A Computationally Efficient Sixth-Order Method for Nonlinear Models

Janak Raj Sharma and Harmandeep Singh

Abstract The aim of the present study is to develop an iterative scheme of high convergence order with minimal computational cost. With this objective, a three-step method has been designed by utilizing only two Jacobian matrices, single matrix inversion, and three function evaluations. Under some standard assumptions, the proposed method is found to possess the sixth order of convergence. The iterative schemes with these characteristics are hardly found in the literature. The analysis is carried out to assess the computational efficiency of the proposed method, and further, outcomes are compared with the efficiencies of existing ones. In addition, numerical experiments are performed by applying the method to some practical nonlinear problems. The entire analysis remarkably favors the new technique compared with existing counterparts in terms of computational efficiency, stability, and CPU time elapsed during execution.

Keywords Nonlinear systems · Iterative techniques · Convergence order · Computational efficiency

1 Introduction

The systems of nonlinear equations arise by virtue of modeling the most of the physical processes or practical situations. The constructed models are generally expressed in mathematical form as

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

J. R. Sharma · H. Singh (✉)
Department of Mathematics, Sant Longowal Institute of Engineering and Technology, Longowal, Punjab 148106, India
e-mail: harman85pau@gmail.com

R. K. Sharma et al. (eds.), *Frontiers in Industrial and Applied Mathematics*,
Springer Proceedings in Mathematics & Statistics 410,
https://doi.org/10.1007/978-981-19-7272-0_39

where $O \in \mathbb{R}^m$ represents the zero vector, $F : \Omega \subseteq \mathbb{R}^m \to \mathbb{R}^m$ is a nonlinear mapping which is commonly represented as $(f_1(x), f_2(x), \ldots, f_m(x))^T$, $x = (x_1, \ldots, x_m)^T \in \Omega$, and $f_i : \mathbb{R}^m \to \mathbb{R}$ $(i = 1, \ldots, m)$ are nonlinear scalar functions.

Knowledge about the solution of the constructed nonlinear model plays an important role in forecasting the future developments of the corresponding physical problem. But, as a matter of fact, obtaining the analytical solutions of nonlinear systems is generally not feasible. To deal with this challenge, iterative methods [8, 13] offer the numerical solution up to the desired precision. The working process of an iterative method is based on the fixed point iteration theory, under which it locates the solution, $x^* \in \Omega$, of the given system (1), as a fixed point of a mapping $\phi : \mathbb{R}^m \to \mathbb{R}^m$, so that

$$x^{(k+1)} = \phi(x^{(k)}), \quad k = 0, 1, 2, \ldots,$$

where, $x^{(0)}$ is the initial estimate to the solution, and the mapping ϕ is constrained to satisfy some prescribed assumptions.

The most widely applied iterative procedure to find the solution to nonlinear equations is Newton's method

$$x^{(k+1)} = \phi(x^{(k)}) = x^{(k)} - F'(x^{(k)})^{-1} F(x^{(k)}), \quad k = 0, 1, 2, \ldots, \tag{2}$$

where $F(x)$ is continuously differentiable in some neighborhood of its solution, and $F'(x) \in \mathcal{L}(\mathbb{R}^m, \mathbb{R}^m)$ is a linear operator which is generally represented as a Jacobian matrix $\left[\frac{\partial f_i}{\partial x_j}\right]_{m \times m}$. This method approximates the simple solution of (1) with the quadratic rate of convergence. To improve the convergence rate of the method (2), numerous iterative schemes have been presented in the literature (see [2, 4–6, 10–12, 14] and references therein). As it is evident that Newton's scheme utilizes evaluation of a function (F), a Jacobian matrix (F'), and a matrix inversion (F'^{-1}) per iteration. An attempt to increase the rate of convergence of an iterative method generally leads to a technique that involves one or more additional evaluations per iteration than its predecessor. For instance, the Potra and Pták method [9], having cubic convergence, is one of the simplest improvements of the method (2), which is expressed as follows:

$$\begin{aligned} y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1} F(x^{(k)}), \\ x^{(k+1)} &= y^{(k)} - F'(x^{(k)})^{-1} F(y^{(k)}). \end{aligned} \tag{3}$$

Clearly, the above-presented two-step scheme utilizes an additional function evaluation over Newton's method.

The practice of designing an iterative scheme, by utilizing additional evaluations, accelerates the convergence order but it certainly increases the computational cost per iteration in terms of mathematical operations. Optimizing the computational cost with the improving convergence speed leads to the construction of computationally efficient techniques. The measure of efficiency is formulated in [8, 13] to analyze and further compare the efficiencies of iterative techniques. In addition, the necessary parameters have been introduced in [11] for the thorough investigation of this concept.

Taking into account the above discussion, in the next section, we shall present a simple and efficient iterative method showing the sixth order of convergence. The computational efficiency of the developed method is determined, analyzed, and compared with the efficiencies of existing methods in Sect. 3. Numerical performance is investigated in Sect. 4, and concluding remarks are given in Sect. 5.

2 Development of Method

The primary objective here is to design an iterative scheme that improves the convergence speed of the Potra and Pták method (3) without utilizing any additional inverse operator. In what follows, we shall present a three-step iterative method involving undetermined parameters, which are to be chosen in order to maximize the convergence order. In view of this, we consider the iterative scheme of type,

$$\begin{aligned} y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}), \\ z^{(k)} &= y^{(k)} - F'(x^{(k)})^{-1}F(y^{(k)}), \\ x^{(k+1)} &= z^{(k)} - \left[aI + F'(x^{(k)})^{-1}F'(y^{(k)})(bI + cF'(x^{(k)})^{-1}F'(y^{(k)}))\right] \\ &\qquad \times F'(x^{(k)})^{-1}F(z^{(k)}) \end{aligned} \tag{4}$$

where a, b, and c are the parameters.

Before proceeding to the convergence analysis, a preliminary result (see [7]) is stated below, which will be followed by the main theorem to show the sixth-order convergence for scheme (4).

Lemma 1 *Assume that the mapping $F : \Omega \subseteq \mathbb{R}^m \to \mathbb{R}^m$ is n-times Fréchet differentiable in a convex neighborhood $\Omega \in \mathbb{R}^m$, and let $x,\ t \in \Omega$, then the following expansion holds:*

$$F(x+t) = F(x) + F'(x)t + \frac{1}{2!}F''(x)t^2 + \ldots + \frac{1}{(n-1)!}F^{(n-1)}(x)t^{n-1} + R_n,$$

where $t^i = (t, \overset{i-times}{\ldots\ldots}, t)$, $F^{(i)}(x) \in \mathcal{L}(\mathbb{R}^m \times \overset{i-times}{\ldots\ldots} \times \mathbb{R}^m, \mathbb{R}^m)$ for each $i = 1, 2, \ldots$, and

$$\|R_n\| \le \frac{1}{n!} \sup_{0<h<1} \|F^{(n)}(x+ht)\| \, \|t\|^n.$$

Theorem 1 *Assume that a nonlinear mapping, $F : \Omega \subseteq \mathbb{R}^m \to \mathbb{R}^m$, is continuously differentiable sufficient number of times in some neighborhood of its simple zero x^*, contained in an open convex region Ω. Further, suppose that $F'(x)$ is non-singular and continuous in that neighborhood, and the initial approximation $x^{(0)}$ is sufficiently close to x^*. Then, the sequence of iterates generated by the method (4) converges to x^* with the sixth order of convergence, provided $a = \frac{7}{2}$, $b = -4$, and $c = \frac{3}{2}$.*

