Computational and Applied Mathematics (2022) 41:46
https://doi.org/10.1007/s40314-021-01739-5

A class of accurate Newton–Jarratt-like methods with applications to nonlinear models

Janak Raj Sharma[1] · **Sunil Kumar**[2]

Received: 17 April 2021 / Revised: 13 November 2021 / Accepted: 10 December 2021 /
Published online: 10 January 2022

Abstract
We propose a class of composite Newton–Jarratt iterative methods with increasing convergence order for approximating the solutions of systems of nonlinear equations. Novelty of the methods is that in each step the order of convergence is increased by an amount of two at the cost of only one additional function evaluation. Moreover, the use of only a single inverse operator in each iteration makes the algorithms computationally more efficient. Theoretical results regarding convergence and computational efficiency are verified through numerical problems, including those that arise from boundary value problems. By way of comparison, it is shown that the novel methods are more efficient than their existing counterparts, especially when applied to solve the large systems of equations.

Keywords Systems of nonlinear equations · Iterative methods · Fast algorithms · Computational efficiency

Mathematics Subject Classification 65H10 · 41A25 · 49M15

1 Introduction

Construction of fixed point iterative methods for solving nonlinear equations or systems of nonlinear equations is an interesting and challenging task in numerical analysis and many applied scientific branches. The importance of this subject has led to the development of many numerical methods, most frequently of iterative nature (see Ortega and Rheinboldt 1970;

Communicated by Justin Wan.

✉ Janak Raj Sharma
jrshira@yahoo.co.in

Sunil Kumar
k_sunil@ch.amrita.edu

[1] Department of Mathematics, Sant Longowal Institute of Engineering and Technology Longowal, Sangrur 148106, India

[2] Department of Mathematics, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Chennai 601103, India

Argyros 2007). With the advancement of computer hardware and software, the problem of solving nonlinear equations by numerical methods has gained an additional importance. In this paper, we consider the problem of approximating a solution t^* of the equation $\mathrm{F}(t) = 0$, where $\mathrm{F} : \Omega \subset \mathbb{R}^m \to \mathbb{R}^m$, by iterative methods of a high order of convergence. The solution t^* can be obtained as a fixed point of some function $\phi : \Omega \subset \mathbb{R}^m \to \mathbb{R}^m$ by means of fixed point iteration

$$t_{k+1} = \phi(t_k), \quad k = 0,\ 1,\ 2, \ldots.$$

There are a variety of iterative methods for solving nonlinear equations. A basic method is the well-known quadratically convergent Newton's method (Argyros 2007)

$$t_{k+1} = M_1^{(2)}(t_k) = t_k - \mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k), \tag{1}$$

where $\mathrm{F}'(t)^{-1}$ is the inverse of Fréchet derivative $\mathrm{F}'(t)$ of the function $\mathrm{F}(t)$. This method converges if the initial approximation t_0 is closer to solution t^* and $\mathrm{F}'(t)^{-1}$ exists in the neighborhood Ω of t^*. To achieve the higher order of convergence, a number of modified Newton's or Newton-like methods have been proposed in the literature, see, for example (Cordero and Torregrosa 2006; Babajee et al. 2010; Behl et al. 2017; Homeier 2004; Darvishi and Barati 2007; Cordero and Torregrosa 2007; Cordero et al. 2010, 2012; Esmaeili and Ahmadi 2015; Noor and Waseem 2009; Alzahrani et al. 2018; Xiao and Yin 2017, 2018; Sharma et al. 2016; Lotfi et al. 2015; Choubey et al. 2018) and references therein. Throughout this paper, we use $M_i^{(p)}$ to denote an i-th iteration method of convergence order p.

The principal goal and motivation in developing iterative methods is to achieve convergence order as high as possible by utilizing number of evaluations as small as possible so that the methods may possess high efficient character. The most obvious barrier in the development of efficient methods is the evaluation of inverse of a matrix since it requires a lengthy calculation work. Therefore, it will turn out to be judicious if we use minimum number of such inversions as possible (Argyros and Regmi 2019; Regmi 2021). With these considerations, here we propose multi-step iterative methods with increasing order of convergence. First, we present a cubically convergent two-step scheme with first step as Jarratt iteration and second is Newton-like iteration. Then on the basis of this scheme a three-step scheme of fifth-order convergence is proposed. Furthermore, in quest of more fast convergence the scheme is generalized to $q+1$ Newton–Jarratt steps with increasing convergence order $2q+1$ ($q \in \mathbb{N}$). The novel feature is that in each step the order of convergence is increased by an amount of two at the cost of only one additional function evaluation. Evaluation of the inverse operator $\mathrm{F}'(t)^{-1}$ remains the same throughout which also points to the name 'methods with frozen inverse operator'.

Rest of the paper is structured as follows. Some basic definitions relevant to the present work are provided in Sect. 2. Section 3 includes development of the third- and fifth-order methods with their analysis of convergence. Then the generalized version consisting of $q+1$-step scheme with convergence order $2q+1$ is presented in Sect. 4. The computational efficiency is discussed and compared with the existing methods in Sect. 5. In Sect. 6, various numerical examples are considered to confirm the theoretical results. Concluding remarks are given in Sect. 7.

2 Basic definitions

2.1 Order of convergence

Let $\{t_k\}_{k\geq 0}$ be a sequence in $\mathbb{R}^m$ which converges to t^*. Then convergence is called of order p, $p > 1$, if there exists M, $M > 0$, and k_0 such that

$$\|t_{k+1} - t^*\| \leq M\|t_k - t^*\|^p \quad \text{for all} \quad k \geq k_0$$

or

$$\|\epsilon_{k+1}\| \leq M\|\epsilon_k\|^p \quad \text{for all} \quad k \geq k_0,$$

where $\epsilon_k = t_k - t^*$. The convergence is called linear if $p = 1$ and there exists M such that $0 < M < 1$.

2.2 Error equation

Let $\epsilon_k = t_k - t^*$ be the error in the k-th iteration, we call the relation

$$\epsilon_{k+1} = L\epsilon_k^p + O(\epsilon_k^{p+1}),$$

as the error equation. Here, p is the order of convergence, L is a p-linear function, i.e. $L \in \mathcal{L}(\mathbb{R}^m \times \overset{p-times}{\cdots} \times \mathbb{R}^m, \mathbb{R}^m)$, $\mathcal{L}$ denotes the set of bounded linear functions.

2.3 Computational order of convergence

Let t^* be a zero of the function F and suppose that t_{n-1}, t_n, t_{n+1} and t_{n+2} are the four consecutive iterations close to t^*. Then, the computational order of convergence (COC) can be approximated using the formula (see Weerkoon and Fernando 2000)

$$\text{COC} = \frac{\log\left(\|t_{k+2} - t_{k+1}\|/\|t_{k+1} - t_k\|\right)}{\log\left(\|t_{k+1} - t_k\|/\|t_k - t_{k-1}\|\right)}.$$

2.4 Computational efficiency

Computational efficiency of an iterative method is measured by the efficiency index $E = p^{1/C}$ or $E = \frac{\log p}{C}$ (see Ostrowski 1960), where p is the order of convergence and C is the computational cost per iteration.

3 Formulation of basic methods

In what follows first we will introduce the basic third- and fifth-order iterative methods. These are called basic methods since they pave the way for the generalized algorithm to be presented in next section.

3.1 Third-order scheme

Our aim is to develop a method which accelerates the convergence rate of Newton method (1) using minimum number of function evaluations and inverse operators. Thus, it will turn out to be judicious if we consider the two-step iteration scheme of the type

$$\begin{aligned} w_k &= t_k - \frac{2}{3}\mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k), \\ t_{k+1} &= w_k - (a_1\,\mathrm{I} + a_2 Q_k)\mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k), \end{aligned} \tag{2}$$

where $Q_k = \mathrm{F}'(t_k)^{-1}\mathrm{F}'(w_k)$, a_1 and a_2 are arbitrary constants and I is an $m \times m$ identity matrix. Idea of the first step is taken from the well-known Jarratt method (Cordero et al. 2010) whereas the second step is based on Newton-like iteration.

We introduce some known notations and results (Cordero et al. 2010), which are needed to obtain the convergence order of new method. Let $\mathrm{F} : \Omega \subseteq \mathbb{R}^m \to \mathbb{R}^m$ be sufficiently differentiable in Ω. The q-th derivative of F at $u \in \mathbb{R}^m$, $q \geq 1$, is the q-linear function $\mathrm{F}^{(q)}(u) : \mathbb{R}^m \times \cdots \times \mathbb{R}^m \to \mathbb{R}^m$ such that $\mathrm{F}^{(q)}(u)(v_1, \ldots, v_q) \in \mathbb{R}^m$. It is easy to observe that

(i) $\mathrm{F}^{(q)}(u)(v_1, \ldots, v_q) \in \mathcal{L}(\mathbb{R}^m)$,
(ii) $\mathrm{F}^{(q)}(u)(v_{\sigma(1)}, \ldots, v_{\sigma(q)}) = \mathrm{F}^{(q)}(u)(v_1, \ldots, v_q)$, for all permutation σ of $\{1, 2, \ldots, q\}$.

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

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

On the other hand, for $t^* + h \in \mathbb{R}^m$ lying in a neighborhood of a solution t^* of $\mathrm{F}(t) = 0$, we can apply Taylor's expansion and assuming that the Jacobian matrix $\mathrm{F}'(t^*)$ is nonsingular, we have

$$\mathrm{F}(t^* + h) = \mathrm{F}'(t^*)\left[h + \sum_{q=2}^{p-1} K_q h^q\right] + O(h^p),$$

where $K_q = \frac{1}{q!}\mathrm{F}'(t^*)^{-1}\mathrm{F}^{(q)}(t^*)$, $q \geq 2$. We observe that $K_q h^q \in \mathbb{R}^m$ since $\mathrm{F}^{(q)}(t^*) \in \mathcal{L}(\mathbb{R}^m \times \cdots \times \mathbb{R}^m, \mathbb{R}^m)$ and $\mathrm{F}'(t^*)^{-1} \in \mathcal{L}(\mathbb{R}^m)$. In addition, we can express F' as

$$\mathrm{F}'(t^* + h) = \mathrm{F}'(t^*)\left[\mathrm{I} + \sum_{q=2}^{p-1} q\,K_q h^{q-1}\right] + O(h^{p-1}), \tag{3}$$

where I is the identity matrix. Therefore, $q\,K_q h^{q-1} \in \mathcal{L}(\mathbb{R}^m)$. From (3), we obtain

$$\mathrm{F}'(t^* + h)^{-1} = \left[\mathrm{I} + X_2 h + X_3 h^2 + X_4 h^3 + \cdots\right]\mathrm{F}'(t^*)^{-1} + O(h^p), \tag{4}$$

where

$$\begin{aligned} X_2 &= -2K_2, \\ X_3 &= 4K_2^2 - 3K_3, \\ X_4 &= -8K_2^3 + 6K_2K_3 + 6K_3K_2 - 4K_4, \\ &\vdots \end{aligned} \tag{5}$$

To analyze the convergence properties of scheme (2), we prove the following theorem:

Theorem 1 *Let the function* $\mathrm{F} : \Omega \subset \mathbb{R}^m \to \mathbb{R}^m$ *be sufficiently differentiable in an open neighborhood* Ω *of its zero* t^*. *Suppose that* $\mathrm{F}'(t)$ *is continuous and nonsingular in* t^*. *If an initial approximation* t_0 *is sufficiently close to* t^*, *then order of convergence of method* (2) *is at least 3, provided* $a_1 = \frac{13}{12}$ *and* $a_2 = -\frac{3}{4}$.

