symmetry

MDPI

Article

Visual Analysis of Mixed Algorithms with Newton and Abbasbandy Methods Using Periodic Parameters

Safeer Hussain Khan [1], Lateef Olakunle Jolaoso [2,3,*] and Maggie Aphane [3]

1 Abdus Salam School of Mathematical Sciences, Government College University, Lahore 54000, Pakistan
2 School of Mathematical Sciences, University of Southampton, Southampton SO17 3AS, UK
3 Department of Mathematics and Applied Mathematics, Sefako Makgatho Health Sciences University, P.O. Box 94, Medunsa 0204, South Africa
* Correspondence: lateef.jolaoso@smu.ac.za

Abstract: In this paper, we proposed two mixed algorithms of Newton's and Abbasbandy's methods using a known iteration scheme from fixed point theory in polynomiography. We numerically investigated some properties of the proposed algorithms using periodic sequence parameters instead of the constant parameters that are mostly used by many authors. Two pseudo-Newton algorithms were introduced based on the mixed iterations for the purpose of generating polynomiographs. The properties of the obtained polynomiographs were studied with respect to their graphics, turning effects and computation time. Moreover, some of these polynomiographs exhibited symmetrical properties when the degree of the polynomial was even.

Keywords: mixed algorithms; Newton's method; Abbasbandy's method; polynomiographs; sequence parameters; periodic parameters

MSC: 65H05; 65D32

check for updates

Citation: Khan, S.H.; Jolaoso, L.O.; Aphane, M. Visual Analysis of Mixed Algorithms with Newton and Abbasbandy Methods Using Periodic Parameters. *Symmetry* **2022**, *14*, 2484. https://doi.org/10.3390/sym14122484

Academic Editors: Dongfang Li, Juan Luis García Guirao and Sergei D. Odintsov

Received: 17 July 2022
Accepted: 15 November 2022
Published: 23 November 2022

1. Introduction

Polynomiography has many applications in our everyday life, from painting to animation. It has been used to create images that have inspired several hand-painted artworks, lead to the creation of elegant carpet designs, provided a great source for the production of many tapestries and served as a useful technique in the content of both education and art in the generation of several animations [1]. Polynomiography may also be used in textile design. According to Kalantari [1], polynomiography is described as the art or science of creating graphics based on the approximation of zeros of complex polynomials. These graphical images can be fractals or nonfractals and are created using mathematical convergence properties of iteration functions. A single image created via polynomiography is referred to as a polynomiograph. Although polynomiographs belong to the same class of images as fractals (an earlier word for an image produced from iteration functions), they, however, differ in the sense that their shapes and designs can be controlled more predictably by using different iteration methods. More so, polynomiographs are sensitive to small changes in the control parameters of the iteration functions, polynomials or scales of the graphics, whereas fractals are self-similar, have a typical structure and are independent of scale. In general, polynomiography has found several applications in design, education, art and science [2–4].

According to the fundamentals of algebra, a complex polynomial p of degree n with coefficients $\{a_n, a_{n-1}, \dots, a_1\}$, which can be written as

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

has n roots (zeros), which may or may not be distinct. In a polynomiograph, the degree of the polynomial describes the number of basins of attraction the polynomial has. In addition, the localization of the basins is controlled by placing roots on the complex plane manually. Polynomiographs are colored based on the number of iterations needed to obtain the approximation of the root within a chosen accuracy. One of the most popular methods for finding the roots of a polynomial p is Newton's method. This method has served as a motivation for the introduction of several generalized methods for finding roots of polynomials in real and complex domains. Recently, polynomiographs have been created using the modified Newton's method together with several iterations from the theory of fixed points. A major contribution towards the development of pseudo-Newton algorithms for locating the maximum modulus of a polynomial on the unit disk is attributed to Kalantari [5]. Gdawiec et al. [6–9] modified and studied Kalanrari's pesudo-Newton method with other known methods for approximating fixed points, resulting in new algorithms to solve the local maxima of polynomial modulus problems. Instead of using the standard Picard iteration, they used several different iterations, such as the Mann, Ishikawa, Noor, S and SP iterations. Many other authors have proposed several polynomiographs obtained using generalized versions of Newton's method with different iteration processes, for instance, [10,11] and references therein. Moreover, a survey of stability analysis for solving systems of nonlinear equations can be found in [12,13].

In this paper, we introduced two mixed algorithms consisting of Newton's method and Abbasbandy's method for generating polynomiographs. We modified the mixed processes by using a known iterative scheme that was used for finding the common fixed points of two nonlinear mappings. We also investigated some properties of these algorithms by using periodic sequence parameters instead of constant parameters, which have been mostly used by many authors. Using these processes, we obtained several polynomiographs with interesting properties from an artistic and aesthetic point of view.

2. Iterative Methods

In this section, some iterative methods for finding the fixed points of a mapping are recalled. Let (X, d) be a complete metric space and $T : X \to X$ be a mapping. A point $x \in X$ is called a fixed point T if $Tx = x$. For approximating the fixed point of T, the following iterative methods are well known. Take $x_0 \in X$ as a starting point, then we have the following:

(i) The standard Picard iteration [14] is defined by

$$z_{n+1} = Tz_n, \quad n \geq 0; \tag{1}$$

(ii) The Mann iteration [15] is given by

$$z_{n+1} = (1 - \xi_n)z_n + \xi_n Tz_n, \quad n \geq 0,$$

where $\xi_n \in (0, 1]$;

(iii) The Ishikawa iteration [16], which is a two-step process, is given by

$$\begin{cases} z_{n+1} = (1 - \xi_n)z_n + \xi_n Ty_n, \\ y_n = (1 - \eta_n)z_n + \eta_n Tz_n, \quad n \geq 0, \end{cases}$$

where $\xi_n \in (0, 1]$ and $\eta_n \in [0, 1]$;

(iv) The Picard–Mann iteration of Khan [17] is given as

$$\begin{cases} z_{n+1} = Ty_n \\ y_n = (1 - \xi_n)z_n + \xi_n Tz_n, \quad n \geq 0, \end{cases}$$

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

In the celebrated Banach fixed-point theorem [14], the standard Picard iteration is used to ensure the existence of a fixed point of a contraction mapping T. The Mann and Ishikawa iterations allow the weakening of the assumption on the mapping T to be nonexpansive. It is easy to see that the Ishikawa iteration with $\eta_n = 0$ is a Mann iteration, and for $\eta_n = 0, \xi_n = 1$ it is a Picard iteration. The Mann iteration with $\xi_n = 1$ is a Picard iteration. In addition, the Picard–Mann iteration with $\xi_n = 0$ is the Picard iteration.

