

An Analysis of a King-based Family of Optimal Eighth-order Methods

Changbum Chun¹ and Beny Neta^{2*}

Received 26 November 2014; Published online 21 March 2015

© The author(s)2015. Published with open access at www.uscip.us

Abstract

In this paper we analyze an optimal eighth-order family of methods based on King's fourth order method to solve a nonlinear equation. This family of methods was developed by Thukral and Petković and uses a weight function. We analyze the family using the information on the extraneous fixed points. Two measures of closeness of an extraneous points set to the imaginary axis are considered and applied to the members of the family to find its best performer. The results are compared to a modified version of Wang-Liu method.

Keywords: Iterative methods; Order of convergence; Basin of attraction; Extraneous fixed points; Weight functions

1 Introduction

The problem of solving a single nonlinear equation $f(x) = 0$ is fundamental in science and engineering. For example, to minimize a function $F(x)$ one has to find the critical points, i.e. to find points where the derivative vanishes, i.e. $F'(x) = 0$. There are many algorithms for the solution of nonlinear equations, see e.g. Traub [23], Neta [14] and the recent book by Petković et al. [19]. The methods can be classified as one step and multistep. In a one step method we have

$$x_{n+1} = \phi(x_n).$$

* Corresponding Author

¹cbchun@skku.edu, Tel:+82-31-299-4523, Fax: +82-31-290-7033, Department of Mathematics, Sungkyunkwan University, Suwon 440-746, Republic of Korea

²bnet@nps.edu, Tel: 1-831-656-2235, Fax: 1-831-656-2355, Naval Postgraduate School, Department of Applied Mathematics, Monterey, CA 93943, USA

The iteration function ϕ depends on the method used. For example, Newton's method is given by

$$x_{n+1} = \phi(x_n) = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (1)$$

Some one point methods allow the use of older points, in such a case we have a one step method with memory. For example, the secant method is given by

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n).$$

In order to increase the order of a one step method, we require higher derivatives. In many cases the function is not smooth enough or the higher derivatives are too complicated. Another way to increase the order is by using multistep. The recent book by Petković et al. [19] is dedicated to multistep methods. A trivial example of a multistep method is a combination of two Newton steps, i.e.

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= y_n - \frac{f(y_n)}{f'(y_n)}. \end{aligned}$$

Of course this is too expensive. The cost of a method is defined by the number (d) of function-evaluations per step. In the latter, the method requires four function-evaluations (including derivatives). The efficiency of a method is defined by

$$I = p^{1/d},$$

where p is the order of the method. Clearly one strives to find the most efficient methods. To this end, Kung and Traub [12] introduced the idea of optimality. A method using d evaluations is optimal if the order is 2^{d-1} . They have also developed optimal multistep methods of increasing order. Newton's method (1) is optimal of order 2. King [11] has developed an optimal fourth order family of methods depending on a parameter β

$$\begin{aligned} w_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= w_n - \frac{f(w_n)}{f'(x_n)} \left[\frac{1 + \beta r_n}{1 + (\beta - 2)r_n} \right], \end{aligned} \quad (2)$$

where

$$r_n = \frac{f(w_n)}{f(x_n)}. \quad (3)$$

There are a number of ways to compare various techniques proposed for solving nonlinear equations. Frequently, authors will pick a collection of sample equations, a collection of algorithms for comparison and a starting point for each of the sample equations. Then comparisons of the various algorithms are based on the number of iterations required for convergence, number of function evaluations, and/or amount of CPU time. ``The primary flaw in this

type of comparison is that the starting point, although it may have been chosen at random, represents only one of an infinite number of other choices" [20]. In recent years the Basin of Attraction method was introduced to visually comprehend how an algorithm behaves as a function of the various starting points. The first comparative study using basin of attraction, to the best of our knowledge, is by Vrscay and Gilbert [24]. They analyzed Schröder and König rational iteration functions. Other work was done by Stewart [21], Amat et al. [1], [2], [3], [4], Chicharro et al. [5], Chun et al. [6], [8], Cordero et al. [10], Neta et al. [16], [18] and Scott et al. [20]. There are also similar results for methods to find roots with multiplicity, see e.g. [7], [15] and [17].

In this paper we empirically analyze a family of optimal eighth order methods based on King's fourth order method (2). We assume that $f : \mathbb{C} \rightarrow \mathbb{C}$. The analysis of King's method (2) was included in [18]. It was found that the best choice (in terms of basin of attraction) for the parameter is $\beta = 3 - 2\sqrt{2}$.

