

Visual Analysis of the Newton's Method with Fractional Order Derivatives

Krzysztof Gdawiec , Wiesław Kotarski  and Agnieszka Lisowska 

Institute of Computer Science, University of Silesia, Będzińska 39, 41-200 Sosnowiec, Poland;
kotarski@ux2.math.us.edu.pl (W.K.); alisow@ux2.math.us.edu.pl (A.L.)

* Correspondence: krzysztof.gdawiec@us.edu.pl

Received: 28 July 2019; Accepted: 5 September 2019; Published: 9 September 2019



Abstract: The aim of this paper is to investigate experimentally and to present visually the dynamics of the processes in which in the standard Newton's root-finding method the classic derivative is replaced by the fractional Riemann–Liouville or Caputo derivatives. These processes applied to polynomials on the complex plane produce images showing basins of attractions for polynomial zeros or images representing the number of iterations required to obtain polynomial roots. These latter images were called by Kalantari as polynomiographs. We use both: the colouring by roots to present basins of attractions, and the colouring by iterations that reveal the speed of convergence and dynamic properties of processes visualised by polynomiographs.

Keywords: fractional derivative; Newton method; root-finding; polynomiography

1. Introduction

Polynomials and the problem of finding their roots are present in both mathematics and engineering. From the history of mathematics, one can learn that already Sumerians in 3000 B.C. and ancient Greeks considered practical problems that can be stated in modern mathematical language as finding roots of polynomials. Newton, in the 17th century, proposed the method for calculating polynomials' zeros approximately. In 1879, Cayley discovered strange and unpredictable chaotic behaviour when applying Newton's method to the simple equation $z^3 - 1 = 0$ in the complex plane. An explanation of Cayley's problem was found by Julia in 1919. Julia sets inspired great discoveries—the Mandelbrot set and fractals in the 1970s [1]. Polynomiography, introduced to science in the early 2000s by Kalantari [2], was the last interesting contribution to the polynomial root-finding history. Kalantari [2,3] defined polynomiography as both the art and science related to visualisation of the root-finding process for a polynomial in complex plane. To generate a polynomiograph, any root-finding method can be used, e.g., the well-known Newton's method, the methods from The Basic or from the Euler–Schröder Families of Iterations. Polynomiographs are nicely looking images that have aesthetic values. Polynomiography as the method of aesthetic pattern generation was patented by Kalantari in the USA in 2005 [3].

In [4,5], the authors presented a survey of some modifications of Kalantari's results based on the classic Newton and the higher-order Newton-like root-finding methods and pseudo-methods for complex polynomials. The standard Picard iteration was replaced there by several different iteration processes. By the use of various kinds of iterations, various convergence criteria, and colouring methods, a great variety of polynomiographs were obtained in [5]. In [6–8], the authors used the methods of polynomiography to visually analyse the behaviour of various roots finding methods, whereas, in [6,9,10],

the authors investigated root-finding methods by using different numerical measures, e.g., average number of iterations or convergence area index.

Recently, Brambila-Paz et al. proposed some modification of the Newton method for root finding of polynomial equations in $\mathbb{R}$ [11,12]. Instead of the classical derivative, a fractional α -order Riemann–Liouville derivative was used. The modified root-finding method is able to find a root with complex part, if it exists, by starting from an initial point lying on the real line. By changing the α -order of the derivative and the starting point on reals the new root-finding method finds all polynomial zeros. Brambila-Paz et al. [12] proved that, for functions with roots of multiplicity one, fractional Newton method converges at least linearly when $\alpha \neq 1$. Inspired by this new idea, the authors of this paper have carried out some investigations concerning polynomiographs generated by methods with fractional order derivatives.

The paper is organised as follows. Section 2 presents some basic information related to fractional derivatives. In Section 3, formulas for the Newton’s method with fractional Riemann–Liouville and Caputo derivatives for complex polynomials are given. Then, in Section 4, some examples of polynomiographs and results of numerical experiments are presented. Section 5 concludes the paper and shows the future directions of research.

2. Fractional Derivatives

The idea of fractional derivatives first appeared in Leibnitz’s letter to L’Hospital in 1695, in which the notion for differentiation of non-integer order is discussed. Integer order derivatives and integrals, oppositely to fractional order derivatives, have clear physical interpretation and are used for describing different concepts in classical physics, e.g., velocity and area. Mathematicians Liouville, Riemann, and Caputo have done major work on fractional calculus. The first practical application of $\frac{1}{2}$ -order derivative appeared in the tautochrone problem [13], in which the shape of a curve was found such that the time of descent of a frictionless point mass sliding along the curve under the gravity force is independent of the starting point. Fractional Calculus became a useful mathematical tool for applied sciences and is attracting more and more attention. Fractional-order models can describe reality more accurately in comparison to classical integer-order models. Many new applications can be found in a variety of fields such as physics, chemistry, biology, economics, signal and image processing, control, porous media, and aerodynamics, as reported in a recently published survey paper ([14] and references therein). Fractional calculus and its applications are presented in many books and papers [15–19].

Integer-order derivative and integral are uniquely determined in the classical analysis. The same is for fractional integral. However, the situation for fractional-order derivatives is more complicated. There are many different definitions for them. In this paper, we use two the most popular definitions: Riemann–Liouville and Caputo. Following the authors of [18,20], we recall their basic properties. We need the special function Γ that is a generalisation of the factorial function $n!$, i.e., $\Gamma(n) = (n-1)!$ for $n \in \mathbb{N}$. For complex arguments with positive real part, it is defined as:

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt, \quad \operatorname{Re}(z) > 0. \quad (1)$$

By analytical continuation, this function can be extended to the whole complex plane except for the points $0, -1, -2, -3, \dots$, where it has single poles.

Some of the most important properties of Γ are the following:

- $\Gamma(1) = \Gamma(2) = 1$,
- $\Gamma(z+1) = z\Gamma(z)$,
- $\Gamma(n) = (n-1)!$ for $n \in \mathbb{N}$,
- $\Gamma(1/2) = \sqrt{\pi}$.

In the next two subsections, Riemann–Liouville and Caputo derivatives are defined with a value 0 in the lower limits of integrals.

2.1. The Riemann–Liouville Derivative

The Riemann–Liouville derivative (RL derivative) of order $\alpha \in (n-1, n]$, $n \in \mathbb{N}$, of a real-valued function f , is defined as:

$$D_{RL}^{\alpha} f(t) := \begin{cases} \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_0^t \frac{f(\tau)}{(t-\tau)^{\alpha+1-n}} d\tau, & \text{if } \alpha \in (n-1, n), \\ \frac{d^n}{dt^n} f(t), & \text{if } \alpha = n. \end{cases} \quad (2)$$

The Riemann–Liouville fractional derivative operation is a linear one, thus has the interpolation property, i.e.,

$$\lim_{\alpha \rightarrow n} D_{RL}^{\alpha} f(t) = f^{(n)}(t), \quad (3)$$

$$\lim_{\alpha \rightarrow n-1} D_{RL}^{\alpha} f(t) = f^{(n-1)}(t), \quad (4)$$

and for a constant function $f(t) = c$ and $\alpha \neq 1$ it is not 0 and equals:

$$D_{RL}^{\alpha} c = \frac{c}{\Gamma(1-\alpha)} t^{-\alpha} \neq 0. \quad (5)$$

For a power function, the Riemann–Liouville α -fractional order derivative is defined as follows:

$$D_{RL}^{\alpha} t^m = \frac{\Gamma(m+1)}{\Gamma(m-\alpha+1)} t^{m-\alpha}, \quad (6)$$

where $\alpha \in (n-1, n)$, $n \in \mathbb{N}$ and $m \in \mathbb{R}$ is such that $m > n-1$.

2.2. The Caputo Derivative

The Caputo derivative (C derivative) of order $\alpha \in (n-1, n]$, $n \in \mathbb{N}$, of a real-valued function f , is defined as:

$$D_C^{\alpha} f(t) := \begin{cases} \frac{1}{\Gamma(n-\alpha)} \int_0^t \frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha+1-n}} d\tau, & \text{if } \alpha \in (n-1, n), \\ \frac{d^n}{dt^n} f(t), & \text{if } \alpha = n. \end{cases} \quad (7)$$

The Caputo fractional derivative operation is a linear one, thus has the interpolation property, i.e.,

$$\lim_{\alpha \rightarrow n} D_C^{\alpha} f(t) = f^{(n)}(t), \quad (8)$$

$$\lim_{\alpha \rightarrow n-1} D_C^{\alpha} f(t) = f^{(n-1)}(t) - f^{(n-1)}(0), \quad (9)$$

and for a constant function it equals 0.

For a power function, the Caputo α -fractional order derivative is defined as follows:

$$D_C^{\alpha} t^m := \begin{cases} \frac{\Gamma(m+1)}{\Gamma(m-\alpha+1)} t^{m-\alpha} = D_{RL}^{\alpha} t^m, & \text{if } m > n-1, \\ 0, & \text{if } m \leq n-1, \end{cases} \quad (10)$$

where $\alpha \in (n-1, n)$, $n \in \mathbb{N}$ and $m \in \mathbb{R}$.

For example, we have:

$$\begin{aligned} D_C^\alpha t &= \frac{1}{\Gamma(2-\alpha)} t^{1-\alpha}, \text{ if } \alpha < 1, \\ D_C^\alpha t^2 &= \frac{2}{\Gamma(3-\alpha)} t^{2-\alpha}, \text{ if } \alpha < 2, \\ D_C^\alpha t^3 &= \frac{6}{\Gamma(4-\alpha)} t^{3-\alpha}, \text{ if } \alpha < 3, \\ D_C^\alpha t^4 &= \frac{24}{\Gamma(5-\alpha)} t^{4-\alpha}, \text{ if } \alpha < 4. \end{aligned}$$

In general, Riemann–Liouville and Caputo fractional derivatives do not coincide, i.e., $D_{RL}^\alpha f(t) \neq D_C^\alpha f(t)$. They are equal if the function f is such that $f^{(s)}(0) = 0$ for all $s \in \{0, 1, \dots, n-1\}$.

For integer-order differentiation, the classical derivative depends only on the point t together with its close neighbourhood, where it is calculated. However, the fractional derivative depends on the graph of the function f in $[0, t]$. Thus, we can look at the behaviour of integer-order derivative as local, whereas of fractional-order derivative as non-local [21], such as for an integral operator. Moreover, fractional-order derivative depends on the choice of an initial point (in Equations (2) and (7), the initial point was set to 0). The above-mentioned properties explain why fractional calculus is used for modelling processes with memory and hereditary effects. It is worth recalling the following example: if $x(t)$ denotes a position of some mass at time t , then the fractional velocity $v(t)$ and the fractional acceleration $a(t)$ can be defined as: $v(t) = D^\alpha x(t)$ and $a(t) = D^\alpha v(t)$ with some α -order derivatives. Thus, it is possible to construct classical mechanics with fractional derivatives as in [22].

2.3. Complex Plane Derivatives

It is interesting that the fractional calculus was developed in the complex plane from the very beginning. For analytic functions represented by exponential series (Liouville's approach) or power series (Hadamard's approach), their integro-differentials are based on a termwise integro-differentiation of the series and the following formulas

$$D^\alpha e^{az} = a^\alpha e^{az} \quad (11)$$

or

$$D^\alpha (z - z_0)^m = \frac{\Gamma(m+1)}{\Gamma(m-\alpha+1)} (z - z_0)^{m-\alpha}, \quad (12)$$

respectively, where α, a , and m are arbitrary numbers [18].

Analysis in complex plane is more complicated in comparison to analysis in real line. Thus, we cannot replace real variable x by complex variable z directly under integrals in Equations (2) and (7). The difficulty is related to multi-valuedness of expressions that are under integrals in RL and C fractional derivatives. Notice that, for example, expression z^α , where α is a fraction, is multivalued. Thus, to define a function, one has to fix a branch cut line and choose a branch (Riemann surface). A common choice of the branch cut line is the negative real half-axis. We take the principal branch as a single-valued function that is continuous above the branch cut. In general, branch cut can be done in many ways. More details on branch cuts can be found, for instance, in [23–25].

In this paper, we are interested in fractional derivatives for a specific class of complex functions, namely in polynomials. Thus, we can define RL α -order fractional derivative for a monomial:

$$D_{RL}^\alpha z^m = \frac{\Gamma(m+1)}{\Gamma(m-\alpha+1)} z^{m-\alpha}, \quad (13)$$

where $z \in \mathbb{C} \setminus \{c \in \mathbb{C} : \text{Im}(c) = 0 \wedge \text{Re}(c) < 0\}$, $m \neq -1, -2, -3, \dots$ and $\alpha \in (n-1, n)$ for some $n \in \mathbb{N}$.

It is easy to see that

$$D_{RL}^\alpha c = c D_{RL}^\alpha z^0 = \frac{c z^{-\alpha}}{\Gamma(1-\alpha)}, \quad (14)$$

where $c = \text{constant}$ and $\alpha \in (n-1, n)$ for some $n \in \mathbb{N}$. Because any complex polynomial p is a sum of monomials multiplied by some coefficients, Equation (13) can be applied to p , term by term, to obtain $D_{RL}^\alpha p(z)$.

