Available online at www.tjnsa.com
J. Nonlinear Sci. Appl. 9 (2016), 6021–6033
Research Article

Journal of Nonlinear Science and Applications
Print: ISSN 2008-1898 Online: ISSN 2008-1901

Frozen jacobian iterative method for solving systems of nonlinear equations: application to nonlinear IVPs and BVPs

Malik Zaka Ullah[a,b,*], Fayyaz Ahmad[b,c,g], Ali Saleh Alshomrani[a], A. K. Alzahrani[a], Metib Said Alghamdi[d], Shamshad Ahmad[e], Shahid Ahmad[f]

[a] *Department of Mathematics, Faculty of Science, King Abdulaziz University, P. O. Box 80203, Jeddah 21589, Saudi Arabia.*
[b] *Dipartimento di Scienza e Alta Tecnologia, Universita dell'Insubria, Via Valleggio 11, Como 22100, Italy.*
[c] *Departament de Física i Enginyeria Nuclear, Universitat Politècnica de Catalunya, Comte d'Urgell 187, 08036 Barcelona, Spain.*
[d] *Mathematics Department, Faculty of science Jazan University, P. O. Box 218, Jazan, Saudi Arabia.*
[e] *Department of Heat and Mass Transfer Technological Center, Technical University of Catalonia, Colom 11, 08222 Terrassa, Spain.*
[f] *Department of Mathematics, Government College University Lahore, Lahore, Pakistan.*
[g] *UCERD Islamabad, Pakistan.*

Communicated by N. Hussain

Abstract

Frozen Jacobian iterative methods are of practical interest to solve the system of nonlinear equations. A frozen Jacobian multi-step iterative method is presented. We divide the multi-step iterative method into two parts namely base method and multi-step part. The convergence order of the constructed frozen Jacobian iterative method is three, and we design the base method in a way that we can maximize the convergence order in the multi-step part. In the multi-step part, we utilize a single evaluation of the function, solve four systems of lower and upper triangular systems and a second frozen Jacobian. The attained convergence order per multi-step is four. Hence, the general formula for the convergence order is $3 + 4(m - 2)$ for $m \geq 2$ and m is the number of multi-steps. In a single instance of the iterative method, we employ only single inversion of the Jacobian in the form of LU factors that makes the method computationally cheaper because the LU factors are used to solve four system of lower and upper triangular systems repeatedly. The claimed convergence order is verified by computing the computational order of convergence for a system of nonlinear equations. The efficiency and validity of the proposed iterative method are narrated by solving many nonlinear initial and boundary value problems.

Keywords: Frozen Jacobian iterative methods, multi-step iterative methods, systems of nonlinear equations, nonlinear initial value problems, nonlinear boundary value problems.
2010 MSC: 65H10, 65N22.

1. Introduction

It is not always possible to get the closed form solution of a nonlinear problem, and iterative methods

*Corresponding author
Email address: mzhussain@kau.edu.sa,malik.zakaullah@uninsubria.it (Malik Zaka Ullah)

Received 2016-07-21

provide an alternative option to solve them. Most of the problems in science and engineering are modeled in the form of initial value problems (IVPs) and boundary value problems (BVPs). When it is hard to solve nonlinear IVPs and BVPs we utilize iterative methods. Usually, nonlinear IVPs and BVPs are discretized, and we get the associated system of nonlinear equations. Solving the system of nonlinear equations means that we solve indirectly nonlinear IVPs and BVPs. The classical iterative method for solving system of nonlinear equation is the Newton method [11]. We denote a system of nonlinear equations by

$$\mathbf{F}(\mathbf{y}) = \mathbf{0}\,,$$

where $\mathbf{F}(\mathbf{y}) = [f_1(\mathbf{y}), f_2(\mathbf{y}), f_3(\mathbf{y}), \cdots, f_n(\mathbf{y})]^T$, $\mathbf{y} = [y_1, y_2, y_3, \cdots, y_n]^T$ and $f_i(\cdot)$ is a nonlinear real-valued function. The classical Newton method can be written as

$$\text{NR} = \begin{cases} \mathbf{y}_0 = \text{initial guess}, \\ \mathbf{y}_{n+1} = \mathbf{y}_n - \mathbf{F}'(\mathbf{y}_n)^{-1}\mathbf{F}(\mathbf{y}_n), \end{cases}$$

$\det(\mathbf{F}'(\mathbf{y}_n)) \neq 0$. The multi-step frozen Jacobian version of the Newton method (MNR) can be written as

$$\text{MNR} = \begin{cases} \text{Number of steps} = m \geq 1, \\ \text{Convergence order} = m+1, \\ \text{Function evaluations} = m, \\ \text{Jacobian evaluations} = 1, \\ \text{Number of LU-factorization} = 1, \\ \text{Number of solutions of lower} \\ \text{ and upper triangular systems } = m, \end{cases} \qquad \begin{cases} \text{Base method} \rightarrow & \begin{cases} \mathbf{y}_0 = \text{initial guess}, \\ \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_1 = \mathbf{F}(\mathbf{y}_0), \\ \mathbf{y}_1 = \mathbf{y}_0 - \boldsymbol{\phi}_1, \end{cases} \\ \text{Multi-step part} \rightarrow & \begin{cases} \text{for } s = 1, m-1, \\ \qquad \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_{s+1} = \mathbf{F}(\mathbf{y}_s), \\ \qquad \mathbf{y}_{s+1} = \mathbf{y}_s - \boldsymbol{\phi}_{s+1}, \\ \text{end}, \\ \mathbf{y}_0 = \mathbf{y}_m. \end{cases} \end{cases}$$

There are also some other classical iterative methods that have convergence order three. For instance, the Halley [7, 10] and Chebyshev [17] methods

$$\text{Halley Method} = \begin{cases} \mathbf{y}_n = \text{initial guess}, \\ \mathbf{F}'(\mathbf{y}_n)\,\boldsymbol{\phi} = \mathbf{F}(\mathbf{y}_n), \\ \mathbf{F}'(\mathbf{y}_n)\,\mathbf{L}(\mathbf{y}_n) = \mathbf{F}''(\mathbf{y}_n)\boldsymbol{\phi}, \\ \mathbf{y}_{n+1} = \mathbf{y}_n - \left[I - \frac{1}{2}\mathbf{L}(\mathbf{y}_n)\right]^{-1}\boldsymbol{\phi}, \end{cases} , \quad \text{Chebyshev Method} = \begin{cases} \mathbf{y}_n = \text{initial guess}, \\ \mathbf{F}'(\mathbf{y}_n)\,\boldsymbol{\phi} = \mathbf{F}(\mathbf{y}_n), \\ \mathbf{F}'(\mathbf{y}_n)\,\mathbf{L}(\mathbf{y}_n) = \mathbf{F}''(\mathbf{y}_n)\boldsymbol{\phi}, \\ \mathbf{y}_{n+1} = \mathbf{y}_n - \left[I + \frac{1}{2}\mathbf{L}(\mathbf{y}_n)\right]\boldsymbol{\phi}. \end{cases}$$

The Halley and Chebyshev iterative methods are computationally expensive. When we ask the question about the optimality of convergence order of an iterative method for a given number of function evaluations, this question has the answer in the case of single nonlinear equations, but in the case of the system of nonlinear equations, we do not have any answer. According to Kunge-Traub[11] conjecture, an iterative method, without memory for solving a single nonlinear equation, could achieve maximum convergence order 2^{s-1}, and s is the total number of function evaluations. The iterative methods for solving nonlinear equations [5, 9, 13, 16, 19] have attained the proper attention of a large community of researchers. Sometimes it is possible to generalize an iterative method for solving nonlinear equations with an iterative method for solving system of nonlinear equations. For instance, in the case of Newton method, we have this nice generalization but there is a significant number of iterative solvers for solving nonlinear equations do not have such kind of generalization and reason is that we can not adopt the arithmetic designed for the scalar numbers to vectors. It means when we develop the iterative method for solving system of nonlinear equations we have constraints on vector arithmetic operations. Many researchers have made good effort to construct iterative methods [1–3, 6, 12, 14, 17, 18] to solve system of nonlinear equations. Some authors have designed frozen Jacobian multi-step iterative methods for solving system of nonlinear equations. The details of some iterative methods are enclosed in this article. Montazeri et al. [12] proposed an iterative method HJ for solving systems of nonlinear equations and this method has convergence order $2m$ for $m \geq 2$, here m is the step number. In fact, this method is a frozen Jacobian multi-step iterative method, and its base method has convergence order four. The per multi-step increment in the convergence order is two. The HJ method is an efficient iterative method because it requires only one inversion of Jacobian (regarding LU factors) and the information of frozen Jacobian is repeatedly utilized to solve lower and upper triangular systems. A new method FTUC [2] has better convergence order comparing with that of HJ. The convergence order of the base method of

FTUC is five and multi-step increment in the convergence order is three. The efficiency of FTUC method is better than HJ method. There is the other method MSF [17] that has $3m$ convergence order but this method is designed only for solving the system of weakly nonlinear equations. The applicability of MSF is limited in the case of the general system of nonlinear equations because it requires the computation of second order Fréchet derivative.