2 Optimal Eighth-order Family of Methods

We analyze the following three-step family of methods based on King's fourth order method (2)

$$\begin{aligned} w_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ s_n &= w_n - \frac{f(w_n)}{f'(x_n)} \left[\frac{1 + \beta r_n}{1 + (\beta - 2)r_n} \right], \\ x_{n+1} &= s_n - \frac{f(s_n)}{f'(x_n)} \left[\phi(r_n) + \frac{f(s_n)}{f(w_n) - af(s_n)} + \frac{4f(s_n)}{f(x_n)} \right], \end{aligned} \quad (4)$$

where r_n is given by (3) and $\phi(r)$ is a real-valued weight function satisfying the conditions (to ensure eighth order convergence)

$$\phi(0) = 1, \phi'(0) = 2, \phi''(0) = 10 - 4\beta, \phi'''(0) = 12\beta^2 - 72\beta + 72. \quad (5)$$

This method was developed by Thukral and Petković [22].

There are many other optimal eighth order methods that are based on King's method with a third step based on interpolating polynomials, e.g. Neta [13] has used inverse interpolation. Wang and Liu [25] have developed a method based on Hermite interpolation to remove the derivative in the third step. They also use Ostrowski's method which is a special case of King's method when $\beta = 0$. Neta et al. [16] has improved the method by replacing the function value in the third step instead of the derivative. The modified method is given by

$$\begin{aligned} w_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ s_n &= w_n - \frac{f(w_n)}{f'(x_n)} \frac{f(x_n)}{f(x_n) - 2f(w_n)}, \\ x_{n+1} &= s_n - \frac{H_3(s_n)}{f'(s_n)}, \end{aligned} \quad (6)$$

where

$$\begin{aligned}
 H_3(s_n) &= f(x_n) + f'(x_n) \frac{(s_n - w_n)^2(s_n - x_n)}{(w_n - x_n)(x_n + 2w_n - 3s_n)} \\
 &+ f'(s_n) \frac{(s_n - w_n)(x_n - s_n)}{x_n + 2w_n - 3s_n} \\
 &- \frac{f(x_n) - f(w_n)}{x_n - w_n} \frac{(s_n - x_n)^3}{(w_n - x_n)(x_n + 2w_n - 3s_n)}.
 \end{aligned} \tag{7}$$

We are only interested in the eighth order method (4) which is based on a weight function. We will compare our results to one of the best methods (see [16]), namely the modified Wang-Liu (WLN) given by (6)-(7). In a previous work [9] a method for choosing a weight function is discussed. It was shown in [7] that one should not use a polynomial as a weight function, but a rational function.

For the method (4) we consider the weight functions

$$\begin{aligned}
 \phi(t) &= \frac{a + bt}{1 + dt + gt^2}, \\
 \phi(t) &= \frac{a + bt + ct^2}{1 + dt + gt^2}.
 \end{aligned} \tag{8}$$

These functions satisfying the conditions (5) are given by

$$\phi(t) = \frac{2\beta(\beta - 2)t + 2\beta - 1}{(1 + 4\beta)t^2 + 2(1 - 4\beta + \beta^2)t + 2\beta - 1}, \tag{9}$$

$$\phi(t) = \frac{(2(g - 2)\beta - g - 1)t^2 + 2(g + \beta^2 - 4\beta + 1)t + 2\beta - 5}{(2\beta - 5)gt^2 + 2(g + 2\beta^2 - 6\beta + 6)t + 2\beta - 5}. \tag{10}$$

We denote the family of methods (4) with the weight function (9) by LQK (Linear over Quadratic using King-based) and the one using the weight function (10) by QQK (Quadratic over Quadratic using King-based). As we have seen in previous works, one should use the extraneous fixed points to find the best choice for the parameters.

3 Extraneous Fixed Points

In solving a nonlinear equation iteratively we are looking for fixed points which are zeros of the given nonlinear function. Many multipoint iterative methods have fixed points that are not zeros of the function of interest. Thus, it is necessary to investigate the number of extraneous fixed points, their location and their properties. In order to find the extraneous fixed point, we rewrite the methods of interest in the form

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} H_f(x_n, w_n, s_n), \tag{11}$$

where the function H_f for King's based method is given by

$$\begin{aligned}
 H_f(x_n, w_n, s_n) &= 1 + r_n \frac{1 + \beta r_n}{1 + (\beta - 2)r_n} \\
 &+ \frac{f(s_n)}{f(x_n)} \left[\phi(r_n) + \frac{f(s_n)}{f(w_n) - af(s_n)} + \frac{4f(s_n)}{f(x_n)} \right],
 \end{aligned} \tag{12}$$

and for WLN is given by

$$H_f(x_n, w_n, s_n) = 1 + \frac{f(w_n)}{f(x_n) - 2f(w_n)} + \frac{H_3(s_n) f'(x_n)}{f(x_n) f'(s_n)}. \quad (13)$$

