Communications in Numerical Analysis 2018 No.2 (2018) 111-130

Available online at www.ispacs.com/cna

Volume 2018, Issue 2, Year 2018 Article ID cna-00362, 20 Pages

doi:10.5899/2018/cna-00362

Research Article

Dynamics of some one-point third-order methods for the solution of nonlinear equations

Beny Neta*

Naval Postgraduate School, Department of Applied Mathematics Monterey, CA 9394

Abstract

In this paper we have considered 32 one-point methods of cubic order to obtain simple zeros of a nonlinear function. These schemes are constructed by decomposition of previously known schemes. We have used the idea of basins of attractions to compare the performance of these methods with Halley's method on 4 polynomial functions and one non-polynomial function. Based on 3 quantitative criteria, namely average number of iterations per point, CPU time required and the number of points for which the method diverge, we have found 4 methods that performed close to best. We also show that decomposing good methods does not necessarily lead to a better one or even to a scheme as good as the original. We found only one example that gave reasonable results and it is the only one with repelling extraneous fixed points on the imaginary axis.

Keywords: Iterative schemes, Nonlinear functions, Simple zeros, Extraneous fixed points, Basin of attraction.

1 Introduction

There are many one-point third-order iterative procedure to find the simple roots of a single nonlinear function. The most famous and most rediscovered is Halley's method [1], see also [2]. The dynamics of this method can be found in [3]. Other well known methods are Euler's [4], Cauchy's [4], Laguerre's [5], Ostrowski's [6], Hansen and Patrick's [7], Popovski's [8, 9, 10], Milovanović and Djordjević's [11] and Neta [12]. Popovski [13] described a way to generate one-point iterative methods starting with the quadratics approximation

$$u + h + A_2 h^2 = 0 \tag{1.1}$$

where

$$h = x_{n+1} - x_n$$
$$u = \frac{f(x_n)}{f'(x_n)}$$
$$A_k = \frac{f^{(k)}(x_n)}{k! f'(x_n)}$$

*Corresponding author. Email address: bneta@nps.ed, Tel:+1 (831) 656-2235

Clearly if we have taken the linear approximation, i.e. neglecting the h^2 term, we get Newton's equation,

$$h = -u. \tag{1.2}$$

Solving (1.1) for h we get Cauchy's method

$$h = \frac{-2u}{1-\sqrt{1-4uA_2}}. \tag{1.3}$$

Alternatively, we rewrite (1.1) as

$$h + u + A_2\bar{h}\hat{h} = 0 \tag{1.4}$$

where $\bar{h}$ and $\hat{h}$ can be chose as some other approximation, say h or $-u$ or from previously found method. For example, if we choose $\bar{h} = h$ and $\hat{h} = -u$ then we get Halley's method

$$h = \frac{u}{uA_2 - 1}. \tag{1.5}$$

If one chooses $\bar{h} = \hat{h} = -u$, we have Euler's equation

$$h = -u(uA_2 + 1). \tag{1.6}$$

Popovski [13] has developed 9 new methods based on this idea. We list these methods along with the approximation for $\bar{h}$ and $\hat{h}$. The methods along with the approximations used are:

- $\bar{h} = h$ and $\hat{h}$ given by (1.5)
$$h = -\frac{u(uA_2 - 1)}{2uA_2 - 1}. \tag{1.7}$$

- $\bar{h} = h$ and $\hat{h}$ given by (1.6)
$$h = \frac{u}{(uA_2 + 1)uA_2 - 1}. \tag{1.8}$$

- $\bar{h} = h$ and $\hat{h}$ given by (1.7)
$$h = \frac{u(2uA_2 - 1)}{(uA_2 - 3)uA_2 + 1}. \tag{1.9}$$

- $\bar{h} = h$ and $\hat{h}$ given by (1.9)
$$h = -\frac{u[(uA_2 - 3)uA_2 + 1]}{3[(uA_2 - 4)uA_2 + 1]}. \tag{1.10}$$

- $\bar{h} = -u$ and $\hat{h}$ given by (1.7)
$$h = -u\frac{(uA_2 + 1)uA_2 - 1}{2uA_2 - 1}. \tag{1.11}$$

- $\bar{h} = -u$ and $\hat{h}$ given by (1.8)
$$h = u\left(\frac{uA_2}{(uA_2 + 1)uA_2 - 1} - 1\right). \tag{1.12}$$

- $\bar{h} = -u$ and $\hat{h}$ given by (1.9)
$$h = \frac{u[(uA_2 + 2)uA_2 - 1]}{(uA_2 - 3)uA_2 + 1}. \tag{1.13}$$

- $\bar{h}$ and $\hat{h}$ are both given by (1.5)
$$h = -u\left[\frac{uA_2}{(uA_2 - 1)^2} + 1\right]. \tag{1.14}$$

- $\bar{h}$ given by (1.5) and $\hat{h}$ given by (1.6)

$$h = u\frac{(uA_2)^2+1}{uA_2-1}. \tag{1.15}$$

Neta [12] has constructed 21 other methods using a combination for $\bar{h}$ and $\hat{h}$ as one of (1.5), (1.6) and

$$h = -u[(2uA_2+1)uA_2+1]. \tag{1.16}$$

We itemize those methods here and suggest two more.

- $\bar{h} = h$ and $\hat{h}$ given by (1.16)

$$h = -\frac{u}{1-uA_2(1+uA_2(1+2uA_2))} \tag{1.17}$$

- $\bar{h} = h$ and $\hat{h}$ given by (1.8)

$$h = -\frac{u}{1+uA_2/[(1+uA_2)uA_2-1]} \tag{1.18}$$

- $\bar{h} = -u$ and $\hat{h}$ given by (1.5)

$$h = -u - u^2A_2[(1+2uA_2)uA_2+1] \tag{1.19}$$

- $\bar{h} = -u$ and $\hat{h}$ given by (1.6)

$$h = -u - u^2A_2(1+uA_2) \tag{1.20}$$

- $\bar{h} = -u$ and $\hat{h}$ given by (1.16)

$$h = -u - u^2A_2[(1+2uA_2)uA_2+1] \tag{1.21}$$

- $\bar{h}$ given by (1.5) and $\hat{h}$ given by (1.16)

$$h = -u + u^2A_2\frac{(1+2uA_2)uA_2+1}{uA_2-1} \tag{1.22}$$

- $\bar{h}$ given by (1.5) and $\hat{h}$ given by (1.8)

$$h = -u - \frac{u^2A_2}{(uA_2-1)[(uA_2+1)uA_2-1]} \tag{1.23}$$

- $\bar{h}$ given by (1.5) and $\hat{h}$ given by (1.9)

$$h = -u - \frac{u^2A_2}{uA_2-1}\frac{2uA_2-1}{(uA_2-3)uA_2+1} \tag{1.24}$$

- $\bar{h}$ given by (1.6) and $\hat{h}$ given by (1.6)

$$h = -u - u^2A_2(1+uA_2)^2 \tag{1.25}$$

- $\bar{h}$ given by (1.6) and $\hat{h}$ given by (1.16)