Proof Let $e^{(k)} = x^{(k)} - x^*$ be the error obtained at the kth iteration of (4). Then, as a consequence of Lemma 1, and the fact that $F(x^*) = O$, Taylor expansions of $F(x^{(k)})$ and $F'(x^{(k)})$, about x^*, are developed as

$$F(x^{(k)}) = F'(x^*)[e^{(k)} + A_2 e^{(k)^2} + A_3 e^{(k)^3} + A_4 e^{(k)^4} + A_5 e^{(k)^5} + A_6 e^{(k)^6}] + O(e^{(k)^7}), \quad (5)$$

$$F'(x^{(k)}) = F'(x^*)[I + 2A_2 e^{(k)} + 3A_3 e^{(k)^2} + 4A_4 e^{(k)^3} + 5A_5 e^{(k)^4} + 6A_6 e^{(k)^5}] + O(e^{(k)^6}), \quad (6)$$

where $e^{(k)^i} = (e^{(k)}, \overset{i-times}{\ldots\ldots}, e^{(k)})$, and $A_i = \frac{1}{i!} F'(x^*)^{-1} F^{(i)}(x^*), \quad i = 2, 3, \ldots,$ and consequently,

$$F'(x^{(k)})^{-1} = [I + B_1 e^{(k)} + B_2 e^{(k)^2} + B_3 e^{(k)^3} + B_4 e^{(k)^4} + B_5 e^{(k)^5}] F'(x^*)^{-1} + O(e^{(k)^6}), \quad (7)$$

where $B_1 = -2A_2$, $B_2 = -3A_3 + 4A_2^2$, $B_3 = -4A_4 + 6A_2A_3 + 6A_3A_2 - 8A_2^3$, $B_4 = -5A_5 + 8A_2A_4 + 9A_3^2 + 8A_4A_2 - 12A_2^2A_3 - 12A_2A_3A_2 - 12A_3A_2^2 + 16A_2^4$, and $B_5 = -6A_6 + 10A_2A_5 + 12A_3A_4 + 12A_4A_3 + 10A_5A_2 - 16A_2^2A_4 - 18A_2A_3^2 - 16A_2A_4A_2 - 18A_3A_2A_3 - 18A_3^2A_2 - 16A_4A_2^2 + 24A_2^3A_3 + 24A_2^2A_3A_2 + 24A_2A_3A_2^2 + 24A_3A_2^3 - 32A_2^5$.

Denoting $e_y^{(k)} = y^{(k)} - x^*$ as the error at the first step of method (4), and using Eqs. (5)–(7), we have that

$$e_y^{(k)} = C_1 e^{(k)^2} + C_2 e^{(k)^3} + C_3 e^{(k)^4} + C_4 e^{(k)^5} + C_5 e^{(k)^6} + O(e^{(k)^7}), \quad (8)$$

where $C_1 = A_2$, $C_2 = 2(A_3 - A_2^2)$, $C_3 = 3A_4 - 4A_2A_3 - 3A_3A_2 + 4A_2^3$, $C_4 = 4A_5 - 6A_2A_4 - 6A_3^2 - 4A_4A_2 + 8A_2^2A_3 + 6A_2A_3A_2 + 6A_3A_2^2 - 8A_2^4$, and $C_5 = 5A_6 - 8A_2A_5 9A_3A_4 - 8A_4A_3 - 5A_5A_2 + 12A_2^2A_4 + 12A_2A_3^2 + 8A_2A_4A_2 + 12A_3A_2A_3 + 9A_3^2A_2 + 8A_4A_2^2 - 16A_2^3A_3 - 12A_2^2A_3A_2 - 12A_2A_3A_2^2 - 12A_3A_2^3 + 16A_2^5$.

Using the expression (8), Taylor developments of $F(y^{(k)})$ and $F'(y^{(k)})$, about x^*, is given by

$$F(y^{(k)}) = F'(x^*)[K_1 e^{(k)^2} + K_2 e^{(k)^3} + K_3 e^{(k)^4} + K_4 e^{(k)^5} + K_5 e^{(k)^6}] + O(e^{(k)^7}), \quad (9)$$

$$F'(y^{(k)}) = F'(x^*)[I + L_1 e^{(k)^2} + L_2 e^{(k)^3} + L_3 e^{(k)^4} + L_4 e^{(k)^5}] + O(e^{(k)^6}), \quad (10)$$

where $K_1 = A_2$, $K_2 = 2(A_3 - A_2^2)$, $K_3 = 3A_4 - 4A_2A_3 - 3A_3A_2 + 5A_2^3$, $K_4 = 4A_5 - 6A_2A_4 - 6A_3^2 - 4A_4A_2 + 10A_2^2A_3 + 8A_2A_3A_2 + 6A_3A_2^2 - 12A_2^4$, $K_5 = 5A_6 - 8A_2A_5 - 9A_3A_4 - 8A_4A_3 - 5A_5A_2 + 15A_2^2A_4 + 16A_2A_3^2 + 11A_2A_4A_2 + 12A_3A_2A_3 + 9A_3^2A_2 + 8A_4A_2^2 - 24A_2^3A_3 - 19A_2^2A_3A_2 - 19A_2A_3A_2^2 - 11A_3A_2^3 + 28A_2^5$, $L_1 = 2A_2^2$, $L_2 = 4(A_2A_3 - A_2^3)$, $L_3 = 6A_2A_4 - 8A_2^2A_3 - 6A_2A_3A_2 + 3A_3A_2^2 + 8A_2^4$, and $L_4 = 8A_2A_5 - 12A_2^2A_4 - 12A_2A_3^2 - 8A_2A_4A_2 + 6A_3A_2A_3 + 6A_3^2A_2 + 16A_2^3A_3 + 12A_2^2A_3A_2 + 12A_2A_3A_2^2 - 12A_3A_2^3 - 16A_2^5$.

Let us denote $e_z^{(k)} = z^{(k)} - x^*$, then using Eqs. (7)–(9), the second step of method (4) yields

$$e_z^{(k)} = M_1 e^{(k)^3} + M_2 e^{(k)^4} + M_3 e^{(k)^5} + M_4 e^{(k)^6} + O(e^{(k)^7}), \quad (11)$$

where $M_1 = 2A_2^2$, $M_2 = 4A_2A_3 + 3A_3A_2 - 9A_2^3$, $M_3 = 6A_2A_4 + 6A_3^2 + 4A_4A_2 - 18A_2^2A_3 - 14A_2A_3A_2 - 12A_3A_2^2 + 30A_2^4$, and $M_4 = 8A_2A_5 + 9A_3A_4 + 8A_4A_3 + 5A_5A_2 - 27A_2^2A_4 - 28A_2A_3^2 - 19A_2A_4A_2 - 24A_3A_2A_3 - 18A_3^2A_2 - 16A_4A_2^2 + 60A_2^3A_3 + 47A_2^2A_3A_2 + 43A_2A_3A_2^2 + 38A_3A_2^3 - 88A_2^5$.

