Filomat 33:4 (2019), 1019–1027
https://doi.org/10.2298/FIL1904019P

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Computational Geometry as a Tool for Studying Root-Finding Methods

Ivan Petković[a], Lidija Z. Rančić[a]

[a]*Faculty of Electronic Engineering, University of Niš, Serbia*

Abstract. We present an efficient method from Computational geometry, a branch of computer science devoted to the study of algorithms, for mathematical visualization of a third order root solver. For many decades the quality of iterative methods for solving nonlinear equations were analyzed only by using numerical experiments. The disadvantage of this approach is the inconvenient fact that convergence behavior strictly depends on the choice of initial approximations and the structure of functions whose zeros are sought, which often makes the convergence analysis very hard and incomplete. For this reason in this paper we apply dynamic study of iterative processes relied on basins of attraction, a new and powerful methodology developed at the beginning of the 21th century. This approach provides graphic visualization of the behavior of convergent sequences and, consequently, offers considerably better insight into the quality of applied root solvers, especially into the domain of convergence. For demonstration, we present dynamic study of one parameter family of Halley's type introduced in the first part of the paper. Characteristics of this family are discussed by basins of attractions for various values of the involved parameter. Special attention is devoted to clusters of polynomial roots, one of the most difficult problems in the topic. The analysis of the methods and presentation of basins of attractions are performed by the computer algebra system *Mathematica*.

1. Introduction

The aim of this paper is the dynamic study of a third order iterative methods for solving nonlinear equations by using methods of Computational geometry. The construction and convergence analysis of iterative root-solvers are among the most important tasks for solving many real life problems in many disciplines of engineering, computer science, physics, biology, chemistry, banking, business, digital signal processing, control theory, insurance, social science, as well as many other fields of human activities, see [1, Sec. 5.1]. Modern analysis of many numerical methods has been extremely improved by the end of twenty century due to the combination of numerical analysis and computing science, such as symbolic computation and computer graphics.

Today, dynamic study of root-finding methods is an indispensable composite part of modern analysis of the quality of these methods. It turns out that the dynamic study is a powerful tool in choosing optimal parameter that appears in some one-parameter iterative methods for solving equations, which is the main goal of this paper. In Section 3 we apply this methodology to a one-parameter third order method, presented

2010 *Mathematics Subject Classification*. Primary 65H05; Secondary 65Y20, 65D18, 68W30, 33F05
Keywords. Computational geometry, parametric iterative methods, graphic visualization, dynamic study, basin of attraction.
Received: 02 July 2018; Accepted: dd 22 July 2018
Communicated by Ljubica S. Velimirović
Research supported by Serbian Ministry of Education and Science

Corresponding author: Lidija Rančić
Email addresses: ivan.petkovic@elfak.ni.ac.rs (Ivan Petković), lidija.rancic@elfak.ni.ac.rs (Lidija Z. Rančić)

in Section 2, in order to find optimal parameter. More precisely, we search for the range of the parameter that gives the best convergence properties.

2. A family of third order one parameter root-finding methods

Let f be a differentiable function with a real or complex simple zero α, and let

$$u = u(x) = \frac{f(x)}{f'(x)}, \;\; A_m = A_m(x) = \frac{f^{(m)}(x)}{m!f'(x)} \;\; (m = 2, 3, \ldots).$$

Newton's iterative method for finding approximations x_k of α is defined by the iterative formula

$$N(x_k) := x_{k+1} = x_k - u(x_k), \quad (k = 0, 1, \ldots). \tag{1}$$

If the initial approximation x_0 is reasonably close to the zero α, the iterative method (1) converges to α quadratically.

In the sequel, we will omit the iteration index k for simplicity and write x and $\hat{x}$ instead of x_k and x_{k+1}. Let us consider the following modification of Newton's method (1)

$$H(x) = x - u(x) \cdot \frac{1 + pu(x)}{1 + (p - A_2(x))u(x)}, \tag{2}$$

presented in [3], where p is a real or complex constant. Starting from a reasonably close initial approximation x_0 to the zero α of f, the following family of iterative methods is obtained:

$$H(x_k; p) := x_{k+1} = x_k - \frac{u(x_k)\Big(1 + pu(x_k)\Big)}{1 + \Big(p - A_2(x_k)\Big)u(x_k)} \quad (k = 0, 1, 2, \ldots). \tag{3}$$

In practice, we choose real value for p since a number of numerical examples shows that handling with complex p has not any advantage.

