

# Multi-step derivative-free preconditioned Newton method for solving systems of nonlinear equations

Fayyaz Ahmad

**Abstract** Preconditioning of systems of nonlinear equations modifies the associated Jacobian and provides rapid convergence. The preconditioners are introduced in a way that they do not affect the convergence order of parent iterative method. The multi-step derivative-free iterative method consists of a base method and multi-step part. In the base method, the Jacobian of the system of nonlinear equation is approximated by finite difference operator and preconditioners add an extra term to modify it. The inversion of modified finite difference operator is avoided by computing LU factors. Once we have LU factors, we repeatedly use them to solve lower and upper triangular systems in the multi-step part to enhance the convergence order. The convergence order of  $m$ -step Newton iterative method is  $m + 1$ . The claimed convergence orders are verified by computing the computational order of convergence and numerical simulations clearly show that the good selection of preconditioning provides numerical stability, accuracy and rapid convergence.

**Keywords** Systems of nonlinear equations · Nonlinear preconditioners · Multi-step iterative methods · Derivative-free

**Mathematics Subject Classification** 65H10 · 65L10 · 65L05 · 65.3 · 65.65 · 80M25 · 80.65

## 1 Introduction

Let  $\mathbf{F} : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}^n$  be a nonlinear function and the system of nonlinear equations can be written as

$$\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x})]^T = \mathbf{0}, \quad (1)$$

where  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ . If  $\mathbf{x}^*$  is a simple root of (1) then  $\mathbf{F}(\mathbf{x}^*) = \mathbf{0}$  and  $\det(\mathbf{F}'(\mathbf{x}^*)) \neq 0$  i.e. Jacobian should not be singular at the root. Newton method [1–4] is the classical iterative method for computing the simple root of system of nonlinear equation. The multi-step Newton method [5] can be written as

$$\begin{aligned} \text{Base method} &\longrightarrow \begin{cases} \mathbf{x}_0 = \text{initial guess} \\ \mathbf{F}'(\mathbf{x}_0) \boldsymbol{\phi}_1 = \mathbf{F}(\mathbf{x}_0) \\ \mathbf{x}_1 = \mathbf{x}_0 - \boldsymbol{\phi}_1 \end{cases} \\ \text{Multi-step part} &\longrightarrow \begin{cases} \text{for } j = 2, m \\ \quad \mathbf{F}'(\mathbf{x}_0) \boldsymbol{\phi}_j = \mathbf{F}(\mathbf{x}_{j-1}) \\ \quad \mathbf{x}_j = \mathbf{x}_{j-1} - \boldsymbol{\phi}_j \\ \text{end} \\ \mathbf{x}_0 = \mathbf{x}_m \end{cases} \end{aligned} \quad (2)$$

and its order of convergence is  $m + 1$ . Many researchers [6–9] have proposed higher order multi-step iterative method for solving system of nonlinear equations. In most of real world problems, the closed form expression for the system of nonlinear equations is not always possible. When we get the information about the system of nonlinear equations from a black box then the computation of Jacobian analytically is no way possible. So it means, we need to compute it numerically. Recently people have proposed derivative-free iterative method [10–14] for the solution of the system of nonlinear equations. Grau-Sánchez et al. [12] have constructed the following multi-step derivative-free iterative method for solving the system of nonlinear equations

$$\begin{aligned} \text{Basemethod} &\longrightarrow \begin{cases} \mathbf{x}_0 = \text{initial guess} \\ \mathbf{u} = \mathbf{x}_0 + \beta \mathbf{F}(\mathbf{x}_0) \\ [\mathbf{u}, \mathbf{x}_0; \mathbf{F}] \boldsymbol{\phi}_1 = \mathbf{F}(\mathbf{x}_0) \\ \mathbf{x}_1 = \mathbf{x}_0 - \boldsymbol{\phi}_1 \end{cases} \\ \text{Multi-step part} &\longrightarrow \begin{cases} \text{for } j = 2, m \\ \quad [\mathbf{u}, \mathbf{x}_0; \mathbf{F}] \boldsymbol{\phi}_j = \mathbf{F}(\mathbf{x}_{j-1}) \\ \quad \mathbf{x}_j = \mathbf{x}_{j-1} - \boldsymbol{\phi}_j \\ \text{end} \\ \mathbf{x}_0 = \mathbf{x}_m \end{cases}, \end{aligned} \quad (3)$$

where  $\beta$  is a scalar parameter and the  $[\cdot, \cdot; \mathbf{F}] : D \times D \subset \mathbb{R}^n \times \mathbb{R}^n \longrightarrow L(\mathbb{R}^n)$  is divided difference operator of  $\mathbf{F}$ . The divided difference operator is defined as

$$\begin{aligned} [\mathbf{x} + \mathbf{h}, \mathbf{x}; \mathbf{F}] &= \int_0^1 \mathbf{F}'(\mathbf{x} + t \mathbf{h}) dt, \quad \forall \mathbf{x}, \mathbf{h} \in \mathbb{R}^n \\ &= \mathbf{F}'(\mathbf{x}) + \frac{1}{2} \mathbf{F}''(\mathbf{x}) \mathbf{h} + \frac{1}{6} \mathbf{F}'''(\mathbf{x}) \mathbf{h}^2 + O(\mathbf{h}^3), \end{aligned} \quad (4)$$

where  $\mathbf{h}^i = \overbrace{(h, h, \dots, h)}^{i \text{ times}}$ . The ideas of preconditioning of system of nonlinear equations are reported by many authors [15–18]. Let  $\mathbf{G}(\mathbf{x}) = [g_1(\mathbf{x}), g_2(\mathbf{x}), \dots, g_n(\mathbf{x})]^T$  be a non-zero

sufficiently differentiable function. We define a new function

$$\mathbf{Q}(\mathbf{x}) = \mathbf{G}(\mathbf{x}) \odot \mathbf{F}(\mathbf{x}) = \llbracket \mathbf{G}(\mathbf{x}) \rrbracket \mathbf{F}(\mathbf{x}) = \llbracket \mathbf{F}(\mathbf{x}) \rrbracket \mathbf{G}(\mathbf{x}), \quad (5)$$

