against the parameter μ for different cases of m. Let us fix the value $\eta = 3$ in each case.

As discussed above, the efficiency comparison is presented below by considering the analytical as well as geometrical approaches.

$\mathrm{M}_{5,1}$ *versus* $\mathrm{M}_{4,1}$ *case*: In this case, the considered ratio is

$$R_{5,1;4,1} = \frac{2\log(5)(-8 + 2m^2 + 6\eta(1+2m) + 9m(1+\mu))}{\log(4)(1 + 2m^2 + 3\eta(5+3m) + 30\mu + 3m(5+2\mu))}.$$

It can be deduced that, for each $\mu > 0$ and $\eta \geq 1$, the inequality $R_{5,1;4,1} > 1$ holds for each $m \geq 3$, which implies that $\mathrm{M}_{5,1}$ is more efficient than $\mathrm{M}_{4,1}$, i.e. $\mathrm{M}_{5,1} \gtrdot \mathrm{M}_{4,1}$, for $m \geq 3$. However, for the case of $m = 2$, it is easy to certify that $\mathrm{M}_{5,1} \gtrdot \mathrm{M}_{4,1}$ holds only if

$$\mu < \big(-(13+11\eta)\log(2) + 2(3+5\eta)\log(5)\big)/\big(2\log(128/125)\big).$$

The developed results are projected geometrically in the Figure 1 for the special cases of $m = 2, 10, 100, 500$, where $0 \leq \mu \leq 500$ and $\eta = 3$. It is evident that the plotted lines, for each case of m, completely lie above the horizontal line $R = 1$ which in fact depicts that $R_{5,1;4,1} > 1$ and hence $\mathrm{M}_{5,1} \gtrdot \mathrm{M}_{4,1}$ in the considered cases.

$\mathrm{M}_{5,1}$ *versus* $\mathrm{M}_{4,2}$ *case*: The ratio is

$$R_{5,1;4,2} = \frac{2\log(5)(-5 + 2m^2 + 3\eta(1 + 3m) + 6\mu + 6m(1 + \mu))}{\log(4)(1 + 2m^2 + 3\eta(5 + 3m) + 30\mu + 3m(5 + 2\mu))}.$$

For the given parameters $\mu > 0$ and $\eta \geq 1$, the inequality $R_{5,1;4,2} > 1$ holds or each $m \geq 3$, which implies that $\mathrm{M}_{5,1} \gtrdot \mathrm{M}_{4,2}$ for $m \geq 3$, whereas for $m = 2$, the inequality $R_{5,1;4,2} > 1$ leads to the condition, $\mu < \frac{-(13+11\eta)\log(2)+(5+7\eta)\log(5)}{2\log(128/125)}$. Geometrically, the results are presented in Figure 2 for the special cases of $m = 2, 10, 100, 500$, wherein it is clear that the plotted curves remain above the line $R = 1$ in the given range, i.e., $\mathrm{M}_{5,1} \gtrdot \mathrm{M}_{4,2}$ for the considered cases.

$\mathrm{M}_{5,1}$ *versus* $\mathrm{M}_{4,3}$ *case*: Now ratio is

$$R_{5,1;4,3} = \frac{\log(5)(-5 + 2m^2 + 15\eta(1 + m) + 12\mu + 3m(7 + 4\mu))}{\log(4)(1 + 2m^2 + 3\eta(5 + 3m) + 30\mu + 3m(5 + 2\mu))}.$$

In this case, the inequality $R_{5,1;4,3} > 1$ and therefore $\mathrm{M}_{5,1} \gtrdot \mathrm{M}_{4,3}$ holds for each $m \geq 3$, whereas for $m = 2$, $\mathrm{M}_{5,1} \gtrdot \mathrm{M}_{4,3}$ is true only if following condition holds: $\mu < \frac{-2(13+11\eta)\log(2)+15(1+\eta)\log(5)}{4\log(128/125)}$. As similar to the previous cases, the geometrical comparison in this particular case is presented in Figure 3.

$\mathrm{M}_{5,1}$ *versus* $\mathrm{M}_{5,2}$ *case*: Here ratio is

$$R_{5,1;5,2} = \frac{-16 + 4m^2 + 18\eta(1 + m) + 18\mu + 6m(3 + 2\mu)}{1 + 2m^2 + 3\eta(5 + 3m) + 30\mu + 3m(5 + 2\mu)}.$$

It can be deduced by the inequality $R_{5,1;5,2} > 1$ that $\mathrm{M}_{5,1} \gtrdot \mathrm{M}_{5,2}$ for each $m \geq 2$, which implies that $\mathrm{M}_{5,1}$ is more efficient than $\mathrm{M}_{5,2}$ for $m \geq 2$. The geometrical comparison in this case is displayed in Figure 4.