Proof Let $\epsilon_k = t_k - t^*$ and $\Gamma = \mathrm{F}'(t^*)^{-1}$ exist. Developing $\mathrm{F}(t_k)$ in a neighborhood of t^*, we have that

$$\mathrm{F}(t_k) = \mathrm{F}'(t^*)[\epsilon_k + A_2\epsilon_k^2 + A_3\epsilon_k^3 + A_4\epsilon_k^4 + A_5\epsilon_k^5 + O(\epsilon_k^6)], \tag{6}$$

where $A_q = \frac{1}{q!}\Gamma\mathrm{F}^{(q)}(t^*)$, $q = 2, 3, \ldots$, $\mathrm{F}^{(q)}(t^*) \in \mathcal{L}(\mathbb{R}^m \times \overset{q-times}{\cdots} \times \mathbb{R}^m, \mathbb{R}^m)$, $\Gamma \in \mathcal{L}(\mathbb{R}^m, \mathbb{R}^m)$ and $A_q(\epsilon_k)^q = A_q(\epsilon_k, \epsilon_k, \overset{q-times}{\cdots}, \epsilon_k) \in \mathbb{R}^m$ with $\epsilon_k \in \mathbb{R}^m$. Also,

$$\mathrm{F}'(t_k) = \mathrm{F}'(t^*)[\mathrm{I} + 2A_2\epsilon_k + 3A_3\epsilon_k^2 + 4A_4\epsilon_k^3 + 5A_5\epsilon_k^4 + O(\epsilon_k^5)], \tag{7}$$

$$\mathrm{F}'(t_k)^{-1} = [\mathrm{I} + B_1\epsilon_k + B_2\epsilon_k^2 + B_3\epsilon_k^3 + B_4\epsilon_k^4 + O(\epsilon_k^5)]\Gamma, \tag{8}$$

where $B_1 = -2A_2$, $B_2 = 4A_2^2 - 3A_3$, $B_3 = -(8A_2^3 - 6A_2A_3 - 6A_3A_2 + 4A_4)$ and $B_4 = (16A_2^4 + 9A_3^2 - 12A_2^2A_3 - 12A_2A_3A_2 - 12A_3A_2^2 + 8A_2A_4 + 8A_4A_2 - 5A_5)$. Applying (6) and (8) in the first step of (2), we have

$$\begin{aligned}\epsilon_{w_k} =& w_k - t^* \\ =& \frac{1}{3}\epsilon_k + \frac{2}{3}A_2\epsilon_k^2 - \frac{4}{3}(A_2^2 - A_3)\epsilon_k^3 + \frac{2}{3}(4A_2^3 - 4A_2A_3 - 3A_3A_2 + 3A_4)\epsilon_k^4 + O(\epsilon_k^5).\end{aligned} \tag{9}$$

Taylor expansions of $\mathrm{F}(w_k)$ and $\mathrm{F}'(w_k)$ about t^* yield

$$\mathrm{F}(w_k) = \mathrm{F}'(t^*)[\epsilon_{w_k} + A_2\epsilon_{w_k}^2 + A_3\epsilon_{w_k}^3 + A_4\epsilon_{w_k}^4 + A_5\epsilon_{w_k}^5 + O(\epsilon_{w_k}^6)], \tag{10}$$

and

$$\mathrm{F}'(w_k) = \mathrm{F}'(t^*)[\mathrm{I} + 2A_2\epsilon_{w_k} + 3A_3\epsilon_{w_k}^2 + 4A_4\epsilon_{w_k}^3 + 5A_5\epsilon_{w_k}^4 + O(\epsilon_{w_k}^5)]. \tag{11}$$

Combining (8) and (11),

$$\begin{aligned}Q_k = \mathrm{F}'(t_k)^{-1}\mathrm{F}'(w_k) =& \mathrm{I} - \frac{4}{3}A_2\epsilon_k + \frac{4}{3}(3A_2^2 - 2A_3)\epsilon_k^2 \\ & - \frac{8}{27}(36A_2^3 - 27A_2A_3 - 18A_3A_2 + 13A_4)\epsilon_k^3 + O(\epsilon_k^4).\end{aligned} \tag{12}$$

Then simple calculations yield

$$\begin{aligned}a_1\mathrm{I} + a_2Q_k = (a_1 + a_2)\,\mathrm{I} - \frac{4a_2}{3}A_2\epsilon_k + \frac{4a_2}{3}(3A_2^2 - 2A_3)\epsilon_k^2 \\ - \frac{8a_2}{27}(36A_2^3 - 27A_2A_3 - 18A_3A_2 + 13A_4)\epsilon_k^3 + O(\epsilon_k^4).\end{aligned} \tag{13}$$

Substituting (6), (8), (9) and (13) in second step of (2), we obtain

$$\begin{aligned}\epsilon_{k+1} =& \Big(\frac{1}{3} - a_1 - a_2\Big)\epsilon_k + \frac{1}{3}(2 + 3a_1 + 7a_2)A_2\epsilon_k^2 \\ & - \frac{2}{3}\big((2 + 3a_1 + 11a_2)A_2^2 - (2 + 3a_1 + 7a_2)A_3\big)\epsilon_k^3 \\ & + O(\epsilon_k^4).\end{aligned} \tag{14}$$

It easy to prove that for the parameters $a_1 = \frac{13}{12}$ and $a_2 = -\frac{3}{4}$, the error equation (14) produces the maximum order of convergence. For these values of parameters above equation reduces to

$$\epsilon_{k+1} = 2A_2^2\epsilon_k^3 + O(\epsilon_k^4). \tag{15}$$

This proves the third order of convergence. □

Thus, the proposed Newton–Jarratt third-order method (2) is finally presented as

$$\begin{aligned} w_k &= t_k - \frac{2}{3}\mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k), \\ x_{k+1} &= M_1^{(3)}(t_k, w_k) = w_k - \frac{1}{12}(13\,\mathrm{I} - 9Q_k)\mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k). \end{aligned} \tag{16}$$

In terms of computational cost this formula uses one function, two derivatives and one matrix inversion per iteration.

3.2 Fifth-order scheme

Based on the third-order scheme (16), we consider the following three-step Newton–Jarratt composition:

$$\begin{aligned} w_k &= t_k - \frac{2}{3}\mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k), \\ y_k &= M_1^{(3)}(t_k, w_k), \\ t_{k+1} &= y_k - \big(d_1\,\mathrm{I} + d_2 Q_k\big)\mathrm{F}'(t_k)^{-1}\mathrm{F}(y_k), \end{aligned} \tag{17}$$

where d_1 and d_2 are some parameters to be determined. This new scheme requires one extra function evaluation in addition to the evaluations of scheme (16). Following theorem proves convergence properties of (17).

Theorem 2 *Let the function* $\mathrm{F} : \Omega \subset \mathbb{R}^m \to \mathbb{R}^m$ *be sufficiently differentiable in an open neighborhood* Ω *of its zero* t^*. *Suppose that* $\mathrm{F}'(t)$ *is continuous and nonsingular in* t^*. *If an initial approximation* t_0 *is sufficiently close to* t^*, *then order of convergence of method* (17) *is at least* 5, *provided that* $d_1 = \frac{5}{2}$ *and* $d_2 = -\frac{3}{2}$.

Proof From (15), error equation of second step of (16) can be written as

$$\epsilon_{y_k} = 2A_2^2\epsilon_k^3 + O(\epsilon_k^4). \tag{18}$$

Let $\epsilon_{y_k} = y_k - t^*$. Then, the Taylor expansion of $\mathrm{F}(y_k)$ about t^* yields

$$\mathrm{F}(y_k) = \mathrm{F}'(t^*)[\epsilon_{y_k} + A_2\epsilon_{y_k}^2 + O(\epsilon_{y_k}^3)]. \tag{19}$$

Using Eqs. (8), (12), (18) and (19) in third step of (17), we obtain

$$\begin{aligned} \epsilon_{k+1} = {} & -2(d_1 + d_2 - 1)A_2^2\epsilon_k^3 + \frac{1}{9}\Big(3(39d_1 + 47d_2 - 27)A_2^2 - 36(d_1 + d_2 - 1)A_2A_3 \\ & - 27(d_1 + d_2 - 1)A_3A_2 - (d_1 + d_2 - 1)A_4\Big)\epsilon_k^4 - \frac{2}{27}\Big(9(84d_1 + 122d_2 - 45)A_2^4 \\ & - 18(48d_1 + 59d_2 - 33)A_3A_2^2 + (123d_1 + 121d_2 - 126)A_2A_4 \end{aligned}$$

$$+ (d_1 + d_2 - 1)(81A_3^2 + 4A_5)\Big)\epsilon_k^5 + O(\epsilon_k^6). \tag{20}$$

It can be easily shown that for parameters $d_1 = \frac{5}{2}$ and $d_2 = -\frac{3}{2}$, the error equation (20) produces the maximum order of convergence. For this set of parameters, Eq. (20) reduces to

$$\epsilon_{k+1} = 2(6A_2^2 - A_3)A_2^2\epsilon_k^5 + O(\epsilon_k^6). \tag{21}$$

Hence the required result follows. □

Thus, the Newton–Jarratt fifth-order method is expressed as

$$\begin{aligned} w_k &= t_k - \frac{2}{3}\mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k), \\ y_k &= M_1^{(3)}(t_k, w_k), \\ t_{k+1} &= M_1^{(5)}(t_k, w_k, y_k) = y_k - \frac{1}{2}\big(5\,\mathrm{I} - 3\,Q_k\big)\mathrm{F}'(t_k)^{-1}\mathrm{F}(y_k). \end{aligned} \tag{22}$$

In terms of computational cost this formula requires two functions, two derivatives and one inverse operator per iteration.

4 Generalized method

The generalized $q+1$-step Newton–Jarratt composite scheme, with the base as three-step scheme (22), can be expressed as follows:

$$\begin{aligned} w_k^{(0)} &= t_k - \frac{2}{3}\mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k), \\ w_k^{(1)} &= w_k - \frac{1}{12}(13\,\mathrm{I} - 9Q_k)\mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k), \\ w_k^{(2)} &= w_k^{(1)} - \varphi(t_k, w_k)\mathrm{F}'(t_k)^{-1}\mathrm{F}(w_k^{(1)}), \\ w_k^{(3)} &= w_k^{(2)} - \varphi(t_k, w_k)\mathrm{F}'(t_k)^{-1}\mathrm{F}(w_k^{(2)}), \\ w_k^{(4)} &= w_k^{(3)} - \varphi(t_k, w_k)\mathrm{F}'(t_k)^{-1}\mathrm{F}(w_k^{(3)}), \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ w_k^{(q-1)} &= w_k^{(q-2)} - \varphi(t_k, w_k)\mathrm{F}'(t_k)^{-1}\mathrm{F}(w_k^{(q-2)}), \\ w_k^{(q)} &= t_{k+1} = w_k^{(q-1)} - \varphi(t_k, w_k)\mathrm{F}'(t_k)^{-1}\mathrm{F}(w_k^{(q-1)}), \end{aligned} \tag{23}$$

where $q \geq 2$, $w_k^{(0)} = w_k$, $Q_k = \mathrm{F}'(t_k)^{-1}\mathrm{F}'(w_k)$ and $\varphi(t_k, w_k) = \frac{1}{2}\big(5\,\mathrm{I} - 3\,Q_k\big)$.

To analyze the convergence property, we prove the following theorem:

Theorem 3 *Let the function* $\mathrm{F} : \Omega \subset \mathbb{R}^m \to \mathbb{R}^m$ *be sufficiently differentiable in an open neighborhood* Ω *of its zero* t^*. *Suppose* $\mathrm{F}'(t)$ *is continuous and nonsingular in* t^*. *If an initial approximation* t_0 *is sufficiently close to* t^*, *the sequence* $\{t_k\}$ *generated by method* (23) *for* $t_0 \in \Omega$ *converges to* t^* *with order* $2q+1$ *for* $q \in \mathbb{N}$.

