



# A general class of optimal eighth-order derivative free methods for nonlinear equations

Ramandeep Behl<sup>1</sup> · Ali Saleh Alshomrani<sup>1</sup> · Changbum Chun<sup>2</sup>

Received: 12 December 2019 / Accepted: 21 February 2020 / Published online: 27 February 2020  
© Springer Nature Switzerland AG 2020

## Abstract

In this paper, we present a new optimal derivative free scheme of eighth-order methods without memory in a general way. The advantage of our scheme over the earlier iteration functions, it is applicable to every optimal fourth-order derivative free scheme whose first sub step should be Steffensen's type method to develop more advanced optimal iteration techniques of order eight. In addition, the theoretical convergence properties of our schemes are fully explored with the help of main theorem that demonstrate the convergence order. Each member of the proposed scheme satisfies the classical Kung and Traub conjecture which is related to multi-point iterative methods without memory. On the basis of average number of iterations required per point and the number of points requiring 40 iterations, we confirmed that our methods are more effective and comparable to the existing robust optimal eighth-order derivative free methods. Further, the dynamical study of these methods also supports the theoretical aspects.

**Keywords** Nonlinear equations · Basin of attraction · Simple roots · Derivative-free method · Order of convergence · Steffensen's type method

## 1 Introduction

Due to the advancement of digital computer, advanced computer arithmetics and symbolic computation, many scholars proposed the eighth-order derivative free modifications or extensions of Steffensen's method or Steffensen's type method [1–6] in a number of ways at the expense of additional evaluations of functions or by increasing the substeps. All these modifications or extensions are targeted at increasing the local order of convergence with a view of increasing their efficiency index [7, 8]. Most

---

✉ Changbum Chun  
cbchun@skku.edu

<sup>1</sup> Department of Mathematics, Faculty of Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia

<sup>2</sup> Department of Mathematics, Sungkyunkwan University, Suwon 16419, Republic of Korea

of them were the extensions or modifications of a particular well known or unknown existing optimal fourth-order derivative free scheme. According to our best knowledge, we don't have any optimal derivative free scheme in a general way which is applicable to every optimal fourth-order derivative free scheme to produce further optimal scheme of order eight till date.

Nowadays, the construction of optimal derivative free schemes which are applicable to every optimal iteration function of a fixed order to further produce higher-order method have more significant importance instead of obtaining a higher-order extensions or modifications of a particular existing iteration function.

Therefore the principal aim of this paper is to present a new interesting scheme in a general way which is capable to produce further new and interesting optimal eighth-order derivative free scheme from each optimal fourth-order derivative free scheme whose first substep employs Steffensen's method or Steffensen type method. In this way, we have given the flexibility to the scholars who can pick any existing optimal derivative free fourth-order method from the available literature to further extend optimal eight-order convergence of the considered scheme. We consider a concrete variety of standard test problems in order to check the reliability and efficiency of the proposed scheme and allow us to compare them with some other existing methods of same order. Finally, the dynamical study of them have a great extent to theoretical aspects.

## 2 Development of eighth-order optimal scheme

This section is devoted to the main contribution of this research article. Our mean to say that we propose a new optimal family of eighth-order derivative free methods for solving nonlinear equations in this section. Therefore, we consider a general eighth-order derivative free scheme in the following way

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[u_n, x_n]}, & u_n = x_n + \beta f(x_n), \quad \beta \in \mathbb{R} \\ z_n = \varphi_4(u_n, x_n, y_n), \\ x_{n+1} = z_n - \frac{f(z_n)(u_n - x_n)(u_n - y_n)(x_n - y_n)}{f[y_n, z_n](u_n - x_n)(u_n - z_n)(x_n - z_n) - a(y_n - z_n)}, \end{cases} \quad (2.1)$$

where  $a = f[x_n, z_n](u_n - y_n)(u_n - z_n) - f[u_n, z_n](x_n - y_n)(x_n - z_n)$ .

In the next Theorem 3.1, we demonstrate that the order of convergence of the proposed scheme will reach at the optimal eighth-order without using any additional functional evaluations. It is interesting to observe that only a single coefficient  $\lambda_0$  of  $\varphi_4(u_n, x_n, y_n)$  contributes its role in the construction of the desired eighth-order convergence (for the details please see Theorem 3.1).

### 3 Convergence analysis

**Theorem 3.1** Let  $f : \mathbb{C} \rightarrow \mathbb{C}$  be have a simple zero  $\alpha$  and analytic function in the neighborhood of the required zero  $\xi$ . In addition, we assume that  $\varphi_4(u_n, x_n, y_n)$  is any optimal fourth-order scheme whose first substep employs (2.1). Moreover, we consider initial guess  $x = x_0$  is sufficiently close to the desired zero  $\xi$  for guaranteed convergence. Then, the scheme (2.1) has an eighth-order convergence.

**Proof** Let us assume that  $e_n = x_n - \xi$  be the error at  $n$ th point. We expand the function  $f(x_n)$  around the point  $x = \xi$  with the help of Taylor's series expansion. Then, we have

$$f(x_n) = c_1 e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + c_5 e_n^5 + c_6 e_n^6 + c_7 e_n^7 + c_8 e_n^8 + O(e_n^9), \quad (3.1)$$

where  $c_j = \frac{f^{(j)}(\xi)}{j!f'(\xi)}$  for  $j = 1, 2, \dots, 8$ .