For Caputo α -order fractional derivatives, an additional postulate is added:

$$D_C^\alpha c = 0, \quad (15)$$

where $c = \text{constant}$. It is easy to see that $D_C^\alpha p(z) = D_{RL}^\alpha p(z)$ if and only if p has the constant term equal to zero. Equation (13) is well-defined because the branch cuts procedure remove the multi-valuedness in it. The branch cuts procedure has been implemented in many Computer Algebra Systems (CAS) (Maple and Matlab) and programming languages (Python and Java). For example, by using FunctionAdvisor tool in Maple [26], one can check the branch cut line and visualise real and imaginary parts of $D_{RL}^{1/2}(z^3 - 1)$. As one can see in Figure 1, the plots are continuous functions excluding the cut line, $\{z \in \mathbb{C} : \text{Im}(z) = 0 \wedge \text{Re}(z) < 0\}$, along which the imaginary part of the plot has the jump.

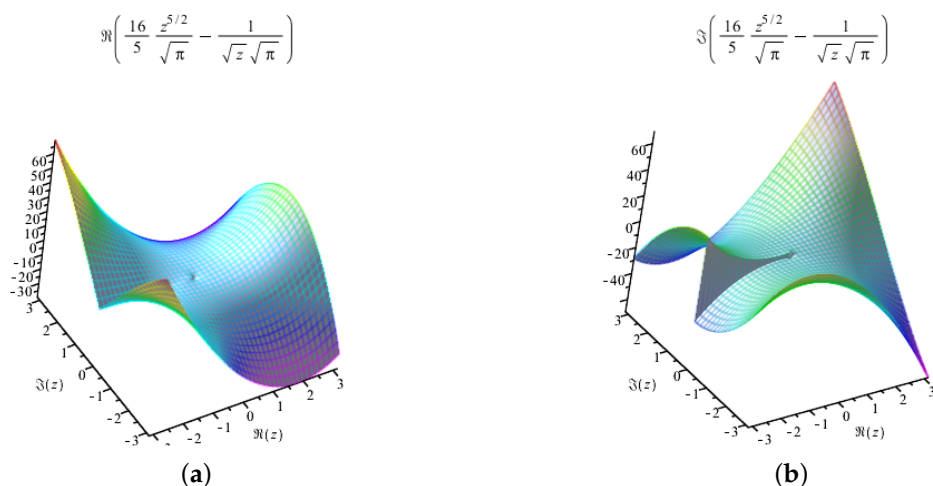


Figure 1. The plots of the (a) real and (b) imaginary parts of $\frac{1}{2}$ -order RL derivative of $z^3 - 1$.

Some general results with the detailed proofs concerning derivatives of analytic functions can be found in [18,27,28]. In [29,30], many new properties of fractional derivatives together with directional fractional derivatives are presented.

3. The Newton Method with Fractional Derivatives

Let us denote any complex polynomial by p . The well-known Newton's method is defined as

$$z_{k+1} = z_k - \frac{p(z_k)}{p'(z_k)}, \quad k = 0, 1, 2, \dots, \quad (16)$$

with a starting point $z_0 \in \mathbb{C}$. Equation (16) can be used to solve equation $p(z) = 0$ in both real and complex domains.

By replacing the classical first-order derivative $p'(z)$ in Equation (16) by the Riemann–Liouville or the Caputo derivative of fractional order α , both denoted as $D^\alpha p(z)$, we get

$$z_{k+1} = z_k - \frac{p(z_k)}{D^\alpha p(z_k)}, \quad k = 0, 1, 2, \dots, \quad (17)$$

with a starting point $z_0 \in \mathbb{C}$. Equation (17) can be used to solve equation $p(z) = 0$ in both the real and the complex domains. Further, Equation (17) for the RL derivative is called the fractional RL-Newton method and for the C derivative as the fractional C-Newton method.

At the beginning, we assumed that p is a complex polynomial. Because polynomials in the complex plane are very regular functions, as in the real case, thus the convergence properties of the fractional Newton's method follow directly from the calculations presented by Brambila-Paz et al. in [12]. All these calculations are valid in our case for both RL and C derivatives. The problem that appears is multivaluedness, which can be fixed by the branch cut method. Observe that the right hand-side of Equation (17) can be treated as an iteration function Φ :

$$\Phi(z, \alpha) = z - \frac{p(z)}{D^\alpha p(z)} \quad (18)$$

with α as a parameter. When Φ is regular enough, the first and the second derivatives can be calculated and Taylor's series expansion of the function Φ around a zero ξ of p can be performed. If Φ is not regular enough, then some regularisation with the use of integral operator is required and Taylor's series expansion is obtained for regularised Φ . Then, similar to in [12], the following conclusion can be formulated: the fractional Newton method in Equation (17) converges at least linearly when $\alpha \neq 1$ and at least quadratically when $\alpha = 1$ for polynomials with zeros of multiplicity one.

4. Numerical Experiments

In this section, the results of graphical and numerical experiments are presented. The experiments were conducted for several polynomials by using the RL-Newton and the C-Newton methods with various values of parameter $\alpha \in (0, 2)$.

The graphical examples are obtained through the methods of polynomiography. We selected polynomiographs obtained with two colouring methods, namely basins of attraction and colouring with respect to the number of iterations (the so-called iteration colouring). The general algorithm for the generation of polynomiograph is presented in Algorithm 1. In the algorithm, the *convergenceTest*($z_{k+1}, z_k, \varepsilon$) returns *TRUE* if the method has converged to the root, and *FALSE* otherwise. The two most widely used convergence tests are the following:

$$|p(z_{k+1})| < \varepsilon, \quad (19)$$

$$|z_{k+1} - z_k| < \varepsilon. \quad (20)$$

The last step of the algorithm colours the point according to the selected colouring method. In the basins of attraction colouring, each root gets a distinct colour (other than black). Next, if the number of performed iterations is less than M , then for the obtained approximation z_{k+1} the algorithm searches for the closest root—by using the modulus metric—and colours the starting point with the colour of the root. If the number of performed iterations is equal to M , then the starting point gets the black colour. In iteration colouring, the starting point is coloured by mapping the number of performed iterations on the colour in the colour map.

Algorithm 1: Polynomiograph generation.

Input: $p \in \mathbb{C}[Z]$ —polynomial, α —derivative order, $A \subset \mathbb{C}$ —area, M —the maximum number of iterations, ε —accuracy, $colourmap[0..C-1]$ —colour map with C colours.

Output: Polynomiograph for the area A .

```

1 for  $z_0 \in A$  do
2    $k = 0$ 
3   while  $k < M$  do
4      $z_{k+1} = z_k - \frac{p(z_k)}{D^\alpha p(z_k)}$ 
5     if  $convergenceTest(z_{k+1}, z_k, \varepsilon, p) = TRUE$  then
6       break
7      $k = k + 1$ 
8   colour  $z_0$  with the use of  $colourmap$ 
```

In the numerical experiments, we used three different measures to compare the use of Riemann–Liouville and Caputo fractional derivatives: average number of iterations [9], convergence area index [6], and generation time [10]. All measures were calculated from the polynomiographs.

To compute the first measure, namely the average number of iterations (ANI), we used polynomiograph P coloured with the iteration colouring. In this type of polynomiograph, each point is coloured according to the number of iterations needed to find a root. We used a linear mapping for this aim, i.e., each iteration was linearly mapped onto a colour in a given colour map. The value of ANI was computed as the mean number of iterations in P .

Similar to ANI, the convergence area index (CAI) was computed from polynomiograph P coloured by using the iteration colouring. First, we counted the number of converging points N_c , i.e., points in P whose number of iterations is less than the maximum number of iterations. Then, the value of CAI was computed in the following way

$$CAI = \frac{N_c}{N}, \quad (21)$$

where N is the total number of points in P . CAI gives us information about the percentage of the polynomiograph's area that has converged to roots. It takes values in $[0, 1]$, where 0 means that no point has converged, and 1 means that all points have converged.

The last measure is the time of generation of the polynomiograph. This measure is very similar to ANI, but it gives us information about the actual time of computations. ANI does not take into account the cost of computations in a single iteration. Thus, two methods with the same value of ANI can have different generation times due to different computational complexities in a single iteration.

Following the authors of [7,8], we used the same polynomials in the experiments, namely

$$p_n(z) = z^n - 1 \text{ for } n = 2, 3, 4,$$

$$p_6(z) = z^6 - \frac{1}{2}z^5 + \frac{11}{4}(1+i)z^4 - \frac{1}{4}(19+3i)z^3 + \frac{1}{4}(11+5i)z^2 - \frac{1}{4}(11+i)z + \frac{3}{2} - 3i.$$

The sets of roots for these polynomials are the following:

- $\{-1, 1\}$ for p_2 ,
- $\{-\frac{1}{2} - \frac{\sqrt{3}}{2}i, -\frac{1}{2} + \frac{\sqrt{3}}{2}i, 1\}$ for p_3 ,
- $\{-1, 1, -i, i\}$ for p_4 ,
- $\{1, 1-i, -1+2i, -\frac{1}{2}(1+i), -\frac{3}{2}i, i\}$ for p_6 .

For polynomials p_2 , p_3 and p_4 , we used the area $[-2, 2] \times [-2, 2]$ for the polynomiographs, whereas for, p_6 , we changed the area to $[-3, 3] \times [-3, 3]$ to cover all polynomial roots. The other parameters used for the generation of polynomiographs were the following: resolution 600×600 pixels, $M = 40$, $\varepsilon = 0.001$, and the convergence test in Equation (20). The colour map used for the colouring of iterations is presented in Figure 2. The values of α were changed with the step 0.01.



Figure 2. The colour map used in the experiments.

All experiments were performed on a computer with the following specifications: Intel Core i5-4570 processor, 16 GB RAM, Windows 10 (64-bit). The software for polynomiograph's generation was implemented in Processing, a programming language based on Java.

4.1. The Results of Experiments

For clear exposition of the obtained results, the polynomials are presented in the following sequence (depending on α -order in $(0, 2)$): basins of attraction for *RL*-Newton and *C*-Newton methods, iteration coloured polynomiographs for *RL*-Newton and *C*-Newton methods and plots of ANI, CAI and the generation time for the *RL*-Newton and *C*-Newton methods. Moreover, for each polynomial, we show polynomiographs and values of ANI, CAI and the generation time for the classical Newton's method (Equation (16)). These results are used as the reference for the results obtained by using the methods with fractional derivatives.

4.1.1. Polynomial p_2

Polynomiographs obtained by the classical Newton's method are presented in Figure 3. The values of ANI, CAI and the generation time are equal to 3.406, 1.0 and 0.366 s, respectively.

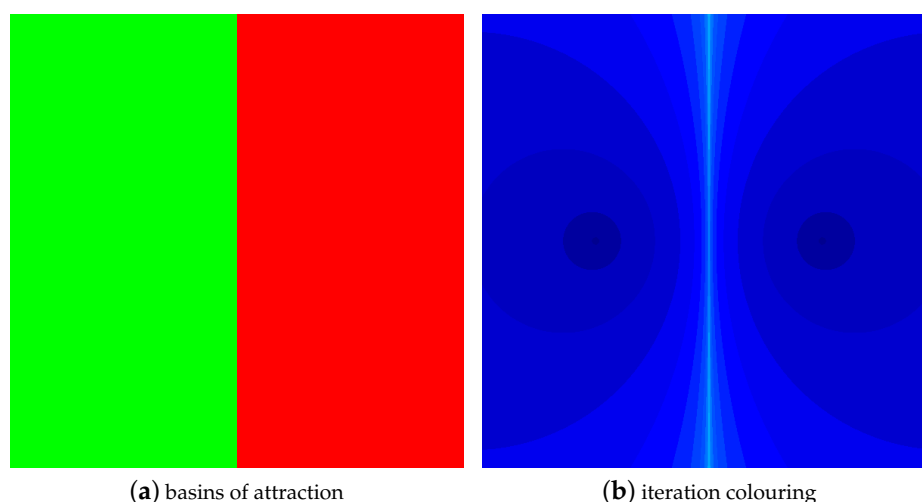


Figure 3. The polynomiographs of the classical Newton's method for p_2 .

The basins of attraction for the *RL*-Newton and *C*-Newton methods are presented in Figures 4 and 5, respectively. The presented basins were generated for various values of α : (a) 0.70; (b) 0.85; (c) 0.99; (d) 1.15; (e) 1.30; and (f) 1.80. By looking at the basins of attraction for α close to 1, i.e. $\alpha = 0.99$ (Figures 4c and 5c),

we see that, compared to the classical Newton's method (Figure 3a), the boundaries between the basins are changing. In the case of the classical Newton's method, this boundary is a straight line. For the RL-Newton method, the boundary bends lightly and some smaller areas are isolating along it, and, for the C-Newton method, the boundary undergo only the bending process. When the value of α is decreasing we see that the boundary for both methods is bending further (Figures 4b and 5b), and, in the case of the RL-Newton method, the isolated areas are enlarging. With a further decreasing of α , the methods stop converging to one of the roots and even stop converging at all (black areas in the polynomiographs). In the case of increasing α above 1, we also see boundary bending, but it is directed in the opposite direction relatively to the direction for $\alpha < 1$ (see Figures 4d and 5d). When one increases the value of α , the basin of one of the roots is shrinking and points for which the methods do not converge to any root are appearing (Figures 4e and 5e). With the further increase of α , we see a continued shrinking of one of the basins and worsening of the convergence for the RL-Newton method (Figure 4f), and, in the case of the C-Newton method, the method converges to only one of the roots or it does not converge at all (Figure 5f).