where  $\odot$  is the component-wise multiplication and  $\llbracket \cdot \rrbracket$  represent the diagonal matrix. The first order Fréchet derivative of (5) can be computed as

$$\begin{aligned} \mathbf{Q}'(\mathbf{x}) &= \llbracket \mathbf{F}(\mathbf{x}) \rrbracket \mathbf{G}'(\mathbf{x}) + \llbracket \mathbf{G}(\mathbf{x}) \rrbracket \mathbf{F}'(\mathbf{x}) \\ &= \llbracket \mathbf{G}(\mathbf{x}) \rrbracket (\mathbf{F}'(\mathbf{x}) + \llbracket \mathbf{F}(\mathbf{x}) \rrbracket \llbracket \mathbf{G}(\mathbf{x}) \rrbracket^{-1} \mathbf{G}'(\mathbf{x})). \end{aligned} \quad (6)$$

The application of Newton method to (5) gives

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - \mathbf{Q}'(\mathbf{x}_k)^{-1} \mathbf{Q}(\mathbf{x}_k) \\ &= \mathbf{x}_k - (\mathbf{F}'(\mathbf{x}_k) + \llbracket \mathbf{F}(\mathbf{x}_k) \rrbracket \llbracket \mathbf{G}(\mathbf{x}_k) \rrbracket^{-1} \mathbf{G}'(\mathbf{x}_k))^{-1} \mathbf{F}(\mathbf{x}_k). \end{aligned} \quad (7)$$

The convergence order of (7) is quadratic because, it is the Newton method for solving preconditioned system of nonlinear equations  $\mathbf{Q}(\mathbf{x}) = \mathbf{0}$ . If we replace  $\mathbf{G}(\mathbf{x})$  by  $\exp(\mathbf{G}(\mathbf{x}))$  then (7) can be written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\mathbf{F}'(\mathbf{x}_k) + \llbracket \mathbf{F}(\mathbf{x}_k) \rrbracket \mathbf{G}'(\mathbf{x}_k))^{-1} \mathbf{F}(\mathbf{x}_k). \quad (8)$$

## 2 Proposed iterative methods

We are interested to propose derivative-free version of (8) with some generalization of preconditioner. Our proposal of multi-step derivative-free preconditioned iterative method is the following

$$\begin{aligned} \text{Base method} &\longrightarrow \left\{ \begin{array}{l} \mathbf{x}_0 = \text{initialguess} \\ \mathbf{u} = \mathbf{x}_0 + \beta \mathbf{F}(\mathbf{x}_0) \\ \mathbf{A} = [\mathbf{u}, \mathbf{x}_0; \mathbf{F}] + \llbracket \mathbf{q}_1(\mathbf{x}_0) \odot \mathbf{q}_2(\mathbf{F}(\mathbf{x}_0)) \rrbracket \\ \mathbf{A} \boldsymbol{\phi}_1 = \mathbf{F}(\mathbf{x}_0) \\ \mathbf{x}_1 = \mathbf{x}_0 - \boldsymbol{\phi}_1 \end{array} \right. \\ \text{Multi-step part} &\longrightarrow \left\{ \begin{array}{l} \text{for } j = 2, m \\ \quad \mathbf{A} \boldsymbol{\phi}_j = \mathbf{F}(\mathbf{x}_{j-1}) \\ \quad \mathbf{x}_j = \mathbf{x}_{j-1} - \boldsymbol{\phi}_j \\ \text{end} \\ \mathbf{x}_0 = \mathbf{x}_m \end{array} \right. , \end{aligned} \quad (9)$$

where  $\mathbf{q}_1, \mathbf{q}_2 : \mathbb{R}^n \longrightarrow \mathbb{R}^n$  are sufficiently differentiable,  $\mathbf{q}_2(\mathbf{0}) = \mathbf{0}$  and  $\mathbf{q}_i(\mathbf{x}) = [q_i(x_1), q_i(x_2) \dots, q_i(x_n)]^T$  for  $i = 1, 2$ . We claim that the convergence order of proposed preconditioned  $m$ -step derivative-free iterative method is  $m + 1$ .

## 3 Convergence analysis

We demonstrate the proof of convergence order of (9) only for  $m = 2$  and the case  $m \geq 3$ , we use mathematical induction.

**Theorem 3.1** Let  $\mathbf{F} : \Gamma \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$  be sufficiently Frechet differentiable on an open convex neighborhood  $\Gamma$  of  $\mathbf{x}^* \in \mathbb{R}^n$  with  $\mathbf{F}(\mathbf{x}^*) = \mathbf{0}$  and  $\det(\mathbf{F}'(\mathbf{x}^*)) \neq 0$ . By taking  $\mathbf{x}_0$  in the vicinity of  $\mathbf{x}^*$ , the sequence  $\{\mathbf{x}_k\}$  generated by (9) converges to  $\mathbf{x}^*$  with local order of convergence at least three for  $m = 2$  and the following error equation

$$\begin{aligned} \mathbf{e}_2 &= \mathbf{B}_1 \mathbf{e}_0 (-\mathbf{C}_2 \mathbf{e}_0^2 + \mathbf{B}_1 \mathbf{e}_0^2 + \mathbf{C}_1^{-1} [\mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1 \mathbf{C}_1 \mathbf{e}_0)] \mathbf{e}_0) \\ &\quad + \mathbf{C}_1^{-1} [\mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1 \mathbf{C}_1 \mathbf{e}_0)] (-\mathbf{C}_2 \mathbf{e}_0^2 + \mathbf{B}_1 \mathbf{e}_0^2 + \mathbf{C}_1^{-1} [\mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1 \mathbf{C}_1 \mathbf{e}_0)] \mathbf{e}_0) \\ &\quad + O((\mathbf{e}_0)_{i_1} (\mathbf{e}_0)_{i_2} (\mathbf{e}_0)_{i_3} (\mathbf{e}_0)_{i_4}) \end{aligned} \quad (10)$$