With the help of above expression (3.1), we further obtain

$$u_n - \xi = (1 + \beta c_1) e_n + \beta (c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + c_5 e_n^5 + c_6 e_n^6 + c_7 e_n^7 + c_8 e_n^8) + O(e_n^9). \quad (3.2)$$

Now, in the combination of the Taylor's series expansion and the above expression (3.2), we have

$$f(u_n) = c_1 (1 + \beta c_1) e_n + c_2 \{ (1 + \beta c_1)^2 + \beta c_1 \} e_n^2 + \sum_{k=1}^6 A_k e_n^{k+2} + O(e_n^9), \quad (3.3)$$

where  $A_k = A_k(\beta, c_1, c_2, \dots, c_8)$ .

By using the expressions (3.1) and (3.3), we have

$$\begin{aligned} y_n - \xi &= \left( \beta + \frac{1}{c_1} \right) c_2 e_n^2 + \frac{c_1 c_3 (\beta^2 c_1^2 + 3\beta c_1 + 2) - c_2^2 (\beta^2 c_1^2 + 2\beta c_1 + 2)}{c_1^2} e_n^3 \\ &+ \sum_{k=1}^5 \bar{A}_k e_n^{k+3} + O(e_n^9). \end{aligned} \quad (3.4)$$

With the help of Taylor's series expansion, we further obtain

$$\begin{aligned} f(y_n) &= c_2 (1 + \beta c_1) e_n^2 + \frac{c_1 c_3 (\beta^2 c_1^2 + 3\beta c_1 + 2) - c_2^2 (\beta^2 c_1^2 + 2\beta c_1 + 2)}{c_1} e_n^3 \\ &+ \sum_{k=1}^5 \bar{\bar{A}}_k e_n^{k+3} + O(e_n^9). \end{aligned} \quad (3.5)$$

As we know that  $\eta(u_n, x_n, y_n)$  is any optimal fourth-order derivative free scheme. Then, it should satisfy the error equation of the following form

$$z_n - \xi = \lambda_0 e_n^4 + \lambda_1 e_n^5 + \lambda_2 e_n^6 + \lambda_3 e_n^7 + \lambda_4 e_n^8 + O(e_n^9), \tag{3.6}$$

where  $\lambda_0 \neq 0$ .

In addition, the Taylor’s series expansion of the function  $f(z_n)$  gives us

$$f(z_n) = c_1 (\lambda_0 e_n^4 + \lambda_1 e_n^5 + \lambda_2 e_n^6 + \lambda_3 e_n^7) + (c_2 \lambda_0^2 + c_1 \lambda_4) e_n^8 + O(e_n^9). \tag{3.7}$$

With the help of expressions (3.1)–(3.7), we further yield

$$f[z_n, x_n] = c_1 + c_2 e_n + c_3 e_n^2 + c_4 e_n^3 + (c_2 \lambda_0 + c_5) e_n^4 + (c_3 \lambda_0 + c_2 \lambda_1 + c_6) e_n^5 + O(e_n^6), \tag{3.8}$$

$$f[z_n, u_n] = c_1 + c_2(\beta c_1 + 1)e_n + \{c_3(\beta c_1 + 1)^2 + \beta c_2^2\} e_n^2 + \{c_4(\beta c_1 + 1)^3 + \beta c_2 c_3(2\beta c_1 + 3)\} e_n^3 + [c_2\{\beta c_4(3\beta^2 c_1^2 + 6\beta c_1 + 4) + \lambda_0\} + \beta^2 c_3 c_2^2 + (\beta c_1 + 1)\{c_5(\beta c_1 + 1)^3 + 2\beta c_3^2\}] e_n^4 + O(e_n^5) \tag{3.9}$$

and

$$f[z_n, y_n] = c_1 + \frac{c_2^2(\beta c_1 + 1)}{c_1} e_n^2 + \frac{c_2(c_1 c_3(\beta^2 c_1^2 + 3\beta c_1 + 2) - c_2^2(\beta^2 c_1^2 + 2\beta c_1 + 2))}{c_1^2} e_n^3 + O(e_n^4) \tag{3.10}$$

By using the expressions (3.1)–(3.10), we obtain

$$\frac{f(z_n)(u_n - x_n)(u_n - y_n)(x_n - y_n)}{f[y_n, z_n](u_n - x_n)(u_n - z_n)(x_n - z_n) - a(y_n - z_n)} = \lambda_0 e_n^4 + \lambda_1 e_n^5 + \lambda_2 e_n^6 + \lambda_3 e_n^7 + \left( \frac{c_1^2 \lambda_4 - c_2 \lambda_0 \{c_4(\beta c_1 + 1)^2 + c_1 \lambda_0\}}{c_1^2} \right) e_n^8 + O(e_n^9). \tag{3.11}$$

Finally, by inserting the expressions (3.6) and (3.11) in the last substep of the proposed scheme (2.1) and after some simplification, we have

$$e_{n+1} = \frac{c_2 \lambda_0 \{c_4(\beta c_1 + 1)^2 + c_1 \lambda_0\}}{c_1^2} e_n^8 + O(e_n^9), \tag{3.12}$$

The above expression (3.12) confirms two things about the scheme (2.1) that it attains an eighth-order convergence and optimal in the sense of Kung–Traub conjecture because it uses only four functional values. This completes the proof.  $\square$

**Remark 3.2** No doubts, one generally expects that some other  $\lambda_j, 1 \leq j \leq 4$  from  $\varphi_4(u_n, x_n, y_n)$  should also be involved in the asymptotic error constant of the scheme

(2.1). However, the expression (3.12) confirm that only  $\lambda_0$  appears in the asymptotic error constant of (2.1).