The polynomiographs obtained with the iteration colouring for the same values of α as for the basins of attraction are presented in Figures 6 and 7. For α close to 1 (Figures 6c and 7c), the area with a light blue colour in the central part bends in comparison to the classical Newton's method (Figure 3b). The bending is similar to the one visible for the basins of attraction. In the other parts of the polynomiograph, the shape of the areas corresponding to the same number of iterations is changing. With the shape change, the dynamics of the methods also changes. When one decreases the value of α , one can observe a similar behaviour as for the basins of attractions, i.e., further bending of the central part (Figures 6b and 7b). Moreover, the dynamics of both methods is increasing, creating very interesting patterns. We can also observe that, to the left of the bent boundary, the number of iterations needed for finding the root is increasing in a significant way in comparison to the right area. After further decreasing of α , the areas where methods stop to converge to the roots are appearing (red colour in Figures 6a and 7a). In the other areas, we see further change of the regions corresponding to the same number of iterations and the increase of the number of iterations performed by the methods. Increasing the value of α above 1 causes the bending of the central part of the polynomiograph (Figures 6d and 7d), which is similar to the bending that we have seen for the basins of attraction. We also see that the number of iterations is increasing, which is especially visible to the left from the bent part. With the further increase of α (Figures 6e and 7e), we see that points for which the methods need large number of iterations or they do not converge to any root appear. Simultaneously, the dynamics of the methods is decreasing. For α close to 2 (Figures 6f and 7f), we can observe further increase of the number of points for which the methods do not converge. In the case of the RL-Newton method, an area with an increased dynamics appears, and, for the C-Newton method, the dynamics is decreasing.

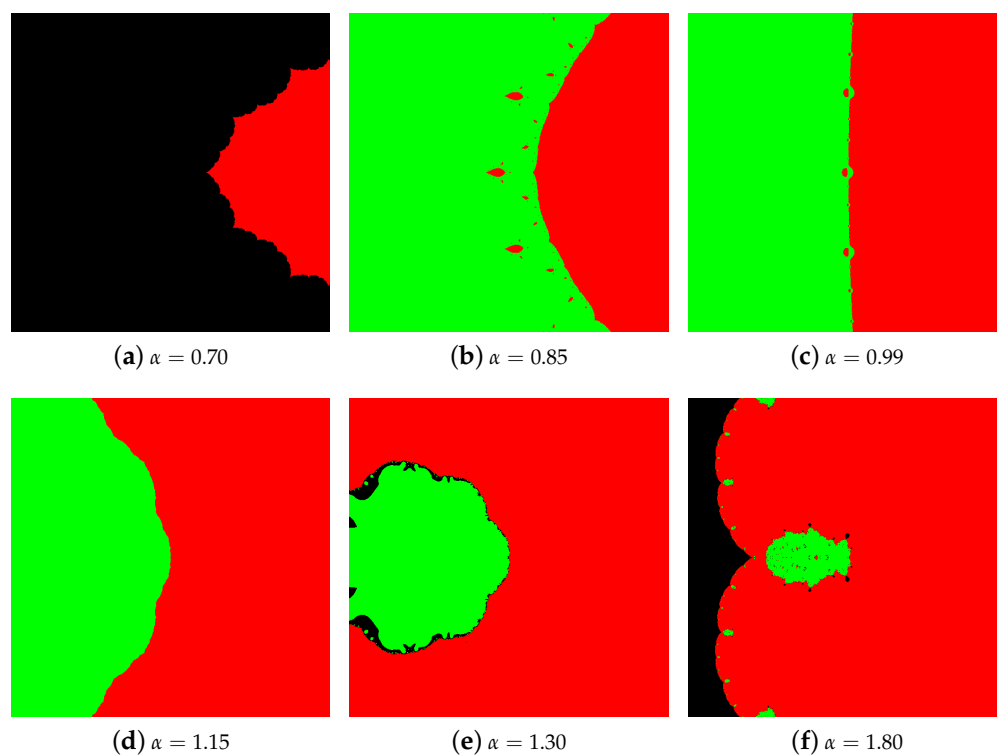


Figure 4. The basins of attraction for the RL-Newton method for p_2 and various values of α .

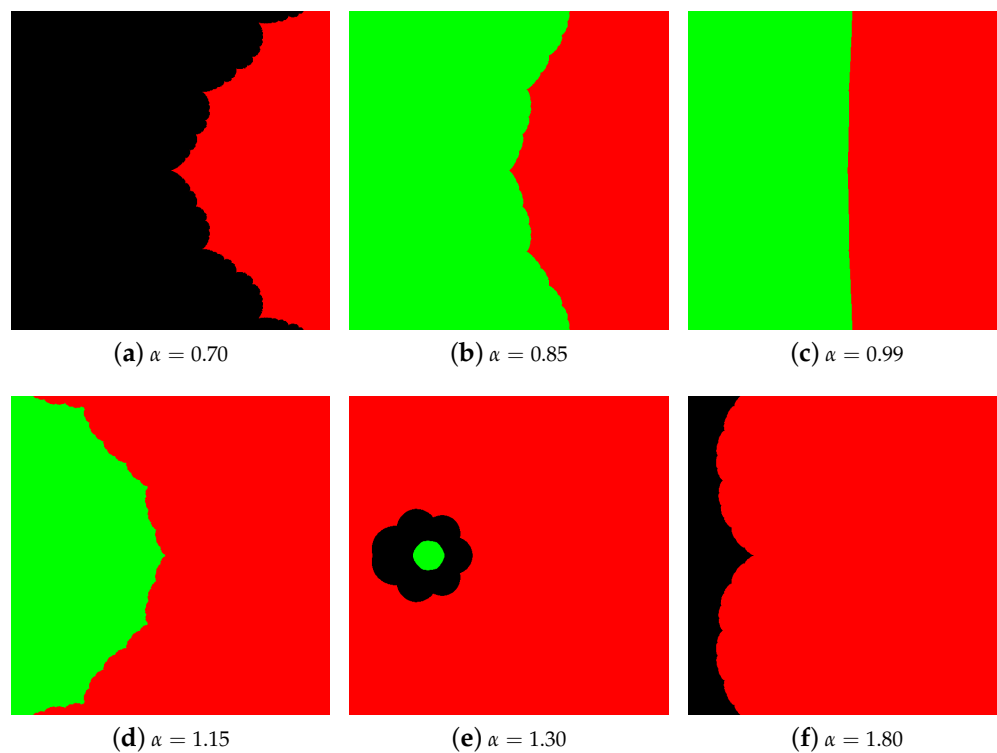


Figure 5. The basins of attraction for the C-Newton method for p_2 and various values of α .

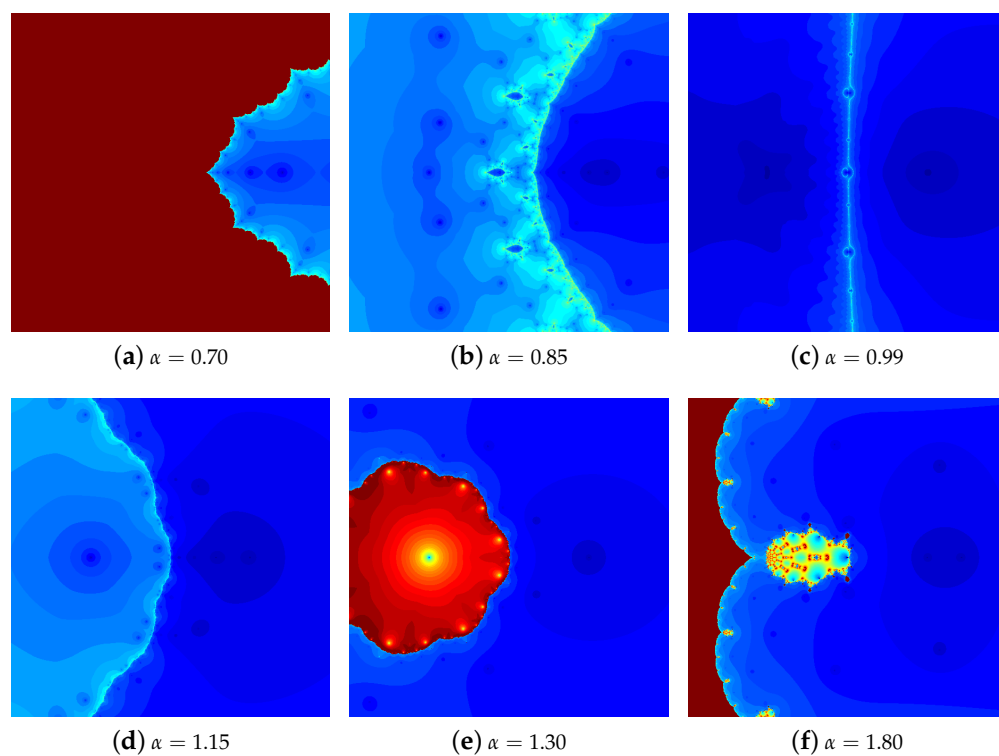


Figure 6. The iteration colouring for the RL-Newton method for p_2 and various values of α .

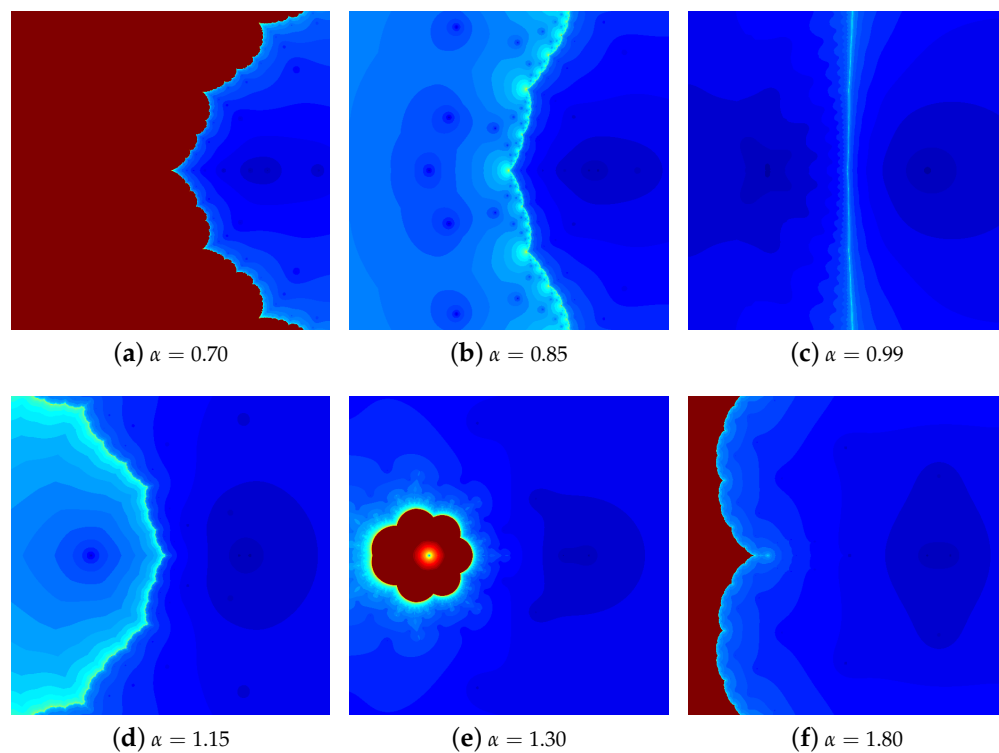


Figure 7. The iteration colouring for the C-Newton method for p_2 and various values of α .

The numerical results obtained for p_2 are presented in Figure 8. In Figure 8a, we see the results for ANI. In the case of both methods, the lowest values of ANI are obtained close to $\alpha = 1$. The minimal value of ANI equals to 4.310 for the RL-Newton method and 4.241 for the C-Newton method. Both are attained at $\alpha = 0.99$. Comparing these results to the classical Newton's method (3.406), we see that the average number of iterations in the methods with the fractional derivatives is larger. Looking at the plots for the both methods, we can observe that the decreasing of α below 1 causes that the ANI is increasing very fast (we can also observe this in Figures 6 and 7 by the change of shades of blue, to lighter shades, and by the appearance of the red colour). For the RL-Newton method for $\alpha < 0.55$, we see that the method needs 40 iterations on average, thus it is not convergent at all. In the case of the C-Newton method, we can observe a similar effect, but for a much lower value of α , namely for $\alpha < 0.04$. When we increase the value of α above 1, we also see the growth of the value of ANI, but not as significant as for $\alpha < 1$. Irrespective of the α value, the C-Newton method obtains lower values of ANI. The results obtained for CAI are presented in Figure 8b. Both methods obtained the maximal value of CAI, equal to 1.0, in the intervals $(0.79, 1.29)$ and $(0.77, 1.25) \cup (1.38, 1.68)$ for the RL-Newton and C-Newton methods, respectively. Therefore, both methods have obtained in a wide range the identical result as the classical Newton's method. Outside of these intervals we can observe decrease of CAI value, wherein for $\alpha < 0.77$ for the RL-Newton method and for $\alpha < 0.74$ for the C-Newton method the decrease is rapid and the methods are reaching value equal to 0.0 for $\alpha < 0.55$ for the RL-Newton method and for $\alpha < 0.06$ for the C-Newton method. Again, irrespective of the α value, the C-Newton method obtains values of CAI larger than the RL-Newton method. In Figure 8c, we see plots with the times of generation of the polynomiographs. For both methods, we can observe a very similar behaviour to the one obtained for ANI. Both methods obtained the shortest times close to $\alpha = 1$, namely 0.924 s at $\alpha = 0.99$ for the RL-Newton method and 0.522 s at $\alpha = 1.0$ for the C-Newton method. The minimal values for both methods are higher than for the classical Newton's method (0.366 s). By changing the values of α we see that the generation times are increasing. For $\alpha < 1$, the growth is significant, obtaining the maximal value of time equal to 6.773 s (at $\alpha = 0.56$) for the RL-Newton method and 4.284 s (at $\alpha = 0.01$) for the C-Newton method. Unlike in the case of ANI, the differences between the RL-Newton and C-Newton methods are more visible, in favour of the C-Newton method. In the case of $\alpha > 1$, the increase is much lower and the methods obtain the maximal value of 2.598 s (at $\alpha = 1.31$) for the RL-Newton method and 1.011 s (at $\alpha = 1.27$) for the C-Newton method. In addition, in this case, the C-Newton method obtains shorter times.

