Journal of Applied Analysis and Computation
Volume 9, Number 2, April 2019, 400–427
Website:http://jaac.ijournal.cn
DOI:10.11948/2156-907X.20160229

COMPARATIVE STUDY OF METHODS OF VARIOUS ORDERS FOR FINDING SIMPLE ROOTS OF NONLINEAR EQUATIONS

Changbum Chun[1] and Beny Neta[2,†]

Abstract Recently there were many papers discussing the basins of attraction of various methods and ideas how to choose the parameters appearing in families of methods and weight functions used. Here we collected many of the results scattered and put a quantitative comparison of methods of orders from 2 to 7. We have used the average number of function-evaluations per point, the CPU time and the number of black points to compare the methods. We also include the best eighth order method. Based on 7 examples, we show that there is no method that is best based on the 3 criteria. We found that the best eighth order method, SA8, and CLND are at the top.

Keywords Iterative methods, nonlinear equations, simple roots, order of convergence, basin of attraction.

MSC(2010) 65H05, 65B99.

1. Introduction

There are many iterative methods for obtaining simple zeros of a single nonlinear equation [68]. We will not discuss derivative-free methods or methods with memory. There are many new methods and families of methods, some of which are just rediscovery of old ones or special cases of known families of methods, see e.g. [61] for examples of such cases.

The usual technique of comparing a new method to existing ones, is by comparing the performance on selected problems using one or two initial points or by comparing the efficiency index (see [68]). In recent work, one can find a visual comparison, by plotting the basins of attraction for the methods. The idea of using basins of attraction appeared first in Stewart [66] and followed by the works of Amat et al. [1,2], and [3], Scott et al. [63], Chun et al. [10–13,19], Chicharro et al. [26], Cordero et al. [27], Neta et al. [49,50], Argyros and Magreñan, [4], Magreñan, [46] and Geum et al. [30–32] and [33]. In later works ([12–16]), we have introduced a more quantitative comparison, by listing the average number of iterations per point, the CPU time and the number of points requiring 40 iterations. We have also discussed methods to choose the parameters appearing in the method and/or the weight function (see, e.g. [17]).

†The corresponding author. Email address:bneta@nps.edu(B. Neta), Tel.: 1 (831) 656-2235, Fax: 1 (831) 656-2355
1Department of Mathematics, Sungkyunkwan University, Suwon 16419, Republic of Korea
2Naval Postgraduate School, Department of Applied Mathematics, Monterey, CA 93943

First we list the methods we consider here with their order of convergence (p), number of function- (and derivative-) evaluation per step (ν) and efficiency (I). We include the best performer out of all eighth order methods (see [19].)

1. Newton's optimal method ($p = 2$, $\nu = 2$, $I = 1.4142$)
2. Hansen-Patrick's family of methods ($p = 3$, $\nu = 3$, $I = 1.4422$)
   - Halley
   - Euler-Chebyshev
   - Basto-Semiao-Calheiros's family of methods
3. Super Halley optimal method ($p = 4$, $\nu = 3$, $I = 1.5874$)
4. King's family (including Ostrowski's method) ($p = 4$, $\nu = 3$, $I = 1.5874$)
5. Kung-Traub's optimal method ($p = 4$, $\nu = 3$, $I = 1.5874$)
6. Maheshwari's method ($p = 4$, $\nu = 3$, $I = 1.5874$)
7. Hermite interpolation Jarratt's based method ($p = 4$, $\nu = 4$, $I = 1.4142$)
8. Chun et al.'s method ($p = 4$, $\nu = 3$, $I = 1.5874$)
   - Jarratt's optimal methods ($p = 4$, $\nu = 3$, $I = 1.5874$)
   - Modified super Halley optimal method ($p = 4$, $\nu = 3$, $I = 1.5874$)
   - Kou et al.
9. Khattri's family of methods ($p = 4$, $\nu = 3$, $I = 1.5874$)
10. Murakami's family of methods ($p = 5$, $\nu = 4$, $I = 1.4953$)
11. Neta's family of methods ($p = 6$, $\nu = 4$, $I = 1.5651$)
12. Chun-Neta's method ($p = 6$, $\nu = 4$, $I = 1.5651$)
13. Method similar to Wang-Liu's except we replaced the function in the last sub-step by the Hermite polynomial ($p = 6$, $\nu = 4$, $I = 1.5651$)
14. Newton's interpolating polynomial and King's method, ($p = 7$, $\nu = 4$, $I = 1.6265$)
15. Hermite based Jarratt's method ($p = 7$, $\nu = 5$, $I = 1.4758$)
16. Bi-Ren-Wu's method, ($p = 7$, $\nu = 4$, $I = 1.6265$)
17. Sharma-Arora's method ($p = 8$, $\nu = 4$, $I = 1.6818$)

We now detail all the above methods.

1. Newton's optimal method (see e.g. Traub [68]) is of second order for simple roots and given by

$$x_{n+1} = x_n - u_n, \tag{1.1}$$

where

$$u_n = \frac{f_n}{f'_n}, \tag{1.2}$$

and $f_n = f(x_n)$ and similarly for the derivative.

2. Hansen-Patrick's family of methods [36] is of third order and given by

$$x_{n+1} = x_n - \frac{(\alpha+1)f_n}{\alpha f_n' \pm \sqrt{(f_n')^2 - (\alpha+1)f_n f_n}}. \tag{1.3}$$

If $\alpha = 1$ and the square root is approximated linearly then we get Halley's method [35]

$$x_{n+1} = x_n - \frac{u_n}{1 - \frac{f_n''}{2f_n'}u_n}. \tag{1.4}$$

It can also be obtained as a member of the family (see Popovski [62])

$$x_{n+1} = x_n + (e-1)\frac{f_n'}{f_n''}\left\{\left[1 - \frac{e}{e-1}\frac{f_n f_n''}{(f_n')^2}\right]^{1/e} - 1\right\}. \tag{1.5}$$

Another form is:

$$x_{n+1} = x_n - \frac{f_n}{f_n'} - \frac{f_n^2 f_n''}{2(f_n')^3 - A f_n f_n' f_n''}. \tag{1.6}$$

Notice that this is just (1.4) with an additional parameter. Upon choosing $A = 1$ we have Halley's method (1.4). The choice $A = 0$ yields the well known Euler-Chebyshev method [37]. This latter method is also a special case of Hansen and Patrick's family (1.3) with $\alpha = 1$ or Popovski's family (1.5) with $e = \frac{1}{2}$. The choice $A = 2$ gives the BSC method [6].

3. Super Halley fourth order (SH4) method [34] is given by

$$\begin{aligned} y_n &= x_n - \frac{2}{3}u_n, \\ x_{n+1} &= x_n - \left(1 + \frac{1}{2}\frac{L_f}{1-L_f}\right)u_n, \end{aligned} \tag{1.7}$$

where

$$L_f = \frac{f_n f_n''}{(f_n')^2}. \tag{1.8}$$

4. King's family of fourth-order methods (K4) [42] is given by

$$\begin{aligned} y_n &= x_n - u_n, \\ x_{n+1} &= y_n - \frac{f(y_n)}{f_n'}\frac{f_n + \beta f(y_n)}{f_n + (\beta-2)f(y_n)}. \end{aligned} \tag{1.9}$$

Ostrowski's method [60] is a special case of (1.9) with $\beta = 0$.

5. Kung and Traub fourth order (KT4) optimal method [44] is given by

$$\begin{aligned} y_n &= x_n - u_n, \\ x_{n+1} &= y_n - \frac{f(y_n)}{f_n'}\frac{1}{[1 - f(y_n)/f_n]^2}. \end{aligned} \tag{1.10}$$

6. Maheshwari's method (M4) [47]

$$\begin{aligned} y_n &= x_n - u_n, \\ x_{n+1} &= x_n - \left[\left(\frac{f(y_n)}{f_n} \right)^2 - \frac{f_n}{f(y_n) - f_n} \right] u_n. \end{aligned} \tag{1.11}$$

7. Hermite interpolation based on Jarratt's method (JHIF4) where the interpolating polynomial replacing the function at the third sub-step is given by

$$\begin{aligned} y_n &= x_n - \frac{2}{3} u_n, \\ z_n &= x_n - \frac{1}{2} u_n - \frac{1}{2} \frac{u_n}{1 + \frac{3}{2}\left(\frac{f'(y_n)}{f'_n} - 1 \right)}, \\ x_{n+1} &= z_n - \frac{H_3(z_n)}{f'(z_n)}, \end{aligned} \tag{1.12}$$

where

$$\begin{aligned} H_3(z_n) = f_n + f'_n \frac{(z_n - y_n)^2 (z_n - x_n)}{(y_n - x_n)(x_n + 2y_n - 3z_n)} + f'(z_n) \frac{(z_n - y_n)(x_n - z_n)}{x_n + 2y_n - 3z_n} \\ - f[x_n, y_n] \frac{(z_n - x_n)^3}{(y_n - x_n)(x_n + 2y_n - 3z_n)}. \end{aligned} \tag{1.13}$$

8. Chun et al.'s fourth order family of methods [10] is given by

$$y_n = x_n - \frac{2}{3} u_n, \tag{1.14}$$

$$x_{n+1} = x_n - \frac{f_n}{f'_n} H(\tilde{t}(x_n)), \tag{1.15}$$

where the weight function H satisfies $H(0) = 1$, $H'(0) = \frac{1}{2}$, $H''(0) = 1$, and

$$\tilde{t}(x_n) = \frac{3}{2} \frac{f'_n - f'(y_n)}{f'_n}. \tag{1.16}$$

The modified Super Halley fourth order (MSH4) optimal method [21] is given by

$$\begin{aligned} y_n &= x_n - \frac{2}{3} u_n, \\ x_{n+1} &= x_n - \left(1 + \frac{1}{2} \frac{\hat{L}_f}{1 - \hat{L}_f} \right) u_n, \end{aligned} \tag{1.17}$$

where

$$\hat{L}_f = \frac{f_n}{(f'_n)^2} \frac{f'(y_n) - f'_n}{y_n - x_n}. \tag{1.18}$$

The method can be rearranged to get Jarratt's method (see Neta et al. [55].)

