Available online at www.tjnsa.com
J. Nonlinear Sci. Appl. 8 (2015), 617–627
Research Article

S-iteration scheme and polynomiography

Shin Min Kang[a], Hamed H. Alsulami[b], Arif Rafiq[c,*], Abdul Aziz Shahid[d]

[a]*Department of Mathematics and the RINS, Gyeongsang National University, Jinju 660-701, Korea.*
[b]*Nonlinear Analysis and Applied Mathematics Research Group, King Abdulaziz University, Jeddah 21589, Saudi Arabia.*
[c]*Department of Mathematics, Lahore Leads University, Lahore 54810, Pakistan.*
[d]*Department of Mathematics, Lahore Leads University, Lahore 54810, Pakistan.*

Abstract

The aim of this paper is to present a modification of the visualization process of finding the roots of a given complex polynomial. In this paper S-iteration scheme is used instead of standard Picard iteration. The word "polynomiography" coined by Kalantari for that visualization process. The images obtained are called polynomiographs. Polynomiographs have importance for both the art and science aspects. By using S-iteration scheme we obtain quite new nicely looking polynomiographs that are different from standard Picard iteration. Presented examples show that we obtain very interesting patterns for complex polynomial equations, permutation and doubly stochastic matrices. We believe that the results of this paper enrich the functionality of the existing polynomiography software.

Keywords: S-iteration scheme, polynomiography, permutation matrix, doubly stochastic matrix.
2010 MSC: 30C10, 30C15

1. Introduction

Polynomials are one of the most significant objects in many fields of mathematics. Isaac Newton proposed the method for finding the roots of polynomial approximately. The behavior of Newton's method in the complex plane as applied to the equation $z^3 - 1 = 0$ initially investigated by Cayley in 1879, which is known as Cayley's problem [3]. This problem was solved by Julia [5] in 1918 and then Mandelbrot [10] in 1982. Kalantari discovered a new method for finding the roots of complex polynomial called polynomiography. In 2005 Kalantari [6] obtained an USA patent on the use of polynomiograph in the generation of nicely looking graphics. Polynomiography is defined to be "the art and science of visualization in approximation of the

*Corresponding author
Email addresses: smkang@gnu.ac.kr (Shin Min Kang), hhaalsalmi@kau.edu.sa (Hamed H. Alsulami), aarafiq@gmail.com (Arif Rafiq), abdulazizhanfi@hotmail.com (Abdul Aziz Shahid)

Received 2015-02-17

zeros of complex polynomials, via fractal and non fractal images created using the mathematical convergence properties of iteration functions" [7]. An individual image is called a "polynomiograph".

Polynomiography gives a new way to solve the ancient problem by using new algorithms and computer technology. Polynomiography is based on the use of one or an infinite number of iteration methods formulated for the purpose of approximation of the root of polynomials, for example, Newton's method, Halley's method, etc. The word "fractal", which partially appeared in the definition of polynomiography, was coined by the famous mathematician Mandelbrot [10]. Both fractal images and polynomiographs can be obtained via different iterative schemes. Fractals are self-similar has typical structure and independent of scale. On the other hand, polynomiographs are quite different. The "polynomiographer" can be controlled the shape and designed in a more predictable way by using different iteration methods to the infinite variety of complex polynomials. Generally, fractals and polynomiographs belong to different classes of graphical objects. Polynomiography has diverse applications in mathematics, science, education, art and design. According to fundamental theorem of algebra, any complex polynomial with complex coefficients:

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + ... + a_1 z + a_0 \tag{1.1}$$

of degree n has n roots (zeros) which may or may not be distinct. The degree of polynomial describes the number of basins of attraction and placing roots on the complex plane manually localization of basins can be controlled. Description of polynomiography, its theoretical background and artistic applications are described in [6, 7].

In this paper we use S-iteration scheme instead of standard Picard iteration to obtain some modifications of the Newton method and iteration methods from Basic Family of iterations [8]. Earlier, Mann [11] and Ishikawa [4] iterations were used to generate polynomiographs [9].