$$
\mathrm{HJ}=\begin{cases}
\text{Number of steps} = m \geq 2,\\
\text{Convergence order} = 2m,\\
\text{Function evaluations} = m-1,\\
\text{Jacobian evaluations} = 2,\\
\text{LU-factorization} = 1,\\
\text{Matrix-vector multiplications} = m,\\
\text{Vector-vector multiplications} = 2m,\\
\text{Number of solutions of lower}\\
\text{ and upper triangular systems}\\
\text{of equations } = 2m-1,
\end{cases}
\quad
\begin{cases}
\text{Base method} \longrightarrow & \begin{cases}
\mathbf{y}_0 = \text{initial guess},\\
\mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_1 = \mathbf{F}(\mathbf{y}_0),\\
\mathbf{y}_1 = \mathbf{y}_0 - \frac{2}{3}\boldsymbol{\phi}_1,\\
\mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_2 = \mathbf{F}'(\mathbf{y}_1)\boldsymbol{\phi}_1,\\
\mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_3 = \mathbf{F}'(\mathbf{y}_1)\boldsymbol{\phi}_2,\\
\mathbf{y}_2 = \mathbf{y}_0 - \frac{23}{8}\boldsymbol{\phi}_1 + 3\boldsymbol{\phi}_2 - \frac{9}{8}\boldsymbol{\phi}_3,
\end{cases}\\
\text{Multi-step part} \rightarrow & \begin{cases}
\text{for } s = 3, m,\\
\quad \mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_4 = \mathbf{F}(\mathbf{y}_{s+1}),\\
\quad \mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_5 = \mathbf{F}'(\mathbf{y}_1)\boldsymbol{\phi}_4,\\
\quad \mathbf{y}_s = \mathbf{y}_{s-1} - \frac{5}{2}\boldsymbol{\phi}_4 + \frac{3}{2}\boldsymbol{\phi}_5,\\
\text{end},\\
\mathbf{y}_0 = \mathbf{y}_m,
\end{cases}
\end{cases}
$$

$$
\mathrm{FTUC}=\begin{cases}
\text{Number of steps} = m \geq 3,\\
\text{Convergence-order} = 3m-4,\\
\text{Function evaluations} = m-1,\\
\text{Jacobian evaluations} = 2,\\
\text{LU-factorization } = 1,\\
\text{Matrix-vector multiplications} = m-1,\\
\text{Vector-vector multiplications} = m+1,\\
\text{Number of solutions of lower}\\
\text{and upper triangular systems}\\
\text{of equations} = 2m-2,
\end{cases}
\quad
\begin{cases}
\text{Base method} \longrightarrow & \begin{cases}
\mathbf{y}_0 = \text{initial guess},\\
\mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_1 = \mathbf{F}(\mathbf{y}_0),\\
\mathbf{y}_1 = \mathbf{y}_0 - \boldsymbol{\phi}_1,\\
\mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_2 = \mathbf{F}(\mathbf{y}_1),\\
\mathbf{y}_2 = \mathbf{y}_1 - 3\boldsymbol{\phi}_2,\\
\mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_3 = \mathbf{F}'(\mathbf{y}_2)\boldsymbol{\phi}_2,\\
\mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_4 = \mathbf{F}'(\mathbf{y}_2)\boldsymbol{\phi}_3,\\
\mathbf{y}_3 = \mathbf{y}_1 - \frac{7}{4}\boldsymbol{\phi}_2 + \frac{1}{2}\boldsymbol{\phi}_3 + \frac{1}{4}\boldsymbol{\phi}_4,
\end{cases}\\
\text{Multi-step part} \rightarrow & \begin{cases}
\text{for } s = 4, m,\\
\quad \mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_5 = \mathbf{F}(\mathbf{y}_s),\\
\quad \mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_6 = \mathbf{F}'(\mathbf{y}_2)\boldsymbol{\phi}_5,\\
\quad \mathbf{y}_s = \mathbf{y}_{s-1} - 2\boldsymbol{\phi}_5 + \boldsymbol{\phi}_6,\\
\text{end},\\
\mathbf{y}_0 = \mathbf{y}_m,
\end{cases}
\end{cases}
$$

$$
\mathrm{MSF}=\begin{cases}
\text{Number of steps} = m,\\
\text{Convergence-order} = 3m,\\
\text{Function evaluations} = m,\\
\text{Jacobian evaluations} = 2,\\
\text{Second-order Fréchet derivative} = 1,\\
\text{LU-factorization} = 1,\\
\text{Matrix-vector multiplications} = 2m-2,\\
\text{Vector-vector multiplications} = m+2,\\
\text{Number of solutions of lower}\\
\text{ and upper triangular systems}\\
\text{ of equations} = 3m-1,
\end{cases}
\quad
\begin{cases}
\text{Base method} \longrightarrow & \begin{cases}
\mathbf{y}_0 = \text{initial guess},\\
\mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_1 = \mathbf{F}(\mathbf{y}_0),\\
\mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_2 = \mathbf{F}''(\mathbf{y}_0)\boldsymbol{\phi}_1^2,\\
\mathbf{y}_1 = \mathbf{y}_0 - \boldsymbol{\phi}_1 - \frac{1}{2}\boldsymbol{\phi}_2,
\end{cases}\\
\text{Multi-step part} \rightarrow & \begin{cases}
\text{for } s = 2, m,\\
\quad \mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_3 = \mathbf{F}(\mathbf{y}_s),\\
\quad \mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_4 = \mathbf{F}'(\mathbf{y}_1)\boldsymbol{\phi}_3,\\
\quad \mathbf{F}'(\mathbf{y}_0)\boldsymbol{\phi}_5 = \mathbf{F}'(\mathbf{y}_1)\boldsymbol{\phi}_4,\\
\quad \mathbf{y}_s = \mathbf{y}_{s-1} - 3\boldsymbol{\phi}_3 + 3\boldsymbol{\phi}_4 - \boldsymbol{\phi}_5,\\
\text{end},\\
\mathbf{y}_0 = \mathbf{y}_m.
\end{cases}
\end{cases}
$$

We design a new iterative method that has better convergence order, and it is a frozen Jacobian multi-step iterative method. The efficiency of the proposed iterative method is hidden in the multi-step part because the multi-step part requires a single evaluation of the system of nonlinear equations and solution of four lower and upper triangular systems.

2. New multi-step iterative method

The new multi-step iterative method (MFAA) can be described as

$$
\text{MFAA} = \begin{cases} \text{Number of steps} = m \geq 2, \\ \text{Convergence order} = 4m-5, \\ \text{Function evaluations} = m, \\ \text{Jacobian evaluations} = 2, \\ \text{LU-factorization} = 1, \\ \text{Matrix-vector }, \\ \ \text{multiplications} = 3(m-2), \\ \text{Vector-vector}, \\ \ \text{multiplications} = 4(m-2), \\ \text{Number of solutions of lower }, \\ \ \text{and upper triangular }, \\ \ \text{systems of equations} = 4m-6, \end{cases}
\begin{cases} \text{Base method} \rightarrow & \begin{cases} \mathbf{y}_0 = \text{initial guess}, \\ \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_1 = \mathbf{F}(\mathbf{y}_0), \\ \mathbf{y}_1 = \mathbf{y}_0 - \boldsymbol{\phi}_1, \\ \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_2 = \mathbf{F}(\mathbf{y}_1), \\ \mathbf{y}_2 = \mathbf{y}_0 - \boldsymbol{\phi}_1 - \boldsymbol{\phi}_2, \\ \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_3 = \mathbf{F}(\mathbf{y}_2), \\ \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_4 = \mathbf{F}'(\mathbf{y}_2)\boldsymbol{\phi}_3, \\ \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_5 = \mathbf{F}'(\mathbf{y}_2)\boldsymbol{\phi}_4, \\ \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_6 = \mathbf{F}'(\mathbf{y}_2)\boldsymbol{\phi}_5, \\ \mathbf{y}_3 = \mathbf{y}_2 - 17/4\boldsymbol{\phi}_3, \\ \qquad +27/4\boldsymbol{\phi}_4 - 19/4\boldsymbol{\phi}_5, \\ \qquad +5/4\boldsymbol{\phi}_6, \end{cases} \\ \text{Multi-step part} \rightarrow & \begin{cases} \text{for } s = 4,\, m, \\ \qquad \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_7 = \mathbf{F}(\mathbf{y}_{s-1}), \\ \qquad \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_8 = \mathbf{F}'(\mathbf{y}_2)\boldsymbol{\phi}_7, \\ \qquad \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_9 = \mathbf{F}'(\mathbf{y}_2)\boldsymbol{\phi}_8, \\ \qquad \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_{10} = \mathbf{F}'(\mathbf{y}_2)\boldsymbol{\phi}_9, \\ \qquad \mathbf{y}_s = \mathbf{y}_{s-1} - 9/2\boldsymbol{\phi}_7, \\ \qquad\quad +15/2\boldsymbol{\phi}_8 - 11/2\boldsymbol{\phi}_9, \\ \qquad\quad +3/2\boldsymbol{\phi}_{10}, \\ \text{end}, \\ \mathbf{y}_0 = \mathbf{y}_m. \end{cases} \end{cases}
$$

The convergence order of MFAA is $4m - 5$ and m is the step number. The proposed iterative method requires two Jacobian evaluations and $4m - 6$ solutions of lower and upper triangular systems. The per multi-step increment in the convergence order is four, and it makes the method highly convergent. The computational cost of different binary operations is given in Table 1.

