Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2015, Article ID 797594, 19 pages
http://dx.doi.org/10.1155/2015/797594

Research Article

Polynomiography Based on the Nonstandard Newton-Like Root Finding Methods

Krzysztof Gdawiec, Wiesław Kotarski, and Agnieszka Lisowska

Institute of Computer Science, University of Silesia, Będzińska 39, 41-200 Sosnowiec, Poland

Correspondence should be addressed to Krzysztof Gdawiec; kgdawiec@ux2.math.us.edu.pl

Received 17 December 2014; Accepted 5 February 2015

Academic Editor: Naseer Shahzad

A survey of some modifications based on the classic Newton's and the higher order Newton-like root finding methods for complex polynomials is presented. Instead of the standard Picard's iteration several different iteration processes, described in the literature, which we call nonstandard ones, are used. Kalantari's visualizations of root finding process are interesting from at least three points of view: scientific, educational, and artistic. By combining different kinds of iterations, different convergence tests, and different colouring we obtain a great variety of polynomiographs. We also check experimentally that using complex parameters instead of real ones in multiparameter iterations do not destabilize the iteration process. Moreover, we obtain nice looking polynomiographs that are interesting from the artistic point of view. Real parts of the parameters alter symmetry, whereas imaginary ones cause asymmetric twisting of polynomiographs.

1. Introduction

Polynomial root-finding has played a key role in the history of mathematics. It is one of the oldest and most deeply studied mathematical problems. In 2000 BC Babylonians solved quadratic equation (quadratics). Seventeen centuries later Euclid solved quadratics with geometrical construction. In 1539 Cardan gave complete solution to cubics. In 1699 Newton introduced numerical iteration for root-finding. About seventy years later Lagrange showed that polynomial of degree 5 or higher cannot be solved by the methods used for quadratics, cubics, and quartics. In 1799 Gauss proved the Fundamental Theorem of Algebra. 27 years later Abel proved the impossibility of generally solving equations of degree higher than 4. General root-finding method has to be iterative and can only be done approximately. Cayley in 1879 observed strange and unpredictable chaotic behaviour of the roots approximation process while applying Newton's method to the equation $z^3 - 1 = 0$ in the complex plane. The solution of Caley's problem was found in 1919 by Julia. Julia sets became an inspiration for the great discoveries in 1970s, the Mandelbrot set and fractals [1]. The last interesting contribution to the polynomials root finding history was made by Kalantari [2], who introduced the polynomiography. It defines the visualization process of the approximation of the roots of complex polynomials, using fractal and nonfractal images created via the mathematical convergence properties of iteration functions. An individual image is called a polynomiograph. Polynomiography combines both art and science aspects. As a method which generates nice looking graphics, it was patented by Kalantari in USA in 2005 [3].

It is known that any complex polynomial p of degree n having n roots, according to the Fundamental Theorem of Algebra, can be uniquely defined by its coefficients $\{a_n, a_{n-1}, \ldots, a_1, a_0\}$:

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0 \tag{1}$$

or by its zeros (roots) $\{r_1, r_2, \ldots, r_{n-1}, r_n\}$:

$$p(z) = (z - r_1)(z - r_2) \cdot \cdots \cdot (z - r_n). \tag{2}$$

Iterative roots finding process can be obviously applied to both representations of p. The polynomiographs are generated as the result of this process' visualization. The degree of the polynomial defines the number of basins of attraction (root's basin of attraction is an area of the complex plane in

which each point is convergent to the root using the root finding method). Localizations of the basins can be controlled by changing the roots positions on the complex plane manually.

Usually, polynomiographs are coloured based on the number of iterations needed to obtain the approximation of some polynomial root with a given accuracy and a chosen iteration method. The description of polynomiography, its theoretical background and artistic applications are described in [2, 4].

Fractals and polynomiographs are generated by iterations. Fractals are self-similar, have complicated and nonsmooth structure, and are not dependent on a resolution. Polynomiographs are different. Their shape can be controlled and designed in a more predictable way in opposition to fractals. Generally, fractals and polynomiographs belong to different classes of graphical objects.

Summing up, polynomiography can be treated as a visualization tool based on the root finding process. It has many possible applications in education, math, sciences, art, and design [2].

In [5] the authors used Mann and Ishikawa iterations instead of the standard Picard iteration to obtain some generalization of Kalantari's polynomiography and presented some polynomiographs for the cubic equation $z^3 - 1 = 0$, permutation, and double stochastic matrices. Latif et al. in [6], using the ideas from [5], have used the S-iteration in polynomiography. Earlier, the other types of iterations have been used in [7] for superfractals and in [8] for fractals generated by IFS. Julia sets and Mandelbrot sets [9] and the antifractals [10] have been also investigated using Noor iteration instead of the standard Picard iteration.

The paper is organised as follows. In Section 2 different kinds of iterations are presented. Section 3 presents the known root finding methods, starting from the known Newton's method up to the different generalizations of it. Section 4 treats different convergence tests used in iteration processes together with their modifications. In Section 5 the colouring methods of polynomiographs are introduced. Section 6 summarizes the theory of polynomiograph generation. As a result the full algorithm of polynomiograph generation is given. Section 7 presents many polynomiographs obtained experimentally as the result of the proposed algorithm. In Section 8 the time complexity of this algorithm is discussed. Section 9 concludes the paper and shows the future directions.

2. Iterations

Obviously, the equation of the form $f(x) = 0$ can be equivalently transformed into a fixed point problem $x = T(x)$, where T is some operator [11]. Then, by applying the approximate fixed point theorem one can get information on existence, or sometimes both on existence and uniqueness, of the fixed point that is the solution of this equation.

Let (X, d) be a complete metric space and $T : X \to X$ a self-map on X. The set $\{x^* \in X : T(x^*) = x^*\}$ is the set of all fixed points of T. Many iterative processes have been described for the approximation of fixed points in the ample literature [12–19]. We recall below some iteration processes known from the literature. Assume that each iteration process starts from any initial point $x_0 \in X$.

(i) The standard Picard iteration [20] introduced in 1890 is defined as

$$x_{n+1} = T(x_n), \quad n = 0, 1, 2, \ldots. \tag{3}$$

(ii) The Mann iteration [16] was defined in 1953 as

$$x_{n+1} = (1 - \alpha_n) x_n + \alpha_n T(x_n), \quad n = 0, 1, 2, \ldots, \tag{4}$$

where $\alpha_n \in (0, 1]$ for all $n \in \mathbb{N}$.

(iii) The Ishikawa iteration [13] was defined in 1974 as a two-step process:

$$\begin{aligned} x_{n+1} &= (1 - \alpha_n) x_n + \alpha_n T(y_n), \\ y_n &= (1 - \beta_n) x_n + \beta_n T(x_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{5}$$

where $\alpha_n \in (0, 1]$ and $\beta_n \in [0, 1]$ for all $n \in \mathbb{N}$.

(iv) The Noor iteration [17] was defined in 2000 as a three-step process:

$$\begin{aligned} x_{n+1} &= (1 - \alpha_n) x_n + \alpha_n T(y_n), \\ y_n &= (1 - \beta_n) x_n + \beta_n T(z_n), \\ z_n &= (1 - \gamma_n) x_n + \gamma_n T(x_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{6}$$

where $\alpha_n \in (0, 1]$ and $\beta_n, \gamma_n \in [0, 1]$ for all $n \in \mathbb{N}$.

(v) The Suantai iteration [19] was defined in 2005 as a three-step iteration process with five parameters:

$$\begin{aligned} x_{n+1} &= (1 - \alpha_n - \beta_n) x_n + \alpha_n T(y_n) + \beta_n T(z_n), \\ y_n &= (1 - a_n - b_n) x_n + a_n T(z_n) + b_n T(x_n), \\ z_n &= (1 - \gamma_n) x_n + \gamma_n T(x_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{7}$$

where $\alpha_n, \beta_n, \gamma_n, a_n, b_n \in [0, 1]$, $\alpha_n + \beta_n \in [0, 1]$, and $a_n + b_n \in [0, 1]$ for all $n \in \mathbb{N}$ and $\sum_{n=0}^{\infty} (\alpha_n + \beta_n) = \infty$.

(vi) In 2007 Agarwal et al. in [21] introduced the S-iteration:

$$\begin{aligned} x_{n+1} &= (1 - \alpha_n) T(x_n) + \alpha_n T(y_n), \\ y_n &= (1 - \beta_n) x_n + \beta_n T(x_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{8}$$

where $\alpha_n \in (0, 1]$ and $\beta_n \in [0, 1]$ for all $n \in \mathbb{N}$.

(vii) The SP iteration [18] was defined in 2011 as the following three-step process:

$$\begin{aligned} x_{n+1} &= (1 - \alpha_n) y_n + \alpha_n T(y_n), \\ y_n &= (1 - \beta_n) z_n + \beta_n T(z_n), \\ z_n &= (1 - \gamma_n) x_n + \gamma_n T(x_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{9}$$

where $\alpha_n \in (0, 1]$ and $\beta_n, \gamma_n \in [0, 1]$ for all $n \in \mathbb{N}$.

(viii) In 2012 Chugh et al. introduced the CR iteration in [22]:

$$\begin{aligned} x_{n+1} &= (1-\alpha_n)\, y_n + \alpha_n T(y_n), \\ y_n &= (1-\beta_n)\, T(x_n) + \beta_n T(z_n), \\ z_n &= (1-\gamma_n)\, x_n + \gamma_n T(x_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{10}$$

where $\alpha_n, \beta_n, \gamma_n \in [0, 1]$ for all $n \in \mathbb{N}$ and $\sum_{n=0}^{\infty} \alpha_n = \infty$.

(ix) In 2013 Khan iteration [15] was defined as the following process:

$$\begin{aligned} x_{n+1} &= T(y_n), \\ y_n &= (1-\alpha_n)\, x_n + \alpha_n T(x_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{11}$$

where $\alpha_n \in (0, 1]$ for all $n \in \mathbb{N}$.

(x) In 2013 Karakaya et al. defined very general three-step iteration process with five parameters in [14]:

$$\begin{aligned} x_{n+1} &= (1-\alpha_n-\beta_n)\, y_n + \alpha_n T(y_n) + \beta_n T(z_n), \\ y_n &= (1-a_n-b_n)\, z_n + a_n T(z_n) + b_n T(x_n), \\ z_n &= (1-\gamma_n)\, x_n + \gamma_n T(x_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{12}$$

where $\alpha_n, \beta_n, \gamma_n, a_n, b_n \in [0, 1]$, $\alpha_n + \beta_n \in [0, 1]$, and $a_n + b_n \in [0, 1]$ for all $n \in \mathbb{N}$ and $\sum_{n=0}^{\infty}(\alpha_n + \beta_n) = \infty$.

(xi) In 2014 Gürsoy and Karakaya introduced the Picard-S iteration in [23]:

$$\begin{aligned} x_{n+1} &= T(y_n), \\ y_n &= (1-\alpha_n)\, T(x_n) + \alpha_n T(z_n), \\ z_n &= (1-\beta_n)\, x_n + \beta_n T(x_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{13}$$

where $\alpha_n \in (0, 1]$ and $\beta_n \in [0, 1]$ for all $n \in \mathbb{N}$.

The standard Picard iteration is used in the Banach Fixed Point Theorem [12] to obtain existence of the fixed point x^* of the operator T. Fixed point approximation is found under additional assumptions on the space X that it has to be a Banach one and the mapping T has to be contractive. The Mann [16], Ishikawa [13], and other iterations [12, 14, 15, 17–19] allow the weakening of the assumptions on the mapping T and generally allow the approximation of fixed points. The dependencies among the presented types of iterations are shown in Figure 1.