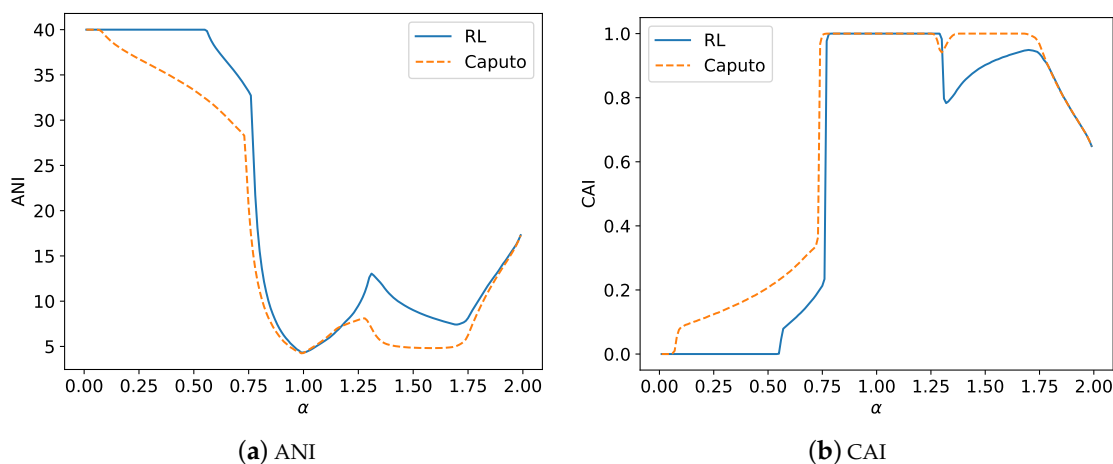


Figure 8. Cont.

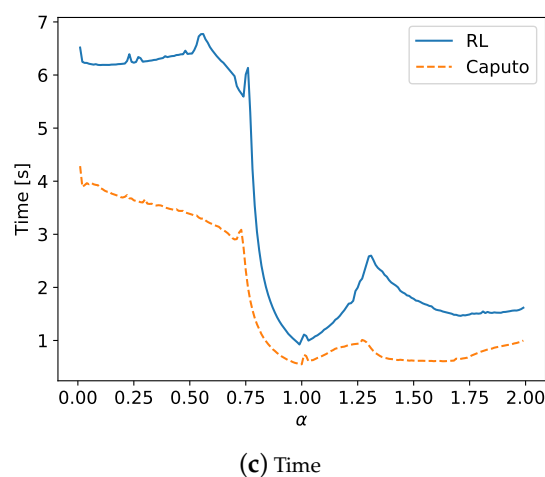


Figure 8. The numerical results for p_2 and the different measures.

4.1.2. Polynomial p_3

Polynomiographs of p_3 obtained by the classical Newton's method are presented in Figure 9. The values of ANI, CAI and the generation time are equal to 5.056, 1.0 and 0.611 s, respectively. Both polynomiographs are symmetric. They present three basins of attraction in Figure 9a and the three roots of unity are clearly visible as dark blue circles in Figure 9b. Boundaries among three areas of the polynomiographs have a complicated fractal structure resembling the shape of braids.

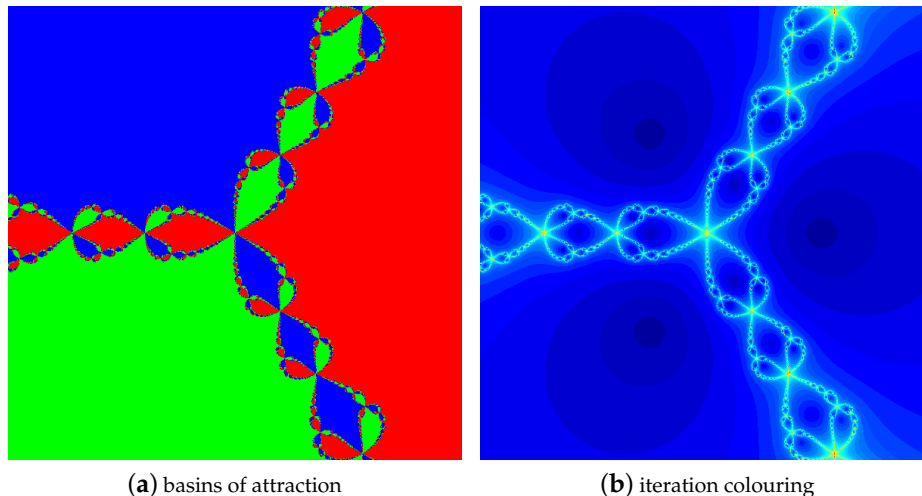


Figure 9. The polynomiographs of the classical Newton's method for p_3 .

Figures 10 and 11 show basins of attraction for the RL-Newton and C-Newton methods, respectively, for fixed α values: (a) 0.70; (b) 0.85; (c) 0.99 (d) 1.30; (e) 1.60; and (f) 1.90. For α of 0.70, 1.60, and 1.90 (RL-Newton method) and for α of 1.60 and 1.90 (C-Newton method), only one red basin of real root 1 exists, covered by black visible areas of non-convergence (larger for RL-Newton method), as shown in Figures 10a,e,f and 11e,f, respectively. Around $\alpha = 0.85$ (RL-Newton method, Figure 10b) and for $\alpha = 0.70$ (C-Newton method, Figure 11a), besides the basin of the real root 1, additionally two basins of conjugate p_3 roots appear. The boundary between them has the form of a wide bunch of braids covered, in the case of the C-Newton method, by small black regularly distributed areas of non-convergence gradually

disappearing for α slightly greater than 0.70. For values of α close to 1, basins for both fractional methods are very similar as for the classical Newton method (Figure 9a). In Figures 10d and 11d, one can see that, for α around 1.30, for the both fractional methods, the boundary between conjugate p_3 roots is a straight segment, whereas, on the other boundaries, small artefacts are visible, larger in the case of the C-Newton method.

Iteration colouring polynomiographs for p_3 presented in Figures 12 and 13 reveal strong dependence of their images on the α order. Generally speaking, polynomiographs generated by both fractional methods are asymmetric apart from the case of α close to 1 corresponding to the classical Newton's method. More blue in the images means higher speed of convergence, whereas yellowish-brownish means slow convergence. Dark blue areas show roots of p_3 , well visible if α is close to 1. Colouring stresses the dynamics of the methods. With α increasing in the range (0.70, 1.90) for RL-Newton method, one can observe in Figure 12 the following sequence of images: (a) very small area of convergence; (b,c) three distinct areas with shrinking braids among them; (d) vanishing boundary among conjugate roots of p_3 ; and (e–f) enlarging areas of non-convergence with visible real root. The sequence of images for C-Newton method (Figure 13) starts from highly dynamical in Figure 13a and continues in Figure 13b–f similarly to those for RL-Newton method, but looking more subtle.

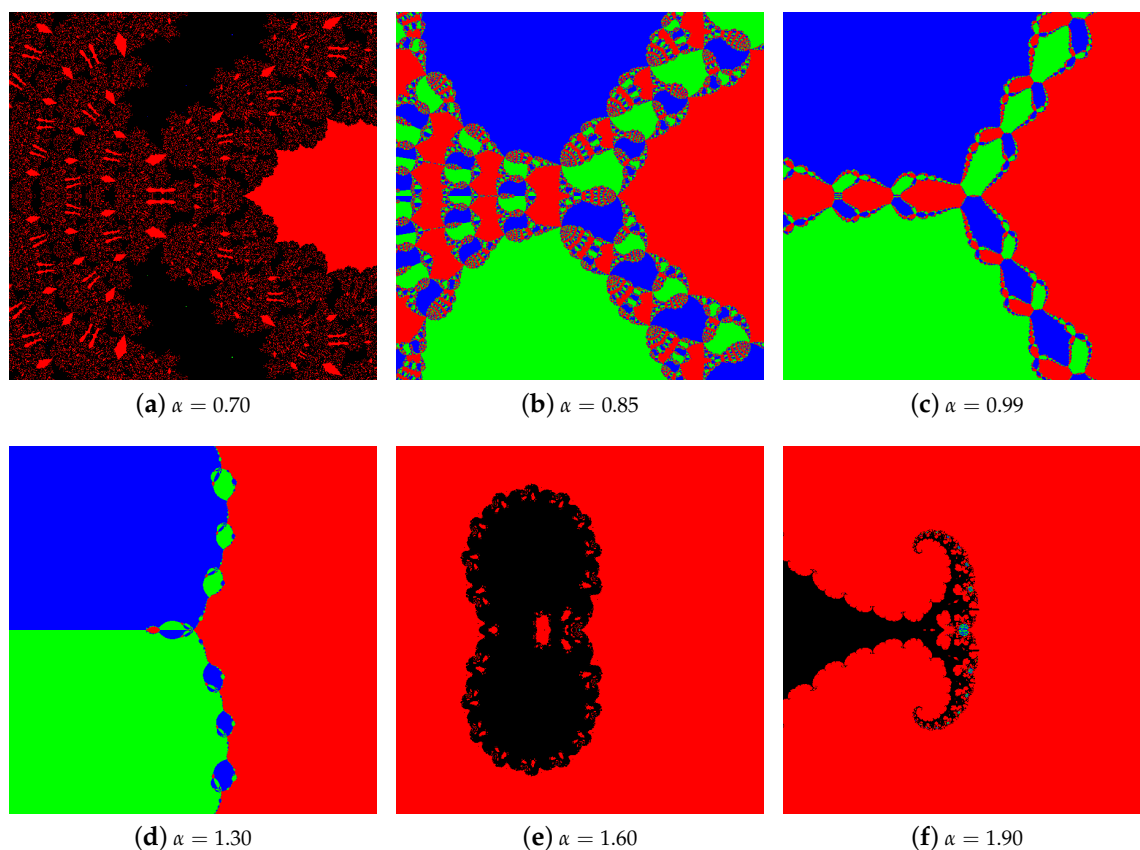


Figure 10. The basins of attraction for the RL-Newton method for p_3 and various values of α .

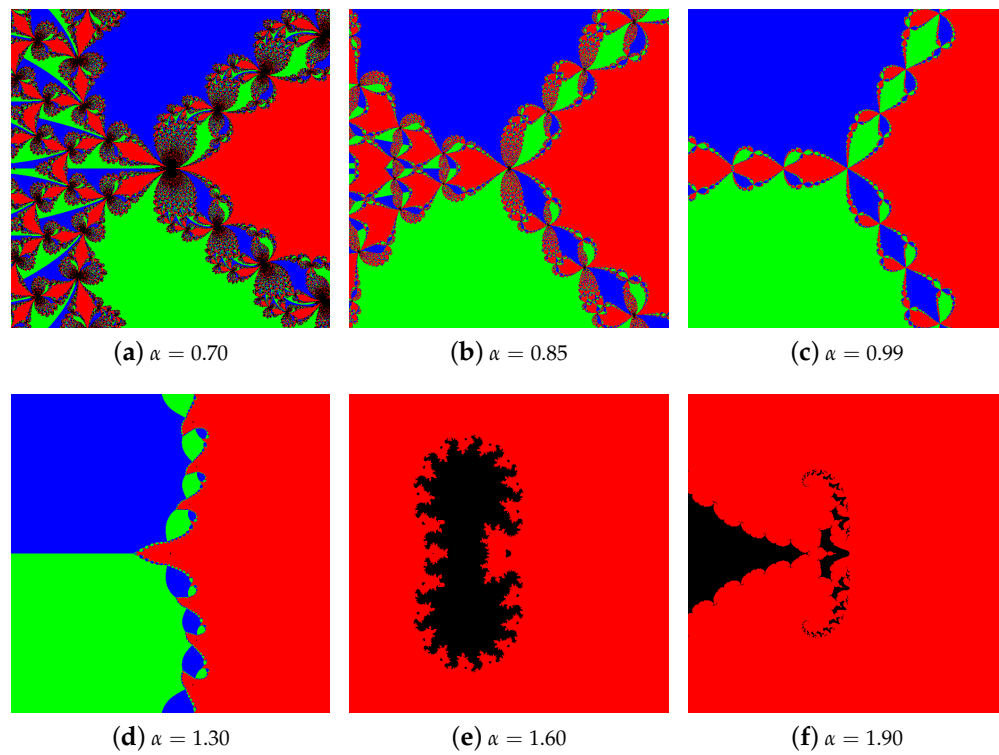


Figure 11. The basins of attraction for the C-Newton method for p_3 and various values of α .

