Applicable Analysis and Discrete Mathematics
available online at http://pefmath.etf.rs

Appl. Anal. Discrete Math. **13** (2019), 399–422.
https://doi.org/10.2298/AADM190111015P

ON GENERALIZED HALLEY-LIKE METHODS FOR SOLVING NONLINEAR EQUATIONS

Miodrag S. Petković *, *Ljiljana D. Petković*, *Beny Neta*

Generalized Halley-like one-parameter families of order three and four for finding multiple root of a nonlinear equation are constructed and studied. This presentation is, actually, a mixture of theoretical results, algorithmic aspects, numerical experiments, and computer graphics. Starting from the proposed class of third order methods and using an accelerating procedure, we construct a new fourth order family of Halley's type. To analyze convergence behavior of two presented families, we have used two methodologies: (i) testing by numerical examples and (ii) dynamic study using basins of attraction.

1. INTRODUCTION

The approximation of zeros of a given scalar function f belongs to the most important problems that occur not only in applied mathematics but also in solving many real life problems in many disciplines of engineering, computer science, physics, biology, chemistry, communication, astronomy, geology, banking, business, digital signal processing, control theory, education, insurance, health care, social science, as well as many other fields of human activities, see [**1**, Sec. 5.1]. In fact, one of the first nonlinear problems the scientists face in their research is related to nonlinear equations. Today, for solving these research problems it is necessary to develop fast root-solvers that produce approximations to the roots of high accuracy. Solution of the mentioned task often requires a suitable combination of numerical analysis and computing science, first of all symbolic computation and computer graphics. Since there is a vast number of papers and books devoted to iterative

*Corresponding author. Miodrag S. Petković
2010 Mathematics Subject Classification. 65H05, 65D18, 68W30, 33F05
Keywords and Phrases. Nonlinear equations, Parametric iterative methods, Convergence analysis, Dynamic study, Basin of attraction.

methods for finding simple and multiple roots of nonlinear equations, see. e.g., [**2**]–[**12**], we will not discuss in details characteristics of existing methods.

The aim of this paper is to present two generalized Halley-like one-parameter families of iterative methods of order three and four for solving nonlinear equations. The developed iterative formulas are simple and enable generating different root-finding methods for the approximation of simple as well as multiple roots of a scalar equation $f(x) = 0$. A simple square-root free iterative formula (in contrast to Laguerre's family) is convenient for the user who can generate various third order methods by changing the involved parameter. The third order family includes as special cases Halley-like and Chebyshev-like methods.

For the practical point of view, one-point third and fourth order methods are the most important. This can be observed from the fact that multi-point methods of very high order are rarely needed in practice. Double-precision arithmetic is most frequently good enough for solving most real life problems giving the accuracy of desired solutions or results of calculation with approximately 16 significant decimal digits, that is, an error of about 10^{-16}.

There are several families of third order root solvers, but two of them should be singled out as pretty general: Laguerre's family [**13**] derived in 1880 for simple zeros, and very recent Traub-Gander's family for simple or multiple zeros [**14**]. Laguerre's family was modified for multiple zeros by Bodewig [**15**] and presented in modern form by the iterative formula

$$L_m(f,\lambda;x) := x - \frac{\lambda f(x)/f'(x)}{1+\operatorname{sgn}(\lambda-m)\sqrt{\Big(\dfrac{\lambda-m}{m}\Big)\Big[\lambda-1-\lambda\dfrac{f(x)f''(x)}{f'(x)^2}\Big]}} \quad (\lambda \neq 0, m),$$

where m is the order of multiplicity of the sought zero, known in advance. This formula was later rediscovered by Hansen and Patrick [**16**].

The second family, which for $\lambda > m$ contains Laguerre's family, reads

$$G_m(f;x) = x - m\frac{f(x)}{f'(x)} \cdot h(T_f(x)),$$

where h is at least two-times differentiable function which satisfies the conditions

$$h(0) = 1, \quad h'(0) = 1/2, \quad |h''(0)| < \infty.$$

The main goal of this paper is to present a new one-parameter general family of iterative methods for finding multiple zeros of a given function, similarly as the previous ones. The approximation of simple zeros appears naturally as a special case. Apart from the construction, we perform extensive analysis of the proposed family in order to obtain as good as possible estimation of the quality of particular methods from the derived family relative to the introduced parameter. We have used two methodologies:

(1)
1) Comparison by numerical examples;
2) Dynamic study using basins of attraction.

We have implored some tools of a computer algebra system such as symbolic computation and computer graphics to realize the study 2).

Why dynamic study? Note that the quality of iterative methods for solving nonlinear equations has been estimated in many papers only using numerical experiments and the study of computational efficiency of the considered root-solvers, always assuming that the chosen initial approximation x_0 is sufficiently close to the sought zero of a given function f. However, it is well known that convergence behavior differs for various methods and strictly depends on the choice of initial approximations and the structure of functions whose zeros are wanted. For this reason we employ the methodology 2) to provide a better insight into the quality of the proposed methods and their visualization. One should emphasize that this approach, often refereed to as *polynomiography* after the excellent Kalantari's book [**17**] and described in the papers cited in Section 5, has brought progress to a better understanding of iterative processes.

This paper is organized as follows. In Section 2 we construct a one-parameter family of iterative methods for finding simple or multiple roots of nonlinear equations and show that its order of convergence is three. Results of numerical experiments for several values of the parameter involved in the proposed family are displayed through three iteration steps in Section 3 using 8 test functions. The dynamic study of iterative methods for multiple zeros is given in Section 4 by constructing basins of attraction for 4 test functions and 8 values of the parameter p. Finally, starting from the proposed third order family, in Section 5 we construct one-parameter family of iterative methods of fourth order for finding multiple zeros. The analysis of convergence properties of this family using numerical examples and dynamic study is given in Section 6. For practical reasons, all figures are grouped at the end of paper.

2. A NEW ONE-PARAMETER FAMILY OF THIRD ORDER

We begin this section with Traub's result given in [**11**, Theorem 2.5].

Theorem 1. *Let $\psi(x)$ be an iteration function which defines iterative method for finding a zero α of multiplicity m of a given function f. Then for that value of m there exists a function $\omega(x)$ such that*

$$\psi(x) = x - u(x)\omega(x), \quad u(x) = \frac{f(x)}{f'(x)}, \quad \omega(\alpha) \neq 0. \tag{2}$$

In this paper we will restrict our attention to iterative methods with cubic convergence. We will often use the abbreviations

$$u(x) = \frac{f(x)}{f'(x)}, \quad v(x) = m\,\frac{f(x)}{f'(x)}, \quad A_\lambda(x) = \frac{f^{(\lambda)}(x)}{\lambda! f'(x)}, \quad A_\lambda = A_\lambda(\alpha) = \frac{f^{(\lambda)}(\alpha)}{\lambda! f'(\alpha)},$$

$$B_\lambda = \frac{f^{(\lambda)}(\alpha)}{\lambda!}, \quad C_{\lambda,m} = \frac{m!}{(m+\lambda)!}\,\frac{f^{(m+\lambda)}(\alpha)}{f^{(m)}(\alpha)} = \frac{B_{m+\lambda}}{B_m} \quad (\lambda = 1, 2, \ldots).$$

The abbreviation AEC(IM) will denote *asymptotic error constant* of the iterative method (IM). First we present two well known cubically convergent methods free of square root:

$$C(x) = x - u(x)\Big(1 + A_2(x)u(x)\Big) \quad \text{(Chebyshev's method)}, \tag{3}$$

$$H(x) = x - \frac{u(x)}{1 - A_2(x)u(x)} \quad \text{(Halley's method)}. \tag{4}$$

Regarding (2) we note that $\omega(u) = 1 + A_2(x)u$ for Chebyshev's method (3) and $\omega(u) = 1/(1 - A_2(x)u)$ for Halley's method (4). Therefore, $\omega(u)$ is a polynomial in (3), while $\omega(u)$ is a rational function in (4) regarding u as an argument. In this paper we will consider a rational approximation to construct a new cubically convergent method for finding simple or multiple zeros in the form

$$G(v(x)) = x - v(x) \cdot \frac{a + p \cdot v(x)}{1 + b \cdot v(x)} \tag{5}$$

with v as the function argument. We allow that the coefficients a and b in (5) take constant values as well as some functions of the argument x, while p is a real or complex parameter.

Let α be the zero of f with the known order of multiplicity $m \geq 1$ and let $\hat{x}$ be a new approximation to α computed by the iterative formula

$$\hat{x} = G\big(v(x)\big), \tag{6}$$

where $G\big(v(x)\big)$ is given by (5). Introduce the errors of approximations $\varepsilon = x - \alpha$, $\hat{\varepsilon} = \hat{x} - \alpha$. For two real or complex numbers z and w we will write $z = O_M(w)$ if z and w have magnitudes of the same order.