The optimal fourth-order family of methods due to Kou et al. [43] is given by

$$x_{n+1} = x_n - \left(1 - \frac{3}{4} \frac{(f'(y_n) - f'_n)(\gamma f'(y_n) + (1 - \gamma) f'_n)}{(\alpha f'(y_n) + (1 - \alpha) f'_n)(\beta f'(y_n) + (1 - \beta) f'_n)} \right) \frac{f_n}{f'_n}. \tag{1.19}$$

This family is a special case of (1.15) when the weight function is given by

$$H(t) = 1 - \frac{3}{4}\frac{t(\gamma t - \frac{3}{2})}{(\alpha t - \frac{3}{2})(\beta t - \frac{3}{2})}, \tag{1.20}$$

where $\gamma = \alpha + \beta - \dfrac{3}{2}, \alpha, \beta \in \mathbf{R}$.

Remark: There was a mistake in our paper [11] in the weight function (1.20).

9. Khattri's family of methods ($p = 4$)
Khattri et al. [41] has developed the following optimal fourth order 3 parameter family of methods

$$\begin{aligned} y_n &= x_n - u_n + \frac{\alpha\beta}{2} f_n^m, \\ x_{n+1} &= y_n - \frac{f_n f(y_n)}{f_n - 2f(y_n)}\left[\frac{\alpha}{f'_n + \beta f_n^m} - \frac{\alpha - 1}{f'_n + \eta f(y_n)}\right]. \end{aligned} \tag{1.21}$$

Chun and Neta [22] have experimented with this method and found that the best choice of parameters is $\alpha = -2$, $\beta = 0$ and $\eta = 0.001$. This method was denoted KB7 in [22], but here we changed the name to KB74 to emphasize it is of order 4.

10. Murakami's fifth order method (M5) [48] is given by

$$x_{n+1} = x_n - a_1 u_n - a_2 w_2(x_n) - a_3 w_3(x_n) - \psi(x_n), \tag{1.22}$$

where

$$\begin{aligned} w_2(x_n) &= \frac{f_n}{f'(x_n - u_n)}, \\ w_3(x_n) &= \frac{f_n}{f'(x_n + \beta u_n + \gamma w_2(x_n))}, \\ \psi(x_n) &= \frac{f_n}{b_1 f'_n + b_2 f'(x_n - u_n)}. \end{aligned} \tag{1.23}$$

To get fifth order, Murakami suggested several possibilities and we picked the following

$$\begin{aligned} &\gamma = 0, \quad a_1 = .3, \quad a_2 = -.5, \quad a_3 = \frac{2}{3}, \\ &b_1 = -\frac{15}{32}, \quad b_2 = \frac{75}{32}, \quad \beta = -\frac{1}{2}. \end{aligned} \tag{1.24}$$

11. Neta's sixth order family of methods (N6) [56] is given by

$$\begin{aligned} y_n &= x_n - u_n, \\ z_n &= y_n - \frac{f(y_n)}{f'_n}\frac{f_n + \beta f(y_n)}{f_n + (\beta - 2)f(y_n)}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{f'_n}\frac{f_n - f(y_n)}{f_n - 3f(y_n)}. \end{aligned} \tag{1.25}$$

Note that the first two sub-steps are King's method. Several choices for the parameter β were discussed.

12. Another sixth order method due to Chun and Neta (CN6) [23] is based on Kung and Traub scheme [44],

$$\begin{aligned} y_n &= x_n - u_n, \\ z_n &= y_n - \frac{f(y_n)}{f'_n} \frac{1}{[1 - f(y_n)/f_n]^2}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{f'_n} \frac{1}{[1 - f(y_n)/f_n - f(z_n)/f_n]^2}. \end{aligned} \tag{1.26}$$

13. Method similar to Wang-Liu's (WL6) except we replaced the function in the last sub-step by the Hermite polynomial instead of replacing the derivative.

$$\begin{aligned} y_n &= x_n - u_n, \\ z_n &= y_n - \frac{f(y_n)}{f'_n} \frac{f_n}{f_n - 2f(y_n)}, \\ x_{n+1} &= z_n - \frac{H_3(z_n)}{f'(z_n)}, \end{aligned} \tag{1.27}$$

where $H_3(z_n)$ is given by (1.13).

14. A seventh order method based on Newton's interpolating polynomial and King's method, denoted NIK7, and is given by (see (3.35) in [61])

$$\begin{aligned} y_n &= x_n - u_n, \\ z_n &= y_n - \frac{f(y_n)}{f'_n} \frac{f_n + \beta f(y_n)}{f_n + (\beta - 2) f(y_n)}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{f[z_n, y_n] + f[z_n, y_n, x_n](z_n - y_n)}, \end{aligned} \tag{1.28}$$

where $f[z,x] = \dfrac{f(z) - f(x)}{z - x}$ and $f[z,y,x] = \dfrac{f[y,x] - f[z,y]}{x - z}$.

15. Hermite based Jarratt's method (JHID7), we added a Newton-like sub-step and replaced the derivative with a Hermite interpolating polynomial. The resulting scheme is of order seven. The method is given by

$$\begin{aligned} y_n &= x_n - \frac{2}{3} u_n, \\ z_n &= x_n - \frac{1}{2} u_n - \frac{1}{2} \frac{u_n}{1 + \frac{3}{2}\left(\frac{f'(y_n)}{f'_n} - 1\right)}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{H'_3(z_n)}, \end{aligned} \tag{1.29}$$

where

$$H'_3(z_n) = 2(f[x_n, z_n] - f[x_n, y_n]) + f[y_n, z_n] + \frac{y_n - z_n}{y_n - x_n}(f[x_n, y_n] - f'_n). \tag{1.30}$$

16. Bi-Ren-Wu's 7th order method [7], denoted BRW7

$$\begin{aligned} y_n &= x_n - u_n, \\ z_n &= x_n - u_n \, \frac{f_n + \beta f(y_n)}{f_n + (\beta - 2) f(y_n)}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{f[z_n, y_n] + f[z_n, x_n, x_n](z_n - y_n)}. \end{aligned} \tag{1.31}$$

17. Sharma-Arora's eighth order method, denoted SA8, [65]

$$\begin{aligned} y_n &= x_n - u_n, \\ z_n &= \phi_4(x_n, y_n), \\ x_{n+1} &= z_n - \frac{f[z_n, y_n]}{f[z_n, x_n]} \, \frac{f(z_n)}{2 f[z_n, y_n] - f[z_n, x_n]}, \end{aligned} \tag{1.32}$$

where

$$\phi_4(x_n, y_n) = y_n - \frac{f(y_n)}{2 f[y_n, x_n] - f'_n}. \tag{1.33}$$

2. Extraneous fixed points

In this section, we introduce the notion of extraneous fixed points and show how to find those for any given method. It is easy to see that any method can be written as

$$x_{n+1} = x_n - H_f \, \frac{f_n}{f'_n}, \tag{2.1}$$

where the function H_f depends on x_n and other intermediate values. In Table 1 we list the function H_f for each of the methods of orders 2–5 and in Table 2 for each of the methods of orders 6–7.

It is clear that if x_n is a zero of the function $f(x)$ then x_n is a fixed point of the iterative method (2.1). But even if x_n is a zero of H_f and not of $f(x)$ it is a fixed point. Those fixed points that are zeroes of H_f and not of $f(x)$ are called extraneous fixed points. For example, Newton method does **not** have any extraneous fixed point, since $H_f = 1$. In order to find the extraneous fixed points, we substitute the quadratic polynomial $z^2 - 1$ for $f(z)$ and then find the zeros of H_f. For example, Super Halley method has extraneous fixed points which are the solution of $L_f = 2$, which are (see [49]) $\pm \frac{\sqrt{3}}{3} i$.

In our previous work, we found that methods without any extraneous fixed points or those having such points on the imaginary axis perform better than others. For families of methods, we showed how to choose the parameter(s) such that the extraneous fixed points are on or close to the imaginary axis. When a method contains a weight function, we suggested a rational function as a weight function. This leads to a one-parameter family of methods. We also demonstrated that a polynomial weight function does not give as good results.

To choose the parameters in the methods, the following criterion can be used, which was developed in [16] and is defined below.

Table 1. The function H_f for each of the methods

Method	H_f
Newton	1
Halley	$1+\frac{f_n f''_n}{2(f'_n)^2-f_n f''_n}$
Euler-Chebyshev	$1+\frac{f_n f''_n}{2(f'_n)^2}$
BSC	$1+\frac{f_n f''_n}{2(f'_n)^2-2f_n f''_n}$
Super Halley	$1+\frac{1}{2}\frac{L_f}{1-L_f}$
King	$1+\frac{f(y_n)}{f_n}\frac{f_n+\beta f(y_n)}{f_n+(\beta-2)f(y_n)}$
Ostrowski	$1+\frac{f(y_n)}{f_n-2f(y_n)}$
KT4	$1+\frac{f(y_n)}{f_n}\frac{1}{(1-f(y_n)/f_n)^2}$
Maheshwari	$\left(\frac{f(y_n)}{f_n}\right)^2-\frac{f_n}{f(y_n)-f_n}$
JHIF4	$\frac{1}{2}+\frac{1}{2+3[f'(y_n)/f'_n-1]}+\frac{H_3(z_n)}{f_n}\frac{f'_n}{f'(z_n)}$
CLND	$H(\tilde{t}(x_n))$
KB74	$1+\frac{f'_n f(y_n)}{f_n-2f(y_n)}\left[\frac{-2}{f'_n}+\frac{3}{f'_n+0.001f(y_n)}\right]$
Murakami	$a_1+a_2\frac{f'_n}{f'(x_n-u_n)}+a_3\frac{f'_n}{f'(x_n+\beta u_n+\gamma w_2(x_n))}$ $+\frac{f'_n}{b_1 f'_n+b_2 f'(x_n-u_n)}$

Let $E=\{z_1,z_2,...,z_n\}$ be the set of the extraneous fixed points corresponding to the values given to the parameters. We define

$$d=\max_{z_i\in E}|Re(z_i)|. \tag{2.2}$$

We look for the parameters which attain the minimum of the function d given in (2.2).