$$h = -u - u^2A_2(uA_2+1)[(1+2uA_2)uA_2+1] \tag{1.26}$$

- $\bar{h}$ given by (1.6) and $\hat{h}$ given by (1.7)

$$h = -u - \frac{u^2A_2(uA_2+1)(uA_2-1)}{2uA_2-1} \tag{1.27}$$

- $\bar{h}$ given by (1.6) and $\hat{h}$ given by (1.9)

$$h = -u + \frac{u^2 A_2 (uA_2 + 1)(2uA_2 - 1)}{(uA_2 - 3)uA_2 + 1} \tag{1.28}$$

- $\bar{h}$ given by (1.16) and $\hat{h}$ given by (1.16)

$$h = -u - u^2 A_2 [(2uA_2 + 1)uA_2 + 1]^2 \tag{1.29}$$

- $\bar{h}$ given by (1.16) and $\hat{h}$ given by (1.7)

$$h = -u - \frac{u^2 A_2 (uA_2 - 1)[(2uA_2 + 1)uA_2 + 1]}{2uA_2 - 1} \tag{1.30}$$

- $\bar{h}$ given by (1.16) and $\hat{h}$ given by (1.8)

$$h = -u + \frac{u^2 A_2 [(2uA_2 + 1)uA_2 + 1]}{(uA_2 + 1)uA_2 - 1} \tag{1.31}$$

- $\bar{h}$ given by (1.16) and $\hat{h}$ given by (1.9)

$$h = -u + \frac{u^2 A_2 (2uA_2 - 1)[(2uA_2 + 1)uA_2 + 1]}{(uA_2 - 3)uA_2 + 1} \tag{1.32}$$

- $\bar{h}$ given by (1.7) and $\hat{h}$ given by (1.7)

$$h = -u - A_2 \frac{u^2 (uA_2 - 1)^2}{(2uA_2 - 1)^2} \tag{1.33}$$

- $\bar{h}$ given by (1.7) and $\hat{h}$ given by (1.8)

$$h = -u + \frac{u^2 A_2 (uA_2 - 1)}{(2uA_2 - 1)[(uA_2 + 1)uA_2 - 1]} \tag{1.34}$$

- $\bar{h}$ given by (1.8) and $\hat{h}$ given by (1.8)

$$h = -u - A_2 \left(\frac{u}{(uA_2 + 1)uA_2 - 1)} \right)^2 \tag{1.35}$$

- $\bar{h}$ given by (1.8) and $\hat{h}$ given by (1.9)

$$h = -u - \frac{u^2 A_2 (2uA_2 - 1)}{[(uA_2 + 1)uA_2 - 1][(uA_2 - 3)uA_2 + 1]} \tag{1.36}$$

- $\bar{h}$ given by (1.9) and $\hat{h}$ given by (1.9)

$$h = -u - A_2 \left(\frac{u(2uA_2 - 1)}{(uA_2 - 3)uA_2 + 1)} \right)^2 \tag{1.37}$$

We now list the new methods

- $\bar{h} = h$ and $\hat{h}$ given by (1.11)

$$h = \frac{u(2uA_2 - 1)}{uA_2[(uA_2 + 1)uA_2 - 3] + 1} \tag{1.38}$$

- $\bar{h}$ is given by (1.7) and $\hat{h}$ given by (1.11)

$$h = -u - \frac{u^2 A_2 (uA_2 - 1)[(uA_2 + 1)uA_2 - 1]}{(2uA_2 - 1)^2} \tag{1.39}$$

Until recently methods were compared by using the idea of efficiency index and/or by showing how a method performs using one or two initial points. We have introduced following the work of Stewart [14] and [15], [16], and [17] a visual method based on the basin of attraction. Here one take a large number of uniformly distributed points in a square containg the zeros of the function and showing how fast a method converge and to which zero from any of the initial points. See [18] – [31]. Chun et al. [32] have shown how to choose a weight function and the parameters of a method.

2 Extraneous fixed points

Clearly any method can be written as

$$x_{n+1} = x_n - H_f \frac{f_n}{f'_n}, \tag{2.40}$$

where the function H_f depends on x_n and other intermediate values. If $f(x_n)$ is a zero then x_n is a fixed point of the iterative method (2.40), but x_n is a fixed point if H_f vanishes. These latter fixed points are called extraneous fixed points (EFPs). 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. In Table 1 –2 we list the extraneous fixed points for each of the methods discussed here. Notice that(1.7) is the only method with purely imaginary extraneous fixed points. The other methods have real or complex extraneous fixed points. We will show that this method performed better than all the others.

3 Numerical experiments

In this section, we give the results for five numerical experiments using each of the methods. All the examples have zeros within a 6 by 6 square centered at the origin. We use 600 by 600 uniformly distributed points in the square as initial points for the schemes and listed the total number of iterations required to converge to which zero. We have also tabulated the number of points for which the iterative procedure did not converge in 40 (the maximum number allowed) iterations and the CPU time (in seconds) required to run each method on all the points using Dell Optiplex 990 desktop computer.

Example 3.1. *The first example is the quadratic polynomial*

$$p_1(z) = z^2 - 1. \tag{3.41}$$

The best method is the one for which the basins are divided by the imaginary axis. We started with this example, because the EFPs are computed based on it. We have plotted the basins in Figures 1 - 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 (40 in our experiments) we color the point black. The best methods are (1.7), (1.9), (1.18) and (1.5). The worst is (1.10) haveing most of the domain black.

Based on Table 3 to see the average number of iteration per point (ANIP). The minimum is 2.74 for (1.37) followed closely by (1.9) (2.84) and (1.18) with 3.01 ANIP. The highest number (37.70) was used by (1.10). All other methods used 3.09 – 6.26 function evaluations per point on average.

Based on Table 4, we find that the fastest method is (1.9) with 113.772 seconds followed by (1.7) with 115.285 seconds. In terms of the number of black points (see Table 5) we find that most methods have 601 such points. The highest number of black points are in (1.10) (334379 points), followed by (1.17) with 31121 points, (1.8) with 10289 points and (1.38) with 7113 points. Therefore, we will not experiment with (1.10) and (1.17).

Example 3.2. *The second example is the cubic polynomial*

$$p_2(z) = z^3 - 1. \tag{3.42}$$

The basins of attraction are displayed in Figures 3 – 4. The best methods seem to be (1.7) and (1.13). The worst are (1.19), (1.20), (1.21), (1.25), (1.26), (1.29) and (1.37). The last one does not have many black point, but it is more chaotic and therefore will be excluded from further study. Consulting Table 3, we find that (1.7) uses the least ANIP (3.78) followed by (1.24) with 3.80. The fastest is (1.7) using 180.04 seconds followed by (1.5) with 180.618 seconds. and the slowest are (1.21) using 336.79 seconds and (1.19) (336.135 seconds). We will eliminate these slow ones from further considerations. We will also exclude (1.20) and (1.25) from further consideration. In terms of black points, there are several method with less than 10 points, namely (1.5), (1.7), (1.9), (1.12)–(1.14), (1.18), (1.23), (1.24), (1.28), (1.33) – (1.36) and (1.39). The worst are (1.38) and (1.8). We will exclude these two from the comparative study.

