

# On Smooth Fractal Coloring Techniques

Jussi Härkönen

Master's Thesis

Department of Mathematics

Åbo Akademi University

2007

## Abstract

Härkönen, Jussi: On Smooth Fractal Coloring Techniques

Master's Thesis, Department of Mathematics, Åbo Akademi University, Turku, 2007, 61 pages.

The thesis was carried out under the supervision of Professor Göran Högnäs.

Keywords: coloring function, coloring algorithm, escape time algorithm, fractal, fractal art, ultra fractal, truncated orbit, iteration count, triangle inequality average, curvature average, stripe average.

This work studies the mathematics of selected techniques for coloring fractal images. The classic escape time algorithm is extended by adding the concepts of coloring, palette and index functions. The coloring function is evaluated for each pixel of an image, whereas palette and index functions map this value to the final RGB color. In addition to good performance, also smoothness and the possibility to adjust the visual appearance are desirable characteristics of coloring functions.

The Smooth Iteration Count coloring is the smooth equivalent of the classic Iteration Count coloring. Its continuity and smoothness are studied thoroughly. The Smooth Iteration Count coloring exhibits small discontinuities at the iteration boundaries of a fractal, and a method for calculating an upper bound for the magnitude of the discontinuity is presented.

Average colorings are a family of coloring functions that use the decimal part of the smooth iteration count to interpolate between average sums. In addition to linear interpolation, a smooth Catmull-Rom spline interpolation method can be used.

The Triangle Inequality Average and Curvature Average colorings are presented and analyzed as examples of branching average colorings. Both colorings exhibit a similar tree-like branching structure, and the three mathematical properties that lead to this kind of structure are presented. A new coloring called the Stripe Average is introduced based on the behavior of the Triangle Inequality Average coloring.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Fundamental Concepts</b>	<b>5</b>
2.1	Iterations and Orbits . . . . .	5
2.2	Fractals . . . . .	6
2.2.1	Coloring and Palette functions . . . . .	7
2.2.2	Calculating Images of Fractals . . . . .	8
2.2.3	The Iteration Count Coloring . . . . .	8
2.3	Iteration Levels and Boundaries . . . . .	9
2.4	Desirable Properties of Colorings . . . . .	10
2.4.1	Performance . . . . .	10
2.4.2	Smoothness . . . . .	11
2.4.3	Degrees of Freedom . . . . .	12
<b>3</b>	<b>The Smooth Iteration Count Coloring</b>	<b>13</b>
3.1	Simplification . . . . .	13
3.1.1	Extending the Results . . . . .	14
3.2	The Continuous Iteration Count . . . . .	16
3.2.1	Continuity and Smoothness . . . . .	16
3.3	The Smooth Iteration Count . . . . .	18
3.3.1	Continuity and Smoothness . . . . .	20
3.3.2	Error Estimation . . . . .	22
<b>4</b>	<b>Branching Average Colorings</b>	<b>26</b>
4.1	Average Colorings . . . . .	26
4.1.1	Using Smooth Iteration Count for Interpolation . . . . .	27
4.2	The Triangle Inequality Average Coloring . . . . .	29

4.2.1	Behavior for Distant Starting Points . . . . .	32
4.2.2	Behavior in the Neighborhood of Origin . . . . .	33
4.3	The Curvature Average Coloring . . . . .	36
4.3.1	Behavior for Distant Starting Points . . . . .	37
4.4	Similarities and Differences . . . . .	38
4.4.1	Emergence of Branches . . . . .	40
4.4.2	Continuity and Smoothness . . . . .	41
4.5	The Stripe Average Coloring . . . . .	42
4.5.1	The Stripe Density Parameter . . . . .	42
4.5.2	Behavior in the Neighborhood of Origin . . . . .	43
<b>5</b>	<b>Svensk sammanfattning</b>	<b>45</b>
<b>A</b>	<b>Using Colorings on a Fractal Image</b>	<b>48</b>
A.1	Intermediate Images . . . . .	49
A.2	Conclusion . . . . .	57



# Chapter 1

## Introduction

This work studies the mathematics of selected algorithms that are used to color fractal images. These algorithms were developed to be instruments for *fractal art*. Javier Barrallo and Damien Jones describe the emergence of fractal art and the role of coloring algorithms in the following way [2].

During the 1980s, fractal enthusiasts began exploring fractals for their artistic merit, not for their mathematical significance. While mathematics was the tool, the focus was art. As the fractal equation itself was the most obvious mathematical element, fractal artists experimented with new equations, introducing hundreds of different fractal types. By carefully choosing parameters to refine form, color, and location, these explorers introduced the concept of fractal art.

After 1995, few new major fractal types have been introduced. This is because the newest innovations in fractal art do not come from changing the fractal equation, but from new ways of coloring the results of those equations. As these coloring algorithms move from simple to complex, fractal artists are often returning to the simpler, classical fractal equations. With the increased flexibility these sophisticated algorithms provide, there is even more room for personal artistic expression.

Countless coloring algorithms have been developed. The development process of a coloring algorithm usually combines experimentation to mathematical understanding and analysis of the visual results. However, thorough mathe-

mathematical understanding is not required in order to develop interesting and useful algorithms.

This work aims to gain better understanding of the selected coloring algorithms, or *colorings*. Their visual appearance is explained based on mathematical analysis. This analysis also led to the development of a new coloring, namely the *Stripe Average* coloring to be introduced in Section 4.5.

Although this work aims to present its contents rigorously and accurately, some of the results are expressed in a somewhat vague form. This is inherent to the applied nature of the subject. To name an example, the words "small" and "distant" are used loosely without an exact definition.

The reader is assumed to be familiar with calculus and complex analysis. Fractals are explained in the extent necessary to introduce the main concepts and notation, whereas details in the theory of chaotic dynamical systems from which fractals emerge fall outside the scope of this work.

The contents are organized as follows: Fundamental concepts and definitions necessary for the later chapters are introduced in Chapter 2. The classic escape time algorithm for calculating fractal images is extended to suit the needs of coloring functions. Heuristic criteria to evaluate the quality of a coloring are also presented.

Chapter 3 presents two continuous colorings that extend the discontinuous Iteration Count coloring. A detailed discussion of *the Smooth Iteration Count* coloring and a proof of its smoothness are presented. It exhibits small discontinuities at the iteration boundaries of a fractal, and a method for calculating an upper bound for the magnitude of the discontinuity is presented.

Chapter 4 first introduces *average colorings* followed by a discussion of linear and smooth Catmull-Rom spline interpolation methods. The Triangle Inequality Average and Curvature Average colorings are described and analyzed as examples of average colorings. Based on the behavior of the Triangle Inequality Average coloring, the Stripe Average coloring is then introduced.

Appendix A illustrates the construction process of a fractal image using the colorings discussed in the earlier chapters.

The thesis was carried out under the supervision of Professor Göran Högnäs. Kerry Mitchell provided some helpful comments.

# Chapter 2

## Fundamental Concepts

This chapter first introduces fundamental definitions that are the foundation for the following chapters. The calculation of images of divergent fractals is then discussed.

For convenience, the presentation of definitions is adapted for the needs of this work. For example, definitions are presented as confined to the complex plane instead of an arbitrary metric space.

### 2.1 Iterations and Orbits

The notation and definitions in this section are adapted from [3].