The order of convergence of the family of iterative methods (3) is third, which can be proved by the following theorem from Traub's book [2, Theorem 2.2].

Theorem 1 *Let ϕ be iteration function such that $\phi^{(r)}$ is continuous in a neigborhood of α. Then ϕ is of order r if and only if*

$$\phi(\alpha) = \alpha; \quad \phi^{(j)}(\alpha) = 0 \quad (j = 1, 2, \ldots, r - 1); \quad \phi^{(r)}(\alpha) \neq 0.$$

We have used symbolic computation in computational algebra system *Mathematica* to find the derivatives $H'(x)$, $H''(x)$ and $H'''(x)$ of the iteration function H. We list

$$H'(x) = \frac{u^2\Big(pu(2A_2p - 3A_3) + 3A_2(A_2 + p) - 3A_3\Big)}{(A_2u - pu - 1)^2} \tag{4}$$

and

$$\begin{aligned} H''(x) \;\; = \;\; & \frac{2u}{(A_2u - pu - 1)^3}\Big\{3{A_2}^3u(3pu + 4) + {A_2}^2\Big[u\Big(p(2pu + 3)^2 - 3A_3u\Big) - 3\Big] \\ & + A_2\Big(-u^2(30A_3p + 6A_4 + p^3) - 3pu^3(3A_3p + 2A_4) - 3u(6A_3 + p^2) - 3p\Big) \\ & + 3(pu + 1)\Big(3{A_3}^2u^2 - A_3\Big(pu(pu + 1) - 1\Big) + 2A_4u(pu + 1)\Big)\Big\}. \end{aligned} \tag{5}$$

The expression of $H'''(x)$ is very lengthy so that it is omitted. Setting $x = \alpha$ in (2), (4) and (5), we obtain

$$H(\alpha) = \alpha, \quad H'(\alpha) = 0, \quad H''(\alpha) = 0, \quad H'''(\alpha) = 6\Big(A_2(\alpha)^2 - A_3(\alpha) + A_2(\alpha)p\Big).$$

Therefore, according to Theorem 1, there follows that the iterative method (2) is of order three.

Taking different values od the parameter p in (3) numerous cubically convergent root solvers can be generated. In the special case setting $p = 0$ in (3), we obtain the well known Halley's method

$$H(x_k; 0) := x_{k+1} = x_k - \frac{u(x_k)}{1 - A_2(x_k)u(x_k)} = x_k - \frac{f(x_k)}{f'(x_k)} \cdot \frac{1}{1 - \dfrac{f(x_k)f''(x_k)}{2f'(x_k)^2}} \,. \tag{6}$$

Furthermore, the choice $p = A_2(x_k)$ in (3) gives the third order method

$$H(x_k; A_2) := x_{k+1} = x_k - u(x_k)\big(1 + A_2(x_k)u(x_k)\big) = x_k - \frac{f(x_k)}{f'(x_k)}\Big(1 + \frac{f(x_k)f''(x_k)}{2f'(x_k)^2}\Big), \tag{7}$$

known as Chebyshev's method.

3. Dynamic study of the third order family

As said above, the main subject of this paper is to find optimal parameter p appearing in the family (3). It is not easy to give a precise explanation what "optimal parameter" in some iterative methods means since it depends on the user's targets. Sometimes, the user want to obtain 1) as great as possible accuracy in minimal number of iterative steps. In other situations the basic aim is 2) to provide guaranteed convergence, which is directly connected to the problem of choice of very good initial approximations. Finally, one of the best qualities of an iterative methods is 3) as large as possible domain of convergence in which the applied method globally converges to all zeros belonging to the domain in acceptable number of iterations consuming as small as possible CPU time.

The request 1) is tightly connected to the request 2); the best initial approximations provide the most accurate approximations to the zeros. Both problems were considered in details in the dissertation [4], the paper [5] and references cited there. For this reason we will not discuss the tasks 1) and 2). However, having in mind specific form of the family of iterative methods (3) which depends of the parameter p, we have performed a number of numerical examples for various values of p by applying the family (3) to algebraic polynomials as well as transcendental functions. According to results of numerical experiments we have drawn the following conclusions:

- We have not found the value of p which gives (at least approximately) the powerful method belonging to the family (3).
- The accuracy of produced approximations strongly depends on the structure of functions to be solved and the quality of initial approximations (the extent of closeness to the sought zeros).
- From (3) we observe that for a very large p the factor $(1 + pu)/(1 + (p - A_2)u)$ (which multiples u) tends to 1 producing $H(x) \approx x - u(x)$ (Newton's method), which means that the choice of large values of p should be avoided since the order of convergence decreases and tends to 2.
- A vast number of numerical examples has shown that the best results (for fixed initial approximations) are obtained for the parameter p belonging to the interval $[-a, a]$ for (empirically found) a (> 0) close to 1.