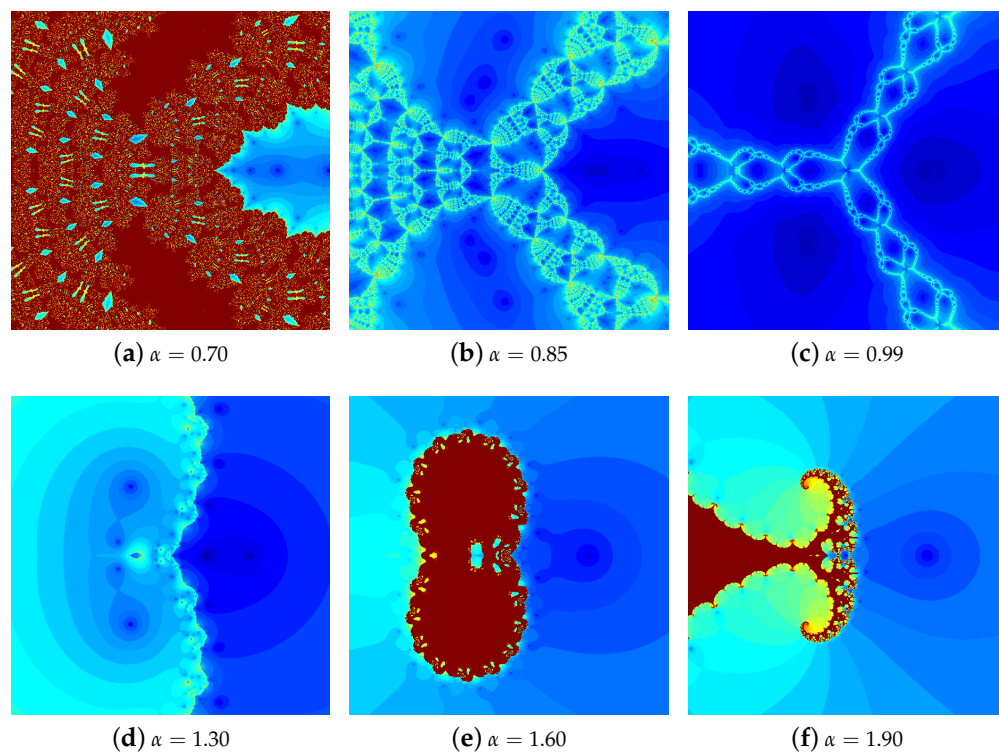


Figure 12. The iteration colouring for the RL-Newton method for p_3 and various values of α .

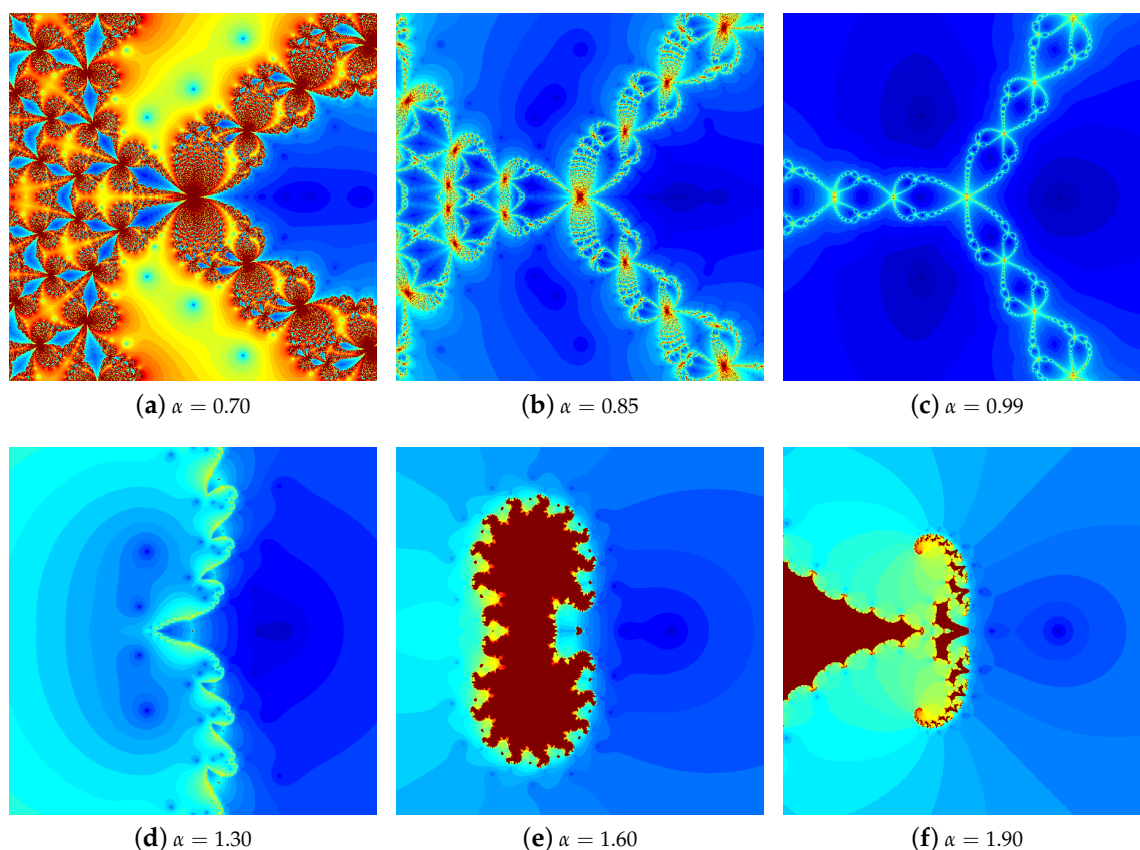


Figure 13. The iteration colouring for the C-Newton method for p_3 and various values of α .

From ANI and CAI plots presented in Figure 14a,b, respectively, one can see that, for α in the range $(0.75, 2)$, in both plots, the runs for both methods are almost identical. That means that both fractional methods behave similarly. In Figure 14c, it is seen that the time of generation is shorter for the C-Newton method. The minimal time of generation of polynomiographs is for α close to 1. More detailed analysis of numerical data shows that, for $\alpha \in (0, 0.6)$ (RL-Newton method) and $\alpha \in (0, 0.36)$ (C-Newton method), the average number of iterations is maximal (40). That means that neither method is convergent. Starting from $\alpha = 0.36$ for the RL-Newton method and $\alpha = 0.36$ for the C-Newton method, the ANI measure quickly decreases and attains its minimal value 5.551 for $\alpha = 1.04$ and 5.745 for $\alpha = 1.02$ for the RL-Newton and the C-Newton method, respectively. If α increases further for both fractional methods, the increase of ANI measure is observed until achieving local maximum equal to 20 iterations at $\alpha = 1.51$ (RL-Newton method) and $\alpha = 1.48$ (C-Newton method). From CAI plots (Figure 14b), one can see that, for $\alpha < 0.36$ (RL-Newton method) and $\alpha < 0.6$ (C-Newton method), neither fractional method is convergent (CAI is zero). Further one can observe that for α in $(0.85, 1.42) \cup (1.72, 1.81)$ almost all points are convergent in both fractional methods. In Figure 14c, presenting the generation times of polynomiographs, it is visible that, for $\alpha \in (0.75, 1.75)$, the generation time plots, in general, are similar to ANI plots. For $\alpha = 0.99$, the times of generation attain their minima equal to 1.302 s (RL-Newton method) and 0.769 s (C-Newton method). It is interesting that ANI and the generation time attain local maximum for α equal approximately 1.5, whereas CAI attains its local minimum for both fractional methods.

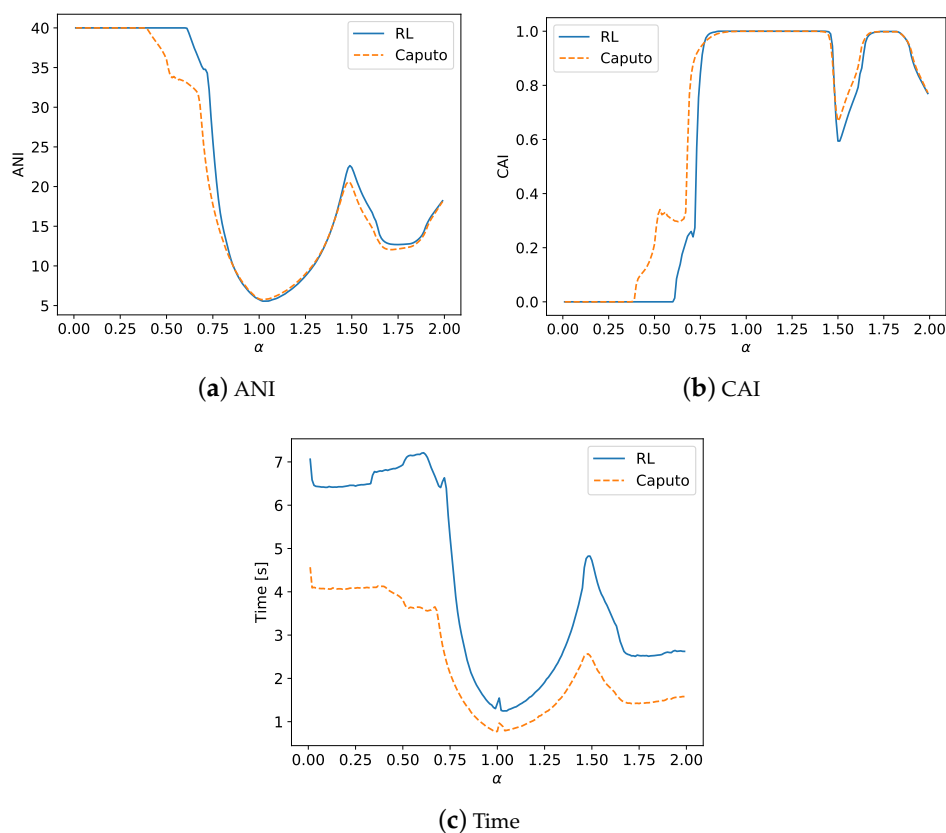


Figure 14. The numerical results for p_3 and the different measures.

4.1.3. Polynomial p_4

The polynomiographs obtained by the classical Newton's method are presented in Figure 15. The values of ANI, CAI and the generation time are equal to 7.440, 0.994 and 0.919 s, respectively.

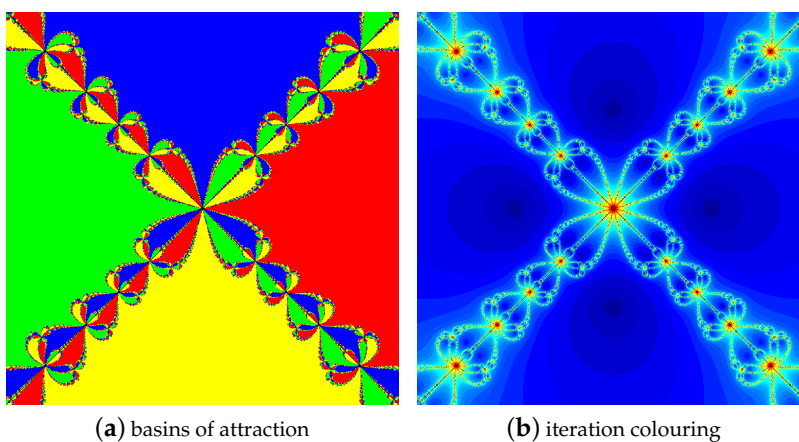


Figure 15. The polynomiographs of the classical Newton's method for p_4 .

The basins of attraction for polynomial p_4 are presented in Figures 16 and 17 for RL-Newton and C-Newton methods, respectively, for $\alpha \in \{0.70, 0.85, 0.99, 1.15, 1.40, 1.65\}$. As one can see for $\alpha = 0.99$,

the general shapes of the basins are quite predictable in comparison to p_3 for both methods. The differences lie only in details. For p_4 , there are four braids and the polynomiographs are also symmetric. For the other values of α , the symmetry is broken bending the braids from the left or the right side depending on whether α is decreased or increased.

In Figures 18 and 19, there are shown polynomiographs for RL-Newton and C-Newton methods, respectively. As one can see, their general shapes are similar as in the case of basins of attraction. Similarly, as in the case of the above presented polynomials p_2 and p_3 , dark blue colour represents the roots of p_4 . They are well visible for $\alpha = 0.99$ (Figures 18c and 19c). Next, the smaller or larger is the value of α , the lesser is the dynamic of convergence (visible as more brown colour).

Figure 20 presents plots of ANI, CAI and the generation time for p_4 . As one can see, for both methods, RL-Newton and C-Newton, for the small value of α , the average number of iterations is maximal (Figure 20a). This means that, for $\alpha < 0.64$ for the RL-Newton method and for $\alpha < 0.5$ for the C-Newton method, these methods have not converged. Starting from $\alpha = 0.65$, ANI intensively decreases for both methods and achieves its minimum for $\alpha = 1.03$ for the RL-Newton method and $\alpha = 0.99$ for the C-Newton method. These minima are equal to 7.638 for the RL-Newton method and 8.106 for the C-Newton method. For $\alpha > 1.03$, ANI increases slowly, achieving the maximum at about 24–25 iterations. As one can expect, by analysing the plot of CAI (Figure 20b), for $\alpha < 0.5$ and for $\alpha < 0.65$, the number of converging points for the RL-Newton and the C-Newton methods, respectively, equal zero. For α from $(0.83, 0.97) \cup (1.01, 1.28)$, all points converge in RL-Newton method and nearly all points (but not all) converge in C-Newton method. By analysing the last plot (Figure 20c), one can observe that the generation time for RL-Newton method has fluctuations for small values of α . Both methods achieve their minimal times for $\alpha = 1.03$ and $\alpha = 0.99$ for RL-Newton and C-Newton methods, respectively. As in the case of the other polynomials, one can see that the C-Newton method is faster than the RL-Newton one. From all these plots, it is easily seen that at $\alpha = 1.65$ in all plots there is maximum or minimum.