The  $n$ -fold composition of  $f$  with itself is denoted by  $f^n(z)$ . That is,

$$f^n(z) = f(f(\dots f(z)\dots))$$

where  $f$  is applied to  $z$   $n$  times. The points  $f(z)$ ,  $f^2(z)$ ,  $f^3(z)$ ,  $\dots$  are called the *iterates* of  $z$ . Repeatedly applying the function to the previous iterate is called *iteration*. Given an initial value  $z_0$  and a function  $f$ , the  $n$ th iterate  $f^n(z_0)$  is denoted  $z_n$ .

**Definition 2.1** (Orbit). *Given a function  $f : \mathbb{C} \mapsto \mathbb{C}$ , the (forward<sup>1</sup>) orbit of a point  $z \in \mathbb{C}$  is the set*

$$O(z) = \{z, f(z), f^2(z), \dots\}.$$

---

<sup>1</sup>It is not necessary to separate forward and backward orbits in this context because only the forward orbit is of interest for this work.

The orbit  $O(z)$  is said to be *periodic* if, for a given  $f$  and  $z$ , there exists a number  $n \in \mathbb{N}$  such that  $f^n(z) = z$ . An orbit contains infinitely many points if it is not periodic. An orbit  $O(z)$  *escapes* if the iterates converge to infinity.

When calculating fractals on a computer, a finite representation of an orbit is needed. This is done by limiting both the magnitude and the maximum number of elements to be contained in an orbit. Define the constants  $M$  and  $N_{\max}$  as the *bailout value* and the *maximum number of iterations*, respectively. Furthermore, define the set  $\mathbb{C}_{N_{\max}}$  as

$$\mathbb{C}_{N_{\max}} = \bigcup_{k=1}^{N_{\max}+1} \mathbb{C}^k.$$

**Definition 2.2** (Truncated Orbit). *Let an orbit*

$$O(z) = \{z, f(z), f^2(z), \dots\}$$

*and constants  $M$  and  $N_{\max} \in \mathbb{N}$  be given. Let  $\bar{N}$  denote the smallest nonnegative integer for which  $|f^{\bar{N}}(z)| > M$ , and define  $N = \min\{\bar{N}, N_{\max}\}$ . The truncated orbit  $O_T(z) \in \mathbb{C}_{N_{\max}}$  is the set*

$$O_T(z) = \{z, f(z), f^2(z), \dots, f^N(z)\}.$$

In addition to the initial point  $z$ , the truncated orbit contains  $N$  of the first  $N_{\max}$  iterates  $z_n$ . The number of elements in a truncated orbit is  $N + 1$ . It is noteworthy that  $N \leq N_{\max}$  and  $N$  is dependent of both the function  $f$  and the initial value  $z$ .

The following inequality describes the magnitude of the two last iterates  $z_{N-1}$  and  $z_N$  of a truncated orbit with respect to the bailout value  $M$ .

**Proposition 2.1** (Bailout Inequality). *Assume  $1 \leq N \leq N_{\max}$  and consider the points  $z_{N-1} \in O_T(z)$  and  $z_N \in O_T(z)$ . The bailout inequality is the inequality*

$$|z_{N-1}| \leq M < |z_N|. \quad (2.1)$$

According to Definition 2.2,  $N$  is the smallest integer for which  $M < |z_N|$ . Thus  $|z_{N-1}| \leq M$  and the Bailout Inequality follows.

## 2.2 Fractals

As an example, consider the function  $f : \mathbb{C} \mapsto \mathbb{C}$ ,

$$f(z) = z^p + c \quad (2.2)$$

where the constant  $p \in \mathbb{N}$ ,  $p \geq 2$ , and the *seed*  $c \in \mathbb{C}$ . The function  $f(z)$  defines the *dynamical system*

$$z_k = z_{k-1}^p + c. \quad (2.3)$$

This system is known to exhibit *chaotic* behavior; depending on the value of the constant  $c$  and the initial value  $z_0$ , the iterates behave differently. The intricate nature of this function can be illustrated by plotting some characteristics of the orbits when varying  $c$  or  $z_0$  over the complex plane.

### 2.2.1 Coloring and Palette functions

An *image* is represented as an  $m \times n$  matrix of discrete points referred to as *pixels*. To each pixel in the matrix is associated an RGB color [4, p. 572]. Let  $z_0$  denote the point corresponding to the position of a pixel in the complex plane. To calculate the RGB color of a pixel in a fractal image, the truncated orbit  $O_T(z_0)$  is first calculated. Then a *coloring function* is evaluated.

**Definition 2.3** (Coloring Function). *A coloring function is a function*

$$u : \mathbb{C}_{N_{\max}} \mapsto \mathbb{R} \quad (2.4)$$

*that maps a truncated orbit to the real line.*

The number of elements  $N + 1$  in the argument of  $u$  is variable. Implementations of coloring functions usually include a loop that is executed for all elements of  $O_T(z_0)$  (cf. Section 2.2.2). The coloring  $u(O_T(z_0))$  is often studied as a function of the point  $z_0$ , whereas certain coloring functions only use the last iterate  $z_N$  of  $O_T(z_0)$ . This is denoted  $u(z_N)$ .

A *color index function*  $I(u)$  maps the value  $u$  of the coloring function to a *color index*. The color index is either contained in or clamped to the interval  $[0, 1]$ . The *color palette function*  $P(I)$  is then used to map the color index to the RGB color space. The red, green and blue components of an RGB color can be represented as a point in  $[0, 1]^3$ .

**Definition 2.4** (Palette Function). *A palette function  $P : \mathbb{R} \mapsto \mathbb{R}^3$  maps the color index  $I$  to the RGB color space. Its domain is  $[0, 1]$  and range  $[0, 1]^3$ . The domain can be extended to the real line by using the decimal part of  $I$ .*

The color index function is merely used as a means to adjust the visual appearance of a coloring without the need to change the palette function.

## 2.2.2 Calculating Images of Fractals

Iterating the function defined by equation 2.3, the algorithm presented below renders an image of a *Julia set* [1, p. 258]. This algorithm extends the escape time algorithm presented in [1, p. 252] by introducing coloring, index and palette functions.

The following operations are carried out for each pixel in the  $m \times n$  matrix representing the image.

1. Set  $z_0$  to correspond to the position of the pixel in the complex plane.
2. Calculate the truncated orbit by iterating the formula  $z_n = f(z_{n-1})$  starting from  $z_0$  until either
  - $|z_n| > M$ , or
  - $n = N_{\max}$ ,

where  $N_{\max}$  is the maximum number of iterations.

3. Using the coloring and color index functions, map the resulting truncated orbit to a color index value.
4. Determine an RGB color of the pixel by using the palette function.

If  $|z_n| > M$ , the pixel is an *outside* point. Otherwise, if  $n = N_{\max}$ , the point is *inside*.

The *Mandelbrot set* can be calculated by setting  $z_0 = 0$  and  $c$  to correspond to the position of the pixel in the complex plane in step 1. In addition to the system 2.2 mainly used in this work, many other functions exhibit chaotic behavior. Also various other types of fractals exist and are calculated with modified or completely different algorithms, but are not discussed here.

## 2.2.3 The Iteration Count Coloring

One of the most intuitive ways to specify the color of a pixel is to choose the color proportional to the number of iterations  $N$ . The Iteration Count coloring is illustrated in Figure 2.1. The coloring function simply returns the number of elements in the truncated orbit divided by 20. The coloring is applied to the

system 2.3 with  $c = 0.5 + 0.25i$  and  $p = 2$ . The bailout value was chosen to be  $M = 10$  and the initial value  $z_0$  was set to the value corresponding to the pixel position. The real and imaginary parts vary between -1.333 and 1.333 in the figure. The color palette function uses the RGB color palette shown in figure 2.2. The color index function

$$I(u) = k(u - u_0),$$

where  $k = 2.5$  and  $u_0 = 0$ , was used.