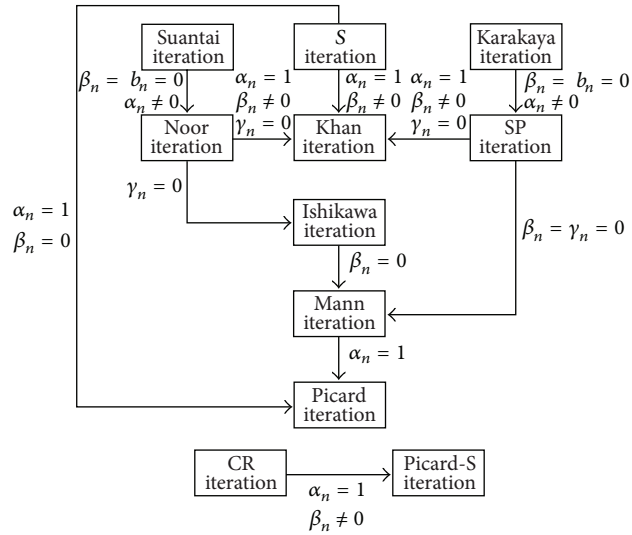


Figure 1: The diagram of iterations' dependencies.

Our further considerations will be conducted in the space $X = \mathbb{C}$ that is obviously a Banach one. We take $z_0 \in \mathbb{C}$ and $\alpha_n = \alpha$, $\beta_n = \beta$, $\gamma_n = \gamma$, $a_n = a$, and $b_n = b$ for all $n \in \mathbb{N}$ such that $\alpha \in (0, 1]$, $\beta, \gamma, a, b \in [0, 1]$, $\alpha + \beta \in (0, 1]$, and $a + b \in [0, 1]$. Naturally, if $\alpha + \beta \in (0, 1]$, then $\sum_{n=0}^{\infty}(\alpha_n + \beta_n) = \sum_{n=0}^{\infty}(\alpha + \beta) = \infty$.

3. Newton's Root Finding Method and Its Generalizations

At first, we recall the well-known Newton's method for finding roots of a complex polynomial. Then, following [2] some generalizations, which use higher order iterations described with the help of the Basic Family of Iterations and Euler-Schröder Family of Iterations, will be presented. At the end of this section a set of formulas for solving polynomial equation with a complex variable will be given. In those formulas the standard Picard iteration will be replaced by different types of nonstandard iterations defined in Section 2.

3.1. The Standard Newton's Method with Picard's Iteration. Let us denote any complex polynomial as p. The standard Newton's root finding procedure for p is given by the formula:

$$z_{n+1} = N(z_n), \quad n = 0, 1, 2, \ldots, \tag{14}$$

where

$$N(z) = z - \frac{p(z)}{p'(z)}, \tag{15}$$

and $z_0 \in \mathbb{C}$ is a starting point.

3.2. The Basic Family of Iterations. Let $\deg p \geq 2$. Define a sequence of functions $D_m : \mathbb{C} \to \mathbb{C}$ in the following way: $D_0(z) = 1$ and for $m > 0$ let

$$D_m(z) = \det \begin{bmatrix} p'(z) & \frac{p''(z)}{2!} & \cdots & \frac{p^{(m-1)}(z)}{(m-1)!} & \frac{p^{(m)}(z)}{m!} \\ p(z) & p'(z) & \ddots & \ddots & \frac{p^{(m-1)}(z)}{(m-1)!} \\ 0 & p(z) & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \frac{p''(z)}{2!} \\ 0 & 0 & \cdots & p(z) & p'(z) \end{bmatrix}. \tag{16}$$

The elements of the Basic Family of Iterations are then defined as

$$B_m(z) = z - p(z)\frac{D_{m-2}(z)}{D_{m-1}(z)}, \quad m = 2, 3, \ldots. \tag{17}$$

Let us see how the first three elements of the Basic Family look like:

$$\begin{aligned} B_2(z) &= z - \frac{p(z)}{p'(z)}, \\ B_3(z) &= z - \frac{2p'(z)p(z)}{2p'(z)^2 - p''(z)p(z)}, \\ B_4(z) &= z - \frac{6p'(z)^2 p(z) - 3p''(z)p(z)^2}{p'''(z)p(z)^2 + 6p'(z)^3 - 6p''(z)p'(z)p(z)}. \end{aligned} \tag{18}$$

One can easily see that B_2 is Newton's method, whereas B_3 is Halley's method.

By using functions D_m in [2], Kalantari defined the Parametric Basic Family:

$$B_{m,\lambda}(z) = z - \lambda p(z)\frac{D_{m-2}(z)}{D_{m-1}(z)}, \tag{19}$$

where $m = 2, 3, \ldots$ and $\lambda \in \mathbb{C}$. Let us note that for $\lambda = 1$ the Parametric Basic Family reduces to the Basic Family.

3.3. Euler-Schröder's Family of Iterations. Now, let us draw our attention to Euler-Schröder's Family of Iterations. The initial elements of this family have the following form:

$$\begin{aligned} E_2(z) &= z - \frac{p(z)}{p'(z)}, \\ E_3(z) &= E_2(z) + \left(\frac{p(z)}{p'(z)}\right)^2 \frac{p''(z)}{2p'(z)}, \\ E_4(z) &= E_3(z) - \left(\frac{p(z)}{p'(z)}\right)^3 \left(\frac{p'''(z)}{6p'(z)} - \frac{p''(z)}{2p'^2(z)}\right), \\ E_5(z) &= E_4(z) + \left(\frac{p(z)}{p'(z)}\right)^4 \\ &\quad \cdot \left(\frac{p^{IV}(z)}{4!p'(z)} - \frac{5p''(z)p'''(z)}{12p'^2(z)} + \frac{5p''^3(z)}{8p'^3(z)}\right). \end{aligned} \tag{20}$$

One can easily see that E_2 is Newton's method. The construction of the other elements of the family can be found in [2].

3.4. Root Finding Methods with Nonstandard Iterations. Let us denote by G one of the operators: N representing the standard Newton's method, B_i for $i = 2, 3, \ldots$ or $B_{m,\lambda}$ for $i = 2, 3, \ldots$, $\lambda \in \mathbb{C}$ or E_i for $i = 2, 3, \ldots$ representing elements of the Basic, Parametric Basic, and Euler-Schöder's Families of Iterations, respectively. And let us replace the standard Picard iteration by one of the nonstandard iterations described in Section 2. Then we get the following formulas for finding roots of complex polynomial p iteratively:

(i) The generalized Newton's method with the Mann iteration (4):

$$z_{n+1} = (1-\alpha)z_n + \alpha G(z_n), \quad n = 0, 1, 2, \ldots, \tag{21}$$

where $\alpha \in (0, 1]$.

(ii) The generalized Newton's method with the Ishikawa iteration (5):

$$\begin{aligned} z_{n+1} &= (1-\alpha)z_n + \alpha G(v_n), \\ v_n &= (1-\beta)z_n + \beta G(z_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{22}$$

where $\alpha \in (0, 1]$ and $\beta \in [0, 1]$.

(iii) The generalized Newton's method with the Noor iteration (6):

$$\begin{aligned} z_{n+1} &= (1-\alpha)z_n + \alpha G(v_n), \\ v_n &= (1-\beta)z_n + \beta G(w_n), \\ w_n &= (1-\gamma)z_n + \gamma G(z_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{23}$$

where $\alpha \in (0, 1]$ and $\beta, \gamma \in [0, 1]$.

(iv) The generalized Newton's method with the Suantai iteration (7):

$$\begin{aligned} z_{n+1} &= (1-\alpha-\beta)\, z_n + \alpha G(v_n) + \beta G(w_n), \\ v_n &= (1-a-b)\, z_n + a G(w_n) + b G(z_n), \\ w_n &= (1-\gamma)\, z_n + \gamma G(z_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{24}$$

where $\alpha \in (0, 1]$, $\beta, \gamma, a, b \in [0, 1]$ and $\alpha + \beta \in (0, 1]$, $a + b \in (0, 1]$.

(v) The generalized Newton's method with the S-iteration (8):

$$\begin{aligned} z_{n+1} &= (1-\alpha)\, G(z_n) + \alpha G(v_n), \\ v_n &= (1-\beta)\, z_n + \beta G(z_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{25}$$

where $\alpha \in (0, 1]$ and $\beta \in [0, 1]$.

(vi) The generalized Newton's method with the SP iteration (9):

$$\begin{aligned} z_{n+1} &= (1-\alpha)\, v_n + \alpha G(v_n), \\ v_n &= (1-\beta)\, w_n + \beta G(w_n), \\ w_n &= (1-\gamma)\, z_n + \gamma G(z_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{26}$$

where $\alpha \in (0, 1]$ and $\beta, \gamma \in [0, 1]$.

(vii) The generalized Newton's method with the CR iteration (10):

$$\begin{aligned} z_{n+1} &= (1-\alpha)\, v_n + \alpha G(v_n), \\ v_n &= (1-\beta)\, G(z_n) + \beta G(w_n), \\ w_n &= (1-\gamma)\, z_n + \gamma G(z_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{27}$$

where $\beta, \gamma \in [0, 1]$ and $\alpha \in (0, 1]$, because if $\alpha \neq 0$, then $\sum_{n=0}^{\infty} \alpha_n = \sum_{n=0}^{\infty} \alpha = \infty$.

(viii) The generalized Newton's method with the Khan iteration (11):

$$\begin{aligned} z_{n+1} &= G(v_n), \\ v_n &= (1-\alpha)\, z_n + \alpha G(z_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{28}$$

where $\alpha \in (0, 1]$.

(ix) The generalized Newton's method with the Karakaya iteration (12):

$$\begin{aligned} z_{n+1} &= (1-\alpha-\beta)\, v_n + \alpha G(v_n) + \beta G(w_n), \\ v_n &= (1-a-b)\, w_n + a G(w_n) + b G(z_n), \\ w_n &= (1-\gamma)\, z_n + \gamma G(z_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{29}$$

where $\alpha \in (0, 1]$, $\beta, \gamma, a, b \in [0, 1]$ and $\alpha + \beta \in (0, 1]$, $a + b \in (0, 1]$.

(x) The generalized Newton's method with the Picard-S iteration (13):

$$\begin{aligned} z_{n+1} &= G(v_n), \\ v_n &= (1-\alpha)\, G(z_n) + \alpha G(w_n), \\ w_n &= (1-\beta)\, z_n + \beta G(z_n), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{30}$$

where $\alpha \in (0, 1]$ and $\beta \in [0, 1]$.

The sequence $\{z_n\}_{n=0}^{\infty}$ (or orbit of the point z_0) either converges or does not to a root of p. If the sequence converges to a root z^* then we say that z_0 is attracted to z^*. The set of all starting points z_0 for which $\{z_n\}_{n=0}^{\infty}$ converges to z^* is called the basin of attraction of z^*. The boundaries among basins usually are fractals in nature.

All the above presented iteration processes are convergent to the roots of polynomial p. Only the speed and the character of the convergence are different and the basins of attraction to roots of p look different for different kinds of iterations used.

The application of nonstandard iterations perturbs the shape of polynomial basins and makes the polynomiographs look more "fractal." The aim of using more general iterations, instead of the Picard iteration, was not to improve the speed of convergence but to create images that are interesting from the aesthetic point of view.

4. Convergence Tests

In the numerical algorithms that are based on iterative processes we need a stop criterion for the process, that is, a test that tells us that the process has converged or it is very near to the solution. This type of test is called a convergence test. Usually, in the iterative process that use a feedback, like the root finding methods, the standard convergence test has the following form:

$$|z_{n+1} - z_n| < \varepsilon, \tag{31}$$

where z_{n+1}, z_n are two successive points in the iteration process and $\varepsilon > 0$ is a given accuracy.

In 1988 Pickover in [24] proposed a different convergence test for Halley's root finding method. By changing the standard convergence test (31) with

$$\left| |z_{n+1}|^2 - |z_n|^2 \right| < \varepsilon. \tag{32}$$

Pickover obtained new and diverse shapes of the polynomiographs. Later, Gdawiec in [25] introduced methods of creating new convergence tests, which we will briefly present in the rest of this section.