The following developments in Taylor series are valid:

$$\begin{cases} f(x) = B_m\varepsilon^m\Big(1 + C_{1,m}\varepsilon + C_{2,m}\varepsilon^2 + C_{3,m}\varepsilon^3 + O_M(\varepsilon^4)\Big), \\ f'(x) = B_m\varepsilon^{m-1}\Big(m + (m+1)C_{1,m}\varepsilon + (m+2)C_{2,m}\varepsilon^2 \\ \qquad\qquad + (m+3)C_{3,m}\varepsilon^3 + O_M(\varepsilon^4)\Big), \\ f''(x) = B_m\varepsilon^{m-2}\Big(m(m-1) + m(m+1)C_{1,m}\varepsilon + (m+1)(m+2)C_{2,m}\varepsilon^2 \\ \qquad\qquad + (m+2)(m+3)C_{3,m}\varepsilon^3 + O_M(\varepsilon^4)\Big). \end{cases} \tag{7}$$

Using (7) we find

$$v(x) = m\frac{f(x)}{f'(x)} = \varepsilon - \frac{C_{1,m}\varepsilon^2}{m} + \frac{((m+1)C_{1,m}^2 - 2mC_{2,m})\varepsilon^3}{m^2} + O_M(\varepsilon^4). \tag{8}$$

Using symbolic computation in computer algebra system *Mathematica*, we find the error of the improved approximation

$$
\begin{aligned}
\hat{\varepsilon} &= \hat{x}-\alpha = G\big(v(x)\big)-\alpha = \frac{(1-a)\varepsilon + a\Big(\big(b+\frac{C_{1,m}}{m}\big)-p\Big)\varepsilon^2}{m^2} \\
&\quad +\frac{1}{m^2}\Big[-a\Big(2bmC_{1,m}+C_{1,m}^2(m+1)+m(-2C_{2,m}+b^2m)\Big) \\
&\quad +m(2C_{1,m}+bm)p\Big]\,\varepsilon^3+O_M(\varepsilon^4).
\end{aligned} \tag{9}
$$

To provide the cubic convergence of the iterative method (6), the coefficients by ε and ε^2 must be 0. It is sufficient to take

$$
a=1,\quad b=p-\frac{C_{1,m}}{m}.
$$

Using these values of a and b we obtain the following iteration function for finding multiple zeros

$$
G(x)=x-v(x)\,\frac{1+p\,v(x)}{1+\Big(p-\frac{C_{1,m}}{m}\Big)\,v(x)}. \tag{10}
$$

The iterative formula (10) is not convenient for implementation since $C_{1,m}=\frac{f^{(m+1)}(\alpha)}{(m+1)f^{(m)}(\alpha)}$ is not available – the zero α is unknown. For this reason we will express $C_{1,m}$ in a more suitable form. From (7) we find

$$
u(x)=\frac{f(x)}{f'(x)}=\frac{\varepsilon}{m}-\frac{C_{1,m}}{m^2}\varepsilon^2+O_M(\varepsilon^3) \tag{11}
$$

and

$$
A_2(x)=\frac{(m-1)m+m(m+1)C_{1,m}\varepsilon+(m+1)(m+2)C_{2,m}\varepsilon^2+O_M(\varepsilon^3)}{2\varepsilon\,(m+(m+1)C_{1,m}\varepsilon+(m+2)C_{2,m}\varepsilon^2+O_M(\varepsilon^3))}. \tag{12}
$$

Combining (11) and (12), we obtain

$$
2mu(x)A_2(x)-m+1=\frac{2C_{1,m}\varepsilon}{m}+O_M(\varepsilon^2).
$$

Hence, taking into account the development (11), we calculate

$$
C_{1,m}=mA_2(x)-\frac{m-1}{2u(x)}+O_M(u(x)). \tag{13}
$$

Replacing (13) in (10) and neglecting higher terms of $u(x)$ (or ε, which is the same), we finally obtain the one-parameter iterative formula

$$
x_{k+1}=\mathcal{G}_m(x_k;p)=x_k-\frac{2mu(x_k)\big(1+mpu(x_k)\big)}{1+m+2m\big(p-A_2(x_k)\big)u(x_k)}\quad (k=0,1,\ldots). \tag{14}
$$

Note that the choice of $p = 0$ in (14) gives Halley-like method for finding multiple zeros [**15**]

$$x_{k+1} = x_k - \frac{u(x_k)}{\dfrac{m+1}{2m} - A_2(x_k)u(x_k)}. \tag{15}$$

Furthermore, from (9) we find the error-relation

$$\hat{\varepsilon} = \frac{2C_{2,m} - C_{1,m}^2 + pC_{1,m}}{m}\,\varepsilon^3 + O_M(\varepsilon^4). \tag{16}$$

From (16) we immediately obtain the following assertion.

Theorem 2. *Let x_0 be sufficiently close initial approximation to the zero α of the known multiplicity $m \geq 1$ of a given function f. Then the iterative method* (14) *is cubically convergent and*

$$\begin{aligned} AEC(14) &= \lim_{k\to\infty} \frac{|x_{k+1}-\alpha|}{\left|x_k - \alpha\right|^3} = \frac{1}{m}\left|2C_{2,m} - C_{1,m}^2 + p\,C_{1,m}\right| \\ &= \frac{1}{m}\left|\frac{pB_{m+1}}{B_m} - \frac{B_{m+2}}{B_m} + \frac{(m+1)B_{m+1}^2}{2mB_m^2}\right| \end{aligned} \tag{17}$$

is valid.

If the zero α is simple, setting $m = 1$ in (14) we obtain the one-parameter family of iterative methods

$$x_{k+1} = x_k - \frac{u(x_k)\Big(1 + p\,u(x_k)\Big)}{1 + \Big(p - A_2(x_k)\Big)u(x_k)} \quad (k = 0, 1, \ldots) \tag{18}$$

with the error-relation

$$\hat{\varepsilon} = (A_2^2 - A_3 + A_2 p)\varepsilon^3 + O_M(\varepsilon^4).$$

Hence, we find the asymptotic error constant

$$AEC(18) = \lim_{k\to\infty} \frac{|x_{k+1}-\alpha|}{\left|x_k - \alpha\right|^3} = \left|A_2^2(\alpha) - A_3(\alpha) + pA_2(\alpha)\right|.$$

Remark 1. In the special case when $p = 0$, from (18) we obtain Halley's method (4) with

$$AEC(4) = |A_2^2(\alpha) - A_3(\alpha)|,$$

which is well-known result. Later we will see that the methods with parameter p in the neighborhood $[-\delta, \delta]$ ($\delta \approx 1$) of "Halley's parameter" $p = 0$, produce the best results. For this reason, the families (18) and (14) will be referred to as *Generalized Halley's methods*. Another special case appears when $p \to \pm\infty$; then the method (14) reduces to quadratically convergent Schröder's method

$$x_{k+1} = x_k - m\,u(x_k).$$

This fact tells us that *one should avoid the choice of the parameter p large in magnitude.*

Remark 2. It is interesting to consider another special case $p = A_2(x_k)$: then the iterative process (18) switches to Chebyshev's method

$$x_{k+1} = x_k - u(x_k)\Big(1 + A_2(x_k)\,u(x_k)\Big) \quad (k = 0, 1, \ldots),$$

see (3). The described case can be helpful in finding suitable range (say, $[A_2(x_k) - \delta, A_2(x_k) + \delta]$) for the parameter p.

The impact of the parameter p to the accuracy of approximations to the zeros of a given function is very complex and it is hard to find its optimal value even within a particular class of functions. From the discussion given in Remark 1 we can conclude that large values of p are not convenient since then the order of convergence decreases and tends to 2. Furthermore, for $p = 0$ the method (18) reduces to Halley's method which belongs to the group of cubically convergent methods with very good convergence behavior, see, e.g, [**18**]–[**21**]. More extensive discussion on the choice of optimal parameter p is given in Sections 3 and 4.

3. THIRD ORDER METHODS – NUMERICAL RESULTS

The theoretical order of convergence of the iterative method (14) is three, see Theorem 2. However, it is always recommendable to check the convergence behavior in practice. For this reason, in our numerical experiments we have calculated the so-called *computational order of convergence* r_c (COC, for brevity) using the approximate formula

$$r_c = \frac{\log|f(x_{k+1})/f(x_k)|}{\log|f(x_k)/f(x_{k-1})|}. \tag{19}$$

Note that the formula (19) is a special case of a general formula given in [**22**]. The selection of tested functions is given in Tables 1 and 2. Functions from Table 2 have been also used for plotting basins of attraction in Section 5.

In Tables 3 and 4 we have presented the errors of approximations

$$\varepsilon_k = |x_k - \alpha| \ (k = 1, 2, 3)$$

produced by the method (14) for 5 values of the parameter p. The denotation $A(-h)$ means $A \times 10^{-h}$.