This example fractal is used throughout this work to illustrate different colorings. The coefficient  $k$  is used to adjust the colors in the image to a suitable range. The offset  $u_0$  is used to adjust the relative location of light and dark areas. The outside areas of fractals are of most interest for artistic purposes. This fractal consists only of outside points and is consequently well suited as an example. The ideas presented can easily be adapted to many other divergent fractals as well.

## 2.3 Iteration Levels and Boundaries

The following definition assumes that a function  $f$  and a bailout value  $M$  are given.

**Definition 2.5** (Iteration Levels and Boundaries). *For  $n \geq 1$ , the  $n$ th iteration level is the set*

$$L_n = \{z_0 \in \mathbb{C} : |f^{n-1}(z_0)| < M < |f^n(z_0)|\}.$$

*The  $n$ th iteration boundary is the set*

$$B_n = \{z_0 \in \mathbb{C} : |f^{n-1}(z_0)| = M\}.$$

*For  $n = 0$ , define*

$$L_0 = \{f(z_0) : z_0 \in L_1\}$$

*and*

$$B_0 = \{f(z_0) : z_0 \in B_1\}.$$

The iteration levels correspond to the areas colored with same color in Figure 2.1. The boundaries lie between the iteration levels. A point in  $L_N \cup B_N$  exceeds the bailout  $M$  after  $N$  iterations.

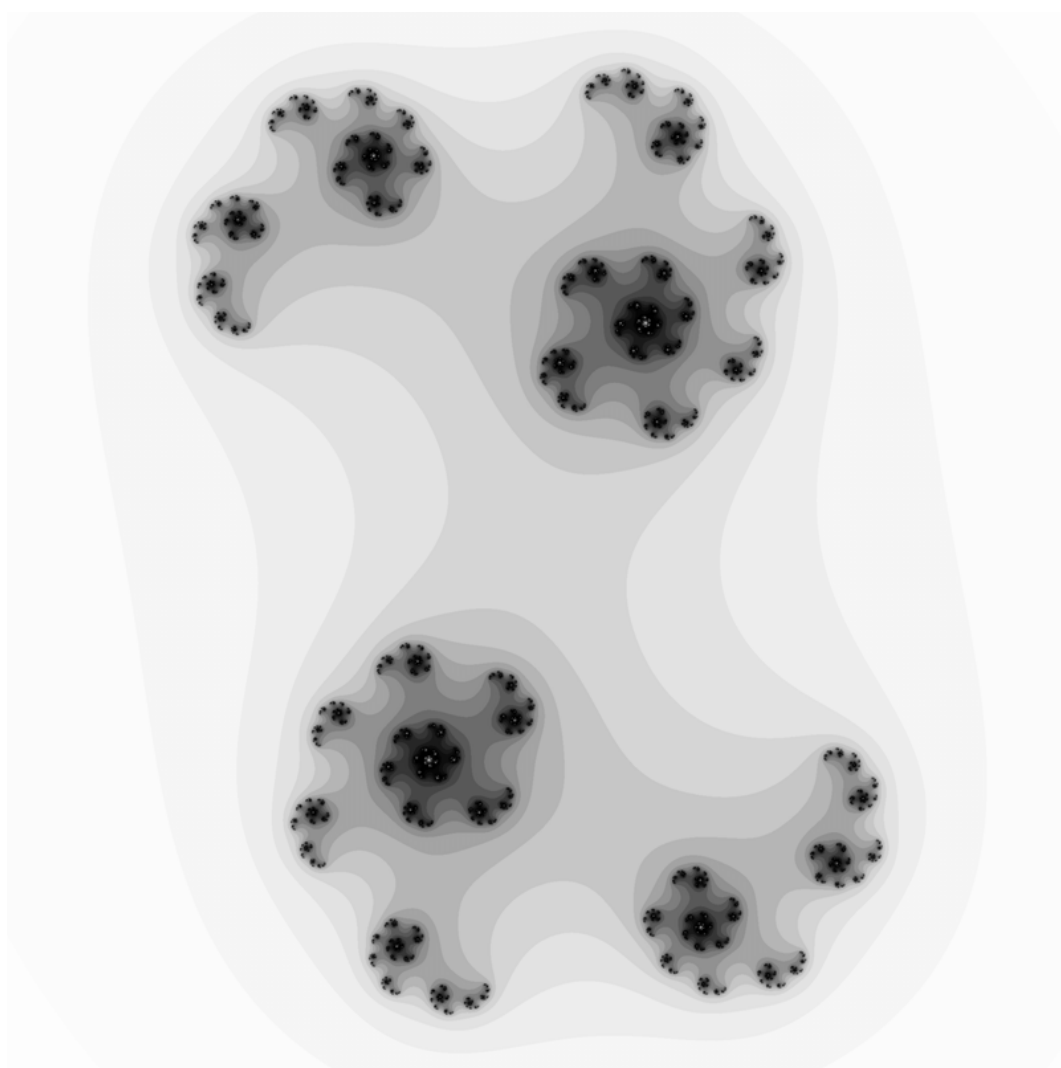


Figure 2.1: Illustration of the Iteration Count coloring.

## 2.4 Desirable Properties of Colorings

The following subsections present criteria to evaluate the goodness and usefulness of a coloring.

### 2.4.1 Performance

The fewer calculations needed to determine the color of a pixel, the better. At the moment of writing, rendering times for fractal images on a normal PC are usually manageable. For example, rendering the final image constructed in Appendix A using both cores of a 2.17MHz Intel Core 2 processor took 8 minutes and 28



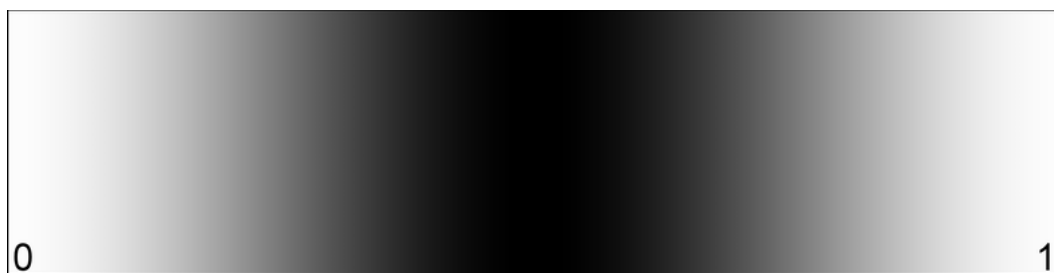


Figure 2.2: The RGB color palette used in Figure 2.1. The colors are indexed from 0 (left) to 1 (right).

seconds. The image was rendered using resolution  $2000 \times 1225$  and normal anti-aliasing settings.

A fractal animation sequence can contain thousands of frames and each frame must be rendered separately. Consequently, rendering times for fractal animation are orders of magnitude greater than rendering times for images. Rendering a short animation sequence can take from hours to days and thus performance remains to be a problem.

## 2.4.2 Smoothness

It is often desirable that the colors of a fractal image change *smoothly*, that is, without noticeable edges. In addition to any discontinuities that are very easy to see, the human eye also identifies discontinuity in the rate of change of colors. Continuity is usually sufficient if the discontinuity in the derivative is small. However, smoother appearance requires the coloring function to be continuously differentiable.