Recently, Khan et al. [18] introduced a new iteration for finding common fixed points of two mappings $S, T : X \to X$, as follows: given $x_0 \in X$,

$$\begin{cases} x_{n+1} = Ty_n \\ y_n = S((1-\xi_n)w_n + \xi_n T w_n) \\ w_n = (1-\eta_n)x_n + \eta_n S x_n, \quad n = 0, 1, 2, \dots, \end{cases} \tag{2}$$

where $\{\xi_n\}$ and $\{\eta_n\}$ are sequences in $(0, 1)$. It is easy to see that (2) is more general than the Picard, Mann, Ishikawa and Picard–Mann iterations. In the sequel, we take the space $X = \mathbb{C}$, which is clearly a Banach one, $z_0 = (x_0, y_0) \in \mathbb{R}^2$ and $\xi_n = \xi$, $\eta_n = \eta$ such that $0 < \xi \leq 1$ and $0 \leq \eta \leq 1$.

Mixed Algorithms

Let $p(z)$ be a complex polynomial. Newton's method for finding the roots of p is given by the formula

$$z_{n+1} = z_n - \frac{p(z_n)}{p'(z_n)}, \quad n = 0, 1, 2, \dots, \tag{3}$$

where $z_0 \in \mathbb{C}$ is a starting point and p' is the first derivative of p. Newton's method was introduced in the 17th century and has continued to be a very important tool in mathematics. It is well known that Newton's method converges quadratically.

Using the technique of domain decomposition, Abbasbandy [19] introduced the following generalization of Newton's method:

$$z_{n+1} = z_n - \frac{p(z_n)}{p'(z_n)} - \frac{p^2(z_n)p''(z_n)}{2(p'(z_n))^3} - \frac{p^3(z_n)p'''(z_n)}{6(p'(z_n))^4}. \tag{4}$$

It was shown that the Abbasbandy method (4) converges cubically, and, hence, has a better rate of convergence than Newton's method.

Next, we define the following mixed iteration process using (2) with Newton's and Abbasbandy's methods:

$$\begin{cases} z_{n+1} = N(v_n), \\ v_n = B((1-\eta)w_n + \eta N(w_n)), \\ w_n = (1-\xi)z_n + \eta B(z_n), \end{cases} \tag{5}$$

and

$$\begin{cases} z_{n+1} = B(v_n), \\ v_n = N((1-\eta)w_n + \eta B(w_n)), \\ w_n = (1-\xi)z_n + \xi N(z_n), \end{cases} \tag{6}$$

where $n \in \mathbb{N}$, $z_0 \in \mathbb{C}$,

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

and

$$B(z) = z - \frac{p(z)}{p'(z)} - \frac{p^2(z)p''(z)}{2(p'(z))^3} - \frac{p^3(z)p'''(z)}{6(p'(z))^4}.$$

If the sequence $\{z_n\}$ (the orbit of point z_0) converges to a root z^* of p, then we say that z_0 is attracted to z^*. The set of all starting points z_0, for which $\{z_n\}$ converges to z^*, is called the basin of attraction of z^*. The boundaries among basins are usually fractals in nature. The above mixed iteration processes are convergent to the roots of the polynomial p. More so, the speed of convergence is different and the polynomiographs created by each process look different. The application of each iteration perturbs the shape of the polynomiographs, which is interesting from an aesthetic point of view.

3. Escape Criterion Results

In this section, we present some escape criterion for the mixed algorithm presented in (5) and (6). We used the general polynomial $z^n + c$ and $z^n + az + r$ to repsent Newton's and Abbassbany's methods, respectively. In the sequel, we took $z_0 \in \mathbb{C}$ and $\xi, \eta \in (0, 1]$. The escape criterion is important in the analysis of Julia sets, Mandelbrot sets and their generalizations.

The following theorem describes the escape criterion for (5).

Theorem 1. *Let $N_c(z) = z_n + c$ and $B_{a,r}(z) = z_n + az + r$. Suppose*

$$|z| \geq |r| > \frac{2(1 + |a|)}{\xi|r^{n-2}|} \quad \text{and} \quad |z| \geq |c| > \frac{2}{\eta|c^{n-2}|},$$

where $\xi, \mu \in (0, 1]$ and $r, c, a \in \mathbb{C}$. Define $\{z^n\}$ by the iteration in (5). Then, $|z^n| \to \infty$ as $n \to \infty$.

Proof. From (5), we consider

$$\begin{aligned}
|w| &= |(1 - \xi)z + \xi B_{a,r}(z)| \\
&= |(1 - \xi)z + \xi(z^n + az + r)| \\
&\geq |(1 - \xi)z + \xi(z^n + az)| - \xi|r| \\
&\geq |(1 - \xi)z + \xi(z^n + az)| - \xi|z| \\
&\geq \xi|z^n| - (1 - \xi + \xi|a|)|z| - \xi|z| \\
&= \xi|z^n| - (1 + \xi|a|)|z| \\
&= |z|(\xi|z^{n-1}| - (1 + \xi|a|)).
\end{aligned}$$

Since $\xi \in (0, 1]$ and $|z| > |r|$, then

$$\begin{aligned}
|w| &\geq |z|(\xi|z||z^{n-2}| - (1 + \xi|a|)) \\
&\geq |z|(\xi|z||r^{n-2}| - (1 + |a|)) \\
&= |z|(1 + |a|)\left(\frac{\xi|z||r^{n-2}|}{1 + |a|} - 1\right).
\end{aligned}$$