In overall, the methodology based only on numerical experiments do not give acceptable estimation of the *quantity* of the applied method.

From the above conclusions, which coincide with the results given in several existing papers published during the last two decades, numerical analysts have recently redirected their attention to the task 3), see, e.g., [6], [7], [8], [9], [10]. A new methodology for estimating the quality and the rank of root solvers, often called *dynamic study*, is based on the concept of *basins of attraction*. This approach provides not only graphical visualization of convergence behavior, but also gives a precise insight to the consuming CPU time for each basin, average number of iterations necessary to satisfy the termination criterion and the percentage of convergent points from the basin. The last characteristic points to the quality of the tested method regarding the size of domain. Altogether, dynamic study of iterative methods led to a better understanding of iterative procedures.

First we give the definition of the basin of attraction.

Definition 1. *Let $S \subseteq \mathbb{C}$ be a complex domain and let f be a given sufficiently many times differentiable function in S having simple zeros $\alpha_1, \alpha_2, \dots, \alpha_m \in S$. For a given root-finding iteration defined by $x_{k+1} = g(x_k)$, the basin of attraction for the zero α_i is the set (or the union of sets)*

$$\mathcal{B}_{f,g}(\alpha_i) = \{\xi \in S \,|\, \textit{the iteration } x_{k+1} = g(x_k) \textit{ with } x_0 = \xi \textit{ converges to } \alpha_i\}.$$

It is preferable that

(i) each basin of attraction consumes as small as possible CPU time counting all iterations and all starting points from S;

(ii) the boundaries of basins of attraction, related to each zero of the test function, are straight lines;

(iii) these basins, together with their boundaries, have fewer number of blobs, fractals and divergent points;

(iv) average number of iterations considering all points from the complex domain S is small as possible.

Convergence analysis of root-finding methods based on the dynamic study has appeared for the first time in the papers by Vrscay and Gilbert [6], Stewart [7] and Varona [8]. In this section we give the dynamic study of the iterative method (3) for the functions f_1, f_2 and f_3, given in Table 1, and the values of the parameter p equals to −2, −1, 0 (Halley's method (6)), 1, 2 and $p = A_2$ (Chebyshev's method (7)). As well known, most root-finding methods work well for polynomials whose roots are not close to each others. To avoid this easy task, in our experiments we chose rather challenging task and worked with test polynomials containing clusters of zeros (see small circles on Figures 1–3) since they are hard to solve.

$f(x)$	all zeros
$f_1(x) = x^5 - 0.00032$	$0.2\big(\cos(2k\pi/5) + i\,\sin(2k\pi/5)\big),\ k = 0,1,2,3,4$
$f_2(x) = (x^4 - 0.001x)(x^2 + 2x + 1.01)$	$0,\ -1 \pm 0.1i,\ 0.1\big(\cos(2k\pi/3) + i\,\sin(2k\pi/3)\big),\ k = 0,1,2$
$f_3(x) = (x^2 + 0.09)(x^3 - 0.01x)$	$0,\ \pm 0.3i,\ \pm 0.1$

Table 1: Tested functions for $f_1 - f_3$

Basins of attraction are plotted using the following rules:

- Each basin is colored by a different color;
- Each basin is divided into several parts, each of which is shaded darker (lighter) as the number of iterations rises (decreases).
- Starting points that lead to the divergence are colored black.
- We set the maximum of 40 iterations as a limit for every initial point; in the case that the number of iterations exceeds 40, we proclaim the considered initial point as divergent and paint it black.

In our dynamic study we tested the methods (3) on the 360 000 equally spaced points of the square

$$S = \{(x, y) \,|\, -3 \le x \le 3,\ -3 \le y \le 3\}$$

centered at the origin. The termination criterion was given by the inequality $|x_k - \alpha| < 10^{-6}$.