where  $\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}^*$ ,  $\mathbf{e}_k^p = \overbrace{(\mathbf{e}_k, \mathbf{e}_k, \dots, \mathbf{e}_k)}^{p \text{ times}}$ ,  $\mathbf{e}_k = [(\mathbf{e}_k)_1, (\mathbf{e}_k)_2, \dots, (\mathbf{e}_k)_n]^T$ ,  $\boldsymbol{\gamma}_1 = \mathbf{q}_2(\mathbf{0})$ ,  $\mathbf{B}_1 = \beta \mathbf{C}_2 \mathbf{C}_1 + 2 \mathbf{C}_2$  and  $\mathbf{B}_2 = \beta^2 \mathbf{C}_3 \mathbf{C}_1^2 + \beta \mathbf{C}_2 \mathbf{C}_1 \mathbf{C}_2 + 3 \beta \mathbf{C}_3 \mathbf{C}_1 + 3 \mathbf{C}_3$ .

*Proof* The  $r$ th Frechet derivative of  $\mathbf{F}$  at  $v \in \mathbb{R}^n$ ,  $r \geq 1$ , is the  $r$ -linear function  $\mathbf{F}^{(r)}(v) : \underbrace{\mathbb{R}^n \mathbb{R}^n \dots \mathbb{R}^n}_{r \text{ times}}$  such that  $\mathbf{F}^{(r)}(v)(u_1, u_2, \dots, u_r) \in \mathbb{R}^n$ . The Taylor's series expansion of  $\mathbf{F}(\mathbf{x}_0)$  around  $\mathbf{x}^*$  can be written as

$$\begin{aligned} \mathbf{F}(\mathbf{x}_0) &= \mathbf{F}(\mathbf{x}^* + \mathbf{x}_0 - \mathbf{x}^*) = \mathbf{F}(\mathbf{x}^* + \mathbf{e}_0), \\ &= \mathbf{F}(\mathbf{x}^*) + \mathbf{F}'(\mathbf{x}^*) \mathbf{e}_0 + \frac{1}{2!} \mathbf{F}''(\mathbf{x}^*) \mathbf{e}_0^2 + \frac{1}{3!} \mathbf{F}^{(3)}(\mathbf{x}^*) \mathbf{e}_0^3 + \dots, \\ &= \mathbf{F}'(\mathbf{x}^*) \left( \mathbf{e}_0 + \frac{1}{2!} \mathbf{F}'(\mathbf{x}^*)^{-1} \mathbf{F}''(\mathbf{x}^*) \mathbf{e}_0^2 + \frac{1}{3!} \mathbf{F}'(\mathbf{x}^*)^{-1} \mathbf{F}^{(3)}(\mathbf{x}^*) \mathbf{e}_0^3 + \dots \right), \\ &= \mathbf{C}_1 (\mathbf{e}_0 + \mathbf{C}_2 \mathbf{e}_0^2 + \mathbf{C}_3 \mathbf{e}_0^3 + O(\mathbf{e}_0^4)), \end{aligned} \quad (11)$$

where  $\mathbf{C}_1 = \mathbf{F}'(\mathbf{x}^*)$  and  $\mathbf{C}_s = \frac{1}{s!} \mathbf{F}^{(s)}(\mathbf{x}^*)^{-1} \mathbf{F}^{(s)}(\mathbf{x}^*)$  for  $s \geq 2$ . From (11), we can calculate the Fréchet derivatives of  $\mathbf{F}(\cdot)$  as

$$\begin{aligned} \mathbf{F}'(\mathbf{x}_0) &= \mathbf{C}_1 (\mathbf{I} + 2\mathbf{C}_2 \mathbf{e}_0 + 3\mathbf{C}_3 \mathbf{e}_0^2 + 4\mathbf{C}_3 \mathbf{e}_0^3 + O(\mathbf{e}_0^4)) \\ \mathbf{F}''(\mathbf{x}_0) &= \mathbf{C}_1 (2\mathbf{C}_2 + 6\mathbf{C}_3 \mathbf{e}_0 + O(\mathbf{e}_0^2)) \\ \mathbf{F}'''(\mathbf{x}_0) &= \mathbf{C}_1 (6\mathbf{C}_3 + O(\mathbf{e}_0)), \end{aligned} \quad (12)$$

where  $\mathbf{I}$  is the identity matrix. By using (4), we get the following expansion

$$[\mathbf{u}, \mathbf{x}_0; \mathbf{F}] = \mathbf{C}_1 (\mathbf{I} + \mathbf{B}_1 \mathbf{e}_0 + \mathbf{B}_2 \mathbf{e}_0^2 + O(\mathbf{e}_0^3)), \quad (13)$$

where  $\mathbf{B}_1 = \beta \mathbf{C}_2 \mathbf{C}_1 + 2 \mathbf{C}_2$  and  $\mathbf{B}_2 = \beta^2 \mathbf{C}_3 \mathbf{C}_1^2 + \beta \mathbf{C}_2 \mathbf{C}_1 \mathbf{C}_2 + 3 \beta \mathbf{C}_3 \mathbf{C}_1 + 3 \mathbf{C}_3$ . Next, we expand  $\mathbf{q}_2(\mathbf{F}(\mathbf{x}))$  as

$$\begin{aligned} \mathbf{q}_2(\mathbf{F}(\mathbf{x})) &= \mathbf{q}'_2(\mathbf{0}) \mathbf{F}(\mathbf{x}) + \mathbf{q}''_2(\mathbf{0}) \mathbf{F}(\mathbf{x}) \odot \mathbf{F}(\mathbf{x}) + \dots \\ \mathbf{q}_1(\mathbf{x}) \odot \mathbf{q}_2(\mathbf{F}(\mathbf{x})) &= \mathbf{q}_1(\mathbf{x}) \odot \mathbf{q}'_2(\mathbf{0}) \mathbf{F}(\mathbf{x}) + \mathbf{q}_1(\mathbf{x}) \odot \mathbf{q}''_2(\mathbf{0}) \mathbf{F}(\mathbf{x}) \odot \mathbf{F}(\mathbf{x}) + \dots \\ &= \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1 \mathbf{C}_1 \mathbf{e}_0) + \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1 \mathbf{C}_1 \mathbf{C}_2 \mathbf{e}_0^2) + \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_2 [\mathbf{C}_1 \mathbf{e}_0] \mathbf{C}_1 \mathbf{e}_0) \\ &\quad + O((\mathbf{e}_0)_{i_1} (\mathbf{e}_0)_{i_2} (\mathbf{e}_0)_{i_3}), \end{aligned} \quad (14)$$