Clearly, if x_n is the root then from (11) we have $x_{n+1} = x_n$ and the iterative process converged. But we can have $x_{n+1} = x_n$ even if x_n is not the root but $H_f(x_n, w_n, s_n) = 0$. Those latter points are called extraneous fixed points. Neta et al. [16] have discussed the extraneous fixed points for WLN and they are all on the imaginary axis. It was demonstrated that it is best to have the extraneous fixed points on the imaginary axis or close to it. For example, in the case of King's method, we found that the best performance is when the parameter $\beta = 3 - 2\sqrt{2}$ since then the extraneous fixed points are closest to the imaginary axis.

We have searched the parameter spaces (β, a in the case of LQK and β, g, a in the case of QQK) and found that the extraneous fixed points are not on the imaginary axis. We have tried to get several measures of closeness to the imaginary axis and experimented with those members from the parameter spaces.

Let $E = \{z_1, z_2, \dots, z_{n_{\beta, g, a}}\}$ be the set of the extraneous fixed points corresponding to the values given to β, g and a . We define

$$d(\beta, g, a) = \max_{z_i \in E} |Re(z_i)|. \quad (14)$$

We look for the parameters β, g and a which attain the minimum of $d(\beta, g, a)$. For the family LQK, the minimum of $d(\beta, a)$ occurs at $\beta = -0.9$ and $a = 1.8$. For the QQK family, the minimum of $d(\beta, g, a)$ occurs at $\beta = -0.8, g = 4, a = 1.5$, and at $\beta = 3 - 2\sqrt{2}, g = -2.2, a = 3.7$.

Another method to choose the parameters is by considering the stability of $z \in E$ defined by

$$dq(z) = \frac{dq}{dz}(z), \quad (15)$$

where q is the iteration function of (11). We define a function, the averaged stability value of the set E by

$$A(\beta, g, a) = \frac{\sum_{z_i \in E} |dq(z_i)|}{n_{\beta, g, a}}. \quad (16)$$

The smaller A becomes, the less chaotic the basin of attraction tends to. For the family LQK, the minimum of $A(\beta, a)$ occurs at $\beta = 2.7$ and $a = -0.8$. For the value $\beta = 3 - 2\sqrt{2}$, the minimum of $A(3 - 2\sqrt{2}, a)$ occurs at $a = 3$.

In the next section we plot the basins of attraction for these five cases along with the basin for WLN to find the best performer.

4 Numerical Experiments

In this section, we give the results of using the 7 cases described in Table 1 on five different polynomial equations. We also compare the results to 4 other methods based on King's

method. The first, denoted Neta6, is Neta's sixth order family of methods [26] given by

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - \frac{f(y_n)}{f'(x_n)} \frac{f(x_n) + \beta f(y_n)}{f_n + (\beta - 2)f(y_n)}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{f'(x_n)} \frac{f(x_n) - f(y_n)}{f_n - 3f(y_n)}. \end{aligned} \tag{17}$$

The second one is a seventh order method based on Newton's interpolating polynomial and King's method, denoted NIK7, and is given by (see (3.35) in [19])

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - \frac{f(y_n)}{f'(x_n)} \frac{f(x_n) + \beta f(y_n)}{f(x_n) + (\beta - 2)f(y_n)}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{f[z_n, y_n] + f[z_n, y_n, x_n](z_n - y_n)} \end{aligned} \tag{18}$$

The third is an eighth order method, denoted Neta8, and is given by [13]

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ t_n &= y_n - \frac{f(y_n)}{f'(x_n)} \frac{f(x_n) + \beta f(y_n)}{f(x_n) + (\beta - 2)f(y_n)}, \\ x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} + \gamma f(x_n)^2 - \rho f(x_n)^3, \end{aligned} \tag{19}$$

where

$$\begin{aligned} \rho &= \frac{\phi_y - \phi_t}{F_y - F_t}, \quad \gamma = \phi_y - \rho F_y, \quad F_y = f(y_n) - f(x_n), \quad F_t = f(t_n) - f(x_n), \\ \phi_y &= \frac{y_n - x_n}{F_y^2} - \frac{1}{F_y f'(x_n)}, \quad \phi_t = \frac{t_n - x_n}{F_t^2} - \frac{1}{F_t f'(x_n)}. \end{aligned} \tag{20}$$