Proof Let $\epsilon_k = t_k - t^*$, $\epsilon_{w_k^{(q-1)}} = w_k^{(q-1)} - t^*$. Taylor's expansion of $\mathrm{F}(w_k^{(q-1)})$ about t^* yields

$$\mathrm{F}(w_k^{(q-1)}) = \mathrm{F}'(t^*)\big[(w_k^{(q-1)} - t^*) + A_2(w_k^{(q-1)} - t^*)^2 + \cdots\big]. \tag{24}$$

Using (8) and (24), we have that

$$\begin{aligned}\mathrm{F}'(t_k)^{-1}\mathrm{F}(w_k^{(q-1)}) &= [\mathrm{I} + B_1\epsilon_k + B_2\epsilon_k^2 + B_3\epsilon_k^3 + B_4\epsilon_k^4 + O(\epsilon_k^5)]\Gamma \\ &\quad \times \mathrm{F}'(t^*)\Big[(w_k^{(q-1)} - t^*) + A_2(w_k^{(q-1)} - t^*)^2 + \cdots\Big] \\ &= (w_k^{(q-1)} - t^*) - \Big(2A_2\epsilon_k - (4A_2^2 - 3A_3)\epsilon_k^2 \\ &\quad + (8A_2^3 - 6A_2A_3 - 6A_3A_2 + 4A_4)\epsilon_k^3 + \cdots\Big) \\ &\quad \times (w_k^{(q-1)} - t^*) + A_2(w_k^{(q-1)} - t^*)^2 - 2A_2\epsilon_k A_2(w_k^{(q-1)} - t^*)^2 + \cdots . \end{aligned} \tag{25}$$

We write Eq. (13) as

$$\begin{aligned}\varphi(t_k, w_k) &= \mathrm{I} + 2A_2\epsilon_k - 2(3A_2^2 - 2A_3)\epsilon_k^2 \\ &\quad + \frac{4}{9}(36A_2^3 - 27A_2A_3 - 18A_3A_2 + 13A_4)\epsilon_k^3 + O(\epsilon_k^4). \end{aligned} \tag{26}$$

Using Eqs. (25) and (26), we get

$$\begin{aligned}&\varphi(t_k, w_k)\mathrm{F}'(t_k)^{-1}\mathrm{F}(w_k^{(q-1)}) \\ &= (w_k^{(q-1)} - t^*) + \Big((-6A_2^2 + A_3)\epsilon_k^2 + \frac{2}{9}\Big(216A_2^3 - 108A_2A_3 - 99A_3A_2 \\ &\quad + 44A_4\Big)\epsilon_k^3 + O(\epsilon_k^4)\Big)(w_k^{(q-1)} - t^*) + \cdots . \end{aligned} \tag{27}$$

Then last step of (23) yields

$$\begin{aligned}w_k^{(q)} - t^* &= \Big[(6A_2^2 - A_3)\epsilon_k^2 - \frac{2}{9}\Big(216A_2^3 - 108A_2A_3 \\ &\quad -99A_3A_2 + 44A_4\Big)\epsilon_k^3 + \cdots\Big](w_k^{(q-1)} - t^*) + \cdots . \end{aligned}$$

For $q = 3, 4, 5$, we obtain the corresponding error equations as

$$\begin{aligned}w_k^{(3)} - t^* &= \Big[(6A_2^2 - A_3)\epsilon_k^2 - \frac{2}{9}\Big(216A_2^3 - 108A_2A_3 \\ &\quad - 99A_3A_2 + 44A_4\Big)\epsilon_k^3 + \cdots\Big](w_k^{(2)} - t^*) + \cdots \\ &= 2(6A_2^2 - A_3)^2A_2^2\epsilon_k^7 + O(\epsilon_k^8), \\ w_k^{(4)} - t^* &= \Big[(6A_2^2 - A_3)\epsilon_k^2 - \frac{2}{9}\Big(216A_2^3 - 108A_2A_3 \\ &\quad - 99A_3A_2 + 44A_4\Big)\epsilon_k^3 + \cdots\Big](w_k^{(3)} - t^*) + \cdots \\ &= 2(6A_2^2 - A_3)^3A_2^2\epsilon_k^9 + O(\epsilon_k^{10}) \end{aligned}$$

and

$$\begin{aligned} w_k^{(5)} - t^* =& \Bigg[(6A_2^2 - A_3)\epsilon_k^2 - \frac{2}{9}\Big(216A_2^3 - 108A_2A_3 \\ & - 99A_3A_2 + 44A_4 \Big)\epsilon_k^3 + \cdots \Bigg] (w_k^{(4)} - t^*) + \cdots \\ =& 2(6A_2^2 - A_3)^4 A_2^2 \epsilon_k^{11} + O(\epsilon_k^{12}). \end{aligned}$$

Proceeding by induction, we have

$$\epsilon_{k+1} = w_k^{(q)} - t^* = 2(6A_2^2 - A_3)^{q-1} A_2^2 \epsilon_k^{2q+1} + O(\epsilon_k^{2q+2}).$$

Hence, the result follows. □

It is clear that the generalized scheme requires the information of q functions, two derivatives and only one matrix inversion per iteration. The single use of inverse operator throughout the iteration also justifies the name 'method with frozen inverse operator' of the above scheme

5 Computational efficiency

Ranking of numerical methods, based on their computational efficiency, is in many cases a difficult task since the quality of an algorithm depends on many parameters. Considering root-solvers, Brent (1973) said that "*...the method with the higher efficiency is not always the method with the higher order*". Sometimes a great accuracy of the sought results is not a main target whereas in many cases numerical stability of implemented algorithms is the preferable feature.

Computational efficiency of a root-solver can be defined in various manners, but always proportional to order of convergence p and inversely proportional to computational cost C per iteration—the number of function evaluations taking with certain weights. Traub (1964) introduced coefficient of efficiency by the ratio

$$E_T = \frac{p}{C},$$

whereas Ostrowski (1960) dealt with alternative definitions

$$E_{O_1} = p^{1/C} \text{ and } E_{O_2} = \frac{\log p}{C}.$$

An interesting question arises: Which of these definitions describes computational efficiency in the best way in practice when iterative methods are implemented on digital computers? This will be clarified in what follows.

To solve nonlinear equations, assuming that the tested equation has a solution t^* contained in an interval (n-cube or n-ball in general) of unit diameter. Starting with an initial approximation t_0 to t^*, a stopping criterion is given by

$$||t_k - t^*|| \leq \tau = 10^{-d},$$

where k is the iteration index, τ is the required accuracy, and d is the number of significant decimal digits of the approximation t_k. Assume that $||t_0 - t^*|| \approx 10^{-1}$ and let p be the order of convergence of the applied iterative method. Then the (theoretical) number of iterative steps, necessary to reach the accuracy τ, can be calculated approximately from the relation

$10^{-d} = 10^{-p^k}$ as $k \approx \log d / \log p$. Taking into account that the computational efficiency is proportional to the reciprocal value of the total computational cost kC of the completed iterative process consisting of k iterative steps, one gets the estimation of computer efficiency,

$$E = \frac{1}{kC} = \frac{1}{\log d} \frac{\log p}{C}. \tag{28}$$

For the function $F(t) = (f_1(t), f_2(t), \ldots, f_m(t))^{\mathrm{T}}$, where $t = (t_1, t_2, \ldots, t_m)^{\mathrm{T}}$, the computational cost C is computed as

$$\mathrm{C}(\mu_0, \mu_1, m, l) = P_0(m)\mu_0 + P_1(m)\mu_1 + P(m, l), \tag{29}$$

where $P_0(m)$ represents the number of evaluations of scalar functions used in the evaluation of F, $P_1(m)$ is the number of evaluations of scalar functions of F', i.e. $\frac{\partial f_i}{\partial t_j}$, $1 \le i, j \le m$ and $P(m, l)$ represents the number of products or quotients needed per iteration. To express the value of $C(\mu_0, \mu_1, m, l)$ in terms of products, the ratios $\mu_0 > 0$ and $\mu_1 > 0$ between products and evaluations and a ratio $l \ge 1$ between products and quotients are required.

It is clear form the above discussion that estimating the computational efficiency of iterative methods for some fixed accuracy, it is sufficient to compare the values of $\log p/C$. This means that the second Ostrowski's formula $E_{O_2} = \log p/C$ is preferable in the sense of the best fitting a real CPU time. Let us note that this formula was used in many manuscripts and books, see, e.g., Brent (1973) and McNamee (2007).

We shall make use of the Definition (28) for assessing the computational efficiency of presented methods. To do this we must consider all possible factors which contribute to the total cost of computation. For example, to compute F in any iterative method we need to calculate m scalar functions. The number of scalar evaluations is m^2 for any new derivative F'. To compute an inverse linear operator we solve a linear system, where we have $m(m-1)(2m-1)/6$ 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. We must add m^2 products for multiplication of a matrix with a vector or of a matrix by a scalar and m products for multiplication of a vector by a scalar.

To demonstrate the computational efficiency we consider the third, fifth- and seventh-order methods of the family (23) and compare the efficiency with existing third, fifth- and seventh-order methods. For example, third-order method $M_1^{(3)}$ is compared with third-order methods by Homeier (2004), Cordero and Torregrosa (2007), Noor and Waseem (2009) and Xiao and Yin (2017); fifth-order method $M_1^{(5)}$ with fifth-order methods by Cordero et al. (2010, 2012), Sharma and Gupta (2014) and Xiao and Yin (2018); and seventh-order $M_1^{(7)}$ with seventh-order method by Xiao and Yin (2015). In addition, the new methods are also compared with each other. The considered existing methods are expressed as follows.

Third-order Homeier method (Homeier 2004):

$$\begin{aligned} y_k &= t_k - \frac{1}{2}\mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k), \\ t_{k+1} &= M_2^{(3)}(t_k, y_k) = t_k - \mathrm{F}'(y_k)^{-1}\mathrm{F}(t_k). \end{aligned}$$

Third-order methods by (Cordero and Torregrosa 2007):

$$\begin{aligned} y_k &= t_k - \mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k), \\ t_{k+1} &= M_3^{(3)}(t_k, y_k) = t_k - 6\left[\mathrm{F}'(t_k) + 4\mathrm{F}'\left(\frac{t_k + y_k}{2}\right) + \mathrm{F}'(y_k)\right]^{-1}\mathrm{F}(t_k) \end{aligned}$$

and

$$y_k = t_k - \mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k),$$
$$t_{k+1} = M_4^{(3)}(t_k, y_k) = t_k - 3\left[2\mathrm{F}'\left(\frac{3t_k + y_k}{4}\right) - \mathrm{F}'\left(\frac{t_k + y_k}{2}\right) + 2\mathrm{F}'\left(\frac{t_k + 3y_k}{4}\right)\right]^{-1}\mathrm{F}(t_k).$$

Third-order Noor–Waseem method (Noor and Waseem 2009):

$$y_k = t_k - \mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k),$$
$$t_{k+1} = M_5^{(3)}(t_k, y_k) = t_k - 4\left[3\mathrm{F}'\left(\frac{2t_k + y_k}{3}\right) + \mathrm{F}'(y_k)\right]^{-1}\mathrm{F}(t_k).$$

Third-order Xiao–Yin method (Xiao and Yin 2017):

$$y_k = t_k - \mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k),$$
$$t_{k+1} = M_6^{(3)}(t_k, y_k) = t_k - \frac{2}{3}\left(\left(3\mathrm{F}'(y_k) - \mathrm{F}'(t_k)\right)^{-1} + \mathrm{F}'(t_k)^{-1}\right)\mathrm{F}(t_k).$$

Fifth-order methods by (Cordero et al. 2010, 2012):

$$y_k = t_k - \frac{2}{3}\mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k),$$
$$z_k = t_k - \frac{1}{2}\left(3\mathrm{F}'(y_k) - \mathrm{F}'(t_k)\right)^{-1}\left(3\mathrm{F}'(y_k) + \mathrm{F}'(t_k)\right)\mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k),$$
$$t_{k+1} = M_2^{(5)}(t_k, y_k, z_k) = z_k - \left(\alpha\mathrm{F}'(y_k) + \beta\mathrm{F}'(t_k)\right)^{-1}\mathrm{F}(z_k)$$

and

$$y_k = t_k - \mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k),$$
$$z_k = t_k - 2\left(\mathrm{F}'(y_k) + \mathrm{F}'(t_k)\right)^{-1}\mathrm{F}(t_k),$$
$$t_{k+1} = M_3^{(5)}(t_k, y_k, z_k) = z_k - \mathrm{F}'(y_k)^{-1}\mathrm{F}(z_k),$$

where $\alpha = \beta = 0.5$.