Example 3.3. *The third example is a quartic polynomial with roots* ± 1 *and* $\pm i$ *given by:*

$$p_3(z) = z^4 - 1. \tag{3.43}$$

The basins of attraction are displayed in Figure 5. Based on this figure it is clear that the worst are (1.9), (1.13), (1.18), (1.23) and (1.28) and should be excluded from further consideration. The fastest method is (1.7) using 235.14 seconds followed by (1.5) using 246.84 seconds and (1.12) (279.695 seconds) and the slowest are (1.30) (1742.858 seconds), (1.36) (1731.689 seconds), (1.24) (1540.156 seconds), (1.28) (1530.142 seconds) and (1.22) (1522.835 seconds). We will exclude all these slow methods. In terms of ANIP, we find that (1.7) is best (4.48 iterations) and the worst are (1.9) (28.27 iterations), (1.30) 928.09 iterations) and (1.36) (27.80 iterations). These three methods are also having the highest number of black points. The lowest number of black points is 1201 for the methods (1.5), (1.7), (1.12), (1.23), (1.32), (1.34) and (1.35).

Example 3.4. *The fourth example is a quintic polynomial with 5 roots of unity.*

$$p_4(z) = z^5 - 1. \tag{3.44}$$

The basins for the remaining 11 methods are displayed in Figure 6. Based on Table 3, (1.7) is best with 4.6 iterations and (1.33) is the worst with 16.28 iterations. The fastest is (1.7) requiring 283.336 seconds and the slowest is (1.33) using 1065.393 seconds. There is only one method with 1 black point, namely (1.7). The highest number of black points, 23400, is for (1.33). Clearly this method will not be used in our last experiment.

Example 3.5. *The fifth example is a non-polynomial function with roots* ± 1

$$p_5(z) = (e^{z+1} - 1)(z - 1). \tag{3.45}$$

The basins for the remaining 10 methods are displayed in Figure 7. Now we can see that most methods prefer the root $z = -1$ *coming from the first factor with the exponential. From Tables 3 – 4 we find that (1.11) requires the lowest ANIP (3.93) and is second fastest (300.599 seconds) after (1.5) (263.314 seconds) and (1.35) requires the highest ANIP (6.26) and is the slowest (455.663 seconds).*

In order to find the best method, we have averaged all these results across the 5 examples. Method (1.7) is best in terms of number of function evaluations and second best in CPU time but in third place in the number of black points. Halley's method (1.5) came first in CPU and lowest in the number of black points but in fifth place in the number of function evaluations per point. The difference between the methods is not vast. If we take a range of 4.1-4.6 ANIP, we find the following methods (1.7), (1.11), (1.12), (1.23), (1.5) and (1.27). The top 5 in CPU time (ranging 222 – 260 seconds) are (1.5), (1.7), (1.14), (1.12) and (1.11). The top 5 is terms of fewer black points (614 – 1564 black points) are (1.5), (1.11), (1.7), (1.12) and (1.27).

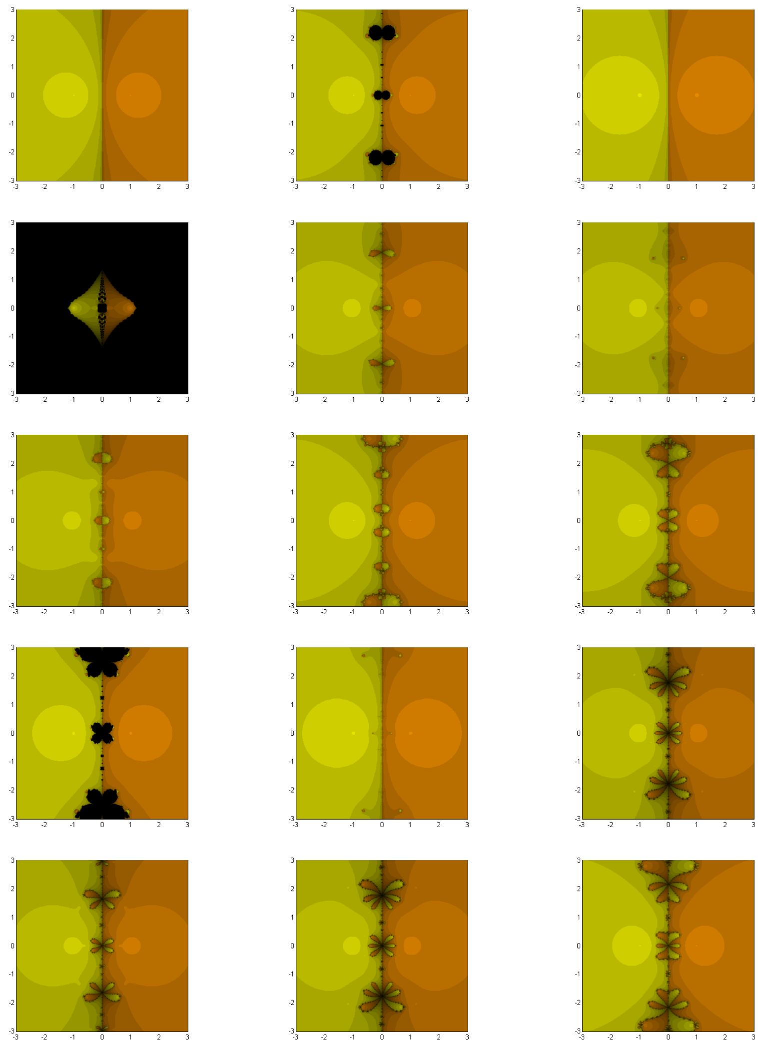

Figure 1: The top row for (1.7) (left), (1.8) (center) and (1.9) (right). The second row for (1.10) (left), (1.11) (center) and (1.12) (right). The third row for (1.13) (left), (1.14) (center) and (1.15) (right). The fourth row for (1.17) (left), (1.18) (center) and (1.19) (right).The bottom row for (1.20) (left), (1.21) (center) and (1.22) (right) for the roots of the polynomial z^2-1.