The last one is a sixteenth order method, denoted Neta16, and is given by [13]

$$\begin{aligned}
 y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
 z_n &= y_n - \frac{f(y_n)}{f'(x_n)} \frac{f(x_n) + \beta f(y_n)}{f(x_n) + (\beta - 2)f(y_n)}, \\
 t_n &= x_n - \frac{f(x_n)}{f'(x_n)} + c_n f(x_n)^2 - d_n f(x_n)^3 \\
 x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} + \rho_n f(x_n)^2 - \gamma_n f(x_n)^3 + q_n f(x_n)^4
 \end{aligned} \tag{21}$$

where c_n and d_n are given

$$\begin{aligned}
 d_n &= \frac{1}{[f(y_n) - f(x_n)][f(y_n) - f(z_n)]} \left[\frac{y_n - x_n}{f(y_n) - f(x_n)} - \frac{1}{f'(x_n)} \right] \\
 &\quad - \frac{1}{[f(y_n) - f(z_n)][f(z_n) - f(x_n)]} \left[\frac{z_n - x_n}{f(z_n) - f(x_n)} - \frac{1}{f'(x_n)} \right] \\
 c_n &= \frac{1}{f(y_n) - f(x_n)} \left[\frac{y_n - x_n}{f(y_n) - f(x_n)} - \frac{1}{f'(x_n)} \right] - d_n [f(y_n) - f(x_n)]
 \end{aligned} \tag{22}$$

and

$$\begin{aligned}
 q_n &= \frac{\frac{\phi(t_n) - \phi(z_n)}{F(t_n) - F(z_n)} - \frac{\phi(y_n) - \phi(z_n)}{F(y_n) - F(z_n)}}{F(t_n) - F(y_n)}, \\
 \gamma_n &= \frac{\phi(t_n) - \phi(z_n)}{F(t_n) - F(z_n)} - q_n (F(t_n) + F(z_n)), \\
 \rho_n &= \phi(t_n) - \gamma_n F(t_n) - q_n F^2(t_n),
 \end{aligned} \tag{23}$$

and for $\delta_n = y_n, z_n, t_n$

$$\begin{aligned}
 F(\delta_n) &= f(\delta_n) - f(x_n), \\
 \phi(\delta_n) &= \frac{(\delta_n - x_n)}{F^2(\delta_n)} - \frac{1}{f'(x_n)F(\delta_n)}.
 \end{aligned} \tag{24}$$

In all these methods we have chosen the parameter $\beta = 3 - 2\sqrt{2}$ as suggested by the analysis in [18].

Table 1: The eleven cases for experimentation

case	method	β	g	a
1	LQK	-0.9	-	1.8
2	LQK	$3 - 2\sqrt{2}$	-	3
3	LQK	2.7	-	-0.8
4	QQK	-0.8	4	1.5
5	QQK	$3 - 2\sqrt{2}$	-2.2	3.7
6	QQK	-1.7	-3.3	-1.7
7	WLN	-	-	-
8	Neta6	-	-	-
9	NIK7	-	-	-
10	Neta8	-	-	-
11	Neta16	-	-	-

We have ran our code for each case and each example on a 6 by 6 square centered at the origin. We have taken 360,000 equally spaced points in the square as initial points for the algorithms. We have recorded the root the method converged to and the numebr of iterations it took. We chose a color for each root and the intensity of the color gives information on the number of iterations. The slower the convergence the darker the shade. If the scheme did not converge in 40 iterations to one of the roots, we color the point black.

Example 1

In our first example, we have taken the quadratic polynomial

$$p_1(z) = z^2 - 1, \quad (25)$$

whose roots, $z = \pm 1$, are both real. The results are given in Figures 1-6. It is clear that WLN (Fig. 4 left) and NIK7 (Fig. 5 left) are the best since there are no black points. Case 3 (Fig. 2 left) has large black region. Case 6 (Fig. 3 right) has points converging to the distant roots and black region. Case 4 (Fig. 2 right), Neta6 (Fig. 4 right), Neta8 (Fig. 5 right), case 1 (Fig. 1 left), and case 2 (Fig. 1 right) are slightly better but are not as good as cases 7 (Fig. 4 left) and 9 (Fig. 5 left). In order to get a more quantitative comparison, we have computed the average number of iterations per point. The smaller the number the faster the method converged on average. These numerical results are given in Table 2. It is clear from the table that WLN is best followed by NIK7 and the worst are cases 6 and 11. Therefore, in the other examples we will not show these cases. We have also computed the CPU time in seconds required to run each program on each example. These results are given in Table 3 and they show that NIK7 and Neta6 are slightly faster than WLN. Again cases 6 and 11 took the most CPU time.