The obtained basins of attraction for the functions $f_1 - f_3$ are given in Figures 1–3 where small circles represents the polynomial roots. As mentioned above, an excellent property of the dynamic study is the ability to provide very useful data given in Table 2: the total CPU time for all 360 000 points, average number of iterations (counting all points of the square S) and the number of divergent points (assuming the limit of 40 iterations) for each method.

The dynamic study was realized using the computer algebra system *Mathematica* and the program presented as Appendix in the paper [11]. During this process the program has measured absolute values of CPU times $t^{(r)}(p)$ (expressed in seconds) for a fixed function f_r ($r \in \{1, 2, 3\}$). However, these data depend on the applied computer algebra system (Mathematica, Maple, Sage, Maxima and others) and the characteristics of the used digital computer such as its configuration, graphical features, a power of the embedded micro-processor, etc. To eliminate the influence of these factors, we have taken relative values of CPU times; more precisely, we calculated *normalized CPU times* $A_r(p)$ in reference to a method that requires minimal CPU time for the considered test function, that is, $A_r(p) = \frac{\mathrm{CPU}_r(p)}{\min\ \mathrm{CPU}_r(p)}$ $(r = 1, 2, 3)$, see Table 2.

$f_1(x) = x^5 - 0.00032$						
	$p = -2$	$p = -1$	$p = 0$	$p = 1$	$p = 2$	Cheb
A	1.76	1	1	1.03	1.83	1.21
B	9.68	8.98	8.94	9.22	9.75	12.62
C	20.02	0.04	0	0.07	20.14	2.92
$f_2(x) = (x^4 - 0.001x)(x^2 + 2x + 1.01)$						
A	1.74	1.03	1.02	1	1.05	1.05
B	10.06	9.50	9.41	9.33	10.06	11.98
C	20.7	0.009	0	0.009	8.88	0.12
$f_3(x) = (x^2 + 0.09)(x^3 - 0.01x)$						
A	1.71	1.01	1.01	1.02	1.73	1
B	14.69	9.42	9.49	9.42	9.96	11.81
C	23	0.007	0	0.007	20.4	0.07
average values						
A*	1.72	1.003	1	1.007	1.52	1.08
B	11.48	9.30	9.28	9.32	9.92	12.44
C	21.24	0.019	0	0.029	16.47	1.04

most preferable methods

Table 2: Iteration data for Examples 1–3

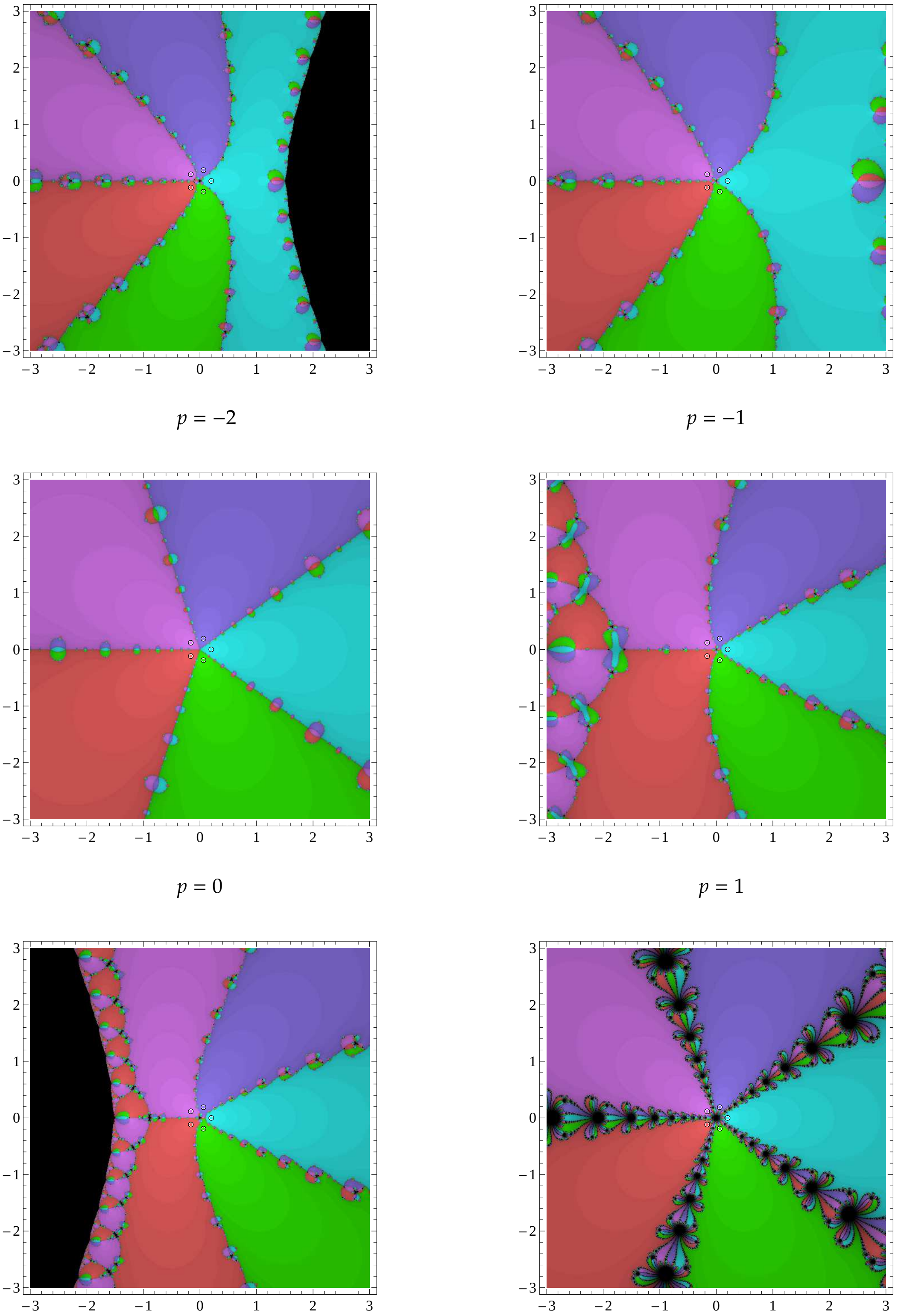
A – CPU time normalized related to CPU$_{min}$;