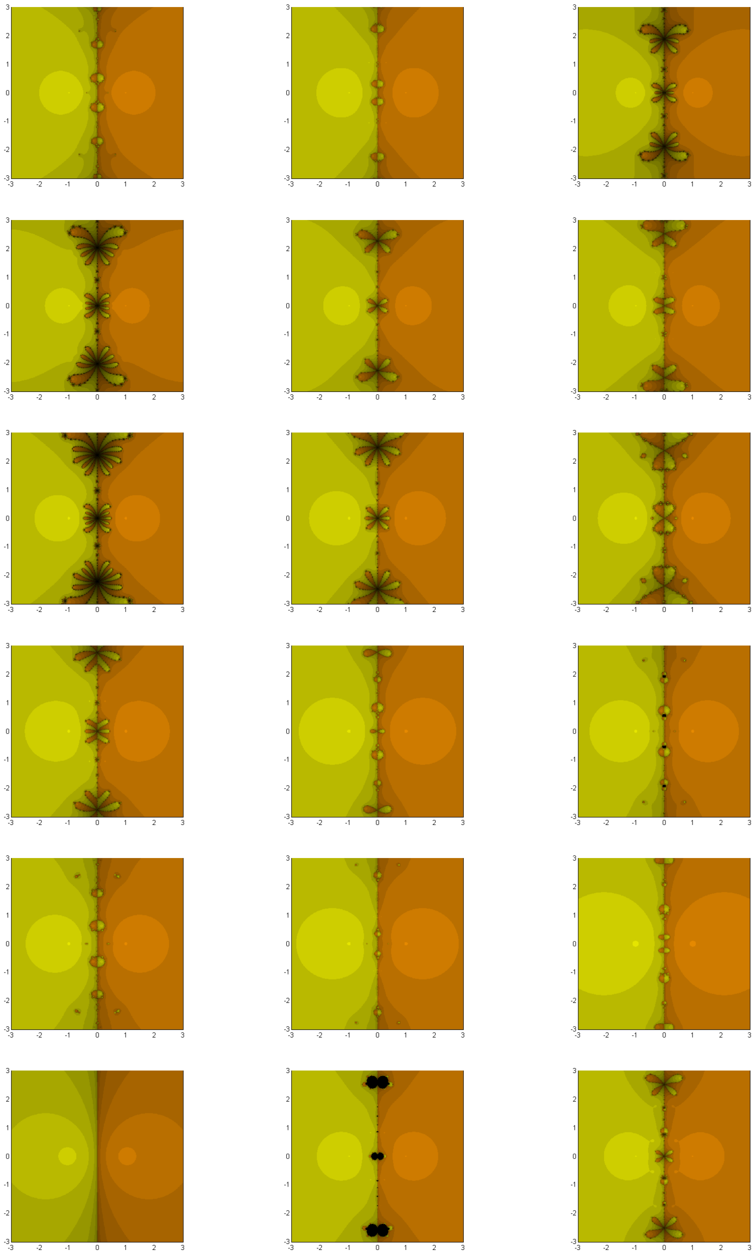

Figure 2: The top row for (1.23) (left), (1.24) (center) and (1.25) (right). The second row for (1.26) (left), (1.27) (center) and (1.28) (right). The third row for (1.29) (left), (1.30) (center) and (1.31) (right). The fourth row for (1.32) (left), (1.33) (center) and (1.34) (right). The fifth row for (1.35) (left), (1.36) (center) and (1.37) (right). The bottom row for (1.5) (left), (1.38) (center) and (1.39) (right) for the roots of the polynomial $z^2 - 1$.

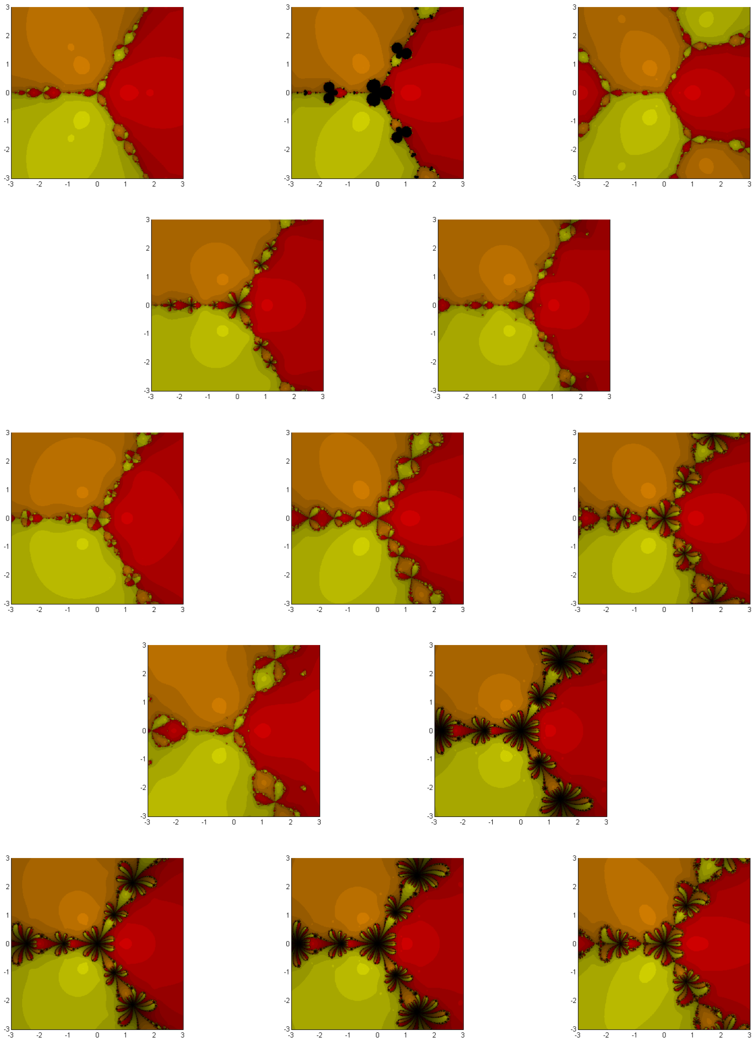

Figure 3: The top row for (1.7) (left), (1.8) (center) and (1.9) (right). The second row for (1.11) (left) and (1.12) (right). The third row for (1.13) (left), (1.14) (center) and (1.15) (right). The fourth row for (1.18) (left) and (1.19) (right).The bottom row for (1.20) (left), (1.21) (center) and (1.22) (right) for the roots of the polynomial z^3-1.