When we look at (31) we can note that the calculation of the modulus is equivalent to the computation of the distance (in the complex plane) between the two elements. So, one way of changing the test is the use of different metrics in $\mathbb{C}$. We know that the complex plane $\mathbb{C}$ is isometric with $\mathbb{R}^2$, where the isometry $\phi : \mathbb{C} \to \mathbb{R}^2$ is given by [26]

$$\phi(z) = (\Re(z), \Im(z)), \tag{33}$$

for every $z \in \mathbb{C}$, and where $\mathfrak{R}(z)$ and $\mathfrak{T}(z)$ denote the real and imaginary part of z, respectively. Using the isometry we can define metric $d : \mathbb{C} \times \mathbb{C} \to [0, +\infty)$ using metric $\rho : \mathbb{R}^2 \times \mathbb{R}^2 \to [0, +\infty)$ in the following way [26]:

$$d(z_1, z_2) = \rho(\phi(z_1), \phi(z_2)), \tag{34}$$

where $z_1, z_2 \in \mathbb{C}$. For instance, we can use some well-known metrics defined in $\mathbb{R}^2$ [26]:

(i) the taxicab metric

$$\rho((x_1, y_1), (x_2, y_2)) = |x_1 - x_2| + |y_1 - y_2|, \tag{35}$$

(ii) the supremum metric

$$\rho((x_1, y_1), (x_2, y_2)) = \max\{|x_1 - x_2|, |y_1 - y_2|\}, \tag{36}$$

(iii) the l_p metric

$$\rho((x_1, y_1), (x_2, y_2)) = \left[|x_1 - x_2|^p + |y_1 - y_2|^p\right]^{1/p}, \tag{37}$$

where $1 \le p \le +\infty$.

Moreover, we can use different facts about metric spaces to create new metrics. For instance, let (X, ρ) be a metric space; then [26]

(i) if $f : X \to X$ is injective, then

$$\eta(x, y) = \rho(f(x), f(y)) \tag{38}$$

is a metric on X;

(ii) if $f : X \to \mathbb{R}$ is a function, then

$$\eta(x, y) = \rho(x, y) + |f(x) - f(y)| \tag{39}$$

is a metric on X.

If we are interested in generating diverse patterns using polynomiography, we can take a function that does not fulfil some of the metric axioms, for example, l_p for $p \in (0, 1)$ does not fulfil the triangle inequality. We can also omit the assumption about the injectivity of f in (38). For instance, if we take the complex plane $\mathbb{C}$ with the modulus metric and $f(z) = |z|^2$, which of course is not injective, we obtain

$$\eta(z_1, z_2) = \left||z_1|^2 - |z_2|^2\right|. \tag{40}$$

When we look at (40), then we can see that this is the function used by Pickover in (32).

Another way to modify the tests is to add some weights in the metric function. The weights could cause that the metric function will lose the properties of the metric. For instance, if we use (38) for creation of the test we can add weights $\xi_1, \xi_2 \in \mathbb{R}$ in the following way:

$$\eta(x, y) = \rho(\xi_1 f(x), \xi_2 f(y)). \tag{41}$$

If $\xi_1 \neq \xi_2$, then we lose the symmetry property of the metric.

All the tests discussed so far were based on a single metric function, but we can create tests using several terms that use metric functions or modified metric functions; for example,

$$\begin{aligned} &|\xi_1 \mathfrak{R}(z_{n+1} - z_n)| < \varepsilon_1 \vee |\xi_2 \mathfrak{T}(z_{n+1} - z_n)| < \varepsilon_2, \\ &|\xi_1 \mathfrak{R}(z_{n+1} - z_n)|^2 < \varepsilon_1 \wedge |\xi_2 \mathfrak{T}(z_{n+1} - z_n)|^2 < \varepsilon_2, \end{aligned} \tag{42}$$

where $\xi_1, \xi_2 \in \mathbb{R}$ and $\varepsilon_1, \varepsilon_2 > 0$.

In the Mandelbrot and Julia sets we use the escape criteria to stop the iterative process. In this criteria we check if the computed value is greater than the given threshold value $R > 0$ [27]. Based on the idea of escape criteria we can create different tests in polynomiography. The examples of this kind of tests are as follows:

$$\begin{aligned} &|z_{n+1} - z_n| + |\arg(z_{n+1}) - \arg(z_n)| > R, \\ &\left|\frac{1}{|z_{n+1}|^2} - \frac{1}{|z_n|^2}\right| + \left||z_{n+1}|^2 - |z_n|^2\right| > R, \\ &\xi_1 |\mathfrak{R}(z_{n+1} - z_n)| > R \wedge \xi_2 |\mathfrak{T}(z_{n+1} - z_n)| > R, \end{aligned} \tag{43}$$

where $\arg(z)$ is an argument of the complex number z, and $\xi_1, \xi_2 \in \mathbb{R}$.

5. Colouring Methods

After satisfying the convergence test in the iteration process of the root-finding method for a considered starting point we need to determine the colour for that point. The method of colour determination for a given point is called the colouring method. We briefly introduce three basic colouring methods [2] in this section.

In the first method we need to know all the roots $\{r_1, r_2, \ldots, r_n\}$ of the polynomial p. So, if we want to use the method, it is comfortable to use the polynomial representation by its roots (2), because we do not need to compute the roots. Each root r_i of the polynomial gets a distinct colour c_i. So, after the iteration process we take the obtained approximation of the root z_m and find the closest root of p using the modulus metric. Having the closest root we colour the starting point with the colour that corresponds to this root. In this way we obtain visualization of the polynomial basins of attraction introduced in Section 3.4.

Unlike the first method, the second colouring method does not need the information about the roots, so we can use any of the two polynomial representation methods (coefficients, roots). In this method we deal with a colour map, that is, the table of different colours. After the iteration process we take the number of iteration m at which the process has stopped and map it to an index of colour in the colour map. If the number of colours in the colour map is equal to the maximum number of iterations, then we have one-to-one correspondence between iterations and colours. In the other case we need to use some mapping. Most often the linear interpolation is used; that is, a mapping $L : \{0, 1, \ldots, M\} \to \{0, 1, \ldots, C - 1\}$, where M is the maximum number of

Input: $p \in \mathbb{C}[Z]$—polynomial, $A \subset \mathbb{C}$—area, M—maximum number of iterations, I_q—iteration method for some Generalized Newton Method, $q \in \mathbb{C}^N$—parameters of the iteration I_q.
Output: Polynomiograph for the area A.
(1) **for** $z_0 \in A$ **do**
(2) $i = 0$
(3) **while** $i \leq M$ **do**
(4) $z_{i+1} = I_q(z_i)$
(5) **if** $convergenceTest(z_i, z_{i+1}) = true$ **then**
(6) break
(7) $i = i + 1$
(8) determine the colour for z_0

Algorithm 1: Polynomiograph generation.

iterations and C is the number of colours in the colour map, of the following form:

$$L(m) = \left\lfloor (C-1)\frac{m}{M} \right\rfloor. \tag{44}$$

By using this method we are able to visualize the speed of convergence of the root-finding method. The use of specific colour maps often reveals a hidden unrepeatable beauty of the root finding visualization process.

In [28] Pickover used this method to create contour lines that helps to visually emphasize different regions of behaviour of the considered function (root-finding method for a given polynomial). For this purpose he used two colours (black, white) and a mapping function $P : \{0, 1, \ldots, M\} \to \{0, 1\}$ of the following form:

$$P(m) = m \bmod 2. \tag{45}$$

The last colouring method combines the features of the two previous ones. It visualizes, at the same time, basins of attractions and speed of convergence of the root-finding method. In this method, like in the first one, we need to know the roots $\{r_1, r_2, \ldots, r_n\}$ of the polynomial p and each root get a distinct colour $\{c_1, c_2, \ldots, c_n\}$. After stopping the iteration process we take the obtained approximation of the root z_m and we find the closest root of p using the modulus metric. Now, we set the colour of the starting point to the colour of the closest root and we use the iteration number m to set the shade of that colour. To have ease in operating on the colours they should be represented in the HSB (Hue, Saturation, Brightness) colour space. In this space hue represents the colour's type (the root colour) and saturation represents the shade of the colour. Moreover, we can use linear interpolation to map the iteration number to the saturation, what would be difficult if we have used the RGB (Red, Green, Blue) colour space. In this way, visualization shows the basins of attraction by using distinct colours and shows the speed of convergence represented by the shade of the colour.

6. Polynomiograph Generation

In the previous sections we introduced several parts of the polynomiograph generation method. Putting them together we obtain algorithm that is presented as a pseudocode in Algorithm 1.

The input for the algorithm consists of the polynomial p given by equation (1) or (2), the area of the complex plane A for which the polynomiograph is generated, and the maximum number of iterations which will be made for each point in A. The last input parameter is the iteration method I_q from Section 3.4 for a chosen root-finding method. The index q is a vector of parameters of the iteration method, that is, $q \in \mathbb{C}^N$, where N is the number of parameters of the iteration. For the Picard's method the iteration I is used instead of I_q. Moreover, some convergence test and a colouring method have to be fixed.

In Section 3.4 the parameters used in the iterations were real numbers, but when we look at Algorithm 1 we see that they are complex. According to the authors' knowledge the iterations with complex parameters have not been studied so far. The experiments carried out for this type of iterations show that the generated polynomiographs form interesting patterns, what will be shown in Section 7.2.

In this algorithm, for each point z_0 in the considered area A, this point is iterated by the method I_q. If the convergence test is satisfied, then it is assumed that the generated sequence converges to a root of p and the iteration is stopped. In the other case the algorithm goes to the next iteration. If the maximum number of iterations M is reached, it is assumed that the generated sequence does not converge to any root of p. At the end a colour is given to the considered point by using a fixed method of colouring.

7. Examples of Polynomiographs

In this section some examples of the polynomiographs obtained by using the methods described in the previous sections are presented. First, we show the use of different iterations with both the real and complex parameters. Next, the influence of the convergence tests on the polynomiograph's shape will be presented. The last example shows the use of different colour maps. In all examples we use the same colouring method—the second method from Section 5 with different colour maps.

7.1. Polynomiographs with Real-Valued Parameters of Iterations. Figure 2 presents examples of the polynomiographs with the use of the following parameters: $p_1(z) = z^7 + z^2 - 1$, $M = 15$, standard convergence test with $\varepsilon = 0.001$, $A = [-1.5, 1.5]^2$, the Newton's root finding method (B_2, E_2). The parameters used in the iterations were fixed as follows:

(b) Mann: $\alpha = 0.9$,

(c) Ishikawa: $\alpha = 0.35$, $\beta = 0.85$,

(d) Noor: $\alpha = 0.75$, $\beta = 0.39$, $\gamma = 0.07$,

(e) Suantai: $\alpha = 0.25$, $\beta = 0.75$, $\gamma = 0.01$, $a = 0.25$, $b = 0.75$,

(f) *S*: $\alpha = 0.45$, $\beta = 0.5$,

(g) SP: $\alpha = 0.45$, $\beta = 0.3$, $\gamma = 0.1$,

(h) CR: $\alpha = 0.1$, $\beta = 0.4$, $\gamma = 0.8$,

(i) Khan: $\alpha = 0.15$,

(j) Karakaya: $\alpha = 0.7$, $\beta = 0.2$, $\gamma = 0.85$, $a = 0.1$, $b = 0.7$,

(k) Picard-S: $\alpha = 0.9$, $\beta = 0.6$.