## 4 Numerical examples

This section is devoted to check the effectiveness and efficiency of our proposed methods with the other existing optimal eighth-order derivative free methods. Therefore, we consider some of the following special cases of  $\varphi(u_n, x_n, y_n)$

- (1)  $\varphi_4(u_n, x_n, y_n) = y_n - \frac{f(y_n)}{\frac{af(y_n)-bf(u_n)}{y_n-u_n} + \frac{cf(y_n)-df(x_n)}{y_n-x_n}}$ , such that  $a = c = 1$  and  $b + d = 1$ ,
- (2)  $\varphi_4(u_n, x_n, y_n) = y_n - \left[ \frac{f[y_n, x_n] + (p-1)f[u_n, y_n] - (p-1)f[u_n, x_n] - b(y_n - x_n)(y_n - u_n)}{f[y_n, x_n] + pf[u_n, y_n] - pf[u_n, x_n] - a(y_n - x_n)(y_n - u_n)} \right] \frac{f(y_n)}{f[y_n, x_n]}$ ,  $a, b, p \in \mathbb{R}$  and
- (3)  $\varphi_4(u_n, x_n, y_n) = y_n - \frac{f(u_n)f(y_n)(y_n - x_n)}{(f(u_n) - f(y_n))(f(y_n) - f(x_n))}$ ,

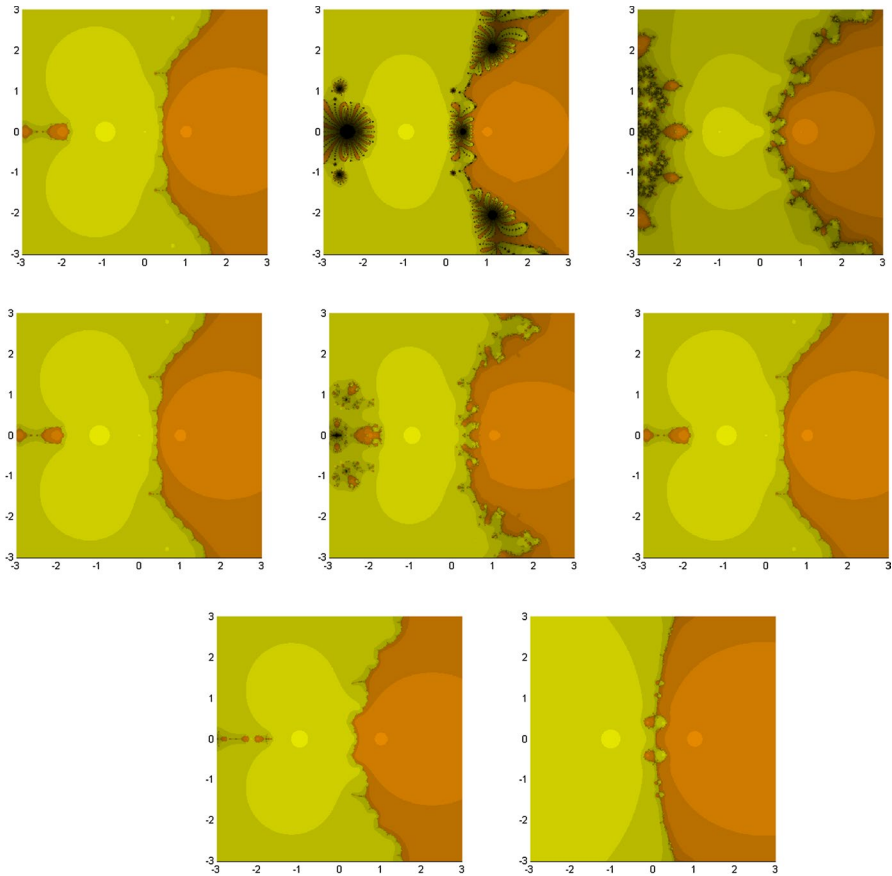
which are chosen from Cordero and Torregrosa [9], Zheng et al. [10], Kung–Traub [3]. We called these cases with our proposed scheme (2.1) by  $OM1_8$  (for  $a = b = c = 1$ ,  $d = 0$ ,  $\beta = 1$ ),  $OM2$  (for  $a = b = 0$ ,  $p = 2$  and  $\beta = 1$ ) and  $OM3_8$  ( $\beta = 1$ ), respectively. Now, we will compare them with the existing optimal derivative free eighth-order methods which were proposed by Zheng et al. [1], Soleymani and Vanani [2], Kung and Traub [3], Khattri and Steihaug [4] and Thukral [5], out of which we have chosen methods namely, method (8) with  $\gamma = 1$ , method (21), method (4.1) with  $\beta = 1$ , method (17) with  $\alpha = 1$  and method (2.8) with  $\beta = 1$ , respectively. We denoted these methods by  $ZM_8$ ,  $SK_8$ ,  $KT_8$ ,  $KM_8$  and  $TM_8$ , respectively.

All the examples have roots within a square of  $[-3, 3]$  by  $[-3, 3]$  except one within another square of  $[-1, 5]$  by  $[-3, 3]$ . We have taken 360,000 equally spaced points in the square as initial points for the methods and we have registered the total number of iterations required to converge to a root and also to which root it converged. We have also collected the CPU time (in seconds) required to run each method on all the points using Samsung desktop computer with Intel(R) Core(TM) i7-8700K CPU. We then computed the average number of iterations required per point and the number of points requiring 40 iterations.

**Example 1** The first example is the quadratic polynomial

$$p_1(z) = z^2 - 1 \quad (4.1)$$

whose roots are at  $\pm 1$ . We have plotted the basins in Fig. 1. We used a different color for each basin, so that we can tell if the method converged to the closest root. We have also used lighter shade when the number of iterations is lower and at the maximum number of iterations we color the point black. Therefore ideally the method should show lighter shades. The best methods are  $OM3_8$ ,  $OM1_8$ ,  $ZM_8$  and  $KM_8$ .