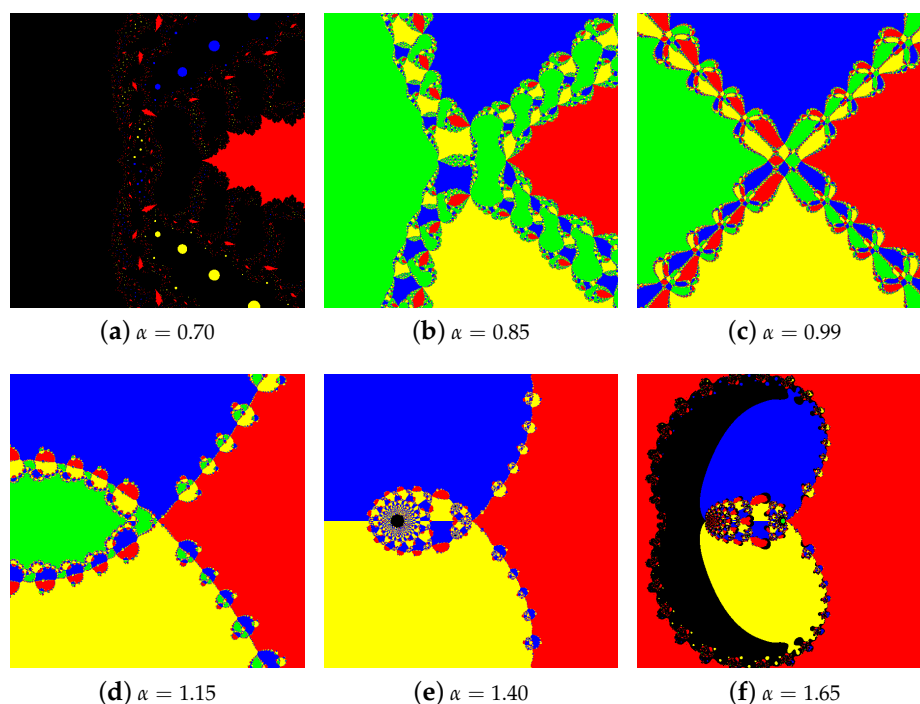


Figure 16. The basins of attraction for the RL-Newton method for p_4 and various values of α .

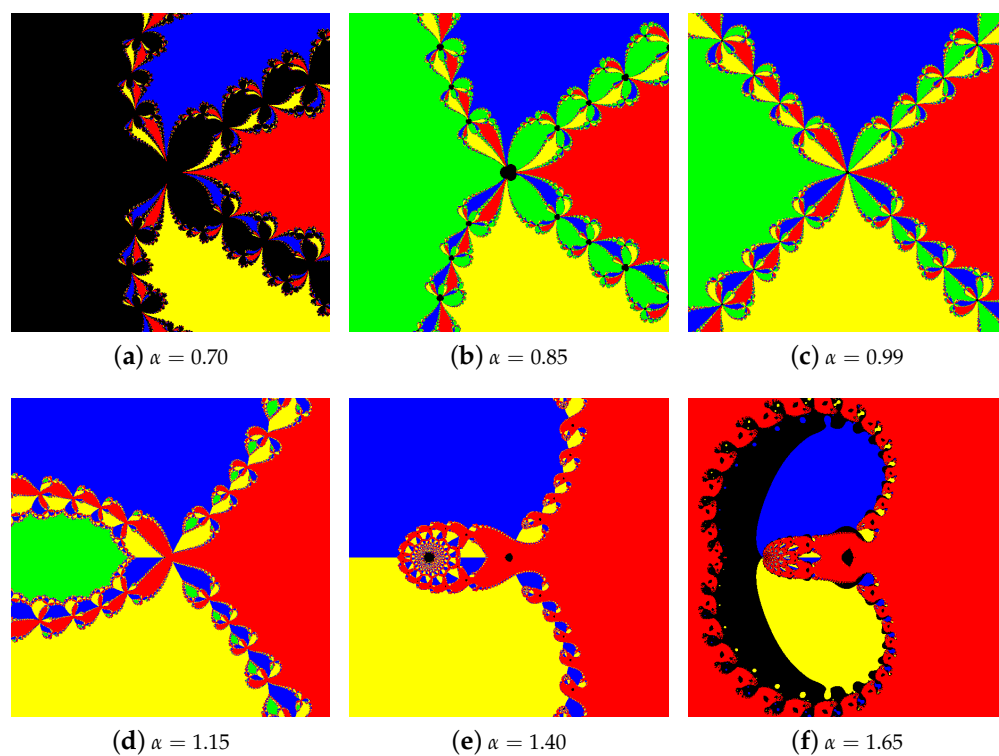


Figure 17. The basins of attraction for the C-Newton method for p_4 and various values of α .

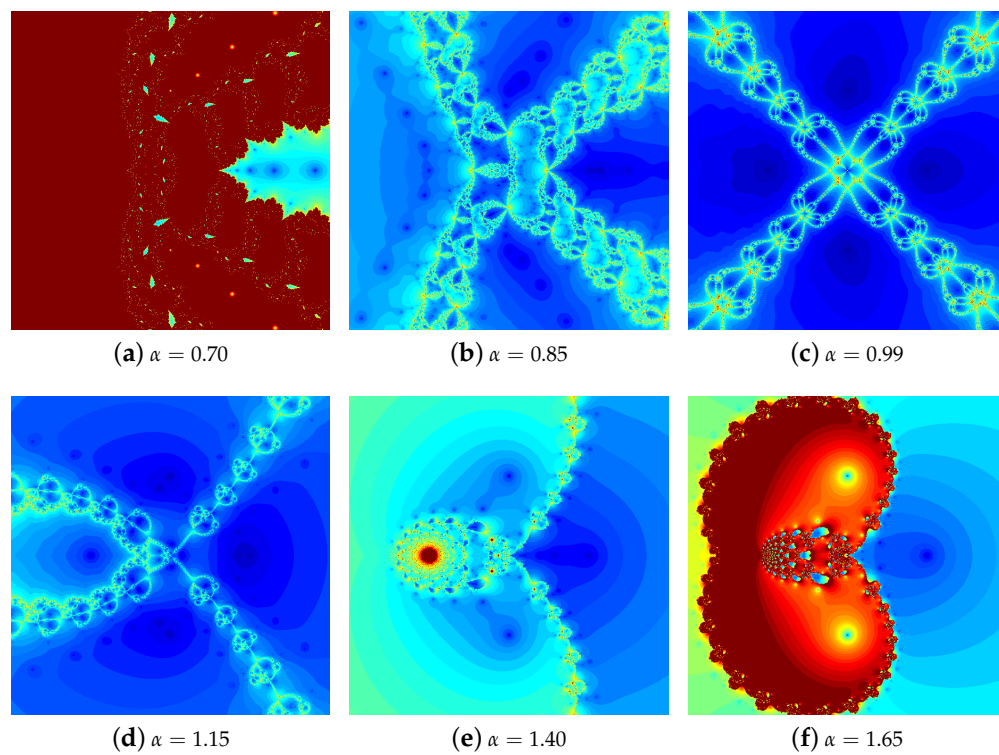


Figure 18. The iteration colouring for the RL-Newton method for p_4 and various values of α .

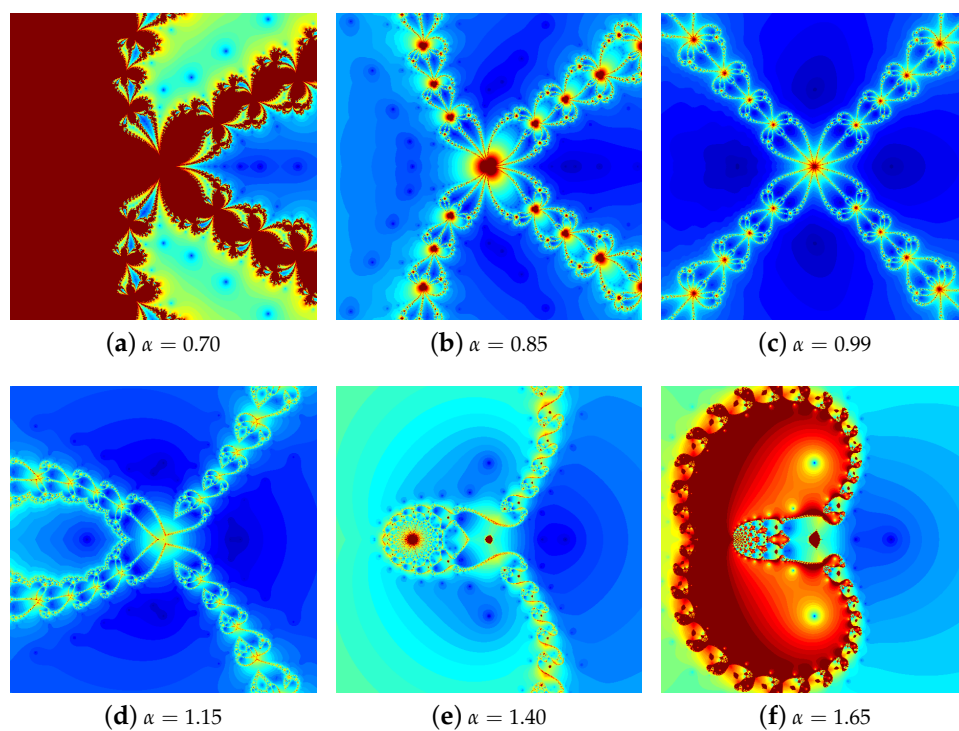


Figure 19. The iteration colouring for the C-Newton method for p_4 and various values of α .

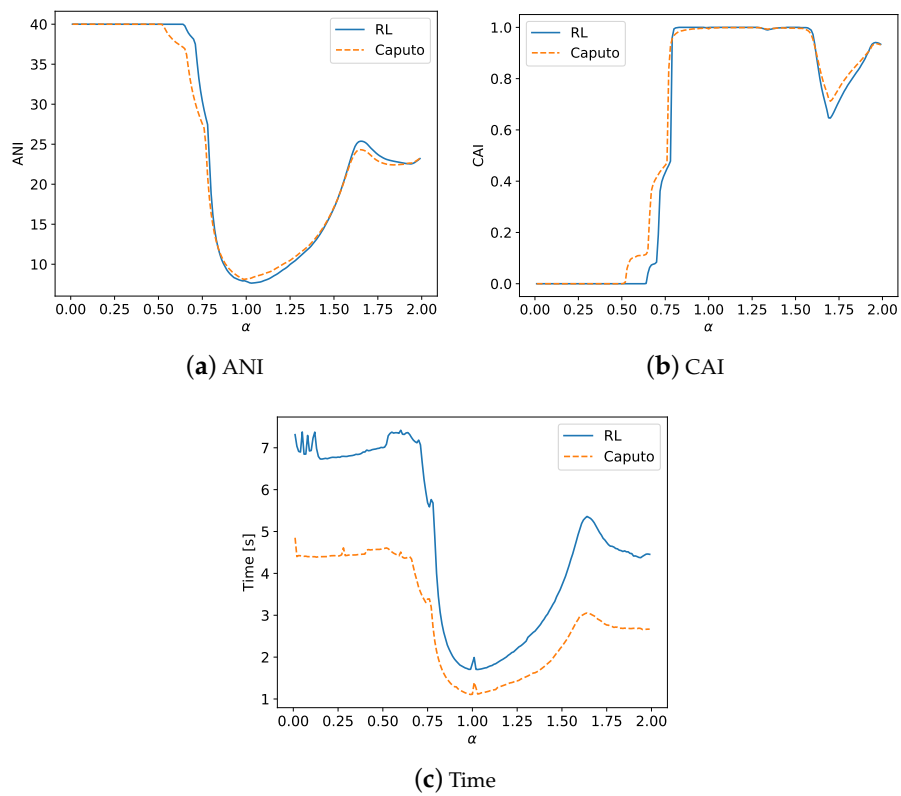


Figure 20. The numerical results for p_4 and the different measures.

4.1.4. Polynomial p_6

The polynomiographs obtained by the classical Newton's method are presented in Figure 21. The values of ANI, CAI and the generation time are equal to 7.984, 1.0 and 1.368 s, respectively.

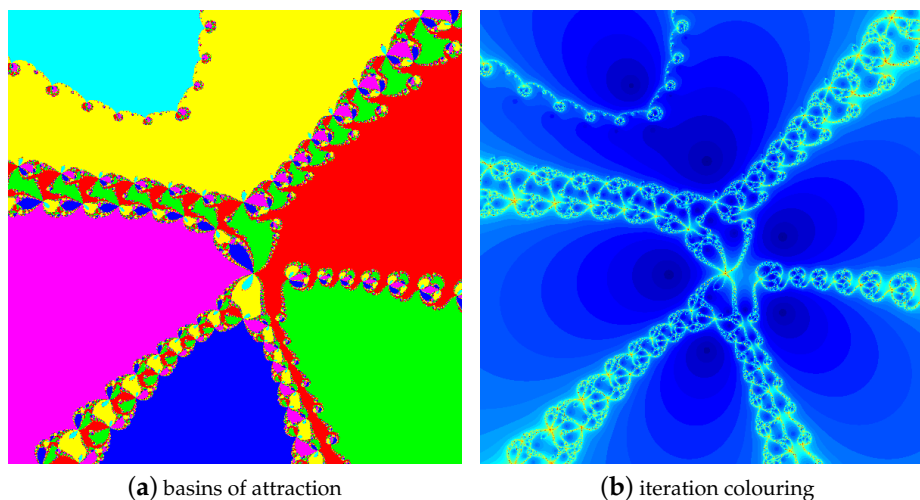


Figure 21. The polynomiographs of the classical Newton's method for p_6 .