Table 1: Computational cost of different operations (the computational cost of a division is l times to multiplication).

LU-factorization	
Multiplications	$\frac{n(n-1)(2n-1)}{6}$
Divisions	$\frac{n(n-1)}{2}$
Total cost	$\frac{n(n-1)(2n-1)}{6} + l\frac{n(n-1)}{2}$
Solution of lower and upper triangular systems	
Multiplications	$n(n-1)$
Divisions	n
Total cost	$n(n-1) + l\,n$
Scalar-vector multiplication	n
Component-wise vector-vector multiplication	n
Matrix-vector multiplication	n^2

To make comparison between different iterative methods, the efficiency index is defined as

$$E.I. = \rho^{1/C},$$

where C is the computational cost. The computational cost of different iterative methods is shown in Table 2 when the number of multi-step are equal. Table 3 displays the computational cost differences of the different iterative method with the computational cost of MFAA iterative method when the convergence orders of all methods are same. The conditions are shown in Table 4 when the cost of our proposed iterative method MFAA is less than the computational cost of other iterative methods.

Table 2: Computational cost of different iterative methods when the number of steps m are equal, the average computational cost of $f_i\left(f'_{ij},\, f''_i\right)$ is $\alpha(\beta,\,\gamma)$ and $\alpha(\beta,\,\gamma)$ is the ratio between the average computational cost of $f_i\left(f'_{ij},\, f''_i\right)$ and the computational cost of a multiplication.

Methods	Total computational cost
MNR	$1/3\,n^3 + (\beta + m - 1/2 + l/2)\,n^2 + (m\alpha + lm - m + 1/6 - l/2)\,n$
HJ	$1/3\,n^3 + (2\,\beta - 3/2 + l/2 + 3\,m)\,n^2 + (m\alpha - \alpha + 7/6 - 3/2\,l + 2\,lm)\,n$
FTUC	$1/3\,n^3 + (2\,\beta - 7/2 + l/2 + 3\,m)\,n^2 + \left(m\alpha - \alpha + \frac{19}{6} - 5/2\,l - m + 2\,lm\right)n$
MSF	$1/3\,n^3 + (2\,\beta - 7/2 + l/2 + 5\,m)\,n^2 + \left(m\alpha + \gamma + \frac{19}{6} - 3/2\,l - 2\,m + 3\,lm\right)n$
MFAA	$1/3\,n^3 + \left(2\,\beta - \frac{25}{2} + l/2 + 7\,m\right)n^2 + \left(m\alpha - \frac{11}{6} - 13/2\,l + 4\,lm\right)n$

Table 3: Difference of computational cost of different iterative methods when the convergence orders of iterative methods is equal to the convergence order of MFAA.

Methods	Difference of total computational costs
MNR-MFAA	$(-\beta - 3\,m + 6)\,n^2 + (3\,m\alpha - 6\,\alpha - 4\,m + 8)\,n$
HJ-MFAA	$(7/2 - m)\,n^2 + (m\alpha + 3 - 7/2\,\alpha)\,n$
FTUC-MFAA	$(8 - 3\,m)\,n^2 + (1/3\,m\alpha + 16/3 + 10/3\,l - 4/3\,lm - 4/3\,\alpha - 4/3\,m)\,n$
MSF-MFAA	$(-m/3 + 2/3)\,n^2 + \left(1/3\,m\alpha + \frac{25}{3} - 5/3\,\alpha + \gamma - 8/3\,m\right)n$

Table 4: Comparison of computational cost of different iterative methods with MFAA method when convergence orders are equal.

Computational cost	Bounds on m	Conditions
$C_{\text{MNR}} - C_{\text{MFAA}} > 0$	$2 \leq \frac{\beta\,n - 6\,n + 6\,\alpha - 8}{3\,\alpha - 3\,n - 4} < m$	$\{n < \alpha - 4/3, 4/3 < \alpha\}$
$C_{\text{HJ}} - C_{\text{MFAA}} > 0$	$m = 3$	$\alpha < n$
	$3 < m < 1/2\,\frac{7\,\alpha - 7\,n - 6}{\alpha - n}$	$\alpha < n$
	$3 \leq m$	$\alpha = n + 2$
	$2 \leq 1/2\,\frac{7\,\alpha - 7\,n - 6}{\alpha - n} < m$	$n + 2 < \alpha$
$C_{\text{FTUC}} - C_{\text{MFAA}} > 0$	$2 \leq 2\,\frac{2\,\alpha - 5\,l - 12\,n - 8}{\alpha - 4\,l - 9\,n - 4} < m$	$4\,l + 9\,n + 4 < \alpha$
$C_{\text{MSF}} - C_{\text{MFAA}} > 0$	$m \geq 3$	$\Big\{\alpha < n + 8\,, -1/3\,m\alpha + 1/3\,mn$
		$+8/3\,m + 5/3\,\alpha - 2/3\,n - \frac{25}{3} < \gamma\Big\}$
	$2 \leq -\frac{2\,n + 25 - 5\,\alpha + 3\,\gamma}{\alpha - n - 8} < m$	$1/3\,n^2 - (\alpha/3 - 8/3)\,n < 0$

3. Convergence analysis

The proof of convergence order of the iterative method MFAA is established in this section via the mathematical induction. First, we will prove the convergence order of the iterative method MFAA for $m = 3$ and then for $m > 3$. In our convergence analysis, Taylor's series helps us in the expansion of the system of nonlinear equation around the simple root and hence, we deal with higher order Fréchet derivatives. The constraint of Fréchet differentiability on the system of nonlinear equations is essential because it is the Fréchet differentiability that is the responsible for linearization of the system of nonlinear equation. On the contrary, in Gâteaux differentiability does not have this nice property of linearization which is the soul of Newton-like methods. A function $\mathbf{F}(\cdot)$ is said to be Fréchet differentiable at a point $\mathbf{y}$ if there exists a linear operator $\mathbf{A} \in \mathbb{L}(\mathbb{R}^n, \mathbb{R}^q)$ such that

$$\lim_{\mathbf{h}\to\mathbf{0}} \frac{||\mathbf{F}(\mathbf{y}+\mathbf{h}) - \mathbf{F}(\mathbf{y}) - \mathbf{A}\mathbf{h}||}{||\mathbf{h}||} = 0\,.$$

The linear operator $\mathbf{A}$ is called the first order Fréchet derivative and we denote it by $\mathbf{F}'(\mathbf{y})$. The higher order Fréchet derivatives can be computed recursively as follows:

$$\begin{aligned} \mathbf{F}'(\mathbf{y}) &= \text{Jacobian}\,(\mathbf{F}(\mathbf{y}))\,, \\ \mathbf{F}^j(\mathbf{y})\mathbf{v}^{j-1} &= \text{Jacobian}\left(\mathbf{F}^{j-1}(\mathbf{y})\mathbf{v}^{j-1}\right), \quad j \geq 2\,, \end{aligned}$$

where $\mathbf{v}$ is a vector independent from $\mathbf{y}$.

Theorem 3.1. *Let $\mathbf{F} : \Gamma \subseteq \mathbb{R}^n \to \mathbb{R}^n$ be a sufficiently Fréchet differentiable function on an open convex neighborhood Γ of $\mathbf{y}^* \in \mathbb{R}^n$ with $\mathbf{F}(\mathbf{y}^*) = 0$ and $\det(\mathbf{F}'(\mathbf{y}^*)) \neq 0$, where $\mathbf{F}'(\mathbf{y})$ denotes the Fréchet derivative of $\mathbf{F}(\mathbf{y})$. Let $\mathbf{A}_1 = \mathbf{F}'(\mathbf{y}^*)$ and $\mathbf{A}_j = \frac{1}{j!}\mathbf{F}'(\mathbf{y}^*)^{-1}\mathbf{F}^{(j)}(\mathbf{y}^*)$ for $j \geq 2$, where $\mathbf{F}^{(j)}(\mathbf{y})$ denotes j-order Fréchet derivative of $\mathbf{F}(\mathbf{y})$. Then, for $m = 3$, with an initial guess in the neighborhood of $\mathbf{y}^*$, the sequence $\{\mathbf{y}_m\}$ generated by MFAA converges to $\mathbf{y}^*$ with local order of convergence at least seven and error*

$$\mathbf{e}_3 = \mathbf{L}{\mathbf{e}_0}^7 + O\left({\mathbf{e}_0}^8\right),$$

where $\mathbf{e}_0 = \mathbf{y}_0 - \mathbf{y}^*$, ${\mathbf{e}_0}^p = \overbrace{(\mathbf{e}_0, \mathbf{e}_0, \ldots, \mathbf{e}_0)}^{p \text{ times}}$, and $\mathbf{L} = 2\mathbf{A}_2^2\mathbf{A}_3\mathbf{A}_2^2 - 6\mathbf{A}_3\mathbf{A}_2^4 + 46\mathbf{A}_2^6$ is a 7-linear function, i.e. $\mathbf{L} \in \mathbb{L}\overbrace{(\mathbb{R}^n, \mathbb{R}^n, \mathbb{R}^n, \cdots, \mathbb{R}^n)}^{7\text{-times}}$ with $\mathbf{L}{\mathbf{e}_0}^7 \in \mathbb{R}^n$.