**Fig. 1** The top row for  $ZM_8$  (left),  $SK_8$  (center) and  $KT_8$  (right). The second row for  $KM_8$  (left),  $TM_8$  (center) and  $OM1_8$  (right). The bottom row for  $OM2_8$  (left),  $OM3_8$  (right) for the roots of the polynomial  $z^2 - 1$

Now we check Table 1 to see the average number of iterations per point. The minimum is 2.34 and it is achieved by  $OM3_8$  followed by methods:  $ZM_8$ ,  $KM_8$  and  $OM1_8$  (2.56), and  $OM2_8$  (2.58). The highest number (4.57) is used by  $KT_8$ . All other methods used 3.01–4.13 number of iterations per point on average.

Based on the CPU time in seconds, we find that the fastest method is  $ZM_8$  with 135.219 s. The slowest is  $SK_8$  with 238.062 s. We can see that the basins for this method have many black points (Fig. 1, middle of top row). In terms of the number of black points (see Table 3) we find that most methods have no such points except  $SK_8$  (30825 points),  $TM_8$  (94 points), and  $KT_8$  (4 points).

**Example 2** The second example is the cubic polynomial

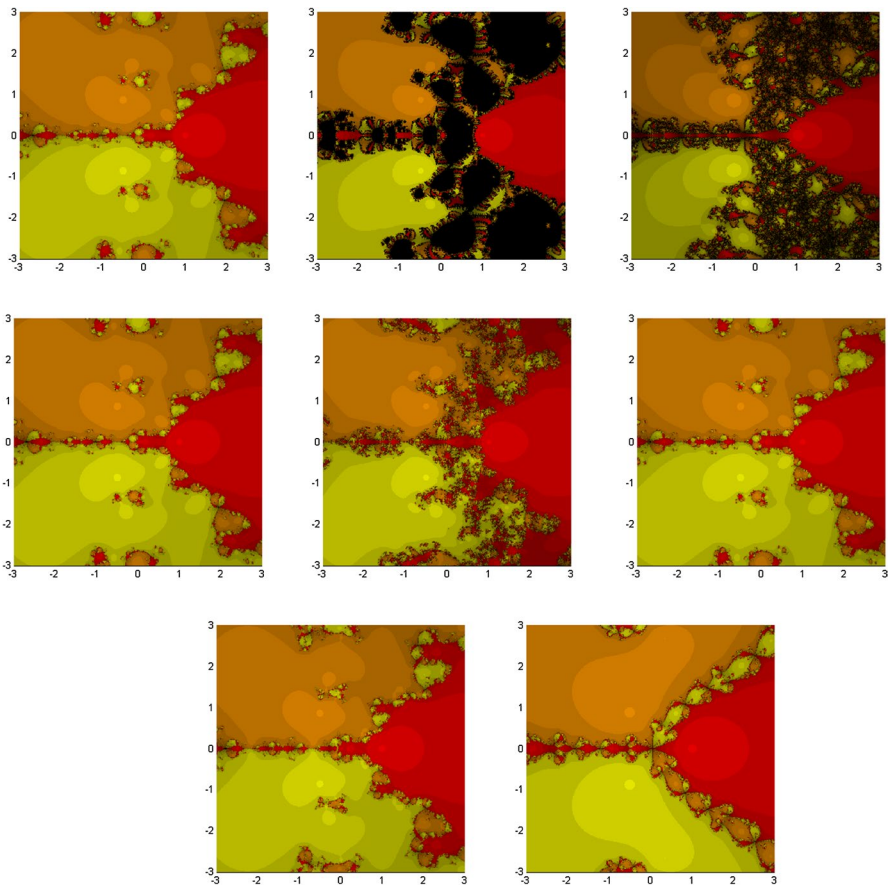
$$p_2(z) = z^3 - 1 \tag{4.2}$$

having the 3 roots of unity.

The basins of attraction are given in Fig. 2. Based on these plots we find that  $OM3_8$ ,  $OM2_8$ ,  $ZM_8$  and  $OM1_8$  are best. Based on Table 1 we find that the minimum number of iterations per point on average is achieved by  $OM3_8$  (3.19) followed by  $ZM_8$  (3.60), and  $OM1_8$  and  $KM_8$  (3.61). The worst are  $SK_8$  (17.28) and  $KT_8$  (11.63). All the other methods use 3.65–4.88.

The fastest method is  $ZM_8$  method (244.796 s) and the slowest are  $SK_8$  (704.48 s),  $KT_8$  (605.61 s),  $TM_8$  (475.14 s), and  $KM_8$  (392.906 s). Based on the number of black points clearly we have  $SK_8$  being the worst with 129371 such points and  $KT_8$  (14258).