The paper is organized as follows. In Section 2, for Newton's method of finding roots and its generalizations, we present iteration formula. In Section 3, the examples of polynomiographs with S-iteration scheme for complex equations, permutation and doubly stochastic matrices are presented. In Section 4, comparison of S-scheme and Ishikawa iterations are presented. The last section, Section 5 describes some conclusions for future work.

2. Newton roots finding method and its generalizations

Newton method for finding the roots of a complex polynomial p is given by the formula:

$$z_{n+1} = z_n - \frac{p(z_n)}{p'(z_n)}, \quad n = 0, 1, 2, ..., \tag{2.1}$$

where $z_0 \in \mathbb{C}$ is a starting point. Generalizations procedures for approximation of all the roots of complex polynomials, making use of a fundamental family of iteration functions given in [7, 8]. This family is called "Basic Family". It is represented as $\{B_m(z)\}_{m=2}^{\infty}$. Let $p(z)$ be a polynomial of degree $n \geq 2$ with complex coefficients. Define $D_0(z) = 1$ and for each natural number $m \geq 1$, let

$$D_m(z) = \det \begin{bmatrix} p'(z) & \frac{p''(z)}{2!} & \cdots & \frac{p^{(m-1)}(z)}{(m-1)!} & \frac{p^{(m)}(z)}{(m)!} \\ p(z) & p'(z) & \ddots & \ddots & \frac{p^{(m-1)}(z)}{(m-1)!} \\ 0 & p(z) & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & p'(z) & \frac{p''(z)}{2!} \\ 0 & 0 & \cdots & p(z) & p'(z) \end{bmatrix}. \tag{2.2}$$

Then Basic Family of iterations is defined as:

$$B_m(z) = z - p(z)\frac{D_{m-2}(z)}{D_{m-1}(z)}, \quad m = 2, 3, 4, ... \tag{2.3}$$

It is clear that the first member of the sequence, $B_2(z)$ is the Newton method and $B_3(z)$ is the Halley method. Members of Basic Family can be modified using S-iteration scheme because that iteration produce only different orbits in comparison to Picard iteration. It has different character of convergence and by using different kinds of iteration we obtain entirely different basins of attraction to roots of complex polynomial p.

Let $T : X \to X$ be a mapping on a metric space (X, d), where d is a metric. Let $u_0 \in X$ be a starting point. Then, we recall the iterative procedures:

Picard iteration scheme [2]:

$$x_{n+1} = Tx_n, \quad n = 0, 1, 2, ... \tag{2.4}$$

Ishikawa iteration scheme [4]:

$$\begin{cases} u_{n+1} = (1-\alpha_n)u_n + \alpha_n T v_n \\ v_n = (1-\beta_n)u_n + \beta_n T u_n, \quad n = 0, 1, 2, ..., \end{cases} \tag{2.5}$$

where $0 < \alpha_n \leq 1$ and $0 \leq \beta_n \leq 1$.

S-iteration scheme [1]:

$$\begin{cases} u_{n+1} = (1-\alpha_n)Tu_n + \alpha_n T v_n \\ v_n = (1-\beta_n)u_n + \beta_n T u_n, \quad n = 0, 1, 2, ..., \end{cases} \tag{2.6}$$

where $0 < \alpha_n \leq 1$ and $0 \leq \beta_n \leq 1$.

To find the fixed point x^* such that $x^* = T(x^*)$ and its approximations under additional assumptions on the space X that should be Banach one and the mapping T should be contractive. Therefore the standard Picard iteration is used in the Banach Fixed Point Theorem [2]. The S-iteration scheme allow to weak the assumptions on the mapping T. We will take the space $X = \mathbb{C}$ or $X = \mathbb{R}^2$ that is Banach one. We take $u_0 = (x_0, y_0)$ and $\alpha_n = \alpha$, $\beta_n = \beta$ such that $0 < \alpha \leq 1$ and $0 \leq \beta \leq 1$. It can be seen that the S-iteration scheme with $\alpha = 1$ and $\beta = 0$ is Picard iteration.

Applying the Ishikawa iteration scheme (2.5) in (2.1) we obtain the following formula [9]:

$$\begin{cases} z_{n+1} = \alpha\left(v_n - \frac{p(v_n)}{p'(v_n)}\right) + (1-\alpha)z_n \\ v_n = \beta\left(z_n - \frac{p(z_n)}{p'(z_n)}\right) + (1-\beta)z_n, \quad n = 0, 1, 2, ..., \end{cases} \tag{2.7}$$

where $0 < \alpha \leq 1$ and $0 \leq \beta \leq 1$.