$f(x)$	m	x_0	α
$f_1(x)=\big(x\,\sin x-2\sin^2(x/\sqrt{2})\big)\big(x^5+x^2+100\big)$	6	-1.2	0
$f_2(x)=(xe^{x^2}-\sin^2 x+3\cos x+5)^2$	2	-1	$-1.2076478271309\ldots$
$f_3(x)=\big(e^{x^2+4x+5}-1\big)^3\sin^2(t+2-i)$	5	$-1.7+0.8i$	$-2+i$
$f_4(x)=\big(x-\sin x\big)^4$	12	0.4	0

Table 1: Tested functions for f_1-f_4

$f(x)$	m	x_0	α	all zeros
$f_5(x)=\big(x^5-x\big)^2$	2	1.2	1	$0,\ \pm1,\ \pm i$
$f_6(x)=\big(x^5+2x^4+2x^3+10x^2+25x\big)^2$	2	$0.8+1.8i$	$1+2i$	$0,\ \pm1\pm2i,\ -2\pm i$
$f_7(x)=\big(x^4-1\big)^5$	5	-1.2	-1	$\pm1,\ \pm i$
$f_8(x)=\big(x(x-1)(x+2)(x^2+4)\big)^3$	3	-2.2	-2	$-2,\ 0,\ 1,\ \pm2i$

Table 2: Tested functions for f_5-f_8

p	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	r_c (19)
$f_1(x)=\big(x\,\sin x-2\sin^2(x/\sqrt{2})\big)\big(x^5+x^2+100\big)$				
-2	2.29(−2)	1.40(−7)	2.84(−23)	3.011
-1	8.91(−4)	7.25(−12)	3.90(−36)	3.000
0	7.08(−2)	3.64(−6)	3.39(−19)	3.000
1	0.111	1.42(−2)	3.06(−8)	3.000
2	0.172	1.19(−5)	1.72(−17)	2.846
$f_2(x)=(xe^{x^2}-\sin^2 x+3\cos x+5)^2$				
-2	4.93(−2)	4.34(−4)	2.66(−10)	3.067
-1	1.87(−2)	1.17(−5)	2.82(−15)	3.013
0	7.99(−4)	1.29(−10)	5.50(−31)	3.000
1	1.10(−2)	1.65(−6)	5.64(−18)	2.994
2	1.93(−2)	2.04(−5)	2.32(−14)	2.991
$f_3(x)=(e^{x^2+4x+5}-1)^3\sin^2(t+2-i)$				
-2	6.17(−2)	1.74(−4)	3.45(−12)	3.031
-1	3.30(−2)	1.44(−5)	1.18(−15)	3.007
0	1.33(−2)	2.94(−7)	5.32(−20)	3.000
1	7.04(−2)	1.36(−7)	9.83(−22)	2.999
2	1.06(−2)	7.59(−7)	2.85(−19)	2.997
$f_4(x)=\big(x-\sin x\big)^4$				
-2	1.38(−2)	4.47(−8)	1.78(−24)	3.067
-1	3.21(−3)	5.59(−10)	2.91(−30)	3.001
0	1.08(−3)	2.08(−11)	1.50(−34)	3.000
1	1.58(−4)	6.52(−14)	4.63(−42)	3.000
2	3.53(−4)	7.37(−13)	6.68(−39)	3.000

Table 3: Errors of approximations; functions f_1-f_4

p	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	r_c (19)
$f_5(x) = (x^5 - x)^2$				
-2	$7.85(-5)$	$6.04(-13)$	$2.75(-37)$	2.999
-1	$9.22(-5)$	$9.80(-13)$	$1.18(-36)$	3.000
0	$2.76(-4)$	$7.93(-11)$	$1.87(-30)$	3.000
1	$4.76(-4)$	$6.75(-10)$	$1.92(-27)$	3.000
2	$6.93(-4)$	$2.92(-9)$	$2.17(-25)$	3.001
$f_6(x) = (x^5 + 2x^4 + 2x^3 + 10x^2 + 25x)^2$				
-2	$4.59(-2)$	$1.77(-4)$	$1.07(-11)$	2.968
-1	$2.70(-2)$	$1.79(-5)$	$5.34(-15)$	2.990
0	$2.84(-2)$	$1.88(-5)$	$5.23(-15)$	3.016
1	$6.93(-2)$	$6.49(-4)$	$4.69(-10)$	3.078
2	0.125	$6.15(-3)$	$6.41(-7)$	3.167
$f_7(x) = (x^4 - 1)^5$				
-2	$1.77(-2)$	$2.20(-5)$	$4.52(-14)$	2.979
-1	$1.33(-2)$	$6.25(-6)$	$6.72(-16)$	2.988
0	$7.25(-3)$	$4.72(-7)$	$1.31(-19)$	2.995
1	$1.68(-3)$	$1.19(-9)$	$4.25(-28)$	3.001
2	$1.62(-2)$	$7.45(-6)$	$7.24(-16)$	3.009
$f_8(x) = \big(x(x-1)(x+2)(x^2+4)\big)^3$				
-2	$1.56(-2)$	$1.34(-5)$	$8.91(-15)$	2.983
-1	$1.16(-2)$	$3.59(-6)$	$1.11(-16)$	2.990
0	$5.96(-3)$	$2.24(-7)$	$1.21(-20)$	2.996
1	$2.32(-3)$	$3.30(-9)$	$9.47(-27)$	3.001
2	$1.59(-2)$	$6.32(-6)$	$4.03(-16)$	3.007

Table 4: Errors of approximations; functions $f_5 - f_8$

The most accurate approximations, obtained after the third iterative step, are shaded in Tables 3 and 4 (and, also, in Tables 7 and 8). We observe that the best results are obtained taking $p = -1$ for f_1, $p = 0$ for f_2 and f_6, $p = 1$ for f_3, f_4, f_7, f_8 and $p = -2$ for f_5. Note that the values of p are given by integers; it is very likely that the use of p having one or two decimal digits of mantissa would give more accurate approximations.

Apart from the functions listed in Tables 1 and 2, we have also tested a number of functions of various structure and found that mainly the most accurate approximations are obtained for $p \in [-1, 1]$. However, we have not found the value of p which defines approximately the best method from the family (14). Having in mind this fact, appart from the accuracy of approximations, we also investigate some other very important characteristics of iterative methods, first of all a domain of initial approximations which guarantees the convergence. For this purpose we study basins of attraction in Section 4 to provide the size of this domain and some useful additional information on the quality of methods from the family (14).

Remark 3. The values of COC r_c in Tables 3 and 4 are taken with 3 decimal digits of mantissa. However, in some cases unexpected values of r_c appear. The explanation is simple: formula (19) works well when the approximations x_{k-1}, x_k, x_{k+1}

are sufficiently close to the zero. One additional iteration would give more realistic value of r_c.

4. DYNAMIC STUDY OF THE THIRD ORDER FAMILY

As mentioned in Section 2, the estimations of quality of iterative methods by classical methods (numerical experiments, computational efficiency) are often not sufficient to give a proper rank of these methods since these approaches strongly depend on the closeness of starting approximations to the sought zero. Actually, finding closed sets in the complex plane consisting of the initial points for which a method converges to the zero of a given function f, is not an easy task. Moreover, these sets most frequently have complex and intricate structure, including blobs and fractals, see, e.g., [**23**] and [**24**].

To give a more precise and realistic insight into the convergence characteristics and behavior of iterative methods, at the beginning of the 21st century a new methodology (denoted as 2) in (1)), based on the notion of *basins of attraction*, was developed. Apart from visualization of convergence behavior, this methodology gives very useful information on iterations such as CPU time for each basin, average number of iterations of the applied iterative methods and the percentage of divergent points regarding all initial points of the basin.

Definition 1. *Let f be a given sufficiently many times differentiable function in some complex domain $S \subseteq \mathbb{C}$ with simple or multiple zeros $\alpha_1, \alpha_2, \ldots, \alpha_\lambda \in S$, and a (convergent) root-finding iteration defined as*

$$x_{k+1} = g(x_k),$$

the basin of attraction for the zero α_i is defined to be

$$\mathcal{B}_{f,g}(\alpha_i) = \{\xi \in S \,|\, \textit{the iteration } x_{k+1} = g(x_k) \textit{ with } z_0 = \xi \textit{ converges to } \alpha_i\}.$$

It is worth emphasizing that an iterative method has a very good convergence behavior if (i) the boundaries of particular basins of attraction, corresponding to each zero of the test function, are straight lines and (ii) these basins, including their boundaries, have fewer number (preferably none) of blobs, fractals and divergent points. Another very important property is small CPU time and small average number of iterations considering all points from the complex domain S.

The dynamic study for the comparison of iterative methods for finding simple zeros has appeared in the papers by Vrscay and Gilbert [**25**], Stewart [**26**] and Varona [**27**]. Basins of attraction for methods for finding multiple zeros can be found in the papers of Neta and Chun [**12**], [**23**], [**28**], [**29**] and [**30**]. In this section we give the dynamic study of the iterative methods (14) for the functions $f_5,\ f_6,\ f_7$ and f_8, given in Table 2, and the values -4, -3, -2, -1, 0, 1, 2, 3 of the parameter p.

Plotting basins of attraction we assign a color to each basin of attraction of a root in the following manner: each basin will have a different color and the shading is as darker as the number of iterations rises. Starting points for which the method does not converge are colored black. We allow the maximum of 40 iterations from every initial point; if the number of iterations exceeds 40 then we treat the initial point as divergent one and paint it black. All methods have been tested on the 360 000 equally spaced points of the square $S = \{-3, 3\} \times \{-3, 3\}$ centered at the origin. The corresponding basins of attraction are given in Figures 1–12. For each basin we collect data concerning the CPU time for all 360 000 points, average number of iterations (for all points of the square S) required to reach the accuracy

$$|x_k - \alpha| < 10^{-7}$$

and the number of divergent points for each method and each example. These data are given in Table 5.