Table 2: Average number of iterations per point for each example (1--6) and each case

case	Ex1	Ex2	Ex3	Ex4	Ex5	Average
1	2.6166	7.4865	6.3671	31.3655	37.3817	17.04348
2	2.6548	3.9913	3.2882	6.9025	8.2288	5.01312
3	4.6811	8.9580	6.5885	15.2082	17.6276	10.61268
4	2.5688	5.7337	6.2696	23.6324	37.8956	15.22002
5	3.9804	4.9690	3.6428	8.1225	10.0625	6.15544
6	5.4280	31.3446	6.1734	22.0437	19.4689	16.89172
WLN	2.2676	2.7084	2.5306	3.7191	4.7871	3.20256
Neta6	2.5744	3.4305	3.1251	13.6225	33.7292	11.2963
NIK7	2.3195	6.3093	2.8332	6.9646	9.3339	5.5521
Neta8	2.5711	7.0367	2.8358	7.9514	9.1620	5.9114
Neta16	5.6447	7.5538	4.7945	11.4577	12.2628	8.3427

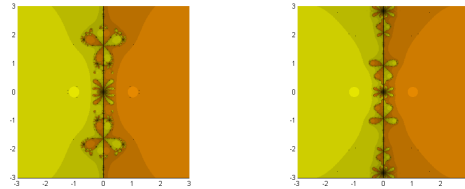


Fig. 1: LQK with $\beta = -0.9$ and $a = 1.8$ (left) and LQK with $\beta = 3 - 2\sqrt{2}$ and $a = 3$ (right) for the roots of the polynomial $z^2 - 1$

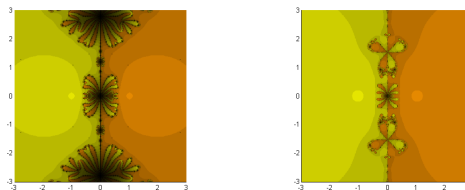


Fig. 2: LQK with $\beta = 2.7$ and $a = -0.8$ (left) and QQK with $\beta = -0.8$, $g = 4$ and $a = 1.5$ (right) for the roots of the polynomial $z^2 - 1$

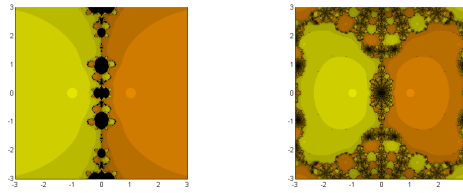


Fig. 3: QK with $\beta = 3 - 2\sqrt{2}$, $g = -2.2$ and $a = 3.7$ (left) and QK with $\beta = -1.7$, $g = -3.3$ and $a = -1.7$ (right) for the roots of the polynomial $z^2 - 1$

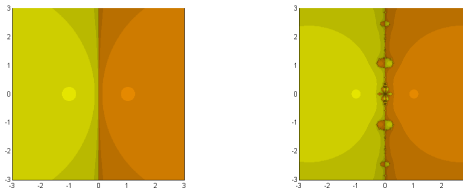


Fig. 4: WLN (left) and Neta6 (right) for the roots of the polynomial $z^2 - 1$

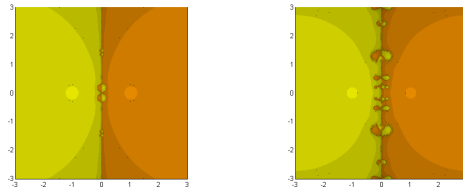


Fig. 5: NIK7 (left) and Neta8 (right) for the roots of the polynomial $z^2 - 1$

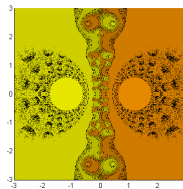


Fig. 6: Neta16 for the roots of the polynomial $z^2 - 1$

Example 2

In the second example we have taken a cubic polynomial with the 3 roots of unity, i.e.

$$p_2(z) = z^3 - 1. \tag{26}$$

The results are presented in Figures 7-11. Again, WLN outperformed the others. Neta6 (Fig. 10 left) and case 2 (Fig. 7 right) are second best since the black regions are only in the boundary between the roots. Case 3 is the worst among those in the Figures. These conclusions are

confirmed quantitatively by the results in Table 2. The CPU time shows a slight advantage of NIK7 over case 2, (see Table 3).