**Definition 2.6** (Smooth Coloring). *A coloring  $u$  is smooth in  $z_0$  if  $u(O_T(z_0)) \in C^1(\mathbb{C})$ , that is, if both  $u(O_T(z_0))$  and*

$$\frac{du(O_T(z_0))}{dz_0}$$

*are continuous [5, Remark 3.1.2, p. 251].*

Colorings are discontinuous in the neighborhood of the periodic points  $P$  of a fractal set. Define

$$P_r = \{z \in \mathbb{C} : \exists p \in P \bullet |z - p| < r\}.$$

The set  $P_r$  includes a neighborhood of radius  $r$  to each periodic point in  $P$ . In this work, a coloring is said to be differentiable if it is differentiable in the set  $\{z \in \mathbb{C} : |z| \leq M\} \setminus P_r$  for a small  $r > 0$ . Thus the neighborhood of the periodic points is ignored when considering the differentiability of a coloring.

Discontinuity usually appears at the iteration boundaries because the number of iterations changes at the boundary. The smooth iteration count and interpolation can effectively be used to eliminate discontinuities, as will be discussed in Chapter 4.

### 2.4.3 Degrees of Freedom

A *parameter* is a variable in the coloring function that affects the visual appearance of the coloring. Parameters offer degrees of freedom for adjusting the visual appearance of the coloring.

Parameters that give smooth coloring only for specific (e.g. integer) values appear frequently in applications. Real parameters that continuously affect the coloring are especially useful for creating motion in fractal animation.

# Chapter 3

## The Smooth Iteration Count Coloring

The smooth iteration count is a smooth equivalent of the iteration count. Both attain integer values on the iteration boundaries, but the smooth iteration count varies smoothly between the iteration boundaries whereas the iteration count changes in a stepwise manner.

The smooth iteration count is useful for creating smooth and soft color transitions in fractal images. Its decimal part can also be used for interpolation in order to introduce continuity and smoothness to various other colorings.

### 3.1 Simplification

Consider the system 2.3. The approximation

$$z_n = z_{n-1}^p + c \approx z_{n-1}^p \quad (3.1)$$

is valid for  $|z_{n-1}^p| \gg |c|$ . This gives the formula

$$z_n = z_{n-1}^p. \quad (3.2)$$

Denoting  $z_0 = r_0 e^{i\theta_0}$  and applying 3.2 iteratively gives

$$z_n = r_0^{p^n} e^{ip^n \theta_0}, \quad (3.3)$$

or

$$|z_n| = r_0^{p^n}. \quad (3.4)$$

The bailout inequality 2.1 states that  $M < |z_N|$ . The function  $|f(z)| = |z|^p$  is strictly increasing for  $|z| > 1$ . Assuming  $M > 1$  and applying the formula  $|z_N| = |z_{N-1}|^p$  to the inequality  $|z_{N-1}| \leq M$  gives  $|z_N| \leq M^p$ . Thus the last iterate  $|z_N|$  fulfills the inequality

$$M < |z_N| \leq M^p. \quad (3.5)$$

Figure 3.1 shows the iteration count coloring applied to the system 3.2. The figure shows the square bounded by the points  $-4 - 4i$  and  $4 + 4i$  in  $\mathbb{C}$ . The black disk  $|z| \leq 1$  in the middle of the image consists of inside points. Consequently, only the distances  $r_0 > 1$  are relevant for this system.

For the system 2.3, it can be shown that all orbits  $O(z)$  escape for  $|z| > 0.5 + \sqrt{0.25 + |c|}$  [1, p. 258]. Assuming  $|c| \leq 2$ , all orbits for  $|z| > 2$  escape so it is reasonable to confine  $r_0$  to values greater than 2.

The approximation 3.1 offers an accurate enough simplification of the system 2.3 for large  $r_0$ . Most importantly, the  $n$ th iterate can be expressed in a very compact form according to equation 3.3, whereas the system 2.3 leads to polynomials of increasing complexity that are difficult to work with.

### 3.1.1 Extending the Results

The Continuous and Smooth Iteration Count colorings are deduced for the simplified system 3.2. It will be shown in Section 3.3.2 that the Smooth Iteration Count coloring is almost smooth also when applied to the system 2.3.

When applied to the system 3.2, the Continuous and Smooth Iteration Count colorings are independent of the argument  $\theta_0$  of the initial point  $z_0 = r_0 e^{i\theta_0}$ . Thus it is sufficient to consider the continuity and smoothness with respect to the initial radius  $r_0$ .

Equation 3.3 allows the representation of a coloring  $u(r_N)$  as a function of  $r_0$ , namely

$$u(r_N) = u(r_0^{p^N}).$$

Expression  $u(r_0^{p^N})$  is often used for deducing results for the system 3.2. However, when extending the results from the system 3.2 to the system 2.3, it is necessary to use  $u(r_N)$  because the simplification 3.1 is likely to be valid only for the last iteration. For this reason, also the discussion of continuity and smoothness of

the colorings operate on  $u(r_N)$ . The requirement  $r_0 > 2$  is also replaced by the requirement  $r_N > 2$ .