Taylor expansion of $F(z^{(k)})$, using Eq. (11), is established as

$$F(z^{(k)}) = F'(x^*)[P_1 e^{(k)^3} + P_2 e^{(k)^4} + P_3 e^{(k)^5} + P_4 e^{(k)^6}] + O(e^{(k)^7}), \quad (12)$$

where $P_1 = 2A_2^2$, $P_2 = 4A_2A_3 + 3A_3A_2 - 9A_2^3$, $P_3 = 6A_2A_4 + 6A_3^2 + 4A_4A_2 - 18A_2^2A_3 - 14A_2A_3A_2 - 12A_3A_2^2 + 30A_2^4$, and $P_4 = 8A_2A_5 + 9A_3A_4 + 8A_4A_3 + 5A_5A_2 - 27A_2^2A_4 - 28A_2A_3^2 - 19A_2A_4A_2 - 24A_3A_2A_3 - 18A_3^2A_2 - 16A_4A_2^2 + 60A_2^3A_3 + 47A_2^2A_3A_2 + 43A_2A_3A_2^2 + 38A_3A_2^3 - 84A_2^5$.

Consequently, the error equation at the $(k+1)^{th}$ iteration is derived by substituting the expressions of (7), (10), (11), and (12) in the final step of method (4), which is given by the expression

$$e^{(k+1)} = x^{(k+1)} - x^* = Q_1 e^{(k)^3} + Q_2 e^{(k)^4} + Q_3 e^{(k)^5} + Q_4 e^{(k)^6} + O(e^{(k)^7}), \quad (13)$$

where $Q_1 = 2(1-a-b-c)A_2^2$, $Q_2 = (1-a-b-c)(4A_2A_3 + 3A_3A_2) - (9 - 13a - 17b - 21c)A_2^3$, $Q_3 = (1-a-b-c)(6A_2A_4 + 6A_3^2 + 4A_4A_2) - 2(9 - 13a - 17b - 21c)A_2^2A_3 - 2(7 - 10a - 13b - 16c)A_2A_3A_2 - 6(2 - 3a - 4b - 5c)A_3A_2^2 + 2(15 - 28a - 47b - 70c)A_2^4$, and the expression of Q_4, being lengthy, is not shown explicitly here.

Ultimately, there should be an optimum selection of parameters' values so as to achieve the maximum possible convergence speed for the proposed scheme. In that sense, if we choose $a = \frac{7}{2}$, $b = -4$, and $c = \frac{3}{2}$, then the coefficients Q_1, Q_2, and Q_3 in Eq. (13) vanish. Further, the error equation is reduced to

$$e^{(k+1)} = 2(A_2A_3A_2^2 - 3A_3A_2^3 + 18A_2^5)e^{(k)^6} + O(e^{(k)^7}).$$

Hence, the sixth order of convergence is proved for the iterative method (4). □

The proposed sixth-order iterative method is finally presented below.

$$\begin{aligned} y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}), \\ z^{(k)} &= y^{(k)} - F'(x^{(k)})^{-1}F(y^{(k)}), \\ x^{(k+1)} &= z^{(k)} - \left[\frac{7}{2}I - 4F'(x^{(k)})^{-1}F'(y^{(k)}) + \frac{3}{2}\left(F'(x^{(k)})^{-1}F'(y^{(k)})\right)^2\right] \\ &\qquad \times F'(x^{(k)})^{-1}F(z^{(k)}). \end{aligned} \quad (14)$$

Clearly, the proposed method utilizes three function evaluations, two Jacobian matrices, and one Jacobian inversion per iteration. For the further reference in this study, the technique (14) is denoted as ϕ_1.

3 Computational Efficiency

Solving nonlinear systems using iterative procedures involves a significantly large number of mathematical calculations or operations. Apart from achieving the high convergence order, an iterative algorithm shall also be evaluated on the basis of its computational aspects. The term computational efficiency relates to the investigation of algorithmic characteristics that how much computing resources it utilizes during its implementation. In what follows, the concept of computational efficiency shall be investigated thoroughly, and further, the analysis shall be carried out in this context for the comparison of the new iterative method with the existing counterparts.

For locating the solution of a nonlinear system using an iterative method, initially, an approximation is selected in the neighborhood of the solution. Then, the iterative process is terminated using a specific criterion, which is generally prescribed as

$$||x^{(k)} - x^*|| \leq \epsilon = 10^{-d},$$

where 'k' is the iteration index, 'ϵ' is the desired precision, and 'd' is the number of significant decimal digits of the obtained approximation. To estimate the number of iterations which are required to achieve the desired accuracy, it is assumed that $||x^{(0)} - x^*|| \approx 10^{-1}$. Then, after the '$k$' number of iterative steps, we have the approximation: $10^{-d} \approx 10^{-r^k}$, and that simply provides the required estimation $k \approx \log d / \log r$, where r is the convergence order. Further, let the computational cost per iteration be represented by 'C', then the completed iterative process constitutes the total computational cost which is equal to 'kC'. The measure of computational efficiency, conventionally known as the efficiency index, is formulated in various manners in the literature. Ostrowski in [8] and Traub in [13] have independently provided this measure in different ways. But, defined in any way, the efficiency index always indicates reciprocal relation with the cost of computation. Therefore, taking into consideration the reciprocal relationship, the efficiency index be evaluated as

$$E = \frac{1}{kC} = \frac{1}{\log d} \frac{\log r}{C}. \tag{15}$$

Consider a m-dimensional function, $F : \mathbb{R}^m \to \mathbb{R}^m$, $F(x) = (f_1(x), ..., f_m(x))^T$, where $x = (x_1, x_2, ..., x_m)^T$, then the estimation of computational cost per iteration is given by the formulation,

$$C(m, \eta_0, \eta_1, \mu) = N_0(m)\eta_0 + N_1(m)\eta_1 + N(m, \mu), \tag{16}$$

where $N_0(m)$ and $N_1(m)$ represent the number of evaluations of scalar functions in the computation of F and F', respectively, and $N(m, \mu)$ stands for the number of product or quotient evaluations per iteration. The ratios $\eta_0 > 0$ and $\eta_1 > 0$, which interrelate the costs of products and functional evaluations, and a ratio $\mu > 1$, interrelating costs of products and quotients, are the necessary parameters in order to express $C(m, \eta_0, \eta_1, \mu)$ in terms of product units. Let us note that evaluations of m and m^2 scalar functions are required, respectively, to compute a function F and a derivative F'. Additionally, to compute an inverse linear operator, and eventually to evaluate $F'^{-1}F$, the technique of LU decomposition is employed that involves $m(m-1)(2m-1)/6$ products and $m(m-1)/2$ quotients, which is followed by the resolution of two triangular linear systems requiring $m(m-1)$ products and m quotients. Further, m products for scalar-vector multiplication and m^2 products for matrix-vector multiplication must be taken into account.

With the purpose to analyze and compare the efficiency of the developed method, we have included the existing sixth-order methods developed by Bahl et al. [2], Cordero et al. [4], Esmaeili and Ahmadi [5], Lofti et al. [6], Soleymani et al. [12], and Wang et al. [14]. For the ready reference, these methods are expressed below, which are denoted by ϕ_i, where $i = 2, 3, \ldots, 7$.