We now quote the results obtained previously for each of the methods. For King's method we found (see [49]) that the best parameters are either $\beta=0$ (Ostrowski's method) or $\beta=3-2\sqrt{2}$.

In Chun et al. [11] it was found that the best choice for the weight function $H(t)$ in (1.15) is given by

$$H(t)=\frac{1+(2g-2c-1/2)t+gt^2}{1+(2g-2c-1)t+ct^2} \tag{2.3}$$

with one of the following combinations of the parameters c and g:

1. $c=0,\ g=0.$
2. $c=2/3,\ g=1/3.$
3. $c=0.76,\ g=0.38.$

Table 2. The function H_f for each of the methods

Method	H_f
N6	$1+\frac{f(y_n)}{f_n}\frac{f_n+\beta f(y_n)}{f_n+(\beta-2)f(y_n)}+\frac{f(z_n)}{f_n}\frac{f_n-f(y_n)}{f_n-3f(y_n)}$
CN6	$1+\frac{f(y_n)}{f_n}\frac{1}{(1-f(y_n)/f_n)^2}+\frac{f(z_n)}{f_n}\frac{1}{(1-f(y_n)/f_n-f(z_n)/f_n)^2}$
WL6	$1+\frac{f(y_n)}{f_n-2f(y_n)}+\frac{H_3(z_n)}{f_n}\frac{f'_n}{f'(z_n)}$
NIK7	$1+\frac{f(y_n)}{f_n}\frac{f_n+\beta f(y_n)}{f_n+(\beta-2)f(y_n)}+\frac{f(z_n)}{f_n}\frac{f'_n}{f[z_n,y_n]+f[z_n,y_n,x_n](z_n-y_n)}$
JHID7	$\frac{1}{2}+\frac{1}{2+3[f'(y_n)/f'_n-1]}+\frac{f'_n}{H'_3(z_n)}\frac{f(z_n)}{f_n}$
BRW7	$\frac{f_n+\beta f(y_n)}{f_n+(\beta-2)f(y_n)}+\frac{f'_n}{f_n}\frac{f(z_n)}{f[z_n,y_n]+f[z_n,x_n,x_n](z_n-y_n)}$

4. $c=1.82,\ g=0.91.$

Here we just use the first case (denoted CLND), which is basically Jarratt's fourth-order (J4) method [39]

$$x_{n+1}=x_n-\left[1-\frac{3}{2}\frac{f'(y_n)-f'_n}{3f'(y_n)-f'_n}\right]\frac{f_n}{f'_n}, \tag{2.4}$$

where y_n is given by (1.14).

For the family of methods K4 we can find the parameters α and β for any of the 4 choices of c and g above.

In our work [14] we have shown that the best parameter for Murakami's family of methods is $\gamma=-0.125$. The other parameters can be found from the following

$$\theta=\frac{16\gamma+5}{4(4\gamma+1)},\ a_1=\frac{1}{6}\left(1+\frac{4\gamma+1}{\theta}\right),\qquad a_2=\frac{1}{\theta-1}\left(\frac{1}{6}\theta-\frac{2}{3}\gamma-\frac{1}{3}\right),$$

$$a_3=\frac{2}{3}, b_1=-\frac{6\theta(\theta-1)^2}{4\gamma+1},\quad b_2=\frac{6\theta^2(\theta-1)}{4\gamma+1},\quad \beta=-\gamma-\frac{1}{2}.$$

For the family N6, Chun and Neta [23] show that $\beta=-\frac{1}{2}$ is best. We have also taken $\beta=0.08$ (denoted N6d).

For the method NIK7, we have used $\beta=3-2\sqrt{2}$ which is the optimal parameter for King's method (see [49]). We have also taken $\beta=0$ (denoted NIK7d).

3. Numerical experiments

In this section, we detail the experiments we have used with each of the methods. For some methods we have taken more than one case. All the examples have roots within a square of [-3, 3] by [-3, 3]. We have taken 601^2 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 Dell Optiplex 990 desktop computer. We then computed the average number of function evaluations required per point and the number of points requiring 40 iterations.

Example 3.1. The first example is the quadratic polynomial

$$p_1(z) = z^2 - 1, \tag{3.1}$$

whose roots are at ± 1. The best results will be when the basins are divided by the imaginary axis. We have plotted the basins for methods of order 2-5 in Figure 1, for methods of order 6-7 and SA8 in Figure 2. 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. Clearly method of lower order uses more iterations in general and one should only compare the shading for methods of the same cost ν. It seems that Halley and BSC are the best third order methods, JHIF4 is the best fourth order method, WL6 is the best sixth order and NIK7d and JHID7 are the best seventh order. For eighth order methods we have SA8 (see [19].)

Now we check Table 3 to see the average number of function-evaluations per point. Note that we have used E-C short for Euler-Chebyshev. The minimum is 8.65 function-evaluations per point on average and it is achieved by SA8 followed by methods: JHIF4, WL6, and NIK7d with 9.06 function-evaluations per point on average. The highest number (13.37) was used by Euler-Chebyshev. All other methods used 9.22–12.78 function-evaluations per point on average.

Table 3. Average number of function evaluations per point for each example (1 – 6) and each of the methods

Method	Ex1	Ex2	Ex3	Ex4	Ex5	Ex6	average
Newton	11.65	15.21	14.62	11.91	22.22	19.50	15.85
Halley	11.63	13.31	14.71	12.71	16.04	15.67	14.01
E-C	13.37	18.48	18.16	-	-	-	-
BSC	9.59	11.35	12.36	13.49	13.81	14.10	12.45
Super Halley	9.59	11.35	12.36	13.49	13.81	14.10	12.45
King	10.33	12.79	12.79	-	-	-	-
Ostrowski	9.59	11.19	11.85	10.21	14.24	13.40	11.75
KT4	10.71	13.79	13.56	11.26	19.67	17.96	14.49
Maheshwari	11.87	19.76	-	-	-	-	-
JHIF4	9.06	10.83	11.23	11.19	14.97	14.10	11.90
CLND	9.59	11.19	11.85	10.19	14.50	13.49	11.80
KB74	9.42	11.28	11.86	10.08	14.83	13.70	11.86
Murakami	10.73	13.10	13.39	11.40	17.92	16.42	13.83
N6	9.84	14.18	12.64	15.09	-	-	-
N6d	10.23	13.08	12.69	12.32	57.79	-	-
CN6	11.63	14.90	14.41	12.56	20.54	19.00	15.65
WL6	9.06	10.83	11.23	10.12	14.88	13.95	11.68
NIK7	9.22	11.32	11.53	11.27	15.71	13.60	12.11
NIK7d	9.06	10.33	11.15	11.34	13.17	14.06	11.52
JHID7	11.33	13.54	14.03	12.44	20.51	17.46	14.89
BRW7	12.78	31.83	-	-	-	-	-
SA8	8.65	9.68	10.46	10.20	12.11	11.57	10.45

Based on the CPU time in seconds, we find that the fastest method is Newton's method (105.628 seconds) followed by Super Halley (110.917 seconds). The slowest is JHIF4 with 230.803 seconds. In terms of the number of black points (see Table 5) we find that KB74 has the lowest number (3 points), most methods have 601 such points except King's method (1653 points).

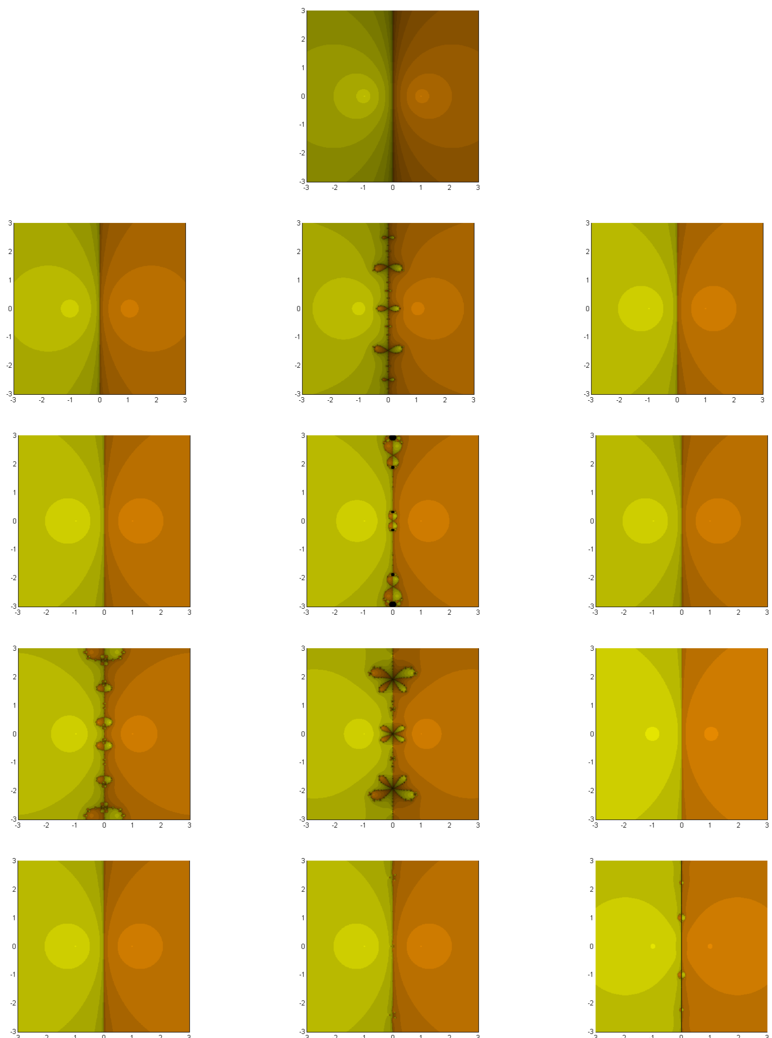

Figure 1. The top row for Newton. Second row for Halley (left), Euler-Chebyshev (center) and BSC (right). Third row for Super Halley (left), King with $\beta = 3 - 2\sqrt{2}$ (center), and Ostrowski (right). Fourth row for Kung-Traub fourth order (left), Maheshwari (center), and JHIF4 (right). Bottom row for CLND (left), KB74 (center) and Murakami (right) for the roots of the polynomial $z^2 - 1$.