Proof. We define the error at the nth step $\mathbf{e}_n = \mathbf{y}_n - \mathbf{y}^*$. To complete the convergence proof, we performed the detailed computations by using Maple and details are provided below in sequence.

$$\begin{aligned}
\mathbf{F}(\mathbf{y}_0) =& \mathbf{A}_1\Big(\mathbf{e}_0 + \mathbf{A}_2\mathbf{e}_0^2 + \mathbf{A}_3\mathbf{e}_0^3 + \mathbf{A}_4\mathbf{e}_0^4 + \mathbf{A}_5\mathbf{e}_0^5 + \mathbf{A}_6\mathbf{e}_0^6 + \mathbf{A}_7\mathbf{e}_0^7 + O\left(\mathbf{e}_0^8\right)\Big) \\
\mathbf{F}^{-1}(\mathbf{y}_0) =& \Big(\mathbf{I} - 2\mathbf{A}_2\mathbf{e}_0 + \Big(-3\mathbf{A}_3 + 4\mathbf{A}_2^2\Big)\mathbf{e}_0^2 + \Big(-4\mathbf{A}_4 + 6\mathbf{A}_3\mathbf{A}_2 + 6\mathbf{A}_2\mathbf{A}_3 - 8\mathbf{A}_2^3\Big)\mathbf{e}_0^3 + \Big(-5\mathbf{A}_5 + 8\mathbf{A}_4\mathbf{A}_2 \\
& + 9\mathbf{A}_3^2 + 8\mathbf{A}_2\mathbf{A}_4 - 12\mathbf{A}_3\mathbf{A}_2^2 - 12\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2 - 12\mathbf{A}_2^2\mathbf{A}_3 + 16\mathbf{A}_2^4\Big)\mathbf{e}_0^4 \\
& + \Big(-6\mathbf{A}_6 + 10\mathbf{A}_5\mathbf{A}_2 + 12\mathbf{A}_4\mathbf{A}_3 + 12\mathbf{A}_3\mathbf{A}_4 + 10\mathbf{A}_2\mathbf{A}_5 - 16\mathbf{A}_4\mathbf{A}_2^2 - 18\mathbf{A}_3^2\mathbf{A}_2 - 16\mathbf{A}_2\mathbf{A}_4\mathbf{A}_2 \\
& - 18\mathbf{A}_3\mathbf{A}_2\mathbf{A}_3 - 18\mathbf{A}_2\mathbf{A}_3^2 - 16\mathbf{A}_2^2\mathbf{A}_4 + 24\mathbf{A}_3\mathbf{A}_2^3 + 24\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2^2 + 24\mathbf{A}_2^2\mathbf{A}_3\mathbf{A}_2 \\
& + 24\mathbf{A}_2^3\mathbf{A}_3 - 32\mathbf{A}_2^5\Big)\mathbf{e}_0^5 + \cdots + O\Big(\mathbf{e}_0^8\Big)\Big)\mathbf{A}_1^{-1}, \\
\mathbf{e}_1 =& \mathbf{A}_2\mathbf{e}_0^2 + \Big(2\mathbf{A}_3 - 2\mathbf{A}_2^2\Big)\mathbf{e}_0^3 + \Big(4\mathbf{A}_2^3 + 3\mathbf{A}_4 - 4\mathbf{A}_2\mathbf{A}_3 - 3\mathbf{A}_3\mathbf{A}_2\Big)\mathbf{e}_0^4 + \Big(6\mathbf{A}_3\mathbf{A}_2^2 + 6\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2 - 8\mathbf{A}_2^4 \\
& + 8\mathbf{A}_2^2\mathbf{A}_3 + 4\mathbf{A}_5 - 6\mathbf{A}_2\mathbf{A}_4 - 6\mathbf{A}_3^2 - 4\mathbf{A}_4\mathbf{A}_2\Big)\mathbf{e}_0^5 + \cdots + O\Big(\mathbf{e}_0^8\Big), \\
\mathbf{F}(\mathbf{y}_1) =& \mathbf{A}_1\Big(\mathbf{A}_2\mathbf{e}_0^2 + \Big(2\mathbf{A}_3 - 2\mathbf{A}_2^2\Big)\mathbf{e}_0^3 + \Big(5\mathbf{A}_2^3 + 3\mathbf{A}_4 - 4\mathbf{A}_2\mathbf{A}_3 - 3\mathbf{A}_3\mathbf{A}_2\Big)\mathbf{e}_0^4 + \Big(6\mathbf{A}_3\mathbf{A}_2^2 \\
& + 8\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2 - 12\mathbf{A}_2^4 + 10\mathbf{A}_2^2\mathbf{A}_3 + 4\mathbf{A}_5 - 6\mathbf{A}_2\mathbf{A}_4 - 6\mathbf{A}_3^2 - 4\mathbf{A}_4\mathbf{A}_2\Big)\mathbf{e}_0^5 + \Big(-19\mathbf{A}_2^2\mathbf{A}_3\mathbf{A}_2 \\
& - 19\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2^2 - 11\mathbf{A}_3\mathbf{A}_2^3 + 11\mathbf{A}_2\mathbf{A}_4\mathbf{A}_2 + 9\mathbf{A}_3^2\mathbf{A}_2 + 8\mathbf{A}_4\mathbf{A}_2^2 + 16\mathbf{A}_2\mathbf{A}_3^2 + 12\mathbf{A}_3\mathbf{A}_2\mathbf{A}_3 \\
& + 15\mathbf{A}_2^2\mathbf{A}_4 + 28\mathbf{A}_2^5 + 5\mathbf{A}_6 - 24\mathbf{A}_2^3\mathbf{A}_3 - 8\mathbf{A}_2\mathbf{A}_5 - 9\mathbf{A}_3\mathbf{A}_4 - 8\mathbf{A}_4\mathbf{A}_3 - 5\mathbf{A}_5\mathbf{A}_2\Big)\mathbf{e}_0^6 \\
& + \cdots + O\Big(\mathbf{e}_0^8\Big)\Big), \\
\boldsymbol{\phi}_2 =& \mathbf{A}_2\mathbf{e}_0^2 + \Big(-4\mathbf{A}_2^2 + 2\mathbf{A}_3\Big)\mathbf{e}_0^3 + \Big(-6\mathbf{A}_3\mathbf{A}_2 + 13\mathbf{A}_2^3 - 8\mathbf{A}_2\mathbf{A}_3 + 3\mathbf{A}_4\Big)\mathbf{e}_0^4 + \Big(-38\mathbf{A}_2^4 - 8\mathbf{A}_4\mathbf{A}_2 \\
& + 20\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2 + 18\mathbf{A}_3\mathbf{A}_2^2 - 12\mathbf{A}_3^2 + 26\mathbf{A}_2^2\mathbf{A}_3 - 12\mathbf{A}_2\mathbf{A}_4 + 4\mathbf{A}_5\Big)\mathbf{e}_0^5 + \Big(-59\mathbf{A}_2^2\mathbf{A}_3\mathbf{A}_2 \\
& - 55\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2^2 - 50\mathbf{A}_3\mathbf{A}_2^3 + 27\mathbf{A}_2\mathbf{A}_4\mathbf{A}_2 + 27\mathbf{A}_3^2\mathbf{A}_2 + 24\mathbf{A}_4\mathbf{A}_2^2 + 40\mathbf{A}_2\mathbf{A}_3^2 + 36\mathbf{A}_3\mathbf{A}_2\mathbf{A}_3 \\
& + 39\mathbf{A}_2^2\mathbf{A}_4 + 104\mathbf{A}_2^5 + 5\mathbf{A}_6 - 76\mathbf{A}_2^3\mathbf{A}_3 - 16\mathbf{A}_2\mathbf{A}_5 - 18\mathbf{A}_3\mathbf{A}_4 - 16\mathbf{A}_4\mathbf{A}_3 - 10\mathbf{A}_5\mathbf{A}_2\Big)\mathbf{e}_0^6 \\
& + \cdots + O\Big(\mathbf{e}_0^8\Big), \\
\mathbf{e}_2 =& 2\mathbf{e}_0^3\mathbf{A}_2^2 + \Big(-9\mathbf{A}_2^3 + 4\mathbf{A}_2\mathbf{A}_3 + 3\mathbf{A}_3\mathbf{A}_2\Big)\mathbf{e}_0^4 + \Big(-12\mathbf{A}_3\mathbf{A}_2^2 - 14\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2 + 30\mathbf{A}_2^4 - 18\mathbf{A}_2^2\mathbf{A}_3 \\
& + 6\mathbf{A}_2\mathbf{A}_4 + 6\mathbf{A}_3^2 + 4\mathbf{A}_4\mathbf{A}_2\Big)\mathbf{e}_0^5 + \Big(47\mathbf{A}_2^2\mathbf{A}_3\mathbf{A}_2 + 43\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2^2 + 38\mathbf{A}_3\mathbf{A}_2^3 - 19\mathbf{A}_2\mathbf{A}_4\mathbf{A}_2 \\
& - 18\mathbf{A}_3^2\mathbf{A}_2 - 16\mathbf{A}_4\mathbf{A}_2^2 - 28\mathbf{A}_2\mathbf{A}_3^2 - 24\mathbf{A}_3\mathbf{A}_2\mathbf{A}_3 - 27\mathbf{A}_2^2\mathbf{A}_4 - 88\mathbf{A}_2^5 + 60\mathbf{A}_2^3\mathbf{A}_3 + 8\mathbf{A}_2\mathbf{A}_5
\end{aligned}$$