$\mathrm{M}_{5,1}$ *versus* $\mathrm{M}_{6,1}$ *case*: The ratio is

$$R_{5,1;6,1} = \frac{2\log(5)(-8 + 2m^2 + 3\eta(2 + 3m) + 9\mu + 3m(3 + 2\mu))}{\log(6)(1 + 2m^2 + 3\eta(5 + 3m) + 30\mu + 3m(5 + 2\mu))}.$$

The inequality $R_{5,1;6,1} > 1$ is true for each $m \geq 3$, which implies that $\mathrm{M}_{5,1} \gtrdot \mathrm{M}_{6,1}$ for $m \geq 3$. However, $R_{5,1;6,1} > 1$ holds for $m = 2$ under the condition, $\mu < \frac{4(3+4\eta)\log(5)-(13+11\eta)\log(6)}{14\log(6/5)}$. These results are projected geometrically in the Figure 5 for the special cases of considered parameters.

$\mathrm{M}_{5,1}$ *versus* $\mathrm{M}_{6,2}$ *case*: The ratio is

$$R_{5,1;6,2} = \frac{\log(5)(-5 + 2m^2 + 3\eta(9 + 5m) + 18\mu + 3m(13 + 4\mu))}{\log(6)(1 + 2m^2 + 3\eta(5 + 3m) + 30\mu + 3m(5 + 2\mu))}.$$

For $3 \leq m \leq 120$, the inequality $R_{5,1;6,2} > 1$ holds true, that means $\mathrm{M}_{5,1} \gtrdot \mathrm{M}_{6,2}$ for $3 \leq m \leq 120$. In particular for $m = 2$, $\mathrm{M}_{5,1} \gtrdot \mathrm{M}_{6,2}$ is true only if

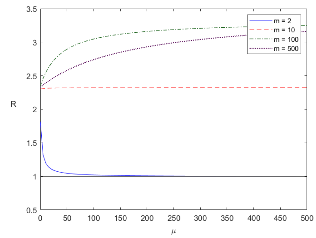

Figure 1. $\mathbf{M}_{5,1}$ vs. $\mathrm{M}_{4,1}$.

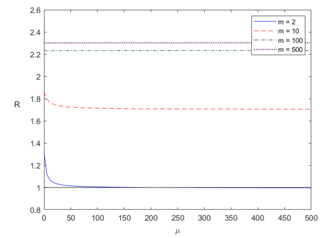

Figure 2. $\mathrm{M}_{5,1}$ vs. $\mathrm{M}_{4,2}$.

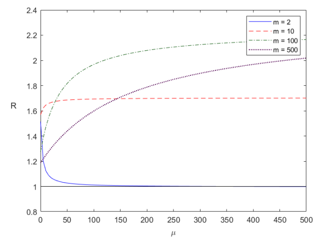

Figure 3. $\mathrm{M}_{5,1}$ vs. $\mathrm{M}_{4,3}$.

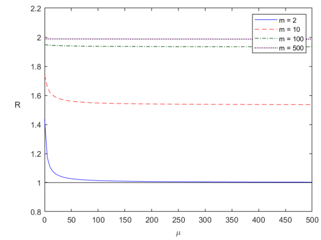

Figure 4. $\mathrm{M}_{5,1}$ vs. $\mathrm{M}_{5,2}$.

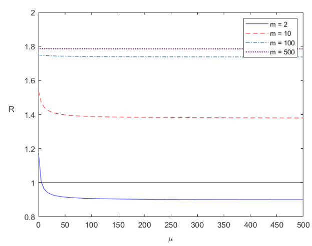

Figure 5. $\mathrm{M}_{5,1}$ vs. $\mathrm{M}_{6,1}$.

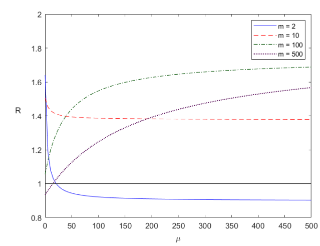

Figure 6. $\mathrm{M}_{5,1}$ vs. $\mathrm{M}_{6,2}$.

$\mu < \frac{(27+19\eta)\log(5)-(13+11\eta)\log(6)}{14\log(6/5)}$, whereas for $m \geq 121$, $\mathrm{M}_{5,1} \gtrdot \mathrm{M}_{6,2}$ holds if $\mu > \frac{(-2m^2-3m(5\eta+13)-27\eta+5)\log(5)+(2m^2+3m(3\eta+5)+15\eta+1)\log(6)}{6m\log(25/6)-6\log(7776/125)}$. The geometrical comparison of methods in this case is depicted in Figure 6.

The analytical as well as geometrical comparison, in each of the above cases, determines the range of parameters for which the new developed method possesses higher efficiency index than the other methods. In general, the efficiency analysis in this section leads to establish that the method $\mathrm{M}_{5,1}$ is computationally more efficient than the existing methods for the wider range of parameters.