Method by Bahl et al. (ϕ_2):

$$\begin{aligned} y^{(k)} &= x^{(k)} - \frac{2}{3}F'(x^{(k)})^{-1}F(x^{(k)}), \\ z^{(k)} &= x^{(k)} - \left[I + \frac{3}{4}[I - F'(x^{(k)})^{-1}F'(y^{(k)})] + \frac{9}{8}[I - F'(x^{(k)})^{-1}F'(y^{(k)})]^2\right] \\ &\qquad \times F'(x^{(k)})^{-1}F(x^{(k)}), \\ x^{(k+1)} &= z^{(k)} - 2[3F'(y^{(k)}) - F'(x^{(k)})]^{-1}F(z^{(k)}). \end{aligned}$$

Method by Cordero et al. (ϕ_3):

$$\begin{aligned} y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}), \\ z^{(k)} &= y^{(k)} - F'(x^{(k)})^{-1}[2I - F'(y^{(k)})F'(x^{(k)})^{-1}]F(y^{(k)}), \\ x^{(k+1)} &= z^{(k)} - F'(y^{(k)})^{-1}F(z^{(k)}). \end{aligned}$$

Method by Esmaeili and Ahmadi (ϕ_4):

$$\begin{aligned} y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}), \\ z^{(k)} &= y^{(k)} + \frac{1}{3}\big[F'(x^{(k)})^{-1} + 2[F'(x^{(k)}) - 3F'(y^{(k)})]^{-1}\big]F(x^{(k)}), \\ x^{(k+1)} &= z^{(k)} + \frac{1}{3}\big[-F'(x^{(k)})^{-1} + 4[F'(x^{(k)}) - 3F'(y^{(k)})]^{-1}\big]F(z^{(k)}). \end{aligned}$$

Method by Lofti et al. (ϕ_5):

$$\begin{aligned}
y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}),\\
z^{(k)} &= x^{(k)} - 2[F'(x^{(k)}) + F'(y^{(k)})]^{-1}F(x^{(k)}),\\
x^{(k+1)} &= z^{(k)} - \left[\frac{7}{2}I - 4F'(x^{(k)})^{-1}F'(y^{(k)}) + \frac{3}{2}(F'(x^{(k)})^{-1}F'(y^{(k)}))^2\right]\\
&\qquad \times F'(x^{(k)})^{-1}F(z^{(k)}).
\end{aligned}$$

Method by Soleymani et al. (ϕ_6):

$$\begin{aligned}
y^{(k)} &= x^{(k)} - \frac{2}{3}F'(x^{(k)})^{-1}F(x^{(k)}),\\
z^{(k)} &= x^{(k)} - \frac{1}{2}[3F'(y^{(k)}) - F'(x^{(k)})]^{-1}[3F'(y^{(k)}) + F'(x^{(k)})]F'(x^{(k)})^{-1}F(x^{(k)}),\\
x^{(k+1)} &= z^{(k)} - \left[\frac{1}{2}[3F'(y^{(k)}) - F'(x^{(k)})]^{-1}[3F'(y^{(k)}) + F'(x^{(k)})]\right]^2\\
&\qquad \times F'(x^{(k)})^{-1}F(z^{(k)}).
\end{aligned}$$

Method by Wang et al. (ϕ_7):

$$\begin{aligned}
y^{(k)} &= x^{(k)} - \frac{2}{3}F'(x^{(k)})^{-1}F(x^{(k)}),\\
z^{(k)} &= x^{(k)} - [6F'(y^{(k)}) - 2F'(x^{(k)})]^{-1}[3F'(y^{(k)}) + F'(x^{(k)})]F'(x^{(k)})^{-1}F(x^{(k)}),\\
x^{(k+1)} &= z^{(k)} - \frac{1}{2}[3F'(y^{(k)})^{-1} - F'(x^{(k)})^{-1}]F(z^{(k)}).
\end{aligned}$$

Denoting the computational costs and the efficiency indices, respectively, by C_i and E_i, $i = 1, 2, \ldots, 7$, and then taking into account the mathematical operations or computations described above, the computational costs and the corresponding efficiency indices are expressed as follows:

$$\begin{aligned}
C_1 &= 3m\eta_0 + 2m^2\eta_1 + \frac{m}{6}(2m^2 + 39m - 11 + 3\mu(9+m)) \;\text{ and }\; E_1 = \frac{1}{D}\frac{\log 6}{C_1}.\\
C_2 &= 2m\eta_0 + 2m^2\eta_1 + \frac{m}{3}(2m^2 + 18m + 4 + 3\mu(3+m)) \;\text{ and }\; E_2 = \frac{1}{D}\frac{\log 6}{C_2}.\\
C_3 &= 3m\eta_0 + 2m^2\eta_1 + \frac{m}{3}(2m^2 + 12m - 8 + 3\mu(3+m)) \;\text{ and }\; E_3 = \frac{1}{D}\frac{\log 6}{C_3}.\\
C_4 &= 2m\eta_0 + 2m^2\eta_1 + \frac{m}{3}(2m^2 + 12m + 1 + 3\mu(3+m)) \;\text{ and }\; E_4 = \frac{1}{D}\frac{\log 6}{C_4}.\\
C_5 &= 2m\eta_0 + 2m^2\eta_1 + \frac{m}{3}(2m^2 + 18m - 2 + 3\mu(4+m)) \;\text{ and }\; E_5 = \frac{1}{D}\frac{\log 6}{C_5}.\\
C_6 &= 2m\eta_0 + 2m^2\eta_1 + \frac{m}{3}(2m^2 + 24m - 5 + 3\mu(4+m)) \;\text{ and }\; E_6 = \frac{1}{D}\frac{\log 6}{C_6}.
\end{aligned}$$

$$C_7 = 2m\eta_0 + 2m^2\eta_1 + \frac{m}{2}(2m^2 + 9m + 1 + \mu(5 + 3m)) \text{ and } E_7 = \frac{1}{D}\frac{\log 6}{C_7}.$$

Here $D = \log d$.

3.1 Comparison of Efficiencies

Consider a ratio, for the comparison of iterative methods, say ϕ_i versus ϕ_j, which is defined as

$$\Pi_j^i = \frac{E_i}{E_j} = \frac{C_j \log(r_i)}{C_i \log(r_j)}, \tag{17}$$

where r_i and r_j, respectively, are the orders of convergence of the methods ϕ_i and ϕ_j. Clearly, if $\Pi_j^i > 1$ holds, then ϕ_i will be more efficient than ϕ_j, and we symbolize it as $\phi_i \gg \phi_j$. The proposed method, $\boldsymbol{\phi}_1$, shall be compared analytically as well as geometrically with the existing methods, ϕ_i $(i = 2, 3, \ldots, 7)$, which are already presented above. The analytical way of comparison is the resolution of inequality $\Pi_i^1 > 1$ for each $i = 2, 3, \ldots, 7$, and the results obtained are presented geometrically by projecting the boundary lines $\Pi_i^1 = 1$, in (η_1, η_0)-plane, corresponding to the special cases of $m = 5,\ 10,\ 25,$ and 50, and fixing $\mu = 3$ in each case. Let us note here that each line will divide the plane into two parts, where $\boldsymbol{\phi}_1 \gg \phi_i$ on one side, whereas $\phi_i \gg \boldsymbol{\phi}_1$ on the other.