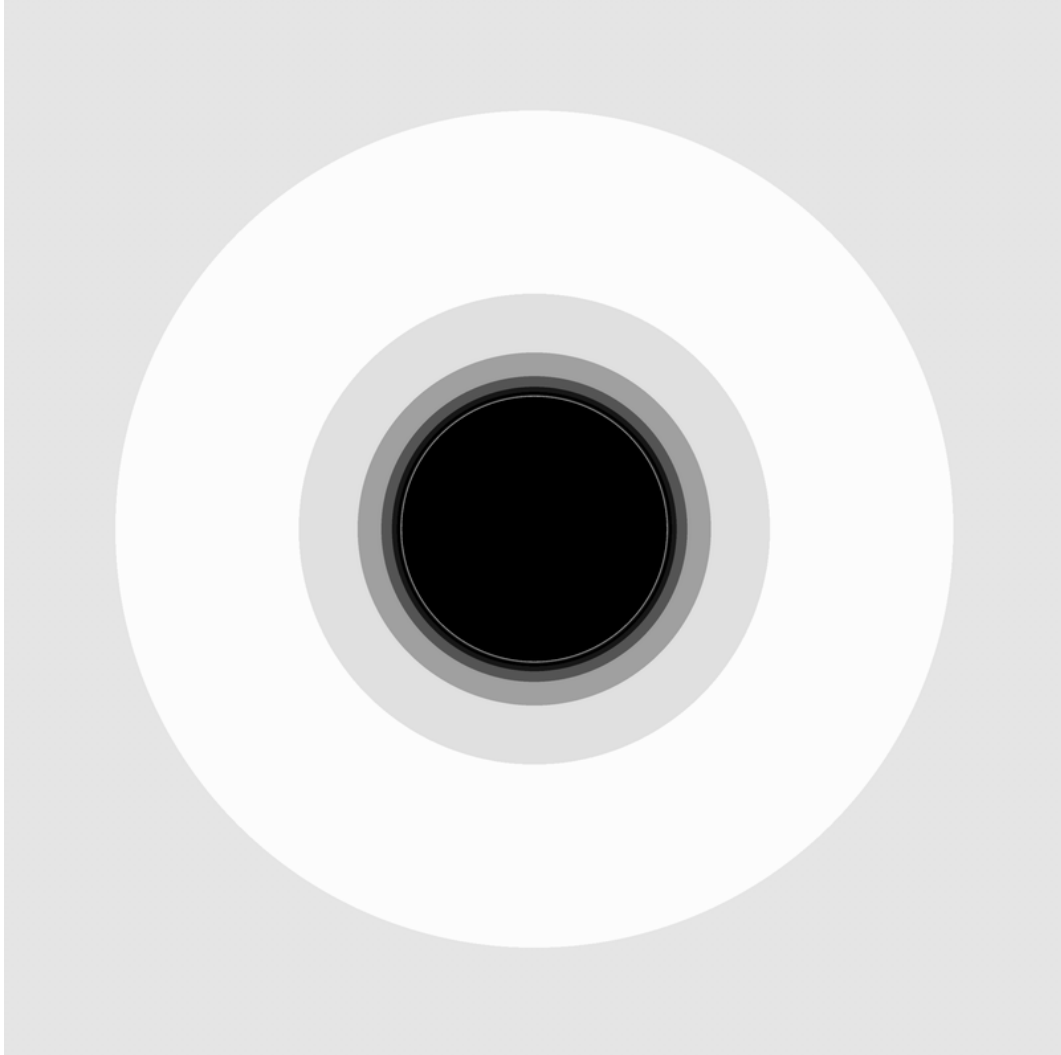


Figure 3.1: The iteration count coloring applied to the system 3.2.

It is noteworthy that for any outside point  $z_0$ , its iterates  $z_1, z_2, \dots$  start to grow at an increasingly fast rate as soon as they escape outside the area where system 2.3 exhibits chaotic behavior. Formally, for an arbitrary bound  $R$  there is a number  $n$  such that the iterates  $z_n, z_{n+1}, \dots$  are outside the disk  $|z| \leq R$ . The requirement  $r_0 > 2$  and the assumption made in the next chapter that  $r_0$  is large are valid for the iterates  $z_n, z_{n+1}, \dots$ . When extending the results to hold for the system 2.3, the iterates  $z_0, \dots, z_{n-1}$  can be omitted because the Continuous and Smooth Iteration Count colorings are functions of the magnitude  $r_N$  of the last

iterate only. Thus the conclusions made are also valid for outside points  $z_0$  for which  $r_0 \leq 2$ .

## 3.2 The Continuous Iteration Count

A continuous but not Smooth Iteration Count coloring is first presented. This coloring exhibits noticeable edges at the iteration boundaries. It is presented in order to illustrate and emphasize the requirement for smoothness.

**Definition 3.1** (Continuous Iteration Count Coloring). *The Continuous Iteration Count coloring is the function*

$$u(r_N) = N + \frac{M^p - r_N}{M^p - M}. \quad (3.6)$$

According to the bailout inequality 3.5, the magnitude  $r_N$  of the last iterate varies between  $M$  and  $M^p$ . The decimal part of the continuous iteration count is given by the quotient

$$\frac{M^p - r_N}{M^p - M}.$$

This expression can be interpreted as linear interpolation in the interval  $[M, M^p]$ . It attains the value 0 when  $r_N = M^p$  and 1 when  $r_N = M$ .

### 3.2.1 Continuity and Smoothness

The following theorem describes the continuity and smoothness of the Continuous Iteration Count coloring when applied to the system 3.2.

**Theorem 3.1.** *The Continuous Iteration Count coloring 3.6 attains values equal to the iteration count at the iteration boundaries. It is continuous but not smooth with respect to  $r_0$ .*

*Proof.* Continuity on the iteration levels follows from the continuity of the expression 3.6. The following shows that it is continuous also in the neighborhood of the iteration boundaries.

Consider a point  $z_0 \in B_N$ . The definition of the iteration boundary gives

$$|z_{N-1}| = M$$

or, combining this with 3.4,

$$|z_N| = M^p. \quad (3.7)$$

Substituting to 3.6 gives

$$\begin{aligned} u(M^p) &= N + \frac{M^p - M^p}{M^p - M} \\ &= N. \end{aligned}$$

This also shows that the continuous iteration count coincides with the iteration count at the iteration boundaries.

Now consider another point  $\hat{z}_0 \in L_{N-1}$  such that  $|\hat{z}_{N-1}| = M + \varepsilon$  for a small  $\varepsilon > 0$ . This orbit escapes outside the bailout  $M$  at the  $(N - 1)$ th iteration. Denote  $\varepsilon \downarrow 0$  as  $\varepsilon \rightarrow 0$  and  $\varepsilon > 0$ . Substituting  $\hat{r}_{N-1} = M + \varepsilon$  to 3.6 gives

$$\begin{aligned} u(\hat{r}_{N-1}) &= N - 1 + \frac{M^p - (M + \varepsilon)}{M^p - M} \\ &\rightarrow N - 1 + 1 \\ &= N \end{aligned}$$

as  $\varepsilon \downarrow 0$ . Thus 3.6 is continuous in the neighborhood of the iteration boundaries.

Using Equation 3.4,  $u(r_N)$  can be expressed as

$$u(r_0^{p^N}) = N + \frac{M^p - r_0^{p^N}}{M^p - M}.$$

The derivative can now be calculated as

$$\frac{du}{dr_0} = -\frac{p^N r_0^{p^N-1}}{M^p - M}.$$

for  $z_0$ . For the point  $\hat{z}_0$ , the derivative is

$$\frac{du}{dr_0} = -\frac{p^{N-1} r_0^{p^{N-1}-1}}{M^p - M}$$

The derivatives differ by a factor of  $pr_0^{p^{N-1}(p-1)}$  so 3.6 is not smooth in the neighborhood of the iteration boundaries.  $\square$

Figure 3.2 illustrates the continuous iteration count as a function of the magnitude of the initial point  $r_0$ . Figure 3.3 shows the Continuous Iteration Count coloring applied to the example fractal described in Section 2.2.3. Although the iteration boundaries are softer than in Figure 2.1, they are still clearly distinguishable despite the continuity of the coloring.

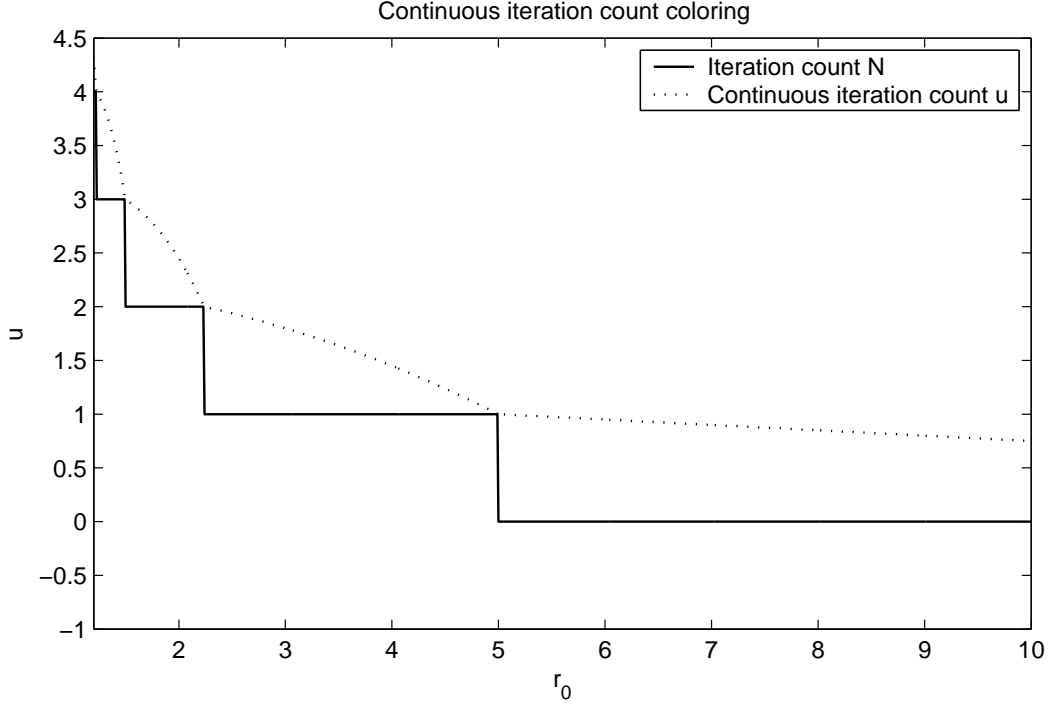


Figure 3.2: The continuous iteration count  $u(r_0^{p^N})$  as a function of  $r_0$ . The curves are calculated for  $M = 5$  and  $p = 2$ .