B – Average number of iterations per point;

C – Percentage of divergent ("black") points;

)* – Subsequent normalization of normalized average values of CPU time

Remark 1. Values of CPU time can vary up to ±4% depending on the activities of interior tasks of employed computer.



$p = -2$

$p = -1$

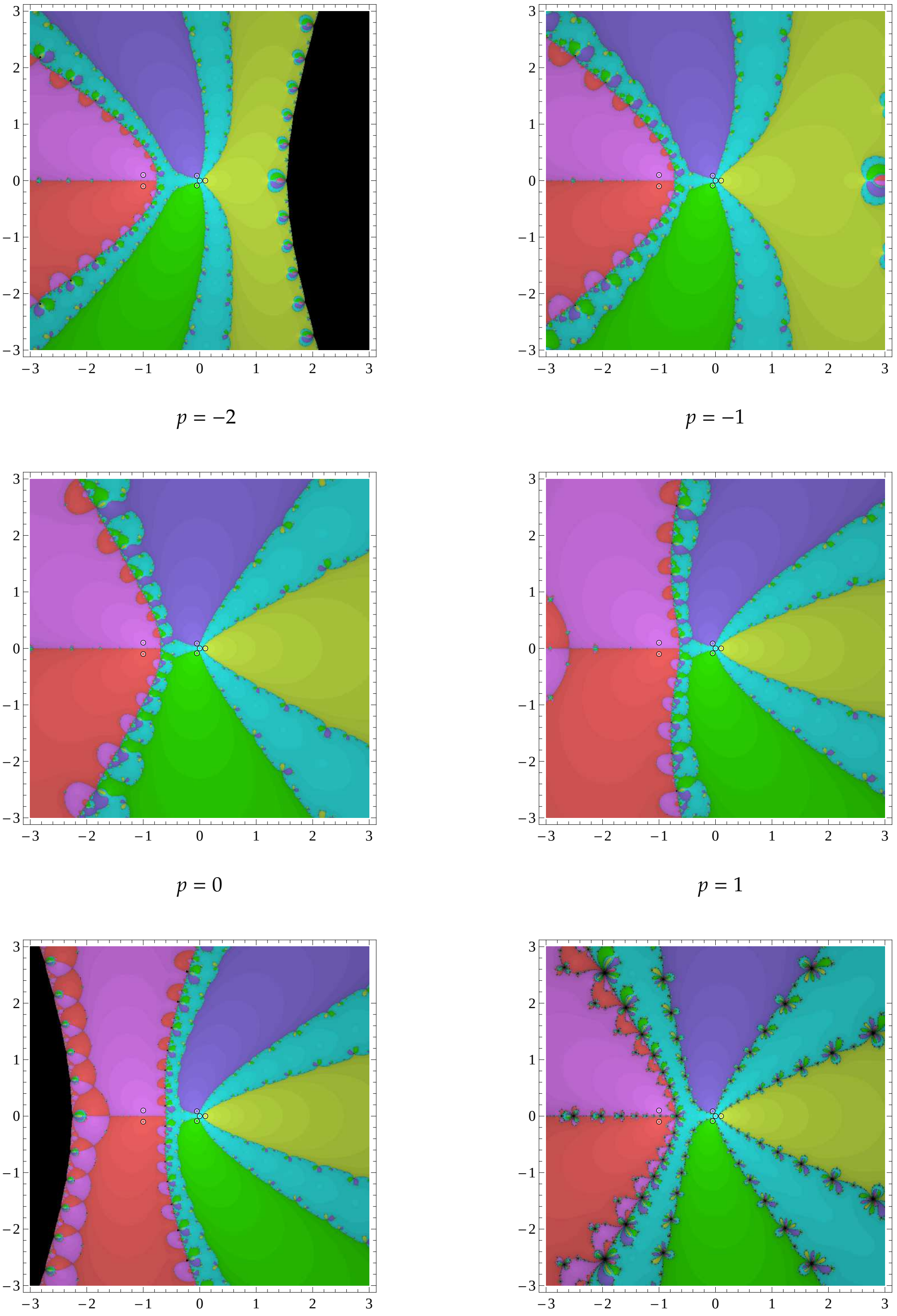
$p = 0$

$p = 1$

$p = 2$

Chebyshev's method (7)