From the hypothesis of our theorem, we get

$$|z| > \frac{2(1 + |a|)}{\xi|r^{n-2}|},$$

which implies that

$$\frac{\xi|z||r^{n-2}|}{1 + |a|} - 1 > 1. \tag{7}$$

Therefore,

$$|w| > |z|.$$

Furthermore, set $u = (1-\eta)w + \eta N_C(w)$, then

$$\begin{aligned}|u| &= |(1-\eta)w + \eta N_c(w)| \\ &= |(1-\eta)w + \eta(w^n + c)| \\ &\geq |(1-\eta)w + \eta w^n| - \eta|w| \\ &\geq \eta|w^n| - (1-\eta)|w| - \eta|w| \\ &\geq |w|(\eta|w||c^{n-2}| - 1).\end{aligned}$$

In addition, since $|w| \geq |z|$, thus

$$|u| \quad \geq |z|(\eta|z||c^{n-2}| - 1).$$

Again, $|z| > \frac{2}{\eta|c^{n-2}|}$, so $\eta|z|c^{n-2} - 1 > 1$, hence,

$$|u| \geq |z|.$$

Then,

$$\begin{aligned}|v| &= |B_{a,r}(u)| \\ &= |u^n + au + r| \\ &\geq |u^n + au| - |r| \\ &\geq |u^n + au| - |u|.\end{aligned}$$

Since $\xi \in (0,1]$, then

$$\begin{aligned}|v| &\geq \xi|u^n| - |a||u| - |u| \\ &= |u|(\xi|u||u^{n-2}| - (1+|a|)).\end{aligned} \tag{8}$$

Since $|u| > |z|$, then

$$\begin{aligned}|v| &\geq |u|(\xi|u||u^{n-2}| - (1+|a|)) \\ &> |z|(\xi|z||r^{n-2}| - (1+|a|)) \\ &\geq |z|\left(\frac{\xi|z||r^{n-2}|}{1+|a|} - 1\right).\end{aligned}$$

Hence (7), we get

$$|v| \geq |z|.$$

More so, from (5), we obtain

$$\begin{aligned}|z_1| &= |N_c(v)| \\ &= |v^n + c| \\ &\geq |v^n| - |c| \\ &\geq \eta|v^n| - |v| \\ &= |v|(\eta|v||c^{n-2}| - 1) \\ &\geq |z|(\eta|z||c^{n-2}| - 1).\end{aligned}$$

Since $|z| > \frac{2}{\eta|c^{n-2}|}$, which implies that $\eta|z||c^{n-2}| - 1 > 1$, then there exists a real number $\lambda > 0$ such that

$$\frac{2}{\eta|c^{n-2}|} - 1 > 1 + \lambda.$$

Therefore, $|z_1| > (1+\lambda)|z|$. Hence by induction, we have

$$|z_n| > (1+\lambda)^n|z|.$$

This means that $|z_n| \to \infty$ as $n \to \infty$. $\square$

Next, we prove the escape criterion for (6).

Theorem 2. *Let $N_c(z) = z_n + c$ and $B_{a,r}(z) = z_n + az + r$. Suppose*

$$|z| \geq |r| > \frac{2(1+|a|)}{\eta|r^{n-2}|} \quad \text{and} \quad |z| \geq |c| > \frac{2}{\xi|c^{n-2}|}, \tag{9}$$

where $\xi, \mu \in (0,1]$ and $r, c, a \in \mathbb{C}$. Define $\{z^n\}$ by the iteration in (5). Then, $|z^n| \to \infty$ as $n \to \infty$.

Proof. From (6), we get

$$\begin{aligned}
|w| &= |(1-\xi)z + \xi N_c(z)| \\
&= |(1-\xi)z + \xi(z^n + c)| \\
&\geq |(1-\xi)z + \xi z^n| - \xi|c| \\
&\geq |(1-\xi)z + \xi z^n| - \xi|z| \\
&\geq \xi|z^n| - |z| \\
&= |z|(\xi|z||c^{n-2}| - 1).
\end{aligned}$$

From (9), we obtain $|z| > \frac{2}{\xi|z||c^{n-2}|}$. Thus $\xi|z||c^{n-2}| - 1 > 1$. Hence,

$$|w| \geq |z|.$$

Similarly, put $p = (1-\eta)w + \eta B_{a,r}(w)$. Then,

$$\begin{aligned}
|p| &= |(1-\eta)w + \eta B_{a,r}(w)| \\
&= |(1-\eta)w + \eta(w^n + aw + r)| \\
&\geq |(1-\eta)w + \eta(w^n + aw)| - \eta|r| \\
&\geq |(1-\eta)w + \eta(w^n + aw)| - \eta|w| \\
&\geq \eta|w^n| - (1-\eta+\eta|a|)|w| - \eta|w| \\
&\geq \eta|w^n| - (1+\eta|a|)|w|.
\end{aligned}$$

Since $\eta \in (0,1]$, then

$$\begin{aligned}
|p| &\geq \eta|w^n| - (1+\eta|a|)|w| \\
&\geq \eta|w^n| - (1+|a|)|w| \\
&= |w|(\eta|w||w^{n-2}| - (1+|a|)).
\end{aligned}$$

In addition, since $|z| \geq |r|$, then

$$\begin{aligned}
|p| &\geq |w|(\eta|w||w^{n-2}| - (1+|a|)) \\
&\geq |w|(\eta|w||r^{n-2}| - (1+|a|)) \\
&= |w|\left(\frac{|w|\eta|r^{n-2}|}{1+|a|} - 1\right).
\end{aligned}$$

Hence by (9), we have

$$|w| > \frac{2(1+|a|)}{\eta|r^{n-2}|},$$

which implies that

$$\frac{|w||\eta||r^{n-2}|}{1+|a|} - 1 > 1. \tag{10}$$

Therefore $|p| > |w|$. Consequently,

$$\begin{aligned} |v| &= |N_c(p)| \\ &= |p^n + c| \\ &\geq |p^n| - |c| \\ &\geq \xi|p^n| - |p| \\ &= |p|(\xi|p||p^{n-2}| - 1) \\ &\geq |z|(\xi|z||c^{n-2}| - 1). \end{aligned}$$