where  $\boldsymbol{\gamma}_1 = \mathbf{q}'_2(\mathbf{0})$ ,  $\boldsymbol{\gamma}_2 = \mathbf{q}''_2(\mathbf{0})$  and  $O((\mathbf{e}_0)_{i_1}(\mathbf{e}_0)_{i_2}(\mathbf{e}_0)_{i_3})$  represent the third order term. The expansion of  $\mathbf{A}$  by using (13) and (14) is

$$\begin{aligned}
\mathbf{A} &= \mathbf{C}_1(\mathbf{I} + \mathbf{B}_1\mathbf{e}_0 + \mathbf{B}_2\mathbf{e}_0^2) + \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{e}_0) \rrbracket + \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{C}_2\mathbf{e}_0^2) \rrbracket \\
&\quad + \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_2 \llbracket \mathbf{C}_1\mathbf{e}_0 \rrbracket \mathbf{C}_1\mathbf{e}_0) \rrbracket + O((\mathbf{e}_0)_{i_1}(\mathbf{e}_0)_{i_3}(\mathbf{e}_0)_{i_3}) \\
\mathbf{A} &= \mathbf{C}_1(\mathbf{I} + \mathbf{B}_1\mathbf{e}_0 + \mathbf{B}_2\mathbf{e}_0^2 + \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{e}_0) \rrbracket) + \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{C}_2\mathbf{e}_0^2) \rrbracket \\
&\quad + \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_2 \llbracket \mathbf{C}_1\mathbf{e}_0 \rrbracket \mathbf{C}_1\mathbf{e}_0) \rrbracket + O((\mathbf{e}_0)_{i_1}(\mathbf{e}_0)_{i_2}(\mathbf{e}_0)_{i_3}) \\
\mathbf{A}^{-1} &= (\mathbf{I} - \mathbf{B}_1\mathbf{e}_0 - \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{e}_0) \rrbracket) - \mathbf{B}_2\mathbf{e}_0^2 - \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{C}_2\mathbf{e}_0^2) \rrbracket \\
&\quad - \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_2 \llbracket \mathbf{C}_1\mathbf{e}_0 \rrbracket \mathbf{C}_1\mathbf{e}_0) \rrbracket + O((\mathbf{e}_0)_{i_1}(\mathbf{e}_0)_{i_2}(\mathbf{e}_0)_{i_3})\mathbf{C}_1^{-1} \\
\mathbf{A}^{-1} &= (\mathbf{I} - \mathbf{B}_1\mathbf{e}_0 - \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{e}_0) \rrbracket) + O((\mathbf{e}_0)_{i_1}(\mathbf{e}_0)_{i_2})\mathbf{C}_1^{-1}. \tag{15}
\end{aligned}$$

By using (15), we get

$$\begin{aligned}
\mathbf{A}^{-1}\mathbf{F}(\mathbf{x}_0) &= (\mathbf{I} - \mathbf{B}_1\mathbf{e}_0 - \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{e}_0) \rrbracket) + O((\mathbf{e}_0)_{i_1}(\mathbf{e}_0)_{i_2})\mathbf{C}_1^{-1}\mathbf{C}_1(\mathbf{e}_0 + \mathbf{C}_2\mathbf{e}_0^2 + O(\mathbf{e}_0^3)) \\
&= \mathbf{e}_0 + \mathbf{C}_2\mathbf{e}_0^2 - \mathbf{B}_1\mathbf{e}_0^2 - \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{e}_0) \rrbracket \mathbf{e}_0 + O((\mathbf{e}_0)_{i_1}(\mathbf{e}_0)_{i_2}(\mathbf{e}_0)_{i_3}) \\
\mathbf{e}_1 &= \mathbf{e}_0 - \mathbf{e}_0 - \mathbf{C}_2\mathbf{e}_0^2 + \mathbf{B}_1\mathbf{e}_0^2 + \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{e}_0) \rrbracket \mathbf{e}_0 + O((\mathbf{e}_0)_{i_1}(\mathbf{e}_0)_{i_2}(\mathbf{e}_0)_{i_3}) \\
\mathbf{e}_1 &= -\mathbf{C}_2\mathbf{e}_0^2 + \mathbf{B}_1\mathbf{e}_0^2 + \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{e}_0) \rrbracket \mathbf{e}_0 + O((\mathbf{e}_0)_{i_1}(\mathbf{e}_0)_{i_2}(\mathbf{e}_0)_{i_3}). \tag{16}
\end{aligned}$$

The error equation  $\mathbf{e}_1$  tells that the order of convergence of base method of (9) is quadratic. The order of convergence of (9) for  $m = 2$  is three which is established as