Applying the S-iteration scheme (2.6) in (2.1) we obtain the following formula:

$$\begin{cases} z_{n+1} = \alpha\left(v_n - \frac{p(v_n)}{p'(v_n)}\right) + (1-\alpha)\left(z_n - \frac{p(z_n)}{p'(z_n)}\right) \\ v_n = \beta\left(z_n - \frac{p(z_n)}{p'(z_n)}\right) + (1-\beta)z_n, \quad n = 0, 1, 2, ..., \end{cases} \tag{2.8}$$

where $0 < \alpha \leq 1$ and $0 \leq \beta \leq 1$.

The sequence $\{z_n\}_{n=0}^{\infty}$ is called the orbit of the point z_0 converges to a root z^* of p then, we say that z_0 is attracted to z^*. A set of all such starting points for which $\{z_n\}_{n=0}^{\infty}$ converges to root z^* is called the basin of attraction of z^*. Fractals by Newton method can be seen at boundaries of basins. In [13] basin boundaries of Newton method (2.1) are discussed. The formula (2.8) is used in the next section to obtain polynomiographs for complex polynomial equations, permutation and doubly stochastic matrices.

3. Applications of polynomiographs

In this section we present some examples of polynomiographs for complex polynomial equations, permutation and doubly stochastic matrices. They are obtained for different parameters α and β via Newton method using Picard and S-iteration scheme.

3.1. Polynomiographs for complex cubic

Complex polynomial equation $z^3 - 1 = 0$ having three roots: 1, $-\frac{1}{2} - \frac{\sqrt{3}}{2}i$ and $-\frac{1}{2} + \frac{\sqrt{3}}{2}i$. Some polynomiographs are presented in the following figures with three distinct basins of attraction to the three roots of the polynomial $z^3 - 1 = 0$. The different colors of a image depend upon number of iterations to reach a root with given accuracy $\varepsilon = 0.001$. One can obtain infinitely many nicely looking polynomiographs by changing parameters $\alpha, \beta, \varepsilon$ and k, where k is the upper bound of the number of iterations was fixed as $k = 12$.

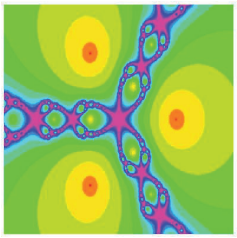

Figure 1: Picard iteration

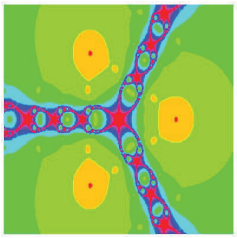

Figure 2: S-iteration scheme for $\alpha = 0.8$, $\beta = 0.6$

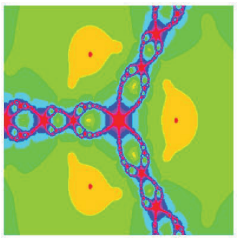

Figure 3: S-iteration scheme for $\alpha = 0.7$, $\beta = 1.0$

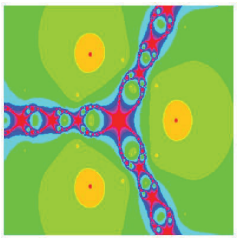

Figure 4: S-iteration scheme for $\alpha = 0.6$, $\beta = 0.8$

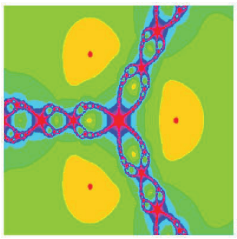

Figure 5: S-iteration scheme for $\alpha = 0.8$, $\beta = 1.0$

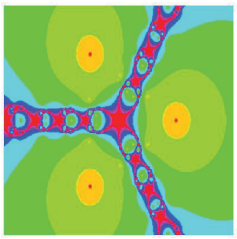

Figure 6: S-iteration scheme for $\alpha = 0.7$, $\beta = 0.5$

3.2. Polynomiographs for $z^3 + z^2 + 1 = 0$

Some polynomiographs are presented in the following figures with three distinct basins of attraction to the three roots of the polynomial $z^3 + z^2 + 1 = 0$.