From the hypothesis of our theorem, we have $|v| \geq |z|$. Hence,

$$\begin{aligned} |z_1| &= |B_{a,r}(v)| \\ &= |v^n + av + r| \\ &\geq |v^n + av| - |r| \\ &\geq |v^n + av| - |v|. \end{aligned}$$

Similarly since $\eta \in (0,1]$, then

$$\begin{aligned} |z_1| &\geq \eta|v^n| - |a||v| - |v| \\ &= |v|(\eta|v||v^{n-2}| - (1+|a|)) \\ &\geq |z|(\eta|z||r^{n-2}| - (1+|a|)) \\ &= |z|\left(\frac{\eta|z||r^{n-2}|}{1+|a|} - 1\right). \end{aligned}$$

From (10), we get

$$\frac{\eta|z||r^{n-2}|}{1+|a|} - 1 > 1.$$

This implies that there exists a real number $\rho > 0$ such that

$$\frac{2(1+|a|)}{\eta|r^{n-2}|} - 1 > 1 + \rho.$$

Therefore,

$$|z_1| > (1+\rho)|z|.$$

Then, by induction, we have

$$|z_n| > (1+\rho)^n|z|.$$

This means that $|z_n| \to \infty$ as $n \to \infty$. □

4. Polynomiographs Generation

4.1. Algorithms

To generate a polynomiograph, we selected a polynomial p, parameters ξ_n and η_n for the mixed algorithms, and the maximum number of iterations M. Then, for each starting point z_0 in the area $A \subset \mathbb{C}$ (the area was discretized depending on the resolution of the graphics), we used the mixed iteration processes to iterate the root of the polynomial. The iteration proceeded until the following convergence test was satisfied:

$$|z_{n+1} - z_n| < \epsilon,$$

where $\epsilon > 0$ is the accuracy of computation, or the maximum number of iterations was reached. Finally, when the iteration process stopped, we determined a color for the starting point by mapping the number n of performed iterations to a color in the color map. The algorithms for generating the polynomiographs were presented as follows.

4.2. Visualization of Polynomiographs

In this subsection, we present some polynomiographs, which were obtained by using Algorithms 1 and 2. Furthermore, we compare the performance of the algorithms using the time taken by each algorithm to generate a polynomiograph.

Algorithm 1: Mixed Algorithm 1 for generating polynomiography

Input: Choose

$$\begin{array}{ccc} p \in \mathbb{C}[\mathbb{Z}] & -- & \text{polynomial of degree } \geq 2, \\ z_0 \in \mathbb{C} & -- & \text{starting point,} \\ \xi_n \in (0,1], \eta_n \in [0,1] & -- & \text{parameters of mixed iteration,} \\ M & -- & \text{maximum number of iteration,} \\ A \in \mathbb{C} & -- & \text{area for the polynomiographs.} \end{array} \tag{11}$$

While: $n \leq M$, compute

$$\begin{cases} w_n = (1-\xi_n)z_n + \eta_n B(z_n) \\ v_n = B((1-\eta_n)w_n + \eta_n N(w_n)) \\ z_{n+1} = N(v_n) \end{cases} \tag{12}$$

Stopping criterion: $|z_{n+1} - z_n| < \epsilon$.
Output: color c of z_0.

Algorithm 2: Mixed Algorithm 2 for generating polynomiography

Input: Choose

$$\begin{array}{ccc} p \in \mathbb{C}[\mathbb{Z}] & -- & \text{polynomial of degree } \geq 2, \\ z_0 \in \mathbb{C} & -- & \text{starting point,} \\ \xi_n \in (0,1], \eta_n \in [0,1] & -- & \text{parameters of mixed iteration,} \\ M & -- & \text{maximum number of iteration,} \\ A \in \mathbb{C} & -- & \text{area for the polynomiographs.} \end{array} \tag{13}$$

While: $n \leq M$, compute

$$\begin{cases} w_n = (1-\xi_n)z_n + \eta_n N(z_n) \\ v_n = N((1-\eta_n)w_n + \eta_n B(w_n)) \\ z_{n+1} = B(v_n) \end{cases} \tag{14}$$

Stopping criterion: $|z_{n+1} - z_n| < \epsilon$.
Output: color c of z_0.

Example 1. *In the first example, we used six various polynomials to generate different polynomiographs with Algorithms 1 and 2. We chose* $\xi_n = \frac{1}{n+1}, \eta_n = \frac{2n}{5n^2+8}, M = 30, \epsilon = 0.0001$ *and*

$A = [-2, 2]^2$. The control parameter was chosen arbitrarily such that $\xi_n, \eta_n \in [0, 1]$. In particular, we considered the following polynomials:

Case I $p(z) = z^3 + 1$;
Case II $p(z) = 2z^4 - 3z^3 + z^2 + 2$;
Case III $p(z) = z^4 + 2z - 1$;
Case IV $p(z) = z^8 - 1$;
Case V $p(z) = z^{12} + 2z - 3$;
Case VI $p(z) = z^{17} + 8z^8 + 2z^7 - z - 1$.

The polynomiographs generated by Algorithm 1 are shown in Figure 1 and those generated by Algorithm 2 are shown in Figure 2. More so, the graph of time of execution against the number of examples is shown in Figure 3.

From Figures 1 and 2, we see that both algorithms were able to generate distinct polynomiographs, which were artistically appealing. Figure 3 shows that the average time of execution for Algorithm 1 was less than the average time of execution for Algorithm 2.

