Available online at www.tjnsa.com
J. Nonlinear Sci. Appl. 9 (2016), 4542–4553
Research Article

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

On the construction of three step derivative free four-parametric methods with accelerated order of convergence

Fiza Zafar*, Saima Akram, Nusrat Yasmin, Moin-ud-Din Junjua

Centre for Advanced Studies in Pure and Applied Mathematics, Bahauddin Zakariya University, Multan, Pakistan.

Communicated by E. Savas

Abstract

In this paper, a general procedure to develop some four-parametric with-memory methods to find simple roots of nonlinear equations is proposed. The new methods are improved extensions of with derivative without memory iterative methods. We used four self-accelerating parameters to boost up the convergence order and computational efficiency of the proposed methods without using any additional function evaluations. Numerical examples are presented to support the theoretical results of the methods. We further investigate the dynamics of the methods in the complex plane.

Keywords: Root finding, four-parametric, accelerated order of convergence, derivative free.
2010 MSC: 65H04, 65H05.

1. Introduction

Approximation of simple roots of nonlinear equations with a suitable level of accuracy has a vital importance in various branches of sciences and engineering. Iterative root solvers are extensively used to find simple roots of nonlinear equations $f(x) = 0$ [11, 12]. According to Traub [18], iterative root solvers can be categorized as one-step and multi-step solvers. One-step root solvers are not fruitful due to their less efficiency. However, multi-step root solvers are of great importance because they produce approximations of great accuracy. Multi-step root solvers that use only information from the recent iteration are called without memory root solvers and the root finding methods that use information from the recent and previous

*Corresponding author
Email addresses: fizazafar@gmail.com (Fiza Zafar), saimaakram@bzu.edu.pk (Saima Akram), nusyasmin@yahoo.com (Nusrat Yasmin), moin_junjua@yahoo.com (Moin-ud-Din Junjua)

Received 2016-05-17

iteration are known as with-memory iterative root solvers. Numerous multi-step with and without memory iterative root solvers have been developed by several researchers . Due to the conjecture of Kung and Traub [8], an optimal multi-step without memory root solver can achieve the order of convergence at most 2^{m-1} requiring m functional evaluations per iteration. Among all discussed methods, multi-step with-memory root solvers are of more significance because they significantly improve the convergence speed and computational efficiency of the without memory root solvers without using any additional functional evaluations. Generally with-memory root solvers are constructed by using one or more free parameters or self-accelerators in any optimal derivative free without memory root solver. Newton's interpolating polynomials are used to approximate these self accelerators by using information from the current and previous iterations. The first without memory derivative free method is the well known Steffensen's method [17], which is a variant of Newton's method [11]:

$$x_{m+1} = x_m - \frac{f^2(x_m)}{(f(x_m + f(x_m)) - f(x_m))}, \quad m \geq 0. \tag{1.1}$$

A lot of derivative free without memory optimal root finding methods have been developed in recent years, for example (see [3, 9, 13, 15, 16, 20, 21]). Sometimes it is not possible to improve the convergence order and the efficiency index of without memory methods without additional functional evaluations based on free parameters [3]. Recently, several multi-step with-memory, specially two-steps and three-steps root solvers based on without memory derivative free methods have been developed, which can be seen in [2, 4, 5, 9, 10, 12, 15, 19]. Traub [18], was the first who developed the first with-memory method by modifying the famous Steffensen's iterative scheme (1.1). Its iterative scheme is:

$$\begin{aligned} w_m &= x_m + p_n f(x_m), \\ x_{m+1} &= x_m - \frac{f(x_m)}{f[x_m, w_m]}, \; m \geq 0, \end{aligned} \tag{1.2}$$

where, x_0, p_0 are given, $p_{m+1} = \frac{-1}{N_1'(x_m)}$, $N_1 = f(x_m) + (x - x_m) f[x_m, w_m]$ and p_m is a self-accelerator. The convergence order of the method (1.2) is 2.41421. A large number of derivative free multi-step methods have been developed by using the Traub's method [18] (1.2), in the first step for example (see [9, 13, 20, 21]). In [2], Cordero et al. presented a two-parametric with-memory family of two-steps methods based on a without memory fourth order method of Zheng et al. [21] as follows:

$$\begin{aligned} w_m &= x_m + \beta_m f(x_m), \beta_m = -\frac{1}{N_3'(x_m)}, \gamma_m = -\frac{N_4''(w_m)}{2N_4'(w_m)}, \quad m \geq 1, \\ y_m &= x_m - \frac{f(x_m)}{f[x_m, w_m] + \gamma_m f(w_m)}, \\ x_{m+1} &= y_m - \frac{f(y_m)}{f[x_m, y_m] + (y_m - x_m) f[x_m, w_m, y_m]}, \end{aligned} \tag{1.3}$$

where, x_0, β_0, γ_0 are given. It is proved that the with-memory method (1.3) has R-order of convergence at least 7 and its index of efficiency is 1.913. Lotfi et al. in [10], presented a new tri-parametric with-memory method based on without memory two-step variant of Steffensen's method:

$$\begin{aligned} w_m &= x_m + \beta_m f(x_m), \; m \geq 0, \\ y_m &= x_m - \frac{f(x_m)}{f[x_m, w_m]} \left(1 + \eta_m \frac{f(w_m)}{f[x_m, w_m]}\right), \\ x_{m+1} &= y_m - \frac{f(y_m)}{f[x_m, y_m] + \zeta_m (y_m - x_m)(y_m - w_m)} \left(D_m + (D_m - 1)^4\right), \end{aligned} \tag{1.4}$$

where x_0, β_0, q_0, t_0 are given $D_m = \frac{f[x_m, w_m]}{f[y_m, w_m]}$ and

$$\beta_m = -\frac{1}{N_4'(x_m)}, \quad \eta_m = \frac{N_5''(w_m)}{2N_5'(w_m)}, \; \zeta_m = -\frac{1}{4}\frac{N_6''(w_m)^2}{N_6'(w_m)} + \frac{1}{6} N_6'''(w_m), \quad m \geq 1, \tag{1.5}$$