In view of the above discussion, we now present the comparison analysis through the following theorem:

Theorem 2 *For all $\eta_0 > 0$, $\eta_1 > 0$, and $\mu > 1$, we have that*

(i) $E_1 > E_2$, *for* $\eta_0 < \frac{1}{6}(2m^2 - 3m + 19 + 3\mu(m - 3))$.
(ii) $E_1 > E_3$ *for* $m \geq 7$, *and* $E_1 < E_3$ *for* $m = 2, 3$, *but otherwise comparison depends on value of* μ.
(iii) $E_1 > E_4$, *for* $\eta_0 < \frac{1}{6}(2m^2 - 15m + 13 + 3\mu(m - 3))$.
(iv) $E_1 > E_5$, *for* $\eta_0 < \frac{1}{6}(2m^2 - 3m + 7 + 3\mu(m - 1))$.
(v) $E_1 > E_6$, *for* $\eta_0 < \frac{1}{6}(2m^2 + 9m + 1 + 3\mu(m - 1))$.
(vi) $E_1 > E_7$, *for* $\eta_0 < \frac{1}{3}(2m^2 - 6m + 7 + 3\mu(m - 2))$.

Proof $\boldsymbol{\phi}_1$ *versus* ϕ_2 *case*:
The ratio in this case is

$$\Pi_2^1 = \frac{2m\eta_0 + 2m^2\eta_1 + \frac{m}{3}(2m^2 + 18m + 4 + 3\mu(3 + m))}{3m\eta_0 + 2m^2\eta_1 + \frac{m}{6}(2m^2 + 39m - 11 + 3\mu(9 + m))}.$$

By resolving of the inequality $\Pi_2^1 > 1$, it is straightforward to deduce that $\eta_0 < \frac{1}{6}(2m^2 - 3m + 19 + 3\mu(m - 3))$, which concludes (i). The boundary lines $\Pi_2^1 = 1$,

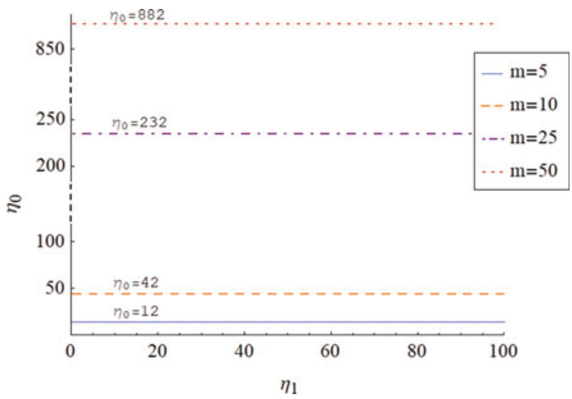

Fig. 1 Boundary lines for comparison of ϕ_1 and ϕ_2

in (η_1, η_0)-plane, are displayed in Fig. 1, where $\phi_1 \gtrdot \phi_2$ in the section which is below the line for each particular case of m.

ϕ_1 *versus* ϕ_3 *case*:

The ratio in this case is

$$\Pi_3^1 = \frac{3m\eta_0 + 2m^2\eta_1 + \frac{m}{3}(2m^2 + 12m - 8 + 3\mu(3+m))}{3m\eta_0 + 2m^2\eta_1 + \frac{m}{6}(2m^2 + 39m - 11 + 3\mu(9+m))}.$$

It is easy to verify that, for $\eta_0 > 0,\ \eta_1 > 0$, and $\mu > 1$, the inequality $\Pi_3^1 > 1$ holds for $m \geq 7$, and $\Pi_3^1 < 1$ holds only for $m = 2, 3$. For $4 \leq m \leq 6$, the inequality $\Pi_3^1 > 1$ holds when $\mu > \frac{2m^2-15m-5}{9-3m}$, and this eventually proves (ii). So, we conclude here that $\phi_1 \gtrdot \phi_3$ for all $m \geq 7$, whereas $\phi_1 \lessdot \phi_3$ for $m = 2, 3$, but otherwise, comparison depends on the value of μ.

ϕ_1 *versus* ϕ_4 *case*:

The ratio in this case is

$$\Pi_4^1 = \frac{2m\eta_0 + 2m^2\eta_1 + \frac{m}{3}(2m^2 + 12m + 1 + 3\mu(3+m))}{3m\eta_0 + 2m^2\eta_1 + \frac{m}{6}(2m^2 + 39m - 11 + 3\mu(9+m))}.$$

Resolution of the inequality $\Pi_4^1 > 1$ results into $\eta_0 < \frac{1}{6}(2m^2 - 15m + 13 + 3\mu(m-3))$, which concludes (iii). The boundary lines for this comparison, in (η_1, η_0)-plane, are shown in Fig. 2, where $\phi_1 \gtrdot \phi_4$ on the lower region of line for each case of m.

ϕ_1 *versus* ϕ_5 *case*:

The ratio in this case is

$$\Pi_5^1 = \frac{2m\eta_0 + 2m^2\eta_1 + \frac{m}{3}(2m^2 + 18m - 2 + 3\mu(4+m))}{3m\eta_0 + 2m^2\eta_1 + \frac{m}{6}(2m^2 + 39m - 11 + 3\mu(9+m))}.$$

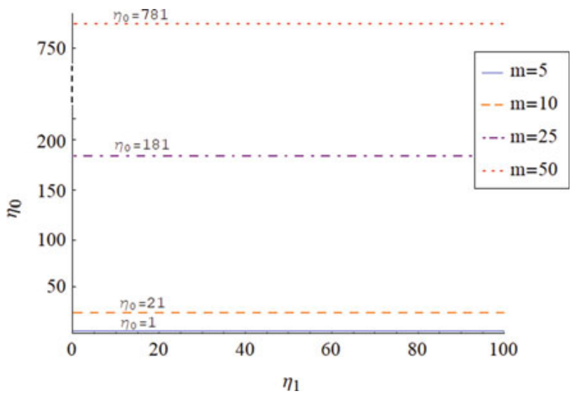

Fig. 2 Boundary lines for comparison of ϕ_1 and ϕ_4

The inequality $\Pi_5^1 > 1$ simply resolves into relation $\eta_0 < \frac{1}{6}(2m^2 - 3m + 7 + 3\mu(m-1))$, and this proves (iv). In this comparison, the boundary lines are displayed in Fig. 3, where $\phi_1 > \phi_5$ holds on the lower section of line for each particular case.

ϕ_1 versus ϕ_6 case:

The ratio in this case is

$$\Pi_6^1 = \frac{2m\eta_0 + 2m^2\eta_1 + \frac{m}{3}(2m^2 + 24m - 5 + 3\mu(4+m))}{3m\eta_0 + 2m^2\eta_1 + \frac{m}{6}(2m^2 + 39m - 11 + 3\mu(9+m))}.$$

It is straightforward to establish the relation $\eta_0 < \frac{1}{6}(2m^2 + 9m + 1 + 3\mu(m-1))$ by resolving $\Pi_6^1 > 1$, which eventually proves (v). The boundary lines, in this case, are presented in Fig. 4 with $\phi_1 > \phi_6$ on the lower side of each line.