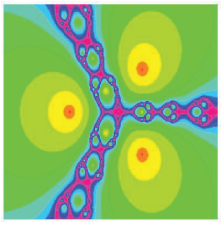

Figure 7: Picard iteration

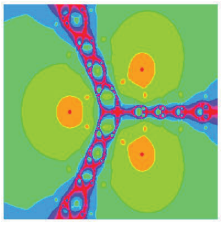

Figure 8: S-iteration scheme for $\alpha = 0.8$, $\beta = 0.6$

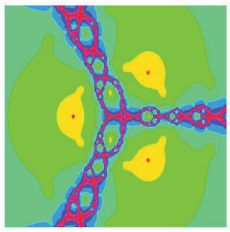

Figure 9: S-iteration scheme for $\alpha = 0.7$, $\beta = 1.0$

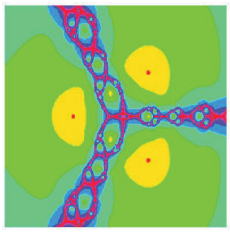

Figure 10: S-iteration scheme for $\alpha = 0.8$, $\beta = 1.0$

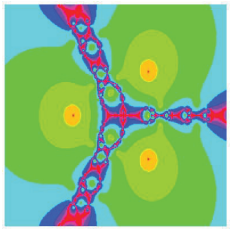

Figure 11: S-iteration scheme for $\alpha = 0.4$, $\beta = 0.8$

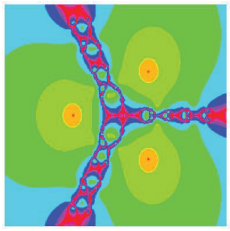

Figure 12: S-iteration scheme for $\alpha = 0.5$, $\beta = 0.7$

3.3. Polynomiographs for $z^4 - 1 = 0$

Some polynomiographs are presented in the following figures with four distinct basins of attraction to the four roots of the polynomial $z^4 - 1 = 0$.

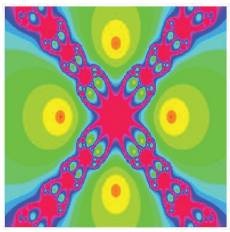

Figure 13: Picard iteration

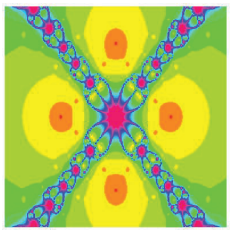

Figure 14: S-iteration scheme for $\alpha = 0.8$, $\beta = 0.6$

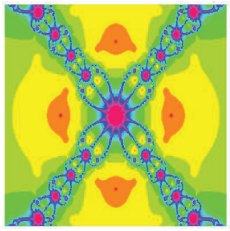

Figure 15: S-iteration scheme for $\alpha = 0.7$, $\beta = 1.0$

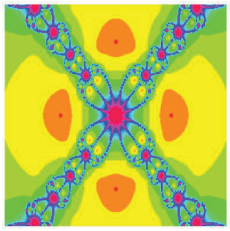

Figure 16: S-iteration scheme for $\alpha = 0.8$, $\beta = 1.0$

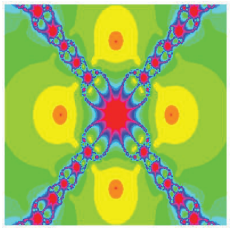

Figure 17: S-iteration scheme for $\alpha = 0.4$, $\beta = 0.8$

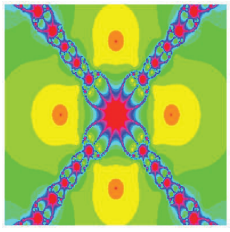

Figure 18: S-iteration scheme for $\alpha = 0.5$, $\beta = 0.7$

3.4. Polynomiographs for $z^5 - 1 = 0$

Some polynomiographs are presented in the following figures with five distinct basins of attraction to the five roots of the polynomial $z^5 - 1 = 0$.