$$
\begin{aligned}
&+ 9\mathbf{A}_3\mathbf{A}_4 + 8\mathbf{A}_4\mathbf{A}_3 + 5\mathbf{A}_5\mathbf{A}_2\Big)\mathbf{e}_0^6 + \cdots + O\Big(\mathbf{e}_0^8\Big),\\
\mathbf{F}(\mathbf{y}_2) =& \mathbf{A}_1\Big(1 + 4\mathbf{A}_2^3\mathbf{e}_0^3 + \Big(6\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2 + 8\mathbf{A}_2^2\mathbf{A}_3 - 18\mathbf{A}_2^4\Big)\mathbf{e}_0^4 + \Big(8\mathbf{A}_2\mathbf{A}_4\mathbf{A}_2 + 12\mathbf{A}_2\mathbf{A}_3^2 - 24\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2^2\\
&- 28\mathbf{A}_2^2\mathbf{A}_3\mathbf{A}_2 + 12\mathbf{A}_2^2\mathbf{A}_4 + 60\mathbf{A}_2^5 - 36\mathbf{A}_2^3\mathbf{A}_3\Big)\mathbf{e}_0^5 + \Big(94\mathbf{A}_2^3\mathbf{A}_3\mathbf{A}_2 + 86\mathbf{A}_2^2\mathbf{A}_3\mathbf{A}_2^2 + 76\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2^3\\
&- 38\mathbf{A}_2^2\mathbf{A}_4\mathbf{A}_2 - 36\mathbf{A}_2\mathbf{A}_3^2\mathbf{A}_2 - 32\mathbf{A}_2\mathbf{A}_4\mathbf{A}_2^2 - 56\mathbf{A}_2^2\mathbf{A}_3^2 - 48\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2\mathbf{A}_3 - 54\mathbf{A}_2^3\mathbf{A}_4\\
&- 176\mathbf{A}_2^6 + 120\mathbf{A}_2^4\mathbf{A}_3 + 16\mathbf{A}_2^2\mathbf{A}_5 + 18\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4 + 16\mathbf{A}_2\mathbf{A}_4\mathbf{A}_3 + 10\mathbf{A}_2\mathbf{A}_5\mathbf{A}_2 + 12\mathbf{A}_3\mathbf{A}_2^4\Big)\mathbf{e}_0^6\\
&+ \cdots + O\Big(\mathbf{e}_0^8\Big)\Big),\\
\boldsymbol{\phi}_3 =& 2\mathbf{A}_2^2\mathbf{e}_0^3 + \Big(-13\mathbf{A}_2^3 + 4\mathbf{A}_2\mathbf{A}_3 + 3\mathbf{A}_3\mathbf{A}_2\Big)\mathbf{e}_0^4 + \Big(-18\mathbf{A}_3\mathbf{A}_2^2 - 20\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2 + 56\mathbf{A}_2^4 - 26\mathbf{A}_2^2\mathbf{A}_3\\
&+ 6\mathbf{A}_2\mathbf{A}_4 + 6\mathbf{A}_3^2 + 4\mathbf{A}_4\mathbf{A}_2\Big)\mathbf{e}_0^5 + \Big(87\mathbf{A}_2^2\mathbf{A}_3\mathbf{A}_2 + 79\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2^2 + 77\mathbf{A}_3\mathbf{A}_2^3 - 27\mathbf{A}_2\mathbf{A}_4\mathbf{A}_2\\
&- 27\mathbf{A}_3^2\mathbf{A}_2 - 24\mathbf{A}_4\mathbf{A}_2^2 - 40\mathbf{A}_2\mathbf{A}_3^2 - 36\mathbf{A}_3\mathbf{A}_2\mathbf{A}_3 - 39\mathbf{A}_2^2\mathbf{A}_4 - 196\mathbf{A}_2^5 + 112\mathbf{A}_2^3\mathbf{A}_3 + 8\mathbf{A}_2\mathbf{A}_5\\
&+ 9\mathbf{A}_3\mathbf{A}_4 + 8\mathbf{A}_4\mathbf{A}_3 + 5\mathbf{A}_5\mathbf{A}_2\Big)\mathbf{e}_0^6 + \cdots + O\Big(\mathbf{e}_0^8\Big),\\
\boldsymbol{\phi}_4 =& 2\mathbf{A}_2^2\mathbf{e}_0^3 + \Big(-17\mathbf{A}_2^3 + 4\mathbf{A}_2\mathbf{A}_3 + 3\mathbf{A}_3\mathbf{A}_2\Big)\mathbf{e}_0^4 + \Big(-24\mathbf{A}_3\mathbf{A}_2^2 - 26\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2 + 90\mathbf{A}_2^4 - 34\mathbf{A}_2^2\mathbf{A}_3\\
&+ 6\mathbf{A}_2\mathbf{A}_4 + 6\mathbf{A}_3^2 + 4\mathbf{A}_4\mathbf{A}_2\Big)\mathbf{e}_0^5 + \Big(139\mathbf{A}_2^2\mathbf{A}_3\mathbf{A}_2 + 127\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2^2 + 128\mathbf{A}_3\mathbf{A}_2^3 - 35\mathbf{A}_2\mathbf{A}_4\mathbf{A}_2\\
&- 36\mathbf{A}_3^2\mathbf{A}_2 - 32\mathbf{A}_4\mathbf{A}_2^2 - 52\mathbf{A}_2\mathbf{A}_3^2 - 48\mathbf{A}_3\mathbf{A}_2\mathbf{A}_3 - 51\mathbf{A}_2^2\mathbf{A}_4 - 368\mathbf{A}_2^5 + 180\mathbf{A}_2^3\mathbf{A}_3 + 8\mathbf{A}_2\mathbf{A}_5\\
&+ 9\mathbf{A}_3\mathbf{A}_4 + 8\mathbf{A}_4\mathbf{A}_3 + 5\mathbf{A}_5\mathbf{A}_2\Big)\mathbf{e}_0^6 + \cdots + O\Big(\mathbf{e}_0^8\Big),\\
\boldsymbol{\phi}_5 =& 2\mathbf{A}_2^2\mathbf{e}_0^3 + \Big(-21\mathbf{A}_2^3 + 4\mathbf{A}_2\mathbf{A}_3 + 3\mathbf{A}_3\mathbf{A}_2\Big)\mathbf{e}_0^4 + \Big(-30\mathbf{A}_3\mathbf{A}_2^2 + 132\mathbf{A}_2^4 - 42\mathbf{A}_2^2\mathbf{A}_3 - 32\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2\\
&+ 6\mathbf{A}_2\mathbf{A}_4 + 6\mathbf{A}_3^2 + 4\mathbf{A}_4\mathbf{A}_2\Big)\mathbf{e}_0^5 + \Big(203\mathbf{A}_2^2\mathbf{A}_3\mathbf{A}_2 + 187\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2^2 + 191\mathbf{A}_3\mathbf{A}_2^3 - 43\mathbf{A}_2\mathbf{A}_4\mathbf{A}_2\\
&- 45\mathbf{A}_3^2\mathbf{A}_2 - 40\mathbf{A}_4\mathbf{A}_2^2 - 64\mathbf{A}_2\mathbf{A}_3^2 - 60\mathbf{A}_3\mathbf{A}_2\mathbf{A}_3 - 63\mathbf{A}_2^2\mathbf{A}_4 - 624\mathbf{A}_2^5 + 264\mathbf{A}_2^3\mathbf{A}_3 + 8\mathbf{A}_2\mathbf{A}_5\\
&+ 9\mathbf{A}_3\mathbf{A}_4 + 8\mathbf{A}_4\mathbf{A}_3 + 5\mathbf{A}_5\mathbf{A}_2\Big)\mathbf{e}_0^6 + \cdots + O\Big(\mathbf{e}_0^8\Big),\\
\boldsymbol{\phi}_6 =& 2\mathbf{A}_2^2\mathbf{e}_0^3 + \Big(-25\mathbf{A}_2^3 + 4\mathbf{A}_2\mathbf{A}_3 + 3\mathbf{A}_3\mathbf{A}_2\Big)\mathbf{e}_0^4 + \Big(-36\mathbf{A}_3\mathbf{A}_2^2 + 182\mathbf{A}_2^4 - 50\mathbf{A}_2^2\mathbf{A}_3 - 38\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2\\
&+ 6\mathbf{A}_2\mathbf{A}_4 + 6\mathbf{A}_3^2 + 4\mathbf{A}_4\mathbf{A}_2\Big)\mathbf{e}_0^5 + \Big(279\mathbf{A}_2^2\mathbf{A}_3\mathbf{A}_2 + 259\mathbf{A}_2\mathbf{A}_3\mathbf{A}_2^2 + 266\mathbf{A}_3\mathbf{A}_2^3 - 51\mathbf{A}_2\mathbf{A}_4\mathbf{A}_2\\
&- 54\mathbf{A}_3^2\mathbf{A}_2 - 48\mathbf{A}_4\mathbf{A}_2^2 - 76\mathbf{A}_2\mathbf{A}_3^2 - 72\mathbf{A}_3\mathbf{A}_2\mathbf{A}_3 - 75\mathbf{A}_2^2\mathbf{A}_4 - 980\mathbf{A}_2^5 + 364\mathbf{A}_2^3\mathbf{A}_3 + 8\mathbf{A}_2\mathbf{A}_5\\
&+ 9\mathbf{A}_3\mathbf{A}_4 + 8\mathbf{A}_4\mathbf{A}_3 + 5\mathbf{A}_5\mathbf{A}_2\Big)\mathbf{e}_0^6 + \cdots + O\Big(\mathbf{e}_0^8\Big),\\
\mathbf{e}_3 =& \Big(2\mathbf{A}_2^2\mathbf{A}_3\mathbf{A}_2^2 - 6\mathbf{A}_3\mathbf{A}_2^4 + 46\mathbf{A}_2^6\Big)\mathbf{e}_0^7 + O\Big(\mathbf{e}_0^8\Big).
\end{aligned}
$$

□

Now we present the proof of convergence of MFAA via the mathematical induction.