4 Numerical experimentation

In this section, the numerical performance of developed method $\mathrm{M}_{5,1}$ is analyzed, and further, the corresponding results are compared with the existing methods to arrive at some logical conclusions. The numerical experimentation is executed by considering the systems of nonlinear equations arising in various practical situations. Additionally, in order to build a relation between the numerical experimentation and computational efficiency, it is required to estimate the values of parameters μ and η for each of the considered problem. To work out this estimation, the evaluation cost of each elementary function needs to be computed in terms of product units. In this regards, the CPU time (in milliseconds) elapsed during the execution of elementary operations along with their estimated cost in product units is displayed in Table 1. All these computations are executed using the software *Mathematica*. It is apparent from the Table 1 that the evaluation cost of division is approximately 2.81 times the product unit.

Table 1. CPU time and estimation of computational cost of elementary functions.

Functions	ty	t/y	$\sqrt{t}$	e^t	$\ln(t)$	$\sin(t)$	$\cos(t)$	$\arctan(t)$
CPU Time	0.017	0.048	0.023	1.556	1.347	1.694	1.690	2.980
Cost	1	2.81	1.36	90.48	78.31	98.48	98.23	173.24

where $t = \sqrt{3} - 1$ *and* $y = \sqrt{5}$

To compare the performance of method $\mathrm{M}_{5,1}$ with the existing methods that are described in previous section, the following comparison parameters are considered: (i) Computational efficiency (E), (ii) Number of iterations (k) required for convergence, (iii) Errors between consecutive iterations ($\|e^{(k)}\| = \|t^{(k+1)} - t^{(k)}\|$), (iv) Residual error ($\|G(t^{(k)})\|$), (v) Approximate computational order of convergence (ACOC), and (vi) CPU time elapsed during the execution of algorithm. Let us note that the computational efficiency is computed by fixing the value $\rho = 10^{-5}$ in each test. Also, the ACOC is computed using the formula (see [5, 10]),

$$\mathrm{ACOC} = \ln\left(\|s^{(k+1)}\|/\|s^{(k)}\|\right) \Big/ \ln\left(\|s^{(k)}\|/\|s^{(k-1)}\|\right), \;\; s^{(k+1)} = t^{(k+1)} - t^{(k)}.$$

The stopping criterion being employed to abort the iterations in each test is as follows: $\|t^{(k+1)} - t^{(k)}\| + \|G(t^{(k)})\| < 10^{-100}$, where $k < 100$. Notice that the parameter β, which is appearing in the methods $\mathrm{M}_{5,1}$, $\mathrm{M}_{4,1}$, $\mathrm{M}_{4,3}$ and $\mathrm{M}_{6,2}$, has been assigned a specific value $\beta = 10^{-2}$.

Now, considering the following examples for the comparison analysis, the numerical results in each case are displayed in Table 2.

Example 1. Starting with the system of two nonlinear equations:

$$t_1 + e^{t_2} - \cos(t_2) = 0, \;\; 3t_1 - t_2 - \sin(t_1) = 0,$$

let us take the initial estimate as $t^{(0)} = (1, 1)^T$, in particular, to obtain solution, $t^* = (0, 0)^T$. In this particular problem, the values of efficiency parameters m, μ and η are obtained in accordance with the estimates as displayed in Table 1. These are given as, $\{m, \mu, \eta\} = \{2, 144, 2.81\}$.

Table 2. Comparison of performance of methods.