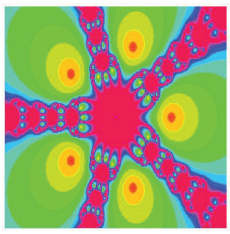

Figure 19: Picard iteration

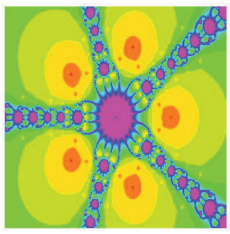

Figure 20: S-iteration scheme for $\alpha = 0.8$, $\beta = 0.6$

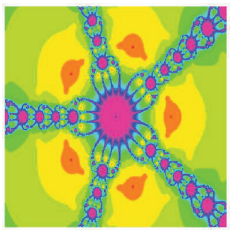

Figure 21: S-iteration scheme for $\alpha = 0.7$, $\beta = 1.0$

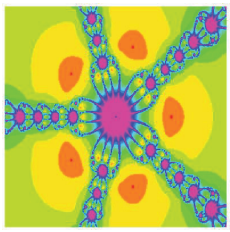

Figure 22: S-iteration scheme for $\alpha = 0.8$, $\beta = 1.0$

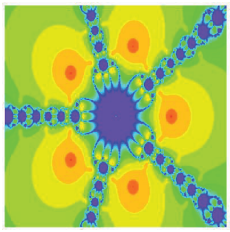

Figure 23: S-iteration scheme for $\alpha = 0.7$, $\beta = 1.0$

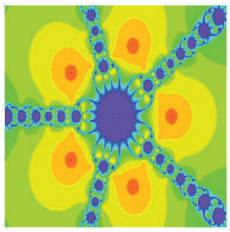

Figure 24: S-iteration scheme for $\alpha = 0.5$, $\beta = 0.7$

3.5. Polynomiographs for permutation matrices

An $n \times n$ matrix $\Pi = (\pi_{ij})$ is a matrix whose rows and columns form a permutation of the identity matrix. To each permutation matrix Π,we associate a complex polynomial defined as follows. First, we set θ_{ij} to be the complex number associated to the location (i, j).

$$\theta_{ij} = i + j\mathbf{i}, \quad \mathbf{i} = \sqrt{-1}.$$

Now ginen the matrix Π we define a corresponding matrix

$$\overline{\Pi} = (\overline{\pi}_{ij}), \quad \overline{\pi}_{ij} = \pi_{j,(n+1-i)}.$$

This matrix is analogus to the transpose, except that the i-th row of Π corresponds to the i-th column of $\overline{\Pi}$ corresponds to the bottom up. Finally, the matrix $\Pi = (\pi_{ij})$ can be associated to a complex polynomial $p_\Pi(z)$ can be defined as

$$p_\Pi(z) = \prod_{\overline{\pi}_{ij}=1} (z - \theta_{ij}),$$

a polynomial of degree n associated as the complex permutation polynomial corresponding to Π. We create polynomiography with permutations polynomials. As an example, for $n = 2$ the permutation matrices Π_1,

Π_2, $\overline{\Pi}_1$ and $\overline{\Pi}_2$ and their corresponding polynomials are

$$\Pi_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \Pi_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \overline{\Pi}_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \overline{\Pi}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$p_{\Pi_1}(z) = (z - (1 + 2\mathbf{i}))(z - (2 + \mathbf{i})), \tag{3.1}$$

$$p_{\Pi_2}(z) = (z - (1 + \mathbf{i}))(z - (2 + \mathbf{i})). \tag{3.2}$$

Many polynomiographs can be associated to these permutation polynomials. The polynomiographs of above mentioned polynomials are presented in the following figures. Polynomiographs obtained via S-iteration scheme for different α, β are different as compared to the Picard iteration. All the images are obtained with accuracy $\varepsilon = 0.01$ and $k = 6$.