For the sake of comparison, dynamic study of the iterative methods from the new family (14) has been carried out using computer algebra systems Maple and Mathematica. As expected, the obtained figures of basins of attraction coincide, and the same holds for the average numbers of iterations and the numbers of divergent ("black") points for some methods, see Table 5. However, absolute values CPU times $t^{(k)}(p)$ (expressed in seconds), in plotting basins for a fixed function f_k, are not equal, which is the consequence of different computer algebra systems employed (in our case Maple and Mathematica). To avoid these non-influential entries in the comparison procedure, we have dealt with *normalized CPU times* $T^{(k)}_{norm}(p)$ calculated relative to a fixed method. In our case we have taken Halley's method ($p = 0$) since this method consumes the minimal CPU time for all tested examples, that is,

$$T^{(k)}_{norm}(p) = \frac{t^{(k)}(p)}{t^{(k)}(0)}, \quad \text{with} \quad T^{(k)}_{norm}(0) = 1,$$

for a fixed $k \in \{5, 6, 7, 8\}$, see Table 5. This approach has an additional important advantage; the influence of characteristics of a micro-processor embedded in digital computer is eliminated.

Comments on the basins of attraction: According to Figures 1–12 and entries given in Table 5 we can conclude that, in most cases, the basins of attraction and their boundaries of Halley's method $(14)_{p=0}$ have fewest numbers of blobs and fractals. Besides, Halley's method has the smallest number of divergent points in comparison with the remaining methods, only 0.08% in average for all 4 examples and even 0 divergent points for the functions f_5, f_6 and f_8, see Table 5. These values show that Halley's method $(14)_{p=0}$ possesses the best convergence properties in regard to the domain of convergence. From Table 5 we observe that Halley's method reaches the given tolerance with the smallest number of iterations for all tested methods and consumes the smallest CPU time (sharing it with the method $(14)_{p=1}$ for f_8). Altogether, in regard to the dynamic study of the methods from the family (14), Halley's method $(14)_{p=0}$ convincingly shows the best convergence characteristics.

$f_5(x) = (x^5 - x)^2$						
	A		B		C	
p	Maple 18	Math. 10	Maple 18	Math. 10	Maple 18	Math. 10
-4	1.89	1.92	10.84	10.85	9.98	10.0
-3	2.13	2.14	12.11	12.11	14.87	14.86
-1	1.01	1.02	5.71	5.72	0.15	0.16
0	1	1	5.65	5.65	0	0
1	1.04	1.02	5.71	5.72	0.15	0.16
2	2.04	2.03	11.66	11.66	15.88	15.05
3	2.05	2.15	12.11	12.11	14.87	14.86
$f_6(x) = (x^5 + 2x^4 + 2x^3 + 10x^2 + 25x)^2$						
-4	1.52	1.53	6.33	6.33	0	0
-3	1.53	1.52	6.32	6.13	0	0
-2	1.88	1.86	7.77	7.77	3.69	3.68
-1	3.49	3.22	13.37	13.40	25.08	25.07
0	1	1	4.22	4.22	0	0
1	1.42	1.41	5.84	5.85	2.98	3.0
2	1.59	1.55	6.57	6.56	1.26	1.16
3	1.41	1.43	5.93	5.53	0	0
$f_7(x) = (x^4 - 1)^5$						
-4	2.01	2.11	10.78	10.76	6.65	6.56
-3	2.18	2.25	11.81	11.80	13.02	12.95
-2	2.01	2.13	11.14	11.14	13.66	13.64
-1	1.24	1.23	6.37	6.36	3.01	3.01
0	1	1	5.25	5.25	0.33	0.31
1	1.24	1.23	6.37	6.36	3.01	3.01
2	2.00	2.19	11.14	11.14	13.66	13.64
3	2.10	2.30	11.81	11.80	13.02	12.95
$f_8(x) = (x(x-1)(x+2)(x^2+4))^3$						
-4	1.78	1.73	8.53	8.53	5.08	5.08
-3	1.86	1.95	9.46	9.46	9.08	9.09
-2	2.23	2.13	10.51	10.51	13.87	13.64
-1	1.21	1.17	5.78	5.78	2.76	2.75
0	1	1	4.90	4.90	0	0
1	0.99	1.01	4.83	4.83	0.34	0.34
2	1.31	1.33	6.43	6.42	1.93	1.94
3	1.40	1.37	6.73	6.72	1.69	1.69
average values						
-4	1.80	1.82	9.12	9.12	5.43	5.41
-3	1.93	1.96	9.93	9.87	9.24	9.22
-2	2.07	2.02	10.27	10.27	11.78	11.56
-1	1.73	1.66	7.81	7.82	7.75	7.70
0	1	1	5.01	5.00	0.08	0.08
1	1.17	1.17	5.69	5.69	1.62	1.63
2	1.73	1.77	8.95	8.95	8.18	7.95
3	1.74	1.81	9.15	9.04	7.40	7.32

A – Normalized CPU time compared to Halley's method ($p = 0$); B – Average number of iterations for all starting points; C – Number of "black" points, expressed also as a percentage (in parenthesis).

Table 5: Iteration data for $f_5 - f_8$ obtained by Maple 18 and Mathematica 10

5. ONE-PARAMETER FAMILY OF ITERATIVE METHODS OF ORDER FOUR

An iterative method for solving nonlinear equations of the form $f(x) = 0$ can be accelerated using the following theorem, see [**32, 33**].

Theorem 3. *Let* $x_{k+1} = \phi_r(x_k) \quad (k = 0, 1, \ldots)$ *be an iterative method of order* r *for finding a simple or multiple zero of a given function* f *(sufficiently many times differentiable). Then the iterative method defined by*

$$x_{k+1} = \phi_{r+1}(x_k) := x_k - \frac{x_k - \phi_r(x_k)}{1 - \dfrac{1}{r}\phi_r'(x_k)} \quad (r \geq 2;\ k = 0, 1, \ldots), \tag{20}$$

has the order of convergence $r + 1$.

Milovanović [**34**] derived the accelerating formula in Banach space. In the recent paper [**35**] Gnang and Dubeau presented a class of iterative formulas which, as a special case, contains Jovanović's formula (20).

Let $r = 3$ and $\mathcal{G}_m(x_k; p) = \phi_3(x_k)$, where $\mathcal{G}_m(x_k; p)$ is defined by (14). Using the accelerating formula (20), by symbolic computation in computer algebra system *Mathematica* we obtain the fourth order one-parameter family of iterative methods (suppressing the iteration index k and the argument x_k for simplicity)

$$\phi_4(x; m, p) = x - \frac{Q(u; m, p)}{R(u; m, p)}, \tag{21}$$

where Q and R are polynomials of variable u given by

$$\begin{aligned} Q(u; m, p) &= 3mu(mpu + 1)(-2A_2mu + 2mpu + m + 1), \\ R(u; m, p) &= 1 + m\Big(3 + 2m + 2(1 + m)\big((2 + m)p - 3A_2\big)u \\ &\qquad + 2m\big(3A_3 - p(2 + m)(3A_2 - p)\big)u^2 + 2m^2p(3A_3 - 2pA_2)u^3\Big). \end{aligned}$$

If $p = 0$, the iterative formula (21) gives Halley-like method of order four

$$\phi_4(x; m, 0) = x - \frac{3mu(1 + m - 2mA_2u)}{1 + 3m + 2m^2 - 6mA_2(1 + m)u + 6m^2A_3u^2}\,, \tag{22}$$

derived in [**14**] in a different way.

Let α be a simple zero of f $(m = 1)$, then (21) becomes

$$\phi_4(x; 1, p) = x - \frac{3u(1 + pu)(1 - A_2u + pu)}{3(1 + pu)(1 + pu + A_3u^2) - uA_2(6 + 9pu + 2p^2u^2)}\,. \tag{23}$$

To our knowledge, the iterative formulas (21) and (23) are new.

In a special case $p = 0$ the iterative formula (23) reduces to

$$\phi_4(0, x) = x - \frac{u(1 - A_2 u)}{1 - 2A_2 u + A_3 u^2}, \tag{24}$$

which is Kiss' iterative method of order four, see [**36**]. It can also be derived from (22) setting $m = 1$.

6. NUMERICAL EXAMPLES AND DYNAMIC STUDY OF THE FOURTH ORDER FAMILY

In this section we present the results of numerical experiments obtained by applying the fourth order family (21) for simple zeros (functions $g_1 - g_4$) and multiple zeros (functions $g_5 - g_7$), and for 5 values of parameters -2, -1, 0, 1, 2. The functions $g_1 - g_7$ are listed in Table 6. The entries of the absolute values of approximation errors are displayed in Tables 7 and 8 for three iterations. The best approximations obtained in the third step are shaded. We observe that the choice of the parameter p in the interval $[-1, 1]$ gives the best results. The Halley-like methods (22) and (24) ($p = 0$) produce the best approximations in 4 (of 7) tested functions.