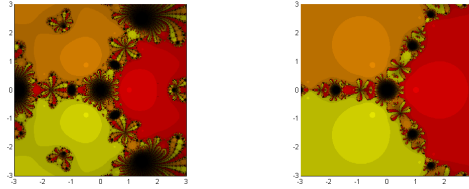


Fig. 7: LQK with $\beta = -0.9$ and $a = 1.8$ (left) and LQK with $\beta = 3 - 2\sqrt{2}$ and $a = 3$ (right) for the roots of the polynomial $z^3 - 1$

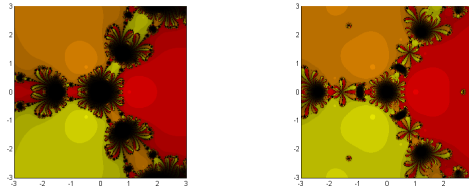


Fig. 8: LQK with $\beta = 2.7$ and $a = -0.8$ (left) and QQK with $\beta = -0.8$, $g = 4$ and $a = 1.5$ (right) for the roots of the polynomial $z^3 - 1$

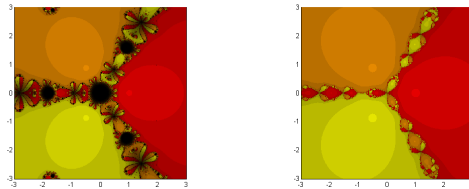


Fig. 9: QQK with $\beta = 3 - 2\sqrt{2}$, $g = -2.2$ and $a = 3.7$ (left) and WLN (right) for the roots of the polynomial $z^3 - 1$

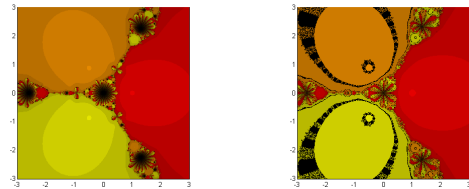


Fig. 10: Neta6 (left) and NIK7 (right) for the roots of the polynomial $z^3 - 1$

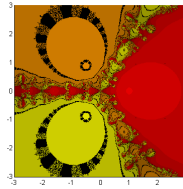


Fig. 11: Neta8 for the roots of the polynomial $z^3 - 1$

Example 3

In the third example we have taken a polynomial of degree 4 with 4 real roots at $\pm 1, \pm 3$, i.e.

$$p_3(z) = z^4 - 10z^2 + 9. \quad (27)$$

The results are presented in Figures 12-16. Notice the chaos in Figures 12 left and 13 right. This is also seen in Table 2. Again, WLN (Fig. 14 right) is the best performer followed by NIK7 (Fig. 15 right) and Neta8 (Fig. 16). The CPU time shows that Neta6 is slightly faster than Neta8. Since cases 1 and 3 were the worst in this example, we will not show the basins for these cases in the last two examples. Case 4 is slightly better than cases 1 and 3, but it has more black points. That is the reason why the CPU time for case 4 is slightly higher than case 3.

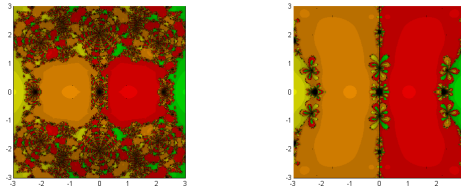


Fig. 12: LQK with $\beta = -0.9$ and $a = 1.8$ (left) and LQK with $\beta = 3 - 2\sqrt{2}$ and $a = 3$ (right) for the roots of the polynomial $z^4 - 10z^2 + 9$

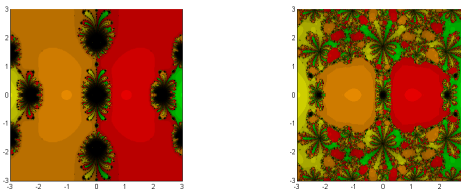


Fig. 13: LQK with $\beta = 2.7$ and $a = -0.8$ (left) and QQK with $\beta = -0.8, g = 4$ and $a = 1.5$ (right) for the roots of the polynomial $z^4 - 10z^2 + 9$

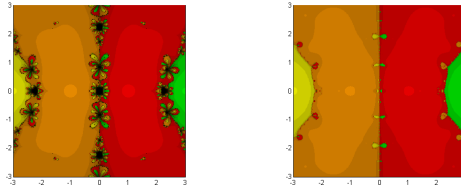


Fig. 14: QK with $\beta = 3 - 2\sqrt{2}$, $g = -2.2$ and $a = 3.7$ (left) and WLN (right) for the roots of the polynomial $z^4 - 10z^2 + 9$

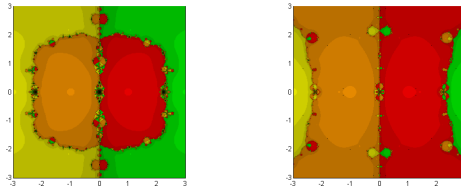


Fig. 15: Neta6 (left) and NIK7 (right) for the roots of the polynomial $z^4 - 10z^2 + 9$

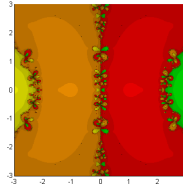


Fig. 16: Neta8 for the roots of the polynomial $z^4 - 10z^2 + 9$

Example 4

In the next example we have taken a polynomial of degree 5 with the 5 roots of unity, i.e.

$$p_4(z) = z^5 - 1. \tag{28}$$

The results are presented in Figures 17-20. Again, WLN (Fig. 18 right) is best, followed by case 2 (Fig. 17 left) and NIK7 (Fig. 19 right). Case 4 (Fig. 17 right) shows the largest black regions and it tooks the most CPU time to run.