where $N_4(t)$, $N_5(t)$ and $N_6(t)$ are fourth, fifth and sixth degree interpolating polynomials respectively which pass through best saved points. It is demonstrated that the R-order of convergence of (1.4) is at least 7.77200 and the efficiency index is 7.77200 $\approx$ 1.98082. Cordero and Torregrosa in [3], developed a three-step two-parametric derivative free method by using the approximation $f'(x_m) \approx f[x_m, z_m]$, where $z_m = x_m + \gamma f(x_m)^4$ in the three-step iterative method of Sharma et al. [14], which is given by:

$$\begin{aligned} y_m &= x_m - \frac{f(x_m)}{f[x_m, z_m]}, \quad m \geq 0 \\ u_{m+1} &= y_m - \frac{f(x_m) + \beta f(y_m)}{f(x_m) + (\beta - 2) f(y_m)} \frac{f(y_m)}{f[x_m, z_m]}, \\ x_{m+1} &= x_m - \frac{(P + Q + R) f(x_m)}{P f[u_m, x_m] + Q f[z_m, x_m] + R f[y_m, x_m]}, \end{aligned} \tag{1.6}$$

where $P = (x_m - y_m) f(x_m) f(y_m)$, $Q = (y_m - u_m) f(y_m) f(u_m)$ and $R = (u_m - x_m) f(u_m) f(x_m)$ with error equation

$$e_{m+1} = c_2^2((1 + 2\beta)c_2^2 - c_3)(c_2^3 - 2c_3c_2 + c_4)e_m^8 + O(e_m^9). \tag{1.7}$$

Obviously the error equation (1.7) cannot allow to improve the convergence order of (1.6) by varying the free parameters γ and β. Thus (1.6) cannot be extended to with-memory root solver.

Since, commonly only the derivative free methods can be extended to with-memory, in this work, firstly, we design a procedure to develop some new optimal without memory derivative free root solvers which can be extended to with-memory root solvers. Secondly, we extend a proposed without memory method to with-memory by approximating the self-accelerating parameters which arise in the error equations of without memory methods. This acceleration is based on the information from current and the previous iterative step. Newton's interpolating polynomials are used to calculate the parameters in such a way that the convergence order and efficiency of the without memory root solvers is significantly increased from 8 to 15.5156. The important advantage of these with-memory root solvers is that the convergence speed is accelerated without additional functional evaluations. Let $f(x)$ be a function defined on an interval D, where D is the smallest interval containing $m + 1$ distinct nodes $x_0, x_1, \ldots, x_m$. Then, the divided difference $f[x_0, x_1, \ldots, x_m]$ with mth-order is defined by $f[x_0] = f(x_0)$, and $f[x_0, x_1, \ldots, x_m] = \frac{f[x_1, x_2, \ldots, x_m] - f[x_0, x_1, \ldots, x_{m-1}]}{x_m - x_0}$.

If ρ is the convergence order of an iterative method and η is the total number of functional evaluations per iterative step then the index $E = \rho^{1/\eta}$ is used to measure the efficiency of an iterative method and thus called efficiency index commonly. The rest of this work is organized as follows. Section 2 is devoted to construct three-step derivative free without memory methods extendable to with-memory methods involving four free parameters. Section 3 provides the four-parametric with-memory methods and their R-order of convergence by extending a without memory method given in Section 2. To support the given theory, some numerical results are presented in Section 4. The dynamical analysis of the iterative methods is given in Section 5 and finally, Section 6 provides the conclusions of this paper.

2. Procedure to Construct The Derivative Free Root Solvers Extendable to With-memory

The main aim of this section is to construct three-step derivative free without memory methods extendable to with-memory based on any optimal two-step derivative-involved method. We use weight function approach at the second step of any optimal two-step with-derivative method followed by Newton's method in the third step and the first derivative arising at each step is calculated using suitable approximations. In this way the convergence order of the proposed methods is preserved and can be increased by varying the involved free parameters. This increase of convergence will be discussed in the next section by with-memorization of the free parameters. We consider the following general optimal two-step method involving first derivative of the function:

$$\begin{aligned} y_m &= \phi_1(x_m), \quad m \geq 0, \\ z_m &= \phi_2(x_m, y_m), \end{aligned} \tag{2.1}$$

where $\phi_1$and ϕ_2 are real functions such that ϕ_1 is the well known Newton's scheme which involves the values

$f(x_m)$ and $f'(x_m)$ providing the quadratic convergence of the sequence x_m and ϕ_2 requires the previously computed values $f(x_m)$, $f'(x_m)$ and the new value $f(y_m)$ to give the fourth order convergence. Based on any two-step with-derivative optimal method like (2.1), we design a general three-step without-derivative method extendable to with-memory as follows. We use a weight function $S(u_m)$ in the second step of (2.1) and add another real function ϕ_3, (Newton's method [11]) in the third step. The values of first derivative involved in ϕ_1, ϕ_2 and ϕ_3 are approximated by $f[x_m, w_m] + qf(w_m)$, $f[y_m, w_m] + qf(w_m) + s(y_m - w_m)(y_m - x_m)$ and $f[y_m, z_m] + f[z_m, y_m, x_m](z_m - y_m) + f[z_m, y_m, x_m, w_m](z_m - y_m)(z_m - x_m) + t(z_m - w_m)(z_m - y_m)(z_m - x_m)$ respectively, where $u_m = \frac{f(y_m)}{f(x_m)}$ and the scalars p, q, s and t are freely chosen parameters. Hence, we obtain the following general three-step method having no derivative:

$$\begin{aligned} y_m &= \phi_1(x_m, w_m), w_m = x_m + pf(x_m), \quad m \geq 0, \\ z_m &= \phi_2(x_m, w_m, y_m), \\ x_{m+1} &= \phi_3(x_m, w_m, y_m, z_m), \end{aligned} \tag{2.2}$$

were ϕ_1 is the famous Steffensen's method [17] employing the values $f(x_m)$ and $f(w_m)$, ϕ_2 is selected such that it requires the previously calculated values $f(x_m)$, $f(w_m)$ and the new value $f(y_m)$ preserving the fourth order convergence and ϕ_3 is chosen such that it uses the already computed values $f(x_m)$, $f(w_m)$, $f(y_m)$ and the new value $f(z_m)$ to provide the optimal eighth order convergence.