$g(x)$	m	x_0	α
$g_1(x) = \Big(\exp(x^2 + 6x + 16) - 1\Big) \sin(x + 2 - i)$	1	$-1.8 + 0.8i$	$-2 + i$
$g_2(x) = \big(x^5 + x^2 + 100\big)\big(x \sin x - 2\sin^2(x/\sqrt{2})\big)^{1/6}$	1	-0.7	0
$g_3(x) = x \exp(x^2) - \sin^2 x + 3\cos x + 5$	1	-1	$-1.2076478\ldots$
$g_4(x) = \sin(x + 5)\big(x^5 - 3x^4 - 5x^2 - 6x + 13\big)$	1	0.8	1
$g_5(x) = \big(x - \sin x\big)^4$	12	0.4	0
$g_6(x) = \big(x^4 - 1\big)^5$	5	-1.2	-1
$g_7(x) = \big(\exp(x^2 + 6x - 16) - 1\big)\big((x - 1)^6 - 1\big)$	2	2.2	2

Table 6: Tested functions for $g_1 - g_7$

Dynamic study of the fourth order family (21) has been performed for the iterative methods (23) (the case of simple zeros) and for 6 parameters -1, -0.5, 0, 0.5, 1, 1.5. The tested functions $g_8 - g_{10}$ are listed in Table 9 together with all zeros. The corresponding basins of attractions are given in Figures 13–18.

As in the case of the third order family (14), for each basin we collect data concerning the CPU time for all 360 000 points, average number of iterations (for all points of the square $S = \{-3, 3\} \times \{-3, 3\}$) required to reach the accuracy $|x_k - \alpha| < 10^{-7}$ and the number of divergent points (given in %) for each method and each example using computer algebra system Maple 18, see Table 10.

p	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	r_c (19)
$g_1(x)=\big(\exp(x^2+6x+16)-1\big)\ \sin(x+2-i)$				
-2	1.74(−2)	4.07(−7)	1.18(−25)	4.007
-1	1.06(−2)	4.60(−8)	1.64(−29)	3.995
0	1.13(−2)	6.87(−8)	9.85(−29)	3.987
1	1.68(−2)	5.42(−7)	6.24(−25)	3.976
2	2.37(−2)	3.53(−6)	1.89(−21)	3.962
$g_2(x)=(x^2+x+100)\big(x\sin x-2\sin^2(x/\sqrt{2})\big)^{1/6}$				
-2	3.85(−4)	2.04(−16)	1.60(−65)	4.000
-1	1.25(−4)	3.01(−19)	1.03(−77)	4.000
0	5.91(−5)	9.78(−22)	7.32(−89)	4.000
1	1.53(−2)	2.78(−10)	3.16(−41)	3.998
2	2.62(−2)	7.31(−9)	4.94(−35)	3.993
$g_3(x)=x\exp(x^2)-\sin^2 x+3\cos x+5$				
-2	1.63(−2)	1.43(−7)	9.13(−28)	3.987
-1	1.60(−3)	2.51(−12)	1.52(−47)	3.999
0	2.71(−4)	2.00(−15)	5.99(−60)	4.000
1	2.80(−4)	7.74(−16)	4.47(−62)	4.000
2	1.39(−3)	4.16(−12)	3.36(−46)	3.999
$g_4(x)=\big(x^9+x^2+1\big)\big(x^6-3x^4-5x^2-6x+13\big)$				
-2	6.94(−3)	2.85(−8)	7.66(−30)	3.994
-1	3.92(−3)	3.04(−9)	1.09(−33)	3.996
0	1.54(−3)	6.20(−11)	1.65(−40)	3.993
1	4.08(−3)	1.83(−9)	7.68(−35)	3.993
2	6.46(−2)	9.97(−5)	1.01(−17)	4.433*

Table 7: Errors of approximations; functions g_1-g_4, simple zeros – IM (23)

p	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	r_c (19)
$g_5(x)=(x-\sin x)^4$				
-2	2.29(−2)	8.39(−9)	1.65(−34)	3.954
-1	1.21(−3)	3.55(−14)	2.65(−56)	4.000
0	6.52(−6)	7.50(−30)	1.51(−149)	5.000**
1	2.15(−4)	3.59(−17)	2.76(−68)	4.000
2	2.63(−4)	1.60(−16)	2.18(−65)	4.000
$g_6(x)=\big(x^4-1\big)^5$				
-2	3.721(−3)	9.69(−10)	4.52(−36)	3.997
-1	2.25(−3)	6.07(−11)	3.22(−31)	3.999
0	8.30(−4)	2.97(−13)	4.83(−51)	3.999
1	3.27(−5)	1.44(−19)	5.32(−77)	4.000
2	1.34(−3)	4.12(−13)	3.60(−51)	4.001
$g_7(x)=\big(\exp(x^2+6x-16)-1\big)\big((x-1)^6-1\big)$				
-2	5.56(−3)	4.28(−9)	1.52(−33)	4.006
-1	8.87(−4)	2.19(−12)	8.03(−47)	4.000
0	4.48(−3)	2.71(−11)	1.80(−45)	4.155
1	9.51(−3)	4.97(−8)	3.69(−29)	3.988
2	1.40(−2)	5.35(−7)	1.20(−24)	3.975

Table 8: Errors of approximations; functions g_5-g_7, multiple zeros – IM (21)

$g(x)$	all zeros
$g_8(x) = x^5 - x$	$0,\ \pm 1,\ \pm i$
$g_9(x) = x^4 + 2x^3 + 2x^2 + 10x + 25$	$\pm 1 \pm 2i,\ -2 \pm i$
$g_{10}(x) = \big(\exp(x+1) - 1\big)(x-1)$	± 1

Table 9: Tested functions for $g_8 - g_{10}$

	$p=-1$	$p=-0.5$	$p=0$	$p=0.5$	$p=1$	$p=1.5$
$g_8(x) = x^5 - x$						
A	1.07	1.08	1	1.08	1.02	2.01
B	4.46	4.50	4.61	4.50	4.46	8.62
C	0.19	0.08	0	0.08	0.19	11.4
$g_9(x) = x^4 + 3x^3 + 2x^2 + 10x + 25$						
A	2.46	1.33	1	1.26	1.80	1.85
B	7.71	3.89	3.22	3.61	5.56	5.79
C	8.63	2.21	0.17	0.70	5.72	3.62
$g_{10}(x) = \big(e^{x+1} - 1\big)(x-1)$						
A	2.46	0.96	1	0.97	1.04	1.06
B	7.99	3.06	3.44	3.24	3.39	3.41
C	12.93	0.09	0.40	0.22	0.16	0.15
average values						
A	2.00	1.12	1	1.10	1.29	1.64
B	6.72	3.82	3.76	3.78	4.47	5.94
C	7.25	0.79	0.19	0.33	2.02	5.06

A – CPU time normalized related to $(23)_{p=0}$; B – Average number of iterations per point; C – Percentage of divergent ("black") points

Table 10: Iteration data for $g_8 - g_{10}$

Comments on the basins of attraction: According to Figures 13–16 and entries given in Table 10 we can conclude that, in most cases, the attraction basins and their boundaries of Halley's method $(24)_{p=0}$ have fewest number of blobs and fractals and require minimal CPU time and minimal average number of iterations. Results of another set of examples, not presented in this paper to save space, have confirmed that the fourth order Halley-like method (24) possesses the best convergence properties. In overall, the data presented in Table 10 show that the choice of the parameter p from the interval $[-0.5, 0.5]$ produce the best results.

Numerical examples presented in Section 3 (for the third order family (14)) and Section 6 (for the fourth order family (21)) have shown that the most accurate approximations are obtained choosing the parameter p in the interval $[-1, 1]$ in the case of the third order family (14) and $[-0.5, 0.5]$ for the fourth order family (21). Having in mind that the executed tests are of numerical nature and the number of tested functions is finite, the bounds of the above intervals are only approximative. Since the parameter $p = 0$, which defines Halley-like methods (3), (15), (22) and (24), belongs to these intervals, we can conclude that the choice of p close to 0 is the best choice in most cases both for the third order family and the fourth order family. Taking into account the results of dynamic study of the considered families presented in Sections 4 and 5, it is indisputable that the choice $p = 0$ (methods of Halley's type) yields the best convergence properties for both families (14) and (21).

The presented characteristics can help the user to choose between (almost) global convergence and more accurate approximations. From the practical point of view, the first option is preferable in solving practical problems since extremely accurate approximations are beyond practical necessity, while safe convergence to the solution is ultimately of great importance – the latter property defines the *reliability* of the applied method, very desirable feature. From this aspect, Halley-like methods ($p = 0$) are the most serious candidates for the best methods from the families (14) and (21).