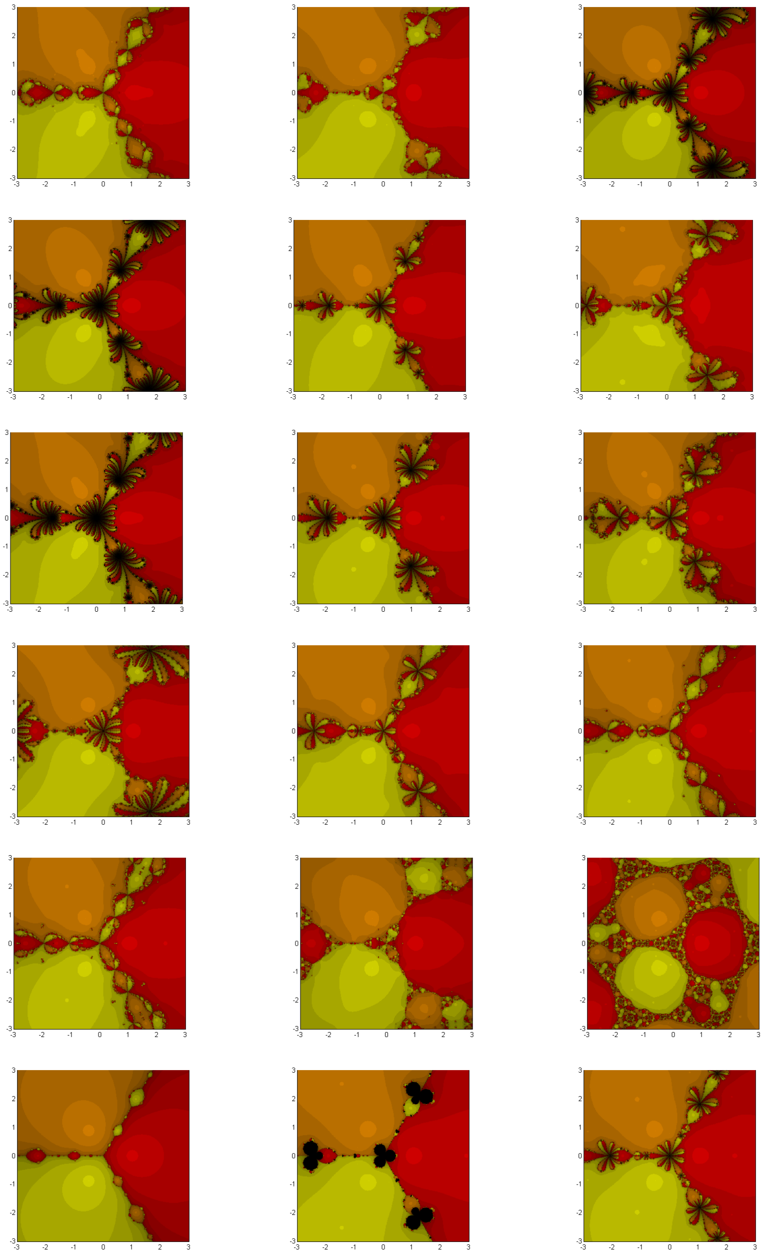

Figure 4: The top row for (1.23) (left), (1.24) (center) and (1.25) (right). The second row for (1.26) (left), (1.27) (center) and (1.28) (right). The third row for (1.29) (left), (1.30) (center) and (1.31) (right). The fourth row for (1.32) (left), (1.33) (center) and (1.34) (right). The fifth row for (1.35) (left), (1.36) (center) and (1.37) (right). The bottom row for (1.5) (left), (1.38) (center) and (1.39) (right) for the roots of the polynomial $z^3 - 1$.

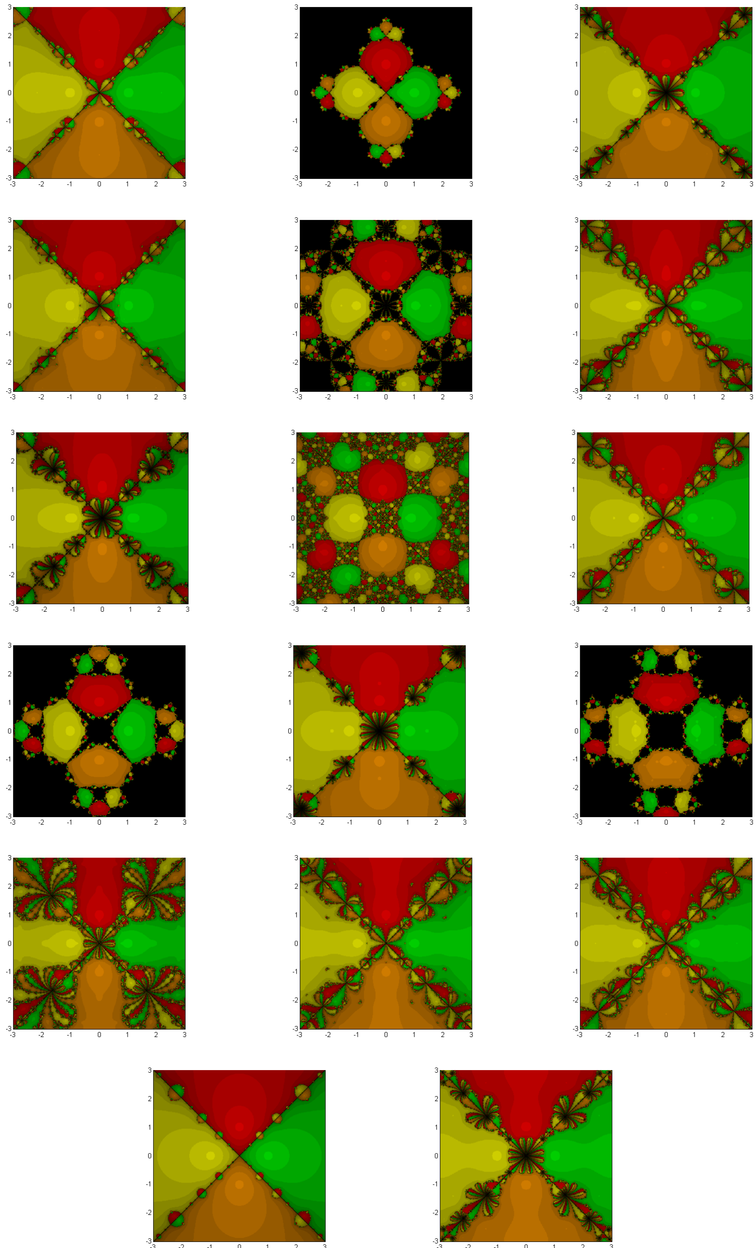

Figure 5: The top row for (1.7) (left), (1.9) (center) and (1.11) (right). The second row for (1.12) (left), (1.13) (center) and (1.14) (right). The third row for (1.15) (left), (1.18) (center) and (1.22) (right). The fourth row for (1.23) (left), (1.27) (center) and (1.28) (right).The fifth row for (1.33) (left), (1.34) (center) and (1.35) (right). The bottom row for (1.5) (left) and (1.39) (right) for the roots of the polynomial $z^4 - 1$.