Method	E	k	$\|e^{(1)}\|$	$\|e^{(2)}\|$	$\|e^{(3)}\|$	$\|G(t^{(3)})\|$	ACOC	CPUt
Ex. 1								
$M_{5,1}$	78.13	4	8.28e−02	3.60e−06	1.10e−27	1.48e−27	5.00	0.159
$M_{4,1}$	77.18	5	7.30e−02	9.07e−06	2.57e−21	3.43e−21	4.00	0.266
$M_{4,2}$	78.00	5	1.24e−01	1.54e−05	8.95e−21	1.17e−20	4.00	0.219
$M_{4,3}$	77.66	5	1.14e−01	1.02e−04	1.08e−16	1.45e−16	4.00	0.217
$M_{5,2}$	77.43	4	6.51e−02	4.20e−07	3.46e−33	4.64e−33	5.00	0.166
$M_{6,1}$	86.43	4	3.30e−02	3.72e−11	2.38e−64	3.15e−64	6.00	0.156
$M_{6,2}$	85.47	4	4.52e−02	1.19e−08	5.79e−48	7.76e−48	6.00	0.161
Ex. 2								
$M_{5,1}$	31.21	4	2.36e−02	9.06e−11	7.23e−53	2.72e−52	5.00	1.11
$M_{4,1}$	17.72	4	2.16e−02	1.61e−08	4.67e−33	1.76e−32	4.00	1.83
$M_{4,2}$	22.20	5	1.28e−01	4.76e−03	5.10e−08	1.94e−08	4.00	1.75
$M_{4,3}$	22.25	4	2.56e−02	4.48e−08	3.67e−31	1.38e−30	4.00	1.31
$M_{5,2}$	23.73	7	1.19E+01	8.05e−01	1.68e−01	1.12e−01	5.00	2.42
$M_{6,1}$	26.47	5	1.63e−01	2.35e−03	7.64e−13	2.91e−13	6.00	1.91
$M_{6,2}$	26.26	4	5.02e−03	7.33e−16	6.83e−93	2.57e−92	6.00	1.31
Ex. 3								
$M_{5,1}$	10.07	3	2.59e−04	6.33e−23	0	0	5.00	6.31
$M_{4,1}$	4.34	4	2.30e−03	1.61e−15	3.83e−64	3.17e−64	4.00	10.33
$M_{4,2}$	5.88	4	2.28e−02	6.05e−11	3.00e−45	2.48e−45	4.00	8.00
$M_{4,3}$	5.95	4	2.08e−03	2.04e−15	1.89e−63	1.56e−63	4.00	7.83
$M_{5,2}$	6.51	3	1.04e−03	4.49e−20	0	0	5.00	8.23
$M_{6,1}$	7.26	3	2.95e−03	6.66e−22	0	0	6.00	7.09
$M_{6,2}$	7.27	3	1.65e−04	9.40e−30	0	0	6.00	6.50
Ex. 4								
$M_{5,1}$	0.52	4	3.97e−01	1.78e−07	3.42e−39	8.93e−39	5.00	0.86
$M_{4,1}$	0.17	4	3.97e−01	8.84e−07	2.07e−29	5.41e−29	4.00	1.23
$M_{4,2}$	*Diverges*	–	–	–	–	–	–	–
$M_{4,3}$	0.26	5	6.49e−01	2.29e−05	3.65e−23	9.55e−23	4.00	1.02
$M_{5,2}$	0.28	5	1.94E+00	2.83e−03	4.23e−17	1.11e−16	5.00	1.41
$M_{6,1}$	*Diverges*	–	–	–	–	–	–	–
$M_{6,2}$	0.33	4	1.80e−01	2.28e−11	9.53e−71	2.49e−70	6.00	0.89
Ex. 5								
$M_{5,1}$	0.0721	3	8.78e−08	2.34e−51	0	0	5.00	4.71
$M_{4,1}$	0.0230	3	7.92e−08	1.62e−39	0	0	4.00	12.72
$M_{4,2}$	0.0323	4	1.13e−04	8.35e−21	8.22e−87	1.62e−84	4.00	15.97
$M_{4,3}$	0.0350	3	4.31e−06	7.32e−32	0	0	4.00	8.00
$M_{5,2}$	0.0373	3	1.04e−07	7.75e−49	0	0	5.00	10.89
$M_{6,1}$	0.0415	3	3.25e−07	1.20e−45	0	0	6.00	12.92
$M_{6,2}$	0.0447	3	1.32e−09	9.32e−68	0	0	6.00	5.82
Ex.6								
$M_{5,1}$	2.55e−03	4	5.01e−03	1.02e−17	3.64e−91	8.95e−91	5.00	19.97
$M_{4,1}$	9.52e−04	4	5.46e−04	1.84e−19	2.34e−81	5.77e−81	4.00	26.72
$M_{4,2}$	*Diverges*	–	–	–	–	–	–	–
$M_{4,3}$	1.66e−03	4	9.27e−03	1.48e−13	9.47e−57	2.33e−56	4.00	22.91
$M_{5,2}$	*Diverges*	–	–	–	–	–	–	–
$M_{6,1}$	*Diverges*	–	–	–	–	–	–	–
$M_{6,2}$	2.12e−03	3	2.80e−04	1.35e−29	0	0	6.00	20.24

Example 2. Consider the system of equations:

$$t_i - \cos\Big(t_i - \sum_{j=1, j\neq i}^{n} t_j\Big) = 0, \quad i = 1, 2, ..., n.$$

Taking n=5, and choosing the initial estimate, $t^{(0)} = \left(\frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4}\right)^T$, the solution so obtained is given as $t^* = (0.390..., 0.390..., 0.390..., 0.390..., 0.390...)^T$. Here, the efficiency parameters are estimated as; $\{m, \mu, \eta\} = \{5, 98.23, 2.81\}$.

Example 3. Consider the nonlinear integral equation $G(t) = 0$, where

$$G(t)(s) = t(s) - 1 + \frac{1}{2}\int_0^1 s\cos(t(x))dx, \quad s \in [0, 1], \tag{4.1}$$

with $t \in C[0, 1]$. Here, $C[0, 1]$ is defined as the space of continuous functions on $[0, 1]$ along with the norm, $\|t\| = \max|t(s)|$, where $s \in [0, 1]$. This kind of integral equation, particularly called Chandrasekhar equation (see [2]), arises in the study of radiative transfer theory, neutron transport theory and kinetic theory of gases.