**Example 3** The third example is another cubic polynomial, but with real roots only, i.e. the polynomial is given by:

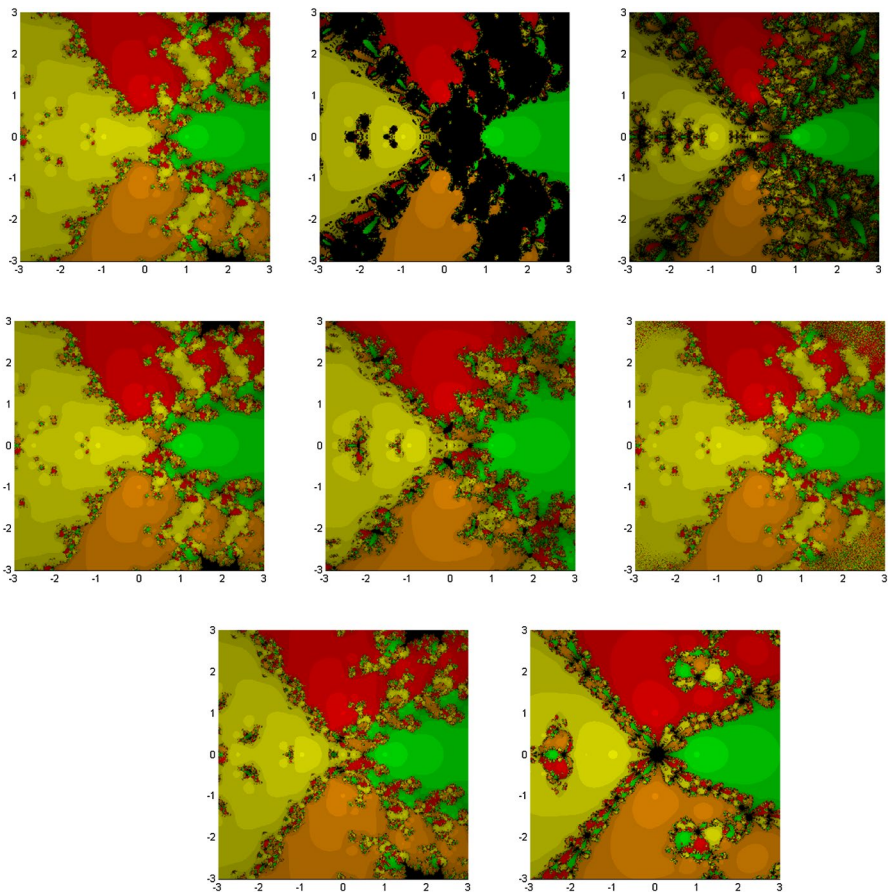


**Fig. 2** The top row for  $ZM_8$  (left),  $SK_8$  (center) and  $KT_8$  (right). The second row for  $KM_8$  (left),  $TM_8$  (center) and  $OM1_8$  (right). The bottom row for  $OM2_8$  (left),  $OM3_8$  (right) for the roots of the polynomial  $z^3 - 1$

$$p_3(z) = z^4 - 1. \quad (4.3)$$

The basins of attraction are displayed in Fig. 3. It seems that the best methods are  $OM3_8$ ,  $OM1_8$  and  $KM_8$ . Consulting the number of iterations per point, we find that  $OM1_8$  is best (4.78) followed by  $KM_8$  and  $ZM_8$  (5.22), and  $OM3_8$  (5.33). The worst is  $SK_8$  (20.41). All the others use 5.52 – 13.22 iterations per point. The fastest method is again  $ZM_8$  (356.656 s). The slowest are  $SK_8$  (798.81),  $KT_8$  (664.89 s), and  $KM_8$  (588.813). All methods have black points.

**Example 4** Let us consider a quartic equation from [11, 12], which describes the fraction of the nitrogen-hydrogen feed that gets converted to ammonia (this fraction is called fractional conversion). By considering 250 atm and 500<sup>0</sup> C, then the mentioned equation can be converted into the following form



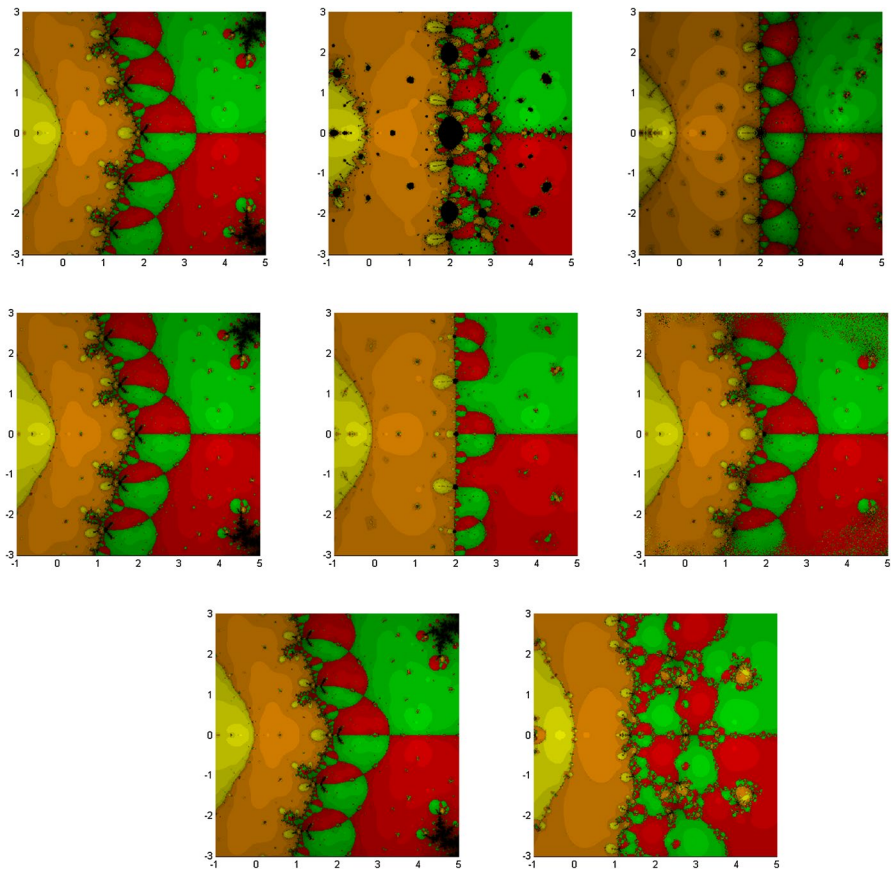
**Fig. 3** The top row for  $ZM_8$  (left),  $SK_8$  (center) and  $KT_8$  (right). The second row for  $KM_8$  (left),  $TM_8$  (center) and  $OM1_8$  (right). The bottom row for  $OM2_8$  (left),  $OM3_8$  (right) for the roots of the polynomial  $z^4 - 1$



$$p_4(z) = z^4 - 7.79075z^3 + 14.7445z^2 + 2.511. \quad (4.4)$$

The above function has total four number of zeros and out of them two are real and other two are complex conjugate to each other. However, our desired zero is  $\xi = 3.9485424455620457727 + 0.3161235708970163733i$ .