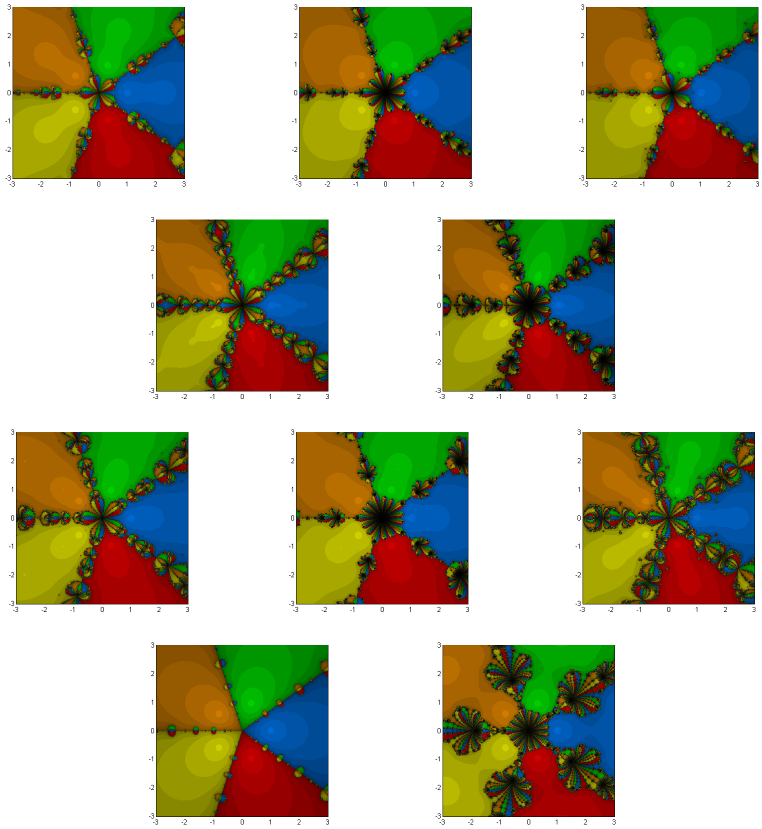

Figure 6: The top row for (1.7) (left), (1.11) (center) and (1.12) (right). The second row for (1.14) (left) and (1.15) (right). The third row for (1.23) (left), (1.27) (center) and (1.35) (right). The bottom row for (1.5) (left), (1.39) (right) for the roots of the polynomial $z^5 - 1$.

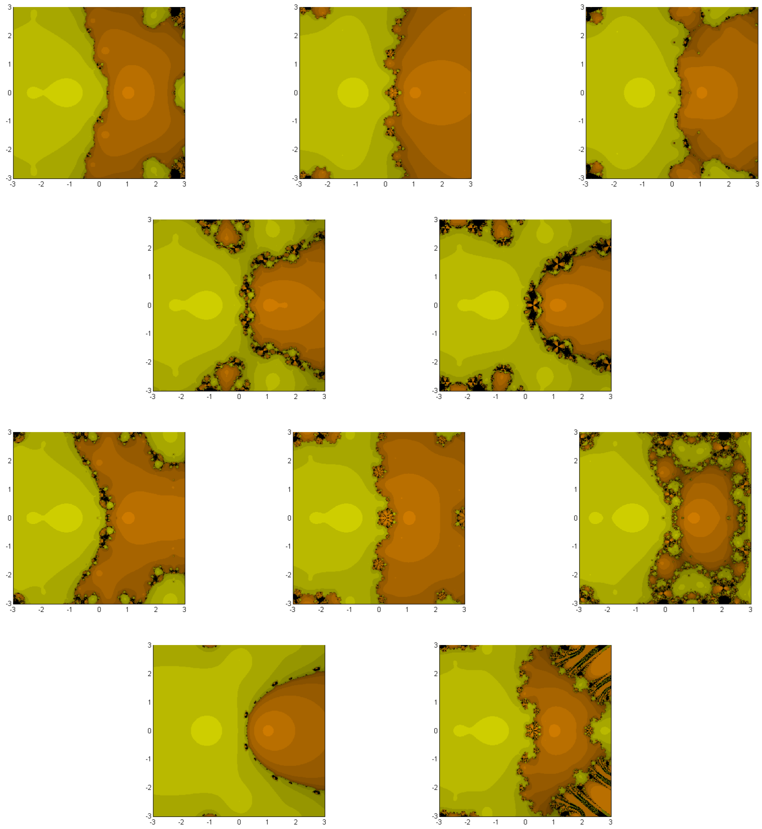

Figure 7: The top row for (1.7) (left), (1.11) (center) and (1.12) (right). The second row for (1.14) (left) and (1.15) (right). The third row for (1.23) (left), (1.27) (center) and (1.35) (right). The bottom row for (1.5) (left), (1.39) (right) for the roots of the nonlinear function $(e^{z+1}-1)(z-1)$.

Table 1: The Extrernal fixed points and the type for each of the methods

Method	Extraneous Fixed Points	Type
(1.5)	0 (double)	parabolic
(1.7)	$\pm\frac{\sqrt{3}}{3}i$	
(1.8)	0 (quadruple)	parabolic
(1.9)	0 (double), $\pm i$	
(1.10)	$\pm 1.376381920i, \pm 0.3249196966i$	first two attractive
(1.11)	$\pm.8241875313i, \pm.3658285681$	
(1.12)	$\pm 0.5773502693i, \pm 0.4472135954$	
(1.13)	$\pm 1.233858886i, \pm.3063271475$	
(1.14)	$\pm 0.3165651760 \pm 0.4208759764i$	
(1.15)	$\pm 0.3881746736 \pm 0.3030776267i$	
(1.17)	0 (sextuple)	parabolic
(1.18)	$\pm.8241875313i, \pm.3658285681$	
(1.19)	$\pm 1(double), \pm\frac{\sqrt{3}}{3}i$	the real are parabolic, the other attractive
(1.20)	$\pm 0.4248970659 \pm 0.1941143315i$	
(1.21)	$\pm.4191292906 \pm .3572718068i, \pm 0.5027770606$	
(1.22)	$\pm 0.4472135954, \pm.4347208721 \pm 0.4347208721i$	
(1.23)	$\pm 0.3984452141, \pm.2706511783 \pm .5341202257i$	
(1.24)	$\pm 1.177988819i, \pm.2344867660 \pm .3493204534i$	

Table 2: The Extrernal fixed points and the type for each of the methods

Method	Extraneous Fixed Points	Type
(1.25)	$\pm 0.3531230488, \pm 0.4660790273 \pm 0.2880076212i$	
(1.26)	$\pm .4344883443 \pm 0.09409191523i$, $\pm 0.4631859437 \pm 0.3993687189i$	
(1.27)	$\pm 0.7604373178i, \pm .3733778531 \pm 0.2289240577i$	
(1.28)	$\pm 1.199162147i, \pm 0.3879289068 \pm 0.2546119411i$	
(1.29)	$\pm 0.5219553315, \pm 0.4154820660 \pm 0.2177317237i$, $\pm 0.4817657461 \pm 0.4651309847i$	
(1.30)	$\pm 0.4703869448, \pm .3790735155 \pm 0.4072690996i$, $\pm 0.6972800367i$	
(1.31)	$\pm 0.4894203836, \pm 0.3750797542 \pm 0.4636770778i$	
(1.32)	$\pm 0.4627362244, \pm 0.4081572503 \pm 0.4045477933i$, $\pm 1.175279534i$	
(1.33)	$\pm 0.2740149501, \pm 0.2370074750 \pm 0.8208012174i$	
(1.34)	$\pm .4164515722, \pm 0.2380307187 \pm 0.7250707001i$	
(1.35)	$\pm 0.4472135954, \pm 0.3228899723$, $\pm 0.2943092590 \pm 0.6500236077i$	
(1.36)	$\pm 0.5773502693i, \pm 0.4086599363, \pm 1.132155499i$, $\pm 0.4013588625i$	
(1.37)	$\pm 0.2124092249 \pm 0.2454578977i$, $\pm 0.2033256984 \pm 1.200351443i$	
(1.38)	0 (quadruple), $\pm i$	the real are parabolic
(1.39)	$\pm 0.3156450459 \pm 0.1941898168i$, $\pm 0.1381819628 \pm 0.8486410446i$	