For instance, consider the optimal fourth order two-step King's method [7],

$$\begin{aligned} y_m &= x_m - \frac{f(x_m)}{f'(x_m)}, \quad m \geq 0, \\ z_m &= y_m - \frac{f(y_m)}{f'(x_m)} \frac{f(x_m) + \lambda f(y_m)}{f(x_m) + (\lambda - 2) f(y_m)}, \quad \lambda \in R. \end{aligned} \tag{2.3}$$

Applying the above procedure (2.2) we propose the following without memory optimal eighth order modification of (2.3) having no derivative:

$$\begin{aligned} y_m &= x_m - \frac{f(x_m)}{f[x_m, w_m] + qf(w_m)}, \quad w_m = x_m + pf(x_m), \quad m \geq 0, \\ z_m &= y_m - S(u_m) \frac{f(x_m) + \lambda f(y_m)}{f(x_m) + (\lambda - 2) f(y_m)} \times \frac{f(y_m)}{f[y_m, w_m] + qf(w_m) + s(y_m - w_m)(y_m - x_m)}, \\ x_{m+1} &= z_m - \frac{f(z_m)}{Q_m}, \end{aligned} \tag{2.4}$$

where $u_m = \frac{f(y_m)}{f(x_m)}, \lambda \in \mathbf{R}, Q_m = f[y_m, z_m] + f[z_m, y_m, x_m](z_m - y_m) + f[z_m, y_m, x_m, w_m](z_m - y_m)(z_m - x_m) + t(z_m - w_m)(z_m - y_m)(z_m - x_m)$, p, q, s and t are free parameters. Obviously, we can state the following theorem by imposing the conditions on $S(u_m)$ to achieve optimal order of convergence for (2.4).

Theorem 2.1. *Let $\omega \in I$ be a simple root of a sufficiently differentiable function $f : I \subseteq \mathbf{R} \to \mathbf{R}$, where $I \subseteq \mathbf{R}$ is an open set and the starting point x_0 is close enough to γ. Then, the scheme (2.4) is eighth order convergent if $S(0) = 1, S'(0) = -1$ and $|S''(0)| < \infty$, and possesses the following error equation:*

$$\begin{aligned} e_{m+1} =& \frac{1}{4c_1^2}(c_2 + q)^2(1 + pc_1)^4 \Big(-4\lambda q^2 pc_1^2 + 2q^2 pc_1^2 + S''(0)q^2 pc_1^2 + 4qc_2 pc_1^2 - 8\lambda qc_2 pc_1^2 \\ &+ 2S''(0)qc_2 pc_1^2 - 4\lambda c_2^2 pc_1^2 + 2c_2^2 pc_1^2 + S''(0)c_2^2 pc_1^2 - 4c_1\lambda q^2 + 2c_1 q^2 + c_1 S''(0)q^2 - 8c_1\lambda qc_2 \\ &+ 2c_1 S''(0)qc_2 - 4c_1\lambda c_2^2 + 2c_1 c_3 - 2c_1 c_2^2 + c_1 S''(0)c_2^2 - 2s\Big)\Big(2r - 2c_1 c_4 - 2sc_2 + 2c_1 c_1 c_3 \\ &- 2c_2^3 c_1 + 2q^2 c_2 c_1 - 4c_1\lambda c_2^3 - 8c_1\lambda qc_2^2 - 4c_2 c_1 \lambda q^2 + c_1 S''(0)c_2^3 + 2c_1 S''(0)qc_2^2 + c_2 c_1 S''(0)q^2 \\ &+ 2c_2^3 pc_1^2 + 4qc_2^2 pc_1^2 + 2q^2 c_1 pc_1^2 - 4\lambda c_2^3 pc_1^2 - 8\lambda qc_1^2 pc_1^2 - 4c_1\lambda q^2 pc_1^2 + S''(0)c_2^3 pc_1^2 \\ &+ 2S''(0)qc_2^2 pc_1^2 + c_2 S''(0)q^2 pc_1^2\Big) e_m^8 + O(e_m^9), \end{aligned} \tag{2.5}$$

where, $c_k = \frac{f^{(k)}(\omega)}{k!f'(\omega)}, k \geq 2.$

Proof. The proof would be similar to those already considered in [9, 13, 15, 16, 20], using the Taylor expansions of the function f in the mth iterative step. Hence, it is omitted. □

In the same fashion, we, now, extend the optimal fourth order iterative method of Kung and Traub [8], which is given by:

$$\begin{aligned} y_m &= x_m - \frac{f(x_m)}{f'(x_m)}, \quad m \geq 0, \\ z_m &= y_m - \frac{f(y_m)}{f'(x_m)} \frac{1}{(1 - f(y_m)/f(x_m))^2}. \end{aligned} \tag{2.6}$$

We obtain the following three-step without memory method based on (2.6):

$$\begin{aligned} y_m &= x_m - \frac{f(x_m)}{f[x_m, w_m] + qf(w_m)}, w_m = x_m + pf(x_m), \quad m \geq 0, \\ z_m &= y_m - S(u_m)\frac{1}{(1 - f(y_m)/f(x_m))^2} \times \frac{f(y_m)}{f[y_m, w_m] + qf(w_m) + s(y_m - w_m)(y_m - x_m)}, \\ x_{m+1} &= z_m - \frac{f(z_m)}{Q_m}, \end{aligned} \tag{2.7}$$

where, p, q, s, t, u_m and Q_m are same as given in (2.4). The error equation of (2.7) is:

$$\begin{aligned} e_{m+1} =& \frac{1}{c_1^2}(c_2 + q)^2(1 + pc_1)^4 \Big(pq^2c_1^2 + 2qc_2pc_1^2 + pc_2^2c_1^2 + c_1q^2 \\ &+ 4qc_1c_2 - c_1c_3 + 3c_1c_2^2 + s\Big)\Big(- t + c_1c_4 + sc_2 - c_1c_2c_3 + 3c_2^3c_1 \\ &+ 4qc_2^2c_1 + q^2c_2c_1 + c_2^3pc_1^2 + 2qc_2^2pc_1^2 + q^2c_2pc_1^2\Big)e_m^8 + O(e_m^9). \end{aligned} \tag{2.8}$$

In [1], Chun suggested the following fourth order two-step derivative involved method:

$$\begin{aligned} y_m &= x_m - \frac{f(x_m)}{f'(x_m)}, \quad m \geq 0, \\ z_m &= y_m - \frac{f(x_m)^3}{f(x_m)^3 - 2f(x_m)^2f(y_m) - f(x_m)f(y_m)^2 - \frac{1}{2}f(y_m)^3} \frac{f(y_m)}{f'(x_m)}. \end{aligned} \tag{2.9}$$