In Figure 2 one can see that different iteration processes produce unique polynomiographs that are different comparing to the polynomiograph obtained with the standard Picard iteration and to each other. In each polynomiograph one can find seven main areas with different shapes and with fractal boundaries. Moreover, looking at the colours and shapes of the polynomiographs one can see that the use of different iteration processes change the speed of convergence of the root finding method, for some points the convergence is faster and for some others it is slower. The speed depends on the iteration and the parameters used. All images, that are very decorative, have visible symmetries what is a consequence of placing the roots in a symmetrical way. The symmetry introduces the order which stress the static appearance of polynomiographs.

The second example (Figure 3) presents the polynomiographs generated with the use of the following parameters: $p_2(z) = z^4+4$, $M = 40$, the standard convergence test with $\varepsilon = 0.001$, $A = [-2, 2]^2$, E_3 root finding method. The parameters used in the iterations were fixed as follows:

(b) Mann: $\alpha = 0.5$,

(c) Ishikawa: $\alpha = 0.1$, $\beta = 0.1$,

(d) Noor: $\alpha = 0.35$, $\beta = 0.7$, $\gamma = 0.4$,

(e) Suantai: $\alpha = 0.01$, $\beta = 0.2$, $\gamma = 0.01$, $a = 0.3$, $b = 0.1$,

(f) *S*: $\alpha = 0.99$, $\beta = 0.01$,

(g) SP: $\alpha = 0.1$, $\beta = 0.99$, $\gamma = 0.01$,

(h) CR: $\alpha = 0.8$, $\beta = 0.45$, $\gamma = 0.8$,

(i) Khan: $\alpha = 0.75$,

(j) Karakaya: $\alpha = 0.9$, $\beta = 0.01$, $\gamma = 0.01$, $a = 0.9$, $b = 0.01$,

(k) Picard-S: $\alpha = 0.75$, $\beta = 0.99$.

In the polynomiographs for p_2 one can see that different iteration processes produce eight main areas with very subtle fractal boundaries. Moreover, one obtains very diverse and different patterns in comparison to the polynomiograph with the standard Picard iteration. We can also observe that the use of different iterations has impact on the speed of convergence. For instance, for the Karakaya or Picard-S iteration one can see that the red areas that occur for the Picard iteration have shrunk and the light blue areas changed the colour to navy blue. This means that in those areas the convergence is faster. The change of speed depends on the iteration used and the value of its parameters.

7.2. Polynomiographs with Complex-Valued Parameters of Iterations. Figure 4 presents the examples of polynomiographs with the same parameters as in Figure 2; that is, $p_1(z) = z^7 + z^2 - 1$, $M = 15$, the standard convergence test with $\varepsilon = 0.001$, $A = [-1.5, 1.5]^2$, the Newton's root finding method (B_2, E_2). The only change is the addition of imaginary part to the values of the iteration parameters:

(a) Mann: $\alpha = 0.9 + 0.3\mathbf{i}$,

(b) Ishikawa: $\alpha = 0.35$, $\beta = 0.85 + \mathbf{i}$,

(c) Noor: $\alpha = 0.75-0.26\mathbf{i}$, $\beta = 0.39-0.26\mathbf{i}$, $\gamma = 0.07-0.5\mathbf{i}$,

(d) Suantai: $\alpha = 0.25$, $\beta = 0.75+0.1\mathbf{i}$, $\gamma = 0.01$, $a = 0.25+\mathbf{i}$, $b = 0.75 + 0.3\mathbf{i}$,

(e) *S*: $\alpha = 0.45 + 0.5\mathbf{i}$, $\beta = 0.5$,

(f) SP: $\alpha = 0.45$, $\beta = 0.3 - 0.5\mathbf{i}$, $\gamma = 0.1 + 0.25\mathbf{i}$,

(g) CR: $\alpha = 0.1$, $\beta = 0.4 + 0.15\mathbf{i}$, $\gamma = 0.8 + 0.13\mathbf{i}$,

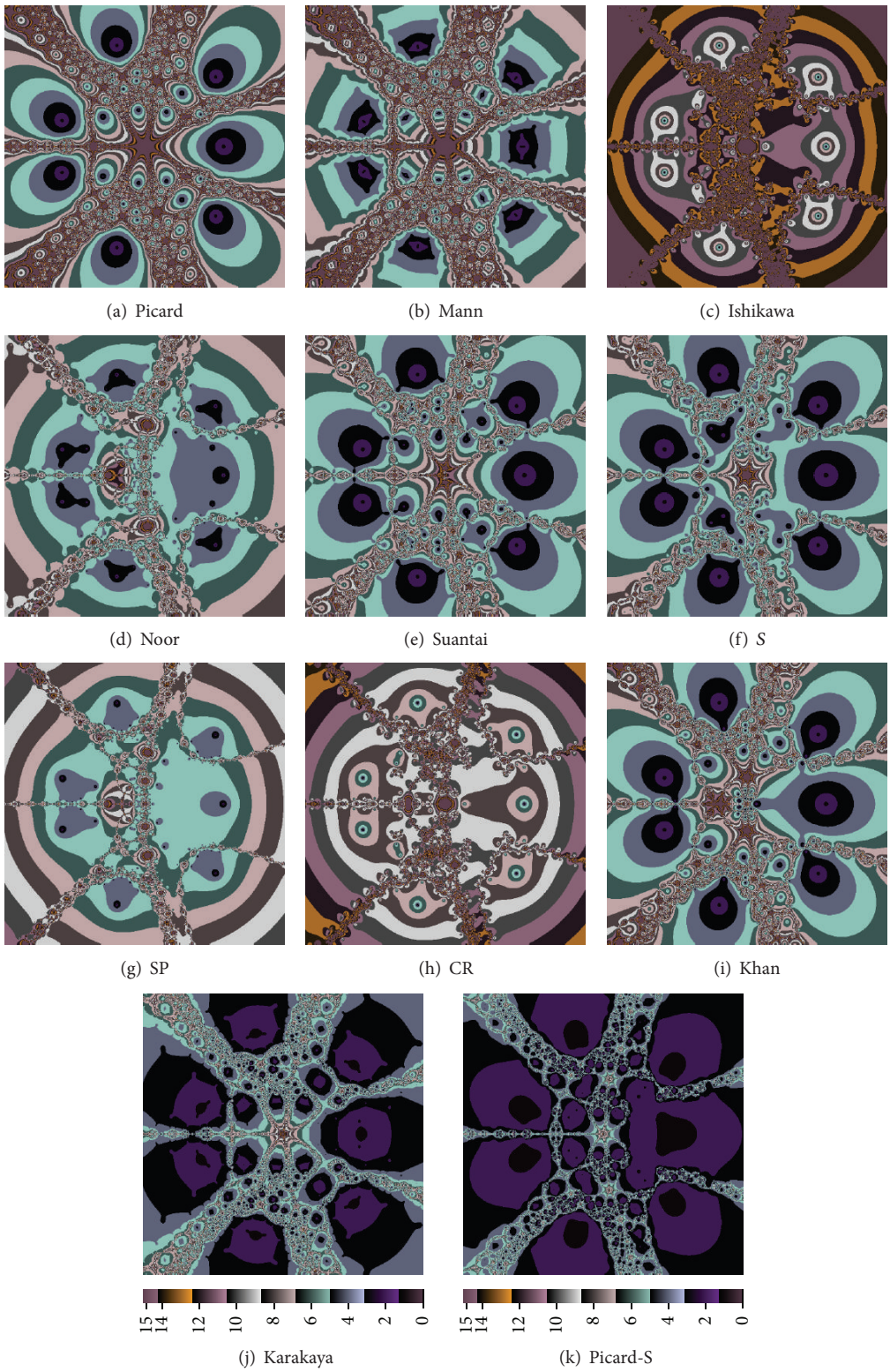
(h) Khan: $\alpha = 0.15 - \mathbf{i}$,

(i) Karakaya: $\alpha = 0.7$, $\beta = 0.2$, $\gamma = 0.85 - \mathbf{i}$, $a = 0.1 - \mathbf{i}$, $b = 0.7 - \mathbf{i}$,

(j) Picard-S: $\alpha = 0.9 + \mathbf{i}$, $\beta = 0.6 - 0.5\mathbf{i}$.

The use of a nonzero imaginary part of the parameters, generally, adds the clockwise or the anticlockwise rotations to the polynomiographs. The angle of rotation is dependent on the value and the sign of the imaginary part. In the case of the multiparameter iterations some of the parameters (their imaginary parts) have global and some have only local influence on the shape of the polynomiograph. The effect is easily seen in Figure 4 by comparison with Figure 2. Swirls and twists present in the polynomiographs from Figure 4 make those images look more dynamical and live in opposite to the polynomiographs from Figure 2. The imaginary parts of the parameters have also the influence on the speed of convergence, which can be easily seen, for instance, in Figure 4(i) by comparison with Figure 2(j).

The next example (Figure 5) presents polynomiographs generated with the use of the same parameters as in Figure 3, that is, $p_2(z) = z^4 + 4$, $M = 40$, the standard convergence test with $\varepsilon = 0.001$, $A = [-2, 2]^2$, E_3 root finding method. The only change is the addition of the imaginary parts to the values of iteration parameters:



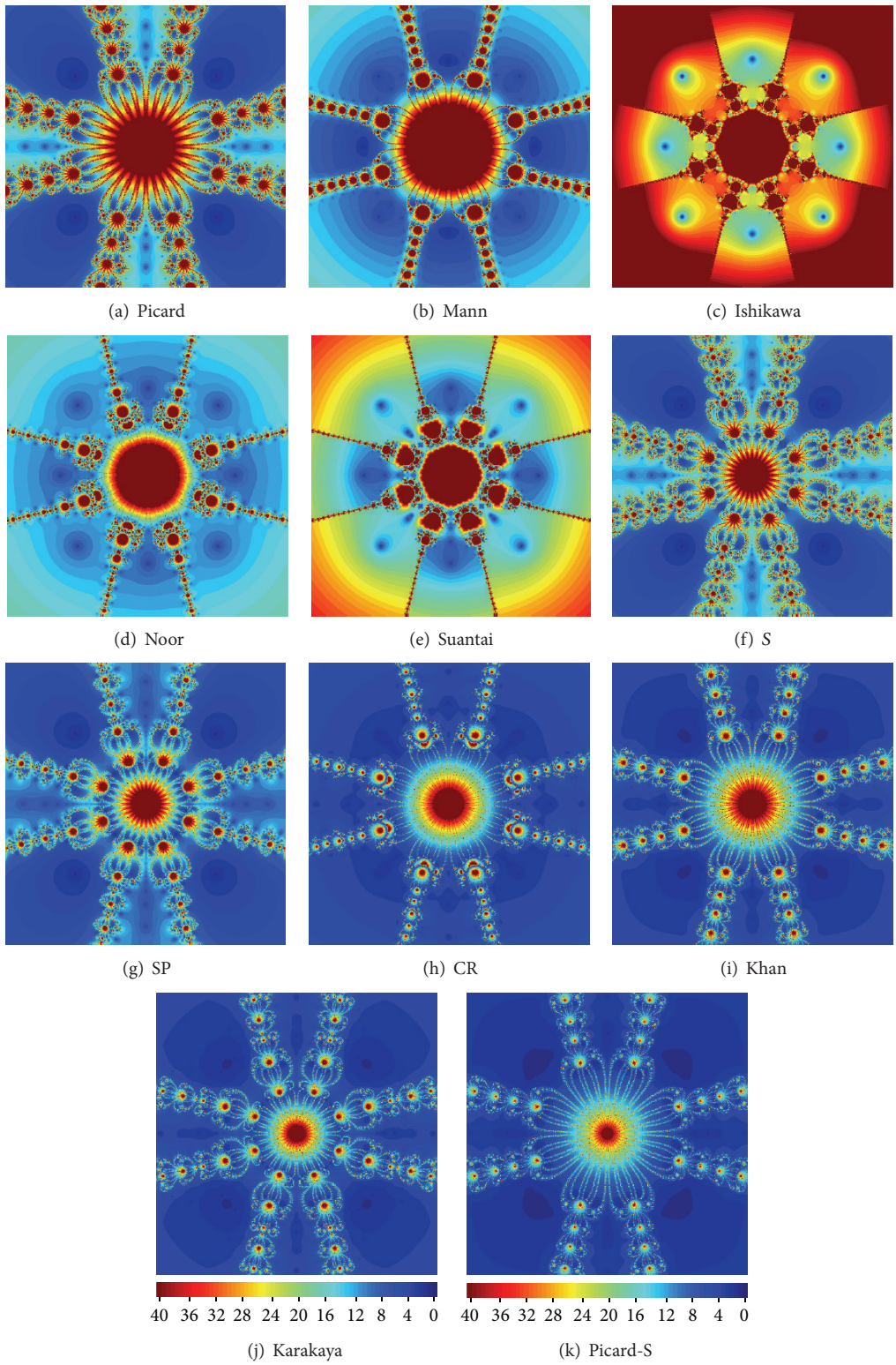
(a) Picard (b) Mann (c) Ishikawa

(d) Noor (e) Suantai (f) S

(g) SP (h) CR (i) Khan

(j) Karakaya (k) Picard-S

FIGURE 2: Examples of polynomiographs for p_1 with real-valued parameters of iterations.