Figure 1: $f_1(z) = z^5 - 0.00032$



$p = -2$

$p = -1$

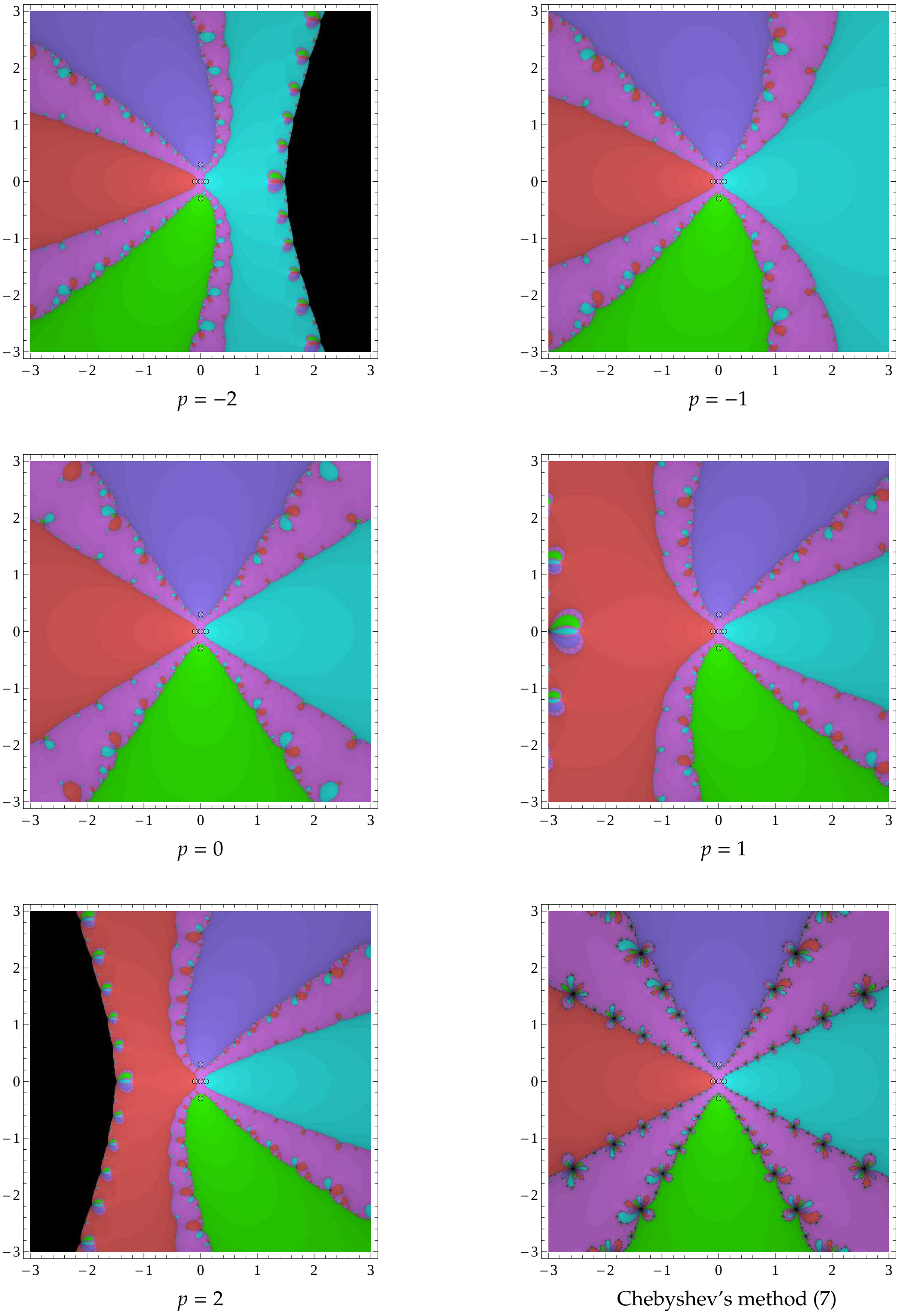
$p = 0$

$p = 1$

$p = 2$

Chebyshev's method (7)

Figure 2: $f_2(z) = (z^4 - 0.0001z)(z^2 + 2z + 1.01)$



$p = -2$

$p = -1$

$p = 0$

$p = 1$

$p = 2$

Chebyshev's method (7)

Figure 3: $f_3(z) = (z^2 + 0.09)(z^3 - 0.01z)$

Comments on the basins of attraction: A careful analysis of basins of attraction displayed on Figures 1–3 and associated data given in Table 2, led to the following conclusions:

(1) For all three test functions Halley's method $(6)_{p=0}$ is the only method without divergent points. This means that it possesses the best convergence properties in regard to the domain of convergence, although only slightly better than the methods obtained for $p = -1$ and $p = 1$.

(2) The attraction basins and their boundaries of Halley's method $(6)_{p=0}$ have fewest numbers of fractals and blobs. These preferable characteristics also appear for $p = -1$ and $p = 1$.

(3) Halley's method (6) consumes the smallest CPU time and converges in the smallest number of iterations (see Table 2, average values). Both of these values are negligibly less than those obtained for $p = -1$ and $p = 1$.

(4) Conclusions similar to (1)–(3) also valid for the methods constructed for any parameter p from the interval $(-1, 1)$ (not listed here to save the space).

(5) The quality of Chebyshev's method (7) is lower compared to the methods constructed for $p \in [-1, 1]$ considering all three features A, B and C (Table 2). However, the method (7) is better than the methods obtained for $p = -2$ and $p = 2$.