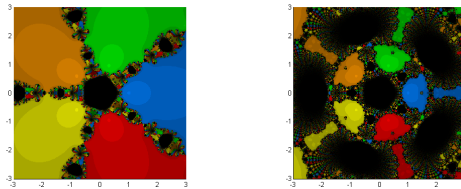


Fig. 17: LQK with $\beta = 3 - 2\sqrt{2}$ and $a = 3$ (left) and QK with $\beta = -0.8$, $g = 4$ and $a = 1.5$ (right) for the roots of the polynomial $z^5 - 1$

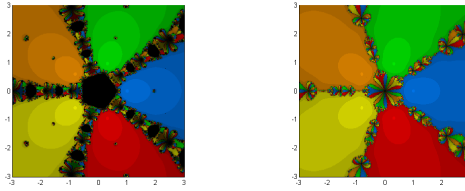


Fig. 18: QK with $\beta = 3 - 2\sqrt{2}$, $g = -2.2$ and $a = 3.7$ (left) and WLN (right) for the roots of the polynomial $z^5 - 1$

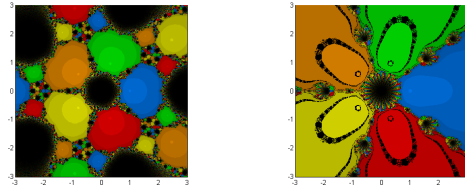


Fig. 19: Neta6 (left) and NIK7 (right) for the roots of the polynomial $z^5 - 1$

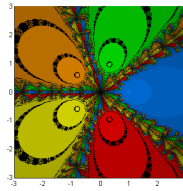


Fig. 20: Neta8 for the roots of the polynomial $z^5 - 1$

Table 3: CPU time in sec. for each example (1--6) and each case using Dell Optiplex 990

case	Ex1	Ex2	Ex3	Ex4	Ex5	Average
1	258.89	904.13	980.49	-	-	-
2	332.70	610.46	592.22	1179.27	1586.51	860.23
3	458.70	1038.40	1014.08	-	-	-
4	294.74	784.95	1065.42	3554.62	-	-
5	558.42	845.22	741.59	1521.88	-	-
6	612.42	-	-	-	-	-
WLN	201.38	318.07	375.85	517.59	778.78	438.33
Neta6	191.14	332.60	412.06	1536.93	-	-
NIK7	186.99	500.27	388.17	671.11	1009.32	551.17
Neta8	277.52	791.48	517.17	1074.86	1483.10	828.82
Neta16	771.84	-	-	-	-	-

Example 5

In the last example we took a polynomial of degree 7 having the 7 roots of unity, i.e.

$$p_6(z) = z^7 - 1. \tag{29}$$

The results for cases 2, 7, 9, and 10 are presented in Figures 21-22. As in all previous examples, WLN (Fig. 21 right) performed better than the others and now case 2 (Fig. 21 left) is slightly better than cases 9 (Fig. 22 left) and 10 (Fig. 22 right).

Based on all these examples, we computed the average of the 5 experiments (see Table 2) and found that WLN is best overall, followed by cases 2, 9, and 10. Case 2 is the only new one that performed well on average. This case is based a rational weight function (LQK) with $\beta = 3 - 2\sqrt{2}$ having a minimum for the measure A . The CPU time results (Table 3) show that case 9 is slightly faster than case 2, but it is of a lower order of convergence than case 2.

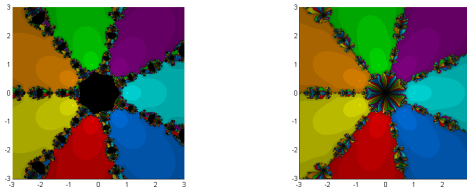


Fig. 21: LQK with $\beta = 3 - 2\sqrt{2}$ and $a = 3$ (left) and WLN (right) for the roots of the polynomial $z^7 - 1$

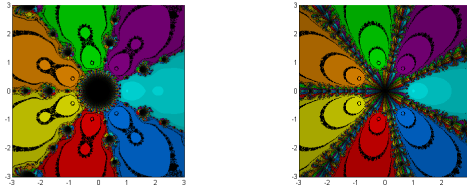


Fig. 22: NIK7 (left) and Neta8 (right) for the roots of the polynomial $z^7 - 1$

5 Conclusion

We have empirically analyzed a family of King-based eighth order methods developed by Thukral and Petković [22] and shown how to choose the weight function and the parameters involved in the family of methods. Several possibilities were suggested for the parameters. One of those came close to WLN, NIK7 and Neta8. This one having the parameter $\beta = 3 - 2\sqrt{2}$ and a weight function being a quotient of linear over quadratic polynomials. This parameter β was found to yield the best performance of King's fourth order method and used in Neta6, NIK7, Neta8 and Neta16.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2005012). The first author thanks the Naval Postgraduate School for the invitation to spend a month in the Applied Mathematics Department.