Fifth-order Sharma–Gupta method (Sharma and Gupta 2014):

$$y_k = t_k - \frac{1}{2}\mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k),$$
$$z_k = t_k - \mathrm{F}'(y_k)^{-1}\mathrm{F}(t_k),$$
$$t_{k+1} = M_4^{(5)}(t_k, y_k, z_k) = z_k - \left(2\mathrm{F}'(y_k)^{-1} - \mathrm{F}'(t_k)^{-1}\right)\mathrm{F}(z_k).$$

Fifth-order Xiao–Yin method (Xiao and Yin 2018):

$$y_k = t_k - \frac{2}{3}\mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k),$$
$$z_k = t_k - \frac{1}{4}\left(3\mathrm{F}'(y_k)^{-1} + \mathrm{F}'(t_k)^{-1}\right)\mathrm{F}(t_k),$$
$$t_{k+1} = M_5^{(5)}(t_k, y_k, z_k) = z_k - \frac{1}{2}\left(3\,\mathrm{F}'(y_k)^{-1} - \mathrm{F}'(t_k)^{-1}\right)\mathrm{F}(z_k).$$

Seventh-order Xiao–Yin method (Xiao and Yin 2015):

$$\begin{aligned} y_k &= t_k - \frac{1}{2}\mathrm{F}'(t_k)^{-1}\mathrm{F}(t_k),\\ z_k &= t_k - \mathrm{F}'(y_k)^{-1}\mathrm{F}(t_k),\\ x_k &= z_k - \left(2\mathrm{F}'(y_k)^{-1} - \mathrm{F}'(t_k)^{-1}\right)\mathrm{F}(z_k),\\ t_{k+1} &= M_2^{(7)}(t_k, y_k, z_k, x_k) = x_k - \left(2\mathrm{F}'(y_k)^{-1} - \mathrm{F}'(t_k)^{-1}\right)\mathrm{F}(x_k). \end{aligned}$$

Denoting the efficiency indices of the methods $M_i^{(p)}$ ($p = 3, 5, 7$ and $i = 1, 2, 3, 4, 5$) by $\mathcal{E}_i^{(p)}$ and computational costs by $C_i^{(p)}$. Then taking into account above considerations, we obtain

$$\begin{aligned}
C_1^{(3)} &= m\mu_0 + 2m^2\mu_1 + \frac{m}{6}(2m^2 + 15m + 7 + 3l(3+m)) \text{ and } \mathcal{E}_1^{(3)} = \frac{1}{D}\frac{\log 3}{C_1^{(3)}},\\
C_2^{(3)} &= m\mu_0 + 2m^2\mu_1 + \frac{m}{3}(2m^2 + 3m - 2 + 3l(1+m)) \text{ and } \mathcal{E}_2^{(3)} = \frac{1}{D}\frac{\log 3}{C_2^{(3)}},\\
C_3^{(3)} &= m\mu_0 + 3m^2\mu_1 + \frac{m}{3}(2m^2 + 6m + 1 + 3l(1+m)) \text{ and } \mathcal{E}_3^{(3)} = \frac{1}{D}\frac{\log 3}{C_3^{(3)}},\\
C_4^{(3)} &= m\mu_0 + 4m^2\mu_1 + \frac{m}{3}(2m^2 + 9m + 13 + 3l(1+m)) \text{ and } \mathcal{E}_4^{(3)} = \frac{1}{D}\frac{\log 3}{C_4^{(3)}},\\
C_5^{(3)} &= m\mu_0 + 3m^2\mu_1 + \frac{m}{3}(2m^2 + 6m + 4 + 3l(1+m)) \text{ and } \mathcal{E}_5^{(3)} = \frac{1}{D}\frac{\log 3}{C_5^{(3)}},\\
C_6^{(3)} &= m\mu_0 + 2m^2\mu_1 + \frac{m}{3}(2m^2 + 6m - 2 + 3l(1+m)) \text{ and } \mathcal{E}_6^{(3)} = \frac{1}{D}\frac{\log 3}{C_6^{(3)}},\\
C_1^{(5)} &= 2m\mu_0 + 2m^2\mu_1 + \frac{m}{6}(2m^2 + 33m + 7 + 3l(7+m)) \text{ and } \mathcal{E}_1^{(5)} = \frac{1}{D}\frac{\log 5}{C_1^{(5)}},\\
C_2^{(5)} &= 2m\mu_0 + 2m^2\mu_1 + \frac{m}{2}(2m^2 + 7m + 1 + 3l(1+m)) \text{ and } \mathcal{E}_2^{(5)} = \frac{1}{D}\frac{\log 5}{C_2^{(5)}},\\
C_3^{(5)} &= 2m\mu_0 + 2m^2\mu_1 + \frac{m}{2}(2m^2 + 3m - 3 + 3l(1+m)) \text{ and } \mathcal{E}_3^{(5)} = \frac{1}{D}\frac{\log 5}{C_3^{(5)}},\\
C_4^{(5)} &= 2m\mu_0 + 2m^2\mu_1 + \frac{m}{3}(2m^2 + 9m - 5 + 3l(3+m)) \text{ and } \mathcal{E}_4^{(5)} = \frac{1}{D}\frac{\log 5}{C_4^{(5)}},\\
C_5^{(5)} &= 2m\mu_0 + 2m^2\mu_1 + \frac{m}{3}(2m^2 + 9m + 1 + 3l(3+m)) \text{ and } \mathcal{E}_5^{(5)} = \frac{1}{D}\frac{\log 5}{C_5^{(5)}},\\
C_1^{(7)} &= 3m\mu_0 + 2m^2\mu_1 + \frac{m}{6}(2m^2 + 51m + 7 + 3l(11+m)) \text{ and } \mathcal{E}_1^{(7)} = \frac{1}{D}\frac{\log 5}{C_1^{(7)}},\\
C_2^{(7)} &= 3m\mu_0 + 2m^2\mu_1 + \frac{m}{3}(2m^2 + 15m - 8 + 3l(5+m)) \text{ and } \mathcal{E}_2^{(7)} = \frac{1}{D}\frac{\log 5}{C_2^{(7)}},
\end{aligned}$$

wherein $D = \log d$.

5.1 Comparison between efficiencies

To compare the efficiencies of iterative methods, say $M_i^{(p)}$ against $M_j^{(q)}$, we consider the ratio

$$R_{i,j}^{p,q} = \frac{\mathcal{E}_i^{(p)}}{\mathcal{E}_j^{(q)}} = \frac{C_j^{(q)} \log p}{C_i^{(p)} \log q}.$$

It is clear that if $R_{i,j}^{p,q} > 1$, the iterative method $M_i^{(p)}$ is more efficient than $M_j^{(q)}$. Mathematically this fact will be denoted by $M_i^{(p)} \succcurlyeq M_j^{(q)}$. Taking into account that the border between two computational efficiencies is given by $R_{i,j}^{p,q} = 1$, this boundary will be given by the equation μ_0 written as a function of μ_1, m and l; $(\mu_0, \mu_1) \in (0, +\infty) \times (0, +\infty)$, m is a positive integer ≥ 2 and $l \geq 1$. In the sequel, we consider the comparison of computational efficiencies of the methods as expressed above.

$M_1^{(3)}$ *versus* $M_2^{(3)}$ *case*:
In this case, the ratio

$$R_{1,2}^{3,3} = \frac{m\mu_0 + 2m^2\mu_1 + \frac{m}{3}(2m^2 + 3m - 2 + 3l(1+m))}{m\mu_0 + 2m^2\mu_1 + \frac{m}{6}(2m^2 + 15m + 7 + 3l(3+m))} > 1,$$

for $m \geq 5$, which implies that $\mathcal{E}_1^3 > \mathcal{E}_2^3$ and hence $M_1^3 \succcurlyeq M_2^3$ for all $m \geq 5$ and $l \geq 1$.

$M_1^{(3)}$ *versus* $M_3^{(3)}$ *case*:
In this case, the ratio

$$R_{1,3}^{3,3} = \frac{m\mu_0 + 3m^2\mu_1 + \frac{m}{3}(2m^2 + 6m + 1 + 3l(1+m))}{m\mu_0 + 2m^2\mu_1 + \frac{m}{6}(2m^2 + 15m + 7 + 3l(3+m))} > 1,$$

for $m \geq 2$, which implies that $\mathcal{E}_1^3 > \mathcal{E}_3^3$ and hence $M_1^3 \succcurlyeq M_3^3$ for all $m \geq 2$ and $l \geq 1$.

$M_1^{(3)}$ *versus* $M_4^{(3)}$ *case*:
For this case, the ratio

$$R_{1,4}^{3,3} = \frac{m\mu_0 + 4m^2\mu_1 + \frac{m}{3}(2m^2 + 9m + 13 + 3l(1+m))}{m\mu_0 + 2m^2\mu_1 + \frac{m}{6}(2m^2 + 15m + 7 + 3l(3+m))} > 1.$$

It is easy to prove that $R_{1,4}^{3,3} > 1$ for $m \geq 1$. Thus, we conclude that $\mathcal{E}_1^3 > \mathcal{E}_4^3$ and consequently $M_1^3 \succcurlyeq M_4^3$ for all $m \geq 2$ and $l \geq 1$.

$M_1^{(3)}$ *versus* $M_5^{(3)}$ *case*:
For this case, the ratio

$$R_{1,5}^{3,3} = \frac{m\mu_0 + 3m^2\mu_1 + \frac{m}{3}(2m^2 + 6m + 4 + 3l(1+m))}{m\mu_0 + 2m^2\mu_1 + \frac{m}{6}(2m^2 + 15m + 7 + 3l(3+m))} > 1.$$

We have that $R_{1,5}^{3,3} > 1$ for $m \geq 1$, which shows $\mathcal{E}_1^3 > \mathcal{E}_5^3$ and consequently $M_1^3 \succcurlyeq M_5^3$ for all $m \geq 2$ and $l \geq 1$.

$M_1^{(3)}$ *versus* $M_6^{(3)}$ *case*:
In this case, the ratio

$$R_{1,6}^{3,3} = \frac{m\mu_0 + 2m^2\mu_1 + \frac{m}{3}(2m^2 + 6m - 2 + 3l(1+m))}{m\mu_0 + 2m^2\mu_1 + \frac{m}{6}(2m^2 + 15m + 7 + 3l(3+m))} > 1,$$

for $m \geq 3$, which implies that $\mathcal{E}_1^3 > \mathcal{E}_6^3$ for all $m \geq 3$ and $l \geq 1$, that is $M_1^3 \succcurlyeq M_6^3$.
$M_1^{(5)}$ *versus* $M_2^{(5)}$ *case*:
In this case, the ratio

$$R_{1,2}^{5,5} = \frac{2m\mu_0 + 2m^2\mu_1 + \frac{m}{2}(2m^2 + 7m + 1 + 3l(1+m))}{2m\mu_0 + 2m^2\mu_1 + \frac{m}{6}(2m^2 + 33m + 7 + 3l(7+m))} > 1,$$

for $m \geq 3$, which implies that $\mathcal{E}_1^5 > \mathcal{E}_2^5$ for all $m \geq 3$ and $l \geq 1$. Therefore $M_1^5 \succcurlyeq M_2^5$.
$M_1^{(5)}$ *versus* $M_3^{(5)}$ *case*:
In this case, the ratio

$$R_{1,3}^{5,5} = \frac{2m\mu_0 + 2m^2\mu_1 + \frac{m}{2}(2m^2 + 3m - 3 + 3l(1+m))}{2m\mu_0 + 2m^2\mu_1 + \frac{m}{6}(2m^2 + 33m + 7 + 3l(7+m))} > 1,$$

for $m \geq 6$, which implies that $\mathcal{E}_1^5 > \mathcal{E}_3^5$ and $M_1^5 \succcurlyeq M_3^5$ for all $m \geq 6$ and $l \geq 1$.
$M_1^{(5)}$ *versus* $M_4^{(5)}$ *case*:
In this case, the ratio

$$R_{1,4}^{5,5} = \frac{2m\mu_0 + 2m^2\mu_1 + \frac{m}{3}(2m^2 + 9m - 5 + 3l(3+m))}{2m\mu_0 + 2m^2\mu_1 + \frac{m}{6}(2m^2 + 33m + 7 + 3l(7+m))} > 1,$$

for $m \geq 8$, which implies that $\mathcal{E}_1^5 > \mathcal{E}_4^5$ and $M_1^5 \succcurlyeq M_4^5$ for all $m \geq 8$ and $l \geq 1$.
$M_1^{(5)}$ *versus* $M_5^{(5)}$ *case*:
In this case, the ratio

$$R_{1,5}^{5,5} = \frac{2m\mu_0 + 2m^2\mu_1 + \frac{m}{3}(2m^2 + 9m + 1 + 3l(3+m))}{2m\mu_0 + 2m^2\mu_1 + \frac{m}{6}(2m^2 + 33m + 7 + 3l(7+m))} > 1,$$

for $m \geq 7$, which implies that $\mathcal{E}_1^5 > \mathcal{E}_5^5$ and $M_1^5 \succcurlyeq M_5^5$ for all $m \geq 7$ and $l \geq 1$.
$M_1^{(7)}$ *versus* $M_2^{(7)}$ *case*:
In this case, the ratio

$$R_{1,2}^{7,7} = \frac{3m\mu_0 + 2m^2\mu_1 + \frac{m}{3}(2m^2 + 15m - 8 + 3l(5+m))}{3m\mu_0 + 2m^2\mu_1 + \frac{m}{6}(2m^2 + 51m + 7 + 3l(11+m))} > 1,$$

for $m \geq 11$, which implies that $\mathcal{E}_1^7 > \mathcal{E}_2^7$ and $M_1^7 \succcurlyeq M_2^7$ for all $m \geq 11$ and $l \geq 1$.
$M_1^{(3)}$ *versus* $M_1^{(5)}$ *case*:
For this case, it is judicious to consider the boundary $R_{1,1}^{3,5} = 1$, which is given by

$$\mu_0 = \frac{-(14 + 18m + m^2)r + (8 + 9m + m^2)s + 6m(s-r)\mu_1}{6r - 3s},$$

where $r = \log 3$ and $s = \log 5$. The comparison of efficiencies $\mathcal{E}_1^3$ and $\mathcal{E}_1^5$ is shown in the (μ_1, μ_0)-plane by drawing some particular boundaries corresponding to $m = 2, 5, 10$ and 20 taking $l = 1$, where $\mathcal{E}_1^3 > \mathcal{E}_1^5$ (that is $M_1^3 \succcurlyeq M_1^5$) on the left (above) and $\mathcal{E}_1^5 > \mathcal{E}_1^3$ (that is, $M_1^5 \succcurlyeq M_1^3$) on the right (below) of each line (see Fig. 1).
$M_1^{(3)}$ *versus* $M_1^{(7)}$ *case*:
Like the previous case we consider the boundary $R_{1,1}^{3,7} = 1$, which yields

$$\mu_0 = \frac{-20r - 27mr - m^2r + 8t + 9mt + m^2t - 6mr\mu_1 + 6mt\mu_1}{3(3r - t)},$$