Example 3.2. The second example is the cubic polynomial

$$p_2(z) = z^3 - 1, \tag{3.2}$$

having the 3 roots of unity.

The basins of attraction are given in Figures 3–4. In Figure 3 we have the basins for methods of order 2-5, in Figure 4 the basins for methods of order 6–7

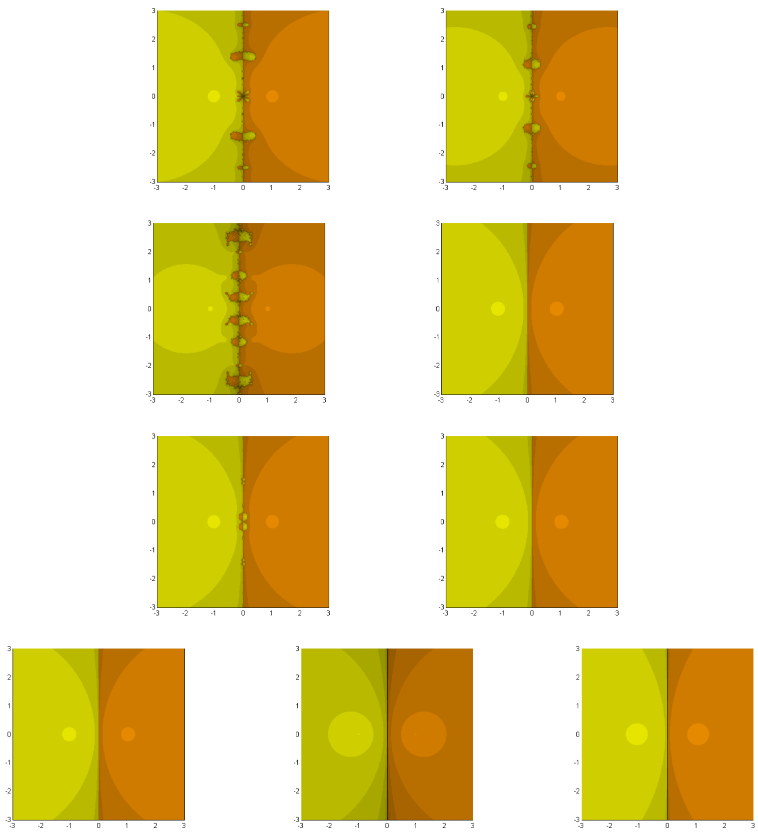

Figure 2. The top row for Neta sixth order with $\beta = -1/2$ (left) and N6d (right). Second row for CN6 (left) and WL6 (right). Third row for NIK7 (left), and NIK7d (right) and bottom row for JHID7 (left), BRW7 (center) and SA8 (right) for the roots of the polynomial $z^2 - 1$.

and SA8 are displayed. Based on these plots we find that Halley, BSC, NIK7d, and JHID7 are best. Based on Table 3 we find that the minimum number of function-evaluations per point is achieved by SA8 (9.68) followed by NIK7d (10.33), JHIF4 and WL6 (10.83). The worst (31.83) is BRW7. All the other methods use 11.19–19.76 function-evaluations per point.

The fastest method is Super Halley's method (162.319 seconds) and the slowest are Maheshwari's method (453.448 seconds) and BRW7 (813.373 seconds). Therefore, we will remove these 2 slowest methods. Based on the number of black points clearly we have BRW7, Maheshwari's method and N6 the worst.

Example 3.3. The third example is another cubic polynomial, but with real roots only, i.e. the polynomial is given by:

$$p_3(z) = z^3 - z. \tag{3.3}$$

The basins of attraction are displayed in Figures 5–6. It seems that the best methods are JHIF4, WL6, NIK7d, and JHID7. Consulting the number of function-evaluations per point, we find that the best is SA8 (10.46), followed by NIK7d (11.15), JHIF4, and WL6 (11.23). The worst is Euler-Chebyshev (18.16). All the others use 11.53–14.71 function-evaluations per point. We will exclude Euler-Chebyshev's method from now on. The fastest method is again Super Halley (175.111 seconds) followed closely by Newton (179.291 seconds). The slowest is JHIF4 (371.032 seconds). The method with the highest number of black points is King's method and it will be excluded too.

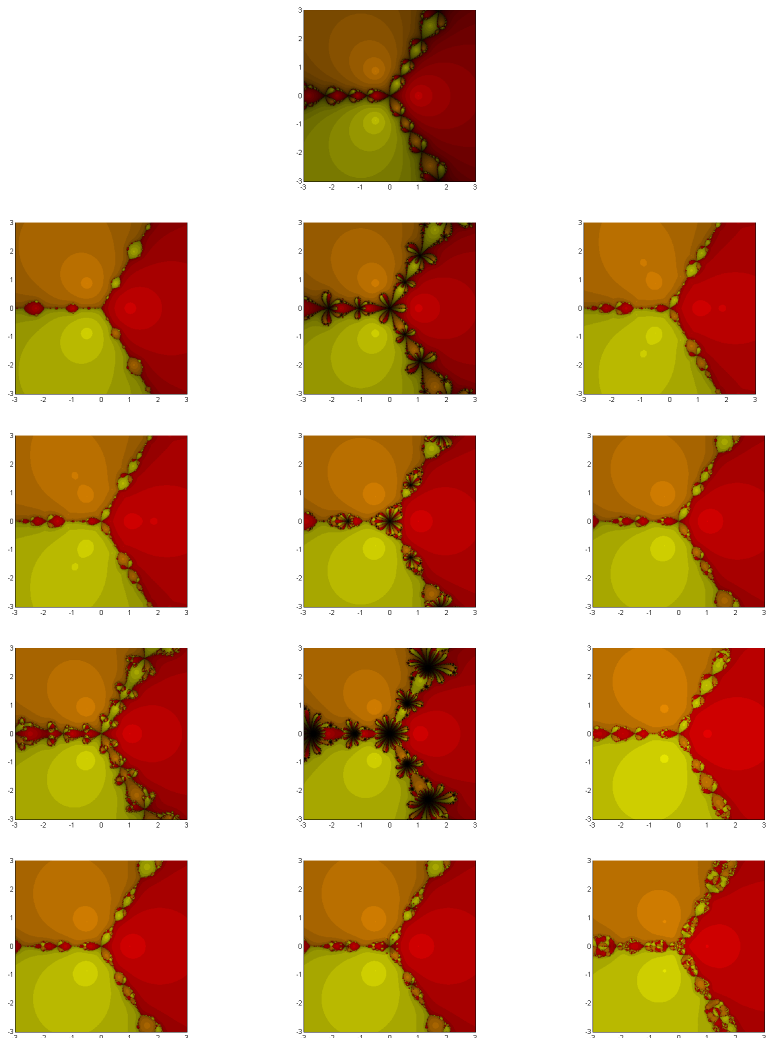

Figure 3. The top row for Newton. Second row for Halley (left), Euler-Chebyshev (center) and BSC (right). Third row for Super Halley (left), King with $\beta = 3 - 2\sqrt{2}$ (center), and Ostrowski (right). Fourth row for Kung-Traub fourth order (left), Maheshwari (center), and JHIF4 (right). Bottom row for CLND (left), KB74 (center) and Murakami (right) for the roots of the polynomial $z^3 - 1$.

Example 3.4. The fourth example is a quartic polynomial with real roots at ± 1, ± 3

$$p_4(z) = z^4 - 10z^2 + 9. \tag{3.4}$$

The basins are displayed in Figures 7–8. The best methods are Ostrowski,

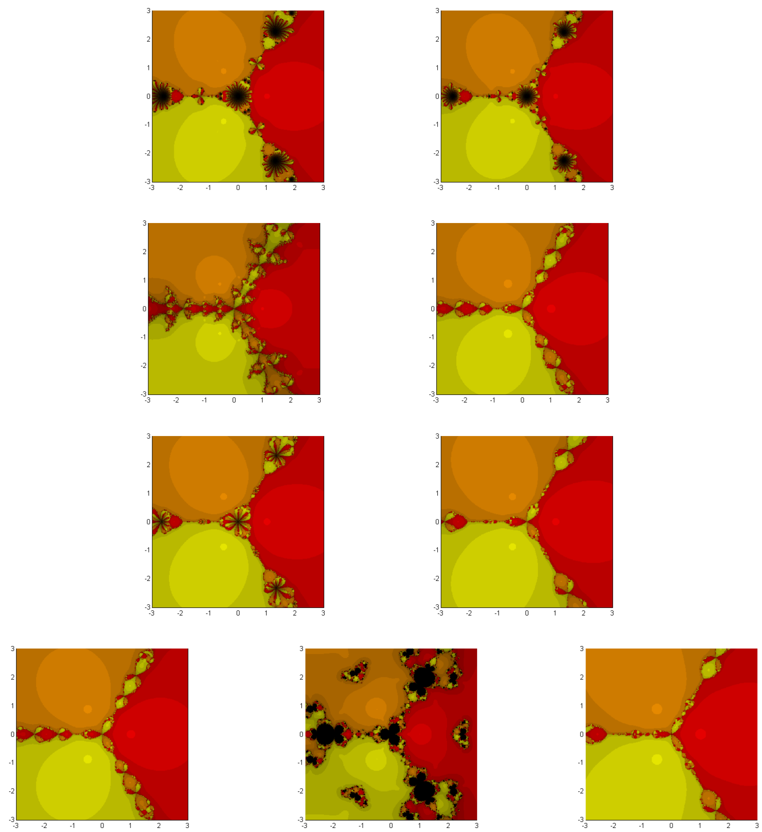

Figure 4. The top row for Neta sixth order with $\beta = -1/2$ (left) and N6d (right). Second row for CN6 (left) and WL6 (right). Third row for NIK7 (left) and NIK7d right) and bottom row for JHID7 (left), BRW7 (center) and SA8 (right) for the roots of the polynomial $z^3 - 1$.