The basins are displayed in Fig. 4. The best methods are  $TM_8$ ,  $OM1_8$  and  $OM2_8$ . Based on the average number of iterations per point (see Table 1) we find that the minimum is achieved by  $OM3_8$  (3.86), followed closely by  $TM_8$  (3.99) and  $OM1_8$  (4.82). The worst method in this sense is  $KT_8$  which uses 7.34 iterations per point on average. The rest of the methods use 5.35 – 6.93 iterations per point on average. In terms of the CPU time, the fastest method is  $OM3_8$  (651.703 s). The slowest are  $KM_8$  with 999.453 s and  $SK_8$  with 912.7 s. All others use 689.39–895.094 s. Based on the



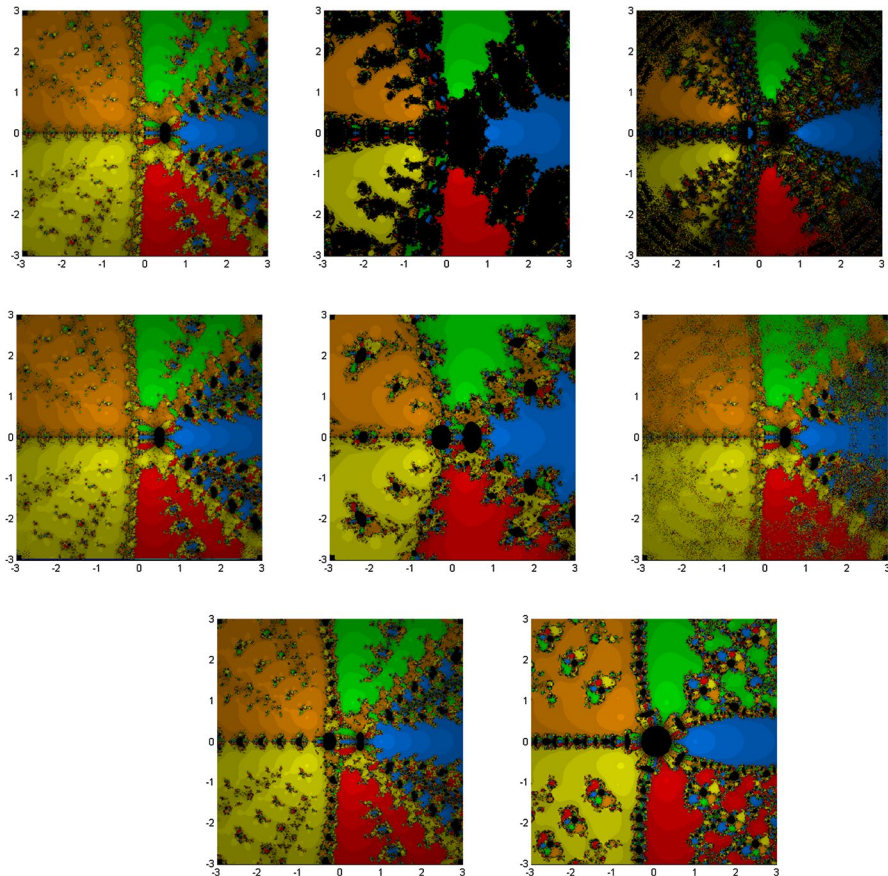
**Fig. 4** The top row for  $ZM_8$  (left),  $SK_8$  (center) and  $KT_8$  (right). The second row for  $KM_8$  (left),  $TM_8$  (center) and  $OM1_8$  (right). The bottom row for  $OM2_8$  (left),  $OM3_8$  (right) for the roots of the polynomial  $z^4 - 7.79075z^3 + 14.7445z^2 + 2.511z - 1.674$

number of black points, the worst method is  $SK_8$  with 23514 black points, and all other methods have between 1075 and 4264 black points.

**Example 5** The fifth example is a fifth degree polynomial

$$p_5(z) = z^5 - 1. \quad (4.5)$$

The basins are displayed in Fig. 5. It seems that the best methods are  $KM_8$  and  $OM1_8$ . The data in Tables 1, 2 and 3 give a quantitative information. Based on Table 1 we find that  $SK_8$  is the worst, requiring 23.18 iterations per point on average. The smallest number of iterations on average is for  $OM1_8$  (8.01) closely followed by  $KM_8$  (8.06). The rest of the methods use between 8.07 and 21.19 iterations per point. The fastest method is  $ZM_8$  (589 s) and the slowest is  $SK_8$  (1003.7 s). The rest



**Fig. 5** The top row for  $ZM_8$  (left),  $SK_8$  (center) and  $KT_8$  (right). The second row for  $KM_8$  (left),  $TM_8$  (center) and  $OM1_8$  (right). The bottom row for  $OM2_8$  (left),  $OM3_8$  (right) for the roots of the polynomial  $z^5 - 1$

use 614.5–1001.94 s. In terms of black points, we find again that the worst are  $SK_8$  (187906) and  $KT_8$  (130717). All other methods have at least 16837 black points.

