Applied Mathematics and Computation 290 (2016) 98–110

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

An improved Newton–Traub composition for solving systems of nonlinear equations

Janak Raj Sharma[a], Rajni Sharma[b,*], Ashu Bahl[c,d]

[a] Department of Mathematics, Sant Longowal Institute of Engineering and Technology, Longowal 148106, Punjab, India
[b] Department of Applied Sciences, D.A.V. Institute of Engineering and Technology, Jalandhar 144008, Punjab, India
[c] Research Scholar, I.K. Gujral Punjab Technical University, Jalandhar–Kapurthala Highway, Kapurthala 144601, Punjab, India
[d] Department of Mathematics, D.A.V. College, Jalandhar 144008, Punjab, India

ARTICLE INFO

Keywords:
Systems of nonlinear equations
Newton method
Traub's method
Order of convergence
Computational efficiency
Basins of attraction

ABSTRACT

In this paper, we present a modified Newton–Traub composition with increasing order of convergence for solving systems of nonlinear equations. The idea is based on the recent development by Sharma et al. (2015). Analysis of convergence shows that the presented method has sixth order of convergence. Computational efficiency of the new method is considered and compared with some well-known existing methods. Numerical tests are performed on some problems of different nature, which confirm robust and efficient convergence behavior of the proposed method. Moreover, theoretical results concerning order of convergence and computational efficiency are verified in the numerical problems. The basins of attraction of existing methods and the presented method are given to demonstrate their performance.

1. Introduction

Constructing fixed point methods for solving nonlinear equations and systems of nonlinear equations is one of the most attractive topics in the theory of numerical analysis, with wide applications in science and engineering. A great importance of this topic has led to the development of many numerical methods, most frequently of iterative nature (see [1–5]). 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 finding solution of the system of nonlinear equations $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ by iterative methods of a high order of convergence. This problem can be precisely stated as to find a vector $\mathbf{r} = (r_1, r_2, \ldots, r_n)^T$ such that $\mathbf{F}(\mathbf{r}) = \mathbf{0}$, where $\mathbf{F} : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ is the given nonlinear vector function $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x}))^T$ and $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$. The solution vector $\mathbf{r}$ of $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ can be obtained as a fixed point of some function $\phi : \mathbb{R}^n \to \mathbb{R}^n$ by means of the fixed point iteration

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

One of the basic procedures for solving systems of nonlinear equations is the quadratically convergent Newton method (see [1–3]), which is given as,

$$\mathbf{x}^{(k+1)} = \phi_1^{(2)}(\mathbf{x}^{(k)}) = \mathbf{x}^{(k)} - \mathbf{F}'(\mathbf{x}^{(k)})^{-1}\mathbf{F}(\mathbf{x}^{(k)}), \tag{1}$$

* Corresponding author. Tel.: +91-9814849123; fax: +91 1812205851.
E-mail addresses: jrshira@yahoo.co.in (J.R. Sharma), rajni_daviet@yahoo.com (R. Sharma), bahl.ashu@rediffmail.com (A. Bahl).

http://dx.doi.org/10.1016/j.amc.2016.05.051
0096-3003/

where $\mathbf{F}'(\mathbf{x})^{-1}$ is the inverse of first Fréchet derivative $\mathbf{F}'(\mathbf{x})$ of the function $\mathbf{F}(\mathbf{x})$. In terms of computational cost the Newton method requires the evaluations of one $\mathbf{F}$, one $\mathbf{F}'$ and one matrix inversion per iteration. Throughout the paper, we use the abbreviation $\phi_i^{(p)}$ to denote an ith iterative function of convergence order p.

To improve the order of convergence of Newton method, a number of higher order methods have been proposed in literature. For example, Frontini and Sormani [6], Homeier [7], Cordero and Torregrosa [8], Noor and Waseem [9], Grau-Sánchez et al. [10] and Xiao and Yin [11] have developed third order methods requiring one $\mathbf{F}$, two $\mathbf{F}'$ and two matrix inversions per iteration. Cordero and Torregrosa have also derived two third-order methods in [12]. One of the methods require one $\mathbf{F}$ and three $\mathbf{F}'$ whereas other requires one $\mathbf{F}$ and four $\mathbf{F}'$ evaluations per iteration. Both the methods also require two matrix inversions in each iteration. Darvishi and Barati [13] have proposed a third order method which uses two $\mathbf{F}$, one $\mathbf{F}'$ and one matrix inversion. Grau-Sánchez et al. presented a fourth order method in [10] utilizing three $\mathbf{F}$, one $\mathbf{F}'$ and one matrix inversion. Cordero et al. presented a fourth order method in [14], which uses two $\mathbf{F}$, two $\mathbf{F}'$ and one matrix inversion. Darvishi and Barati [15] developed a fourth order method requiring two $\mathbf{F}$, three $\mathbf{F}'$ and two matrix inversions per iteration. Cordero et al. [16] have implemented fourth order Jarratt's method [17] for scalar equations to systems of equations which requires one $\mathbf{F}$, two $\mathbf{F}'$ and two matrix inversions. Sharma et al. [18] developed a fourth order method requiring one $\mathbf{F}$, two $\mathbf{F}'$ and two matrix inversions. Recently, Narang et al. [19] developed a fourth order family of methods requiring one $\mathbf{F}$, two $\mathbf{F}'$ and two matrix inversions per iteration.