The basins of attraction for the RL-Newton and C-Newton methods are presented in Figures 22 and 23, respectively. The presented basins were generated for various values of α : (a) 0.70; (b) 0.85; (c) 0.99; (d) 1.20; (e) 1.50; and (f) 1.80. For $\alpha = 0.99$ (Figures 22c and 23c), by comparing the basins with the basins obtained with the classical Newton's method, we see that the change of their shape is very small. The most noticeable change is visible in the braids on the left, i.e., they move a little bit away from the real axis. When the value of α is decreasing (Figures 22b and 23b), we see that the lower-left braid is moving further away from the real axis, whereas the upper-left is splitting into many smaller braids. For the RL-Newton method, the splitting is more chaotic than for the C-Newton method. With a further decrease of the α order, the methods stop converging to some of the roots; we see only basins for three roots (red, green and yellow) out of six. In the case of the RL-Newton method, the yellow basins are very small compared to the C-Newton method. Moreover, many of the points in the considered area do not converge for either method (black colour in the polynomiographs). In the case of increasing α above 1, we see more interesting changes in the shape of the basins. The braids are getting smaller and, at the same time they, are bent (see Figures 22d and 23d). For $\alpha = 1.50$ (Figures 22e and 23e), we can observe that the braids from the upper-left part of Figures 22c and 23c are joining together and that some of the basins are shrinking. Moreover, we see that a spiral pattern appears near the centre. For high values of α , i.e., $\alpha = 1.80$ (Figures 22f and 23f), we see that both fractional methods converge mostly to only two roots. For the other roots, the basins are very small or they disappear: the cyan basin is not present in either polynomiograph, whereas the blue and magenta basins disappear for the C-Newton method. Furthermore, the area in which the methods do not converge to any root increases, but not so much as in the case of $\alpha < 1$.

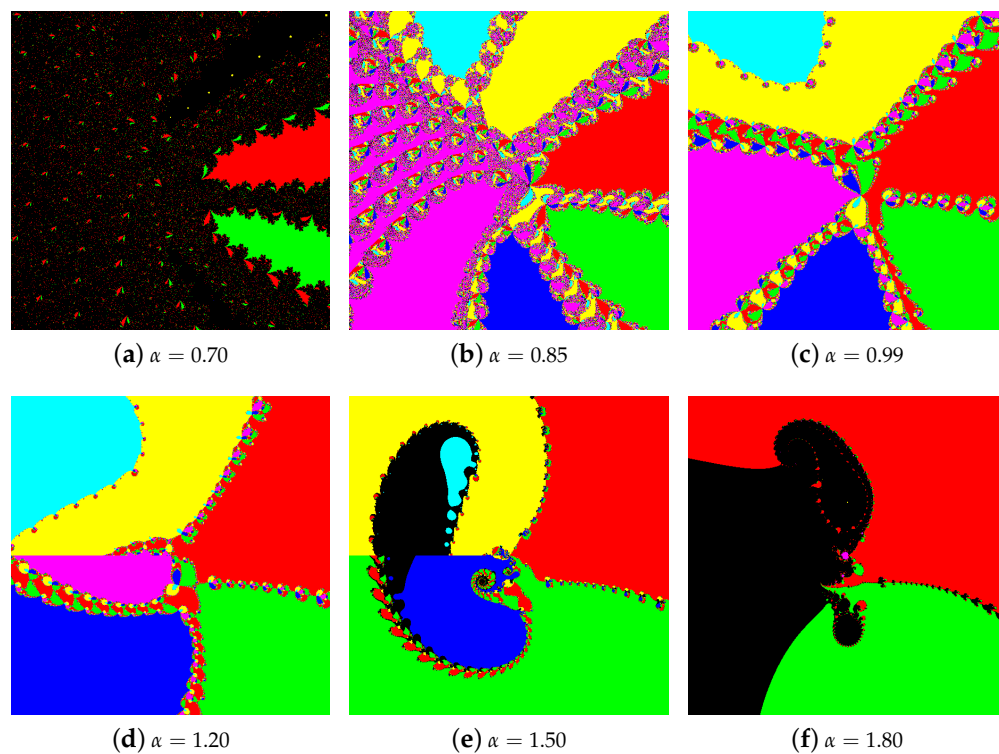


Figure 22. The basins of attraction for the RL-Newton method for p_6 and various values of α .

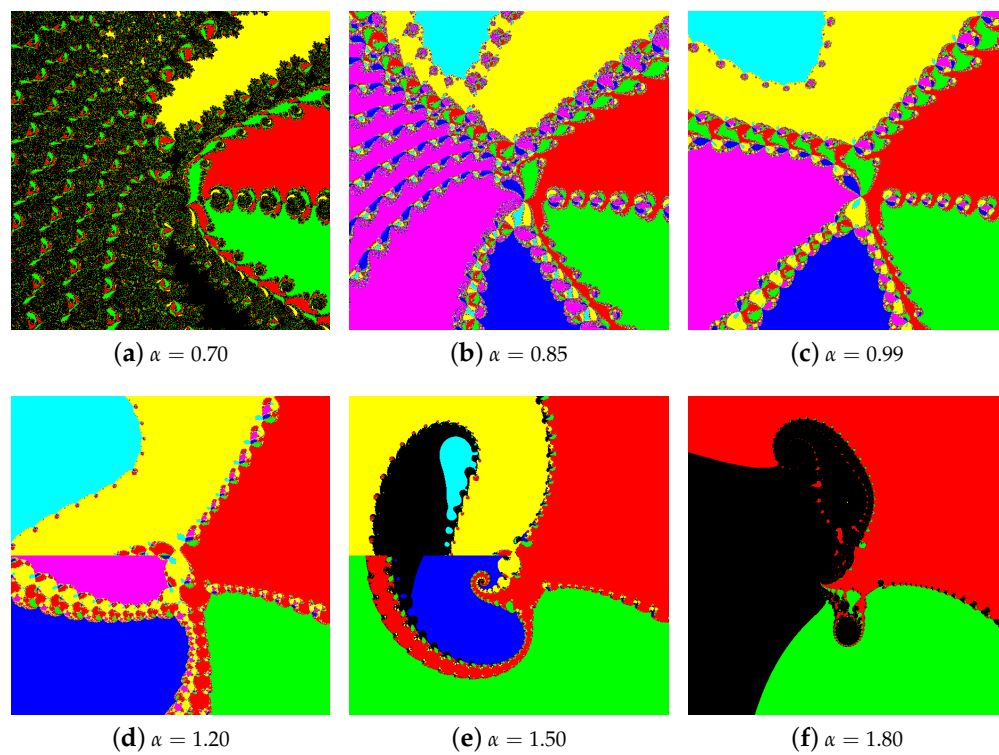


Figure 23. The basins of attraction for the C-Newton method for p_6 and various values of α .

The polynomiographs obtained with the iteration colouring for the same values of α as for the basins of attraction are presented in Figures 24 and 25. From these polynomiographs we can observe that the braids behave in a similar fashion as in the case of basins of attraction. For $\alpha < 1$, the left braids are moving away from the real axis and they are splitting, whereas, for $\alpha > 1$, the braids are getting smaller and, moreover, they are bending and joining together, forming interesting patterns, especially in Figures 24e and 25e. Furthermore, with the change of the α -order the dynamics of the methods also changes. The change is different for miscellaneous values of this parameter. For the value near 1 (Figures 24c and 25c), we can observe only a small change in dynamics in comparison to the classical Newton's method (Figure 21b). When we decrease the value to 0.85, we see increase in dynamics in all areas of the polynomiographs (see Figures 24b and 25b). Then, for $\alpha = 0.70$, we see a low dynamic for both methods (Figures 24a and 25a) because most of the points in the considered area do not converge, i.e., the points with red colour. In this case, the RL-Newton method has much lower dynamics than the C-Newton one. Increasing the value of α above 1 changes the dynamics in the areas outside the braids. For both methods, the dynamics is increasing in the areas laying on the right side. In the left part of the polynomiographs, the dynamics is decreasing due to the presence of points that do not converge to any root.

The numerical results obtained for p_6 are presented in Figure 26. In Figure 26a, we see the results for ANI. The lowest value of ANI is obtained for values of α near 1. The minimal value of ANI, equal to 8.406 for the RL-Newton method, is attained at $\alpha = 1.02$, whereas the minimal value 8.480 for the C-Newton method is attained at $\alpha = 0.99$. By comparing these results to the classical Newton's method (7.984), we see that the ANI value for the methods with fractional derivatives is larger. When we look at the plots, we see that both methods have almost identical behaviour. For $\alpha < 1$, the C-Newton method obtained slightly lower values of ANI, whereas, for $\alpha > 1$, the situation is inverted, i.e., the RL-Newton method obtained lower value, but only for $\alpha \in (1.0, 1.39)$. Both methods lose their convergence properties, i.e., obtain the maximal number of iterations for all points in polynomiographs equal to 40, for different values. The threshold for the RL-Newton method is equal to 0.57 and for the C-Newton method is equal to 0.53. The results obtained for CAI are presented in Figure 26b. Similar to the previous polynomials, both methods obtained the highest values of CAI equal to 1.0 in the neighbourhood of $\alpha = 1.0$, i.e., for $\alpha \in (0.96, 0.99) \cup (1.01, 1.33)$ and $\alpha \in (0.94, 1.22)$ for the RL-Newton and C-Newton methods, respectively. For $\alpha > 1.0$, both methods behave almost identically, whereas, for $\alpha < 1$, we see that the C-Newton method obtains greater values of CAI. The decrease of the value of CAI outside the intervals in which the methods obtained the values equal to 1.0 is very rapid. Moreover, for $\alpha < 0.65$ in the case of the RL-Newton method and $\alpha < 0.55$ in the case of the C-Newton method, the value of CAI is equal to 0.0, thus, for these values of α , the methods do not converge to any root. In Figure 26c, we see the plots with the times of generation of the polynomiographs. The best times for both methods, as in the previous examples, are near $\alpha = 1.0$, i.e., at $\alpha = 1.02$ they are equal to 5.081 s and 4.244 s for the RL-Newton and C-Newton method, respectively. These times are higher than for the classical Newton's method (1.368 s). By increasing or decreasing the value of α above or below 1.02, we can observe a rapid increase of the generation time. Both methods behave very similarly, but the growth in the case of the C-Newton method is smaller than for the RL-Newton method. Both methods obtained the longest times for $\alpha < 1.02$, i.e., the RL-Newton obtained time equal to 21.684 s at $\alpha = 0.57$, whereas the C-Newton method obtained the time equal to 18.961 s at $\alpha = 0.56$. Irrespective of the α value, the C-Newton method obtains shorter times than the RL-Newton method.

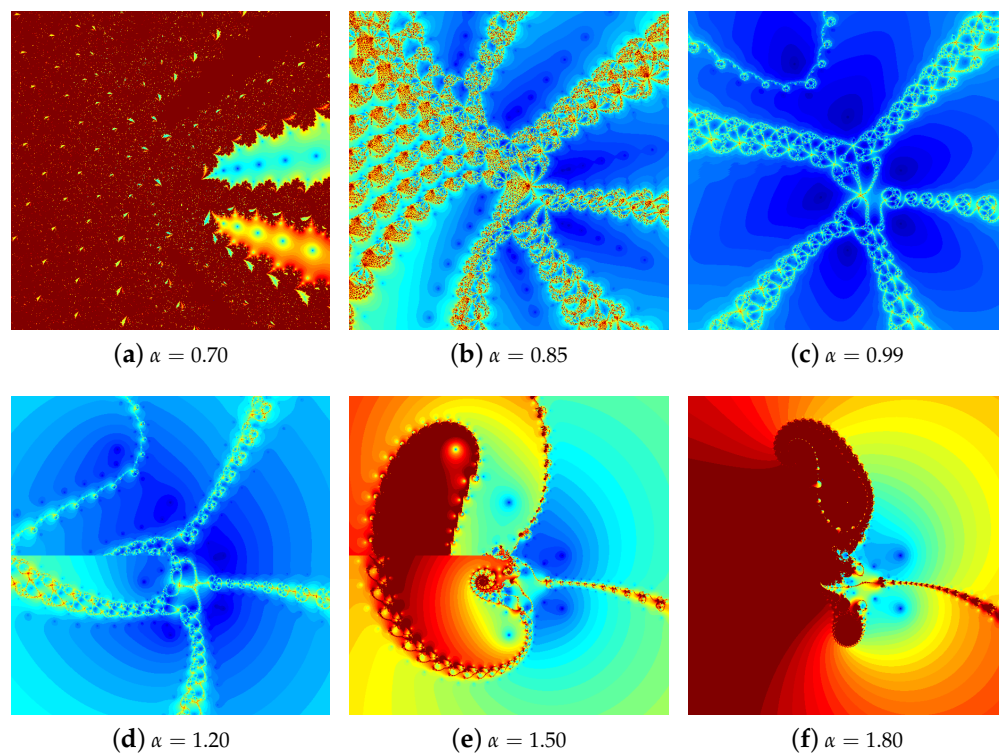


Figure 24. The iteration colouring for the RL-Newton method for p_6 and various values of α .