### 3.3 The Smooth Iteration Count

The following discussion is based on the simplification 3.4. After the smooth iteration count expression has been deduced for the simplified system, the result is shown to be quite accurate for the original system as well.

Applying Equation 3.4 to the bailout inequality gives

$$M < r_0^{p^N} \leq M^p. \quad (3.8)$$

By taking a logarithm twice over this inequality, the bounds for  $N$  can be solved;

$$\begin{aligned} \ln M &< p^N \ln r_0 \leq p \ln M \\ \ln \frac{\ln M}{\ln r_0} &< N \ln p \leq \ln p + \ln \frac{\ln M}{\ln r_0} \\ \frac{1}{\ln p} \ln \frac{\ln M}{\ln r_0} &< N \leq 1 + \frac{1}{\ln p} \ln \frac{\ln M}{\ln r_0}. \end{aligned} \quad (3.9)$$

The number of iterations  $N$  is an integer and its value changes by one when  $r_0$  crosses an iteration boundary. The function

$$v(r_0) = 1 + \frac{1}{\ln p} \ln \frac{\ln M}{\ln r_0} \quad (3.10)$$



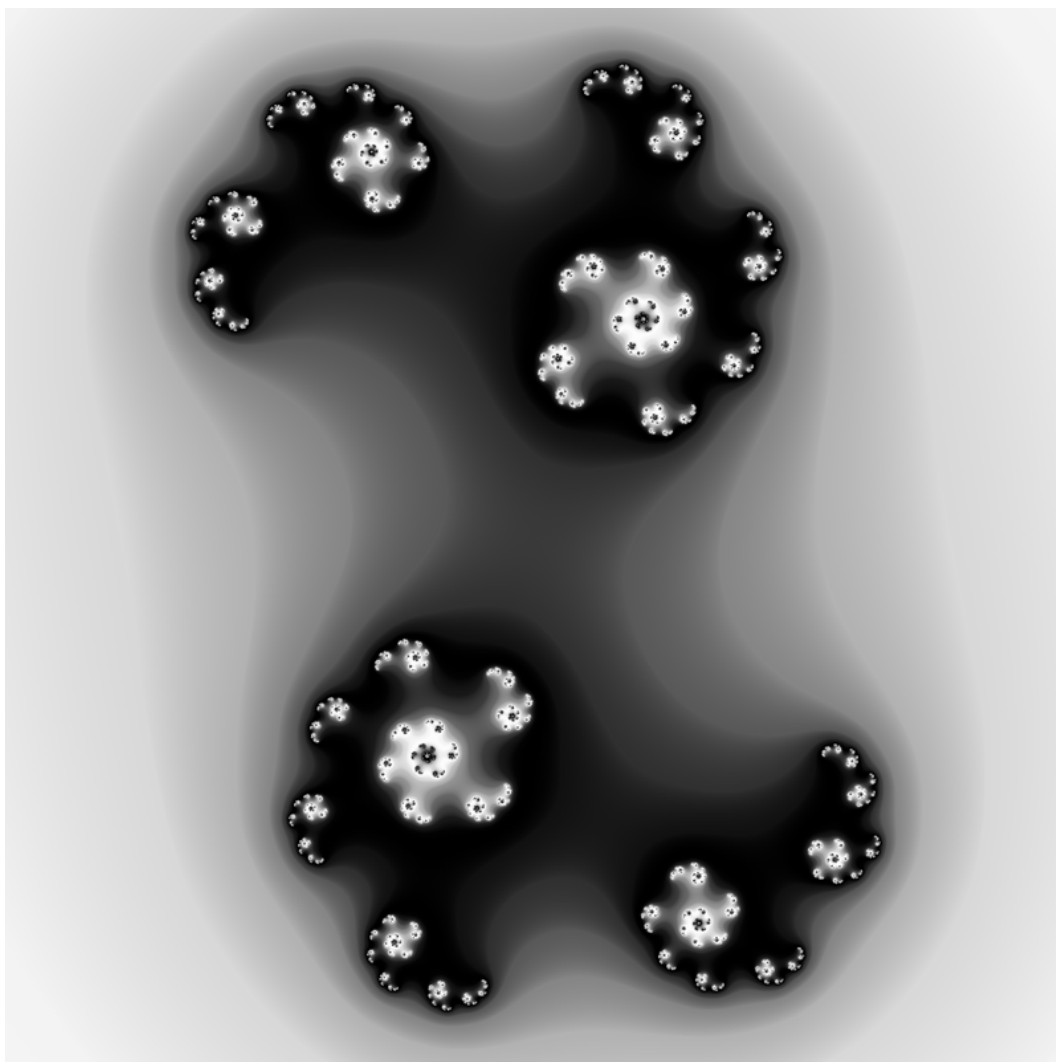


Figure 3.3: The Continuous Iteration Count coloring applied to the example fractal.

attains positive real values for  $2 < r_0 \leq M$ . Because  $N = N(r_0)$  is bounded by the inequality  $v(r_0) - 1 < N(r_0) \leq v(r_0)$ , it is apparent that  $v(r_0)$  must follow  $N(r_0)$ . Also,  $v(r_0) = N(r_0)$  holds at the iteration boundaries. This is illustrated in Figure 3.4.

Using 3.4,  $r_0$  can be expressed as a function of the last iteration  $r_N$ . Substi-