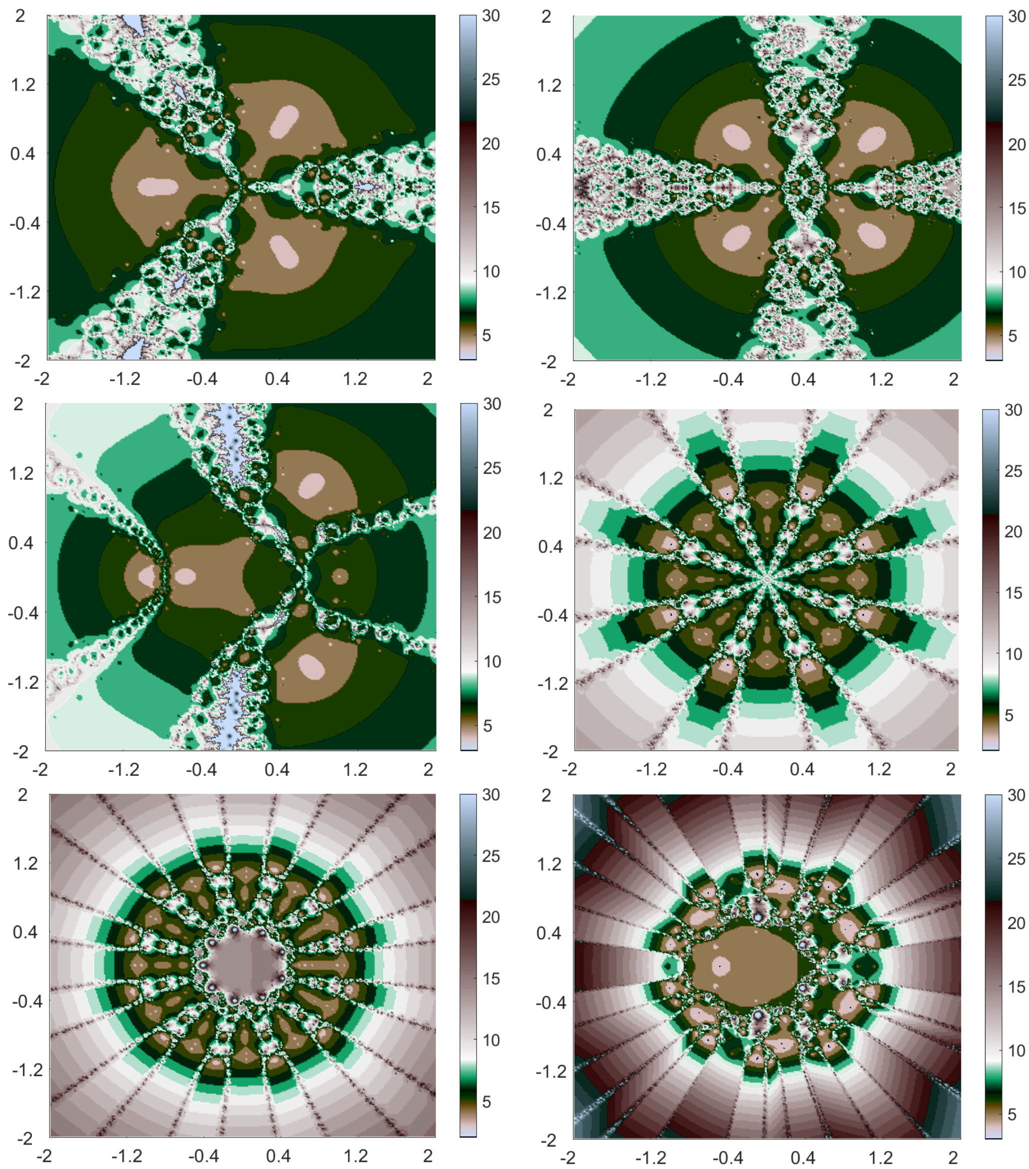

Figure 1. (Example 1) Visualization of polynomiographs by Algorithm 1: Case I (**top left**); Case II (**top Right**); Case III (**middle left**); Case IV (**middle right**); Case V (**bottom left**) and Case VI (**bottom right**).