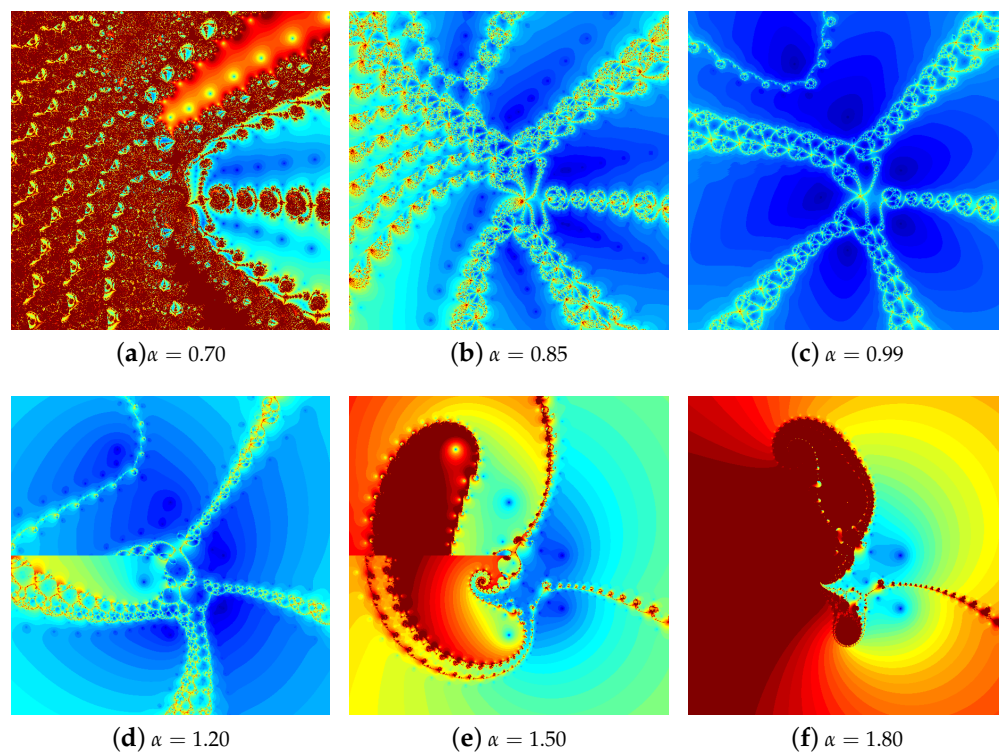


Figure 25. The iteration colouring for the C-Newton method for p_6 and various values of α .

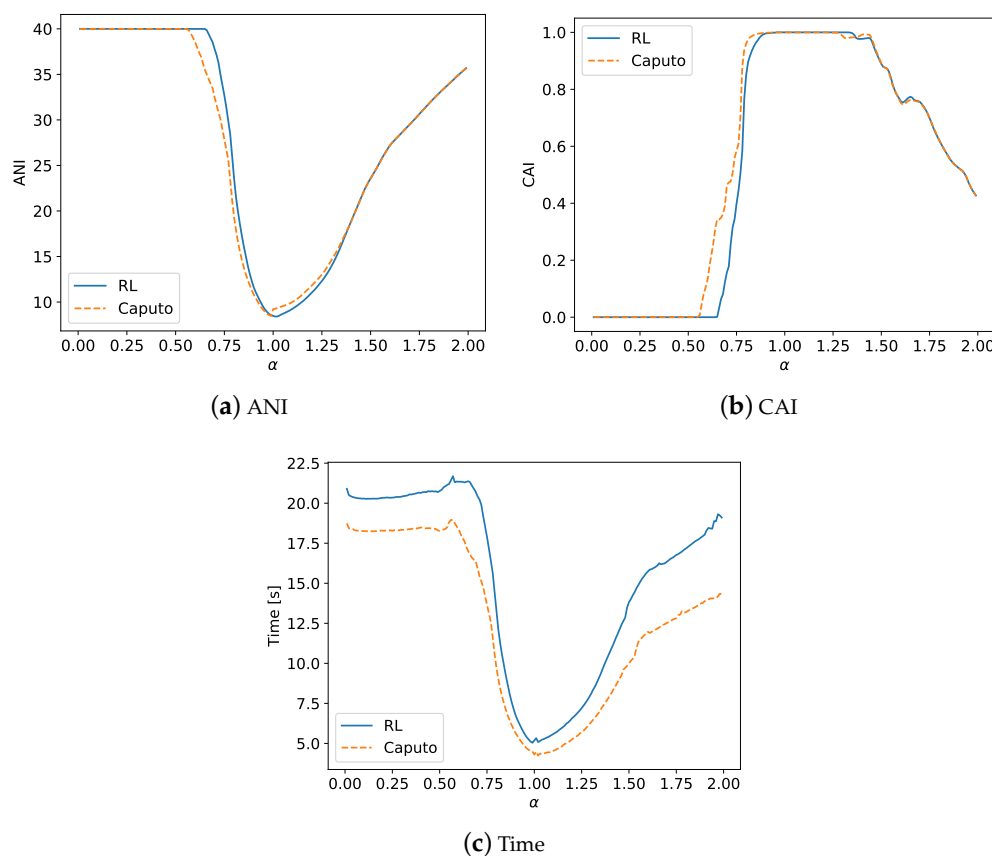


Figure 26. The numerical results for p_6 and the different measures.

4.2. Summary of the Experiments

The polynomiographs (the basins of attraction and the polynomiographs with the iteration colouring) generated with the fractional methods for α approaching to 1 from the above or from the below are approaching the set of polynomiographs obtained for the classical Newton's method. When we move away from $\alpha = 1.0$ in either direction, we can observe that the shapes of the polynomiographs change in a significant way in the left part of the images. Together with the shape change, the dynamics of the methods also changes. The dynamics is high for $\alpha \in (0.8, 1.4)$, and then the dynamics is lower due to the presence of many points for which the methods do not converge to any root. For the basins of attraction, when we move away from $\alpha = 1.0$, some of the roots are not found by the methods, i.e., the basins of attraction of these roots just vanish. The polynomiographs for p_2, p_3, p_4 are not symmetric as in the classical Newton method and contain more areas with turbulences and chaos. They look more "fractally". Moreover, in many of the polynomiographs, we can observe a straight line along negative real axis, e.g., in Figures 10d, 11d, 16e and 17e, among others. These straight lines might be related with the choice of the negative real half-axis as the branch cut line. From the numerical experiments, we can observe that the C-Newton method behaves better than the RL-Newton method. The former method obtains shorter times, needs fewer iterations on average (lower values of ANI), and, at the same time, more points are converging (greater value of CAI) to the roots. The best values of each of the three measures are attained for α in the neighbourhood of 1.0.

5. Conclusions and Future Work

In this paper, some modifications of the root-finding process based on fractional Riemann–Liouville and Caputo derivatives are investigated via polynomiographs. For several polynomials, the basins of attractions and the images coloured with respect to the number of iterations together with the plots showing the dependence of the α -order on the three different measures (ANI, CAI and the generation time) are presented.

In general, fractional derivatives in the complex plane can be well-defined not only by choosing the principal branch of multivalued expressions that are in their definitions, as usually is implemented in programming languages and CAS systems. Therefore, the question concerning the behaviour of the considered methods with fractional derivatives for the choice of the other branches different from the principal one is an interesting.

As in [5], the results of this paper can be further modified by the use of multi-parameter iterations, various convergence tests or different colour maps. In addition, the application of the higher order methods based on the Basic or Euler–Schröder families of iterations [2] with non-standard iteration processes lead to further possible modifications. Next, it is worth checking experimentally the use of complex parameters instead of real ones in multi-parameter iterations. Such a change probably does not destabilise convergence of the investigated processes. At the end, the artistic aspects of polynomiographs should be mentioned. They are images looking nicely and their appearance is greatly influenced by the different colour palettes used to generate them.

Author Contributions: Conceptualisation, K.G., W.K. and A.L.; methodology, K.G., W.K. and A.L.; software, K.G.; formal analysis, K.G., W.K. and A.L.; investigation, K.G.; writing—original draft preparation, K.G., W.K. and A.L.; writing—review and editing, K.G. and A.L.; and visualisation, K.G.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mandelbrot, B. *The Fractal Geometry of Nature*; W.H. Freeman and Company: New York, NY, USA, 1983.
2. Kalantari, B. *Polynomial Root-Finding and Polynomiography*; World Scientific: Singapore, 2009. [CrossRef]
3. Kalantari, B. Polynomiography: From the Fundamental Theorem of Algebra to Art. *Leonardo* **2005**, *38*, 233–238. [CrossRef]
4. 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]
5. Gdawiec, K.; Kotarski, W.; Lisowska, A. Polynomiography Based on the Non-standard Newton-like Root Finding Methods. *Abstr. Appl. Anal.* **2015**, *2015*, 797594. [CrossRef]
6. Ardelean, G.; Cosma, O.; Balog, L. A Comparison of Some Fixed Point Iteration Procedures by using the Basins of Attraction. *Carpath. J. Math.* **2016**, *32*, 277–284.
7. Chun, C.; Neta, B. Comparison of Several Families of Optimal Eighth Order Methods. *Appl. Math. Comput.* **2016**, *274*, 762–773. [CrossRef]
8. Chun, C.; Neta, B. On the New Family of Optimal Eighth Order Methods Developed by Lotfi et al. *Numer. Algorithms* **2016**, *72*, 636–637. [CrossRef]
9. Ardelean, G.; Balog, L. A Qualitative Study of Agarwal et al. Iteration Procedure for Fixed Points Approximation. *Creat. Math. Inform.* **2016**, *25*, 135–139.
10. Gdawiec, K. Fractal Patterns from the Dynamics of Combined Polynomial Root Finding Methods. *Nonlinear Dyn.* **2017**, *90*, 2457–2479. [CrossRef]
11. Brambila-Paz, F.; Torres-Hernandez, A. Fractional Newton–Raphson Method. 2017. Available online: <https://arxiv.org/abs/1710.07634> (accessed on 28 July 2019).

12. Brambila-Paz, F.; Torres-Hernandez, A.; Iturrarán-Viveros, U.; Caballero-Cruz, R. Fractional Newton-Raphson Method Accelerated with Aitken's Method. 2018. Available online: <https://arxiv.org/abs/1804.08445> (accessed on 28 July 2019).
13. Abel, N. Opløsning af et Par Opgaver ved Hjælp af bestemte Integraler. *Mag. Naturvidensk.* **1823**, *2*, 205–215.
14. Sun, H.; Zhang, Y.; Baleanu, D.; Chen, W.; Chen, Y. A New Collection of Real World Applications of Fractional Calculus in Science and Engineering. *Commun. Nonlinear Sci. Numer. Simul.* **2018**, *64*, 213–231. [[CrossRef](#)]
15. Herrmann, R. *Fractional Calculus: An Introduction for Physicists*, 2nd ed.; World Scientific: Singapore, 2014. [[CrossRef](#)]
16. Hilfer, R. *Applications of Fractional Calculus in Physics*; World Scientific: Singapore, 2000. [[CrossRef](#)]
17. Oldham, K.; Spanier, J. The Fractional Calculus: Theory and Applications of Differentiation and Integration to Arbitrary Order. In *Mathematics in Science and Engineering*; Academic Press: San Diego, CA, USA, 1974; Volume 111.
18. Samko, S.; Kilbas, A.; Marichev, O. *Fractional Integrals and Derivatives: Theory and Applications*; Gordon and Breach Science Publishers: Amsterdam, The Netherlands, 1993.
19. Wheeler, N. Construction and Physical Application of the Fractional Calculus. 1997. Available online: <https://www.reed.edu/physics/faculty/wheeler/documents/Miscellaneous%20Math/Fractional%20Calculus/A.%20Fractional%20Calculus.pdf> (accessed on 28 July 2019).
20. Ishteva, M. Properties and Applications of the Fractional Caputo Operator. Master's Thesis, Department of Mathematics, Universität Karlsruhe, Karlsruhe, Germany, 2005.
21. Diethelm, K. *The Analysis of Fractional Differential Equations*; Springer: Heidelberg, Germany, 2010. [[CrossRef](#)]
22. Vance, D. Fractional Derivatives and Fractional Mechanics. 2014. Available online: https://sites.math.washington.edu/~morrow/336_14/papers/danny.pdf (accessed on 28 July 2019).
23. Chong, Y. Branch Points and Branch Cuts. 2016. Available online: http://www1.spms.ntu.edu.sg/~ydcjong/teaching/07_branch_cuts.pdf (accessed on 28 July 2019).
24. Le Bourdais, F. Understanding Branch Cuts in the Complex Plane. 2016. Available online: <http://flothesof.github.io/branch-cuts-with-square-roots.html> (accessed on 28 July 2019).
25. Kahan, W. Branch Cuts for Complex Elementary Functions or Much Ado About Nothing's Sign Bit. In *Proceedings of the Joint IMA/SIAM Conference on the State of the Art in Numerical Analysis*; Iserles, A., Powell, M., Eds.; Clarendon Press: New York, NY, USA, 1987; pp. 165–211.
26. England, M.; Cheb-Terrab, E.; Bradford, R.; Davenport, J.; Wilson, D. Branch Cuts in Maple 17. *ACM Commun. Comput. Algebra* **2014**, *48*, 24–27. [[CrossRef](#)]
27. Munkhammar, J. Fractional Calculus and the Taylor-Riemann Series. *Rose Hulman Undergrad. Math. J.* **2005**, *6*, 6.
28. Ortigueira, M.; Rodríguez-Germá, L.; Trujillo, J. Complex Grünwald-Letnikov, Liouville, Riemann-Liouville, and Caputo Derivatives for Analytic Functions. *Commun. Nonlinear Sci. Numer. Simul.* **2011**, *16*, 4174–4182. [[CrossRef](#)]
29. Li, C.; Daoa, X.; Guoa, P. Fractional Derivatives in Complex Planes. *Nonlinear Anal.* **2009**, *71*, 1857–1869. [[CrossRef](#)]
30. Ortigueira, M.D. A Coherent Approach to Non-integer Order Derivatives. *Signal Process.* **2006**, *86*, 2505–2515. [[CrossRef](#)]