We suggest the following three-step without memory derivative free method based on (2.9):

$$\begin{aligned} y_m =& x_m - \frac{f(x_m)}{f[x_m, w_m] + qf(w_m)}, \quad w_m = x_m + pf(x_m), \\ z_m =& y_m - S(u_m)\frac{f(x_m)^3}{f(x_m)^3 - 2f(x_m)^2f(y_m) - f(x_m)f(y_m)^2 - \frac{1}{2}f(y_m)^3} \times \\ & \frac{f(y_m)}{f[y_m, w_m] + qf(w_m) + s(y_m - w_m)(y_m - x_m)}, \\ x_{m+1} =& z_m - \frac{f(z_m)}{Q_m}, \end{aligned} \tag{2.10}$$

where, p, q, s, t, u_m and Q_m are same as given in (2.4). The error equation of (2.10) is:

$$\begin{aligned} e_{m+1} =& \frac{1}{c_1^2}(c_2 + q)^2(1 + pc_1)^4 \Big(2q^2pc_1^2 + 4qc_2pc_1^2 + 2c_2^2pc_1^2 + 2c_1q^2 \\ &+ 2c_1qc_2 + c_1c_3 - s\Big)\Big(t - c_1c_4 - sc_2 + c_1c_2c_3 + 2qc_2^2c_1 + 2q^2c_2c_1 \\ &+ 2c_2^3pc_1^2 + 4qc_2^2pc_1^2 + 2q^2c_2pc_1^2\Big)e_m^8 + O(e_m^9). \end{aligned} \tag{2.11}$$

3. Four-Parametric With-Memory Root Solvers

In this section, we give our main contribution by extracting a with-memory method from our newly suggested without memory method (2.4). For this, we approximate the involved parameters in such a way that the local order of convergence is increased. It can be noted that the coefficient of e_m^8 in (2.5) disappears if $p = \frac{-1}{c_1}, q = -c_2, s = c_1 c_3$ and $t = c_1 c_4$, where $c_1 = f'(\omega)$ and $c_k = \frac{f^{(k)}(\omega)}{k! f'(\omega)}, k \geq 2$. Therefore, to construct with-memory method the free parameters p, q, s and t are calculated by the formulas $p_n = \frac{-1}{\overline{f'(\omega)}}, q_n = -\frac{\overline{f''(\omega)}}{2\overline{f'(\omega)}}, s_n = \frac{\overline{f'''(\omega)}}{6}$ and $t_n = \frac{\overline{f^{iv}(\omega)}}{24}$ respectively, for $n = 1, 2, \cdots$. Where $\overline{f'}, \overline{f''}(\omega), \overline{f'''}(\omega)$ and $\overline{f^{iv}}(\omega)$ are the best approximations to $f'(\omega), f''(\omega), f'''(\omega)$ and $f^{iv}(\omega)$, since exact value of simple root is not known and consequently the derivatives of the function cannot be computed. The approximations $\overline{f'}, \overline{f''}(\omega), \overline{f'''}(\omega)$ and $\overline{f^{iv}}(\omega)$ are computed by Newton's interpolating polynomials of appropriate degrees respectively. Hence, we replace the free parameters p, q, s and t in (2.4) with self-accelerators p_n, q_n, s_n and t_n and present the following with-memory root solver:

$$
\begin{aligned}
y_m =& x_m - \frac{f(x_m)}{f[x_m, w_m] + q_n f(w_m)}, \quad w_m = x_m + p_n f(x_m), \quad m \geq 2, \\
z_m =& y_m - S(u_m) \frac{f(x_m) + \lambda f(y_m)}{f(x_m) + (\lambda - 2) f(y_m)} \times \frac{f(y_m)}{f[y_m, w_m] + q_n f(w_m) + s_n (y_m - w_m)(y_m - x_m)}, \\
x_{m+1} =& z_m - \frac{f(z_m)}{Q_m},
\end{aligned} \tag{3.1}
$$

where u_m and Q_m are same as given in (2.4) and

$$
p_n = \frac{-1}{N_4'(x_m)}, \; q_n = -\frac{N_5''(w_m)}{2N_5'(w_m)}, \; s_n = \frac{N_6'''(y_m)}{6}, \; t_n = \frac{N_7^{iv}(z_m)}{24}. \tag{3.2}
$$

The self-accelerators are calculated recursively using available information in the current and previous iterations. Hence, we use Newton's interpolation method to approximate the derivatives of f, where $N_4(x_m), N_5(w_m), N_6(y_m)$ and $N_7(z_m)$ are Newton's interpolation polynomials of degree four, five, six and seven respectively defined by:

$$
\begin{aligned}
N_4(t) &= N_4(t; x_m, z_{m-1}, y_{m-1}, w_{m-1}, z_{m-2}), \\
N_5(t) &= N_5(t; w_m, x_m, z_{m-1}, y_{m-1}, w_{m-1}, z_{m-2}), \\
N_6(t) &= N_6(t; y_m, w_m, x_m, z_{m-1}, y_{m-1}, w_{m-1}, z_{m-2}), \\
N_7(t) &= N_7(t; z_m, y_m, w_m, x_m, z_{m-1}, y_{m-1}, w_{m-1}, z_{m-2}),
\end{aligned}
$$

for any $m \geq 2$. We will now prove that the with-memory method (3.1) has convergence order 15.51560 by applying the Herzberger's matrix method [6] provided that self-accelerators given in (3.2) are used.

Theorem 3.1. *Let x_0 be an initial approximation sufficiently close to the root ω of the function $f(x)$. If the parameters p_n, q_n, s_n and t_n are recursively computed by the forms given in* (3.2)*, then the convergence R-order of* (3.1) *is at least* 15.51560 *with the efficiency index* $15.51560^{\frac{1}{4}} \approx 1.98468$.