Discretizing the Equation (4.1) using the Trapezoidal rule of integration with the step size $h = 1/n$, the system of nonlinear equations is obtained as

$$t_i - 1 + \frac{s_i}{2n}\Big(\frac{1}{2}\cos(t_0) + \sum_{j=1}^{n-1}\cos(t_j) + \frac{1}{2}\cos(t_n)\Big) = 0, \quad i = 1, 2, ..., n, \tag{4.2}$$

where $t_i = t(x_i)$, $s_i = x_i = \frac{i}{n}$ and $t_0 = \frac{1}{2}$. Selecting $n = 10$, the solution of the transformed problem (4.2) is

$$\begin{aligned} t^* = (&0.9654..., 0.9309..., 0.8964..., 0.8619..., 0.8274...,\\ &0.7929..., 0.7584..., 0.7239..., 0.6894..., 0.6549...)^T. \end{aligned}$$

For the comparison analysis, let us set the initial estimate as $(2, \overset{10}{\cdots}, 2)^T$. In this problem, the specific values of parameters are; $\{m, \mu, \eta\} = \{10, 99.33, 2.81\}$.

Example 4. Next, considering the system of nonlinear equations as:

$$\begin{cases} t_i t_{i+1} - e^{-t_i} - e^{-t_{i+1}} = 0, \quad i = 1, 2, ..., n-1, \\ t_n t_1 - e^{-t_n} - e^{-t_1} = 0. \end{cases}$$

Taking $n = 50$, and choosing the initial estimate $t^{(0)} = \left(\frac{5}{2}, \overset{50}{\cdots}, \frac{5}{2}\right)^T$ to obtain the following solution,

$$t^* = \big(0.9012..., 0.9012..., \cdots\cdots, 0.9012...\big)^T.$$

The values of parameters in this particular problem are computed as; $\{m, \mu, \eta\} = \{50, 91.48, 2.81\}$.

Example 5. Now, consider the system of equations as:

$$\tan^{-1}(t_i) + 1 - 2\left(\sum_{j=1, j\neq i}^{n} t_j\right) = 0, \quad i = 1, 2, ..., n.$$

Selecting $n = 100$, let us choose the estimate $t^{(0)} = (-1, \overset{100}{\cdots}, -1)^T$ for solution,

$$t^* = (0.00507..., 0.00507...., \cdots\cdots, 0.00507....)^T.$$

Here, the estimated values of parameters are; $\{m, \mu, \eta\} = \{100, 174.24, 2.81\}$.

Example 6. Lastly, we consider a large system of 500 equations,

$$\begin{cases} t_i + \log(2 + t_i + t_{i+1}) = & 0, \quad i = 1, 2, ..., 499, \\ t_{500} + \log(2 + t_{500} + t_1) = & 0. \end{cases}$$

To obtain its solution,

$$t^* = (-0.3149..., -0.3149..., \cdots\cdots, -0.3149...)^T,$$

we select the initial estimate $t^{(0)} = \left(-\frac{1}{2}, \overset{500}{\cdots}, -\frac{1}{2}\right)^T$. In this case, the values of parameters are; $\{m, \mu, \eta\} = \{500, 78.31, 2.81\}$.

Analyzing the findings from Table 2, it can be deduced that the proposed method $\mathrm{M}_{5,1}$ is computationally more efficient in comparison to the existing methods. The efficiency index of $\mathrm{M}_{5,1}$ is highest in each case barring a case of Example 1, wherein the efficiency of sixth-order methods is comparatively higher. It can be clearly observed that the proposed method requires less or equal number of iterations to converge, whereas there are some cases wherein the existing methods are diverging in accordance with the specified criteria. Further, the errors in successive iterations evidently imply the high degree of precision of the new method, even for the sufficiently large systems. In majority of the cases, the proposed method utilizes less CPU time. In addition, the fifth order convergence is also proven numerically through computing ACOC.

4.1 Applications

Here, we examine the efficacy of new method by obtaining solution to some initial and boundary value problems, and display the outcome in respect of: (i) Number of iterations (k), (ii) Efficiency index (E) and (iii) CPU time.

4.1.1 Van der Pol-Rayleigh problem

Consider a problem of hybrid Van der Pol-Rayleigh oscillator [8] with external sinusoidal forcing, which has applications to many physical systems including the bipedal robot locomotion. It is presented as,

$$\frac{d^2x}{dt^2} + x - \epsilon\left(1 - ax^2 - b\left(\frac{dx}{dt}\right)^2\right)\frac{dx}{dt} = A\sin(\omega t), \ t \geq 0, \tag{4.3}$$
$$x(0) = 1, \ \ x'(0) = 1,$$

where, $x(t)$ is the position coordinate, ϵ is a damping parameter, A represents the amplitude of oscillations with pulsation ω, and a, b are constants. The given problem is investigated in [8] for the different sets of parameters' values. To be specific, we take $\epsilon = 1/100$, $A = 1$, $\omega = 3$ and $a = b = 1$.