$$\begin{aligned}
\mathbf{F}(\mathbf{x}_1) &= \mathbf{C}_1(\mathbf{e}_1 + O(\mathbf{e}_1^2)) \\
\mathbf{e}_2 &= \mathbf{e}_1 - \mathbf{A}^{-1}\mathbf{C}_1(\mathbf{e}_1 + O(\mathbf{e}_1^2)) \\
&= \mathbf{e}_1 - (\mathbf{I} - \mathbf{B}_1\mathbf{e}_0 - \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{e}_0) \rrbracket) + O((\mathbf{e}_0)_{i_1}(\mathbf{e}_0)_{i_2})\mathbf{C}_1^{-1}\mathbf{C}_1(\mathbf{e}_1 + O(\mathbf{e}_1^2)) \\
&= \mathbf{B}_1\mathbf{e}_0\mathbf{e}_1 + \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{e}_0) \rrbracket \mathbf{e}_1 + O((\mathbf{e}_0)_{i_1}(\mathbf{e}_0)_{i_2}(\mathbf{e}_0)_{i_3}(\mathbf{e}_0)_{i_4}) \\
&= \mathbf{B}_1\mathbf{e}_0(-\mathbf{C}_2\mathbf{e}_0^2 + \mathbf{B}_1\mathbf{e}_0^2 + \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{e}_0) \rrbracket \mathbf{e}_0) \\
&\quad + \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{e}_0) \rrbracket (-\mathbf{C}_2\mathbf{e}_0^2 + \mathbf{B}_1\mathbf{e}_0^2 + \mathbf{C}_1^{-1} \llbracket \mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1\mathbf{C}_1\mathbf{e}_0) \rrbracket \mathbf{e}_0) \\
&\quad + O((\mathbf{e}_0)_{i_1}(\mathbf{e}_0)_{i_2}(\mathbf{e}_0)_{i_3}(\mathbf{e}_0)_{i_4}). \tag{17}
\end{aligned}$$

It can be seen from (17) that it involves third order terms in  $\mathbf{e}_0$ . □

The proof of convergence when  $m > 3$  can be carried via mathematical induction. Suppose the proposed iterative method (9) has convergence order  $s$  when  $m = s - 1$ . The error equation for  $s$ -step iterative method (9) can be written as

$$\mathbf{e}_s = \mathbf{e}_{s-1} - \mathbf{A}^{-1}\mathbf{F}(\mathbf{x}_{s-1}). \tag{18}$$

The Taylor series expansion of  $\mathbf{F}(\cdot)$  around  $\mathbf{x}^*$  is

$$\mathbf{F}(\mathbf{x}_{s-1}) = \mathbf{F}(\mathbf{x}_{s-1} - \mathbf{x}^* + \mathbf{x}^*) = \mathbf{F}(\mathbf{e}_{s-1} + \mathbf{x}^*) = \mathbf{C}_1(\mathbf{e}_{s-1} + O(\mathbf{e}_{s-1}^2)).$$

By substituting the expressions of  $\mathbf{A}^{-1}$  and  $\mathbf{F}(\mathbf{x}_{s-1})$  in (18) we get

$$\begin{aligned}
\mathbf{e}_s &= \mathbf{e}_{s-1} - (\mathbf{I} - \mathbf{B}_1 \mathbf{e}_0 - \mathbf{C}_1^{-1} [\mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1 \mathbf{C}_1 \mathbf{e}_0)]) + O((\mathbf{e}_0)_{i_1} (\mathbf{e}_0)_{i_2}) \mathbf{C}_1^{-1} \mathbf{C}_1 (\mathbf{e}_{s-1} + O(\mathbf{e}_{s-1}^2)) \\
&= \mathbf{e}_{s-1} - (\mathbf{I} - \mathbf{B}_1 \mathbf{e}_0 - \mathbf{C}_1^{-1} [\mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1 \mathbf{C}_1 \mathbf{e}_0)]) + O((\mathbf{e}_0)_{i_1} (\mathbf{e}_0)_{i_2}) (\mathbf{e}_{s-1} + O(\mathbf{e}_{s-1}^2)) \\
&= (\mathbf{B}_1 \mathbf{e}_0 + \mathbf{C}_1^{-1} [\mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1 \mathbf{C}_1 \mathbf{e}_0)]) + O((\mathbf{e}_0)_{i_1} (\mathbf{e}_0)_{i_2}) \mathbf{e}_{s-1} \\
&\quad - (\mathbf{I} - \mathbf{B}_1 \mathbf{e}_0 - \mathbf{C}_1^{-1} [\mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1 \mathbf{C}_1 \mathbf{e}_0)]) + O((\mathbf{e}_0)_{i_1} (\mathbf{e}_0)_{i_2}) O(\mathbf{e}_{s-1}^2) \\
&= (\mathbf{B}_1 \mathbf{e}_0 + \mathbf{C}_1^{-1} [\mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1 \mathbf{C}_1 \mathbf{e}_0)]) \mathbf{e}_{s-1} + O(\mathbf{e}_{s+1}) \\
&= (\mathbf{B}_1 \mathbf{e}_0 + \mathbf{C}_1^{-1} [\mathbf{q}_1(\mathbf{x}) \odot (\boldsymbol{\gamma}_1 \mathbf{C}_1 \mathbf{e}_0)]) O((\mathbf{e}_0)_{i_1} (\mathbf{e}_0)_{i_2} \cdots (\mathbf{e}_0)_{i_s}) + O(\mathbf{e}_{s+1}) \\
&= O(\mathbf{e}_0) O((\mathbf{e}_0)_{i_1} (\mathbf{e}_0)_{i_2} \cdots (\mathbf{e}_0)_{i_s}) + O(\mathbf{e}_{s+1}) \\
\mathbf{e}_s &= O((\mathbf{e}_0)_{i_1} (\mathbf{e}_0)_{i_2} \cdots (\mathbf{e}_0)_{i_s} (\mathbf{e}_0)_{i_{s+1}}) + O(\mathbf{e}_{s+1}) \tag{19}
\end{aligned}$$

According to our assumption the order of convergence of  $(s - 1)$ -step method is  $s$ . It means  $\mathbf{e}_{s-1} = O((\mathbf{e}_0)_{i_1} (\mathbf{e}_0)_{i_2} \cdots (\mathbf{e}_0)_{i_s})$ . It is clearly evident from (19) that  $\mathbf{e}_s = O((\mathbf{e}_0)_{i_1} (\mathbf{e}_0)_{i_2} \cdots (\mathbf{e}_0)_{i_{s+1}})$ . So  $m$ -step iterative method (9) has convergence order  $m + 1$ .