Polynomiographs for equation (3.1)

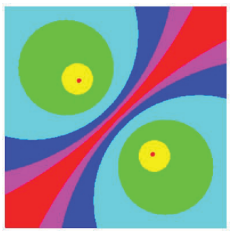

Figure 25: Picard iteration

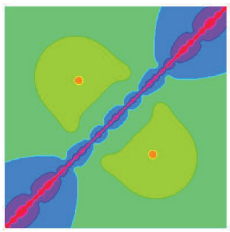

Figure 26: S-iteration scheme for $\alpha = 0.7$, $\beta = 0.9$

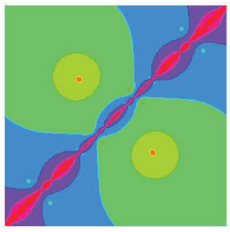

Figure 27: S-iteration scheme for $\alpha = 0.5$, $\beta = 0.6$

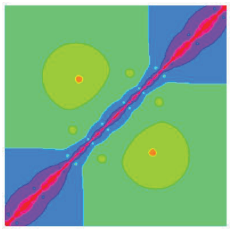

Figure 28: S-iteration scheme for $\alpha = 0.6$, $\beta = 0.9$

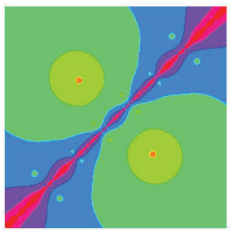

Figure 29: S-iteration scheme for $\alpha = 0.7$, $\beta = 0.5$

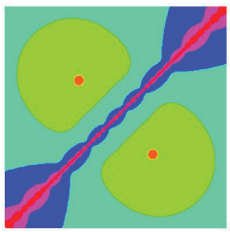

Figure 30: S-iteration scheme for $\alpha = 0.8$, $\beta = 1.0$

Polynomiographs for equation (3.2)

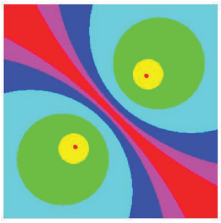

Figure 31: Picard iteration

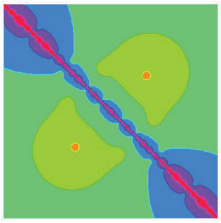

Figure 32: S-iteration scheme for $\alpha = 0.7$, $\beta = 0.9$

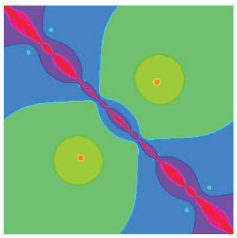

Figure 33: S-iteration scheme for $\alpha = 0.5$, $\beta = 0.6$

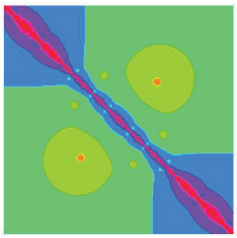

Figure 34: S-iteration scheme for $\alpha = 0.6$, $\beta = 0.9$

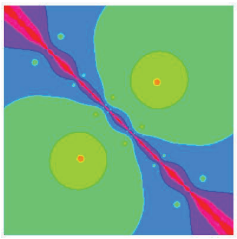

Figure 35: S-iteration scheme for $\alpha = 0.7$, $\beta = 0.5$

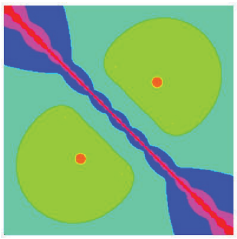

Figure 36: S-iteration scheme for $\alpha = 0.8$, $\beta = 1.0$

3.6. Polynomiographs for doubly stochastic matrix

An $n \times n$ matrix is doubly stochastic matrix if all elements are non negative reals and the sum of the entries in each row and each column equals 1. We will associate a corresponding polynomial to each doubly stochastic matrix. According to Birkhoff-von Neumann theorem [12] any double stochastic matrix A can be represented as a convex combination of permutation matrices $A = \sum_{i=1}^{k} \alpha_i \Pi_i$,

where $\sum_{i=1}^{k} \alpha_i = 1$ and $\alpha_i \geq 0$ for $i = 1, 2, 3, ..., k$. The corresponding complex polynomial p_A to a doubly stochastic matrix $A = (a_{ij})$ is defined as follows:

$$p_A(z) = \prod_{\overline{a}_{ij}>0} (z - \overline{a}_{ij}\theta_{ij}),$$

where θ_{ij} be as defined previously and $\overline{A}$ to A is constructed as matrix $\overline{\Pi}$ to Π. As an example, for $n = 2$ take the following doubly stochastic matrix A :

$$A = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The corresponding complex polynomial p_A to the matrix A has the following form:

$$p_A(z) = \left(z - \frac{1+\mathbf{i}}{2}\right)\left(z - \frac{1+2\mathbf{i}}{2}\right)\left(z - \frac{2+\mathbf{i}}{2}\right)\left(z - \frac{2+2\mathbf{i}}{2}\right). \tag{3.3}$$

Polynomiographs for a doubly stochastic matrix A are presented in following figures.