JHIF4, CLND, Murakami, and JHID7. Based on the average number of function-evaluations per point (see Table 3) we find that the minimum is achieved by KB74 (10.08), following closely by WL6 (10.12), and CLND (10.19). The worst method in this sense is N6 which uses 15.09 function-evaluations per point on average. All other methods use between 10.20 and 13.49 function-evaluations per point on average. In terms of the CPU time, the fastest method is Newton (201.039 seconds) followed by Super Halley (247.698 seconds). The slowest is N6 with over 515 seconds and we will exclude it in the rest of the examples. All others use 285.481–475.694 seconds. Based on the number of black points, we see that most methods have 601 black points and the worst is N6 with 709 points.

Example 3.5. The fifth example is a fifth degree polynomial

$$p_5(z) = z^5 - 1. \tag{3.5}$$

The basins are displayed in Figures 9–10. It seems that the best methods are Halley, BSC, Ostrowski, JHIF4, CLND, WL6 and NIK7d. The data in Tables 3–5 give a quantitative information. Based on Table 3 we find that N6d is the worst, requiring over 57 function-evaluations per point on average. For this reason we will exclude this method in the last two examples. The smallest number of function-evaluations on average is for SA8 (12.11) followed by NIK7d (13.17), BSC and Super Halley (13.81 both). All other methods use between 14.24 and 22.22 function-evaluations per point. The fastest method is again Super Halley (274.452 seconds) and the slowest is N6d (1565.767 seconds). The rest use 359.240 – 608.513 seconds. In terms of black points, we find again that the worst are N6d (71711) and Newton (5160) then NIK7 (1445). All other methods have between 1 and 363 black points.

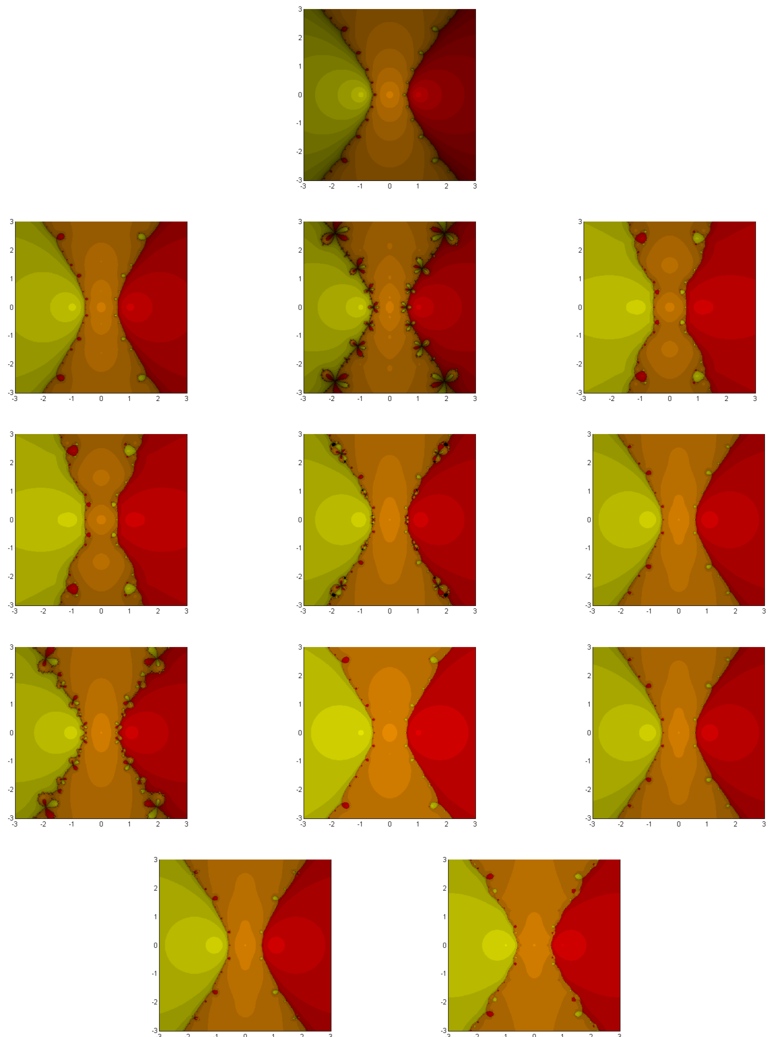

Figure 5. The top row for Newton. Second row for Halley (left), Euler-Chebyshev (center) and BSC (right). Third row for Super Halley (left), King with $\beta = 3-2\sqrt{2}$ (center), and Ostrowski (right). Fourth row for Kung-Traub fourth order (left), JHIF4 (center) and CLND (right). Bottom row for KB74 (left) and Murakami (right) for the roots of the polynomial $z^3 - z$.

Example 3.6. The next example is a polynomial of degree 6 with complex coeffi-

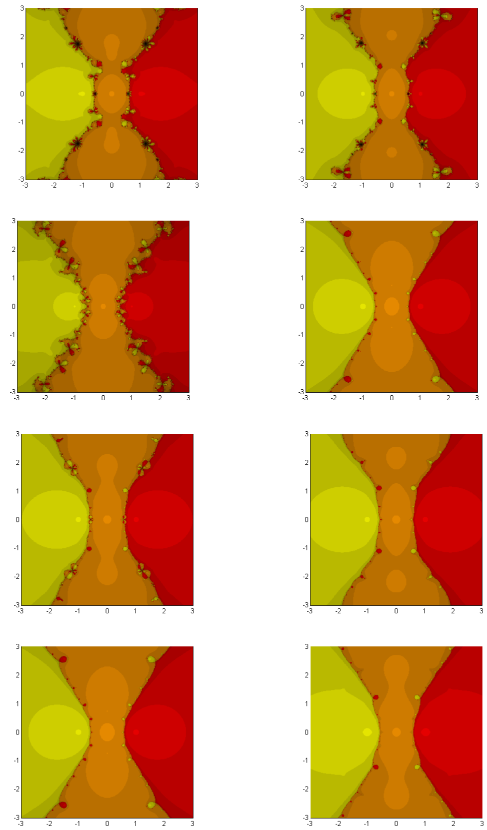

Figure 6. The top row for Neta sixth order with $\beta = -1/2$ (left) and N6d (right). Second row for CN6 (left) and WL6 (right). Third row for NIK7 (left), and NIK7d (right) and bottom row for JHID7 (left) and SA8 (right) for the roots of the polynomial $z^3 - z$.

cients

$$p_6(z) = z^6 - \frac{1}{2}z^5 + \frac{11(i+1)}{4}z^4 - \frac{3i+19}{4}z^3 + \frac{5i+11}{4}z^2 - \frac{i+11}{4}z + \frac{3}{2} - 3i. \quad (3.6)$$

This is an example that was difficult for many methods. The basins are displayed in Figure 11. The best methods seem to be Halley, Ostrowski, KT4, CLND, KB74 and Murakami. In terms of average number of function-evaluations per point, SA8 is the best method with 11.57 followed by Ostrowski (13.40), CLND (13.49), NIK7 (13.6) and KB74 (13.70). The worst is Newton with 19.50 function-evaluations per point on average. The fastest method (Table 4) is again Super Halley (695.218 seconds) and the slowest is CN6 (2183.733 seconds). 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 3–5 across the 6 examples. There was no method that performed best based on the 3 criteria used. The method with the fewest number of function-evaluations per point is SA8 (10.45) followed by NIK7d (11.52), WL6 (11.68) and Ostrowski (11.75). The fastest method is again Super Halley (277.62 seconds) followed by

Table 4. CPU time (in seconds) required for each example (1 – 6) and each of the methods

Method	Ex1	Ex2	Ex3	Ex4	Ex5	Ex6	average
Newton	105.628	195.344	179.291	201.039	388.068	893.012	377.06
Halley	174.222	281.051	316.775	381.968	462.028	1519.42	522.58
E-C	153.458	291.769	285.325	-	-	-	-
BSC	147.577	249.617	278.071	402.857	423.090	1373.293	479.08
Super Halley	110.917	162.319	175.111	247.698	274.452	695.218	277.62
King	207.465	341.845	331.361	-	-	-	-
Ostrowski	163.255	261.566	272.222	331.096	447.239	1681.66	526.17
KT4	146.578	252.752	249.851	287.463	480.748	1535.393	492.13
Maheshwari	194.346	453.448	-	-	-	-	-
JHIF4	230.803	359.114	371.032	475.694	579.543	1865.631	646.97
CLND	164.752	257.542	275.529	309.646	426.382	1216.745	441.77
KB74	177.061	295.217	307.400	351.861	479.298	1698.898	551.62
Murakami	150.229	249.742	251.676	285.481	412.982	1289.770	439.98
N6	176.765	337.695	310.660	515.225	-	-	-
N6d	174.331	290.568	303.593	413.558	1565.767	-	-
CN6	200.679	358.007	326.37	405.197	595.378	2183.733	678.23
WL6	218.51	321.565	338.819	405.602	538.593	1705.029	588.02
NIK7	197.606	311.675	311.627	404.666	514.757	1691.862	572.03
NIK7d	193.753	274.234	290.505	406.024	446.194	1720.941	555.28
JHID7	199.572	328.195	325.481	403.855	608.513	1963.459	638.18
BRW7	228.136	813.373	-	-	-	-	-
SA8	152.381	224.969	241.178	318.273	359.240	1272.032	428.01

Newton (377.06 seconds) and SA8 (428.01 seconds). The methods with the least number of black points on average are KB74 (61.83 points) followed by BSC, Super Halley, WL6, NIK7d and SA8 (200.67 points).

Another way to get the closest method to the best in all 3 categories is to look at range of values for each criterion. For example, let us take a range of 10.45–11.90 for the average number of function-evaluations per point (7 methods) and a range of 277.62–441.77 seconds for the CPU time (5 methods) and 61.83–203.17 black points on average (9 methods), then we can find that SA8 and CLND are close to best performer in all categories. NIK7d, WL6, KB74 and JHIF4 are close to best in 2 out of the 3 categories. Clearly this conclusion is based on the ranges (or tolerances) one is willing to take. The CPU range excludes all methods of order higher than 5 except SA8.