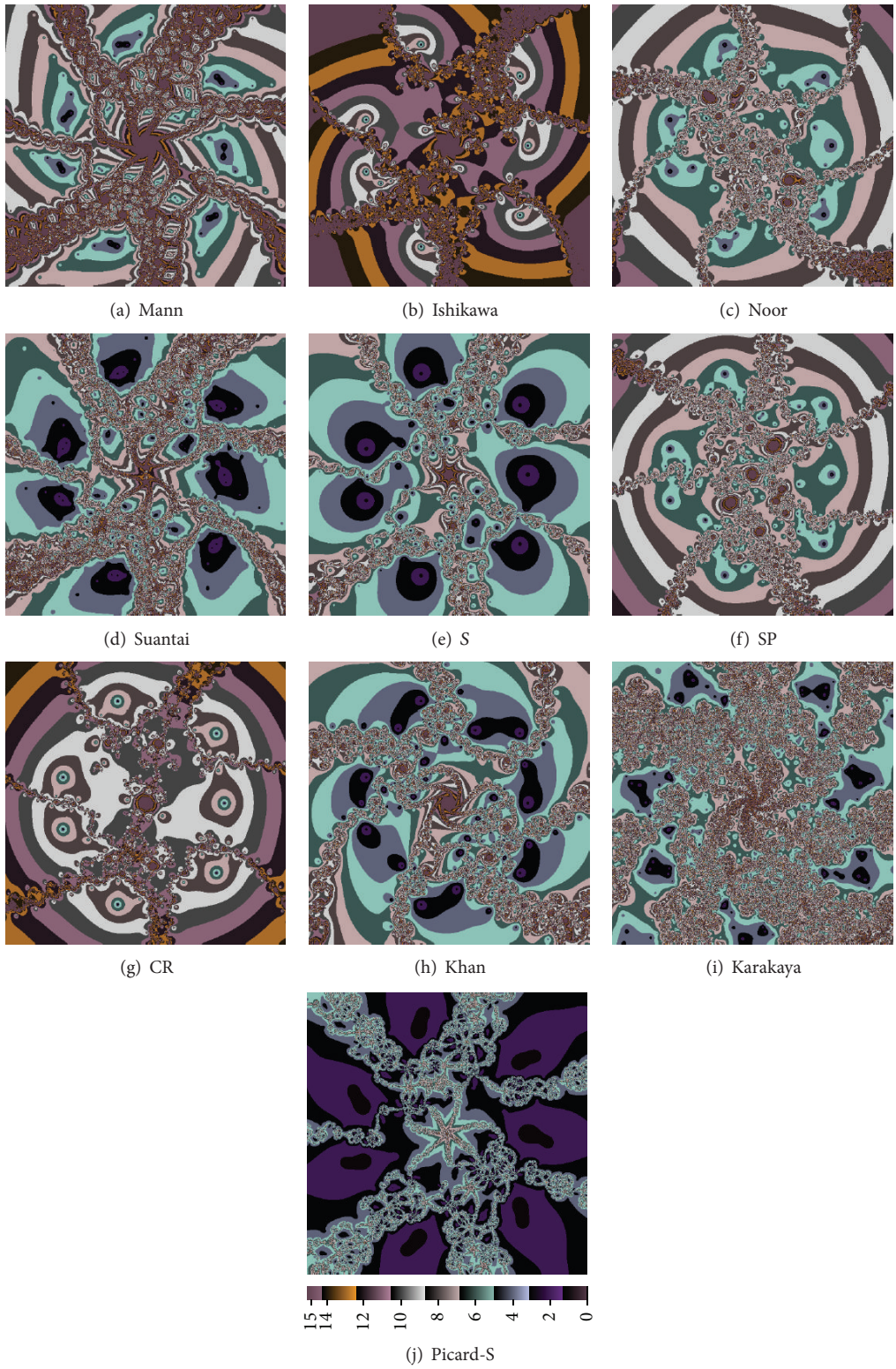
(a) Picard (b) Mann (c) Ishikawa

(d) Noor (e) Suantai (f) S

(g) SP (h) CR (i) Khan

(j) Karakaya (k) Picard-S

FIGURE 3: Examples of polynomiographs for p_2 with real-valued parameters of iterations.



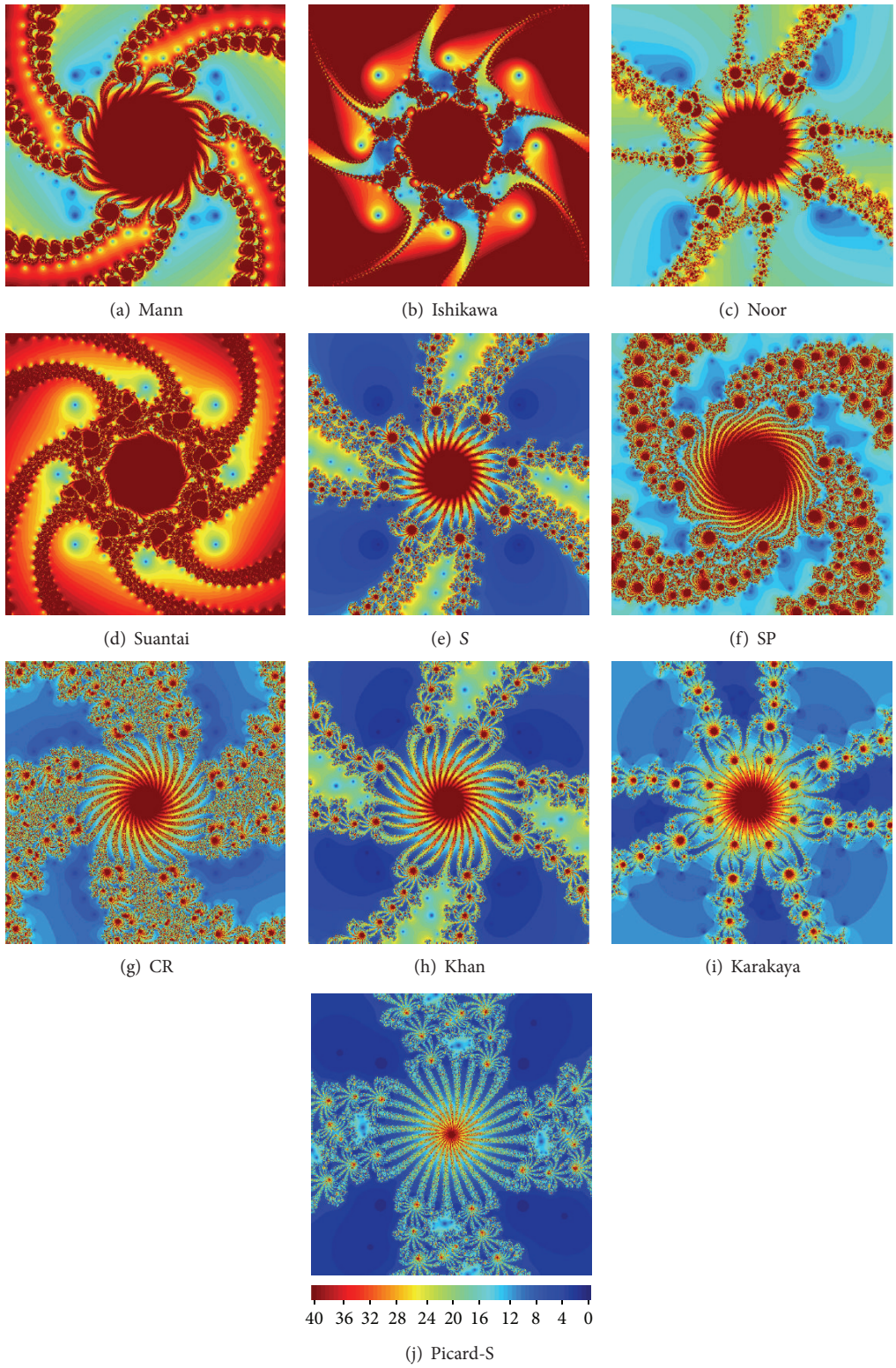
(a) Mann (b) Ishikawa (c) Noor

(d) Suantai (e) S (f) SP

(g) CR (h) Khan (i) Karakaya

(j) Picard-S

Figure 4: Examples of polynomiographs for p_1 with complex-valued parameters of iterations.



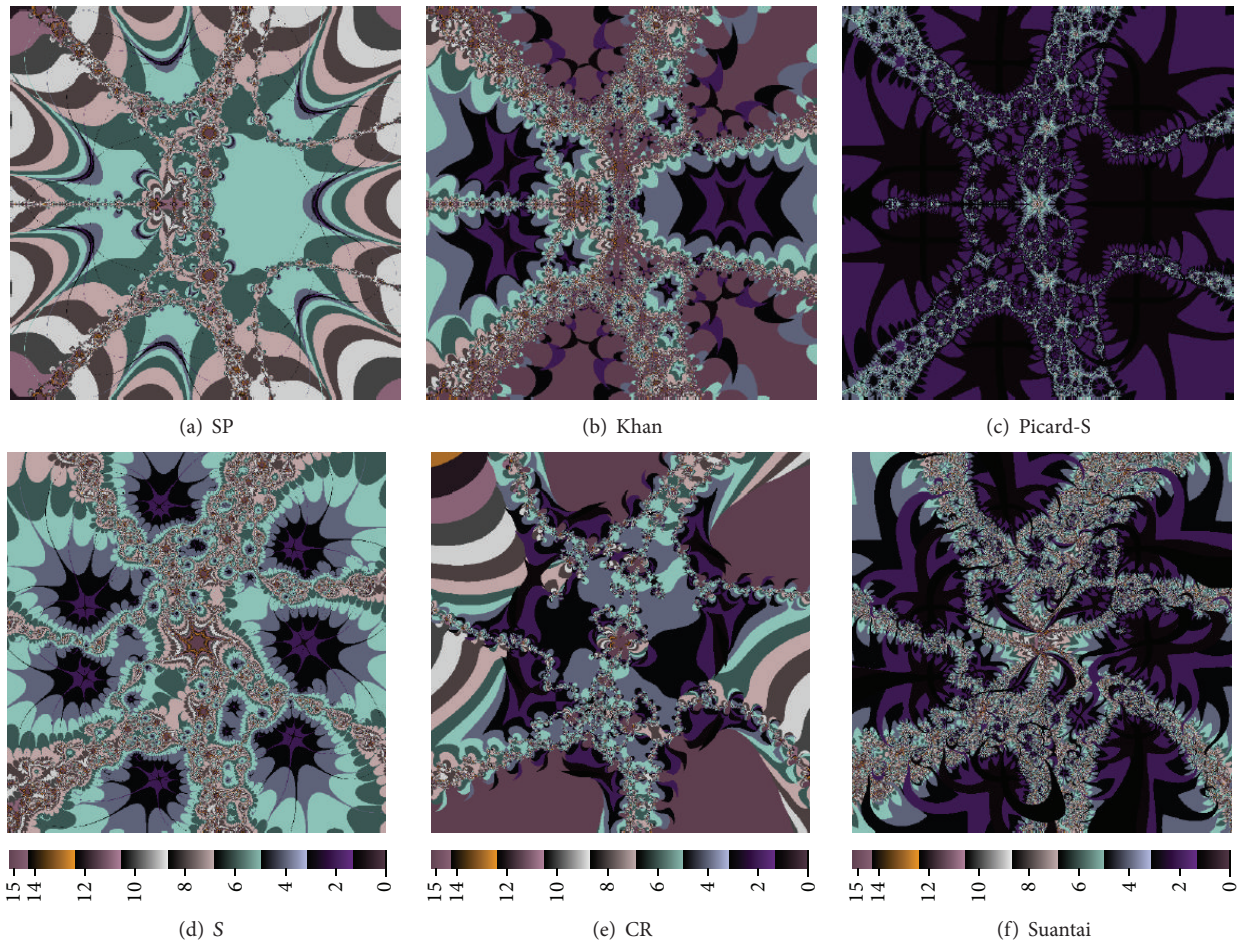
(a) Mann (b) Ishikawa (c) Noor

(d) Suantai (e) S (f) SP

(g) CR (h) Khan (i) Karakaya

(j) Picard-S

Figure 5: Examples of polynomiographs for p_1 with complex-valued parameters of iterations.