We end this paper with the comment that the developed Halley-like iterative methods can be useful as the base for constructing efficient algorithms for finding all zeros of polynomials, simultaneously, a subject for further work. For illustration, let us consider the iterative formula (23) of the fourth order. Let $f(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_1x + a_0$ be a monic polynomial of degree n with simple zeros $\alpha_1, \ldots, \alpha_n$ and let $x_1, \ldots, x_n$ be their respective approximations. Define the rational functions in the complex domain

$$\delta_{q,i}(x) = \frac{f^{(q)}(x_i)}{f_(x_i)} \quad (q = 1, 2, 3), \quad W_i(x) = \frac{f(x)}{\prod\limits_{j=1, j\neq i}^{n} (x - x_j)}.$$

We observe that $W_i(x)$ has the same zeros as the polynomial f. For $x = x_i$ introduce

$$\mathcal{B}_{k,i} = \frac{W_i^{(k)}}{W_i'} \quad (k = 0, 1, \ldots), \quad S_{q,i} = \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{1}{(x_i - x_j)^q} \quad (q = 1, 2, 3),$$

$$U_{0,i} = \delta_{1,i} - S_{1,i}, \quad U_{1,i} = \delta_{2,i} - \delta_{1,i}^2 + S_{2,i}, \quad U_{2,i} = \delta_{3,i} - 3\delta_{1,i}\delta_{2,i} + 2\delta_{1,i}^3 - 2S_{3,i},$$

where the argument x_i is omitted for simplicity. Then we find

$$\mathcal{B}_{0,i} = \frac{1}{U_{0,i}}, \quad \mathcal{B}_{2,i} = U_{0,i} + \mathcal{B}_{0,i}U_{1,i}, \quad \mathcal{B}_{3,i} = \mathcal{B}_{2,i}U_{0,i} + 2U_{1,i} + \mathcal{B}_{0,i}U_{2,i}.$$

Substituting $u(x_i)$ with $\mathcal{B}_{0,i}$, $A_2(x_i)$ by $\mathcal{B}_{2,i}/2$ and $A_3(x_i)$ by $\mathcal{B}_{3,i}/6$ in (23), we construct the following fifth order simultaneous methods for approximating all the zeros of f :

$$\hat{x}_i = x_i - \frac{3\mathcal{B}_{0,i}(1 + p\mathcal{B}_{0,i})\big(2 + \mathcal{B}_{0,i}(2p - \mathcal{B}_{2,i})\big)}{(1 + p\mathcal{B}_{0,i})(6 + \mathcal{B}_{0,i}^2\mathcal{B}_{3,i} + 6p\mathcal{B}_{0,i}) - \mathcal{B}_{0,i}\mathcal{B}_{2,i}\big(6 + p\mathcal{B}_{0,i}(9 + 2p\mathcal{B}_{0,i})\big)}.$$

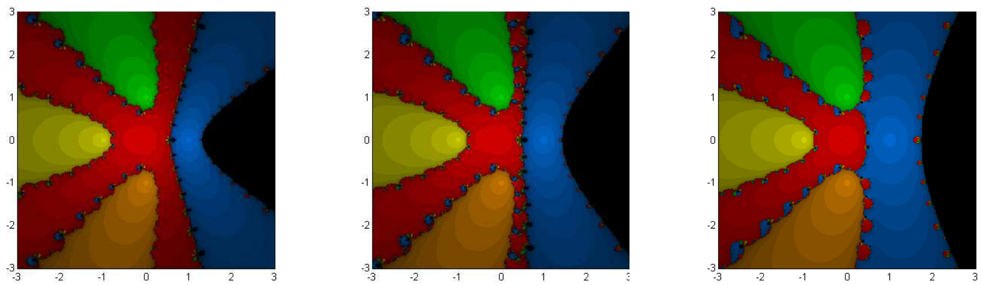

Figure 1: $p = -4,\ p = -3,\ p = -2,\ \ f_5(x) = \left(x^5 - x\right)^2$

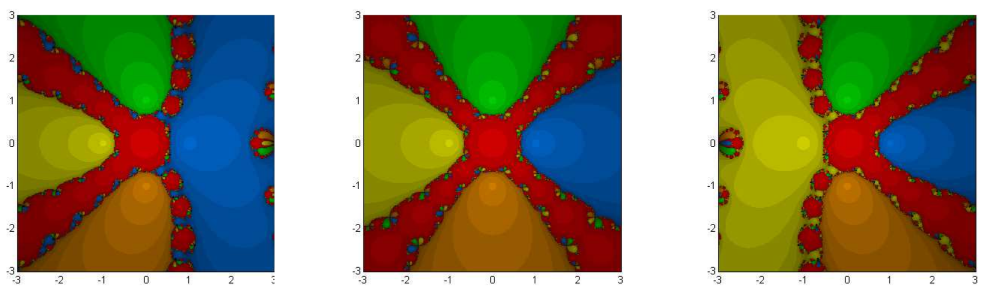

Figure 2: $p = -1,\ p = 0,\ p = 1,\ \ f_5(x) = \left(x^5 - x\right)^2$

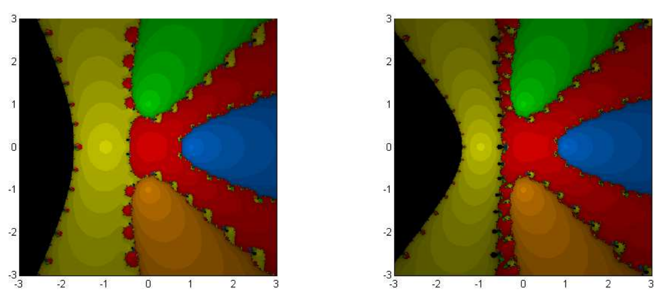

Figure 3: $p = 2,\ p = 3,\ \ f_5(x) = \left(x^5 - x\right)^2$

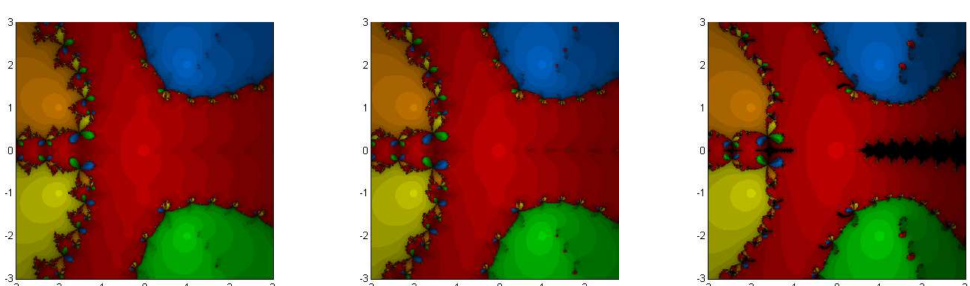

Figure 4: $p = -4,\ p = -3,\ p = -2,\ \ f_6(x) = \left(x^5 + 2x^4 + 2x^3 + 10x^2 + 25x\right)^2$

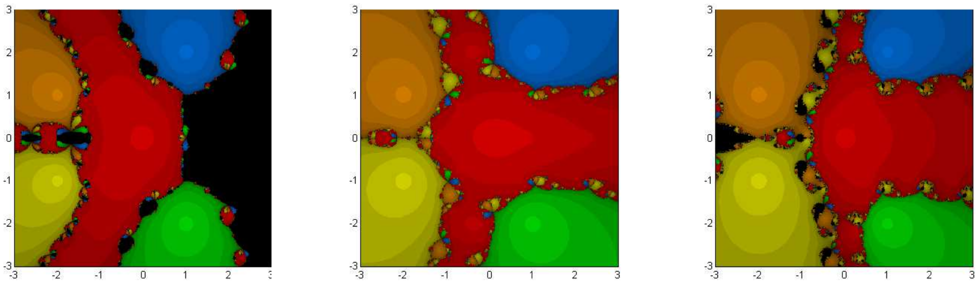

Figure 5: $p = -1,\ p = 0,\ p = 1,\ \ f_6(x) = \left(x^5 + 2x^4 + 2x^3 + 10x^2 + 25x\right)^2$

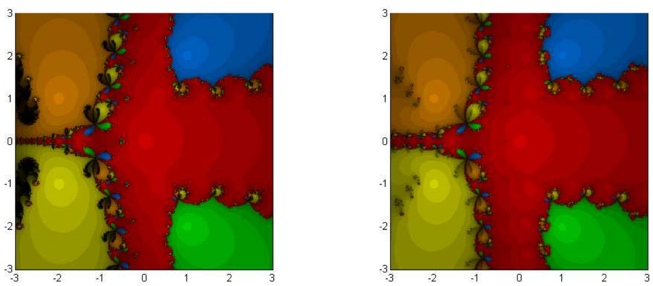

Figure 6: $p = 2,\ p = 3,\ \ f_6(x) = \left(x^5 + 2x^4 + 2x^3 + 10x^2 + 25x\right)^2$