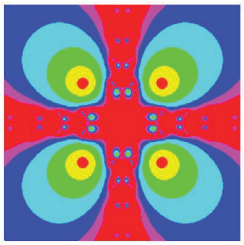

Figure 37: Picard iteration

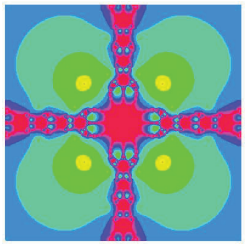

Figure 38: S-iteration scheme for $\alpha = 0.5$, $\beta = 1.0$

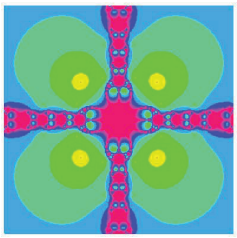

Figure 39: S-iteration scheme for $\alpha = 0.6$, $\beta = 0.8$

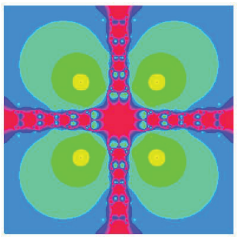

Figure 40: S-iteration scheme for $\alpha = 0.8$, $\beta = 0.5$

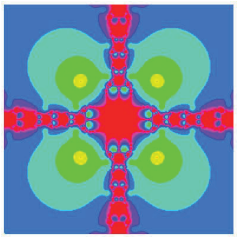

Figure 41: S-iteration scheme for $\alpha = 0.4$, $\beta = 0.8$

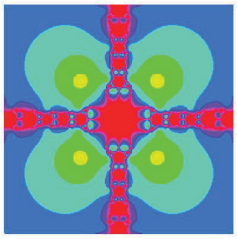

Figure 42: S-iteration scheme for $\alpha = 0.5$, $\beta = 0.7$

4. Comparison of Ishikawa and S-iteration schemes

Time for generating polynomiographs in core i5, 2.60GHz by using MAPLE software given in the following tables. Our results shows that convergence of S-iteration scheme relatively faster than Ishikawa iteration.

Table 1: Time for generating polynomiographs of $z^3 - 1 = 0$

(α, β)	Time for Ishikawa Iteration	Time for S-iteration scheme
$(0.8, 0.6)$	$5.56s$	$3.62s$
$(0.7, 1.0)$	$6.67s$	$3.60s$
$(0.6, 0.8)$	$8.40s$	$3.85s$
$(0.8, 1.0)$	$5.60s$	$3.57s$
$(0.7, 0.5)$	$6.70s$	$4.04s$

Table 2: Time for generating polynomiographs of $z^3 + z^2 + 1 = 0$

(α, β)	Time for Ishikawa Iteration	Time for S-iteration scheme
$(0.8, 0.6)$	$8.18s$	$5.40s$
$(0.7, 1.0)$	$9.65s$	$5.26s$
$(0.8, 1.0)$	$7.76s$	$4.87s$
$(0.4, 0.8)$	$15.5s$	$5.85s$
$(0.5, 0.7)$	$14.73s$	$5.76s$

Table 3: Time for generating polynomiographs of $z^4 - 1 = 0$

(α, β)	Time for Ishikawa Iteration	Time for S-iteration scheme
$(0.8, 0.6)$	$7.79s$	$5.70s$
$(0.7, 1.0)$	$9.06s$	$5.57s$
$(0.8, 1.0)$	$7.95s$	$5.40s$
$(0.4, 0.8)$	$13.89s$	$6.76s$
$(0.5, 0.7)$	$13.23s$	$6.88s$

Table 4: Time for generating polynomiographs of $z^5 - 1 = 0$

(α, β)	Time for Ishikawa Iteration	Time for S-iteration scheme
$(0.8, 0.6)$	$8.48s$	$6.64s$
$(0.7, 1.0)$	$9.89s$	$6.39s$
$(0.8, 1.0)$	$9.26s$	$6.71s$
$(0.4, 0.8)$	$14.79s$	$7.40s$
$(0.5, 0.7)$	$14.23s$	$7.42s$