Proof. To determine the R-order of convergence, we use the Herzberger's matrix method. It can be seen that the spectral radius of a matrix $A^{(s)} = (h_{ij})(1 \leq i, j \leq s)$ associated with a with-memory one-step $s-$point method $x_k = \phi(x_{k-1}, x_{k-2}, \ldots, x_{k-s})$ is the lower bound of the its order of convergence. The elements of this matrix are given by:

$$
\begin{aligned}
h_{1,j} &= \text{number of function evaluations required at point } x_{k-j,}\ j = 1, 2, \ldots, s, \\
h_{i,i-1} &= 1 \text{ for } i = 2, 3, \ldots, s, \\
h_{i,j} &= 0. \text{ otherwise.}
\end{aligned}
$$

On the other hand, the spectral radius of product of the matrices $A_1, A_2, \ldots, A_s$, is the lower bound of order of an $s-$step method $\phi = \phi_1 \circ \phi_2 \circ \cdots \circ \phi_s$, where the matrices A_r correspond to the iteration steps

$\phi_r, 1 \leq r \leq s$. From the relations (3.1) and (3.2), we construct the corresponding matrices as follows:

$$x_{m+1} = \phi_1\left(z_m, y_m, w_m, x_m, z_{m-1}, y_{m-1}, w_{m-1}, x_{m-1}\right)$$
$$\to A_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$
$$z_m = \phi_2\left(y_m, w_m, x_m, z_{m-1}, y_{m-1}, w_{m-1}, x_{m-1}, z_{m-2}\right)$$
$$\to A_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$
$$y_m = \phi_3\left(w_m, x_m, z_{m-1}, y_{m-1}, w_{m-1}, x_{m-1}, z_{m-2}, y_{m-2}\right)$$
$$\to A_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$
$$w_m = \phi_4\left(x_m, z_{m-1}, y_{m-1}, w_{m-1}, x_{m-1}, z_{m-2}, y_{m-2}, w_{m-2}\right)$$
$$\to A_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Hence, we obtain

$$A^{(4)} = A_1 \cdot A_2 \cdot A_3 \cdot A_4 = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 0 & 0 & 0 \\ 4 & 4 & 4 & 4 & 4 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The eigen values of $A^{(4)}$ are $\{0,0,0,0,0,0,15.51560977,-.515609770\}$. Therefore, the lower bound of the R-order of convergence of the method (3.1) is the spectral radius of matrix $A^{(4)}$, which is $\rho(A^{(4)}) = 15.51560$. □

Remark 3.2. It can be seen that, we have obtained the R-order 15.51560, which gives the highest efficiency index 1.98468 for the presented with-memory method (3.1).

To develop a special case of (3.1), we choose $S(u_m)$ in such a way that the following conditions are satisfied:

$$S(0) = 1, \quad S'(0) = -1, \quad S''(0) < \infty.$$

For example we choose $S(u_m) = 1 - u_m$ in (3.1), where, $u_m = \frac{f(y_m)}{f(x_m)}$. Now, we can define a particular with memory method as follows:

FWM:

$$\begin{aligned} y_m &= x_m - \frac{f(x_m)}{f[x_m, w_m] + q_n f(w_m)}, w_m = x_m + p_n f(x_m), m \geq 2, \\ z_m &= y_m - (1 - u_m) \frac{f(x_m) + \lambda f(y_m)}{f(x_m) + (\lambda - 2) f(y_m)} \\ &\quad \times \frac{f(y_m)}{f[y_m, w_m] + q_n f(w_m) + s_n (y_m - w_m)(y_m - x_m)}, \\ x_{m+1} &= z_m - \frac{f(z_m)}{Q_m}, \end{aligned} \tag{3.3}$$

where Q_m is same as given in (2.4) and

$$p_n = \frac{-1}{N_4'(x_m)}, \ q_n = -\frac{N_5''(w_m)}{2N_5'(w_m)}, \ s_n = \frac{N_6'''(y_m)}{6}, \ t_n = \frac{N_7^{iv}(z_m)}{24}. \tag{3.4}$$

4. Numerical Results

In this section, numerical examples are taken from [10] to test the proposed with-memory root solver (3.3) (FWM) in comparison with the with-memory family of methods of Kung and Traub [8] and with-memory method of Lotfi et al. [10] (1.4). Kung and Traub [8] presented the iterative method $\Phi_r(f)$ $(r = -1, 0, \ldots, n)$ as follows:

$$\begin{aligned} y_{k,0} &= \Phi_0(f)(x_k) = x_k, \ y_{k,-1} = \Phi_{-1}(f)(x_k) = x_k + \gamma_k f(x_k), \quad k \geq 0, \\ y_{k,r} &= \Phi_r(f)(x) = P_r(0), \ r = 1, \ldots, n, \quad \text{for } n > 0, \\ x_{k+1} &= y_{k,n} = \Phi_n(f)(x_k), \end{aligned} \tag{4.1}$$

where x_0 is an initial approximation and n is an arbitrary natural number. The free parameter, γ_k is calculated by Newton's interpolation polynomial of third degree and $P_r(t)$ is the rth degree inverse interpolating polynomial such that $P_r(f(y_{k,m})) = y_{k,m}, \ m = -1, 0, \ldots, r-1$. All numerical computations are performed using the programming package Maple16 with multiple-precision arithmetic by applying 3000 fixed floating point arithmetic. For the comparison, we have taken the following test functions:

$$\begin{aligned} f_1(x) &= e^{x^2-3x} \sin(x) + \log(x^2+1), \ x_0 = 0.35, \ \omega = 0, \\ f_2(x) &= e^{x^2+x\cos(x)-1} \sin(\pi x) + x \log(x \sin(x) + 1), \ x_0 = 0.6, \ \omega = 0, \\ f_3(x) &= e^{-x^2+x+2} + \sin(\pi x) e^{x^2+x\cos(x)-1} + 1, \ x_0 = 1.3, \ \omega \approx 1.55031 \cdots. \end{aligned}$$

Tables 1–3 display the behavior of the approximate values for the test functions, where $A(-d)$ denotes $A \times 10^{-d}$. For all the compared with-memory methods, we have considered $p_0 = s_0 = t_0 = 0.01, q_0 = 0.1$. From the obtained results it is evident that the proposed with-memory method (3.1) has very fast convergence behavior than the with-memory method of Kung and Traub [8] and Lotfi et al. [10] (1.4).