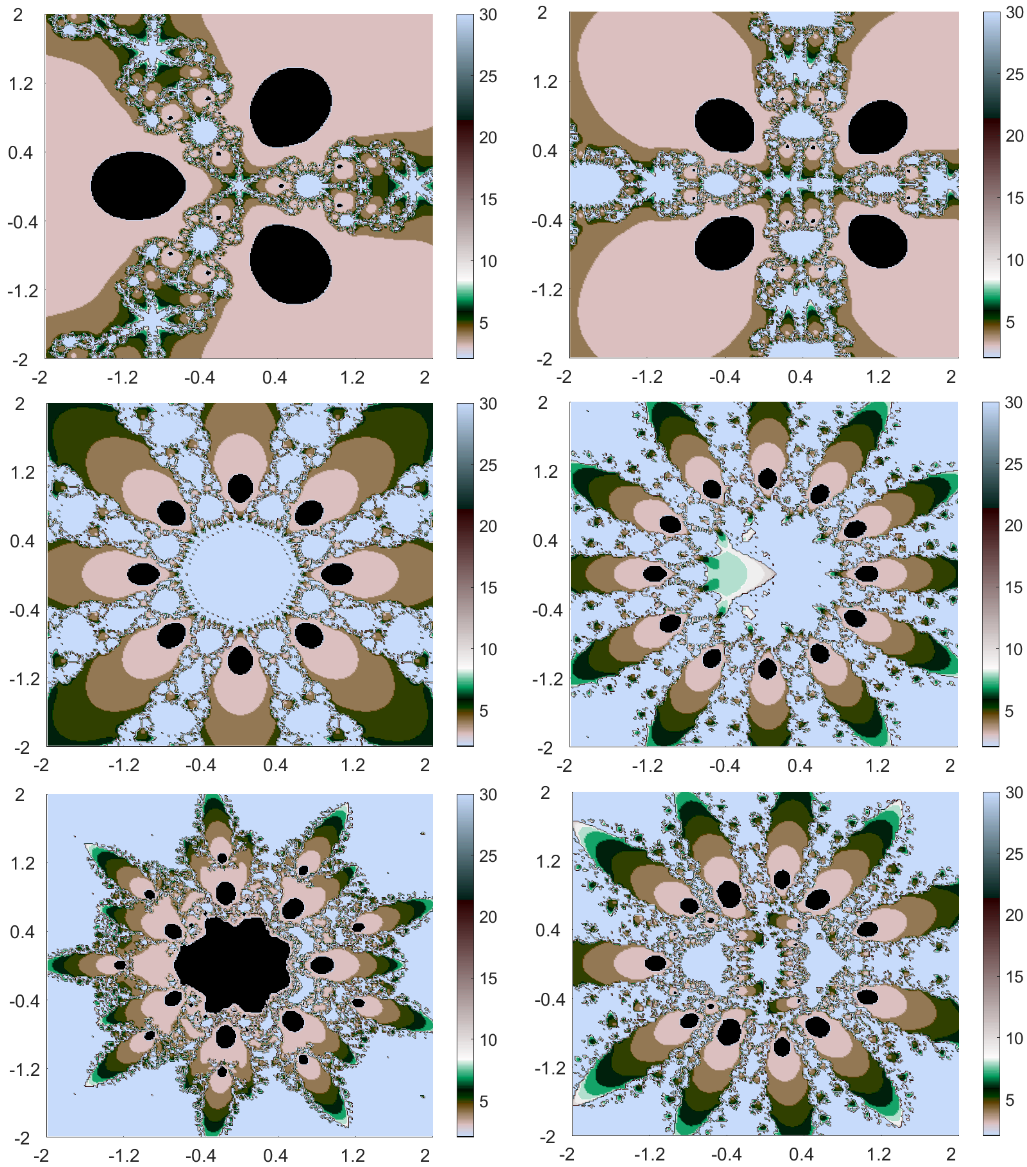

Figure 2. (Example 1) Visualization of polynomiographs by Algorithm 2: Case I (**top left**); Case II (**top right**); Case III (**middle left**); Case IV (**middle right**); Case V (**bottom left**) and Case VI (**bottom right**).

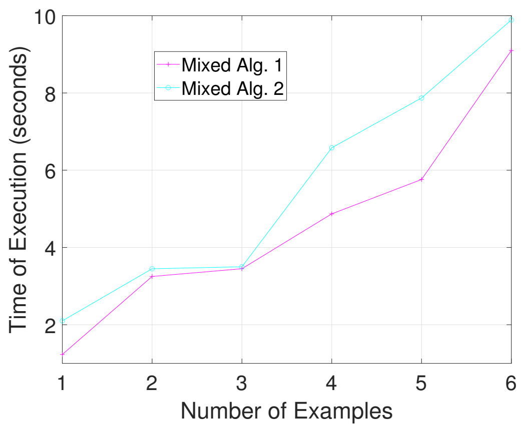

Figure 3. Graph of time of execution against the number of examples (in ascending order) for Example 1.

Example 2. *For the second example, we used periodic control parameter* $\xi_n = \sin(2n), \eta_n = \cos(n+1)$ *together with the following polynomials to generate our polynomiographs:*

Case I: $p(z) = z^3 - 5z^2 - 2z + 1$;
Case II: $p(z) = 10z^5 - 2z^4 + 3\pi z + 1$;
Case III: $p(z) = -5z^6 + 7z^3 + 2z^2 + z - 10$;
Case IV: $p(z) = -50z^8 + 5z^7 + 6z^6 + 7z^5 + 3z^3 - 7z^2 + 4z - 10$;
Case V: $p(z) = z^{12} - 1$;
Case VI: $p(z) = 20z^{12} + 9z^3 + 2z + 3$.

Similar to Example 1, the mixed algorithms were used for generating distinct polyomiographs. The polynomiographs generated by Algorithm 1 are shown in Figure 4 and the polynomiographs generated by Algorithm 2 are shown in Figure 5.The time of execution against the number of cases is shown in Figure 6. In this case, the average time of execution for Algorithm 2 was less than the average time of execution for Algorithm 1.