Altogether, according to the dynamic study of the methods from the family (3), we conclude that Halley's method $(6)_{p=0}$ shows the best convergence behavior. However, taking into account negligible differences among the entries displayed in Table 2 for $p \in [-1, 1]$, from a practical point of view it is more correct to say that the best convergence characteristics are shared among methods from the family (3) obtained for the parameter p belonging to the interval $(-1, 1)$.

Acknowledgements. This work was supported by the Serbian Ministry of Education and Science under Grant 174022.

References

[1] M.S. Petković, B. Neta, L.D. Petković, J. Džunić, Multipoint Methods for Solving Nonlinear Equations, Elsevier/Academic Press, Amsterdam, 2013.

[2] J.F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, New Jersey, 1964.

[3] L. D. Petković, M. S. Petković, A note on a new cubically convergent one-parameter root solver, arXiv:1706.00403v1 [math.NA] 26 May 2017.

[4] M. Farmer, Computing the Zeros of Polynomials using the Divide and Conquer Approach (Ph.D. thesis), Department of Computer Science and Information Systems, Birkbeck, University of London, 2013.

[5] M. Farmer, G. Loizou, S. Maybank, Iteration functions re-visited, J. Comp. Appl. Math. 311 (2017) 484–496.

[6] 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.

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

[8] J.L. Varona, Graphic and numerical comparison between iterative methods, Math. Intelligencer 24 (2002) 37–46.

[9] B. Kalantari, Polynomial Root-Finding and Polynomiography, World Scientific, New Jersey, 2009.

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

[11] 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.