Table 5: Time for generating polynomiographs of permutation matrix 1

(α, β)	Time for Ishikawa Iteration	Time for S-iteration scheme
$(0.7, 0.9)$	$6.37s$	$4.05s$
$(0.5, 0.6)$	$6.45s$	$3.73s$
$(0.6, 0.9)$	$6.46s$	$3.51s$
$(0.7, 0.5)$	$6.15s$	$3.64s$
$(0.8, 1.0)$	$5.15s$	$3.18s$

Table 6: Time for generating polynomiographs of permutation matrix 2

(α, β)	Time for Ishikawa Iteration	Time for S-iteration scheme
$(0.7, 0.9)$	$5.98s$	$3.98s$
$(0.5, 0.6)$	$6.51s$	$3.67s$
$(0.6, 0.9)$	$5.54s$	$3.25s$
$(0.7, 0.5)$	$5.95s$	$3.54s$
$(0.8, 1.0)$	$4.87s$	$3.06s$

Table 7: Time for generating polynomiographs of doubly stochastic matrix

(α, β)	Time for Ishikawa Iteration	Time for S-iteration scheme
$(0.5, 1.0)$	$22.56s$	$13.39s$
$(0.6, 0.8)$	$19.40s$	$13.31s$
$(1.0, 0.5)$	$15.23s$	$13.46s$
$(0.4, 0.8)$	$23.95s$	$14.35s$
$(0.5, 0.7)$	$22.50s$	$13.92s$

5. Conclusions

In this paper we presented some modifications of the complex polynomial roots finding visualization process and generalizations of the Newton method obtained by the S-iteration scheme instead of the standard Picard iteration. Polynomiographs obtained for different complex equations, permutation and doubly stochastic matrices are quite different in comparison to Picard iteration. S-iteration scheme can be used to generalize Basic Family of iteration. We think that the results of this paper can be inspired those who are interested in creating automatically aesthetic patterns. We believe that polynomiography can be used to teach about polynomials and polynomial roots finding, also the functionality of the existing polynomiography software can be increased using S-iteration scheme.

References

[1] R.P. Agarwal, D. O'Regan, D.R. Sahu, *Iterative construction of fixed points of nearly asymptotically nonexpansive mappings*, J. Nonlinear Convex Anal., **8** (2007), 61–79. 2

[2] V. Berinde, *Iterative Approximation of Fixed Points*, Second Edition, Lectures Notes in Mathematics 1912, Springer-Verlag, Berlin, (2007). 2, 2

[3] A. Cayley, *The Newton-Fourier imaginary problem*, Amer. J. Math., **2** (1879), 97. 1

[4] S. Ishikawa, *Fixed point by a new iteration method*, Proc. Amer. Math. Soc., **4** (1974), 147–150. 1, 2

[5] G. Julia, *Mémoire sur l'iteration des fonctions rationnelles*, J. Math. Pures Appl., **8** (1918), 47–246. 1

[6] B. Kalantari, *Polynomiography: From the Fundamental Theorem of Algebra to Art*, Leonardo, **38** (2005), 233–238. 1, 1

[7] B. Kalantari, *Polynomial Root-finding and Polynomiography*, World Scientific Publishing Co. Pte. Ltd., New Jersey, (2009). 1, 1, 2

[8] B. Kalantari, *Alternating sign matrices and polynomiography*, Electron. J. Combin., **2011** (2011), 22 pages. 1, 2

[9] W. Kotarski, K. Gwawiec, A. Lisoska, *Polynomiography via Ishikawa and Mann terations*, **7431** (2012), 305–313. 1, 2

[10] B. Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freeman and Company, New York, (1982). 1

[11] W.R. Mann, *Mean value methods in iteration*, Proc. Amer. Math. Soc., **4** (1953), 506–510. 1

[12] H. Minc, *Nonnegative Matrices*, John Wiley & Sons Inc., New York, (1988). 3.6

[13] H. Susanto, N. Karjanto, *Newton's method's basins of attraction revisited*, Appl. Math. Comput., **215** (2009), 1084–1090. 2