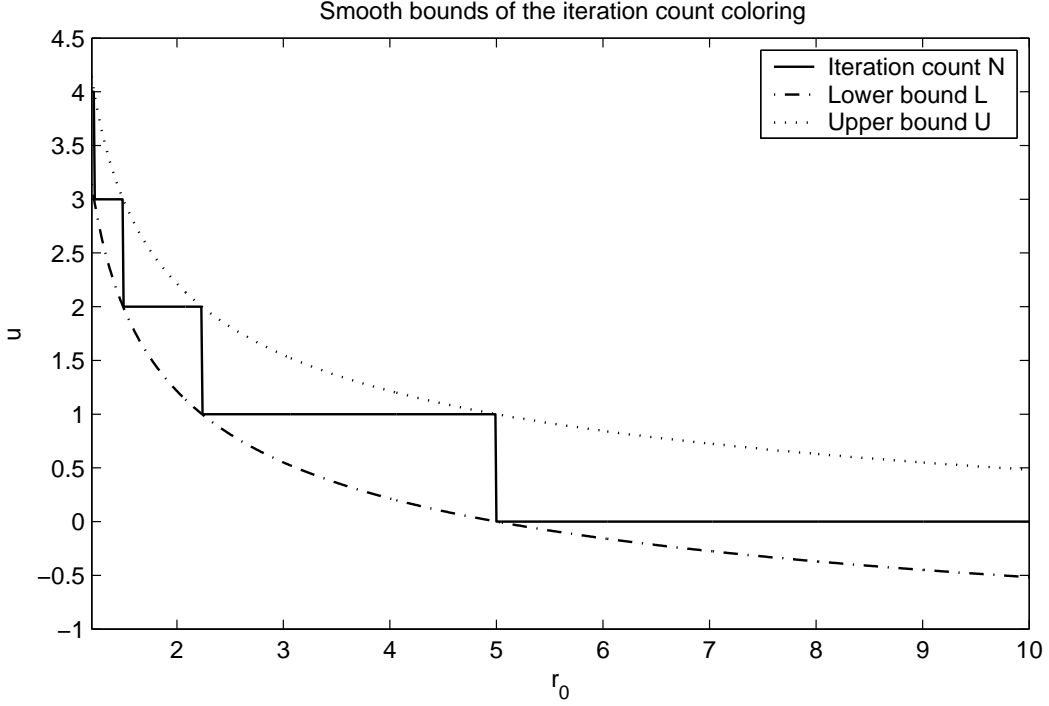


Figure 3.4: The iteration count  $N$  is bounded by the upper bound  $U = v(r_0)$  and the lower bound  $L = v(r_0) - 1$ . The curves are plotted for  $M = 5$  and  $p = 2$ .

tuting  $r_0 = r_N^{p^{-N}}$  to  $v(r_0)$  and denoting  $u(r_N) = v(r_N^{p^{-N}})$  gives

$$\begin{aligned}
 u(r_N) &= 1 + \frac{1}{\ln p} \ln \frac{\ln M}{\ln r_N^{p^{-N}}} \\
 &= 1 + \frac{1}{\ln p} \ln \frac{p^N \ln M}{\ln r_N} \\
 &= 1 + \frac{1}{\ln p} (N \ln p + \ln \frac{\ln M}{\ln r_N}) \\
 &= N + 1 + \frac{1}{\ln p} \ln \frac{\ln M}{\ln r_N}.
 \end{aligned}$$

**Definition 3.2** (Smooth Iteration Count). *The Smooth Iteration Count coloring is the function*

$$u(r_N) = N + 1 + \frac{1}{\ln p} \ln \frac{\ln M}{\ln r_N}. \quad (3.11)$$

### 3.3.1 Continuity and Smoothness

**Theorem 3.2.** *The Smooth Iteration Count coloring 3.11 attains values equal to the iteration count at the iteration boundaries. It is continuous and smooth with respect to  $r_0$ .*

*Proof.* In analogy with the proof of Theorem 3.1,  $r_N = M^p$  for an arbitrary point  $z_0 \in B_N$ . Substituting to 3.11 gives

$$\begin{aligned} u(M^p) &= N + 1 + \frac{1}{\ln p} \ln \frac{\ln M}{\ln M^p} \\ &= N + 1 + \frac{1}{\ln p} (\ln \frac{\ln M}{\ln M} - \ln p) \\ &= N + 1 - 1 + \frac{1}{\ln p} \ln 1 \\ &= N. \end{aligned}$$

All outside points on any of the iteration levels  $L_N$ ,  $N = 1, 2, \dots$ , are mapped by  $f^N$  to the last iteration level  $L_0$ . Similarly, all bounds  $B_N$  are mapped to the last bound  $B_0$ . For any fixed outside point, the number of iterations  $N$  is finite so the mapping  $f^N$  is well defined and smooth because  $f$  is smooth.

Also  $u(r_N)$  is smooth for a fixed  $N$ . Smoothness on any outside point  $z_0 \in L_N$ ,  $N = 1, 2, \dots$ , contained in any of the iteration levels then follows from the smoothness of  $f^N$ .

The number of iterations  $N$  changes at the iteration boundaries. Because points in  $L_{N-1}$  and  $B_N$  are mapped smoothly to  $L_0$  and  $B_1$ , respectively, the continuity of  $u$  and  $\frac{du}{dr_0}$  needs to be shown only between  $L_0$  and  $B_1$ .

Consider the points  $z_0 \in B_1$  and  $\hat{z}_0 \in L_0$  such that  $r_0 = M$  and  $\hat{r}_0 = M + \varepsilon$ . It was already shown that  $u(r_N) = N$  for  $r_N = M^p$ , so  $u(r_1) = 1$ . Substituting  $\hat{r}_0 = M + \varepsilon$  to 3.11 gives

$$\begin{aligned} u(M + \varepsilon) &= 0 + 1 + \frac{1}{\ln p} \ln \frac{\ln M}{\ln(M + \varepsilon)} \\ &\rightarrow 1 + \frac{1}{\ln p} \ln 1 \\ &= 1 \end{aligned}$$

as  $\varepsilon \downarrow 0$ . Thus 3.11 is continuous in the neighborhood of iteration boundaries.

Differentiating 3.11 with respect to  $r_N$  gives

$$\begin{aligned} \frac{du}{dr_N} &= \frac{1}{\ln p} \cdot \frac{1}{\ln M / \ln r_N} \cdot \frac{\ln M}{r_N (\ln r_N)^2} \\ &= \frac{1}{r_N \ln r_N \ln p}. \end{aligned} \tag{3.12}$$

Using the chain rule, the derivative with respect to  $r_0$  can be written

$$\frac{du}{dr_0} = \frac{du}{dr_N} \cdot \frac{dr_N}{dr_0}.$$

For  $r_0 = M$ , the last iterate is  $r_N = r_1 = M^p$ . Thus

$$\frac{dr_1}{dr_0} = pM^{p-1}$$

and

$$\frac{du}{dr_0} = \frac{pM^{p-1}}{M^p \ln M^p \ln p} \quad (3.13)$$

$$= \frac{1}{M \ln M \ln p}. \quad (3.14)$$

For  $\hat{r}_0 = M + \varepsilon$ , the number of iterations is  $N = 0$ . The derivatives now evaluate to

$$\frac{dr_0}{dr_0} = 1$$

and

$$\begin{aligned} \frac{du}{dr_0} &= \frac{1}{(M + \varepsilon) \ln(M + \varepsilon) \ln p} \\ &\rightarrow \frac{1}{M \ln M \ln p} \end{aligned}$$

as  $\varepsilon \downarrow 0$ . That is,  $\frac{du}{dr_0}$  is continuous between  $B_1$  and  $L_0$ .  $\square$

The Smooth Iteration Count coloring is illustrated in Figure 3.5. The iteration boundaries are now indistinguishable from the figure. This is a consequence of the fact that the derivative  $\frac{du}{dr_0}$  is continuous at the iteration boundaries for the Smooth Iteration count coloring but discontinuous for the Continuous Iteration Count coloring.

The  $N - 1$  first iterations only increase the iteration count, whereas the decimal part of  $u(r_N)$  only depends on  $r_N$ . Thus the  $N - 1$  first iterations are independent of the approximation 3.1, which is important only for the last iteration. The statement made above about the continuity of  $f^N$  holds for the function  $f(z) = z^p$ , that defines the simplified system 3.2, as well as for  $f(z) = z^p + c$ . Theorem 3.2 can thus be extended to the system  $z_{n+1} = z_n^p + c$  as long as the approximation 3.1 is valid for the last iteration.