## 4 Numerical simulations

It is important to verify the claimed order of convergence of the proposed iterative method. We adopt the following definition of computational order of convergence

$$\text{COC} = \frac{\log(\|\mathbf{F}(\mathbf{x}_{k+1})\|_\infty / \|\mathbf{F}(\mathbf{x}_k)\|_\infty)}{\log(\|\mathbf{F}(\mathbf{x}_k)\|_\infty / \|\mathbf{F}(\mathbf{x}_{k-1})\|_\infty)}. \tag{20}$$

To check the performance of our proposed iterative method, we solve three problems.

$$\text{Problem 1} = \begin{cases} F_i(\mathbf{x}) = x_i^2 x_{i+1} - 1 = 0, & i = 1, 2, \dots, n-1 \\ F_n(\mathbf{x}) = x_n x_1 - 1 = 0 \end{cases}$$

$$\text{Problem 2} = \begin{cases} F_1(\mathbf{x}) = (3 - 0.5 x_1) x_1 - 2 x_2 + 1 \\ F_n(\mathbf{x}) = (3 - 0.5 x_n) x_n - 2 x_{n-1} + 1 \\ F_i(\mathbf{x}) = (3 - 0.5 x_i) x_i - x_{i-1} + 2 x_{i+1} + 1, & i = 2, 3, \dots, n-1 \end{cases}$$

$$\text{Problem 3} = \begin{cases} F_i(\mathbf{x}) = 2 \left( n + i (1 - \cos(x_i)) - \sin(x_i) - \sum_{j=1}^n \cos(x_j) \right) \\ \quad \times (2 \sin(x_i) - \cos(x_i)), & i = 1, 2, \dots, n \end{cases}$$

The performance comparison is demonstrated between iterative methods (3) and (9). The first order divided difference operator (4) can be computed as

$$\begin{aligned}
[\mathbf{x}, \mathbf{y}; \mathbf{F}]_{ij} &= \frac{F_i(y_1, y_2, \dots, y_{j-1}, y_j, y_{j+1}, \dots, x_n) - F_i(y_1, y_2, \dots, y_{j-1}, x_j, x_{j+1}, \dots, x_n)}{y_j - x_j}, \tag{21}
\end{aligned}$$

$\mathbf{x} = [x_1, x_2, \dots, x_n]^T$  and  $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$  and  $i, j = 1, 2, \dots, n$ . If  $\mathbf{F}(\mathbf{x})$  and  $\mathbf{F}(\mathbf{y})$  are provided separately then number of scalar function evaluation in (21) are  $n(n - 1)$ . The divided difference approximation (21) of (4) is first order accurate. A detailed discussion about different divided difference approximation of Jacobian can be found in [14]. Mathematica

**Table 1** Problem 1: Initial guess:  $x_i = 15/10$ ,  $n = 10$ , Iter= 5,  $\beta = 1/100$ 

Method (9)		Method (3)				
$q_1(\mathbf{x})$	$q_2(\mathbf{F}(\mathbf{x}))$	$m$	$\ \mathbf{F}(\mathbf{x})\ _\infty$	COC	$\ \mathbf{F}(\mathbf{x})\ _\infty$	COC
1	$-\mathbf{F}(\mathbf{x})$	1	$1.41e-46$	2.0	$9.12e-14$	2.0
1	$-\mathbf{F}(\mathbf{x}) + \mathbf{F}(\mathbf{x})^3/100$	1	$4.77e-52$	2.0	$9.12e-14$	2.0
1	$-\mathbf{F}(\mathbf{x})$	2	$9.23e-220$	3.0	$4.24e-81$	3.0
1	$-\mathbf{F}(\mathbf{x}) + \mathbf{F}(\mathbf{x})^3/100$	2	$5.61e-245$	3.0	$4.24e-81$	3.0
1	$-\mathbf{F}(\mathbf{x})$	3	$4.99e-754$	4.0	$3.63e-310$	4.0
1	$-\mathbf{F}(\mathbf{x}) + \mathbf{F}(\mathbf{x})^3/100$	3	$3.63e-827$	4.0	$3.63e-310$	4.0
1	$-\mathbf{F}(\mathbf{x})$	4	$7.95e-2062$	5.0	$1.19e-900$	5.0
1	$-\mathbf{F}(\mathbf{x}) + \mathbf{F}(\mathbf{x})^3/100$	4	$6.44e-2225$	5.0	$1.19e-900$	5.0
1	$-\mathbf{F}(\mathbf{x})$	5	$2.21e-4799$	6.0	$6.53e-2175$	6.0
1	$-\mathbf{F}(\mathbf{x}) + \mathbf{F}(\mathbf{x})^3/100$	5	$6.48e-5105$	6.0	$6.53e-2175$	6.0
$\sin(\mathbf{x})$	$-\mathbf{F}(\mathbf{x})$	5	$2.21e-4536$	6.0	$6.53e-2175$	6.0
$\cos(\mathbf{x})$	$-\mathbf{F}(\mathbf{x})$	5	$3.66e-2464$	6.0	$6.53e-2175$	6.0
$\exp(-\mathbf{x}/10)$	$-\mathbf{F}(\mathbf{x})$	5	$6.90e-5217$	6.0	$6.53e-2175$	6.0
1	$-\sin(\mathbf{F}(\mathbf{x}))$	6	$4.56e-6550$	7.0	$4.79e-4608$	7.0
1	$-\tan(\mathbf{F}(\mathbf{x}))$	6	$7.98e-5136$	7.0	$4.79e-4608$	7.0
1	$-\mathbf{F}(\mathbf{x})/(1 + \mathbf{F}(\mathbf{x}))$	6	$4.34e-6513$	7.0	$4.79e-4608$	7.0
1	$-\mathbf{F}(\mathbf{x})/(1 +  \mathbf{F}(\mathbf{x})/100 )$	6	$2.20e-10069$	7.0	$4.79e-4608$	7.0

code is given in the Appendix that can be used to compute the first order divided difference operator (21).