Theorem 3.2. *The convergence order of MFAA method is $4m-5$ for $m \geq 2$.*

Proof. All the computations are made under the assumption of Theorem 3.1. We know from Theorem 3.1 that the convergence order of MFAA method is seven for $m = 3$. By performing the computations in a similar manner we get the following error equation for $m = 4$

$$\mathbf{e}_4 = \Big(30\mathbf{A}_2^4 - 6\mathbf{A}_3\mathbf{A}_2^2 + 2\mathbf{A}_2^2\mathbf{A}_3\Big)\Big(2\mathbf{A}_2^2\mathbf{A}_3\mathbf{A}_2^2 - 6\mathbf{A}_3\mathbf{A}_2^4 + 46\mathbf{A}_2^6\Big)\mathbf{e}_0^{11} + O\left(\mathbf{e}_0^{12}\right) .$$

Now we assume that the convergence order of MFAA is $4s-5$ for $s \geq 4$, and we will prove that the convergence order of MFAA is $4s-1$ for $(s+1)$-th step. If the convergence order of MFAA is $4s-5$ then

$$\mathbf{e}_s = \mathbf{y}_s - \mathbf{y}^* \sim d_1 \mathbf{e}_0^{4s-5}, \tag{3.1}$$

where d_1 is the asymptotic constant and symbol $\sim$ means the approximation. By using (3.1), we perform the following steps to complete the proof.

$$\begin{aligned}
\mathbf{F}(\mathbf{y}_0)^{-1} &\sim (\mathbf{I} - 2\mathbf{A}_2\mathbf{e}_0)\,\mathbf{A}_1^{-1}, \\
\mathbf{F}(\mathbf{y}_s) &\sim \mathbf{A}_1 d_1 \mathbf{e}_0^{4s-5}, \\
\boldsymbol{\phi}_7 &\sim (\mathbf{I} - 2\mathbf{e}_0\mathbf{A}_2) d_1 \mathbf{e}_0^{4s-5}, \\
\mathbf{F}'(\mathbf{y}_2) &\sim \mathbf{A}_1\left(\mathbf{I} + 4\mathbf{A}_2^3\mathbf{e}_0^3\right), \\
\boldsymbol{\phi}_8 &\sim -16\mathbf{A}_2^4 d_1\mathbf{e}_0^{4s-1} + 4\mathbf{A}_2^2 d_1\mathbf{e}_0^{4s-3} + 4\mathbf{A}_2^3 d_1\mathbf{e}_0^{4s-2} + 16\mathbf{A}_2^5 d_1\mathbf{e}_0^{4s} - 4\mathbf{A}_2 d_1\mathbf{e}_0^{4s-4} + d_1\mathbf{e}_0^{4s-5}, \\
\boldsymbol{\phi}_9 &\sim -128\mathbf{A}_2^9 d_1\mathbf{e}_0^{4s+4} + 192\mathbf{A}_2^8 d_1\mathbf{e}_0^{4s+3} - 96\mathbf{A}_2^7 d_1\mathbf{e}_0^{4s+2} - 48\mathbf{A}_2^6 d_1\mathbf{e}_0^{4s+1} - 48\mathbf{A}_2^4 d_1\mathbf{e}_0^{4s-1} + 12\mathbf{A}_2^2 d_1\mathbf{e}_0^{4s-3} \\
&\quad + 96\mathbf{A}_2^5 d_1\mathbf{e}_0^{4s} - 6\mathbf{A}_2 d_1\mathbf{e}_0^{4s-4} + d_1\mathbf{e}_0^{4s-5}, \\
\boldsymbol{\phi}_{10} &\sim 1024\mathbf{A}_2^{13}\mathbf{e}_0^{4s+8} d_1 - 2048\mathbf{A}_2^{12} d_1\mathbf{e}_0^{4s+7} + 1536\mathbf{A}_2^{11} d_1\mathbf{e}_0^{4s+6} + 256\mathbf{A}_2^{10} d_1\mathbf{e}_0^{4s+5} - 1472\mathbf{A}_2^9 d_1\mathbf{e}_0^{4s+4} \\
&\quad + 1152\mathbf{A}_2^8 d_1\mathbf{e}_0^{4s+3} - 192\mathbf{A}_2^7 d_1\mathbf{e}_0^{4s+2} - 336\mathbf{A}_2^6 d_1\mathbf{e}_0^{4s+1} - 80\mathbf{A}_2^4 d_1\mathbf{e}_0^{4s-1} - 20\mathbf{A}_2^3 d_1\mathbf{e}_0^{4s-2} \\
&\quad + 24\mathbf{A}_2^2 d_1\mathbf{e}_0^{4s-3} + 288\mathbf{A}_2^5 d_1\mathbf{e}_0^{4s} - 8\mathbf{A}_2 d_1\mathbf{e}_0^{4s-4} + d_1\mathbf{e}_0^{4s-5}, \\
\mathbf{e}_{s+1} &\sim 1536\mathbf{A}_2^{13} d_1\mathbf{e}_0^{4s+8} - 3072\mathbf{A}_2^{12} d_1\mathbf{e}_0^{4s+7} + 2304\mathbf{A}_2^{11} d_1\mathbf{e}_0^{4s+6} - 1504\mathbf{A}_2^9 d_1\mathbf{e}_0^{4s+4} + 384\mathbf{A}_2^{10} d_1\mathbf{e}_0^{4s+5} \\
&\quad + 672\mathbf{A}_2^8 d_1\mathbf{e}_0^{4s+3} + 240\mathbf{A}_2^7 d_1\mathbf{e}_0^{4s+2} - 240\mathbf{A}_2^6 d_1\mathbf{e}_0^{4s+1} + 24\mathbf{A}_2^4 d_1\mathbf{e}_0^{4s-1} + 24\mathbf{A}_2^5 d_1\mathbf{e}_0^{4s}, \\
\mathbf{e}_{s+1} &\sim 24\mathbf{A}_2^4 d_1\mathbf{e}_0^{4s-1},
\end{aligned}$$

which completes the proof. □

4. Numerical testing

In this section, we will verify the claimed convergence order and solve some famous nonlinear initial and boundary value problems. We adopt the following definition of computational convergence order (COC)

$$\text{COC} = \frac{\log\left(||\mathbf{F}(\mathbf{x}_{k+1})||_\infty / ||\mathbf{F}(\mathbf{x}_k)||_\infty\right)}{\log\left(||\mathbf{F}(\mathbf{x}_k)||_\infty / ||\mathbf{F}(\mathbf{x}_{k-1})||_\infty\right)}.$$

4.1. Verification of computational convergence order

Consider the following system of nonlinear equations $\mathbf{F}(\mathbf{x}) = [F_1(\mathbf{x}), F_2(\mathbf{x}), F_3(\mathbf{x}), F_4(\mathbf{x})]^T = \mathbf{0}$,

$$\begin{aligned}
F_1(\mathbf{x}) &= x_2\, x_3 + x_4\,(x_2 + x_3) = 0, \\
F_2(\mathbf{x}) &= x_1\, x_3 + x_4\,(x_1 + x_3) = 0, \\
F_3(\mathbf{x}) &= x_1\, x_2 + x_4\,(x_1 + x_2) = 0, \\
F_4(\mathbf{x}) &= x_1\, x_2 + x_3\,(x_1 + x_2) = 1.
\end{aligned} \tag{4.1}$$

Table 5 shows that the computational convergence orders are according to theoretical convergence order of the iterative method MFAA. Next we consider the Lane Emden boundary value problem

$$u''(x) + \frac{2}{x}\,u'(x) + u(x)^p = 0\,, \quad u'(0) = 0\,, \quad u(0) = 1. \tag{4.2}$$

In Table 6, we fix the number of steps and solve the Lane-Emden problem. We find that computational order of convergence of the iterative method MFAA agrees with theoretical convergence order. In Figure 1, we plotted the numerical solution of the Lane-Emden equation for different indices ranging from two to five. It is noticeable that it is not always possible to confirm the convergence order by solving boundary and initial value problems.

Table 5: MFAA : verification of convergence order for the problem (4.1).

Iter \ Steps		$m=2$	$m=3$	$m=4$	$m=5$
1	$\lVert\mathbf{F}(\mathbf{x}_k)\rVert_\infty$	3.33e-3	2.46e-6	2.31e-9	2.16e-12
2	-	4.63e-9	6.19e-42	1.31e-98	1.64e-179
3	-	1.24e-26	3.92e-291	2.58e-1080	2.46e-2686
COC		3	7	11	15
Theoretical convergence order $(4m-5)$		3	7	11	15

Table 6: MFAA : verification of convergence order for the problem (4.2) over the domain [0,3], number of grid points 50.