**Example 6** An undesirable RF breakdown which may happen in the high power microwave devices working under the vacuum condition is known as is multi factor [13]. For example, multi factor appears inside a parallel plate waveguide. There exists an electric field with an electric potential difference which creates the electron movement between these two plates. An interesting case in the study of the electron trajectories is when the electron reaches a plate with root of multiplicity 2. The trajectory of an electron in the air gap between two parallel plates is as follows

$$y(t) = y_0 + \left( v_0 + e \frac{E_0}{m\omega} \sin(\omega t_0 + \alpha) \right) (t - t_0) + e \frac{E_0}{m\omega^2} (\cos(\omega t + \alpha) - \cos(\omega t_0 + \alpha)) \quad (4.6)$$

where  $t$  is time,  $m$  and  $e$  are the mass and charge of the electron at rest,  $E_0 \sin(\omega t + \alpha)$  is the RF electric field between plates and  $y_0$  and  $v_0$  are the position and velocity of the electron at time  $t_0$ . We consider the following particular case of (4.6), where the parameters have been normalized:

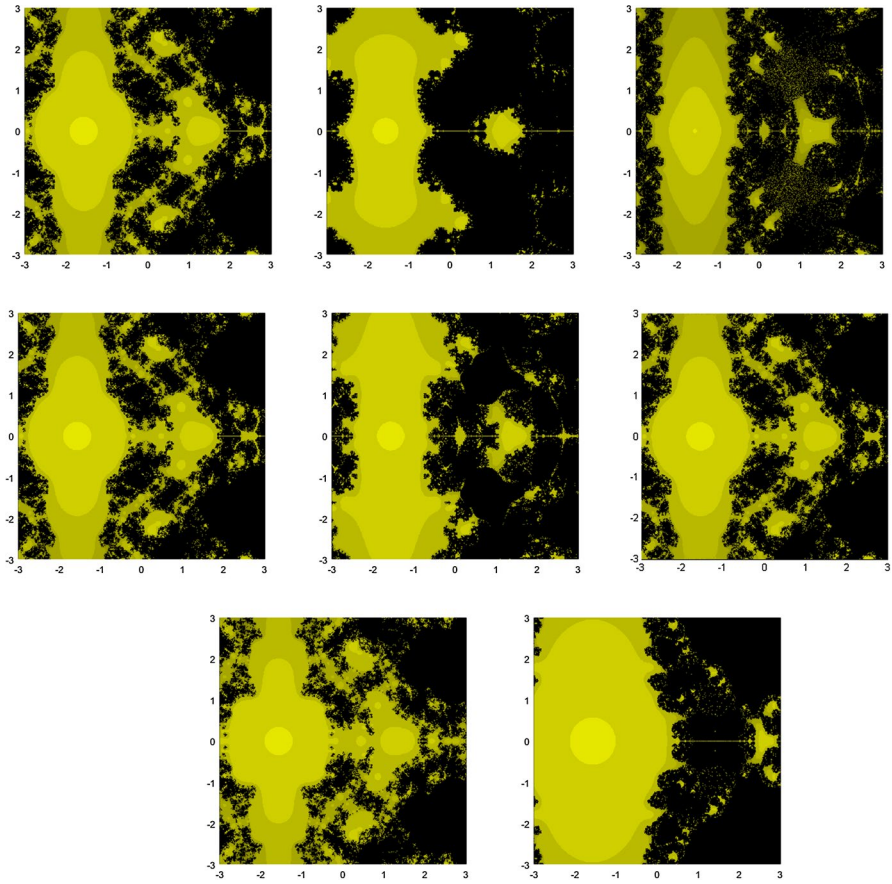
$$p_6(z) = z + \cos(z) + \frac{\pi}{2}. \quad (4.7)$$

with the simple zero  $\xi = -1.570796326794896619231322$ . so, our last example is a non-polynomial function defined by (4.7). This is an example that was difficult for many methods. The basins are displayed in Fig. 6. The best methods seem to be  $OM3_8$  and  $TM_8$ . In terms of average number of iterations per point,  $OM2_8$  is the best method with 18.96 followed by  $OM3_8$  with 19.59. The worst is  $SK_8$  and  $KT_8$  with 24.45 iterations per point on average. The fastest method (Table 2) is  $OM3_8$  (394.750 s) and the slowest is  $KM_8$  (770.365 s).  $SK_8$  and  $KT_8$  have the highest number of black points (Table 3). It is clear that one has to use quantitative measures to distinguish between methods, since we have a different conclusion when just viewing the basins of attraction.

In order to pick the best method overall, we have averaged the results in Tables 1, 2 and 3 across the 6 examples. The best method based on the 3 criteria used is  $OM3_8$ . The method with the fewest number of iterations per point is  $OM3_8$  (7.19) followed closely by  $OM1_8$  (7.50) and  $OM2_8$  (7.52). The fastest method is  $ZM_8$  (415.487 s) closely followed by  $OM3_8$  (418.597 s). The methods with the least number of black points on average are  $OM2_8$  (32082.5 points) and  $OM1_8$  (32673.7 points).