Table 1: Results of With-memory Method (1.4), $\beta_0 = \eta_0 = \zeta_0 = -0.1$

Functions	$\|f(x_1)\|$	$\|f(x_2)\|$	$\|f(x_3)\|$	$\|f(x_4)\|$
$f_1(x)$	0.2(−3)	0.32(−21)	0.11(−155)	0.44(−1196)
$f_2(x)$	0.58(−4)	0.24(−27)	0.12(−206)	0.98(−1592)
$f_3(x)$	0.45(−2)	0.41(−17)	0.10(−129)	0.10(−1007)

Table 2: Results of With-Memory Method for (4.1), $n = 3$ $\gamma_0 = 0.1$

Functions	$\|f(x_1)\|$	$\|f(x_2)\|$	$\|f(x_3)\|$	$\|f(x_4)\|$
$f_1(x)$	0.45(−6)	0.16(−51)	0.87(−435)	0.95(−3683)
$f_2(x)$	0.37(−1)	0.67(−13)	0.13(−110)	0.23(−939)
$f_3(x)$	0.79(−3)	0.16(−33)	0.21(−291)	0.34(−2477)

Table 3: Results of Proposed With-memory Method (FWM)

Functions	$\|f(x_1)\|$	$\|f(x_2)\|$	$\|f(x_3)\|$	$\|f(x_4)\|$
$f_1(x)$	0.18(−7)	0.15(−58)	0.45(−905)	0.53(−9180)
$f_2(x)$	0.11(−2)	0.14(−22)	0.69(−356)	0.65(−5434)
$f_3(x)$	0.18(−4)	0.13(−43)	0.58(−702)	0.20(−3998)

5. Dynamical Behavior

To visualize the stability of iterative methods, we analyze the dynamical properties of rational functions associated to these methods. We present the comparison of the dynamical planes of discussed iterative methods in this section. For this, we associate a rational function obtained by applying the proposed with-memory method (3.3) and Kung-Traub Method with-memory [8] (4.1) to a complex function in the complex plane. Two different approaches of dynamical planes are obtained on Matlab R2013a software as follows: By taking a rectangle $[-2, 2] \times [-2, 2]$ of the complex plane, we define a mesh of 1000×1000 initial approximations. The starting point is in the basin of attraction of a root to which the sequence of the iterative method converges with an error approximation lower than 10^{-5} and at most 30 iterations. In the first technique this initial point is assigned with a specific color which is already selected for the corresponding root. If the sequence of the iterative method converges in less number of iterations then the color will be more intense and if it is not converging to any of the roots after maximum number of 30 iterations, then initial point is marked with dark blue color. For the second technique, maximum number of iterations are 25 with an error estimation lower than 10^{-5} and each initial guess is assigned with a color depending upon to the number of iterations for the iterative method to converge to any of the root of the given function. In this technique we use colormap 'Hot'. The color of the initial point will be more intense if the sequence of the iterative method converges in less number of iterations and if it is not converging to any of the roots after maximum number of 25 iterations, then initial point is assigned with black color. The proposed with-memory method (2.3) and Kung-Traub Method with-memory [8] (4.1) are applied to the following complex functions: $p_1(z) = z^3 - 1$, with roots 1.0, $-0.5000 + 0.86605I$, $-0.5000-0.86605I$, $p_2(z) = z^5-1$, with roots 1.0, $0.3090+0.95105I$, $-0.8090+0.58778I$, $-0.8090-0.58778I$, $0.30902 - 0.95105I$ and $p_3(z) = z^6 - \frac{1}{2}z^5 + \frac{11(i+1)}{4}z^4 - \frac{3i+19}{4}z^3 + \frac{5i+11}{4}z^2 + \frac{i-11}{4}z + \frac{3}{2} - 3i$, with the solutions $-1.0068 + 2.0047i, 0.0281 + 0.9963i, 0.0279 - 1.5225i, 1.0235 - 0.9556i, 0.9557 - 0.0105i, -0.5284 - 0.5125i$.

Dynamical planes of the with-memory methods (3.3) and (4.1) for $n = 3$ and $\beta_0 = 0.01$ applied to the functions $p_1(z), p_2(z), p_3(z)$ are shown in the Figures 1–6. Two types of attraction basins are given in all the figures. Color maps for both types are provided with each figure which show the root to which an initial guess converges and the number of iterations in which the convergence occurs. From the appearance of darker region shown in the Figures 1–6, we conclude that, the iterative method (3.3) consumes less number of iterations in comparison with (4.1). As a conclusion, the proposed with-memory method (3.3) is highly efficient than the with-memory family of Kung and Traub (4.1) (KT).

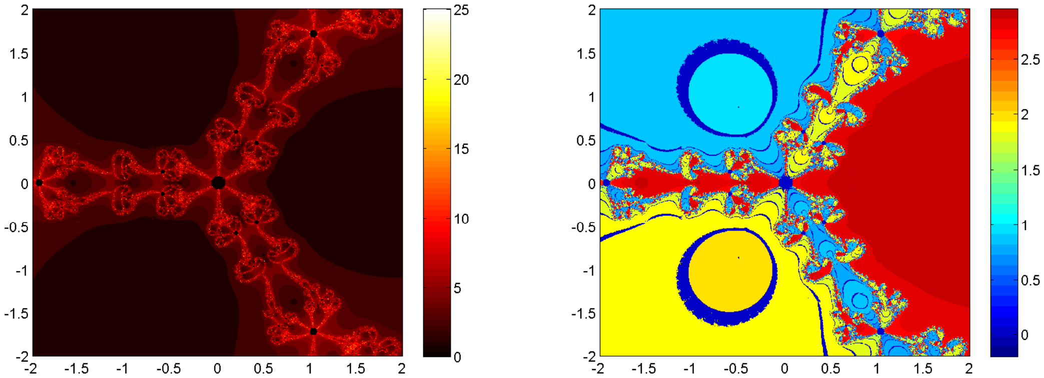

Figure 1: Dynamical Planes for (4.1) for $n = 3$ and $\beta_0 = 0.01$ on p_1.