We now run a non-polynomial example.

Example 3.7.

$$p_7(z) = (e^{z+1} - 1)(z - 1). \tag{3.7}$$

The roots are ± 1 and the basins are given in Figure 12. Notice that in all but 5 methods the basin for $z = +1$ is much smaller. BSC and Super Halley have 2 basins of about the same size but there are points close to $z = +1$ that converge to $z = -1$. Similarly for SA8. The only ones of the 5 for which this is not the case are NIK7 and NIK7d. Now we can see that SA8 came in the top 4 performers in all 3 categories and Ostrowski's method in the top 5 performers in all 3 categories. CLND came in the top 4 in 2 categories, see Table 6. Newton's method was the fastest (230.179 seconds) followed by Super Halley (307.728 seconds), KT4 (311.315 seconds), SA8 (333.967 seconds) and Ostrowski (338.241 seconds). Method JHID7 had the least number of black points (469) followed by Murakami (510), SA8 (514 points), CLND (627 points) and Ostrowski (655). SA8 uses the least number of function-evaluations per point (9.12) followed by Ostrowski (9.32), CLND (9.34), KB74 (9.40) and NIK7d (9.88).

There is no best perfomer based on all 3 criteria for Example 7. SA8 came in the top 4, Ostrowski's method in the top 5 and CLND in the top 6 in all 3 criteria.

Table 5. Number of points requiring 40 iterations for each example (1 – 6) and each of the methods

Method	Ex1	Ex2	Ex3	Ex4	Ex5	Ex6	average
Newton	601	8	0	601	5160	108	1079.67
Halley	601	2	0	601	21	0	204.17
E-C	601	184	0	-	-	-	-
BSC	601	1	0	601	1	0	200.67
Super Halley	601	1	0	601	1	0	200.67
King	1653	453	928	-	-	-	-
Ostrowski	601	1	0	601	18	0	203.5
KT4	601	1	0	601	220	20	240.5
Maheshwari	601	3458	-	-	-	-	-
JHIF4	601	1	0	601	4	0	201.17
CLND	601	1	0	601	16	0	203.17
KB74	3	4	0	1	363	0	61.83
Murakami	601	1	0	609	17	0	204.67
N6	601	754	4	709	-	-	-
N6d	601	534	8	617	71711	-	-
CN6	601	1	0	605	8	0	202.5
WL6	601	1	0	601	1	0	200.67
NIK7	601	1	0	601	1445	3	441.83
NIK7d	601	1	0	601	1	0	200.67
JHID7	601	1	0	601	20	0	203.83
BRW7	601	36099	-	-	-	-	-
SA8	601	1	0	601	1	0	200.67

Conclusions. We have compared the basins of several methods of orders 2 to 7 and the best performer of the eighth order methods (SA8) using 3 quantitative measures and found that there is no best method based on all 3 criteria. If instead of picking the absolute best we allow range of values for each criterion, we found that SA8 and CLND are at the top.

For the non-polynomial example, we found that SA8 is in the top 4, Ostrowski's method in the top 5 and CLND in the top 6 in all 3 categories. We conclude that the best methods to use are SA8 and CLND.

Acknowledgements. The authors are grateful to the anonymous referees for their useful suggestions which improve the contents of this article.

Table 6. Results for Example 7

Method	number of function evaluation per point	CPU	number of black points
Newton	10.43	230.179	666
Halley	11.91	411.390	1244
BSC	13.23	451.139	3570
Super Halley	13.20	307.728	3496
Ostrowski	9.32	338.241	655
KT4	10.42	311.315	2577
JHIF4	11.41	567.656	1523
CLND	9.34	361.438	627
KB74	9.40	375.916	909
Murakami	10.69	376.914	510
CN6	13.01	459.158	5021
WL6	10.36	471.045	896
NIK7	12.24	406.507	914
NIK7d	9.88	409.284	659
JHID7	12.43	451.607	469
SA8	9.12	333.967	514

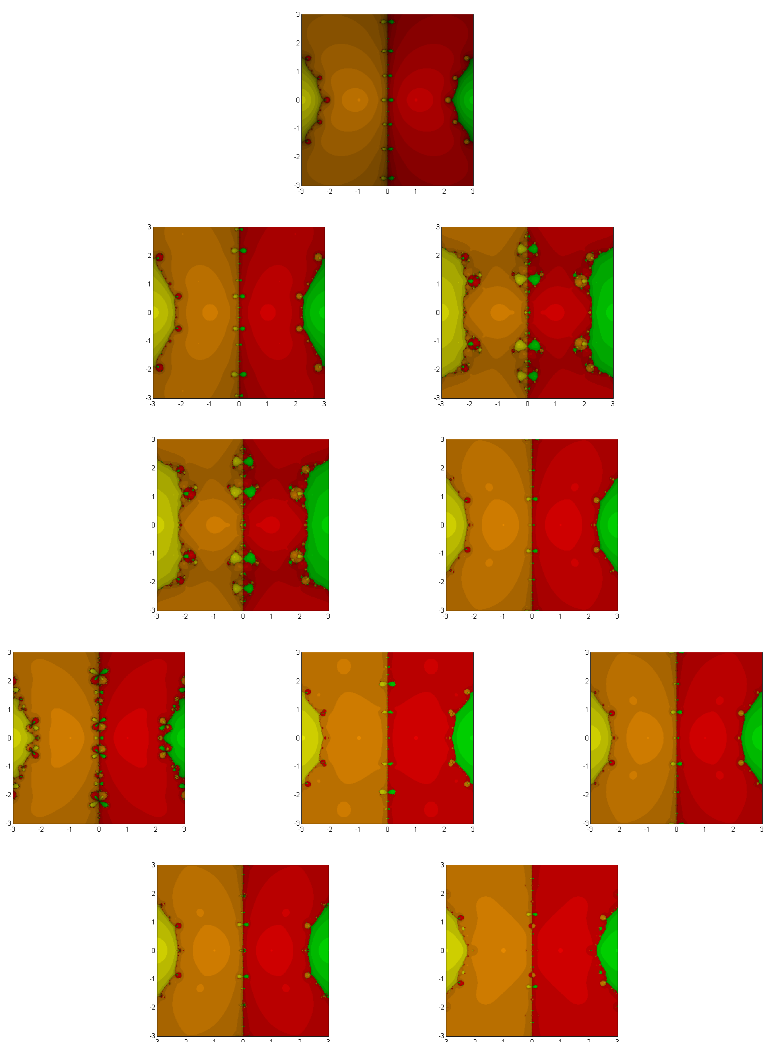

Figure 7. The top row for Newton. Second row for Halley (left) and BSC (right). Third row for Super Halley (left) and Ostrowski (right). Fourth row for Kung-Traub fourth order (left), JHIF4 (center) and CLND (right). Bottom row for KB74 (left) and Murakami (right) for the roots of the polynomial $z^4 - 10z^2 + 9$.

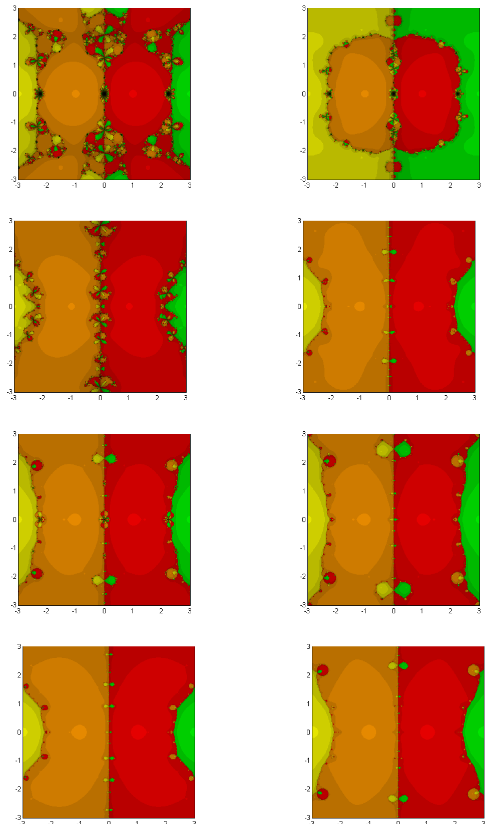

Figure 8. The top row for Neta sixth order with $\beta = -1/2$ (left) and N6d (right). Second row for CN6 (left) and WL6 (right). Third row for NIK7 (left), and NIK7d (right) and bottom row for JHID7 (left) and SA8 (right) for the roots of the polynomial $z^4 - 10z^2 + 9$.