Iter \ Index(p)		2	3	4	5
1	$\lVert\mathbf{F}(\mathbf{x}_k)\rVert_\infty$	1.02	9.91e-1	1.18	5.86e-1
2	-	5.76e-16	7.59e-13	2.00e-11	1.62e-10
3	-	9.93e-123	3.20e-100	4.75e-90	2.64e-83
COC (m=3)		7.00	7.21	7.30	7.30
Theoretical convergence order $(4m-5)$		7	7	7	7

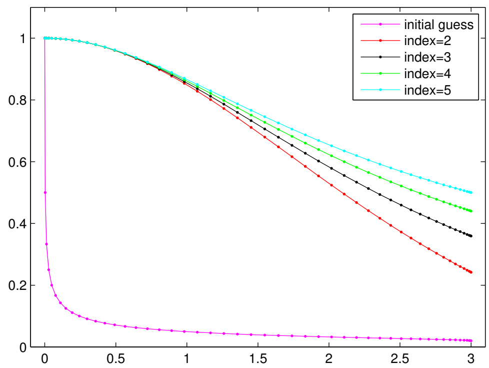

Figure 1: MFAA: plot of Lane Emden equation with different indices, number of gird points 50.

4.2. 3-D nonlinear Poisson problem

We study the following nonlinear Poisson Dirichlet boundary value problem

$$u_{xx} + u_{yy} + u_{zz} + f(u) = p(x, y, z), \qquad (x, y, z) \in (0, 1)^3, \tag{4.3}$$

where $p(x, y, z)$ is the source term and $f(u) = u^s$ is a nonlinear function. Using Chebyshev pseudo-spectral collocation method [4, 8, 15], we performed the following discretization of (4.3)

$$\begin{aligned} \mathbf{F}(\mathbf{U}) &= \big((\mathbf{T}_{xx} \otimes \mathbf{I}_y \otimes \mathbf{I}_z) + (\mathbf{I}_x \otimes \mathbf{T}_{yy} \otimes \mathbf{I}_z) + (\mathbf{I}_x \otimes \mathbf{I}_y \otimes \mathbf{T}_{zz})\big)\mathbf{U} + f(\mathbf{U}) - \mathbf{p} = \mathbf{0}, \\ \mathbf{F}'(\mathbf{U}) &= \big((\mathbf{T}_{xx} \otimes \mathbf{I}_y \otimes \mathbf{I}_z) + (\mathbf{I}_x \otimes \mathbf{T}_{yy} \otimes \mathbf{I}_z) + (\mathbf{I}_x \otimes \mathbf{I}_y \otimes \mathbf{T}_{zz})\big) + \mathrm{diag}\big(f'(\mathbf{U})\big), \end{aligned} \tag{4.4}$$

where $\mathbf{I}$ denotes the identity matrix, and $\otimes$ is a Kronecker product and $\mathbf{T}_{..}$ is the discretization of second order derivative. In Tables 7 and 8, we show the error in the numerical solution of the problem (4.3) for different nonlinear terms against the various grid sizes. We achieved almost 15-digit accuracy in the computed solution. It is important to note that we perform only one iteration and multi-steps. It means we compute only once the LU factors of the Jacobian at the initial guess and use these LU factors repeatedly in the multi-step part to solve the system of linear equations to achieve the high order of convergence.

Table 7: MFAA: absolute error in the solution of (4.4) versus different grid sizes, number of iterations = 1, initial guess $\mathbf{U} = \mathbf{0}$, $f(u) = u^3$.

m \ N		$8 \times 8 \times 8$	$10 \times 10 \times 10$	$12 \times 12 \times 12$
2	$\|\|\mathbf{U}_k - \mathbf{U}_{\text{analytical}}\|\|_\infty$	4.28e-03	4.55e-03	4.69e-03
3	-	1.10e-06	1.22e-06	1.28e-06
4	-	9.30e-10	5.40e-10	5.75e-10
5	-	5.62e-10	3.20e-13	2.48e-13
6	-	5.62e-10	3.22e-13	9.55e-15

Table 8: MFAA: absolute error in the solution of (4.4) versus different grid sizes, number of iterations = 1, initial guess $\mathbf{U} = \mathbf{0}$, $f(u) = u^4$.

m \ N		$8 \times 8 \times 8$	$10 \times 10 \times 10$	$12 \times 12 \times 12$
2	$\|\|\mathbf{U}_k - \mathbf{U}_{\text{analytical}}\|\|_\infty$	5.22e-03	5.54e-03	5.71e-03
3	-	3.70e-06	4.07e-06	4.27e-06
4	-	4.85e-09	4.92e-09	5.20e-09
5	-	5.63e-10	5.74e-12	6.19e-12
6	-	5.63e-10	3.24e-13	7.44e-15

4.3. 2-D nonlinear wave equation

The 2-D nonlinear wave equation can be written as

$$u_{tt} - c^2 (u_{xx} + u_{yy}) + f(u) = p(x, y), \qquad (x, y, t) \in (-1, 1)^2 \times (0, 2], \tag{4.5}$$

where nonlinear function $f(u) = u^s$ and c, s are constants. By assuming the solution $u = \exp(-t)\sin(x+y)$, we compute source term $p(x, y)$. The 2-D nonlinear wave equation is solved by imposing Dirichlet boundary conditions. By the application of Chebyshev pseudo-spectral collocation method, we discretize (4.5) and get the following system of nonlinear equations

$$\begin{aligned} \mathbf{F}(\mathbf{U}) &= \big(\big(\mathbf{T}_{tt} \otimes \mathbf{I}_x \otimes \mathbf{I}_y\big) - c^2\big(\big(\mathbf{I}_t \otimes \mathbf{T}_{xx} \otimes \mathbf{I}_y\big) + \big(\mathbf{I}_t \otimes \mathbf{I}_x \otimes \mathbf{T}_{yy}\big)\big)\big)\mathbf{U} + f(\mathbf{U}) - \mathbf{p} = \mathbf{0}, \\ \mathbf{F}'(\mathbf{U}) &= \big(\mathbf{T}_{tt} \otimes \mathbf{I}_x \otimes \mathbf{I}_y\big) - c^2\big(\big(\mathbf{I}_t \otimes \mathbf{T}_{xx} \otimes \mathbf{I}_y\big) + \big(\mathbf{I}_t \otimes \mathbf{I}_x \otimes \mathbf{T}_{yy}\big)\big) + \mathrm{diag}\big(f'(\mathbf{U})\big). \end{aligned} \tag{4.6}$$

The 2-D nonlinear wave equation is solved over three different size grids. Table 9 depicts that we achieved 11-digit accuracy by performing a single iteration of our proposed iterative method MFAA. In all tables, we stop to perform further step when we see that there is not the reduction in the norm of the error of the numerical solution.

Table 9: MFAA: absolute error in the solution of (4.6) versus different grid sizes, number of iterations = 1, $f(u) = u^3$, $c = 1$, initial guess $\mathbf{U} = \mathbf{0}$.

m \ N		$10 \times 10 \times 20$	$20 \times 20 \times 20$	$20 \times 20 \times 30$
2	$\|\|\mathbf{U}_k - \mathbf{U}_{\text{analytical}}\|\|_\infty$	3.85e-03	3.80e-03	3.69e-03
3	-	1.13e-06	1.59e-04	1.90e-05
4	-	5.71e-10	5.85e-06	1.45e-06
5	-	6.78e-10	2.13e-07	1.14e-07
6	-	6.78e-10	7.69e-09	8.93e-09
7	-	6.78e-10	2.81e-10	7.01e-10
8	-	6.78e-10	5.38e-11	5.43e-11

4.4. Stiff nonlinear boundary value problem

The iterative method FTUC has comparable efficiency with the iterative method MFAA. For the purpose of comparison, we solve a stiff nonlinear boundary value problem which is written as

$$\lambda z''(t) + f_1(z(t))z'(t) - \frac{\pi}{2}\sin\left(\frac{\pi}{2}t\right) f_2(z) = 0, \quad t \in [0,1], \tag{4.7}$$

where $f_1(z) = \exp(z)$, $f_2(z) = \exp(2z)$, $\lambda = 1e-3$, $z(0) = 1$, and $z(1) = 0$. For the discretization of this problem we use finite difference method. The discrete form of (4.7) is

$$\mathbf{F}(\mathbf{Z}) = \lambda\left(\mathbf{D}_2\mathbf{Z}\right) + f_1(\mathbf{Z}) \odot \left(\mathbf{D}_1\mathbf{Z}\right) - \frac{\pi}{2}\left(\sin\left(\frac{\pi}{2}t\right)\right) \odot f_2(\mathbf{Z}) = \mathbf{0},$$
$$\mathbf{F}'(\mathbf{Z}) = \lambda\mathbf{D}_2 + \mathrm{diag}\left(f_1'(\mathbf{Z}) \odot (\mathbf{D}_1\mathbf{Z})\right) + \mathrm{diag}\left(f_1(\mathbf{Z})\right)\mathbf{D}_1 - \mathrm{diag}\left(\frac{\pi}{2}\left(\sin\left(\frac{\pi}{2}t\right)\right) \odot f_2'(\mathbf{Z})\right),$$

where $\odot$ stands for the point-wise multiplication and diag for the diagonal matrix. To make the nonlinear function expensive, we adopt the following Matlab code

```
big=1e7;
f1=@(z) exp(z)+0*sum(rand(1,big));
f2=@(z) exp(2*z)+0*sum(rand(1,big));
df1=@(z) exp(z)+0*sum(rand(1,big));
df2=@(z) 2*exp(2*z)+0*sum(rand(1,big));
```

We want to show that if the evaluation of the nonlinear function is expensive, then a method that has fast convergence will take less simulation compare to slow convergence method. The convergence order of the iterative method MFAA is higher than that of FTUC. We choose 500 as the size of the associated system of nonlinear equations in the stiff nonlinear boundary value problem. Both methods use a single iteration to achieve 13-digit accuracy in the infinity norm of $\mathbf{F}(\cdot)$. Table 10 shows the computational cost of different operations and sequences of error on steps for both methods. Few successive approximations of the solution for the considered stiff nonlinear boundary value problem are plotted in Figure 2. The simulation time of the iterative method MFAA is less because it has the higher order of convergence relatively and the evaluation of the nonlinear function is expensive. We made the nonlinear function computationally costly to show the effectiveness of our proposed iterative method.