Tables 1, 2 and 3 clearly shows that the computational convergence orders confirm our claim that  $m$ -step iterative methods (9) and (3) have convergence order  $m + 1$ . In all test problems, the selection of preconditioners show that the performance of proposed iterative method (9) is better than competitor iterative method (3). The computational cost of preconditioners is reasonable because we use diagonal preconditioners. We observed that the leading coefficients in all preconditioners are of order one or less than one in magnitude. By choosing properly preconditioners we obtain high accuracy in numerical results. The value of parameter  $\beta$  is set to  $1/100$  to approximate divided difference operator in all three problems. One may use a smaller value of  $\beta$  but at some point there is a degradation in the accuracy of numerical results and we found that  $1/100$  is a reasonable choice for set of selected problems. It is hard to devise some rule to select nonlinear preconditioners for system of nonlinear equations but we observed that a small contribution of higher powers (greater than one) of  $\mathbf{F}(\mathbf{x})$  provides better accuracy in the numerical results.

## 5 Conclusions

The derivative-free iterative methods become important when the system of nonlinear equations is a black box and we compute Jacobian numerically. The multi-step iterative methods are efficient and provide a high order of convergence. The high efficiency of multi-step iterative methods is hidden in the fact that we repeatedly use LU factors of frozen Jacobian from the base method in the multi-step part. The computational cost that we pay is the per

**Table 2** Problem 2: Initial guess:  $x_i = -1$ ,  $n = 100$ , Iter = 5,  $\beta = 1/100$

Method (9)		Method (3)				
$\mathbf{q}_1(\mathbf{x})$	$\mathbf{q}_2(\mathbf{F}(\mathbf{x}))$	$m$	$\ \mathbf{F}(\mathbf{x})\ _\infty$	COC	$\ \mathbf{F}(\mathbf{x})\ _\infty$	COC
<b>1</b>	$\mathbf{F}(\mathbf{x})/10$	1	$6.29e-36$	2.0	$2.25e-24$	2.0
<b>1</b>	$\mathbf{F}(\mathbf{x})/10 - (\mathbf{F}(\mathbf{x})/10)^2 + (\mathbf{F}(\mathbf{x})/10)^3$	1	$2.47e-41$	2.0	$2.25e-24$	2.0
<b>1</b>	$\mathbf{F}(\mathbf{x})/10$	2	$5.04e-224$	3.0	$3.60e-161$	3.0
$\exp(-\mathbf{x}/10)$	$\mathbf{F}(\mathbf{x})/10$	2	$5.31e-231$	3.0	$3.60e-161$	3.0
<b>1</b>	$\mathbf{F}(\mathbf{x})/10 - (\mathbf{F}(\mathbf{x})/10)^2 + (\mathbf{F}(\mathbf{x})/10)^3$	2	$4.72e-233$	3.0	$3.60e-161$	3.0
$\exp(-\mathbf{x}/10)$	$\mathbf{F}(\mathbf{x})/10 - (\mathbf{F}(\mathbf{x})/10)^2 + (\mathbf{F}(\mathbf{x})/10)^3$	2	$2.52e-243$	3.0	$3.60e-161$	3.0



**Table 3** Problem 3: Initial guess:  $x_i = 1, n = 20, \text{Iter} = 10, \beta = 1/100$

Method (9)				Method (3)		
$q_1(\mathbf{x})$	$q_2(\mathbf{F}(\mathbf{x}))$	$m$	$\ \mathbf{F}(\mathbf{x})\ _\infty$	COC	$\ \mathbf{F}(\mathbf{x})\ _\infty$	COC
<b>1</b>	$-\mathbf{F}(\mathbf{x})$	1	1.49e-130	2.0	5.06e-73	2.0
$exp(\mathbf{x}/10)$	$-\mathbf{F}(\mathbf{x})$	1	1.33e-137	2.0	5.06e-73	2.0
<b>1</b>	$-\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{x})^2/100$	1	1.76e-146	2.0	5.06e-73	2.0
<b>1</b>	$-\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{x})^2/100 - \mathbf{F}(\mathbf{x})^3/10,000$	1	1.87e-152	2.0	5.06e-73	2.0
$cos(\mathbf{x})$	$-\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{x})^2/100 - \mathbf{F}(\mathbf{x})^3/10,000$	1	2.01e-116	2.0	5.06e-73	2.0
$sin(\mathbf{x})$	$-\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{x})^2/100 - \mathbf{F}(\mathbf{x})^3/10,000$	1	7.53e-122	2.0	5.06e-73	2.0
$sinh(\mathbf{x})$	$-\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{x})^2/100 - \mathbf{F}(\mathbf{x})^3/10,000$	1	3.72e-153	2.0	5.06e-73	2.0
$exp(\mathbf{x}/10)$	$-\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{x})^2/100 - \mathbf{F}(\mathbf{x})^3/10,000$	1	1.62e-165	2.0	5.06e-73	2.0
						⋮

step single evaluation of the system of nonlinear equation and solution of lower and upper triangular systems. The proposed preconditioners offer high numerical accuracy in the computed solutions with very low computational cost. It can be seen that the embedding of our proposed preconditioners modifies the Jacobian without altering the convergence order with very low computational cost.

**Acknowledgements** We are grateful to reviewer for his valuable comments to enhance the quality of the research article.

## 6 Appendix

```

1
2 (*
3 Problem 1
4
5 iterations = 5
6 m = 5
7 n = 10
8 beta = 1/100
9 initial guess x_i = 15/10
10 q1(x) = -F(x)
11 q2(x) = sin(x)
12 *)
13
14 ClearAll["Global '*"]
15 percision = 2*3600;
16 $MinPrecision = percision;
17
18 F[xz_, n_] := (
19   yyz = Table[0, {i, 1, n}];
20   Do[ yyz[[i]] = xz[[i]]^2*xz[[i + 1]] - 1, {i, 1, n - 1}];
21   yyz[[n]] = xz[[n]]^2*xz[[1]] - 1;
22   Return[yyz];
23 );
24
25 Fi[xs_, i_, n_] := (
26   If[i == n, zs = xs[[n]]^2*xs[[1]] - 1,
27     zs = xs[[i]]^2*xs[[i + 1]] - 1];
28   Return[zs];
29 );
30
31 dF[xq_, yq_, F1q_, F2q_, n_] := (
32   Mq = Table[0, {i, 1, n}, {j, 1, n}];
33   Do[
34     z1q = yq;
35     z1q[[1]] = xq[[1]];
36     dumq = Fi[z1q, i, n];
37     Mq[[i, 1]] = (dumq - F2q[[i]])/(xq[[1]] - yq[[1]]);
38     Do[ z1q[[j]] = xq[[j]];
39         dum1q = Fi[z1q, i, n];
40         Mq[[i, j]] = (dum1q - dumq)/(xq[[j]] - yq[[j]]);
41         dumq = dum1q;
42     , {j, 2, n - 1}];