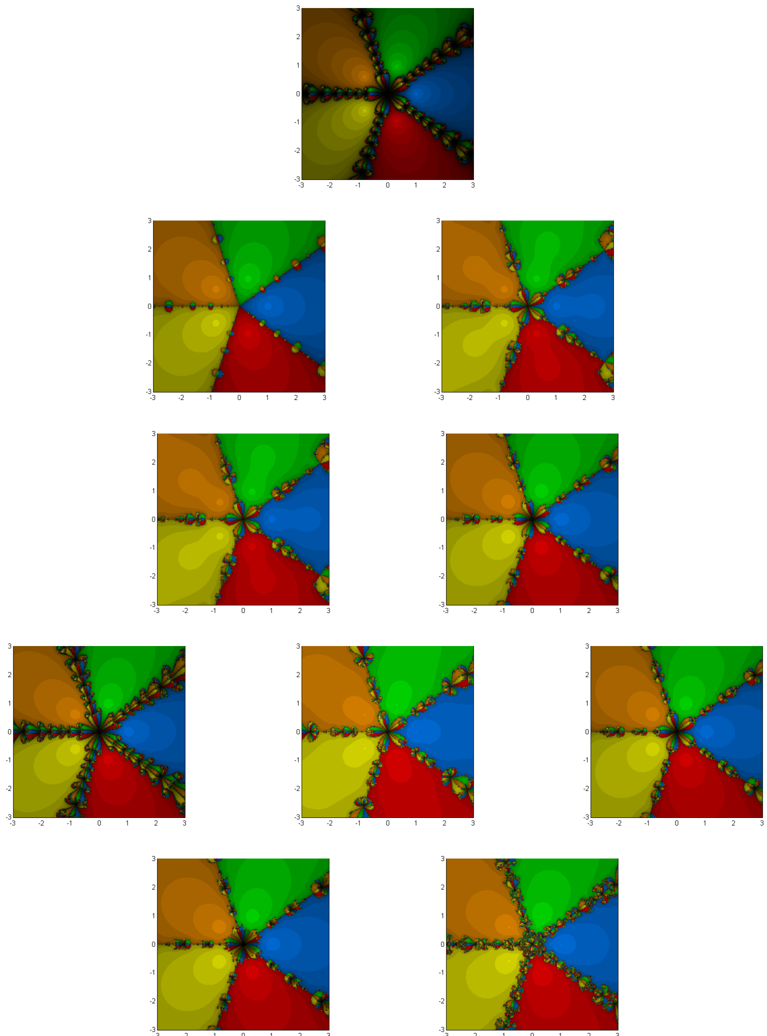

Figure 9. The top row for Newton. Second row for Halley (left) and BSC (right). Third row for Super Halley (left) and Ostrowski (right). Fourth row for Kung-Traub fourth order (left), JHIF4 (center) and CLND (right). Bottom row for KB74 (left) and Murakami (right) for the roots of the polynomial $z^5 - 1$.

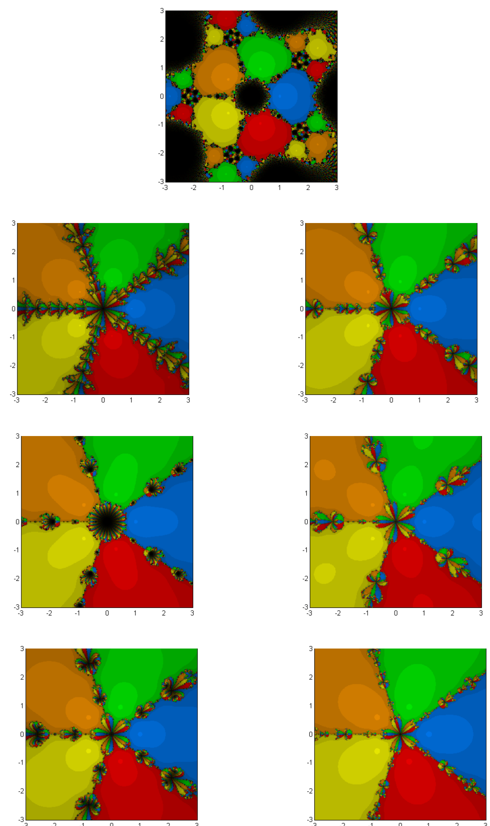

Figure 10. The top row for N6d. Second row for CN6 (left) and WL6 (right). Third row for NIK7 (left), and NIK7d (right) and bottom row for JHID7 (left) and SA8 (right) for the roots of the polynomial $z^5 - 1$.

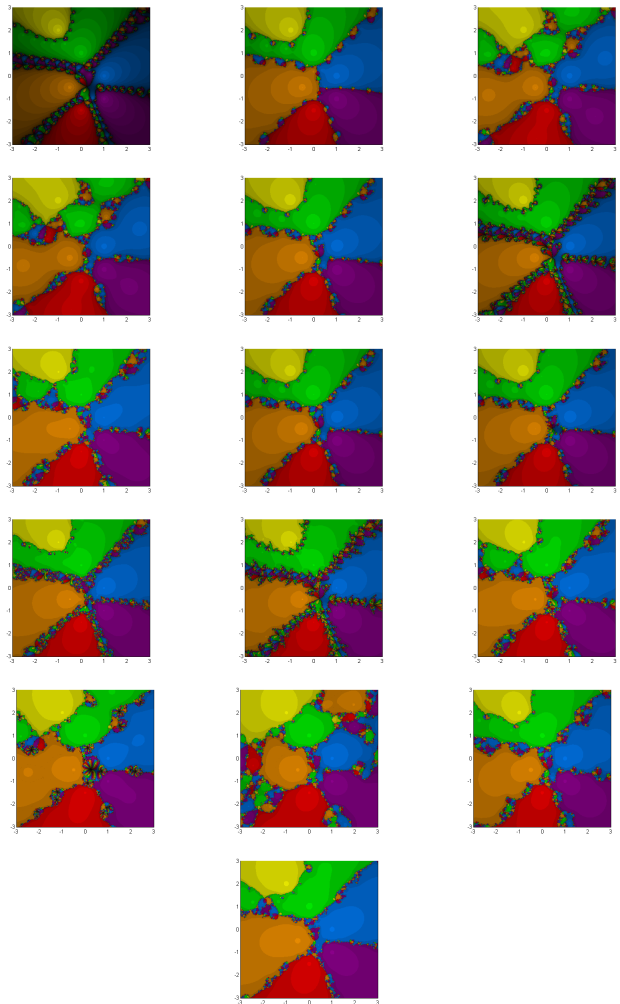

Figure 11. The top row for Newton (left) Halley (center) and BSC (right). Second row for Super Halley (left) and Ostrowski (center) and Kung-Traub fourth order (right). Third row for JHIF4 (left), CLND (center) and KB74 (right). Fourth row for Murakami (left), CN6 (center) and WL6 (right). Fifth row for NIK7 (left), NIK7d (center) and JHID7 (right) and bottom row for SA8 for the roots of the polynomial $z^6 - \frac{1}{2}z^5 + \frac{11(i+1)}{4}z^4 - \frac{3i+19}{4}z^3 + \frac{5i+11}{4}z^2 - \frac{i+11}{4}z + \frac{3}{2} - 3i$.

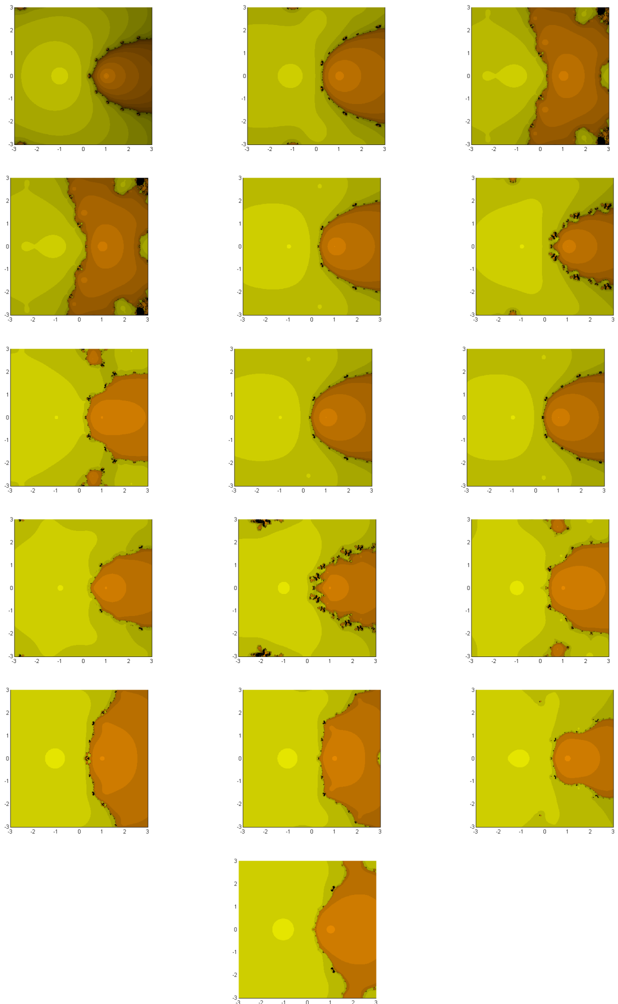

Figure 12. The top row for Newton (left), Halley (center) and BSC (right). The second row for Super Halley(left), Ostrowski (center) and KT4 (right). The third row for JHIF4 (left), CLND (center) and KB74 (right). The fourth row for Murakami (left), CN6 (center) and WL6 (right). The fifth row for NIK7 (left), NIK7d (center) and JHID7 (right). The bottom row for SA8 for the roots of the function $(e^{z+1} - 1)(z - 1)$.

References

[1] S. Amat, S. Busquier and S. Plaza, *Review of some iterative root-finding methods from a dynamical point of view*, Scientia, 2004, 10, 3–35.

[2] S. Amat, S. Busquier and S. Plaza, *Dynamics of a family of third-order iterative methods that do not require using second derivatives*, Appl. Math. Comput., 2004, 154, 735–746.

[3] S. Amat, S. Busquier and S. Plaza, *Dynamics of the King and Jarratt iterations*, Aeq. Math., 2005, 69, 212–2236.

[4] I. K. Argyros and A. A. Magreñan, *On the convergence of an optimal fourth-*

order family of methods and its dynamics, Appl. Math. Comput., 2015, 252, 336–346.

[5] D. K. R. Babajee, A. Cordero, F. Soleymani and J. R. Torregrosa, *On improved three-step schemes with high efficiency index and their dynamics*, Numer. Algorithms, 2014, 65, 153–169.

[6] M. Basto, V. Semiao and F. Calheiros, *A new iterative method to compute nonlinear equations*, Appl. Math. Comput., 2006, 173, 468–483.

[7] W. Bi, H. Ren and Q. Wu, *New family of seventh-order methods for nonlinear equations*, Appl. Math. Comput., 2008, 203, 408–412.

[8] W. Bi, H. Ren and Q. Wu, *Three-step iterative methods with eight-order convergence for solving nonlinear equations*, J. Comput. Appl. Math., 2009, 225, 105–112.

[9] W. Bi, Q. Wu and H. Ren, *A new family of eight-order iterative methods for solving nonlinear equations*, Appl. Math. Comput. 2009, 214, 236–245.

[10] C. Chun, M. Y. Lee, B. Neta and J. Džunić, *On optimal fourth-order iterative methods free from second derivative and their dynamics*, Appl. Math. Comput., 2012, 218, 6427–6438.