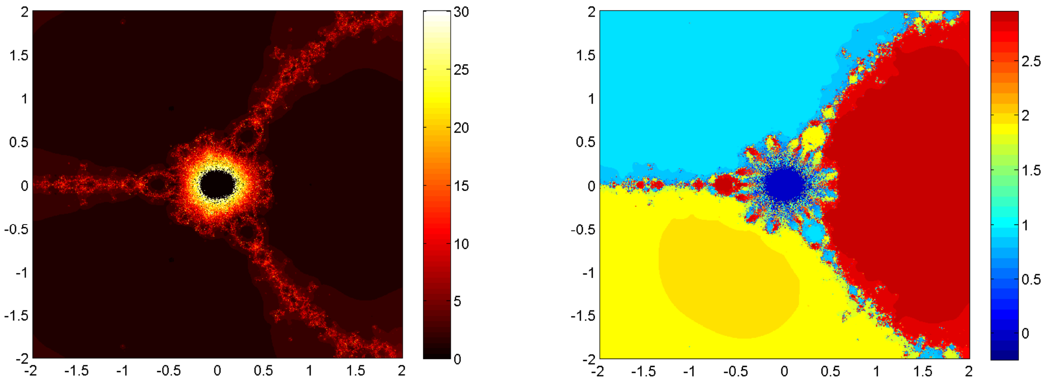

Figure 2: Dynamical Planes for (3.3) for $p_0 = -0.01, q_0 = 0.01, s_0 = 0.01$, and $t_0 = 0.01$ on p_1.

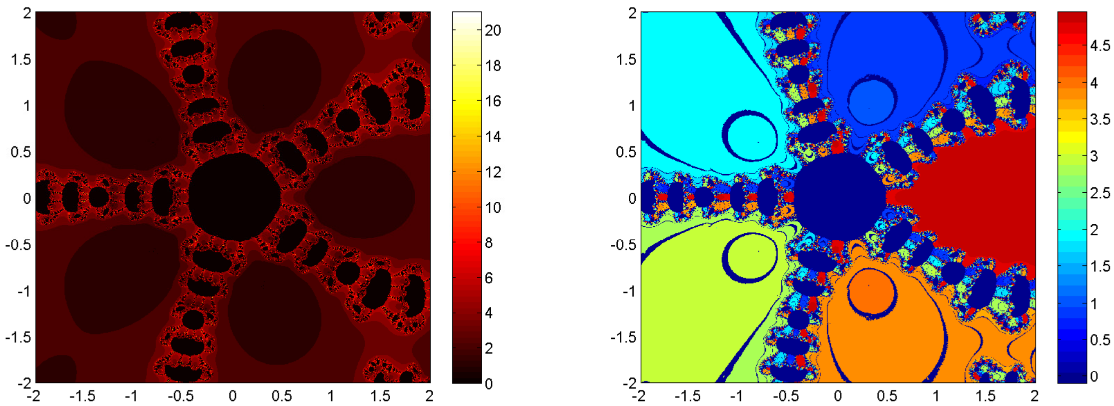

Figure 3: Dynamical Planes for (4.1) for $n = 3$ and $\beta_0 = 0.01$ on p_2.

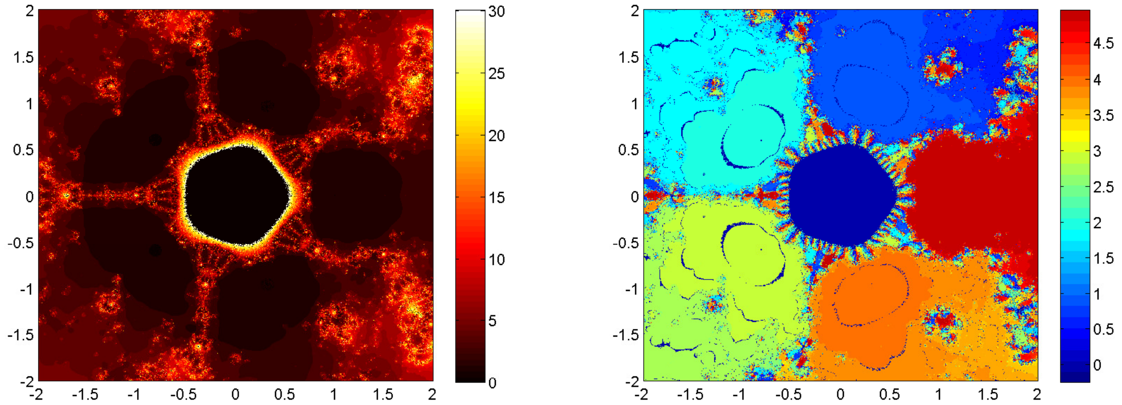

Figure 4: Dynamical Planes for (3.3) for $p_0 = -0.01, q_0 = 0.01, s_0 = 0.01$, and $t_0 = 0.01$ on p_2.

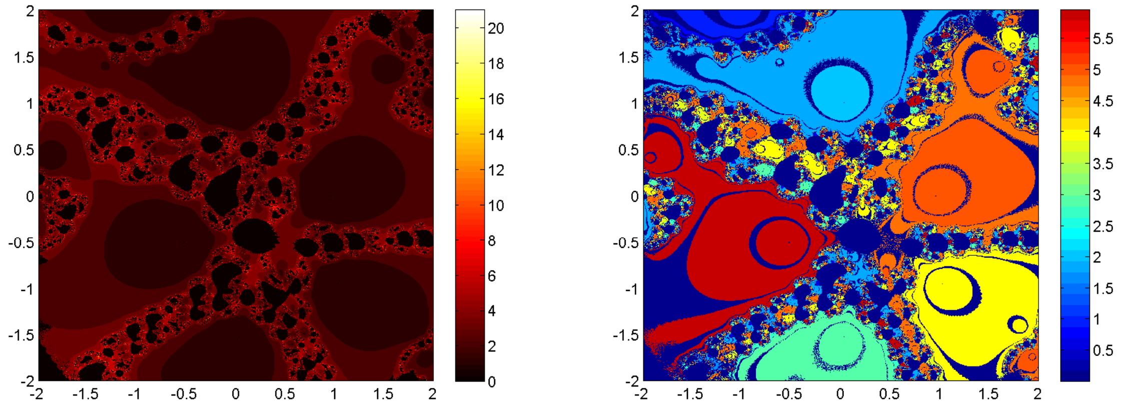

Figure 5: Dynamical Planes for (4.1) for $n = 3$ and $\beta_0 = 0.01$ on p_3.