(a) SP (b) Khan (c) Picard-S

(d) S (e) CR (f) Suantai

FIGURE 6: Examples of different convergence tests application for p_1, the iterations with: real-valued parameters (a, b, and c) and complex-valued parameters (d, e, and f).

(a) Mann: $\alpha = 0.5 + 0.5\mathbf{i}$,

(b) Ishikawa: $\alpha = 0.1 + 0.11\mathbf{i}$, $\beta = 0.1 - 0.23\mathbf{i}$,

(c) Noor: $\alpha = 0.35 - 0.21\mathbf{i}$, $\beta = 0.7 + 0.8\mathbf{i}$, $\gamma = 0.4 + 0.8\mathbf{i}$,

(d) Suantai: $\alpha = 0.01 - 0.33\mathbf{i}$, $\beta = 0.2$, $\gamma = 0.01 + 0.2\mathbf{i}$, $a = 0.3 + 0.16\mathbf{i}$, $b = 0.1$,

(e) S: $\alpha = 0.99 + 0.1\mathbf{i}$, $\beta = 0.01 + 0.35\mathbf{i}$,

(f) SP: $\alpha = 0.1 + 0.61\mathbf{i}$, $\beta = 0.99 + 0.41\mathbf{i}$, $\gamma = 0.01 + 0.93\mathbf{i}$,

(g) CR: $\alpha = 0.8 - 0.7\mathbf{i}$, $\beta = 0.45 - 0.45\mathbf{i}$, $\gamma = 0.8 + \mathbf{i}$,

(h) Khan: $\alpha = 0.75 + 0.7\mathbf{i}$,

(i) Karakaya: $\alpha = 0.9 + 0.75\mathbf{i}$, $\beta = 0.01$, $\gamma = 0.01$, $a = 0.9 - 0.34\mathbf{i}$, $b = 0.01 - 0.7\mathbf{i}$,

(j) Picard-S: $\alpha = 0.75 - \mathbf{i}$, $\beta = 0.99 + 0.73\mathbf{i}$.

As in the previous example, in this case the addition of imaginary parts to the iterations' parameters causes those swirls and twists to appear in the polynomiographs from Figure 5. This makes the polynomiographs more dynamic and vivid in comparison to those from Figure 3. When one looks at the colours of the polynomiographs one can observe that in some cases the speed of convergence has increased in some areas (e.g., Figure 5(b)) and in some cases it has decreased (e.g., Figure 5(g)) in comparison to the polynomiographs from Figure 3.

7.3. Polynomiographs with Different Convergence Tests. In the next examples we present the use of different convergence tests. The tests that were used are as follows:

(1) $||z_{n+1}|^2 - |z_n|^2| < \varepsilon$ (Pickover's test),

(2) $|0.01(z_{i+1} - z_i)| + |0.029|z_{i+1}|^2 - 0.03|z_i|^2| < \varepsilon$,

(3) $|0.05/|z_{i+1}|^2 - 0.045/|z_i|^2| < \varepsilon$,

(4) $|0.04\Re(z_{i+1} - z_i)| < \varepsilon \vee |0.05\Im(z_{i+1} - z_i)| < \varepsilon$,

(5) $|0.4\Re(z_{i+1} - z_i)|^2 < \varepsilon \wedge |\Im(z_{i+1} - z_i)|^2 < \varepsilon$

and in all cases $\varepsilon = 0.001$.

Figure 6 presents the examples of the use of different convergence tests in the polynomiographs for $p_1(z) = z^7 + z^2 - 1$. Figures 6(a), 6(b), and 6(c) were obtained with the use of the parameters that have been used to generate Figures 2(g), 2(i), and 2(k) but with the use of the tests 1, 2, 4, respectively. On the other hand, Figures 6(d), 6(e), and 6(f) were obtained with the use of the parameters that have been used to generate

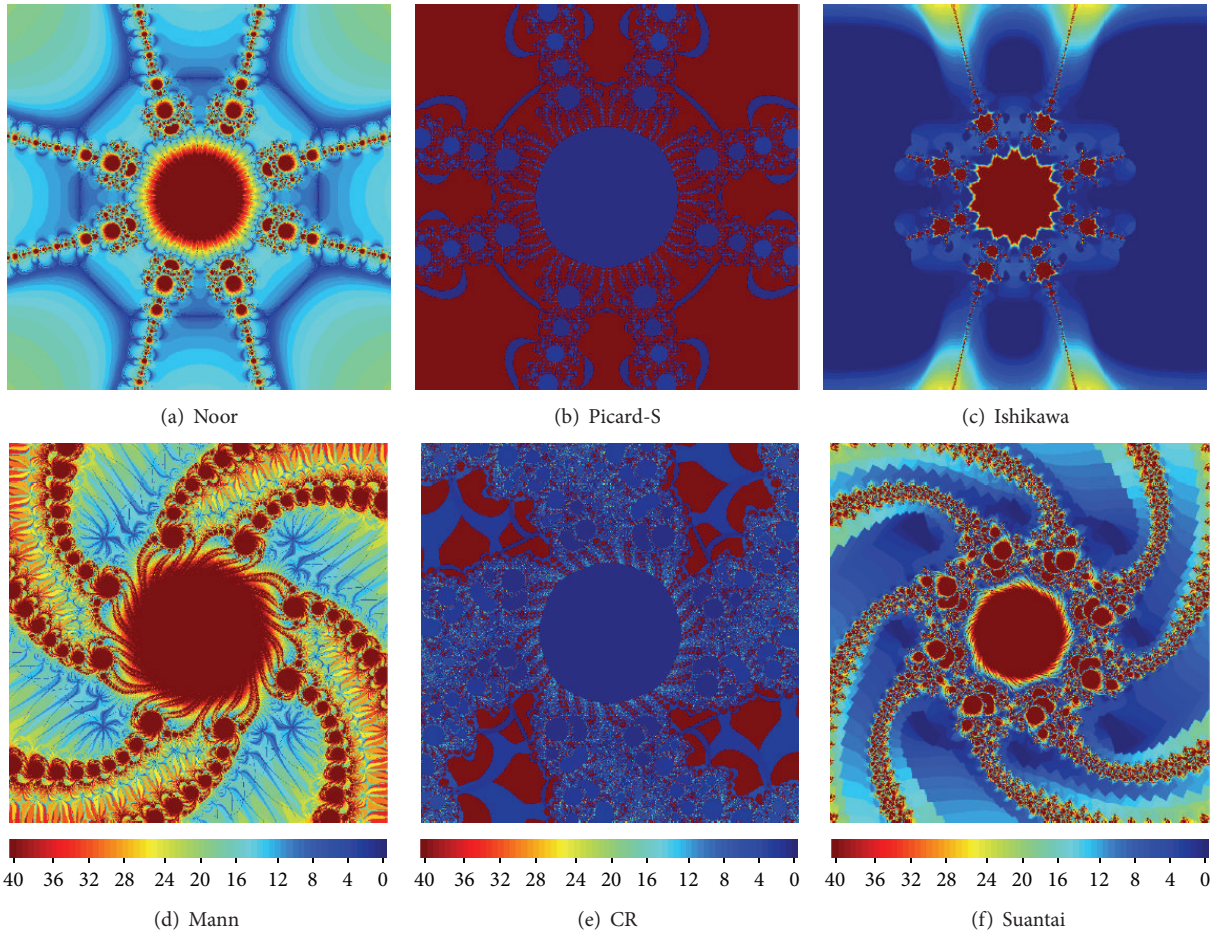


FIGURE 7: Examples of different convergence tests application for p_2, the iterations with: real-valued parameters (a, b, and c) and complex-valued parameters (d, e, and f).

Figures 4(e), 4(g), and 4(d) but with the use of the tests 1, 2, 4, respectively.

Figure 7 presents the examples of the use of different convergence tests in the polynomiographs for $p_2(z) = z^4 + 4$. Figures 7(a), 7(b), and 7(c) were obtained with the use of the parameters that have been used to generate Figures 3(d), 3(k), and 3(c) but with the use of the tests 1, 3, 5, respectively. On the other hand Figures 7(d), 7(e), and 7(f) were obtained with the use of the parameters that have been used to generate Figures 5(a), 5(g), and 5(d) but with the use of the tests 1, 3, 5, respectively.

From the examples presented in Figures 6 and 7 one can see that the different convergence tests have significantly changed the shape of regions of fast convergence. Moreover, one can observe a small change in the areas of slow convergence. When one looks at the polynomiographs obtained with the standard modulus test and with the use of the tests 1–5 one can observe that the areas from the original polynomiographs are very regular and circular, whereas in Figures 6 and 7 the areas have an irregular nature. The change of the shape is very different when one compares the images obtained with different convergence tests.