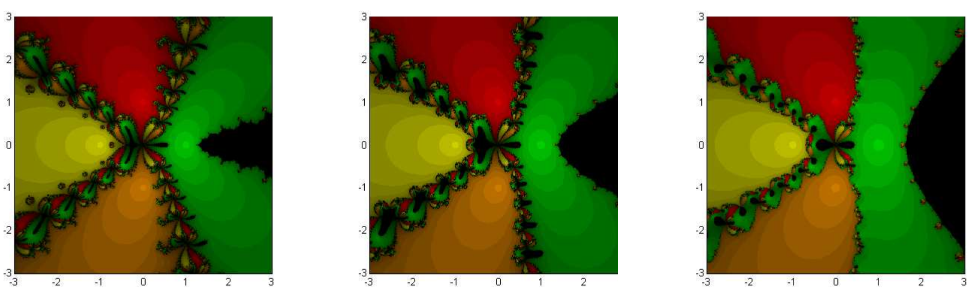

Figure 7: $p = -4,\ p = -3,\ p = -2,\ \ f_7(x) = \left(x^4 - 1\right)^5$

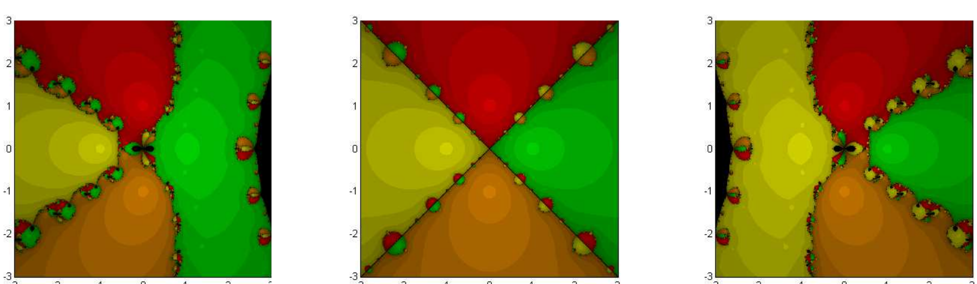

Figure 8: $p = -1,\ p = 0,\ p = 1,\ \ f_7(x) = \left(x^4 - 1\right)^5$

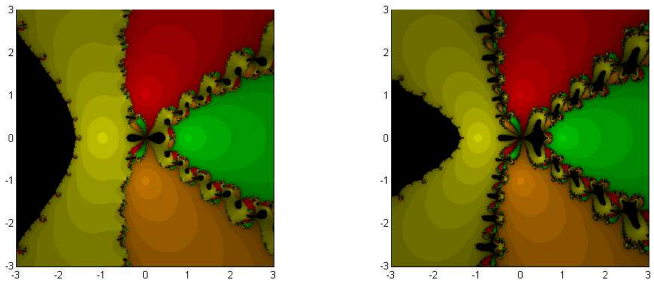

Figure 9: $p=2,\ p=3,\ \ f_7(x)=\left(x^4-1\right)^5$

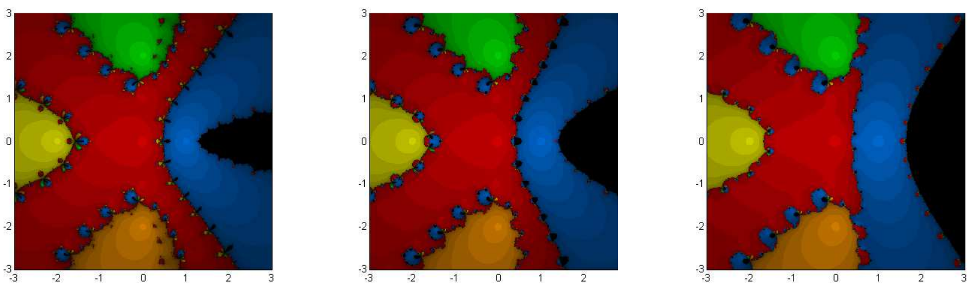

Figure 10: $p=-4,\ p=-3,\ p=-2,\ \ f_8(x)=\left(x(x-1)(x+2)(x^2+4)\right)^3$

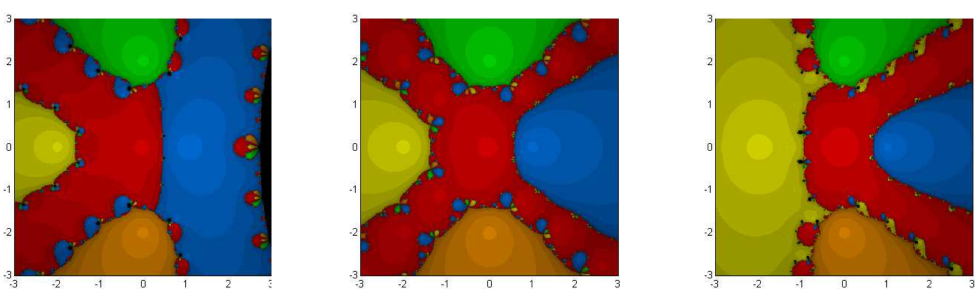

Figure 11: $p=-1,\ p=0,\ p=1,\ \ f_8(x)=\left(x(x-1)(x+2)(x^2+4)\right)^3$

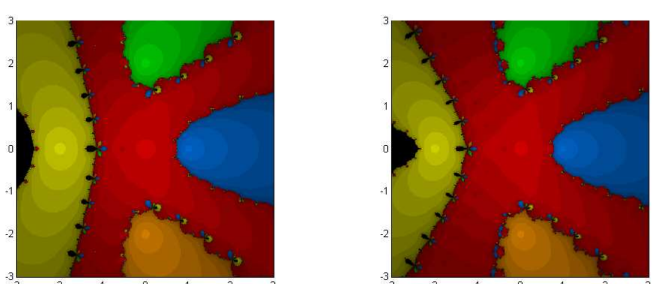

Figure 12: $p=2,\ p=3,\ \ f_8(x)=\left(x(x-1)(x+2)(x^2+4)\right)^3$

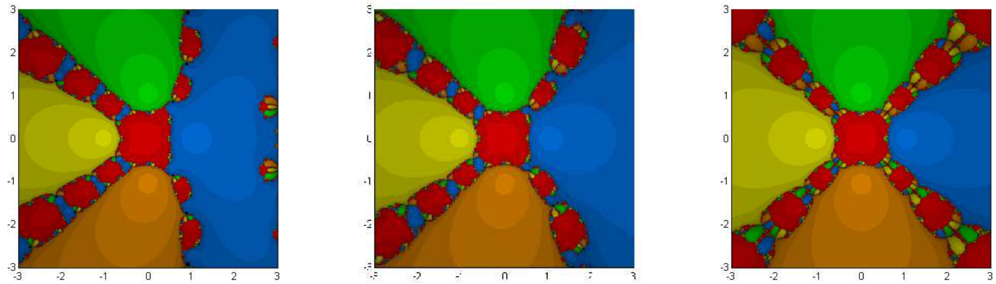

Figure 13: $p = -1,\ p = -0.5,\ p = 0,\ g_8(x) = x^5 - x$

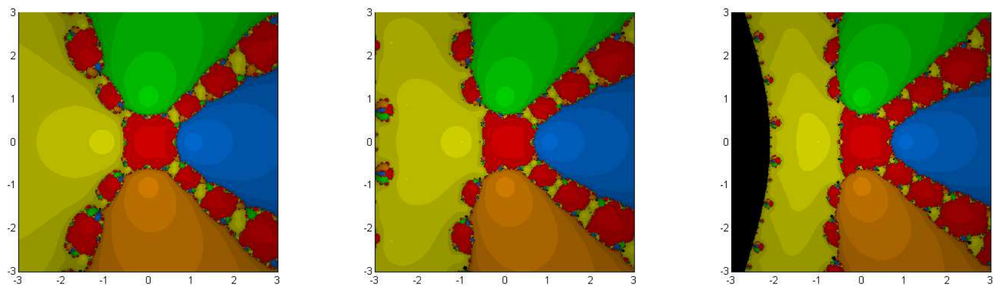

Figure 14: $p = 0.5,\ p = 1,\ p = 1.5,\ g_8(x) = x^5 - x$

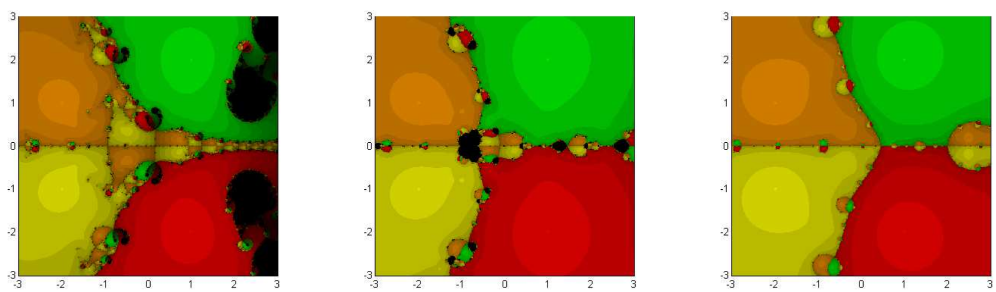

Figure 15: $p = -1,\ p = -0.5,\ p = 0,\ g_9(x) = x^4 + 2x^3 + 2x^2 + 10x + 25$

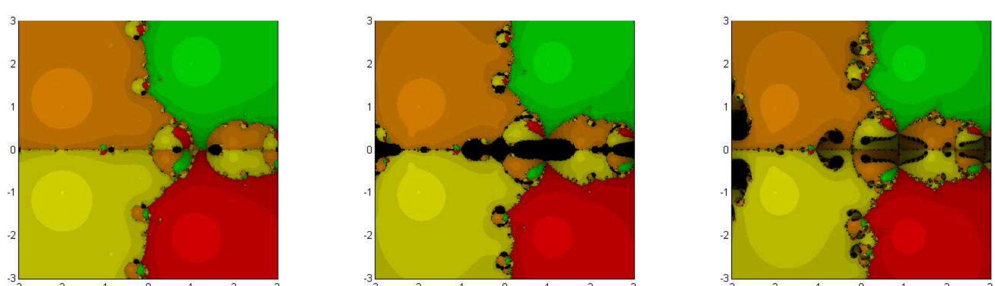

Figure 16: $p = 0.5,\ p = 1,\ p = 1.5,\ g_9(x) = x^4 + 2x^3 + 2x^2 + 10x + 25$

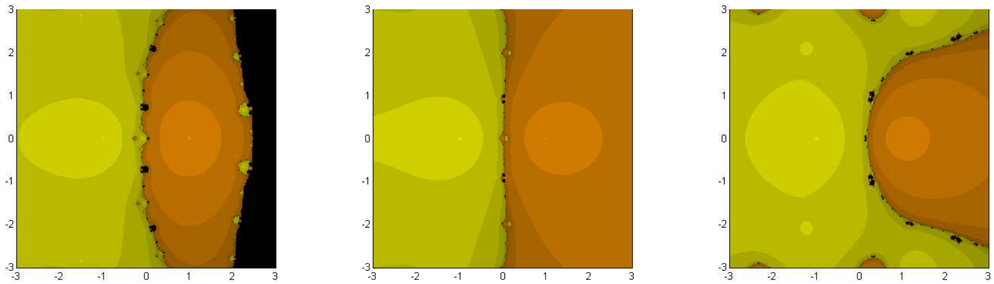

Figure 17: $p=-1,\ p=-0.5,\ p=0,\ g_{10}=f_z=\big(e^{x+1}-1\big)(x-1)$

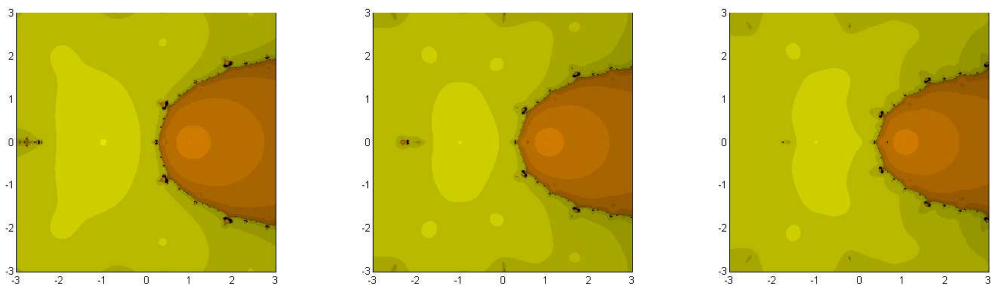

Figure 18: $p=0.5,\ p=1,\ p=1.5,\ g_{10}=\big(e^{x+1}-1\big)(x-1)$

Acknowledgements. This work was supported by the Serbian Ministry of Education and Science.

REFERENCES

1. M. S. Petković, B. Neta, L. D. Petković, J. Džunić: *Multipoint Methods for Solving Nonlinear Equations.* Elsevier/Academic Press, Amsterdam, 2013.
2. C. Chun, B. Neta: *A third-order modification of Newton's method for multiple roots.* Appl. Math. Comput., **211** (2009), 474–479.
3. S. Li, L. Cheng, B. Neta: *Some fourth-order nonlinear solvers with closed formulae for multiple roots.* Comp. Math. Appl., **59** (2010), 126–135.
4. S. Li, X. Liao, L. Cheng: *A new fourth-order iterative method for finding multiple roots of nonlinear equations.* Appl. Math. Comput., **215** (2009), 1288–1292.
5. B. Neta, A. N. Johnson: *High-order nonlinear solver for multiple roots.* Comp. Math. Appl., **55** (2008), 2012–2017.
6. J. R. Sharma, R. Sharma: *Modified Jarratt method for computing multiple roots.* Appl. Math. Comutut., **217** (2010), 878–881.

7. X. Zhou, X. Chen, Y. Song: *Construction of higher order methods for multiple roots of nonlinear equations.* J. Comput. Appl. Math., **235** (2011), 4199–4206.

8. P. Kravanja, M. Van Barel: *Computing the Zeros of Analytic Functions.* Lecture Notes in Mathematics 1727, Springer, Berlin, 2000.

9. B. I. Yun: *A non-iterative method for solving nonlinear equations.* Appl. Math. Comput., **198** (2008), 691–699.

10. B. I. Yun: *Transformation methods for finding multiple roots of nonlinear equations.* Appl. Math. Comput., **217** (2010), 599–606.

11. J. F. Traub: *Iterative Methods for the Solution of Equations.* Prentice-Hall, New Jersey, 1964.

12. B. Neta, C. Chun: *On a family of Laguerre methods to find multiple roots of nonlinear equations.* Appl. Math. Comput., **219** (2013), 10987–11004.

13. E. N. Laguerre: *Ouvres de Laguerre, Vol. 1*, 1880, pp. 87–103.

14. I. Petković, Đ. Herceg: *Symbolic computation and computer graphics as tools for developing and studying new root-finding methods.* Appl. Math. Comput., **295** (2017), 95–113.

15. E. Bodewig: *Sur la méthode Laguerre pour l'approximation des racines de certaineséquations algébriques et sur la critique d'Hermite.* Indag. Math., **8** (1946), 570–580.