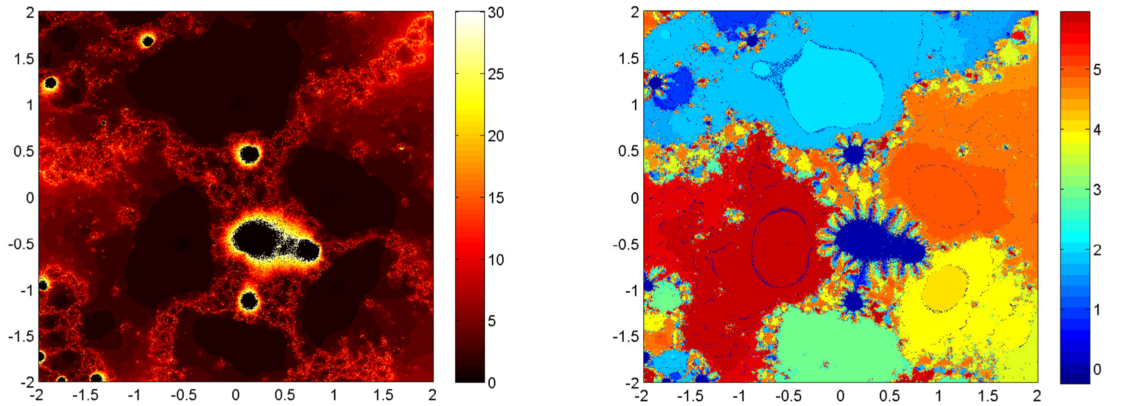

Figure 6: Dynamical Planes for (3.3) for $p_0 = -0.01, q_0 = 0.01, s_0 = 0.01$, and $t_0 = 0.01$ on p_3.

6. Conclusions

In this paper, we have designed a procedure to develop without memory optimal iterative methods extendable to with-memory methods. Some examples are given for this design. We also extend a proposed three-step without memory optimal eighth-order method to an efficient with-memory method. To achieve an efficient method with low computational load, we proposed a with-memory method including three steps involving four accelerators. It has been shown that the new method in Section 3 possesses very high computational efficiency index $15.51560^{\frac{1}{4}} \approx 1.98468$ which is even higher than many of the developed with-memory methods in the literature, e.g. $7^{\frac{1}{3}} \approx 1.913$, of two-step with-memory method using two accelerators discussed in [2]. Finally numerical results and dynamical behavior are presented which illustrate that the proposed with-memory iterative method have good enough behavior for finding roots of nonlinear functions.

References

[1] C. Chun, *Some variants of King's fourth-order family of methods for nonlinear equations*, Appl. Math. Comput., **190** (2007), 57–62. 2

[2] A. Cordero, T. Lotfi, P. Bakhtiari, J. R. Torregrosa, *An efficient two-parametric family with-memory for nonlinear equations*, Numer. Algorithms, **68** (2015), 323–335. 1, 1, 6

[3] A. Cordero, J. R. Torregrosa, *Low-complexity root-finding iteration functions with no derivatives of any order of convergence*, J. Comput. Appl. Math., **275** (2015), 502–515. 1, 1

[4] J. Džunić, M. S. Petković, *On generalized multipoint root-solvers with memory*, J. Comput. Appl. Math., **236** (2012), 2909–2920. 1

[5] J. Džunić, M. S. Petković, *On generalized biparametric multipoint root finding methods with-memory*, J. Comput. Appl. Math., **255** (2014), 362–375. 1

[6] J. Herzberger, *Über Matrixdarstellungen für Iterationsverfahren bei nichtlinearen Gleichungen (German)*, Computing (Arch. Elektron. Rechnen), **12** (1974), 215–222. 3

[7] R. F. King, *A family of fourth-order methods for nonlinear equations*, SIAM J. Numer. Anal., **10** (1973), 876–879. 2

[8] H. T. Kung, J. F. Traub, *Optimal order of one-point and multipoint iteration*, J. Assoc. Comput. Mach., **21** (1974), 643–651. 1, 2, 4, 4, 5

[9] T. Lotfi, K. Mahdiani, Z. Noori, F. Khaksar Haghani, S. Shateyi, *On a new three-step class of methods and its acceleration for nonlinear equations*, Sci. World J., **2014** (2014), 9 pages. 1, 1, 2

[10] T. Lotfi, F. Soleymani, M. Ghorbanzadeh, P. Assari, *On the construction of some tri-parametric iterative methods with-memory*, Numer. Algorithms, **70** (2015), 835–845. 1, 1, 4, 4

[11] J. M. Ortega, W. C. Rheiboldt, *Iterative solution of nonlinear equations in several variables*, Academic Press, New York-London, (1970). 1, 2

[12] M. S. Petković, J. Džunić, L. D. Petković, *A family of two-point methods with-memory for solving nonlinear equations*, Appl. Anal. Discrete Math., **5** (2011), 298–317. 1, 1

[13] M. Sharifi, S. Karimi Vanani, F. Khaksar Haghani, M. Arab, S. Shateyi, *On a new iterative scheme without memory with optimal eighth Order*, Sci. World J., **2014** (2014), 6 pages. 1, 1, 2

[14] J. R. Sharma, R. K. Guha, P. Gupta, *Improved King's methods with optimal order of convergence based on rational approximations*, Appl. Math. Lett., **26** (2013), 473–480. 1

[15] J. R. Sharma, P. Gupta, *On some highly efficient derivative free methods with and without memory for solving nonlinear equations*, Int. J. Comput. Methods, **12** (2015), 28 pages. 1, 2

[16] F. Soleymani, S. Karimi Vanani, M. Jamali Paghaleh, *A class of three-step derivative-free root solvers with optimal convergence Order*, J. Appl. Math., **2012** (2012), 15 pages. 1, 2

[17] J. F. Steffensen, *Remarks on iteration*, Scand. Actuar. J., **1933** (1933), 64–72. 1, 2

[18] J. F. Traub, *Iterative methods for the solution of equations*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, (1964). 1, 1, 1

[19] X. Wang, T. Zhang, *Some Newton-type iterative methods with and without memory for solving nonlinear equations*, Int. J. Comput. Methods, **11** (2014), 20 pages. 1

[20] F. Zafar, N. Yasmin, S. Akram, M. D. Junjua, *A general class of derivative free optimal root finding methods based on rational interpolation*, Sci. World J., **2015** (2015), 12 pages. 1, 1, 2

[21] Q. Zheng, J. Li, F. Huang, *An optimal Steffensen-type family for solving nonlinear equations*, Appl. Math. Comput., **217** (2011), 9592–9597. 1, 1