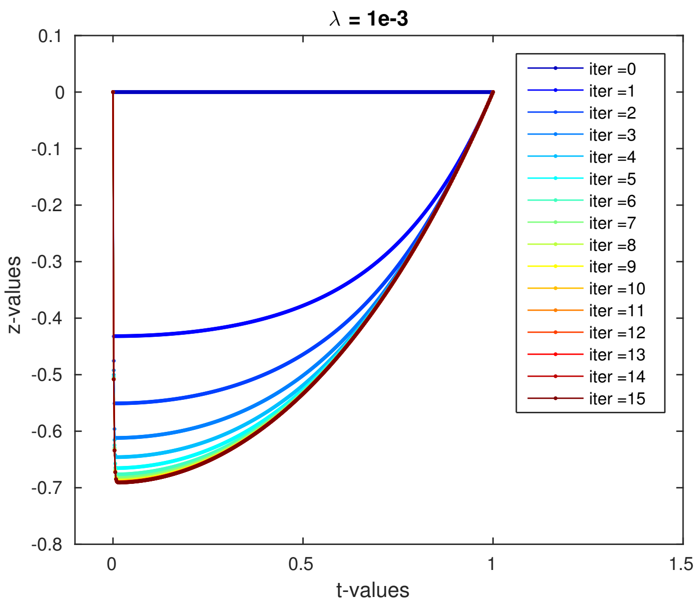

Figure 2: Successive iteration for stiff nonlinear boundary value problem, $\lambda = 1e-3$.

Table 10: Stiff nonlinear boundary value problem: comparison of performances of MFAA and FTUC iterative methods, initial guess $\mathbf{Z} = \mathbf{0}$, $\lambda = 1e-3$, $n = 500$, $x \in [0,1]$.

Iterative methods		FTUC	MFAA
Problem size		500	500
Number of iterations		1	1
Number of steps		24	18
Theoretical convergence-order		68	67
Number of function evaluations per iteration		23	18
Solutions of system of linear equations per iteration		46	66
Number of Jacobian evaluations per iteration		2	2
Number of Jacobian LU-factorizations per iteration		1	1
Number of vector-vector multiplications per iteration		25	64
Number of matrix-vector multiplications per iteration		23	48
	Steps(iter=1)		
$\|\|\mathbf{F}(\mathbf{y}_k)\|\|_\infty$	4	$1.00e+00$	$1.78e-01$
	5	$2.78e-01$	$2.31e-02$
	7	$1.69e-02$	$4.08e-04$
	9	$1.03e-03$	$7.47e-06$
	11	$6.25e-05$	$1.40e-07$
	12	$1.53e-05$	$1.92e-08$
	13	$3.71e-06$	$2.64e-09$
	14	$8.98e-07$	$3.63e-10$
	15	$2.17e-07$	$5.02e-11$
	16	$5.22e-08$	$6.88e-12$
	17	$1.25e-08$	$1.01e-12$
	18	$3.00e-09$	$1.97e-13$
	19	$7.18e-10$	
	20	$1.72e-10$	
	21	$4.10e-11$	
	22	$9.83e-12$	
	23	$2.29e-12$	
	24	$5.35e-13$	
Simulation time (sec)		5.05	4.139

5. Conclusions

We solved four nonlinear problems to show the efficiency and validity of our proposed iterative method. The numerical results show that the proposed iterative method is an efficient iterative method for solving system of nonlinear equations. The soul of the proposed iterative method is hidden in the idea of frozen Jacobian. To make the frozen Jacobian computationally efficient, we add multi-steps in a way that we use LU factor information from the base method and perform a single evaluation of the function per multi-step to achieve a high order of convergence. The per step increment of our proposed iterative method is four and what we pay is the solution of four lower and upper triangular system of equation with a single evaluation of the system of nonlinear equations. In general, it is hard to verify the convergence order for all boundary value problem that is why we did not show COC for the problems (4.3) and (4.5).

Acknowledgment

The authors are thankful to reviewers for their suggestions to enhance the quality of the article.

References

[1] F. Ahmad, E. Tohidi, J. A. Carrasco, *A parameterized multi-step Newton method for solving systems of nonlinear equations*, Numer. Algorithms, **71** (2016), 631–653. 1

[2] F. Ahmad, E. Tohidi, M. Z. Ullah, J. A. Carrasco, *Higher order multi-step Jarratt-like method for solving systems of nonlinear equations: application to PDEs and ODEs*, Comput. Math. Appl., **70** (2015), 624–636. 1

[3] E. S. Alaidarous, M. Z. Ullah, F. Ahmad, A. S. Al-Fhaid, *An efficient higher-order quasilinearization method for solving nonlinear BVPs*, J. Appl. Math., **2013** (2013), 11 pages. 1

[4] A. H. Bhrawy, *An efficient Jacobi pseudospectral approximation for nonlinear complex generalized Zakharov system*, Appl. Math. Comput., **247** (2014), 30–46. 4.2

[5] W. H. Bi, H. M. Ren, Q. B. Wu, *Three-step iterative methods with eighth-order convergence for solving nonlinear equations*, J. Comput. Appl. Math., **225** (2009), 105–112. 1

[6] A. Cordero, J. L. Hueso, E. Martínez, J. R. Torregrosa, *A modified Newton-Jarratt's composition*, Numer. Algorithms, **55** (2010), 87–99. 1

[7] M. Davies, B. Dawson, *On the global convergence of Halley's iteration formula*, Numer. Math., **24** (1975), 133–135. 1

[8] M. Dehghan, F. Fakhar-Izadi, *The spectral collocation method with three different bases for solving a nonlinear partial differential equation arising in modeling of nonlinear waves*, Math. Comput. Modelling, **53** (2011), 1865–1877. 4.2

[9] Y. H. Geum, Y. I. Kim, *A multi-parameter family of three-step eighth-order iterative methods locating a simple root*, Appl. Math. Comput., **215** (2010), 3375–3382. 1

[10] E. Halley, *A new exact and easy method of finding the roots of equations generally and without any previous reduction*, Philos. Trans. Roy. Soc. London, **18** (1964), 136–148. 1

[11] H. T. Kung, J. F. Traub, *Optimal order of one-point and multipoint iteration*, J. Assoc. Comput. Mach., **21** (1974), 643–651. 1

[12] H. Montazeri, F. Soleymani, S. Shateyi, S. S. Motsa, *On a new method for computing the numerical solution of systems of nonlinear equations*, J. Appl. Math., **2012** (2012), 15 pages. 1

[13] F. Soleymani, *On a new class of optimal eighth-order derivative-free methods*, Acta Univ. Sapientiae Math., **3** (2011), 169–180. 1

[14] F. Soleymani, T. Lotfi, P. Bakhtiari, *A multi-step class of iterative methods for nonlinear systems*, Optim. Lett., **8** (2014), 1001–1015. 1

[15] E. Tohidi, S. Lotfi Noghabi, *An efficient Legendre pseudospectral method for solving nonlinear quasi bang-bang optimal control problems*, J. Appl. Math. Stat. Inform., **8** (2012), 73–85. 4.2

[16] M. Z. Ullah, A. S. Al-Fhaid, F. Ahmad, *Four-point optimal sixteenth-order iterative method for solving nonlinear equations*, J. Appl. Math., **2013** (2013), 5 pages. 1

[17] M. Z. Ullah, S. Serra-Capizzano, F. Ahmad, *An efficient multi-step iterative method for computing the numerical solution of systems of nonlinear equations associated with ODEs*, Appl. Math. Comput., **250** (2015), 249–259. 1

[18] M. Z. Ullah, F. Soleymani, A. S. Al-Fhaid, *Numerical solution of nonlinear systems by a general class of iterative methods with application to nonlinear PDEs*, Numer. Algorithms, **67** (2014), 223–242. 1

[19] X. Wang, L. P. Liu, *Modified Ostrowski's method with eighth-order convergence and high efficiency index*, Appl. Math. Lett., **23** (2010), 549–554. 1