To obtain the numerical solution of (4.3) in region $0 \leq t \leq 1$, consider a partitioning of $[0, 1]$, with each sub-interval of uniform length $h = 1/n$, as

$$0 = t_0 < t_1 < t_2 < \cdots < t_{n-1} < t_n = 1, \quad \text{where } t_i = t_0 + ih, \quad (i = 1, 2, \ldots, n).$$

Denote $x(t_i) = x_i$ for each $i = 1, \ldots, n$. Then, approximating the first and second order derivatives by using the finite differences

$$x_i' = \frac{x_i - x_{i-1}}{h}, \quad x_i'' = \frac{x_i - 2x_{i-1} + x_{i-2}}{h^2},$$

the following system of nonlinear equations in n variables is obtained by transforming the Equation (4.3),

$$\frac{x_i - 2x_{i-1} + x_{i-2}}{h^2} + x_i - \frac{1}{100}\left(1 - x_i^2 - \frac{(x_i - x_{i-1})^2}{h^2}\right)\frac{x_i - x_{i-1}}{h} - \sin(3t_i) = 0,$$

where $i = 1, \ldots, n$. Setting $n = 25$, the above system reduces to 25 nonlinear equations for which the approximate solution is plotted in Figure 7. Initial estimate to the given solution is selected as; $(\frac{3}{2}, \overset{25}{\cdots}, \frac{3}{2})^T$, and estimated values of parameters for this problem are; $\{m, \mu, \eta\} = \{25, 6.96, 2.81\}$. Numerical performance of the methods is displayed in the Table 3, wherein it can be observed that the method $\mathrm{M}_{5,1}$ has highest efficiency index for the given case. Further, it takes least number of iterations to converge whereas most of the considered comparison methods are diverging.

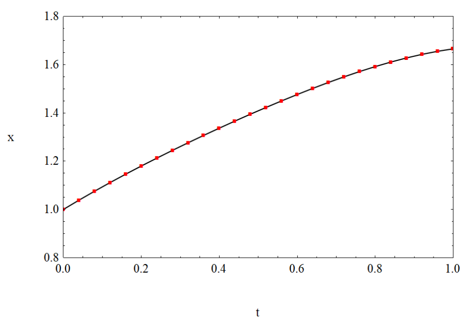

Figure 7. Approximate solution of Van der Pol-Rayleigh oscillator equation.

4.1.2 Fisher's equation

Consider the Fisher's equation [16], which specifically models the growth of particles in reaction-diffusion system and it is expressed as

$$\frac{\partial u}{\partial t} = \delta \frac{\partial^2 u}{\partial x^2} + u(1 - u), \tag{4.4}$$

Table 3. Comparison of performance of methods for boundary value problems

Method	$M_{5,1}$	$M_{4,1}$	$M_{4,2}$	$M_{4,3}$	$M_{5,2}$	$M_{6,1}$	$M_{6,2}$
Van der Pol problem							
k	5	*Diverges*	*Diverges*	6	*Diverges*	*Diverges*	7
E	2.065	–	–	1.336	–	–	1.601
CPUt	0.719	–	–	1.407	–	–	1.328
Fisher equation							
k	4	5	4	5	5	4	4
E	0.365	0.150	0.164	0.269	0.187	0.209	0.328
CPUt	1.125	1.734	1.281	1.566	2.266	1.75	1.659
Heat conduction problem							
k	3	4	3	4	3	3	3
E	6.89e−03	2.90e−03	3.00e−03	5.57e−03	3.47e−03	3.87e−03	7.06e−03
CPUt	5.063	17.172	13.719	9.187	9.406	13.157	8.112

where $u = u(x,t)$ with x and t as spatial and temporal domains, and δ is the diffusion coefficient. The boundary conditions on $u(x,t)$ are imposed as: $\frac{\partial u}{\partial x}(0,t) = 0$ and $\frac{\partial u}{\partial x}(1,t) = 0$ for all $t \geq 0$, along with the initial condition as: $u(x,0) = \frac{3}{2} + \frac{1}{2}\cos(\pi x)$ for $0 \leq x \leq 1$. Consider the domain $D = \{(x,t) \mid (x,t) \in [0,1] \times [0,1]\}$ with the partition of step size $h = 1/p$ and $s = 1/q$, respectively for the domains x and t, i.e.,

$$x_i = 0 + ih,\ i{=}0,1,\ldots,p,\ h{=}1/p,\ \text{and}\ \ t_j{=}0{+}jk,\quad j = 0,1,\ldots,q,\quad k = 1/q.$$

Denote $u(x_i,t_j) = u_{i,j}$ for each i, j and then approximating the Equation (4.4) at any point (x_i,t_j) using the finite differences,

$$\frac{\partial u}{\partial t} = \frac{u_{i,j} - u_{i,j-1}}{s},\quad \frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2},$$

the given equation reduces into a system of $(p-1)\times q$ equations. Then, selecting $p = 11$, $q = 10$ and taking $\delta = 1$ in particular, the numerical solution of the system is plotted in the Figure 8. Starting with the approximation $(3, \overset{100}{\cdots}, 3)^T$, the performance of methods is depicted in Table 3. The estimated values of parameters are; $\{m, \mu, \eta\} = \{100, 3.8, 2.81\}$. Results so obtained clearly show the better computational efficiency of new method than existing ones.