## 5 Conclusions

In this work, we present a new interesting general optimal eighth-order scheme that is applicable to each earlier optimal multi-point derivative free scheme of order four whose first sub step should be Steffensen's type method to construct further new optimal iteration functions of order eight. In addition, it does not use any kind



**Fig. 6** The top row for  $ZM_8$  (left),  $SK_8$  (center) and  $KT_8$  (right). The second row for  $KM_8$  (left),  $TM_8$  (center) and  $OM1_8$  (right). The bottom row for  $OM2_8$  (left),  $OM3_8$  (right) for the root of the non-polynomial  $z + \cos(z) + \frac{\pi}{2}$

**Table 1** Average number of iterations per point for each test function (1–6) and each of the methods

Method	Ex1	Ex2	Ex3	Ex4	Ex5	Ex6	Average
$ZM_8$	2.56	3.60	5.22	5.39	8.07	21.17	7.67
$SK_8$	4.13	17.28	20.41	6.93	23.18	24.45	16.06
$KT_8$	4.57	11.63	13.22	7.34	21.19	24.45	13.73
$KM_8$	2.56	3.61	5.22	5.40	8.06	21.18	7.67
$TM_8$	3.01	4.88	6.14	3.99	9.36	21.96	8.22
$OM1_8$	2.56	3.61	4.78	4.82	8.01	21.20	7.50
$OM2_8$	2.58	3.68	5.52	5.35	9.03	18.96	7.52
$OM3_8$	2.34	3.19	5.33	3.86	8.84	19.59	7.19

**Table 2** CPU time (in seconds) required for each test function (1–6) and each of the methods

Method	Ex1	Ex2	Ex3	Ex4	Ex5	Ex6	Average
$ZM_8$	135.219	244.796	356.656	689.39	589	477.859	415.487
$SK_8$	238.062	704.48	798.81	912.7	1003.7	396.469	675.704
$KT_8$	168.406	605.61	664.89	828.125	1001.94	485.937	625.818
$KM_8$	228.203	392.906	588.813	999.453	897.625	770.375	646.229
$TM_8$	215.39	475.14	564.828	794.609	779.313	515.641	557.487
$OM1_8$	171.266	308.453	422.36	772.359	736.265	632.171	507.146
$OM2_8$	190.89	339.844	508.719	895.094	791.187	544.000	544.956
$OM3_8$	156.313	269.86	424.453	651.703	614.5	394.750	418.597

**Table 3** Number of points requiring more than 40 iterations for each test function (1–6) and each of the methods

Method	Ex1	Ex2	Ex3	Ex4	Ex5	Ex6	Average
$ZM_8$	0	81	5735	4195	17,713	176,360	34,014
$SK_8$	2575	129,371	160,807	23,514	187,906	211,194	119,227.8
$KT_8$	4	14,258	34,059	2900	130,717	202,768	64,117.7
$KM_8$	0	88	5759	4264	17,623	176,410	34,024
$TM_8$	94	19	7329	1075	42,044	186,986	39,591.2
$OM1_8$	0	82	1015	1508	16,837	176,600	32,673.7
$OM2_8$	0	98	7136	3856	27,641	153,764	32,082.5
$OM3_8$	0	40	11,712	1187	40,698	16,5630	36,544.5

of derivative of the considered function and also satisfy the classical Kung–Traub optimality conjecture. We can easily find many new optimal eighth-order derivative free iteration functions. We have compared the basins of our methods with several existing methods of same order using 3 quantitative measure and found that the best method based on all 3 criteria is  $OM3_8$ . Therefore, we conclude that our proposed iteration scheme provides methods at least as better as the ones available in the literature.

**Acknowledgements** This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, Saudi Arabia under Grant No. (KEP-15-130-40). The authors, therefore, acknowledge with thanks DSR technical and financial support.

## References

1. Q. Zheng, J. Li, F. Huang, An optimal Steffensen-type family for solving nonlinear equations. *Appl. Math. Comput.* **217**, 9592–9597 (2011)
2. F. Soleymani, S.K. Vanani, Optimal Steffensen-type methods with eighth order of convergence. *Comput. Math. Appl.* **62**, 4619–4626 (2011)

3. H.T. Kung, J.F. Traub, Optimal order of one-point and multi-point iteration. *J. ACM* **21**, 643–651 (1974)
4. S.K. Khattri, T. Steihaug, Algorithm for forming derivative-free optimal methods. *Numer. Algorithms* **65**, 809–824 (2014)
5. R. Thukral, in *Eighth-Order Iterative Methods Without Derivatives for Solving Nonlinear Equations, International Scholarly Research Network*. ISRN Applied Mathematics (2011) Article ID 693787 <https://doi.org/10.5402/2011/693787>
6. M. Kansal, V. Kanwar, S. Bhatia, An optimal eighth-order derivative-free family of Potra–Pták’s method. *Algorithms* **8**, 309–320 (2015)
7. M.S. Petković, B. Neta, L.D. Petković, J. Džunić, *Multipoint Methods for Solving Nonlinear Equations* (Academic Press, Cambridge, 2012)
8. J.F. Traub, *Iterative Methods for the Solution of Equations* (Prentice-Hall, Englewood Cliffs, 1964)
9. A. Cordero, J.R. Torregrosa, A class of Steffensen type methods with optimal order of convergence. *Appl. Math. Comput.* **217**, 7653–7659 (2011)
10. Q. Zheng, P. Zhao, F. Huang, A family of fourth-order Steffensen-type methods with the applications on solving nonlinear ODEs. *Appl. Math. Comput.* **217**, 8196–8203 (2011)
11. G.V. Balaji, J.D. Seader, Application of interval Newton’s method to chemical engineering problems. *Rel. Comput.* **1**(3), 215–223 (1995)
12. M. Shacham, An improved memory method for the solution of a nonlinear equation. *Chem. Eng. Sci.* **44**(7), 1495–1501 (1989)
13. S. Anza, C. Vicente, B. Gimeno, V.E. Boria, J. Armendáriz, Long-term multipactor discharge in multicarrier systems. *Phys Plasmas* **14**(8), 082–112 (2007)

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.