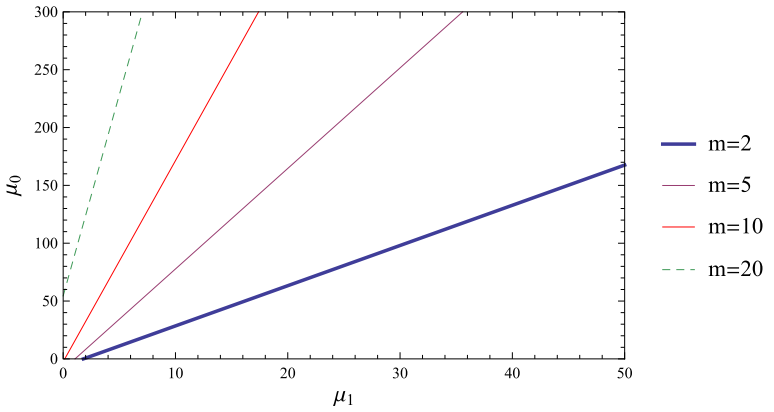

Fig. 1 Boundary lines in (μ_1, μ_0) - plane for M_1^3 versus M_1^5

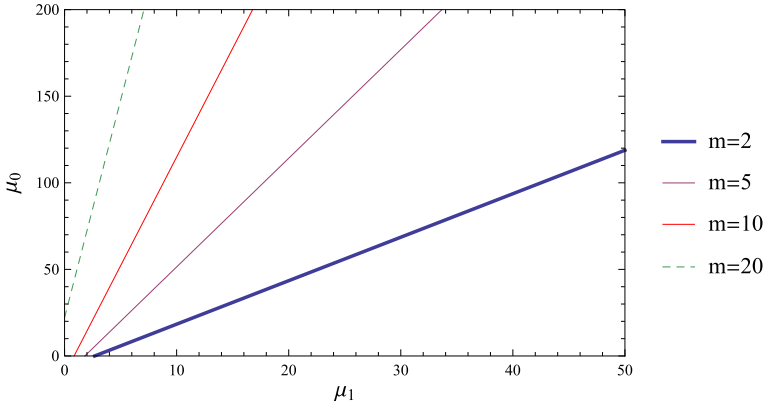

Fig. 2 Boundary lines in (μ_1, μ_0) - plane for M_1^3 versus M_1^7

where $r = \log 3$ and $t = \log 7$. To compare the efficiencies $\mathcal{E}_1^3$ and $\mathcal{E}_1^7$ in the (μ_1, μ_0)-plane, we draw some particular boundaries corresponding to $m = 2, 5, 10$ and 20 taking $l = 1$. These boundaries are shown in Fig. 2, wherein $\mathcal{E}_1^3 > \mathcal{E}_1^7$ (that is $M_1^3 \succ M_1^7$) on the left (above) and $\mathcal{E}_1^7 > \mathcal{E}_1^3$ (that is $M_1^7 \succ M_1^3$) on the right (below) of each line.

$M_1^{(5)}$ versus $M_1^{(7)}$ case:

The boundary $R_{1,1}^{5,7} = 1$ is given by

$$\mu_0 = \frac{-20s - 27ms - m^2 s + 14t + 18mt + m^2 t - 6ms\mu_1 + 6mt\mu_1}{3(3s - 2t)},$$

where $s = \log 5$ and $t = \log 7$. As before we draw boundary lines in (μ_1, μ_0)-plane corresponding to cases $m = 2, 5, 10$ and 20 taking $l = 1$. Boundaries are shown in Fig. 3, wherein $\mathcal{E}_1^5 > \mathcal{E}_1^7$ (that is $M_1^5 \succ M_1^7$) on the left (above) and $\mathcal{E}_1^7 > \mathcal{E}_1^5$ (that is $M_1^7 \succ M_1^5$) on the right (below) of each line.

We summarize the preceding results in following theorem:

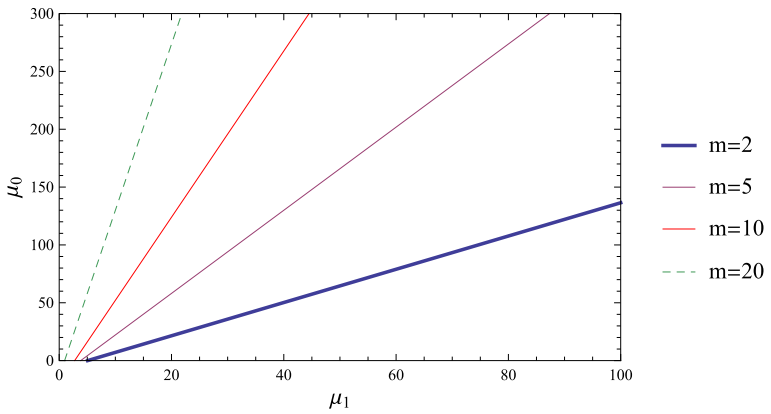

Fig. 3 Boundary lines in (μ_1, μ_0) - plane for M_1^5 versus M_1^7

Theorem 4 (a) *For all* $\mu_0, \mu_1 > 0$ *and* $l \geq 1$ *we have that:*

$$\text{(i)}\mathcal{E}_1^3 > \mathcal{E}_2^3 \Longleftrightarrow M_1^3 \succ M_2^3 \; \forall \; m \geq 5.$$

$$\text{(ii)} \; \{\mathcal{E}_1^3 > \mathcal{E}_3^3, \mathcal{E}_1^3 > \mathcal{E}_4^3, \mathcal{E}_1^3 > \mathcal{E}_5^3\} \Longleftrightarrow \{M_1^3 \succ M_3^3, M_1^3 \succ M_4^3, M_1^3 \succ M_5^3\} \; \forall \; m \geq 2.$$

$$\text{(iii)}\mathcal{E}_1^3 > \mathcal{E}_6^3 \Longleftrightarrow M_1^3 \succ M_6^3 \; \forall \; m \geq 3.$$

$$\text{(iv)}\mathcal{E}_1^5 > \mathcal{E}_2^5 \Longleftrightarrow M_1^5 \succ M_2^5 \; \forall \; m \geq 3 \; \textit{and} \; \mathcal{E}_1^5 > \mathcal{E}_3^5 \Longleftrightarrow M_1^5 \succ M_3^5 \; \forall \; m \geq 6.$$

$$\text{(v)}\mathcal{E}_1^5 > \mathcal{E}_4^5 \Longleftrightarrow M_1^5 \succ M_4^5 \; \forall \; m \geq 8 \; \textit{and} \; \mathcal{E}_1^5 > \mathcal{E}_5^5 \Longleftrightarrow M_1^5 \succ M_5^5 \; \forall \; m \geq 7.$$

$$\text{(vi)}\mathcal{E}_1^7 > \mathcal{E}_2^7 \Longleftrightarrow M_1^7 \succ M_2^7 \; \forall \; m \geq 11.$$

Otherwise, the comparison depends on μ_0, μ_1 *and* l.
(b) *For all* $m \geq 2$ *we have:*

$$\text{(i)}\mathcal{E}_1^3 > \mathcal{E}_1^5 \Longleftrightarrow M_1^3 \succ M_1^5 \; \forall \; \mu_0 > \beta_1 \; \textit{and} \; \mathcal{E}_1^5 > \mathcal{E}_1^3 \Longleftrightarrow M_1^5 \succ M_1^3 \; \forall \; \mu_0 < \beta_1,$$

$$\text{(ii)}\mathcal{E}_1^3 > \mathcal{E}_1^7 \Longleftrightarrow M_1^3 \succ M_1^7 \; \forall \; \mu_0 > \beta_2 \; \textit{and} \; \mathcal{E}_1^7 > \mathcal{E}_1^3 \Longleftrightarrow M_1^7 \succ M_1^3 \; \forall \; \mu_0 < \beta_2,$$

$$\text{(iii)}\mathcal{E}_1^5 > \mathcal{E}_1^7 \Longleftrightarrow M_1^5 \succ M_1^7 \; \forall \; \mu_0 > \beta_3 \; \textit{and} \; \mathcal{E}_1^7 > \mathcal{E}_1^5 \Longleftrightarrow M_1^7 \succ M_1^5 \; \forall \; \mu_0 < \beta_3,$$

where $\beta_1 = \frac{-(14+18m+m^2)r+(8+9m+m^2)s+6m(s-r)\mu_1}{3(2r-s)}$,
$\beta_2 = \frac{-(20+27m+m^2)r+(8+9m+m^2)t+6m(t-r)\mu_1}{3(3r-t)}$ *and*
$\beta_3 = \frac{-(20+27m+m^2)s+(14+18m+m^2)t+6m(t-s)\mu_1}{3(3s-2t)}$.

6 Numerical results

To illustrate the convergence behavior and computational efficiency of the proposed methods $M_1^{(3)}$, $M_1^{(5)}$ and $M_1^{(7)}$, we consider some numerical examples and compare the performance with $M_2^{(3)}$, $M_3^{(3)}$, $M_4^{(3)}$, $M_5^{(3)}$, $M_6^{(3)}$, $M_2^{(5)}$, $M_3^{(5)}$, $M_4^{(5)}$, $M_5^{(5)}$ and $M_2^{(7)}$. All computations are performed in the programming package Mathematica (Wolfram 2003) in a PC with Intel(R) Pentium(R) CPU B960 @ 2.20 GHz, 2.20 GHz (32-bit Operating System) Microsoft Windows 7 Professional and 4 GB RAM. For every method, we analyze the number of iterations (k) needed to converge to the solution such that $\|t_{k+1} - t_k\| + \|\mathrm{F}(t_k)\| < 10^{-100}$.

Table 1 CPU time and estimation of computational cost of the elementary functions, where $x = \sqrt{3} - 1$ and $y = \sqrt{5}$

Functions	$x * y$	x/y	$\sqrt{x}$	e^x	$\ln(x)$	$\sin(x)$	$\cos(x)$	$\cos^{-1}(x)$	$\tan^{-1}(x)$
CPU-time	0.0733	0.1793	0.0718	3.2448	3.0514	3.4039	3.7284	6.4943	6.5193
Cost	1	2.4461	0.9795	44.2674	41.6289	46.4379	50.8649	88.5989	88.9400

To verify the theoretical order of convergence (p), we calculate the computational order of convergence COC by using the formula given in Definition 2.3. In numerical results, we also include CPU time (e-time) utilized in the execution of program which is computed by the Mathematica command TimeUsed[ ]. To calculate $\mathcal{E}_i^p$ for all methods in numerical examples we take $D = 10^{-5}$ and $A(\pm m)$ denotes $A \times 10^{\pm m}$.

The results of theorem 4 are also verified through numerical experiments. To do this, we need an estimation of the factors μ_0 and μ_1. To claim this estimation, we express the cost of the evaluation of the elementary functions in terms of products, which depends on the computer, the software and the arithmetics used (Fousse et al. 2007). In Table 1, an estimation of the cost of the elementary functions in product unit is shown, where the running time of the one product is measured in milliseconds (ms). It is evident from Table 1 that the computational cost of the quotient with respect to product is, $l \approx 2.4$.