In quest of more fast algorithms, researchers have also proposed fifth and sixth order methods in [10,11,14,16,19–24]. The fifth order methods by Grau-Sánchez et al. [10] and Cordero et al. [16,20] require four evaluations namely, two $\mathbf{F}$ and two $\mathbf{F}'$ per iteration. The fifth order method by Cordero et al. [14] requires three $\mathbf{F}$ and two $\mathbf{F}'$. Xiao and Yin [11] and Sharma and Gupta [21] developed fifth order methods requiring two $\mathbf{F}$ and two $\mathbf{F}'$. In addition, the fifth order methods in [10,11,21] require two matrix inversions, in [14] one matrix inversion and in [16,20] three matrix inversions. Sixth order methods by Cordero et al. [16], Narang et al. [19], Sharma and Arora [22], Lotfi et al. [23] and Esmaeili and Ahmadi [24] use two $\mathbf{F}$ and two $\mathbf{F}'$, whereas the sixth order method by Cordero et al. [20] uses three $\mathbf{F}$ and two $\mathbf{F}'$. The sixth order methods, apart from the mentioned evaluations, also require one matrix inversion per iteration in Sharma–Arora method [22] while the others require two matrix inversions per one iteration.

Recently, Sharma et al. [25] proposed the following fifth order composite Newton–Traub family of methods:

$$\begin{aligned}
\mathbf{y}^{(k)} &= \mathbf{x}^{(k)} - \theta\,\mathbf{F}'(\mathbf{x}^{(k)})^{-1}\mathbf{F}(\mathbf{x}^{(k)}),\\
\mathbf{z}^{(k)} &= \mathbf{x}^{(k)} - \left[\left(1+\frac{1}{2\theta}\right)\mathbf{I} - \frac{1}{2\theta}\mathbf{F}'(\mathbf{x}^{(k)})^{-1}\mathbf{F}'(\mathbf{y}^{(k)})\right]\mathbf{F}'(\mathbf{x}^{(k)})^{-1}\mathbf{F}(\mathbf{x}^{(k)}),\\
\mathbf{x}^{(k+1)} &= \mathbf{z}^{(k)} - \left[\left(1+\frac{1}{\theta}\right)\mathbf{I} - \frac{1}{\theta}\mathbf{F}'(\mathbf{x}^{(k)})^{-1}\mathbf{F}'(\mathbf{y}^{(k)})\right]\mathbf{F}'(\mathbf{x}^{(k)})^{-1}\mathbf{F}(\mathbf{z}^{(k)}),\ \theta \in R\setminus\{0\}.
\end{aligned} \tag{2}$$

This scheme consists of three steps of which first two steps are generalizations of Traub's third order two step scheme ([1], p. 181) for solving scalar equation $f(x) = 0$, where as third step is based on Newton-like scheme. In terms of computational cost, this formula requires two $\mathbf{F}$, two $\mathbf{F}'$ and one matrix inversion per one iteration. However, with the same number of function evaluations, we can develop yet another three-step scheme with improved sixth order of convergence. This is the main motivation of present work.

The paper is organized as follows. Section 2 includes the development of the sixth order method with its analysis of convergence. The computational efficiency is discussed and compared with some well-known existing methods in Section 3. In Section 4, we present various numerical examples to confirm the theoretical results and to compare convergence properties of the proposed method with existing methods. In Section 5, we will compare the proposed method with the existing methods by using basins of attraction in the complex plane. Concluding remarks are given in Section 6.

2. Development of the method

We here consider the following three-step iterative scheme for solving the nonlinear system $\mathbf{F}(\mathbf{x}) = \mathbf{0}$,

$$\begin{aligned}
\mathbf{y}^{(k)} &= \mathbf{x}^{(k)} - \theta\mathbf{F}'(\mathbf{x}^{(k)})^{-1}\mathbf{F}(\mathbf{x}^{(k)}),\\
\mathbf{z}^{(k)} &= \mathbf{x}^{(k)} - \left[\left(1+\frac{1}{2\theta}\right)\mathbf{I} - \frac{1}{2\theta}\mathbf{F}'(\mathbf{x}^{(k)})^{-1}\mathbf{F}'(\mathbf{y}^{(k)})\right]\mathbf{F}'(\mathbf{x}^{(k)})^{-1}\mathbf{F}(\mathbf{x}^{(k)}),\\
\mathbf{x}^{(k+1)} &= \mathbf{z}^{(k)} - \left[\alpha\mathbf{I} + \left(\beta\mathbf{I} + \gamma\mathbf{F}'(\mathbf{x}^{(k)})^{-1}\mathbf{F}'(\mathbf{y}^{(k)})\right)\mathbf{F}'(\mathbf{x}^{(k)})^{-1}\mathbf{F}'(\mathbf{y}^{(k)})\right]\mathbf{F}'(\mathbf{x}^{(k)})^{-1}\mathbf{F}(\mathbf{z}^{(k)}),
\end{aligned} \tag{3}$$

where $\theta \in R\setminus\{0\}$, $\mathbf{I}$ denotes an $n \times n$ identity matrix, and α, β and γ are some parameters. Needless to say that its first two steps are the steps of generalized Traub's third order scheme that also forms part of (2). The third step, which is a Newton-like step, is built in such a way that no additional computations of any Jacobian matrix is required so that it speeds up the convergence order from three to six. In order to explore the convergence property of this three-step scheme, we need the following results of Taylor's expansion on vector functions (see [2]).