References

- [1] 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.
- [2] S. Amat, S. Busquier, S. Plaza, Iterative root-finding methods, unpublished report, 2004.
- [3] S. Amat, S. Busquier, S. Plaza, Review of some iterative root-finding methods from a dynamical point of view, *Sci. Ser. A Math. Sci.*, **10**, (2004), 3--35.
- [4] S. Amat, S. Busquier, S. Plaza, Dynamics of the King and Jarratt iterations, *Aeq. Math*, **69**, (2005), 212--223.
- [5] F. Chircharro, A. Cordero, J. M. Gutiérrez, J. R. Torregrosa, Complex dynamics of derivative-free methods for nonlinear equations, *Appl. Math. Comput.*, **219**, (2013), 7023--7035.
- [6] C. Chun, M. Y. Lee, B. Neta, Džunić, J., On optimal fourth-order iterative methods free from second derivative and their dynamics, *Appl. Math. Comput.*, **218**, (2012), 6427--6438.
- [7] B. Neta, C. Chun, Basins of attraction for Zhou-Chen-Song fourth order family of methods for multiple roots, *Math. Comput. Simulat.*
- [8] C. Chun, B. Neta, Sujin Kim, On Jarratt's family of optimal fourth-order iterative methods and their dynamics, *Fractals* doi:10.1142/S0218348X14500133
- [9] C. Chun, B. Neta, J. Kozdon, M. Scott, Choosing weight functions in iterative methods for simple roots, *Appl. Math. Comput.*, **227**, (2014), 788--800.
- [10] 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.
- [11] R. F. King, A family of fourth-order methods for nonlinear equations, *SIAM Numer. Anal.*, **10**, (1973), 876-879.
- [12] H. T. Kung, J. F. Traub, Optimal order of one-point and multipoint iterations, *J. Assoc. Comput. Mach.*, **21**, (1974), 643-651.
- [13] B. Neta, On a family of multipoint methods for nonlinear equations, *Intern. J. Computer Math.*, **9**, (1981), 353-361.
- [14] B. Neta, Numerical Methods for the Solution of Equations. net-a-sof, Monterey, 1983.
- [15] B. Neta, C. Chun, Basins of attraction for several optimal fourth order methods for multiple roots, *Math. Comput. Simulat.*, **103**, (2014) 39--59

- [16] B. Neta, C. Chun, M. Scott, Basins of attraction for optimal eighth order methods to find simple roots of nonlinear equations, *Appl. Math. Comput.*, **227**, (2014), 567--592.
- [17] B. Neta, M. Scott, C. Chun, Basin attractors for various methods for multiple roots, *Appl. Math. Comput.*, **218**, (2012), 5043--5066.
- [18] 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.
- [19] M. S. Petković, B. Neta, L. D. Petković, J. Džunić, *Multipoint Methods for Solving Nonlinear Equations*, Elsevier, 2013.
- [20] M. Scott, B. Neta, C. Chun, Basin attractors for various methods, *Appl. Math. Comput.*, **218**, (2011), 2584--2599.
- [21] B. D. Stewart, *Attractor Basins of Various Root-Finding Methods*, M.S. thesis, Naval Post-graduate School, Department of Applied Mathematics, Monterey, CA, June 2001.
- [22] R. Thukral, M. S. Petković, Family of three-point methods of optimal order for solving nonlinear equations, *J. Comput. Appl. Math.*, **233**, (2010), 2278--2284.
- [23] J.F. Traub, *Iterative Methods for the Solution of Equations*, Chelsea Publishing Company, New York, 1977.
- [24] E. R. Vrscay, W. J. Gilbert, Extraneous fixed points, basin boundaries and chaotic dynamics for Schröder and önic rational iteration functions, *Numer. Math.*, **52**, (1988), 1--16.
- [25] X. Wang, L. Liu, Modified Ostrowski's method with eighth-order convergence and high efficiency index, *Appl. Math. Lett.*, **23**, (2010), 549--554.
- [26] B. Neta, A sixth order family of methods for nonlinear equations, *Int. J. Comput. Math.*, **7**, (1979), 157--161.