ϕ_1 versus ϕ_7 case:

The ratio in this case is

$$\Pi_7^1 = \frac{2m\eta_0 + 2m^2\eta_1 + \frac{m}{2}(2m^2 + 9m + 1 + \mu(5+3m))}{3m\eta_0 + 2m^2\eta_1 + \frac{m}{6}(2m^2 + 39m - 11 + 3\mu(9+m))}.$$

Resolution of the inequality $\Pi_7^1 > 1$ results into the relation $\eta_0 < \frac{1}{3}(2m^2 - 6m + 7 + 3\mu(m-2))$. This concludes (vi), and the boundary lines for this case are shown in Fig. 5, where $\phi_1 > \phi_7$ in the region which is below each boundary line. □

From the above comparison analysis, it can be clearly observed that the proposed iterative method shows an increase in the efficiency index with the increasing values of m. We conclude this section with a note that, as large as the system is, the proposed sixth-order method exhibits superiority over the existing methods in the subject of computational efficiency.

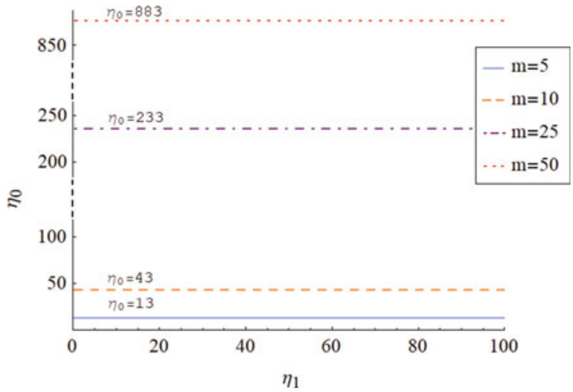

Fig. 3 Boundary lines for comparison of ϕ_1 and ϕ_5

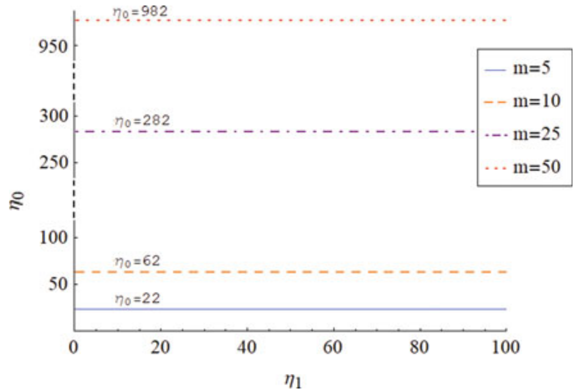

Fig. 4 Boundary lines for comparison of ϕ_1 and ϕ_6

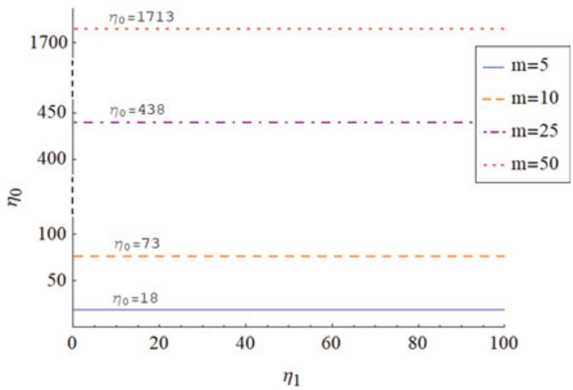

Fig. 5 Boundary lines for comparison of ϕ_1 and ϕ_7

Table 1 CPU time and computational cost for the execution of elementary operations

Functions	$x * y$	x/y	$\sqrt{x}$	e^x	$\log(x)$	$\sin(x)$	$\cos(x)$	$\arctan(x)$
CPU time	0.0172	0.0484	0.0234	1.5562	1.3469	1.6938	1.6896	2.9797
Cost	1	2.81	1.36	90.48	78.31	98.48	98.23	173.24

Here $x = \sqrt{3} - 1$ *and* $y = \sqrt{5}$ (*with* 4096 *digits of accuracy*)

4 Numerical Experimentation

In this section, the numerical experimentation shall be executed to assess the performance of the developed method. The nonlinear problems arising in different physical situations have been selected for this purpose. Moreover, to arrive at some valid conclusion, the outcomes of this testing need to be analyzed and further compared with the corresponding results of the existing methods. Two of the most significant factors which contribute toward the numerical performance of an iterative technique are (i) Stability and (ii) CPU time elapsed during its execution on the digital platform. Let us note that all the numerical computations, in our case, are being executed using the software *Mathematica* [15] installed on the computer equipped with specifications: Intel(R) Core (TM) i5-9300H processor and Windows 10 operating system.

In what follows, the comparison analysis shall be illustrated by locating the solutions of nonlinear problems, and for the termination of iterations, the stopping criterion being employed is described as follows:

$$\|x^{(k)} - x^{(k-1)}\| + \|F(x^{(k)})\| < 10^{-100}.$$

In addition, the approximated computational order of convergence (ACOC) is required to validate the convergence order established by analytical means, which is computed by the formula (see [5]),

$$\text{ACOC} = \frac{\ln\left(\|x^{(k)} - x^{(k-1)}\| / \|x^{(k-1)} - x^{(k-2)}\|\right)}{\ln\left(\|x^{(k-1)} - x^{(k-2)}\| / \|x^{(k-2)} - x^{(k-3)}\|\right)}.$$

To make connection between the computational efficiency and the performance of technique, it is necessary to estimate the parameters, η_0, η_1, and μ, as defined in Sect. 3. These parameters are essential to express the mathematical operations and functional evaluations in terms of product units. In order to achieve this, Table 1 displays the CPU time elapsed during the execution of elementary mathematical operations and their estimated cost of computation in units of products. Note that the estimated cost of division is approximately thrice the cost of the product.

Now, we consider the following nonlinear problems to demonstrate the performance analysis and display results in respect of the following: (i) Number of iterations (k), (ii) ACOC, (iii) Computational cost (C_i), (iv) Efficiency index (E_i), and

Table 2 Comparison of performance of methods for Problem 1

Method	k	ACOC	C_i	E_i	CPU time
ϕ_1	4	5.993	145.58	1230.77	0.0260
ϕ_2	4	5.996	152.58	1174.31	0.0313
ϕ_3	4	5.996	129.58	1382.74	0.0363
ϕ_4	4	5.989	131.58	1361.73	0.0417
ϕ_5	4	5.994	155.01	1155.90	0.0310
ϕ_6	4	5.999	170.01	1053.91	0.0363
ϕ_7	4	5.996	154.01	1163.40	0.0467

(v) Elapsed CPU time (in seconds). To illustrate the efficiency indices of techniques, we have conveniently chosen $D = 10^{-5}$ for each of the problems.

Problem 1 Starting with the system of three nonlinear equations:

$$x^2 + y^2 + z^2 = 1,$$
$$2x^2 + y^2 + 4z = 0,$$
$$3x^2 - 4y^2 + z^2 = 0,$$

the initial estimate is taken as $\left(-\frac{3}{2}, -\frac{3}{2}, -\frac{3}{2}\right)^T$ to locate the particular solution,

$$x^* = (-0.6982..., -0.6285..., -0.3425...)^T.$$

For this particular problem, the parameters used in the equation (16) are estimated as $(m, \eta_0, \eta_1, \mu) = (3, 2.33, 0.67, 2.81)$. Numerical results for the comparison are displayed in Table 2.