[11] C. Chun, B. Neta and Sujin Kim, *On Jarratt's family of optimal fourth-order iterative methods and their dynamics*, Fractals, 2014. DOI:10.1142/S0218348X14500133.

[12] C. Chun and B. Neta, *On the new family of optimal eighth order methods developed by Lotfi et al.*, Numer. Algorithms, 2016, 72, 363–376.

[13] C. Chun and B. Neta, *Comparison of several families of optimal eighth order methods*, Appl. Math. Comput., 2016, 274, 762–773.

[14] C. Chun and B. Neta, *The basins of attraction of Murakami's fifth order family of methods*, Appl. Numer. Math., 2016, 110, 14–25.

[15] C. Chun and B. Neta, *An analysis of a new family of eighth-order optimal methods*, Appl. Math. Comput., 2014, 245, 86–107.

[16] C. Chun and B. Neta, An analysis of a King-based family of optimal eighth-order methods, Amer. J. Algorithms and Computing, 2015, 2, 1–17.

[17] C. Chun, B. Neta, J. Kozdon and M. Scott, *Choosing weight functions in iterative methods for simple roots*, Appl. Math. Comput., 2014, 227, 788–800.

[18] C. Chun and B. Neta, *Basin of attraction for several third order methods to find multiple roots of nonlinear equations*, Appl. Math. Comput., 2015, 268, 129–137.

[19] C. Chun and B. Neta, *Comparative study of eighth order methods for finding simple roots of nonlinear equations*, Numer. Algorithms, 2017, 74(4), 1169C1201.

[20] C. Chun and B. Neta, *Comparing the basins of attraction for Kanwar-Bhatia-Kansal family to the best fourth order method*, Appl. Math. Comput., 2015, 266, 277–292.

[21] C. Chun and Y. Ham, *Some second-derivative-free variants of super-Halley method with fourth-order convergence*, Appl. Math. Comput., 2008, 195, 537–541.

[22] C. Chun and B. Neta, *An analysis of a Kharrri's 4th order family of methods*, Appl. Math. Comput., 2016, 279, 198–207.

[23] C. Chun and B. Neta, *A new sixth-order scheme for nonlinear equations*, Appl. Math. Lett., 2012, 25, 185–189.

[24] C. Chun and B. Neta, *An analysis of a family of Maheshwari-based optimal eighth-order methods*, Appl. Math. Comput., 2015, 253, 294–307.

[25] C. Chun and M. Y. Lee, *A new optimal eighth-order family of iterative methods for the solution of nonlinear equations*, Appl. Math. Comput., 2013, 223, 506–519.

[26] F. Chicharro, A. Cordero, J. M. Gutiérrez and J. R. Torregrosa, *Complex dynamics of derivative-free methods for nonlinear equations*, Appl. Math. Comput., 2013, 219, 7023–7035.

[27] A. Cordero, J.,García-Maimó, J. R. Torregrosa, M. P. Vassileva and P. Vindel, *Chaos in King's iterative family*, Appl. Math. Lett., 2013, 26, 842–848.

[28] A. Cordero, M. Fardi, M. Ghasemi and J. R. Torregrosa, *Accelerated iterative methods for finding solutions of nonlinear equations and their dynamical behavior*, Calcolo, 2014, 51, 17–30.

[29] A. Cordero, J. Torregrosa and M. Vassileva, *Three-step iterative methods with optimal eighth-order convergence*, J. Comput. Appl. Math., 2011, 235, 3189–3194.

[30] Y. H. Geum, Y. I. Kim and B. Neta, *On developing a higher-order family of double-Newton methods with a bivariate weighting function*, Appl. Math. Comput., 2015, 254, 277–290.

[31] Y. H. Geum, Y. I. Kim and B. Neta, *A class of two-point sixth-order multiple-zero finders of modified double-Newton type and their dynamics*, Appl. Math. Comput., 2015, 270, 387–400.

[32] Y. H. Geum, Y. I. Kim and B. Neta, *A family of optimal quartic-order multiple-zero finders with a weight function of the principal k^{th} root of a derivative-to-derivative ratio and their basins of attraction*, Math. Comput. Simulat., 2017, 136, 1–21.

[33] Y. H. Geum, Y. I. Kim, A and A. Magreñan, *A biparametric extension of Kings fourth-order methods and their dynamics*, Appl. Math. Comput., 2016, 282, 254–275.

[34] J. M. Gutiérrez and M. A. Hernández, *An acceleration of Newton's method: Super-Halley method*, Appl. Math. Comput., 2001, 117, 223–239.

[35] E. Halley, *A new, exact and easy method of finding the roots of equations generally and that without any previous reduction*, Phil. Trans. Roy. Soc. London, 1694, 18, 136–148.

[36] E. Hansen and M. Patrick, *A family of root finding methods*, Numer. Math., 1977, 27, 257–269.

[37] A. S. Householder, *The Numerical Treatment of a Single Equation*, McGraw-Hill, New York, NY, 1970.

[38] P. Jarratt, *Some efficient fourth-order multipoint methods for solving equations*, BIT, 1969, 9, 119–124.

[39] P. Jarratt, *Some fourth-order multipoint iterative methods for solving equations*, Math. Comput., 1966, 20, 434–437.

[40] Y. Khan, M. Fardi and K. Sayevand, *A new general eighth-order family of iterative methods for solving nonlinear equations*, App. Math. Lett., 2012, 25, 2262–2266.

[41] S. K. Khattri and D. K. R. Babajee, *Fourth-order family of iterative methods with four parameters*, International J. of Mathematics and Computation, 2013, 19, 1–10.

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

[43] J. Kou, Y. Li and X. Wang, *Fourth-order iterative methods free from second derivative*, Appl. Math. Comput., 2007, 184, 880–885.

[44] H. T. Kung and J. F. Traub, *Optimal order of one-point and multipoint iterations*, J. Assoc. Comput. Mach., 1974, 21, 643–651.

[45] T. Lotfi, S. Sharifi, M. Salimi and S. Siegmund, *A new class of three-point methods with optimal convergence order eight and its dynamics*, Numer. Algorithms, 2015, 68, 261–288.

[46] A. A. Magreñan, *Different anomalies in a Jarratt family of iterative root-finding methods*, Appl. Math. Comput., 2014, 233, 29–38.

[47] A. K. Maheshwari, *A fourth-order iterative method for solving nonlinear equations*, Appl. Math. Comput., 2009, 211, 383–391.

[48] T. Murakami, *Some fifth order multipoint iterative formulae for solving equations*, J. of Information Processing, 1978, 1, 138–139.

[49] B. Neta, M. Scott and C. Chun, *Basin of attractions for several methods to find simple roots of nonlinear equations*, Appl. Math. Comput., 2012, 218, 10548–10556.

[50] B. Neta, C. Chun and M. Scott, *Basins of attractions for optimal eighth order methods to find simple roots of nonlinear equations*, Appl. Math. Comput., 2014, 227, 567–592.

[51] B. Neta, M. Scott and C. Chun, *Basin attractors for various methods for multiple roots*, Appl. Math. Comput., 2012, 218, 5043–5066.

[52] B. Neta and C. Chun, *On a family of Laguerre methods to find multiple roots of nonlinear equations*, Appl. Math. Comput., 2013, 219, 10987–11004.

[53] B. Neta and C. Chun, *Basins of attraction for several optimal fourth order methods for multiple roots*, Math. Comput. Simulat., 2014, 103, 39–59.

[54] B. Neta and C. Chun, *Basins of attraction for Zhou-Chen-Song fourth order family of methods for multiple roots*, Math. Comput. Simulat., 2015, 109, 74–91.

[55] B. Neta, C. Chun and M. Scott, *A note on the modified super-Halley method*, Appl. Math. Comput., 2012, 218, 9575–9577.

[56] B. Neta, *A sixth order family of methods for nonlinear equations*, Int. J. Computer Math., 1979, 7, 157–161.

[57] B. Neta and A. N. Johnson, *High order nonlinear solver*, J. Comput. Methods Science and Engineering, 2008, 8(4-6), 245–250.

[58] B. Neta and M. S. Petković, *Construction of optimal order nonlinear solvers using inverse interpolation*, Appl. Math. Comput., 2010, 217, 2448–2455.

[59] B. Neta, *On a family of multipoint methods for nonlinear equations*, Int. J. Computer Math., 1981, 9, 353–361.

[60] A. M. Ostrowski, *Solution of Equations and Systems of Equations*, 3rd ed. Academic Press, New York London, 1973.

[61] M. S. Petković, B. Neta, L. D. Petković, J. Džunić, *Multipoint Methods for Solving Nonlinear Equations*, Elsevier, 2013.

[62] D. B. Popovski, *A family of one point iteration formulae for finding roots*, Int. J. Computer Math., 1980, 8, 85–88.

[63] M. Scott, M, B. Neta and C. Chun, *Basin attractors for various methods*, t Appl. Math. Comput., 2011, 218, 2584–2599.

[64] J. R. Sharma and R. Sharma, *A new family of modified Ostrowski's methods with accelerated eighth order convergence*, Numer. Algorithms, 2010, 54, 445–458.

[65] J. R. Sharma and H. Arora, *A new family of optimal eighth order methods with dynamics for nonlinear equations*, Appl. Math. Comput., 2016, 273, 924–933.

[66] B. D. Stewart, *Attractor Basins of Various Root-Finding Methods*, M.S. thesis, Naval Postgraduate School, Department of Applied Mathematics, Monterey, CA, June 2001.

[67] R. Thukral and M. S. Petković, *Family of three-point methods of optimal order for solving nonlinear equations*, J. Comput. Appl. Math., 2010, 233, 2278–2284.

[68] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1964.

[69] X. Wang and L. Liu, *Modified Ostrowski's method with eighth-order convergence and high efficiency index*, Appl. Math. Lett., 2010, 23, 549–554.

[70] X. Wang and L. Liu, *New eighth-order iterative methods for solving nonlinear equations*, J. Comput. Appl. Math., 2010, 234, 1611–1620.