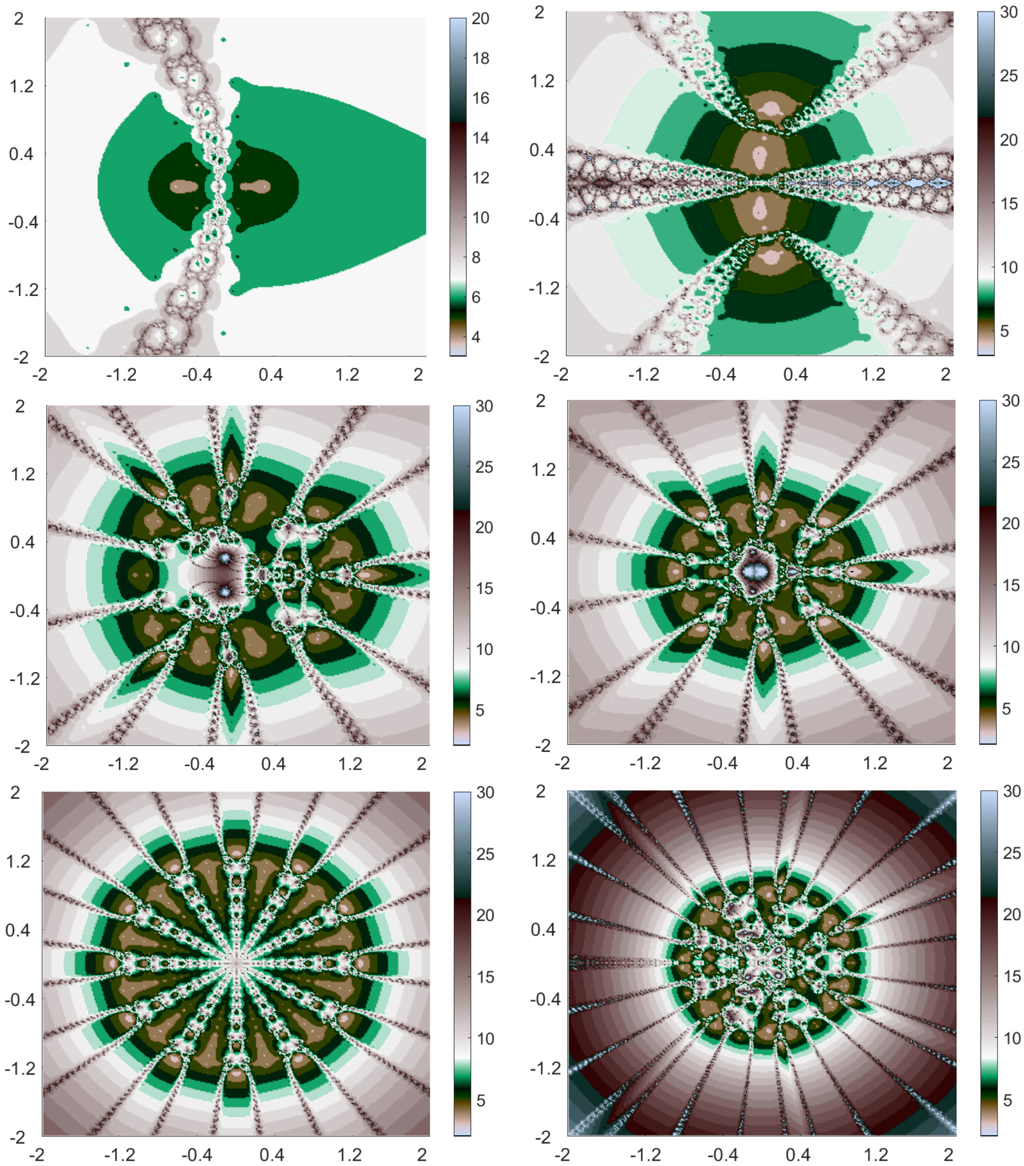

Figure 4. (Example 2) Visualization of polynomiographs by Algorithm 1: Case I (**top left**); Case II (**top right**); Case III (**middle left**); Case IV (**middle right**); Case V (**bottom left**) and Case VI (**bottom right**).

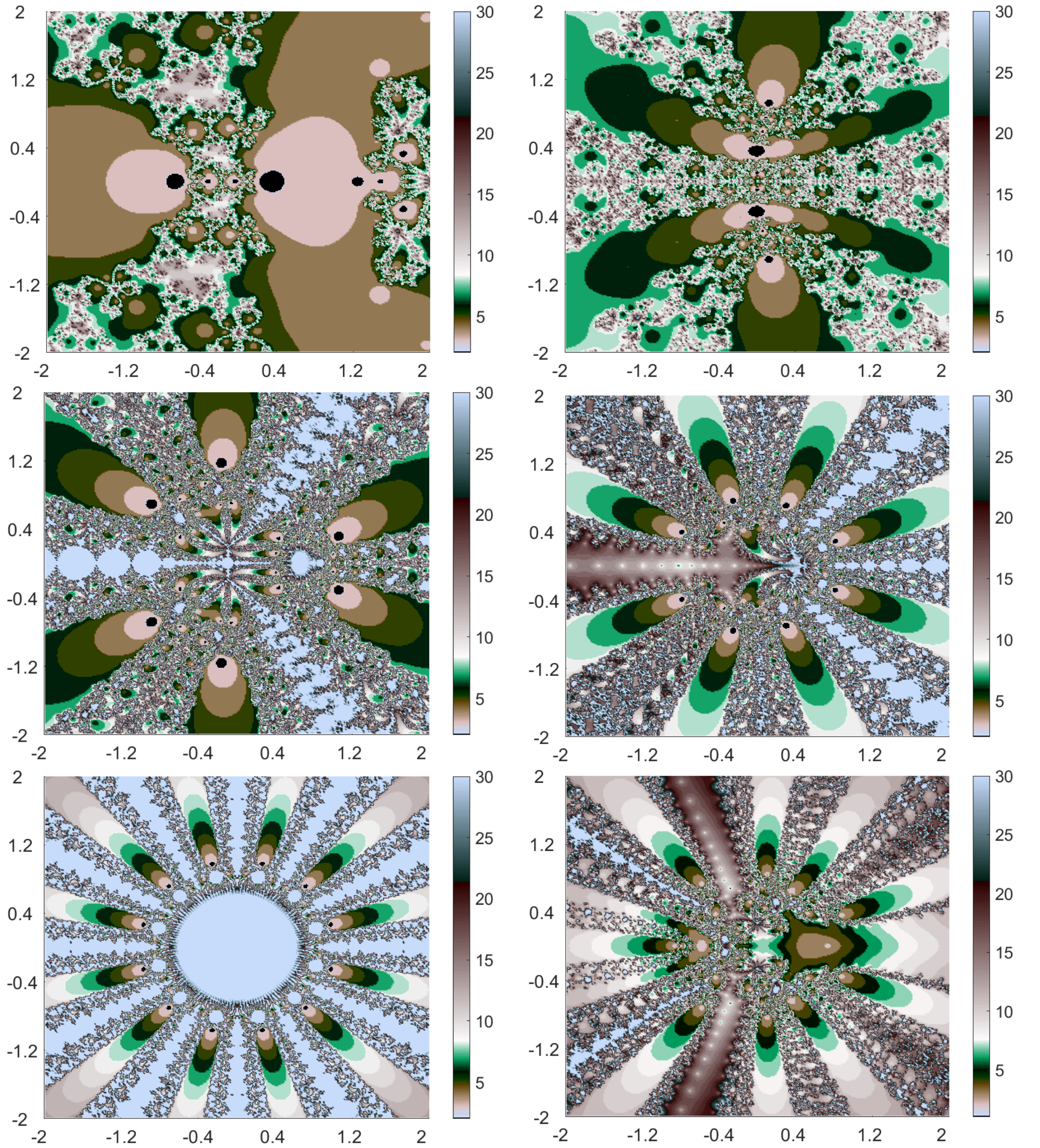

Figure 5. (Example 2) Visualization of polynomiographs by Algorithm 2: Case I (**top left**); Case II (**top right**); Case III (**middle left**); Case IV (**middle right**); Case V (**bottom left**) and Case VI (**bottom right**).