Problem 2 Consider the nonlinear integral equation (see [1]),

$$u(t) = \frac{7}{8}t + \frac{1}{2}\int_0^1 t\ s\ u(s)^2 ds, \tag{18}$$

where $t \in [0, 1]$, and $u \in C[0, 1]$, with $C[0, 1]$ being a space of all continuous functions defined on the interval [0, 1].

The given integral equation can be transformed into a finite-dimensional problem by partitioning the given interval [0, 1] uniformly as follows:

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

where $h = 1/k$ is the sub-interval length. Approximating the integral, appearing in the equation (18), using the trapezoidal rule of integration, and denoting $u(t_i) = u_i$ for each i, we obtain the system of nonlinear equations as

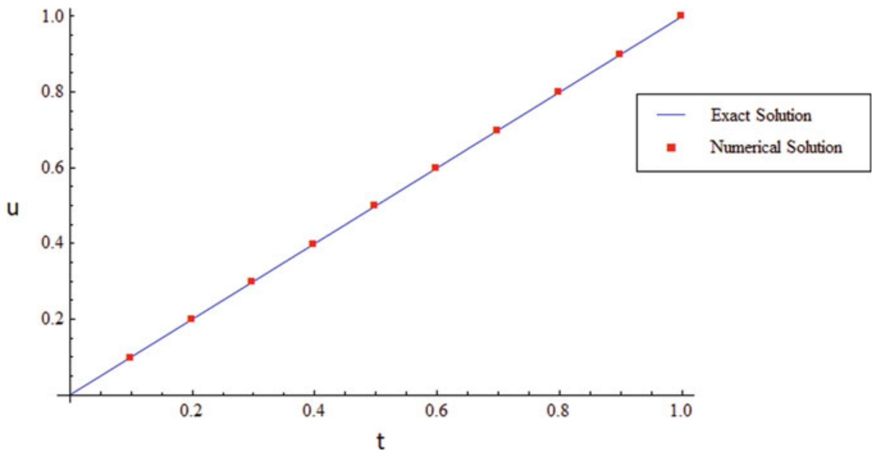

Fig. 6 Graphical comparison of exact and numerical solution of Problem 2

Table 3 Comparison of performance of methods for Problem 2

Method	k	ACOC	C_i	E_i	CPU time
ϕ_1	3	6.000	1521.95	117.73	0.141
ϕ_2	3	6.000	1905.30	94.04	0.188
ϕ_3	3	6.000	1695.30	105.69	0.177
ϕ_4	3	6.000	1695.30	105.69	0.162
ϕ_5	3	6.000	1913.40	93.64	0.187
ϕ_6	3	6.000	2103.40	85.18	0.203
ϕ_7	3	6.000	2206.75	81.19	0.235

$$\frac{7}{8}t_i - u_i + \frac{ht_i}{2}\left(\frac{1}{2}u_k^2 + \sum_{j=1}^{k-1} s_j u_j^2\right) = 0, \quad (i = 1, 2, \ldots, k), \tag{19}$$

where $t_i = s_i = i/k$ for each i.

We solve this problem in particular by taking $k = 10$. Setting the initial approximation as $(\frac{1}{2}, \overset{10}{\cdots\cdots}, \frac{1}{2})^T$, the approximate numerical solution of the reduced system (19) is obtained as,

$$x^* = (0.1001..., 0.2003...,0.3004..., 0.4006..., 0.5008..., 0.6009...,$$
$$0.7011..., 0.8013..., 0.9014..., 1.0016...)^T.$$

The numerical solution, so obtained, is compared graphically with the exact solution in Fig. 6, and further, numerical results are depicted in Table 3. Moreover, the parameters of Eq. (16) are estimated as $(m, \eta_0, \eta_1, \mu) = (10, 3, 1, 2.81)$.

Table 4 Comparison of performance of methods for Problem 3

Method	k	ACOC	C_i	E_i	CPU time
ϕ_1	4	6.000	6.27E+04	2.86	1.385
ϕ_2	4	6.000	1.06E+05	1.68	2.401
ϕ_3	4	6.000	1.01E+05	1.77	2.271
ϕ_4	4	6.000	1.01E+05	1.77	2.327
ϕ_5	4	6.000	1.06E+05	1.68	2.344
ϕ_6	4	6.000	1.11E+05	1.61	2.250
ϕ_7	4	6.000	1.48E+05	1.21	3.344

Problem 3 Consider the boundary value problem (see [3]), which models the finite deflections of an elastic string under the transverse load, and it is presented as follows:

$$u''(t) + a^2(u'(t))^2 + 1 = 0, \quad u(0) = 0, \quad u(1) = 0, \tag{20}$$

where 'a' is a parameter. The exact solution of the given problem is $u(t) = \frac{1}{a^2}\ln\left(\frac{\cos(a(t-1/2))}{\cos(a/2)}\right)$. We intend to remodel the problem (20) into a finite-dimensional problem by considering the partition of [0, 1], with equal sub-interval length $h = 1/k$, as

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

Denoting $u(t_i) = u_i$ for each $i = 1, 2, \ldots, k-1$, and approximating the derivatives involved in (20) by the second-order divided differences,

$$u_i' = \frac{u_{i+1} - u_{i-1}}{2h}, \quad \text{and} \quad u_i'' = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2},$$

the system of $k-1$ nonlinear equations in $k-1$ variables is obtained as

$$u_{i-1} - 2u_i + u_{i+1} + \frac{a^2}{4}(u_{i+1} - u_{i-1})^2 + h^2 = 0, \quad (i = 1, 2, \ldots, k-1),$$

where $u_0 = 0$ and $u_k = 0$ are the transformed boundary conditions. In particular, setting $k = 51$, the above system reduces to 50 nonlinear equations. Further, choosing $a = 2$, and selecting the initial approximation as $(-1, \overset{50}{\cdots\cdots}, -1)^T$, the approximate numerical solution so obtained, along with the exact solution, is plotted in Fig. 7. Further, the numerical performance of the methods is displayed in Table 4. The estimated values of parameters, used in Eq. (16), are given as $(m, \eta_0, \eta_1, \mu) = (50, 2, 0.078, 2.81)$.

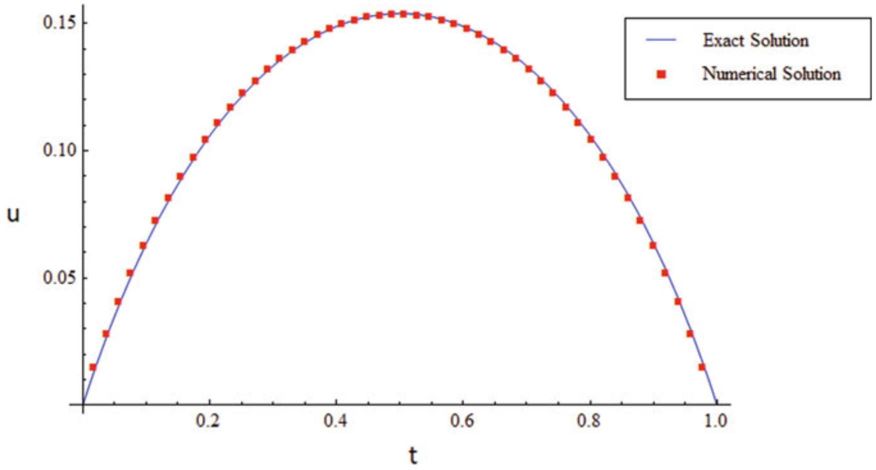

Fig. 7 Graphical comparison of exact and numerical solution of Problem 3

Table 5 Comparison of performance of methods for Problem 4

Method	k	ACOC	C_i	E_i	CPU time
ϕ_1	5	6.000	4.67E+05	0.384	45.53
ϕ_2	5	6.000	7.92E+05	0.226	74.14
ϕ_3	5	6.000	7.89E+05	0.227	75.09
ϕ_4	5	6.000	7.72E+05	0.232	72.88
ϕ_5	5	6.000	7.92E+05	0.226	74.36
ϕ_6	5	6.000	8.12E+05	0.221	75.73
ϕ_7	5	6.000	1.12E+06	0.159	105.05