16. E. Hansen, M. Patrick: *A family of root finding methods.* Numer. Math., **27** (1977), 257–269.

17. B. Kalantari: *Polynomial Root-Finding and Polynomiography.* World Scientific, New Jersey, 2009.

18. M. Davies, B. Dawson: *On the global convergence of Halley's iteration formula.* Numer. Math., **24** (1975), 133–135.

19. G. Alefeld: *On the convergence of Halley's method.* Amer. Math. Monthly, **88** (1981), 530–536.

20. W. Gander: *On Halley's iteration method.* Amer. Math. Monthly, **92** (1985), 131–134.

21. A. Melman: *Geometry and convergence of Euler's and Halley's methods.* SIAM Rev., **39** (1997), 728–735.

22. L. O. Jay: *A note on Q-order of convergence.* BIT, **41** (2001), 422–429.

23. B. Neta, M. Scott, C. Chun: *Basin attractors for various methods for multiple roots.* Appl. Math. Comput., **218** (2012), 5043–5066.

24. F. Dubeau, C. Gnang: *On the Chebyshev-Halley family of iteration functions. Int. J. Pure Appl. Math.*, **85** (2013), 1051–1059.

25. E. R. Vrscay, W. J. Gilbert: *Extraneous fixed points, basin boundaries and chaotic dynamics for Schöder and König rational functions.* Numer. Math., **52** (1988), 1–16.

26. B. D. Stewart: *Attractor Basins of Various Root-finding Methods.* M.S. thesis, Naval Postgraduate School, Department of Applied Mathematics, Monterey, CA, June 2001.

27. J. L. Varona: *Graphic and numerical comparison between iterative methods.*, Math. Intelligencer, **24** (2002), 37–46.

28. B. Neta, C. Chun: *Basins of attraction for several optimal fourth order methods for multiple roots.* Math. Comput. Simul., **103** (2014), 39–59.

29. C. Chun, B. Neta: *Basins of attraction for Zhou-Chen-Song fourth order family of methods for multiple roots.* Math. Comput. Simul., **109** (2015), 74–91.

30. C. Chun, B. Neta: *Basins of attraction for several third order methods to find multiple roots of nonlinear equations.* Appl. Math. Comput., **268** (2015), 129–137.

31. A. M. Ostrowski: *Solution of Equations in Euclidean and Banach space.* Academic Press, New York, 1973.

32. B. Jovanović: *A method for obtaining iterative formulas of higher order.* Mat. Vesnik, **9** (1972), 365–369.

33. I. Petković, B. Neta: *On an application of symbolic computation and computer graphics to root-finders: The case of multiple roots of unknown multiplicity.* J. Comput. Appl. Math., **308** (2016), 215–230.

34. G. V. Milovanović: *A method to accelerated iterative processes in Banach space.* Univ. Beograd. Publ. Elektrotehn. Fak. Ser. Mat. Fiz., No **461**–No **479** (1974), 67–71.

35. C. Gnang, F. Dubeau: *On the rediscovery of Halley's iterative method for computing the zero of an analytic function.* J. Comput. Appl. Math., **335** (2018), 129–141.

36. I. Kiss: *Über die Verallgemeinerung des Newtonschen Näherungsver-fahrens.* Z. Angew. Math. Mech., **34** (1954), 68–69.

37. M. S. Petković: *Point Estimation of Root Finding Methods.* Springer, Berlin-Heidelberg, 2008.

Miodrag S. Petković
Faculty of Electronic Engineering,
University of Niš,
18000 Niš
Serbia
E-mail: *miodragpetkovic@gmail.com*

(Received 11.01.2019)
(Revised 14.05.2019)

Ljiljana D. Petković
Faculty of Mechanical Engineering,
University of Niš,
18000 Niš
Serbia
E-mail: *ljiljana@masfak.ni.ac.rs*

Beny Neta
Naval Postgraduate School
Department of Applied Mathematics,
Monterey, CA 93943
United States
E-mail: *bneta@nps.edu*