4.1.3 Heat conduction problem

Lastly, let us consider a particular case of heat conduction problem (see [10]). Let $u = u(x,t)$, where x and t are the spatial and temporal domains, respectively, then the equation is described as follows:

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial x} + \frac{\partial u}{\partial t} - u^2 + g(x,t),\quad 0 \leq x \leq 1,\ t \geq 0, \tag{4.5}$$

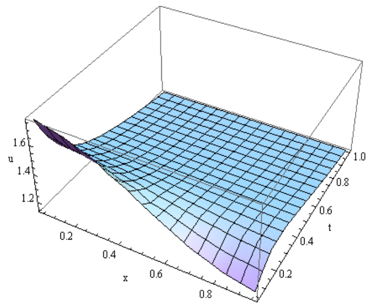

Figure 8. Approximate solution of Fisher equation.

where $g(x,t)$ is defined as, $g(x,t) = e^{-t}(-\pi\cos(\pi x) - (\pi^2 - 2)\sin(\pi x))$. The boundary conditions are $u(0,t) = 0$ and $u(1,t) = 0$, whereas the initial condition is used as $u(x,0) = \sin(\pi x)$. By choosing the maximum value of $t = 1$, we select the partition of domain $[0,1] \times [0,1]$ as $x_i = 0 + ih$, $t_j = 0 + jk$, for $i = 0,1,\ldots,p$ and $j = 0,1,\ldots,q$, where $h = 1/p$ and $k = 1/q$. Let us denote $u_{i,j} = u(x_i,t_j)$ and $g_{i,j} = g(x_i,t_j)$ for each i and j. Then, the initial condition is transformed as: $u_{i,0} = \sin(\pi ih)$ for each $i = 0,1,2,\ldots,p$, and the boundary conditions as: $u_{0,j} = 0$, $u_{p,j} = 0$ for each $j = 0,1,2,\ldots,q$. To discretize the Equation (4.5), consider the following approximations:

$$\frac{\partial u}{\partial x} = \frac{u_{i+1,j} - u_{i-1,j}}{2h}, \quad \frac{\partial u}{\partial t} = \frac{u_{i,j} - u_{i,j-1}}{k} \text{ and } \frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2},$$

and consequently, the system of $(p-1)\times q$ nonlinear equations is obtained as,

$$(2-h)ku_{i+1,j} - 2(2k+h^2)u_{i,j} + (2+h)ku_{i-1,j} + 2kh^2u_{i,j}^2 + 2h^2u_{i,j-1} = 2kh^2g_{i,j},$$

where $i = 1,2,\ldots,p-1$, and $j = 1,2,\ldots,q$. Fixing $p = 21$ and $q = 20$, the system reduces to 20×20 nonlinear equations.

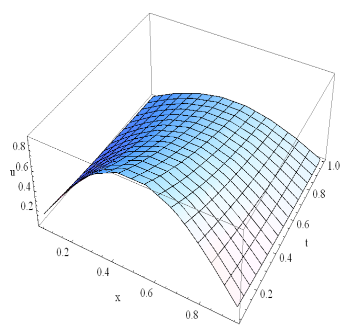

Figure 9. Approximate solution of heat conduction equation.

The approximate solution of the problem is presented in Figure 9, for which the initial estimate is selected as; $(\frac{1}{2}, \overset{400}{\cdots}, \frac{1}{2})^T$. The estimated values of parameters in this problem are; $\{m,\mu,\eta\} = \{400, 5.8, 2.81\}$. Numerical performance of the methods is displayed in Table 3, which clearly signifies the efficient behavior of the proposed method, even for such a large system of equations.

5 Conclusions

With an objective to develop the derivative-free method for solving systems of nonlinear equations, a multi-step formulation is presented initially in particular for the solution of uni-variate equations. The basic principle of construction is to extend the Traub-Steffensen method to a higher-order technique by introducing some undetermined parameters at the subsequent sub-steps. Determining these parameters so as to achieve the highest possible convergence order, the given scheme is found to possess the fifth order of convergence. Generalizing the presented scheme to a multidimensional case, the method is proven to maintain its fifth order of convergence. The generalized scheme is then analyzed comprehensively for its computational aspects and numerical performance. Introducing some necessary parameters to further compute the efficiency index, the computational efficiency is examined in analytical as well as geometrical approach. It is found that the proposed method achieves the higher efficiency index as compared to the existing methods of similar nature, even for the sufficiently large values of m. Finally, the performance is tested and compared by solving the systems of nonlinear equations along with some boundary value problems. In general, the proposed method exhibits remarkable results in comparison to the existing counterparts by showing the better numerical accuracy, higher efficiency index and less CPU time. Similar numerical experimentations have been carried out for a number of problems and results are found to be on a par with those presented here.