Problem 4 Now let us take a system of nonlinear equations as follows:

$$\tan^{-1}(x_i) - 1 + 2\left(\sum_{j=1, j\neq i}^{m} x_j^2\right) = 0, \quad i = 1, 2, ..., m.$$

By taking $m = 100$, we select the initial approximation $(1, \overset{100}{\cdots\cdots}, 1)^T$ to obtain the particular solution,

$$x^* = (0.06859..., \overset{100}{\cdots\cdots}, 0.06859...)^T.$$

The estimated values of the parameters in this problem are $(m, \eta_0, \eta_1, \mu) = (100, 175.24, 0.048, 2.81)$. Further, Table 5 exhibits the comparison of the performance of methods.

Table 6 Comparison of performance of methods for Problem 5

Method	k	ACOC	C_i	E_i	CPU time
ϕ_1	3	6.000	4.38E+07	4.09E-03	16.34
ϕ_2	3	6.000	8.56E+07	2.09E-03	19.42
ϕ_3	3	6.000	8.52E+07	2.10E-03	21.59
ϕ_4	3	6.000	8.51E+07	2.10E-03	19.16
ϕ_5	3	6.000	8.56E+07	2.09E-03	19.27
ϕ_6	3	6.000	8.61E+07	2.08E-03	20.45
ϕ_7	3	6.000	1.27E+08	1.41E-03	24.33

Problem 5 At last, we consider a large system of equations:

$$\begin{aligned} & x_i + \log(2 + x_i + x_{i+1}) = 0, \quad i = 1, 2, ..., m-1, \\ \text{and} \quad & x_m + \log(2 + x_m + x_1) = 0, \end{aligned}$$

where $m = 500$. The above given system has a particular solution,

$$x^* = \left(-0.3149..., \overset{500}{\cdots\cdots}, -0.3149...\right)^T,$$

and to obtain this solution, the initial estimate is taken as $\left(\frac{1}{10}, \overset{500}{\cdots\cdots}, \frac{1}{10}\right)^T$. Numerical results for the performance of methods are depicted in Table 6. Further, the values of parameters are estimated as

$$(m, \eta_0, \eta_1, \mu) = (500, 78.31, 0.0056, 2.81).$$

The findings of numerical experimentation signify the efficient and stable nature of the proposed sixth-order method. The results are remarkable with respect to the efficiency index and CPU time, and certainly favor the new method over its existing counterparts. Furthermore, computation of ACOC validates the theoretically established convergence order.

5 Conclusions

A three-step iterative technique, involving some undetermined parameters, has been designed for the solution of nonlinear equations. The methodology to design the technique is based on the objective to accelerate the convergence rate of the well-known third-order Potra-Pták scheme. Analysis of convergence leads to establishing the sixth order of convergence for a particular set of parametric values. Utilizing only a single Jacobian inversion per iteration, the proposed iterative method exhibits highly economical nature when analyzed in the context of computational complexity. This is affirmed by comparing the computational efficiency of the new method, by analytical as well as visual approach, with the efficiencies of existing methods. Further, numerical performance is examined by locating the solutions of some selected nonlinear problems. The findings of this testing clearly indicate the superiority of the proposed technique over its existing counterparts, especially for large-scale nonlinear systems.

References

1. Avazzadeh, Z., Heydari, M., Loghmani, G.B.: Numerical solution of Fredholm integral equations of the second kind by using integral mean value theorem. Appl. Math. Model. **35**, 2374–2383 (2011). https://doi.org/10.1016/j.apm.2010.11.056
2. Bahl, A., Cordero, A., Sharma, R., Torregrosa, J.R.: A novel bi-parametric sixth order iterative scheme for solving nonlinear systems and its dynamics. Appl. Math. Comput. **357**, 147–166 (2019). https://doi.org/10.1016/j.amc.2019.04.003
3. Cordero, A., Hueso, J.L., Martínez, E., Torregrosa, J.R.: Efficient high-order methods based on golden ratio for nonlinear systems. Appl. Math. Comput. **217**, 4548–4556 (2011). https://doi.org/10.1016/j.amc.2010.11.006
4. Cordero, A., Hueso, J.L., Martínez, E., Torregrosa, J.R.: Increasing the convergence order of an iterative method for nonlinear systems. Appl. Math. Lett. **25**, 2369–2374 (2012). https://doi.org/10.1016/j.aml.2012.07.005
5. Esmaeili, H., Ahmadi, M.: An efficient three-step method to solve system of nonlinear equations. Appl. Math. Comput. **266**, 1093–1101 (2015). https://doi.org/10.1016/j.amc.2015.05.076
6. Lotfi, T., Bakhtiari, P., Cordero, A., Mahdiani, K., Torregrosa, J.R.: Some new efficient multipoint iterative methods for solving nonlinear systems of equations. Int. J. Comput. Math. **92**, 1921–1934 (2014). https://doi.org/10.1080/00207160.2014.946412
7. Ortega, J.M., Rheinboldt, W.C.: Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York (1970)
8. Ostrowski, A.M.: Solution of Equation and Systems of Equations. Academic Press, New York (1960)
9. Potra, F.A., Pták, V.: On a class of modified Newton processes. Numer. Funct. Anal. Optim. **2**, 107–120 (1980). https://doi.org/10.1080/01630568008816049
10. Sharma, J.R., Gupta, P.: An efficient fifth order method for solving systems of nonlinear equations. Comput. Math. Appl. **67**, 591–601 (2014). https://doi.org/10.1016/j.camwa.2013.12.004
11. Sharma, R., Sharma, J.R., Kalra, N.: A modified Newton-Özban composition for solving nonlinear systems. Int. J. Comput. Methods **17**, 1950047 (2020). https://doi.org/10.1142/S0219876219500476
12. Soleymani, F., Lotfi, T., Bakhtiari, P.: A multi-step class of iterative methods for nonlinear systems. Optim. Lett. **8**, 1001–1015 (2014). https://doi.org/10.1007/s11590-013-0617-6
13. Traub, J.F.: Iterative Methods for the Solution of Equations. Chelsea Publishing Company, New York (1982)
14. Wang, X., Kou, J., Gu, C.: Semilocal convergence of a sixth-order Jarratt method in Banach spaces. Numer. Algor. **57**, 441–456 (2011). https://doi.org/10.1007/s11075-010-9438-1
15. Wolfram, S.: The Mathematica Book. Wolfram Media, USA (2003)