7.4. Polynomiographs with Different Colour Maps. Figure 8 presents polynomiographs generated with the same parameters that have been used to obtain polynomiograph from Figure 2(h) but with the use of different colour maps. From this example we see that the polynomiographs are strongly dependent not only on iterations but also on the colour maps used. The same graphical information contained in the polynomiograph may be drastically different for different colour maps. The explanation of this is the following. The colours made of the red hues, such as red, magenta, and orange, are warm colours. They are vivid and energetic, and tend to advance in space. The colours made of blue hues, such as blue, cyan, and green, are cold colours. They give an impression of calm and appear to recede from the viewer, so they are good to use for backgrounds. The complementary colours that are opposite to each other on the colour wheel (e.g., red and green) are used to obtain contrast in the image. The analogous colours that are close to each other on the colour wheel (e.g., yellow and orange) create harmony and they are pleasing to the eye. By adding white or black to any colour it is lightened or darkened and called the tint or shade, respectively. Summing up, colours create the depth,

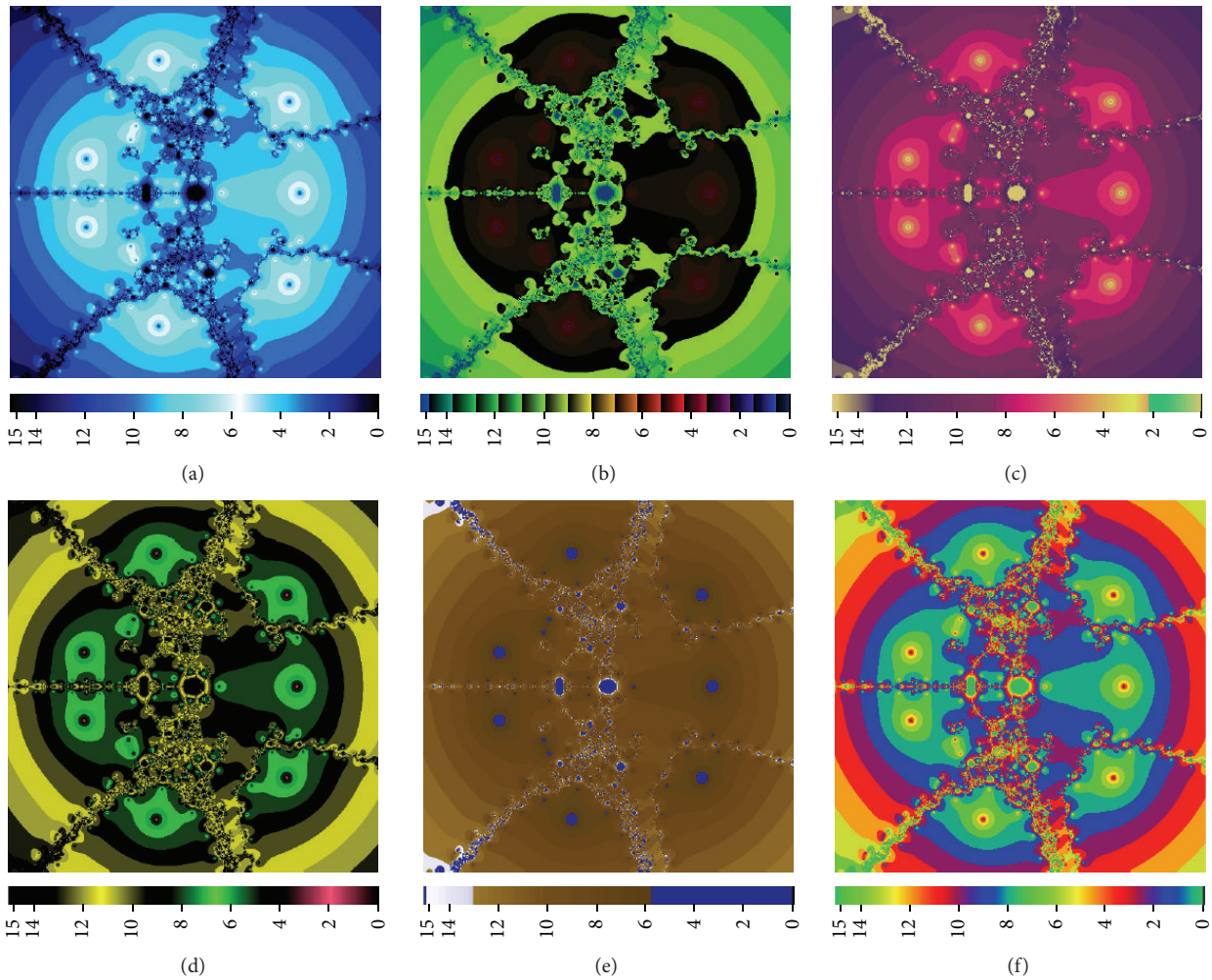


Figure 8: Examples of different colour maps application for the fixed polynomiograph.

movement and mood of an image. Those effects, made by different colour maps, can be observed on the polynomiographs from Figure 8.

8. Time Tests of Polynomiographs Generation

It is a difficult task to estimate theoretically the complexity of the algorithms used for polynomiographs generation. It is because of many factors that should be taken into account. Among these factors are the following: the degree of the polynomial, the root finding method, the computation accuracy, the type of iteration, the type of convergence test, the maximal number of iterations, the polynomiograph's resolution, to mention a few. So, instead of the theoretical complexity estimates we performed some time tests which are presented in this section.

All the experiments were performed on a computer with the following specification: Intel Core i5-4570 processor, 16 GB RAM, and Windows 7 (64-bit). The software for polynomiographs' generation has been implemented in Processing, a programming language based on Java.

The experiments were performed for two sets of parameters:

(1) $p_1(z) = z^7 + z^2 - 1$, $A = [-1.5, 1.5]^2$, $M = 15$, the standard convergence test with $\varepsilon = 0.001$, the Newton's method, and the resolution 600×600 pixels,

(2) $p_2(z) = z^4 + 4$, $A = [-2, 2]^2$, $M = 40$, the standard convergence test with $\varepsilon = 0.001$, the Newton's method, and the resolution 600×600 pixels.

In our experiments we focused on the comparison of the different iteration processes. The iterations have been compared in four groups depending on the number of their parameters: 1 parameter (Mann, Khan), 2 parameters (Ishikawa, S, Picard-S), 3 parameters (Noor, SP, CR), and 5 parameters (Suantai, Karakaya). The experiments have been limited to parameters with real parts only and performed for all possible combinations of discrete values of parameters changed by different steps within each test group. The steps' values and the total numbers of parameters' values combinations are given in Table 1. From a huge number of the experiments only the most representative results are presented in Tables 2–9. Moreover, each polynomiograph has

Table 1: Steps' values and the total numbers of values combinations used in the time tests.

Test group	Step	Total number of combinations
1 Parameter	0.01	99
2 Parameters	0.05	420
3 Parameters	0.10	1210
5 Parameters	0.10	47190

Table 2: Times of polynomiograph's generation for p_1 using one parameter iterations.

α	Time [s]	
	Mann	Khan
1.0	0.760	0.827
0.9	0.768	0.819
0.8	0.803	0.816
0.7	0.855	0.823
0.6	0.931	0.840
0.5	1.042	0.877
0.4	1.158	0.921
0.3	1.257	0.964
0.2	1.317	1.044
0.1	1.319	1.149

Table 3: Times of polynomiograph's generation for p_2 using one parameter iterations.

α	Time [s]	
	Mann	Khan
1.0	0.509	0.510
0.9	0.538	0.515
0.8	0.582	0.520
0.7	0.646	0.529
0.6	0.732	0.567
0.5	0.830	0.560
0.4	0.972	0.575
0.3	1.190	0.600
0.2	1.582	0.627
0.1	2.337	0.666

been generated several times to decrease the influence of other processes and the Virtual Java Machine running on the computer. Then, the worst and the best times have been rejected to obtain the average time generation. It should be added that polynomiographs' time generation for p_1 and p_2 with the standard Picard iteration was 0.663 s and 0.418 s, respectively.

Tables 2 and 3 present slices of the results obtained for the iterations with one parameter for polynomials p_1 and p_2, respectively. From the results we see that for the values of α close to 1 the Mann iteration is faster than the Khan iteration. Then, starting from α equal to about 0.77 for p_1 and 0.97 for p_2 the Khan iteration is faster than the Mann iteration and this remains true up to the lowest values of α. Moreover, we can observe that when the value of α decreases then the time

Table 4: Times of polynomiograph's generation for p_1 using two parameters iterations.

α	β	Time [s]		
		Ishikawa	S	Picard-S
1.0	0.1	1.234	1.249	1.466
1.0	0.5	0.971	0.997	1.300
1.0	1.0	0.915	0.915	1.274
0.8	0.1	1.333	1.285	1.488
0.8	0.5	1.145	0.974	1.268
0.8	1.0	1.129	0.935	1.232
0.5	0.1	1.861	1.356	1.538
0.5	0.5	1.829	1.082	1.349
0.5	1.0	1.719	1.019	1.297
0.2	0.1	2.511	1.403	1.583
0.2	0.5	2.552	1.253	1.467
0.2	1.0	2.514	1.187	1.414

Table 5: Times of polynomiograph's generation for p_2 using two parameters iterations.

α	β	Time [s]		
		Ishikawa	S	Picard-S
1.0	0.1	0.772	0.771	0.880
1.0	0.5	0.627	0.641	0.793
1.0	1.0	0.574	0.580	0.778
0.8	0.1	0.939	0.836	0.924
0.8	0.5	0.863	0.705	0.857
0.8	1.0	0.877	0.657	0.827
0.5	0.1	1.350	0.862	0.930
0.5	0.5	1.350	0.724	0.853
0.5	1.0	1.354	0.688	0.823
0.2	0.1	2.944	0.945	0.977
0.2	0.5	2.882	0.823	0.914
0.2	1.0	2.869	0.775	0.877

Table 6: Times of polynomiograph's generation for p_1 using three parameters iterations.

α	β	γ	Time [s]		
			Noor	SP	CR
0.2	0.4	0.2	3.671	2.120	2.487
0.2	0.4	0.5	3.676	1.689	2.440
0.2	0.4	0.8	3.693	1.459	2.388
0.5	0.5	0.0	2.569	1.807	1.804
0.5	0.5	0.5	2.470	1.365	1.655
0.5	0.5	1.0	2.472	1.188	1.636
0.7	0.9	0.3	1.720	1.172	1.167
0.7	0.9	0.6	1.715	1.068	1.130
0.7	0.9	0.9	1.742	1.064	1.134
1.0	0.2	0.2	1.552	1.416	1.576
1.0	0.4	0.6	1.318	1.112	1.344
1.0	0.9	0.5	1.024	1.026	1.032

Table 7: Times of polynomiograph's generation for p_2 using three parameters iterations.

α	β	γ	Time [s]		
			Noor	SP	CR
0.2	0.4	0.2	4.111	1.571	1.836
0.2	0.4	0.5	4.134	1.265	1.845
0.2	0.4	0.8	4.164	1.031	1.846
0.5	0.5	0.0	1.946	1.356	1.355
0.5	0.5	0.5	1.866	1.005	1.219
0.5	0.5	1.0	1.906	0.826	1.235
0.7	0.9	0.3	1.268	0.783	0.797
0.7	0.9	0.6	1.271	0.737	0.793
0.7	0.9	0.9	1.299	0.702	0.792
1.0	0.2	0.2	0.977	0.875	0.992
1.0	0.4	0.6	0.849	0.745	0.867
1.0	0.9	0.5	0.668	0.675	0.678

Table 8: Times of polynomiograph's generation for p_1 using five parameters iterations.

α	β	γ	a	b	Time [s]	
					Suantai	Karakaya
0.8	0.2	0.7	0.6	0.3	1.791	1.781
0.8	0.2	0.1	0.0	0.6	2.320	2.300
0.8	0.2	0.0	0.0	0.0	3.602	3.599
0.3	0.7	1.0	0.7	0.1	1.979	1.939
0.3	0.7	0.7	1.0	0.0	1.924	1.916
0.3	0.7	0.2	0.8	0.2	2.352	2.372
0.6	0.1	0.7	0.2	0.2	3.059	2.180
0.6	0.2	0.7	0.2	0.2	2.638	2.104
0.6	0.3	0.7	0.2	0.2	2.313	2.026
0.1	0.5	0.3	0.5	0.2	3.676	2.575
0.2	0.5	0.3	0.5	0.2	3.131	2.402
0.3	0.5	0.3	0.5	0.2	2.735	2.295
0.4	0.5	0.3	0.5	0.2	2.337	2.193

Table 9: Times of polynomiograph's generation for p_2 using five parameters iterations.

α	β	γ	a	b	Time [s]	
					Suantai	Karakaya
0.8	0.2	0.7	0.6	0.3	1.196	1.190
0.8	0.2	0.1	0.0	0.6	1.522	1.511
0.8	0.2	0.0	0.0	0.0	2.356	2.347
0.3	0.7	1.0	0.7	0.1	1.250	1.245
0.3	0.7	0.7	1.0	0.0	1.277	1.284
0.3	0.7	0.2	0.8	0.2	1.472	1.477
0.6	0.1	0.7	0.2	0.2	2.198	1.524
0.6	0.2	0.7	0.2	0.2	1.915	1.457
0.6	0.3	0.7	0.2	0.2	1.661	1.393
0.1	0.5	0.3	0.5	0.2	2.640	1.721
0.2	0.5	0.3	0.5	0.2	2.256	1.651
0.3	0.5	0.3	0.5	0.2	1.902	1.544
0.4	0.5	0.3	0.5	0.2	1.599	1.442

difference between the two iterations increases and the time of polynomiograph's generation also increases.