Table 3: Average number of iterations per point (ANIP) for each example (3.1 – 3.5) and each of the methods

Method	Ex1	Ex2	Ex3	Ex4	Ex5	average
(1.5)	3.88	4.44	5.25	5.35	3.97	4.58
(1.7)	3.20	3.78	4.48	4.60	4.41	4.10
(1.8)	4.42	5.48	-	-	-	-
(1.9)	2.84	4.20	28.27	-	-	-
(1.10)	37.70	-	-	-	-	-
(1.11)	3.82	4.24	5.24	5.27	3.93	4.50
(1.12)	3.76	4.27	4.98	5.17	4.35	4.51
(1.13)	3.74	3.86	16.48	-	-	-
(1.14)	3.57	4.28	5.55	5.60	5.19	4.84
(1.15)	3.87	4.98	6.54	7.09	5.96	5.69
(1.17)	6.26	-	-	-	-	-
(1.18)	3.01	4.04	7.47	-	-	-
(1.19)	4.42	6.71	-	-	-	-
(1.20)	4.29	6.21	-	-	-	-
(1.21)	4.42	6.71	-	-	-	-
(1.22)	3.90	5.22	25.04	-	-	-
(1.23)	3.35	4.06	5.23	5.43	4.64	4.54
(1.24)	3.18	3.80	25.04	-	-	-
(1.25)	4.17	6.45	-	-	-	-
(1.26)	4.24	6.32	9.24	-	-	-
(1.27)	3.55	4.12	5.47	5.60	4.23	4.59
(1.28)	3.47	3.97	24.86	-	-	-
(1.29)	4.34	6.35	10.79	-	-	-
(1.30)	3.62	4.71	28.09	-	-	-
(1.31)	3.58	4.72	6.90	-	-	-
(1.32)	3.41	4.78	5.50	-	-	-
(1.33)	3.09	4.13	6.90	16.28	-	-
(1.34)	3.17	4.08	5.50	-	-	-
(1.35)	3.22	4.16	5.37	5.84	6.26	4.97
(1.36)	2.97	4.35	27.80	-	-	-
(1.37)	2.74	5.43	-	-	-	-
(1.38)	3.84	5.5	-	-	-	-
(1.39)	3.36	4.16	5.56	6.97	5.80	5.17

Table 4: CPU time (in seconds) required for each example (3.1 – 3.5) and each of the methods

Method	Ex1	Ex2	Ex3	Ex4	Ex5	average
(1.5)	125.191	180.618	246.84	295.451	263.314	222.283
(1.7)	115.285	180.04	235.14	283.336	303.984	224.557
(1.8)	152.741	243.33	-	-	-	-
(1.9)	113.772	206.171	1452.104	-	-	-
(1.10)	1393.431	-	-	-	-	-
(1.11)	153.178	208.23	297.868	338.944	300.599	259.764
(1.12)	141.009	211.957	279.695	329.662	319.38	256.341
(1.13)	159.62	211.179	804.697	-	-	-
(1.14)	232.724	207.809	304.935	355.838	355.635	244.843
(1.15)	153.646	243.939	360.877	451.389	409.003	323.771
(1.17)	241.568	-	-	-	-	-
(1.18)	128.997	202.864	394.042	-	-	-
(1.19)	176.687	336.135	-	-	-	-
(1.20)	156.266	278.556	-	-	-	-
(1.21)	173.691	336.79	-	-	-	-
(1.22)	177.42	288.898	1522.835	-	-	-
(1.23)	147.031	228.073	313.874	371.329	354.575	282.976
(1.24)	161.227	235.265	1540.156	-	-	-
(1.25)	148.606	304.03	-	-	-	-
(1.26)	188.293	328.086	517.643	-	-	-
(1.27)	166.016	229.399	340.644	386.851	343.343	293.251
(1.28)	168.106	242.847	1530.142	-	-	-
(1.29)	186.718	311.580	615.970	-	-	-
(1.30)	180.322	288.274	1742.858	-	-	-
(1.31)	172.833	287.431	414.323	-	-	-
(1.32)	183.176	297.821	349.723	-	-	-
(1.33)	156.501	229.867	403.044	1065.393	-	-
(1.34)	155.83	242.035	349.723	-	-	-
(1.35)	147.015	225.11	322.532	373.732	455.663	304.810
(1.36)	162.179	270.35	1731.689	-	-	-
(1.37)	143.755	324.342	-	-	-	-
(1.38)	171.445	284.031	-	-	-	-
(1.39)	170.696	249.711	377.756	511.106	415.337	344.921

Table 5: Number of points requiring 40 iterations for each example (3.1 – 3.5) and each of the methods

Method	Ex1	Ex2	Ex3	Ex4	Ex5	average
(1.5)	601	1	1201	21	1244	614
(1.7)	601	1	1201	1	3642	1089
(1.8)	10289	15448	-	-	-	-
(1.9)	601	1	242289	-	-	-
(1.10)	334379	-	-	-	-	-
(1.11)	601	14	1285	521	1897	864
(1.12)	601	1	1201	35	4335	1235
(1.13)	601	5	115541	-	-	-
(1.14)	601	1	1209	18	9967	2359
(1.15)	601	28	1661	2590	18040	4584
(1.17)	31121	-	-	-	-	-
(1.18)	601	1	1321	-	- -	
(1.19)	601	2465	-	-	-	-
(1.20)	601	804	-	-	-	-
(1.21)	601	2465	-	-	-	-
(1.22)	601	289	209569	-	-	-
(1.23)	601	1	1201	21	6665	1698
(1.24)	601	1	209569	-	-	-
(1.25)	605	1801	-	-	-	-
(1.26)	629	2784	14237	-	-	-
(1.27)	601	17	1673	1387	4143	1564
(1.28)	601	1	107809	-	-	-
(1.29)	669	3182	21313	-	-	-
(1.30)	601	119	240761	-	-	-
(1.31)	601	19	1217	-	-	-
(1.32)	601	17	1201	-	-	-
(1.33)	601	2	1217	23400	-	-
(1.34)	953	1	1201	-	-	-
(1.35)	601	1	1201	29	13339	3034
(1.36)	601	1	237309	-	-	-
(1.37)	601	22	-	-	-	-
(1.38)	7113	19295	-	-	-	-
(1.39)	601	7	1293	1089	15796	3757

4 Conclusion

This comparative study of 32 methods of cubic order with Halley's method using 3 quantitative measures found that there is no method standing out based on all 3 criteria. If we allow range of values in each category, we can recommned (1.5), (1.7), (1.11) and (1.12). This comparative study shows that the idea of composing successful methods to get an even better one is not working. Notice that Halley's method came close to be best, but composing two of those to get (1.14) did not bear fruits. The reason that (1.7) was close to best is that it has only purely imaginary extraneous fixed points. All other methods have a combination of real and complex extraneous fixed points.