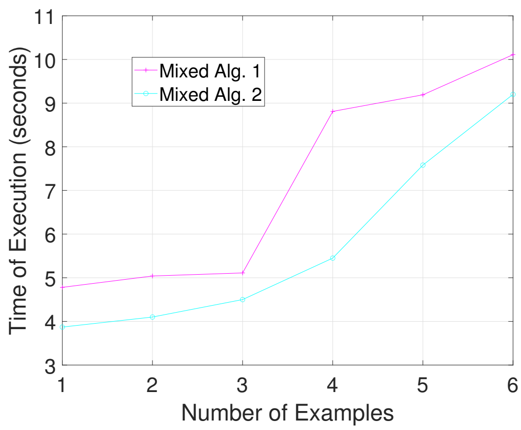

Figure 6. Graph of time of execution against the number of examples (in ascending order) for Example 2.

5. Conclusions and Future Work

In this paper, we introduced two new mixed algorithms, which consisted of Newton's and Abbasbandy's methods with a known iteration process in fixed point theory. We investigated some properties of the algorithms while generating polynomiographs using periodic parameters. The use of periodic parameters gave the obtained polynomiographs more interesting patterns. More so, numerical results showed that the change in the parameters did not have any significant effect on the convergence of the algorithms. These results were new and interesting from an artistic point of view.

The results of this paper could be further modified by using other known methods for approximating the roots of complex polynomials. In addition, the mixed algorithms could be modified by using other iteration methods from fixed point theory. We could also define different periodic functions for the parameters, as well as measure the comparison of the mixed algorithms using the mean number of iterations and the convergence area index.

Author Contributions: Conceptualization, S.H.K.; methodology, L.O.J.; validation, S.H.K., M.A. and L.O.J.; formal analysis, L.O.J.; writing—original draft preparation, L.O.J. and S.H.K.; writing—review and editing, L.O.J., S.H.K. and M.A.; visualization, L.O.J.; supervision, M.A.; project administration, M.A.; funding acquisition, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Sefako Makgatho Health Sciences University Postdoctoral research fund, and the APC was funded by the Department of Mathematics and Applied Mathematics, Sefako Makgatho Health Sciences University, Pretoria, South Africa.

Data Availability Statement: Not applicable.

Acknowledgments: The authors appreciated the assistance of Professor Krzystof Gdawiec on MATLAB programming for Polynomiography. L.O. Jolaoso is supported by the Postdoctoral Fellowship Research Grant at the Sefako Makgatho Health Sciences University, Ga-Rankuwa, Pretoria, South Africa.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript or in the decision to publish the results.

References

1. Kalantari, B. Two and three-dimensional art inspired by polynomiography. In *Renaissance Banff: Mathematics, Music, Art, Culture, Bridges Conference*; Sarhangi, R., Moody, R., Eds.; Southwestern College: Winfield, KS, USA, 2005; pp. 321–328.
2. Jolaoso, L.O.; Khan, S.H. Some Escape Time Results for General Complex Polynomials and Biomorphs Generation by a New Iteration Process. *Mathematics* **2020**, *8*, 2172. [CrossRef]
3. Kalantari, B. Polynomiography: From the fundamental theorem of Algebra to art. *Leonardo* **2005**, *38*, 233–238. [CrossRef]
4. Kalantari, B. Polynomiography and applications in art, education and science. *Comput. Graph.* **2004**, *28*, 417–430. [CrossRef]
5. Kalantari, B. A Necessary and Suffcient Condition for Local Maxima of Polynomial Modulus Over Unit Disc. *arXiv* **2016**, arXiv:1605.00621.
6. Gdawiec, K. Fractal patterns from the dynamics of combined polynomial root finding methods. *Nonlinear Dyn.* **2017**, *90*, 2457–2479. [CrossRef]
7. Gdawiec, K.; Kotarski, W. Polynomiography for the polynomial infinity norm via Kalantari's formula and nonstandard iterations. *Appl. Math. Comput.* **2017**, *307*, 17–30. [CrossRef]
8. Gdawiec, K.; Kotarski, W.; Lisowski, A. Polynomiography based on the nonstandard Newton-like root finding methods, *Abstr. Appl. Anal.* **2015**, *2015*, 797594.
9. Kotarski, W.; Gdawiec, K.; Lisowska, A. Polynomiography via Ishikawa and Mann iterations. In Proceedings of the International Symposium on Visual Computing, Crete, Greece, 16–18 July 2012; Springer: Berlin, Germany, 2012; pp. 305–313.
10. Ardelean, G. Comparison between iterative methods by using the basins of attraction. *Appl. Math. Comput.* **2011**, *218*, 88–95. [CrossRef]
11. Gdawiec, K.; Shahid, A.A.; Nazeer, W. Higher order methods of the Basic Family of Iterations via S-iteration with s -convexity. *Mediterr. J. Math.* **2020**, *17*, 43. [CrossRef]
12. Amiri, A.; Cordero, A.; Darvishi, M.T.; Torregrosa, J.R. Stability analysis of a paramteric family of seventh-order iterative methods for solving nonlinear systems. *Appl. Math. Comput.* **2018**, *323*, 43–57.

13. Amiri, A.; Cordero, A.; Darvishi, M.T.; Torregrosa, J.R. Stability analysis of Jacobian-free iterative methods for solving nonlinear systems by using families of mth power divided differences. *J. Math. Chem.* **2019**, *57*, 1344–1373. [CrossRef]
14. Banach, S. Sur les opérations dan les ensembles abstracts ey lear applications aux équations integrales. *Fund. Math* **1922**, *3*, 133–181. [CrossRef]
15. Mann, W. Mean value methods in iteration. *Proc. Am. Math. Soc.* **1953**, *4*, 506–510. [CrossRef]
16. Ishikawa, S. Fixed points by a new iteration method. *Proc Am. Math Soc.* **1974**, *44*, 147–150. [CrossRef]
17. Khan, S.H. A Picard-Mann hybrid iterative process. *Fixed Point Theory Appl.* **2013**, *2013*, 69. [CrossRef]
18. Garodia, C.; Uddin, I.; Khan, S.H. Approximating common fixed point by a new faster iteration. *Filomat* **2020**, *34*, 2047–2060. [CrossRef]
19. Abbasbandy, S. Improving Newton-Raphson method for nonlinear equations by modified Adomain decomposition method. *Appl. Math. Comput.* **2003**, *145*, 887–893.