For numerical tests we consider the following examples:

Example 1 Consider a system of 2 equations (Sharma and Arora 2016a):

$$\begin{aligned} e^{-t_1^2} + 8\,t_1\,\sin t_2 &= 0, \\ t_1 + t_2 - 1 &= 0. \end{aligned}$$

With the initial approximations $t_0 = \{\frac{4}{10}, -\frac{4}{10}\}^{\mathrm{T}}$, we obtain the solution $t^* = \{1.0407\ldots, -0.0407\ldots\}^{\mathrm{T}}$. The concrete values of parameters (m, μ_0, μ_1), obtained with the help of estimates of elementary functions displayed in Table 1, are $(2, 46.8527, 36.6426)$. These values are used to calculate computational costs and efficiency indices of the methods shown in previous section.

Example 2 Consider a system of 3 equations (Sharma and Arora 2016c):

$$\begin{aligned} 10\,t_1 + \sin(t_1 + t_2) - 1 &= 0, \\ 8\,t_2 - \cos^2(t_3 - t_2) - 1 &= 0 \\ 12\,t_3 + \sin(t_3) - 1 &= 0. \end{aligned}$$

The initial approximation $t_0 = \{0, 1, 0\}^{\mathrm{T}}$ is chosen to find the solution

$$t^* = \{0.0690\ldots, 0.2464\ldots, 0.0769\ldots\}^{\mathrm{T}}.$$

Corresponding calculated values of the parameters (m, μ_0, μ_1) are $(3, 49.2469, 22.3370)$.

Example 3 Let us consider the Van der Pol equation (see Burden and Faires 2001), which is defined as follows:

$$t'' - \mu(t^2 - 1)t' + t = 0,\ \mu > 0, \tag{30}$$

which governs the flow of current in a vacuum tube, with the boundary conditions $t(0) = 0$, $t(2) = 1$. Further, we consider the partition of the given interval [0, 2], which is given by

$$y_0 = 0 < y_1 < y_2 < \cdots < y_{n-1} < y_n, \text{ where } y_i = y_0 + ih, \quad h = \frac{2}{n}.$$

Moreover, we assume that

$$t_0 = t(y_0) = 0, t_1 = t(y_1), \ldots, t_{n-1} = t(y_{n-1}), t_n = t(y_n) = 1.$$

If we discretize the problem (30) using the second-order divided difference for the first and second derivatives, which are given by

$$t'_k = \frac{t_{k+1} - t_{k-1}}{2h}, \quad t''_k = \frac{t_{k-1} - 2t_k + t_{k+1}}{h^2}, \quad (k = 1, 2, 3, \ldots, n-1),$$

then we obtain a system of $(n-1)$ nonlinear equations

$$2h^2 t_k - h\mu(t_k^2 - 1)(t_{k+1} - t_{k-1}) + 2(t_{k-1} + t_{k+1} - 2t_k) = 0, \quad (k = 1, 2, 3, \ldots, n-1).$$

Let us consider $\mu = \frac{1}{2}$ and initial approximation $t_0 = (\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})^{\mathrm{T}}$. In particular, we solve this problem for $n = 6$ so we obtain a system of 5 nonlinear equations. The solution of this problem is

$$t^* = \{0.4393\ldots, 0.7775\ldots, 1.0105\ldots, 1.1319\ldots, 1.1302\ldots\}^{\mathrm{T}}$$

and parametric values are $(m, \mu_0, \mu_1) = (5, 6.9784, 1.3157)$.

Example 4 Next, consider a system of 10 equations (Xiao and Yin 2016):

$$\tan^{-1}(t_i) + 1 - 2\sum_{j=1, j\neq i}^{10} t_j^2, \quad 1 \leq i \leq 10.$$

The solution $t^* = \{0.2644\ldots, 0.2644\ldots, \overset{10}{\cdots}, 0.2644\ldots\}^{\mathrm{T}}$ is obtained by assuming the initial approximations $t_0 = \{\frac{7}{10}, \frac{7}{10} \overset{10}{\cdots}, \frac{7}{10}\}^{\mathrm{T}}$. Computed values of the parameters (m, μ_0, μ_1) are (10, 90.9400, 0.4446).

Example 5 The boundary value problem (see Ortega and Rheinboldt 1970):

$$t'' + a^2(t')^2 + 1 = 0, \quad t(0) = 0, \quad t(1) = 0$$

is studied. Consider the following partitioning of the interval [0,1]:

$$y_0 = 0 < y_1 < y_2 < \cdots < y_{n-1} < y_n = 1, \quad y_{j+1} = y_j + h, \quad h = 1/n.$$

Let us define $t_0 = t(y_0) = 0$, $t_1 = t(y_1), \ldots, t_{n-1} = t_(y_{n-1})$, $t_n = t(y_n) = 1$. If we discretize the problem by using the numerical formulae for first and second derivatives

$$t'_k = \frac{t_{k+1} - t_{k-1}}{2h}, \quad t''_k = \frac{t_{k-1} - 2t_k + t_{k+1}}{h^2}, \quad (k = 1, 2, 3, \ldots, n-1),$$

we obtain a system of $n-1$ nonlinear equations in $n-1$ variables:

$$t_{k-1} - 2t_k + t_{k+1} + \frac{a^2}{4}(t_{k+1} - t_{k-1})^2 + h^2 = 0, \quad (k = 1, 2, 3, \ldots, n-1).$$

In particular, we solve this problem for $n = 51$ so that $k = 50$ by selecting $t_0 = (2, 2, \overset{50}{\cdots}, 2)^{\mathrm{T}}$ as the initial value and $a = 2$. Solution of this problem is

$$\begin{aligned} t^* = \Big\{ & 0.9584\ldots, 0.9719\ldots, 0.9843\ldots, 0.9958\ldots, 1.0063\ldots, \\ & 1.0161\ldots, 1.0252\ldots, 1.0335\ldots, \\ & 1.0412\ldots, 1.0483\ldots, 1.0549\ldots, 1.0609\ldots, 1.0664\ldots, 1.0714\ldots, \\ & 1.0759\ldots, 1.0799\ldots, 1.0835\ldots, \\ & 1.0867\ldots, 1.0895\ldots, 1.0918\ldots, 1.0938\ldots, 1.0953\ldots, 1.0965\ldots, \\ & 1.0972\ldots, 1.0976\ldots, 1.0976\ldots, \\ & 1.0972\ldots, 1.0965\ldots, 1.0953\ldots, 1.0938\ldots, 1.0918\ldots, 1.0895\ldots, \\ & 1.0867\ldots, 1.0835\ldots, 1.0799\ldots, \\ & 1.0759\ldots, 1.0714\ldots, 1.0664\ldots, 1.0609\ldots, 1.0549\ldots, 1.0483\ldots, \\ & 1.0412\ldots, 1.0335\ldots, 1.0252\ldots, \\ & 1.0161\ldots, 1.0063\ldots, 0.9958\ldots, 0.9843\ldots, 0.9719\ldots, 0.9584\ldots \Big\}^{\mathrm{T}} \end{aligned}$$

and $(m, \mu_0, \mu_1) = (50, 3, 0.0392)$.

Example 6 Consider the following Burger's equation (see Sauer 2012):

$$\frac{\partial^2 f}{\partial u^2} + f\frac{\partial f}{\partial u} - \frac{\partial f}{\partial v} + g(u, v) = 0, \quad (u, v) \in [0, 1]^2,$$

where $g(u, v) = -10e^{-2v}[e^v(2 - u + u^2) + 10u(1 - 3u + 2u^2)]$ and function $f = f(u, v)$ satisfies the boundary conditions

$$f(0, v) = f(1, v) = 0,\ f(u, 0) = 10u(u - 1) \text{ and } f(u, 1) = 10u(u - 1)/e.$$

Assuming the following partitioning of the domain $[0, 1]^2$:

$$\begin{aligned} & 0 = u_0 < u_1 < u_2 < \cdots < u_{n-1} < u_n = 1,\ u_{k+1} = u_k + h, \\ & 0 = v_0 < v_1 < v_2 < \cdots < v_{n-1} < v_n = 1,\ v_{l+1} = v_l + h,\ h = 1/n. \end{aligned}$$

Let us define $f_{k,l} = f(u_k, v_l)$ and $g_{k,l} = g(u_k, v_l)$ for $k, l = 0, 1, 2, \ldots, n$. Then the boundary conditions will be $f_{0,l} = f(u_0, v_l) = 0$, $f_{n,l} = f(u_n, v_l) = 0$, $f_{k,0} = f(u_k, v_0) = 10u_k(u_k - 1)$ and $f_{k,n} = f(u_k, v_n) = 10u_k(u_k - 1)/e$. If we discretize Burger's equation by using the numerical formulae for the partial derivatives, we obtain the following system of $(n - 1)^2$ nonlinear equations in $(n - 1)^2$ variables:

$$f_{i-1,j}(2 - hf_{i,j}) + h(f_{i,j-1} - f_{i,j+1}) - f_{i,j}(4 - hf_{i+1,j}) + 2f_{i+1,j} + 2h^2 g_{i,j} = 0 \quad (31)$$

where $i, j = 1, 2, \ldots, n - 1$. In particular, we solve the nonlinear system (31) for $n = 11$ so that $m = 100$ by selecting $f_{i,j} = -\frac{5}{2}$ (for $i, j = 1, 2, \ldots, 10$) as the initial value towards the required solution t^* given by

$$\begin{aligned} \Big\{ & -0.7546\ldots, -0.6892\ldots, -0.6290\ldots, -0.5750\ldots, -0.5235\ldots, \\ & -0.4817\ldots, -0.4306\ldots, -0.4213\ldots, \\ & -0.2583\ldots, -0.3068\ldots, -1.3583\ldots, -1.2405\ldots, -1.1322\ldots, \end{aligned}$$

$$
\begin{aligned}
&-1.0351\ldots, -0.9422\ldots, -0.8675\ldots,\\
&-0.7741\ldots, -0.7598\ldots, -0.4358\ldots, -0.5505\ldots, -1.8111\ldots,\\
&-1.6541\ldots, -1.5096\ldots, -1.3803\ldots,\\
&-1.2559\ldots, -1.1573\ldots, -1.0309\ldots, -1.0110\ldots, -0.6951\ldots,\\
&-0.7356\ldots, -2.1130\ldots, -1.9298\ldots,\\
&-1.7611\ldots, -1.6106\ldots, -1.4649\ldots, -1.3511\ldots, -1.2014\ldots,\\
&-1.1768\ldots, -0.8755\ldots, -0.8605\ldots,\\
&-2.2639\ldots, -2.0678\ldots, -1.8869\ldots, -1.7258\ldots, -1.5690\ldots,\\
&-1.4485\ldots, -1.2860\ldots, -1.2582\ldots,\\
&-0.9783\ldots, -0.9244\ldots, -2.2639\ldots, -2.0678\ldots, -1.8868\ldots,\\
&-1.7261\ldots, -1.5686\ldots, -1.4494\ldots,\\
&-1.2850\ldots, -1.2558\ldots, -1.0040\ldots, -0.9268\ldots, -2.1129\ldots,\\
&-1.9300\ldots, -1.7609\ldots, -1.6112\ldots,\\
&-1.4637\ldots, -1.3534\ldots, -1.1987\ldots, -1.1700\ldots, -0.9534\ldots,\\
&-0.8672\ldots, -1.8111\ldots, -1.6544\ldots,\\
&-1.5093\ldots, -1.3812\ldots, -1.2544\ldots, -1.1604\ldots, -1.0272\ldots,\\
&-1.0010\ldots, -0.8270\ldots, -0.7454\ldots,\\
&-1.3583\ldots, -1.2408\ldots, -1.1320\ldots, -1.0359\ldots, -0.9407\ldots,\\
&-0.8704\ldots, -0.7706\ldots, -0.7492\ldots,\\
&-0.6255\ldots, -0.5609\ldots, -0.7546\ldots, -0.6893\ldots, -0.6289\ldots,\\
&-0.5755\ldots, -0.5227\ldots, -0.4834\ldots,\\
&-0.4285\ldots, -0.4152\ldots, -0.3496\ldots, -0.3128\ldots \Big\}^{\mathrm{T}}.
\end{aligned}
$$

The approximate solution found is also plotted in Fig. 4. The concrete values of parameters for this system are $(m, \mu_0, \mu_1) = (100, 10.0503, 0.0492)$.