References

[1] I.K. Argyros. *The Theory and Applications of Iteration Methods*. CRC Press, New York, 2022. https://doi.org/10.1201/9780203719169.

[2] S. Chandrasekhar. *Radiative Transfer*. Dover Publications Inc., New York, 1960.

[3] R. Erfanifar, K. Sayevand and H. Esmaeili. On modified two-step iterative method in the fractional sense: some applications in real world phenomena. *International Journal of Computer Mathematics*, **97**(10):2109–2141, 2020. https://doi.org/10.1080/00207160.2019.1683547.

[4] M. Grau and J.L. Díaz-Barrero. An improvement to Ostrowski root-finding method. *Applied Mathematics and Computation*, **173**(1):450–456, 2006. https://doi.org/10.1016/j.amc.2005.04.043.

[5] M. Grau, M. Noguera and S. Amat. On the approximation of derivatives using divided difference operators preserving the local convergence order of iterative methods. *Journal of Computational and Applied Mathematics*, **237**(1):363–372, 2013. https://doi.org/10.1016/j.cam.2012.06.005.

[6] Y. He and C. Ding. Using accurate arithmetics to improve numerical reproducibility and stability in parallel applications. *The Journal of Supercomputing*, **18**:259–277, 2001. https://doi.org/10.1023/A:1008153532043.

[7] M.A. Hernández-Verón, S. Yadav, Á.A. Magreñán, E. Martínez and S. Singh. An algorithm derivative-free to improve the Steffensen-type methods. *Symmetry*, **14**(1):1–26, 2022. https://doi.org/10.3390/sym14010004.

[8] L.A. Hinvi, C.H. Miwadinou, A.V. Monwanou and J.B. Chabi Orou. Nonlinear dynamics of system oscillations modeled by a forced Van der Pol generalized oscillator. *International Journal of Engineering and Applied Sciences*, **4**:257398, 2017. https://doi.org/10.48550/arXiv.1402.4392.

[9] C.L. Howk, J.L. Hueso, E. Martínez and C. Teruel. A class of efficient high-order iterative methods with memory for nonlinear equations and their dynamics. *Mathematical Methods in the Applied Sciences*, **41**:7263–7282, 2018. https://doi.org/10.1002/mma.4821.

[10] D. Kumar, J.R. Sharma and H. Singh. Higher order Traub–Steffensen type methods and their convergence analysis in Banach spaces. *International Journal of Nonlinear Sciences and Numerical Simulation*, **24**(4):1565–1587, 2023. https://doi.org/10.1515/ijnsns-2021-0202.

[11] M. Narang, S. Bhatia, A.S. Alshomrani and V. Kanwar. General efficient class of Steffensen type methods with memory for solving systems of nonlinear equations. *Journal of Computational and Applied Mathematics*, **352**:23–39, 2019. https://doi.org/10.1016/j.cam.2018.10.048.

[12] J.M. Ortega and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, New York, 1970. https://doi.org/10.1137/1.9780898719468.

[13] N. Revol and F. Rouillier. Motivation for an arbitrary precision interval arithmetic and the mpfi library. *Reliable Comput.*, **11**:275–290, 2005.

[14] J.R. Sharma and H. Arora. Efficient derivative-free numerical methods for solving systems of nonlinear equations. *Comp. Appl. Math.*, **35**:269–284, 2016. https://doi.org/10.1007/s40314-014-0193-0.

[15] A. Singh. An efficient fifth-order Steffensen-type method for solving systems of nonlinear equations. *Int. J. Comput. Sci. Math.*, **9**(5):501–514, 2018. https://doi.org/10.1504/IJCSM.2018.095502.

[16] H. Singh, J.R. Sharma and S. Kumar. A simple yet efficient two-step fifth-order weighted-Newton method for nonlinear models. *Numer. Algor.*, **93**:203–225, 2023. https://doi.org/10.1007/s11075-022-01412-w.

[17] J.F. Traub. *Iterative Methods for the Solution of Equations.* Chelsea Publishing Company, New York, 1982.

[18] X. Wang and T. Zhang. A family of Steffensen type methods with seventh-order convergence. *Numer. Algor.*, **62**:429–444, 2013. https://doi.org/10.1007/s11075-012-9597-3.

[19] X.Y. Xiao. New techniques to develop higher order iterative methods for systems of nonlinear equations. *Comp. Appl. Math.*, **41**(243):1–19, 2022. https://doi.org/doi.org/10.1007/s40314-022-01959-3.

[20] Y. Zhang and P. Huang. High-precision time-interval measurement techniques and methods. *Progress in Astronomy*, **24**:1–15, 2006.