```

```

43     Mq[[i, n]] = (F1q[[i]] - dumq)/(xq[[n]] - yq[[n]]);
44     , {i, 1, n}];
45     Return[Mq];
46 );
47 n = 10;
48 iter = 5;
49 m = 5;
50 guess = Table[15/10, {i, 1, n}];
51 guess = SetPrecision[guess, percision];
52 err = Table[0, {i, 1, iter}];
53 b = 1/100;
54 x0 = guess;
55
56 Do[
57     fx0 = F[x0, n];
58     q1 = Sin[x0];
59     q2 = -fx0;
60     u = x0 + b*fx0;
61     fu = F[u, n];
62     A = dF[u, x0, fu, fx0, n] + DiagonalMatrix[q1*q2];
63     lusolve = LinearSolve[A];
64     phil = lusolve[fx0];
65     x1 = x0 - phil;
66     Do[
67         fx1 = F[x1, n];
68         x1 = x1 - lusolve[fx1];
69         , {i, 2, m}];
70         x0 = x1;
71         err[[j]] = Norm[F[x0, n], Infinity];
72         , {j, 1, iter}];
73
74 (* Compute the computational order of convergence *)
75
76 Do[
77     rhof = Log[err[[i + 2]]/err[[i + 1]]]/Log[err[[i + 1]]/err[[i]]];
78     , {i, 1, iter - 2}];
79
80 (* Display the results *)
81 percision = 10;
82 $MinPrecision = percision;
83 N[err, percision] // MatrixForm
84 Print[Computational order of convergence];
85 N[rhof, percision] // MatrixForm
86 (* ***** Program finish here ***** *)
87 (*Output of the program*)
88
89 Error(1) = 0.001151877320
90 Error(2) = 3.639375119e-21
91 Error(3) = 3.597261495e-126
92 Error(4) = 3.354618470e-756
93 Error(5) = 2.206327013e-4536
94
95 (* *** Computational order of Convergence *** *)
96
97 COC = 6

```

## References

1. Ortega, J.M., Rheinboldt, W.C.: Iterative Solution of Nonlinear Equations in Several Variables. Academic Press limited 24-28 Oval road, London NW1 7DX, United Kingdom (1970)
2. Traub, J.F.: Iterative Methods for the Solution of Equations. Prentice-Hall, Englewood Cliffs (1964)
3. Burden, R.L., Faires, J.D.: Numerical Analysis. PWS Publishing Company, Boston (2001)
4. McNamee, J.M.: Numerical Methods for Roots of Polynomials, Part I. Elsevier, Amsterdam (2007)
5. Amat, S., Busquier, S., Grau, À., Grau-Sánchez, M.: Maximum efficiency for a family of Newton-like methods with frozen derivatives and some applications. *Appl. Math. Comput.* **219**(15), 7954–7963 (2013)
6. Ullah, M.Z., Soleymani, F., Al-Fhaid, A.S.: Numerical solution of nonlinear systems by a general class of iterative methods with application to nonlinear PDEs. *Numer. Algorithms* **67**(1), 223–242 (2014)
7. Ahmad, F., Tohidi, E., Carrasco, J.A.: A parameterized multi-step Newton method for solving systems of nonlinear equations. *Numer. Algorithms* **71**(3), 1017–1398 (2015)
8. Ullah, M.Z., Serra-Capizzano, S., Ahmad, F.: An efficient multi-step iterative method for computing the numerical solution of systems of nonlinear equations associated with ODEs. *Appl. Math. Comput.* **250**, 249–259 (2015)
9. Ahmad, F., Tohidi, E., Ullah, M.Z., Carrasco, J.A.: Higher order multi-step Jarratt-like method for solving systems of nonlinear equations: application to PDEs and ODEs. *Comput. Math. Appl.* **70**(4), 624–636 (2015)
10. Sharma, J.R., Arora, H.: A novel derivative free algorithm with seventh order convergence for solving systems of nonlinear equations. *Numer. Algorithms* **67**(4), 917–933 (2014)
11. Sharma, J.R., Arora, H.: An efficient derivative free iterative method for solving systems of nonlinear equations. *Appl. Anal. Discrete Math.* **7**, 390–403 (2013)
12. Grau-Sánchez, M., Grau, À., Noguera, M.: Frozen divided difference scheme for solving systems of nonlinear equations. *J. Comput. Appl. Math.* **235**(6), 1739–1743 (2011)
13. Grau-Sánchez, M., Noguera, M.: A technique to choose the most efficient method between secant method and some variants. *Appl. Math. Comput.* **218**, 6415–6426 (2012)
14. Grau-Sánchez, M., Noguera, M., Amat, S.: On the approximation of derivatives using divided difference operators preserving the local convergence order of iterative methods. *J. Comput. Appl. Math.* **237**, 363–372 (2013)
15. Wu, X.: Note on the improvement of Newton's method for systems of nonlinear equations. *Appl. Math. Comput.* **189**, 1476–1479 (2007)
16. Hueso, J.L., Martínez, E., Torregrosa, J.R.: Modified Newton's method for systems of nonlinear equations with singular Jacobian. *J. Comput. Appl. Math.* **224**, 77–83 (2009)
17. Noor, M.A., Shah, F.A.: A family of iterative schemes for finding zeros of nonlinear equations having unknown multiplicity. *Appl. Math. Inf. Sci.* **8**(5), 2367–2373 (2014)
18. Noor, M.A., Waseem, M., Noor, K.I., Al-Said, E.: Variational iteration technique for solving a system of nonlinear equations. *Optim. Lett.* **7**(5), 991–1007 (2013)