Example 7 Next, the methods are applied to solve the Poisson equation (Cordero et al. 2015)

$$u_{xx} + u_{yy} = u^2, \quad (x, y) \in [0, 1] \times [0, 1], \tag{32}$$

with boundary conditions

$$
\begin{aligned}
u(x, 0) &= 2x^2 - x + 1, \quad u(x, 1) = 2,\\
u(0, y) &= 2y^2 - y + 1, \quad u(1, y) = 2.
\end{aligned} \tag{33}
$$

The solution can be found using finite difference discretization. Let $u = u(x, y)$ be the exact solution of this Poisson equation. Let $w_{i,j} = u(x_i, y_j)$ be its approximate solution at the grid points of the mesh. Let M and N be the number of steps in x and y directions, and h and k be the respective step size ($h = \frac{1}{M}$, $k = \frac{1}{N}$). If we discretize the problem by using the central divided differences, i.e., $u_{xx}(x_i, y_j) = (w_{i+1,j} - 2w_{i,j} + w_{i-1,j})/h^2$ and $u_{yy}(x_i, y_j) = (w_{i,j+1} - 2w_{i,j} + w_{i,j-1})/k^2$, we get the following system of nonlinear equations:

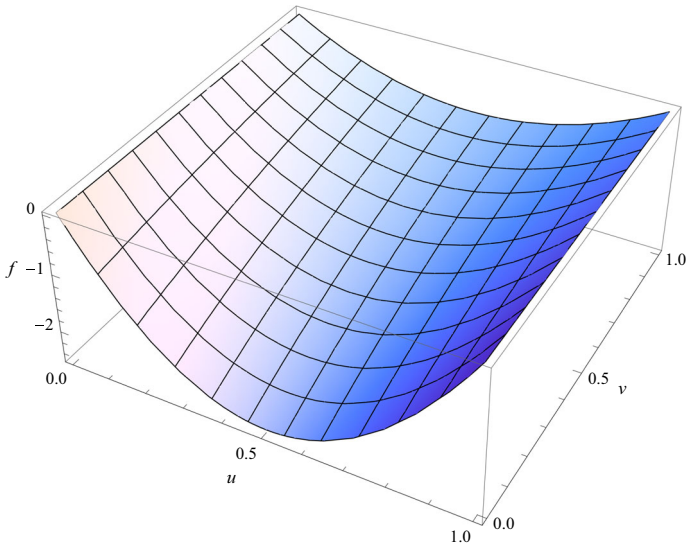

Fig. 4 Approximate solution of Burger equation

$$w_{i+1,j} - 4w_{i,j} + w_{i-1,j} + w_{i,j+1} + w_{i,j-1} - h^2 w_{i,j}^2 = 0,$$
$$i = 1, \ldots, M-1,\ j = 1, \ldots, N-1. \tag{34}$$

We consider $M = 15$ and $N = 15$ and thereby solve the resulting nonlinear system of 196 equations in 196 unknowns. The approximate solution

$$\Big\{0.9306\ldots, 0.9209\ldots, 0.9193\ldots, 0.9289\ldots, 0.9513\ldots, 0.9873\ldots,$$
$$1.0377\ldots, 1.1030\ldots, 1.1834\ldots,$$
$$1.2793\ldots, 1.3909\ldots, 1.5185\ldots, 1.6623\ldots, 1.8226\ldots, 0.9209\ldots, 0.9351\ldots,$$
$$0.9513\ldots, 0.9734\ldots,$$
$$1.0039\ldots, 1.0446\ldots, 1.0965\ldots, 1.1606\ldots, 1.2375\ldots,$$
$$1.3279\ldots, 1.4322\ldots, 1.5510\ldots, 1.6847\ldots,$$
$$1.8341\ldots, 0.9193\ldots, 0.9513\ldots, 0.9815\ldots, 1.0137\ldots,$$
$$1.0510\ldots, 1.0954\ldots, 1.1484\ldots, 1.2113\ldots,$$
$$1.2851\ldots, 1.3704\ldots, 1.4682\ldots, 1.5791\ldots, 1.7041\ldots,$$
$$1.8439\ldots, 0.9289\ldots, 0.9734\ldots, 1.0137\ldots,$$
$$1.0535\ldots, 1.0958\ldots, 1.1429\ldots, 1.1964\ldots, 1.2578\ldots,$$
$$1.3283\ldots, 1.4089\ldots, 1.5006\ldots, 1.6044\ldots,$$
$$1.7214\ldots, 1.8528\ldots, 0.9513\ldots, 1.0039\ldots, 1.0510\ldots,$$
$$1.0958\ldots, 1.1413\ldots, 1.1897\ldots, 1.2428\ldots,$$
$$1.3022\ldots, 1.3692\ldots, 1.4451\ldots, 1.5310\ldots, 1.6280\ldots,$$
$$1.7375\ldots, 1.8609\ldots, 0.9873\ldots, 1.0446\ldots,$$
$$1.0954\ldots, 1.1429\ldots, 1.1897\ldots, 1.2380\ldots, 1.2897\ldots,$$

$$
\begin{aligned}
&1.3465\ldots, 1.4097\ldots, 1.4806\ldots, 1.5606\ldots,\\
&1.6509\ldots, 1.7531\ldots, 1.8688\ldots, 1.0377\ldots, 1.0965\ldots,\\
&1.1484\ldots, 1.1964\ldots, 1.2428\ldots, 1.2897\ldots,\\
&1.3391\ldots, 1.3924\ldots, 1.4512\ldots, 1.5168\ldots, 1.5906\ldots,\\
&1.6741\ldots, 1.7689\ldots, 1.8768\ldots, 1.1030\ldots,\\
&1.1606\ldots, 1.2113\ldots, 1.2578\ldots, 1.3022\ldots, 1.3465\ldots,\\
&1.3924\ldots, 1.4415\ldots, 1.4952\ldots, 1.5550\ldots,\\
&1.6222\ldots, 1.6984\ldots, 1.7854\ldots, 1.8852\ldots, 1.1834\ldots,\\
&1.2375\ldots, 1.2851\ldots, 1.3283\ldots, 1.3692\ldots,\\
&1.4097\ldots, 1.4512\ldots, 1.4952\ldots, 1.5432\ldots, 1.5965\ldots,\\
&1.6565\ldots, 1.7248\ldots, 1.8034\ldots, 1.8943\ldots,\\
&1.2793\ldots, 1.3279\ldots, 1.3704\ldots, 1.4089\ldots, 1.4451\ldots,\\
&1.4806\ldots, 1.5167\ldots, 1.5550\ldots, 1.5965\ldots,\\
&1.6425\ldots, 1.6946\ldots, 1.7543\ldots, 1.8234\ldots, 1.9044\ldots,\\
&1.3909\ldots, 1.4322\ldots, 1.4682\ldots, 1.5006\ldots,\\
&1.5310\ldots, 1.5606\ldots, 1.5906\ldots, 1.6222\ldots, 1.6565\ldots,\\
&1.6946\ldots, 1.7379\ldots, 1.7879\ldots, 1.8465\ldots,\\
&1.9162\ldots, 1.5185\ldots, 1.5510\ldots, 1.5791\ldots, 1.6044\ldots,\\
&1.6280\ldots, 1.6509\ldots, 1.6741\ldots, 1.6984\ldots,\\
&1.7248\ldots, 1.7543\ldots, 1.7879\ldots, 1.8270\ldots, 1.8736\ldots,\\
&1.9301\ldots, 1.6623\ldots, 1.6847\ldots, 1.7041\ldots,\\
&1.7214\ldots, 1.7375\ldots, 1.7531\ldots, 1.7689\ldots, 1.7854\ldots,\\
&1.8034\ldots, 1.8234\ldots, 1.8465\ldots, 1.8736\ldots,\\
&1.9064\ldots, 1.9472\ldots, 1.8226\ldots, 1.8341\ldots, 1.8439\ldots,\\
&1.8528\ldots, 1.8609\ldots, 1.8688\ldots, 1.8768\ldots,\\
&1.8852\ldots, 1.8943\ldots, 1.9044\ldots, 1.9162\ldots, 1.9301\ldots, 1.9472\ldots, 1.9693\ldots
\end{aligned}
\Bigg\}^{\mathrm{T}}
$$

is evaluated with the initial vector $w_{i,j} = \{2, 2, \overset{196}{\cdots}, 2\}^{\mathrm{T}}$ $(i, j = 1, 2, \ldots, 14)$. The approximate solution found has also been displayed in Fig. 5. For this problem the concrete values of parameters are $(m, \mu_0, \mu_1) = (196, 3, 0.005102)$.

Example 8 Consider a system of 200 nonlinear equations (see Sharma and Arora 2016b)

$$\sum_{j=1, j\neq i}^{200} t_j - e^{-t_i} = 0,\ \ 1 \leq i \leq 200,$$

with initial value $t_0 = (\frac{3}{2}, \frac{3}{2}, \overset{m}{\cdots}, \frac{3}{2})^{\mathrm{T}}$ towards the required solution $t^* = (0.0050\ldots, \overset{200}{\cdots}, 0.0050\ldots)^{\mathrm{T}}$. The concrete values of parameters are $(m, \mu_0, \mu_1) = (200, 44.2674, 0.2213)$.

Example 9 Lastly, consider a system of 500 equations (see Sharma and Arora 2016b)

$$\begin{cases} t_i^2 t_{i+1} - 1 = 0,\ \ 1 \leq i \leq 499, \\ t_i^2 t_1 - 1 = 0,\ \ i = 500. \end{cases}$$

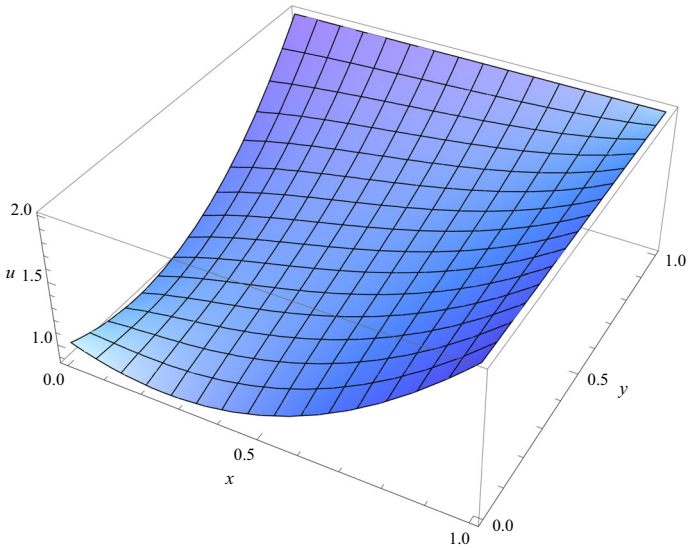

Fig. 5 Approximate solution of Poisson equation

To obtain the required solution $t^* = \{1, 1, \overset{500}{\cdots}, 1\}^{\mathrm{T}}$ of the system, the initial value chosen is $t_0 = (\frac{18}{10}, \frac{18}{10}, \overset{500}{\cdots}, \frac{18}{10})^{\mathrm{T}}$. Computed values of the parameters (m, μ_0, μ_1) are $(500, 2, 0.006)$.

The numerical results of the performance of various methods, applied to solve the above mentioned problems, are displayed in Tables 2 and 3. Here the values like $A(\pm m)$ denote $A \times 10^{\pm m}$. The numerical results clearly indicate the stable convergence behavior of the methods. Also observe that in each method the computational order of convergence overwhelmingly supports the theoretical order of convergence. The elapsed CPU time (e-time in seconds) used in the execution of program shows the efficient nature of proposed methods as compared to other methods. In fact speaking of the efficient nature of an iterative method we mean that the method with high efficiency uses less computing time than that of the method with low efficiency. Similar numerical experimentations, carried out for a number of problems of different type, confirmed the above conclusions to a large extent.

7 Conclusions

In the foregoing study, we have considered the problem of solving systems of nonlinear equations and developed two- and three-step composite Newton–Jarratt iterative methods of convergence order three and five, respectively. Furthermore, in quest of fast algorithms a generalized $q + 1$-step scheme with increasing convergence order $2q + 1$ is proposed and analyzed. Novelty of the $q + 1$-step algorithm is that in each step order of convergence is increased by an amount of two at the cost of only one additional function evaluation. Moreover, evaluation of inverse operator remains the same throughout the iteration which makes the algorithm more attractive and computationally efficient.