Acknowledgements

The authors thank referee and editor for their useful technical comments and valuable suggestions to improve the readability of the paper.

References

[1] 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, 18 (1694) 136-148.
https://doi.org/10.1098/rstl.1694.0029

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

[3] B. Neta, M. Scott, On a family of Halley-like methods to find simple roots of nonlinear equations, Applied Mathematics and Computation, 219 (2013) 7940-7944.
https://doi.org/10.1016/j.amc.2013.02.035

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

[5] E. Laguerre, Sur la résolution des équations numériques, Nouvelles Annales de Mathématiques, sér. 2, tome 17 (1878) 97-101.

[6] A. M. Ostrowski, Solution of equations in Euclidean and Banach space, Academic Press, New York, (1973).

[7] E. Hansen, M. Patrick, A family of root finding methods, Numer. Math, 27 (1977) 257-269.
https://doi.org/10.1007/BF01396176

[8] D. B. Popovski, Solution of equations on the basis of one-point logarithmic approximation, Proc. 14th Yug. Int. Symp. on Information Processing - Informatics 79, Bled 1. - 6. October 1979, 3/202.

[9] D. B. Popovski, Numericl solution of equations on the basis of approximation by the curve $(x-p_1)[(y(x)-p_2]^2-p_3=0$, Internat. J. Numer. Meth. Engng, 14 (1979) 1574.
https://doi.org/10.1002/nme.1620141013

[10] D. B. Popovski, A family of one-point iteration formulae for finding roots, Internat. J. Computer Math, 8 (1980) Sec. B, 85-88.
https://doi.org/10.1080/00207168008803193

[11] G. V. Milovanović, Dj. R. Djordjević, The solution of nonlinear equations using iterative processes derived from exponential approximation (Serbo-Croatian), Proc. f4t Bos.-Herc. Symp. on Informatics - Jahorina 80, Jahorina, 24-28 March 1980, Vol. 2, 465/1-465/5.

[12] B. Neta, Several new methods for solving equations, Intern. J. Computer Math, 23 (1987) 265-282.
https://doi.org/10.1080/00207168808803622

[13] D. B. Popovski, P. B. Popovski, Some new one-point iteration functions of order three for solving equations, ZAMM, 62 (1982) T344-T345.

[14] B. D. Stewart, Attractor Basins of Various Root-Finding Methods, M.S. thesis, Naval Postgraduate School, Department of Applied Mathematics, Monterey, CA, June (2001).
https://calhoun.nps.edu/handle/10945/10997

[15] S. Amat, S. Busquier, S. Plaza, Review of some iterative root-finding methods from a dynamical point of view, Scientia, 10 (2004) 3-35.
http://www.mat.utfsm.cl/scientia/archivos/vol10/part_2.pdf

[16] S. Amat, S. Busquier, S. Plaza, Dynamics of a family of third-order iterative methods that do not require using second derivatives, Appl. Math. Comput, 154 (2004) 735-746.
https://doi.org/10.1016/S0096-3003(03)00747-1

[17] S. Amat, S. Busquier, S. Plaza, Dynamics of the King and Jarratt iterations, Aeq. Math, 69 (2005) 212-223.
https://doi.org/10.1007/s00010-004-2733-y

[18] M. Scott, B. Neta, C. Chun, Basin attractors for various methods, Appl. Math. Comput, 218 (2011) 2584-2599. *https://doi.org/10.1016/j.amc.2011.07.076*

[19] C. Chun, M. Y. Lee, B. Neta, J. Džunić, On optimal fourth-order iterative methods free from second derivative and their dynamics, Appl. Math. Comput, 218 (2012) 6427-6438. *https://doi.org/10.1016/j.amc.2011.12.013*

[20] F. Chicharro, A. Cordero, J. M. Gutiérrez, J. R. Torregrosa, Complex dynamics of derivative-free methods for nonlinear equations, Appl. Math. Comput, 219 (2013) 7023-7035. *https://doi.org/10.1016/j.amc.2012.12.075*

[21] A. Cordero, J. García-Maimó, J. R. Torregrosa, M. P. Vassileva, P. Vindel, Chaos in King's iterative family, Appl. Math. Lett, 26 (2013) 842-848. *https://doi.org/10.1016/j.aml.2013.03.012*

[22] B. Neta, M. Scott, C. Chun, Basin of attractions for several methods to find simple roots of nonlinear equations, Appl. Math. Comput, 218 (2012) 10548-10556. *https://doi.org/10.1016/j.amc.2012.04.017*

[23] B. Neta, C. Chun, M. Scott, Basins of attractions for optimal eighth order methods to find simple roots of nonlinear equations, Appl. Math. Comput, 227 (2014) 567-592. *https://doi.org/10.1016/j.amc.2013.11.017*

[24] C. Chun, B. Neta, S. Kim, On Jarratt's family of optimal fourth-order iterative methods and their dynamics, Fractals 22 1450013 (2014) (16 pages). *https://doi.org/10.1142/S0218348X14500133*

[25] C. Chun, B. Neta, On the new family of optimal eighth order methods developed by Lotl et al, Numerical Algorithms, 72 (2016) 363-376. *https://doi.org/10.1007/s11075-015-0048-9*

[26] C. Chun, B. Neta, Comparison of several families of optimal eighth order methods, Appl. Math. Comput, 274 (2016) 762-773. *https://doi.org/10.1016/j.amc.2015.10.092*

[27] I. K. Argyros, A. A. Magreñan, On the convergence of an optimal fourth-order family of methods and its dynamics, Appl. Math. Comput, 252 (2015) 336-346. *https://doi.org/10.1016/j.amc.2014.11.074*

[28] Á. A. Magrenán, Different anomalies in a Jarratt family of iterative root-finding methods Appl, Math. Comput, 233 (2014) 29-38. *https://doi.org/10.1016/j.amc.2014.01.037*

[29] C. Chun, B. Neta, The basins of attraction of Murakami's fifth order family of methods, Appl. Math. Comp. accepted for publication. *https://doi.org/10.1016/j.apnum.2016.07.012*

[30] C. Chun, B. Neta, An analysis of a new family of eighth-order optimal methods, Appl. Math. Comput. 245 (2014) 86-107. *https://doi.org/10.1016/j.amc.2014.07.068*

[31] C. Chun, B. Neta, An analysis of a King-based family of optimal eighth-order methods, Amer. J. Algorithms and Computing, 2 (2015) 117. *https://doi.org/10.7726/ajac.2015.1001*

[32] C. Chun, B. Neta, J. Kozdon, M. Scott, Choosing weight functions in iterative methods for simple roots, Appl. Math. Comput, 227 (2014) 788-800. *https://doi.org/10.1016/j.amc.2013.11.084*