Tables 4 and 5 present slices of the results obtained for the iterations with two parameters for polynomials p_1 and p_2, respectively. When we look at the times for Ishikawa and S iteration we can observe that for the fixed value of $\alpha < 1$ the time difference between those two iterations is increasing when β is also increasing and that the S iteration is faster. We can also observe that the lower the value of α the greater the time difference. In the case of Ishikawa and Picard-S iteration for the values of α close to 1 the Ishikawa iteration is faster. Then, starting from α equal to about 0.8 for p_1 and 0.7 for p_2 the Picard-S iteration is faster than the Ishikawa iteration and the lower the values of α the greater the time differences. Finally, for the S and Picard-S iterations we can observe that for the fixed value of α the S iteration is faster and that the time difference is increasing when the value of β is also increasing. The difference is the smallest for small values of α and increases with the growth of α. Moreover, we can observe that the time of Ishikawa iteration has downward trend with the growth of α, and that the times for S and Picard-S iterations are oscillating is some interval.

For three and five parameter iterations the comparison among iteration types is more complex and difficult to make. Despite the difficulties we made some observations concerning the times. Tables 6 and 7 present slices of the results obtained for the iterations with three parameters for the polynomials p_1 and p_2, respectively. When we look at the times for Noor and SP iterations we can observe that the Noor iteration is slower and that for fixed values of α and β the time difference is increasing with the growth of γ. Moreover, we can observe that with the value drop of α the time differences are getting larger. In the case of the Noor and CR iteration we can observe similar dependencies as in the previous case (substituting SP iteration with CR). Furthermore, for $\alpha = 1$ the two iterations have comparable times. Finally, for the SP and CR iterations we can observe that the SP iteration is faster and that for the fixed values of α and β the time difference is increasing with the growth of γ. Furthermore, when we look at the times of consecutive iterations we see that the iterations have downward trend with the value growth of α for the Noor iteration, $\alpha + \beta + \gamma$ for the SP iteration, and $\alpha + \beta$ for the CR iteration.

Finally, slices of the results for the last group of iterations (with five parameters) for polynomials p_1 and p_2 are presented in Tables 8 and 9, respectively. From these results we can observe that if $\alpha + \beta = 1$, then the times of the iterations are close to each other. When $\alpha + \beta < 1$ then the time difference between the iterations is noticeable. The lower the sum the greater the difference in favour of the Karakaya iteration. When we look at the formulas of the iterations we see that the parameters a and b play similar role as α and β, but in the other step of the iteration process. This may suggest that there may be a similar dependency between the sum of those parameters and the time difference, but the obtained results show that there is no such a dependency.

Additionally, the examples showed that polynomiographs change their shapes in a smooth way with the change of parameters lying in their admissible intervals.

9. Conclusions and Future Work

It is known that mathematics and art are closely connected to each other. A good example of such a connection delivers polynomials and related to them polynomiographs. In the paper by the use of different iteration processes, colouring methods, convergence tests and colourmaps we generalized Kalantari's polynomiography. The polynomiographs presented in this paper look nicely and many of them can be classified as aesthetic ones. Patterns of polynomiographs can be altered by changing the parameters of iterations and convergence tests. Those parameters effect on the complexity, level of details, and more or less fractal appearance of the polynomiographs. Real parts of the parameters alter symmetry, whereas imaginary ones cause asymmetric twisting of the polynomiographs and influence on statics or dynamics of the images. Colourmaps with cold or warm colours alter drastically a visual expression of polynomiographs. This expression can be classified as nostalgic, sad, calm or cheerful, full of energy or sometimes flat or spatial.

Polynomiographs can be helpful to those who are interested in generating of nicely looking images in an automatic way. Those images can inspire graphics designers in their designs.

The results of this paper can be further extended by using the multipoint methods [29–31]. Similar investigations to those presented in this paper can be carried out for complex fractals (Julia and Mandelbrot sets) and biomorphs. Some aspects of such investigations have been reported in the literature [9, 32–34]. Another interesting direction rely on replacing complex numbers by more general: dual and double numbers used in [35] for defining the Q-Systems Fractals. Additionally, by following Kalantari's paper [36] one can generalize polynomiography to analytic functions or even to the quasipolynomials [37] that have infinitely many roots on complex plane, in contrary to polynomials. The problems mentioned above show the possible future work on the polynomiography generalization.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] B. B. Mandelbrot, *The Fractal Geometry of Nature*, W.H. Freeman and Company, New York, NY, USA, 1983.

[2] B. Kalantari, *Polynomial Root-Finding and Polynomiography*, World Scientific, Singapore, 2009.

[3] B. Kalantari, "Method of creating graphical works based on polynomials," U.S. Patent 6,894,705, 2005.

[4] B. Kalantari, "Polynomiography: from the fundamental theorem of Algebra to art," *Leonardo*, vol. 38, no. 3, pp. 233–238, 2005.

[5] W. Kotarski, K. Gdawiec, and A. Lisowska, "Polynomiography via Ishikawa and Mann iterations," in *Advances in Visual Computing, Part I*, G. Bebis, R. Boyle, B. Parvin et al., Eds., vol. 7431 of *Lecture Notes in Computer Science*, pp. 305–313, Springer, Berlin, Germany, 2012.

[6] A. Latif, A. Rafiq, and A. A. Shahid, "Polynomiography via S-iteration Scheme," *Abstract and Applied Analysis*. In press.

[7] S. L. Singh, S. Jain, and S. N. Mishra, "A new approach to superfractals," *Chaos, Solitons and Fractals*, vol. 42, no. 5, pp. 3110–3120, 2009.

[8] B. Prasad and B. Katiyar, "Fractals via Ishikawa iteration," in *Control, Computation and Information Systems*, P. Balasubramaniam, Ed., vol. 140 of *Communications in Computer and Information Science*, pp. 197–203, Springer, Berlin, Germany, 2011.

[9] R. M. Ashish and R. Chugh, "Julia sets and Mandelbrot sets in Noor orbit," *Applied Mathematics and Computation*, vol. 228, no. 1, pp. 615–631, 2014.

[10] M. Rani and R. Chugh, "Dynamics of antifractals in Noor orbit," *International Journal of Computer Applications*, vol. 57, no. 4, pp. 11–15, 2012.

[11] R. L. Burden and J. D. Faires, *Numerical Analysis*, Brooks-Cole, Boston, Mass, USA, 9th edition, 2011.

[12] V. Berinde, *Iterative Approximation of Fixed Points*, Springer, Heidelberg, Germany, 2nd edition, 2007.

[13] S. Ishikawa, "Fixed points by a new iteration method," *Proceedings of the American Mathematical Society*, vol. 44, no. 1, pp. 147–150, 1974.

[14] V. Karakaya, K. Doğan, F. Gürsoy, and M. Ertürk, "Fixed point of a new three-step iteration algorithm under contractive-like operators over normed spaces," *Abstract and Applied Analysis*, vol. 2013, Article ID 560258, 9 pages, 2013.

[15] S. H. Khan, "A Picard-Mann hybrid iterative process," *Fixed Point Theory and Applications*, vol. 2013, article 69, 10 pages, 2013.

[16] W. R. Mann, "Mean value methods in iteration," *Proceedings of the American Mathematical Society*, vol. 4, no. 3, pp. 506–510, 1953.

[17] M. A. Noor, "New approximation schemes for general variational inequalities," *Journal of Mathematical Analysis and Applications*, vol. 251, no. 1, pp. 217–229, 2000.

[18] W. Phuengrattana and S. Suantai, "On the rate of convergence of Mann, Ishikawa, Noor and SP-iterations for continuous functions on an arbitrary interval," *Journal of Computational and Applied Mathematics*, vol. 235, no. 9, pp. 3006–3014, 2011.

[19] S. Suantai, "Weak and strong convergence criteria of Noor iterations for asymptotically nonexpansive mappings," *Journal of Mathematical Analysis and Applications*, vol. 311, no. 2, pp. 506–517, 2005.

[20] E. Picard, "Mémoire sur la théorie des équations aux dérivées partielles et la méthode des approximations successives," *Journal de Mathématiques Pures et Appliquées*, vol. 6, no. 4, pp. 145–210, 1890.

[21] R. P. Agarwal, D. O'Regan, and D. R. Sahu, "Iterative construction of fixed points of nearly asymptotically nonexpansive mappings," *Journal of Nonlinear and Convex Analysis*, vol. 8, no. 1, pp. 61–79, 2007.

[22] R. Chugh, V. Kumar, and S. Kumar, "Strong convergence of a new three step iterative scheme in Banach spaces," *The American Journal of Computational Mathematics*, vol. 2, no. 4, pp. 345–357, 2012.

[23] F. Gürsoy and V. Karakaya, "A Picard-S hybrid type iteration method for solving a differential equation with retarded argument," http://arxiv.org/abs/1403.2546.

[24] C. A. Pickover, "A note on chaos and Halley's method," *Communications of the Association for Computing Machinery*, vol. 31, no. 11, pp. 1326–1329, 1988.

[25] K. Gdawiec, "Polynomiography and various convergence tests," in *Proceedings of the WSCG Communication*, pp. 15–20, 2013.

[26] M. Ó. Searcóid, *Metric Spaces*, Springer Undergraduate Mathematics Series, Springer, London, UK, 2007.

[27] M. Barnsley, *Fractals Everywhere*, Academic Press, Boston, Mass, USA, 1988.

[28] C. A. Pickover, *Computers, Pattern, Chaos, and Beauty: Graphics from an Unseen World*, Dover Publications, Mineola, NY, USA, 2001.

[29] T. Lotfi, F. Soleymani, S. Sharifi, S. Shateyi, and F. K. Haghani, "Multipoint iterative methods for finding all the simple zeros in an interval," *Journal of Applied Mathematics*, vol. 2014, Article ID 601205, 13 pages, 2014.

[30] M. S. Petković, B. Neta, L. D. Petković, and J. Džunić, "Multipoint methods for solving nonlinear equations: a survey," *Applied Mathematics and Computation*, vol. 226, pp. 635–660, 2014.

[31] F. Zafar, N. Hussain, Z. Fatimah, and A. Kharal, "Optimal sixteenth order convergent method based on quasi-hermite interpolation for computing roots," *The Scientific World Journal*, vol. 2014, Article ID 410410, 18 pages, 2014.

[32] Y. S. Chauhan, R. Rana, and A. Negi, "New Julia sets of Ishikawa iterates," *International Journal of Computer Applications*, vol. 7, no. 13, pp. 34–42, 2010.

[33] M. Rani and V. Kumar, "Superior Julia set," *Journal of the Korea Society of Mathematical Education Series D: Research in Mathematical Education*, vol. 8, no. 4, pp. 261–277, 2004.

[34] M. Rani and V. Kumar, "Superior Mandelbrot set," *Journal of the Korea Society of Mathematical Education Series D: Research in Mathematical Education*, vol. 8, no. 4, pp. 279–291, 2004.

[35] M. Levin, "Discontinuous and alternate q-system fractals," *Computers & Graphics*, vol. 18, no. 6, pp. 873–884, 1994.

[36] B. Kalantari, "Voronoi diagram properties in polynomials with polynomiography applications and extensions," in *Proceedings of the 9th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD '12)*, pp. 32–40, June 2012.

[37] A. B. Pitcher and R. M. Corless, "Quasipolynomial root-finding: a numerical homotopy method," in *Proceedings of the Canadian Undergraduate Mathematics Conference*, Ontario, Canada, 2005.