Applied Mathematics and Computation 269 (2015) 520–535

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

A novel family of composite Newton–Traub methods for solving systems of nonlinear equations

Janak Raj Sharma [a,*], Rajni Sharma [b], Nitin Kalra [b,c]

[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, Punjab, India

ARTICLE INFO

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

ABSTRACT

We present a family of three-step iterative methods of convergence order five for solving systems of nonlinear equations. The methodology is based on the two-step Traub's method with cubic convergence for solving scalar equations. Computational efficiency of the new methods 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 methods. Moreover, theoretical results concerning order of convergence and computational efficiency are verified in the numerical problems. Stability of the methods is tested by drawing basins of attraction in a two-dimensional polynomial system.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

The construction of fixed point methods for solving nonlinear equations and systems of nonlinear equations is an interesting and challenging task in numerical analysis and many applied scientific branches. 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's method (see [1–3]), which is given as,

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

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 Newton's method requires the evaluations of one $\mathbf{F}$, one $\mathbf{F}'$ and one matrix inversion per iteration. Throughout the paper, we shall use the abbreviation $\phi_i^{(p)}$ to denote an ith iterative function of convergence order p.

* Corresponding author. Tel.: +91 1672 253256; fax: +91 1672 280057.
E-mail addresses: jrshira@yahoo.co.in (J.R. Sharma), rajni_daviet@yahoo.com (R. Sharma), kalranitin17@yahoo.com (N. Kalra).

http://dx.doi.org/10.1016/j.amc.2015.07.092
0096-3003/© 2015 Elsevier Inc. All rights reserved.

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

In quest of more fast algorithms, researchers have also proposed fifth and sixth order methods in [10,13,15,18,19]. The fifth order methods by Grau et al. [10] and Cordero et al. [15,18] require four evaluations namely, two **F** and two **F**′ per iteration. The fifth order method by Cordero et al. [13] requires three **F** and two **F**′. In addition, the fifth order method in [10] requires two matrix inversions, in [13] one matrix inversion and in [15,18] three matrix inversions. One sixth order method by Cordero et al. [15] uses two **F** and two **F**′ while other sixth order method [18] uses three **F** and two **F**′. The sixth order methods, apart from the mentioned evaluations, also require two matrix inversions per one iteration. Recently, Sharma and Gupta [19] proposed a fifth order method requiring two **F**, two **F**′ and two matrix inversions per one iteration.

The main goal of this paper is to develop iterative methods of high computational efficiency, which may assume a high convergence order and low computational cost. To do so, we here propose a one-parameter family of methods with fifth order of convergence by employing the iterative scheme that utilizes the number of function evaluations and inverse operators as minimum as possible. In this way, we attain low computational cost and hence an increased computational efficiency. Moreover, we show that the proposed methods are efficient than existing methods of similar nature in general. The scheme of present contribution consists of three steps of which the first two steps are the generalizations of Traub's third order two-step scheme ([1], p. 181) for solving scalar equation $f(x) = 0$, which is given as

$$\begin{aligned} y_k &= x_k - \theta\frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} &= x_k - \left[\left(1+\frac{1}{2\theta}\right) - \frac{1}{2\theta}\frac{f'(y_k)}{f'(x_k)}\right]\frac{f(x_k)}{f'(x_k)}, \quad \theta \in \mathbb{R}\setminus\{0\}, \end{aligned} \tag{2}$$

whereas the third step is based on Newton-like scheme. In terms of computational cost the proposed technique utilizes two **F**, two **F**′ and only one matrix inversion per iteration.

Our presentation is organized as follows. Some basic definitions relevant to the present work are given in Section 2. In Section 3, the fifth order scheme is developed and its behavior is analyzed. Computational efficiency of the new methods is studied and then compared with some well-known existing methods in Section 4. In Section 5, we present various numerical examples to confirm the theoretical results and to compare convergence properties of the proposed methods with existing methods. In Section 6, we study dynamics of the methods on a two-dimensional polynomial system. Concluding remarks are given in Section 7.

2. Basic definitions

2.1. Order of convergence

Let $\{\mathbf{x}^{(k)}\}_{k\geq 0}$ be a sequence in $\mathbb{R}^n$ which converges to $\mathbf{r}$. Then, convergence is called of order p, $p > 1$, if there exists M, $M > 0$, and k_0 such that

$$||\mathbf{x}^{(k+1)} - \mathbf{r}|| \leqslant M||\mathbf{x}^{(k)} - \mathbf{r}||^p \quad \forall \ k \geqslant k_0$$

or

$$||\mathbf{e}^{(k+1)}|| \leqslant M||\mathbf{e}^{(k)}||^p \quad \forall \ k \geqslant k_0,$$

where $\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - \mathbf{r}$. The convergence is called linear if $p = 1$ and there exists M such that $0 < M < 1$.

2.2. Error equation

Let $\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - \mathbf{r}$ be the error in the kth iteration, we call the relation

$$\mathbf{e}^{(k+1)} = L(\mathbf{e}^{(k+1)})^p + O((\mathbf{e}^{(k)})^{p+1}),$$

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

Download English Version:

https://daneshyari.com/en/article/6420148

Download Persian Version:

https://daneshyari.com/article/6420148

Daneshyari.com