Computational efficiency of the methods is discussed in its general form. Then a comparison of efficiencies of the new schemes with existing schemes is shown. It is observed that

Table 2 Comparison of performance between third-order methods

Methods	$M_1^{(3)}$	$M_2^{(3)}$	$M_3^{(3)}$	$M_4^{(3)}$	$M_5^{(3)}$	$M_6^{(3)}$
Ex. 1						
k	5	5	5	5	5	6
COC	3.000	3.000	3.000	3.000	3.000	3.000
e-time	0.999	1.154	1.420	1.592	1.342	1.342
$\mathcal{E}_i^{(p)}$	2.655(2)	2.684(2)	1.955(2)	1.525(2)	1.949(2)	2.658(2)
Ex. 2						
k	4	5	4	4	4	5
COC	3.000	3.000	3.000	3.000	3.000	3.000
e-time	1.514	2.028	2.090	2.512	2.043	1.950
$\mathcal{E}_i^{(p)}$	1.812(2)	1.820(2)	1.345(2)	1.058(2)	1.340(2)	1.793(2)
Ex. 3						
k	5	5	5	5	5	5
COC	3.000	2.994	2.996	2.996	2.996	3.004
e-time	0.234	0.390	0.390	0.358	0.374	0.328
$\mathcal{E}_i^{(p)}$	4.247(2)	3.956(2)	3.226(2)	2.625(2)	3.179(2)	3.630(2)
Ex. 4						
k	6	6	6	6	6	6
COC	3.000	3.000	3.000	3.000	3.000	3.000
e-time	5.257	7.098	7.192	7.145	7.114	7.082
$\mathcal{E}_i^{(p)}$	6.280(1)	5.432(1)	5.047(1)	4.653(1)	5.024(1)	5.176(1)
Ex. 5						
k	6	6	6	6	6	6
COC	3.000	3.000	3.000	3.000	3.000	3.000
e-time	5.772	10.686	10.654	10.718	10.701	10.421
$\mathcal{E}_i^{(p)}$	2.133	1.191	1.157	1.124	1.157	1.159
Ex. 6						
k	5	5	5	5	5	5
COC	3.005	3.002	3.002	3.002	3.002	3.004
e-time	22.183	32.745	33.072	36.067	32.604	32.012
$\mathcal{E}_i^{(p)}$	2.947(−1)	1.563(−1)	1.540(−1)	1.517(−1)	1.540(−1)	1.541(−1)
Ex. 7						
k	4	4	4	4	4	4
COC	3.000	3.000	3.000	3.000	3.000	3.000
e-time	27.472	50.446	50.202	50.326	49.967	50.326
$\mathcal{E}_i^{(p)}$	4.140(−2)	2.133(−2)	2.117(−2)	2.101(−2)	2.117(−2)	2.117(−2)

Table 2 continued

Methods	$M_1^{(3)}$	$M_2^{(3)}$	$M_3^{(3)}$	$M_4^{(3)}$	$M_5^{(3)}$	$M_6^{(3)}$
Ex. 8						
k	4	4	4	4	4	4
COC	3.000	3.000	3.000	3.000	3.000	3.000
e-time	594.11	1138.32	1143.36	1154.00	1142.13	1148.75
$\mathcal{E}_i^{(p)}$	3.865(−2)	1.999(−2)	1.981(−2)	1.964(−2)	1.981(−2)	1.984(−2)
Ex. 9						
k	6	6	6	6	6	6
COC	3.000	3.000	3.000	3.000	3.000	3.000
e-time	41.718	73.929	77.563	77.673	75.287	75.847
$\mathcal{E}_i^{(p)}$	2.579(−3)	1.305(−3)	1.301(−3)	1.297(−3)	1.301(−3)	1.301(−3)

Table 3 Comparison of performance between fifth- and seventh-order methods

Methods	$M_1^{(5)}$	$M_2^{(5)}$	$M_3^{(5)}$	$M_4^{(5)}$	$M_5^{(5)}$	$M_1^{(7)}$	$M_2^{(7)}$
Ex. 1							
k	4	4	4	4	4	4	4
COC	5.000	5.000	5.000	5.000	5.000	7.000	7.000
e-time	0.936	1.107	1.139	1.154	1.217	0.920	1.342
$\mathcal{E}_i^{(p)}$	3.042(2)	2.601(2)	3.136(2)	3.104(2)	3.080(2)	3.019(2)	3.099(2)
Ex. 2							
k	3	3	3	4	3	3	3
COC	5.000	5.001	4.981	5.001	5.000	7.000	7.003
e-time	1.357	1.404	1.575	1.918	1.576	1.311	1.701
$\mathcal{E}_i^{(p)}$	2.023(2)	1.697(2)	2.072(2)	2.061(2)	2.046(2)	1.976(2)	2.031(2)
Ex. 3							
k	3	3	3	3	3	3	3
COC	5.005	5.013	5.001	4.996	4.990	7.011	6.995
e-time	0.187	0.359	0.375	0.280	0.281	0.188	0.297
$\mathcal{E}_i^{(p)}$	4.100(2)	3.262(2)	4.038(2)	4.218(2)	4.110(2)	3.696(2)	4.008(2)
Ex. 4							
k	4	4	4	4	4	4	4
COC	5.000	5.000	5.000	5.000	5.000	7.000	7.000
e-time	4.961	6.052	6.646	7.035	6.973	5.741	8.736
$\mathcal{E}_i^{(p)}$	5.353(1)	3.523(1)	4.680(1)	5.078(1)	5.046(1)	4.563(1)	4.507(1)

Table 3 continued

Methods	$M_1^{(5)}$	$M_2^{(5)}$	$M_3^{(5)}$	$M_4^{(5)}$	$M_5^{(5)}$	$M_1^{(7)}$	$M_2^{(7)}$
Ex. 5							
k	4	4	4	4	4	4	4
COC	5.000	5.000	5.000	5.000	5.000	7.000	7.000
e-time	4.352	10.093	10.577	7.800	7.753	3.791	6.599
$\mathcal{E}_i^{(p)}$	2.710	1.121	1.163	1.649	1.647	2.892	1.890
Ex. 6							
k	3	3	3	3	3	3	3
COC	5.024	5.032	5.024	5.035	5.024	7.013	7.016
e-time	22.744	40.046	40.950	31.543	22.744	17.004	25.568
$\mathcal{E}_i^{(p)}$	3.981(−1)	1.497(−1)	1.527(−1)	2.222(−1)	2.222(−1)	4.465(−1)	2.610(−1)
Ex. 7							
k	3	3	3	3	3	3	3
COC	5.000	5.000	5.000	5.000	5.000	7.000	7.000
e-time	24.929	59.920	59.733	43.353	43.227	27.362	45.460
$\mathcal{E}_i^{(p)}$	5.809(−2)	2.062(−2)	2.083(−2)	3.077(−2)	3.077(−2)	6.739(−2)	3.666(−2)
Ex. 8							
k	3	3	3	3	3	3	3
COC	5.000	5.000	5.000	5.000	5.000	7.000	7.000
e-time	486.10	1365.21	1356.99	1034.63	921.34	484.68	934.38
$\mathcal{E}_i^{(p)}$	5.415(−2)	1.932(−2)	1.953(−2)	2.881(−2)	2.881(−2)	6.273(−2)	3.429(−2)
Ex. 9							
k	4	4	4	4	4	4	4
COC	5.000	5.000	5.000	5.000	5.000	7.000	7.000
e-time	33.510	80.387	76.659	54.750	55.598	37.081	58.921
$\mathcal{E}_i^{(p)}$	3.713(−3)	1.269(−3)	1.274(−3)	1.900(−3)	1.900(−3)	4.412(−3)	2.284(−3)

the presented methods are more efficient than similar existing methods in general. Numerical examples are presented and the performance is compared with existing methods. Theoretical order of convergence is verified in the examples by calculating computational order of convergence. Comparison of the elapsed CPU time shows that, in general, the method with large efficiency uses less computing time than the method with small efficiency. This shows that the efficiency results are in agreement with the CPU time utilized in the execution of program.

References

Alzahrani AKH, Behl R, Alshomrani AS (2018) Some higher-order iteration functions for solving nonlinear models. Appl Math Comput 334:80–93

Argyros IK (2007) Computational theory of iterative methods. In: Chui CK, Wuytack L (eds) Series: studies in computational mathematics, vol 15. Elsevier Publ. Co., New York

Argyros IK, Regmi S (2019) Undergraduate research at Cameron University on iterative procedures in banach and other spaces. Nova Science Publisher, New York
Babajee DKR, Dauhoo MZ, Darvishi MT, Karami A, Barati A (2010) Analysis of two Chebyshev-like third order methods free from second derivatives for solving systems of nonlinear equations. J Comput Appl Math 233(8):2002–2012
Behl R, Cordero A, Motsa SS, Torregrosa JR (2017) Stable high-order iterative methods for solving nonlinear models. Appl Math Comput 303(15):70–80
Brent RP (1973) Some efficient algorithms for solving systems of nonlinear equations. SIAM J Numer Anal 10:327–344
Burden RL, Faires JD (2001) Numerical analysis. PWS Publishing Company, Boston
Choubey N, Panday B, Jaiswal JP (2018) Several two-point with memory iterative methods for solving nonlinear equations. Afr Mat 29(3–4):435–449
Cordero A, Torregrosa JR (2006) Variants of Newton's method for functions of several variables. Appl Math Comput 183(1):199–208
Cordero A, Torregrosa JR (2007) Variants of Newton's method using fifth-order quadrature formulas. Appl Math Comput 190(1):686–698
Cordero A, Hueso JL, Martínez E, Torregrosa JR (2010) A modified Newton–Jarratt's composition. Numer Algorithm 55(1):87–99
Cordero A, Hueso JL, Martínez E, Torregrosa JR (2012) Increasing the convergence order of an iterative method for nonlinear systems. Appl Math Lett 25(12):2369–2374
Cordero A, Feng L, Magreñán ÁA, Torregrosa JR (2015) A new fourth-order family for solving nonlinear problems and its dynamics. J Math Chem 53:893–910
Darvishi MT, Barati A (2007) Super cubic iterative methods to solve systems of nonlinear equations. Appl Math Comput 188(2):1678–1685
Esmaeili H, Ahmadi M (2015) An efficient three-step method to solve system of non linear equations. Appl Math Comput 266(1):1093–1101
Fousse L, Hanrot G, Lefèvre V, Pélissier P, Zimmermann P (2007) MPFR: a multiple-precision binary floating-point library with correct rounding. ACM Trans Math Softw 33(2):15
Homeier HHH (2004) A modified Newton method with cubic convergence: the multivariate case. J Comput Appl Math 169(1):161–169
Lotfi T, Bakhtiari P, Cordero A, Mahdiani K, Torregrosa JR (2015) Some new efficient multipoint iterative methods for solving nonlinear systems of equations. Int J Comput Math 92:1921–1934
McNamee JM (2007) Numerical methods for roots of polynomials, Part I. Elsevier, Amsterdam
Noor MA, Waseem M (2009) Some iterative methods for solving a system of nonlinear equations. Comput Math Appl 57(1):101–106
Ortega JM, Rheinboldt WC (1970) Iterative solutions of nonlinear equations in several variables. Academic Press, New York
Ostrowski AM (1960) Solution of equation and systems of equations. Academic Press, New York
Regmi S (2021) Optimized iterative methods with applications in diverse disciplines. Nova Science Publisher, New York
Sauer T (2012) Numerical analysis, 2nd edn. Pearson, Hoboken
Sharma JR, Arora H (2016a) Improved Newton-like methods for solving systems of nonlinear equations. SeMA 74(2):147–163
Sharma JR, Arora H (2016b) Efficient derivative-free numerical methods for solving systems of nonlinear equations. Comput Appl Math 35(1):269–284
Sharma JR, Arora H (2016c) A simple yet efficient derivative-free family of seventh order methods for systems of nonlinear equations. SeMA 73:59–75
Sharma JR, Gupta P (2014) An efficient fifth order method for solving systems of nonlinear equations. Comput Math Appl 67:591–601
Sharma JR, Sharma R, Bahl A (2016) An improved Newton–Traub composition for solving systems of nonlinear equations. Appl Math Comput 290:98–110
Traub JF (1964) Iterative methods for the solution of equations. Prentice-Hall, Hoboken
Weerkoon S, Fernando TGI (2000) A variant of Newton's method with accelerated third-order convergence. Appl Math Lett 13:87–93
Wolfram S (2003) The mathematica book, 5th edn. Wolfram Media, Champaign
Xiao X, Yin H (2015) A new class of methods with higher order of convergence for solving systems of nonlinear equations. Appl Math Comput 264:300–309
Xiao XY, Yin HW (2016) Increasing the order of convergence for iterative methods to solve nonlinear systems. Calcolo 53(3):285–300

Xiao X, Yin H (2017) Achieving higher order of convergence for solving systems of nonlinear equations. Appl Math Comput 311(C):251–261
Xiao X, Yin H (2018) Accelerating the convergence speed of iterative methods for solving nonlinear systems. Appl Math Comput 333:8–19