### 3.3.2 Error Estimation

**Theorem 3.3.** *Applying the Smooth Iteration Count coloring given by 3.11 to the system 2.2 results in a discontinuity between the iteration levels. the magnitude*

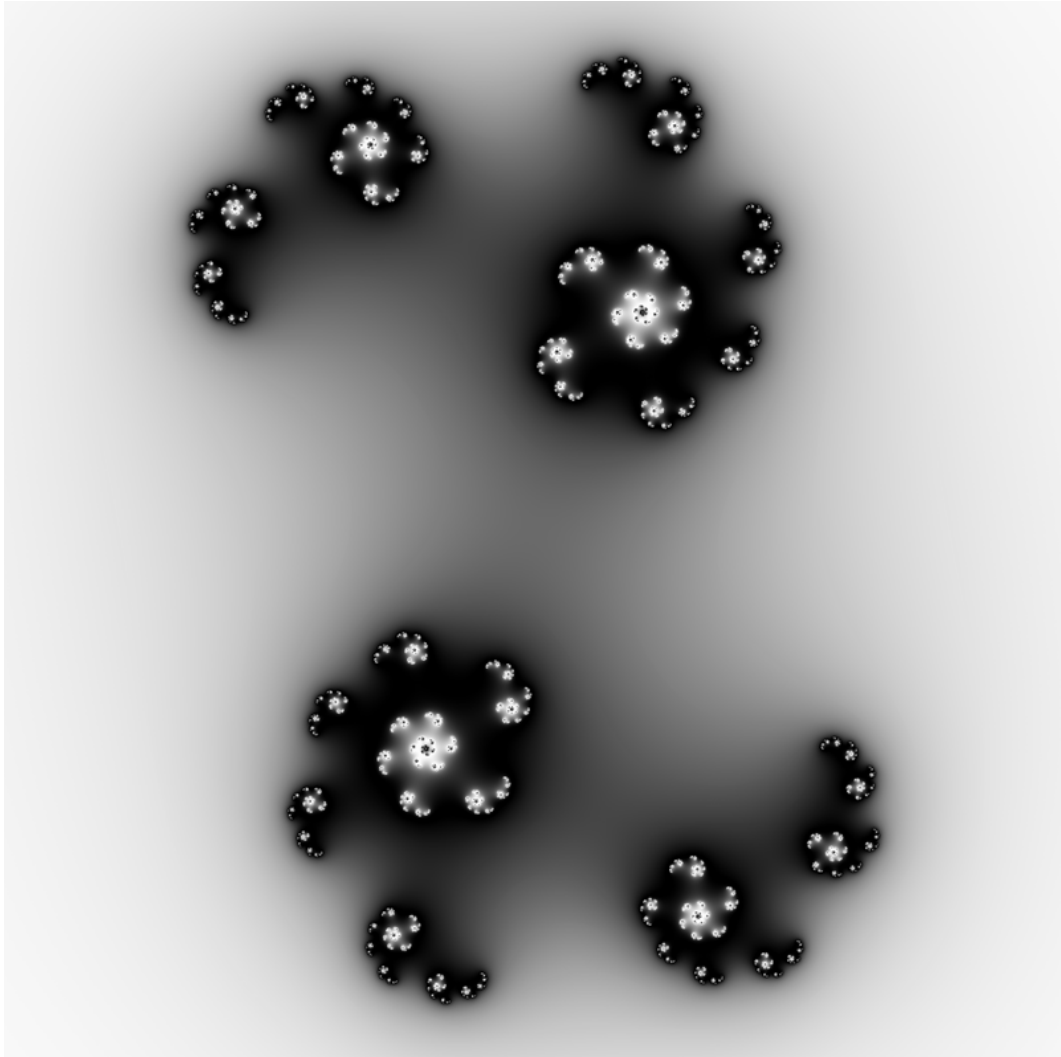


Figure 3.5: Smooth Iteration Count coloring applied to the example fractal.

of the discontinuity is bounded by

$$\Delta_{\max} = \frac{1}{\ln p} \ln \frac{\ln M^p}{\ln(M^p - |c|)}. \quad (3.15)$$

*Proof.* Consider again the two points  $z_0 \in B_1$ ,  $r_0 = M$ , and  $\hat{z}_0, \hat{r}_0 = M + \varepsilon$ . If the system 2.3 is iterated instead of the system 3.2, a discontinuity is introduced between the iteration levels. For  $z_0$ , the magnitude of the first iterate is  $r_1 = |z_0^p + c| \approx M^p$ , so  $N = 1$ . This gives the smooth iteration count

$$\begin{aligned} u(r_1) &= 2 + \frac{1}{\ln p} \ln \frac{\ln M}{\ln |z_0^p + c|} \\ &= 1 + \frac{1}{\ln p} \ln \frac{\ln M^p}{\ln |z_0^p + c|} \end{aligned} \quad (3.16)$$

For  $\hat{z}_0$ ,  $N = 0$  and

$$\begin{aligned} u(\hat{r}_0) &= 1 + \frac{1}{\ln p} \ln \frac{\ln M}{\ln(M + \varepsilon)} \\ &\rightarrow 1 + \frac{1}{\ln p} \ln \frac{\ln M}{\ln M} \\ &= 1. \end{aligned} \tag{3.17}$$

The magnitude  $\Delta$  of the discontinuity is given by

$$\begin{aligned} \Delta &= |u(r_1) - u(\hat{r}_0)| \\ &= \left| 1 + \frac{1}{\ln p} \ln \frac{\ln M^p}{\ln |z_0^p + c|} - 1 \right| \\ &= \left| \frac{1}{\ln p} \ln \frac{\ln M^p}{\ln |z_0^p + c|} \right|. \end{aligned} \tag{3.18}$$

The triangle inequality gives bounds for  $|z_0^p + c|$ , namely

$$|z_0^p| - |c| \leq |z_0^p + c| \leq |z_0^p| + |c|,$$

or, substituting  $r_0 = M$ ,

$$M^p - |c| \leq |z_0^p + c| \leq M^p + |c|.$$

The magnitude of the discontinuity  $\Delta$  is to be maximized on this interval in order to find an upper bound for it. Denoting  $x = |z_0^p + c|$  and differentiating  $\Delta$  gives

$$\frac{d\Delta(x)}{dx} = \frac{1}{x \ln x \ln p}.$$

Assuming  $x > 1$ , the derivative is strictly decreasing and 3.18 attains its maximum value at  $|z_0^p + c| = M^p - |c|$ . Thus the maximum error due to the approximation 3.1 is

$$\Delta_{\max} = \frac{1}{\ln p} \ln \frac{\ln M^p}{\ln(M^p - |c|)}.$$

□

For the example fractal with parameters  $p = 2$ ,  $M = 10$  and  $|c| = 0.5590$ , 3.15 gives the maximum error of  $\Delta_{\max} = 0.00176$ . This is illustrated in Figure 3.6. Error equal to 0 is colored white whereas error equal to 0.00176 is colored black.

Suppose the palette function maps the unit interval  $[0, 1]$  to, say, 500 discrete colors and the index function is  $I(u) = u$ . Then the colors are separated by steps

of  $1/500 = 0.002$  in  $u$ . This is greater than the magnitude of the discontinuity  $\Delta_{\max} = 0.00176$ , so the extension from the simplified system 3.2 to the chaotic system 2.3 indeed only leads to an insignificant error. In addition, the accuracy can be further improved by increasing the value of  $M$ .

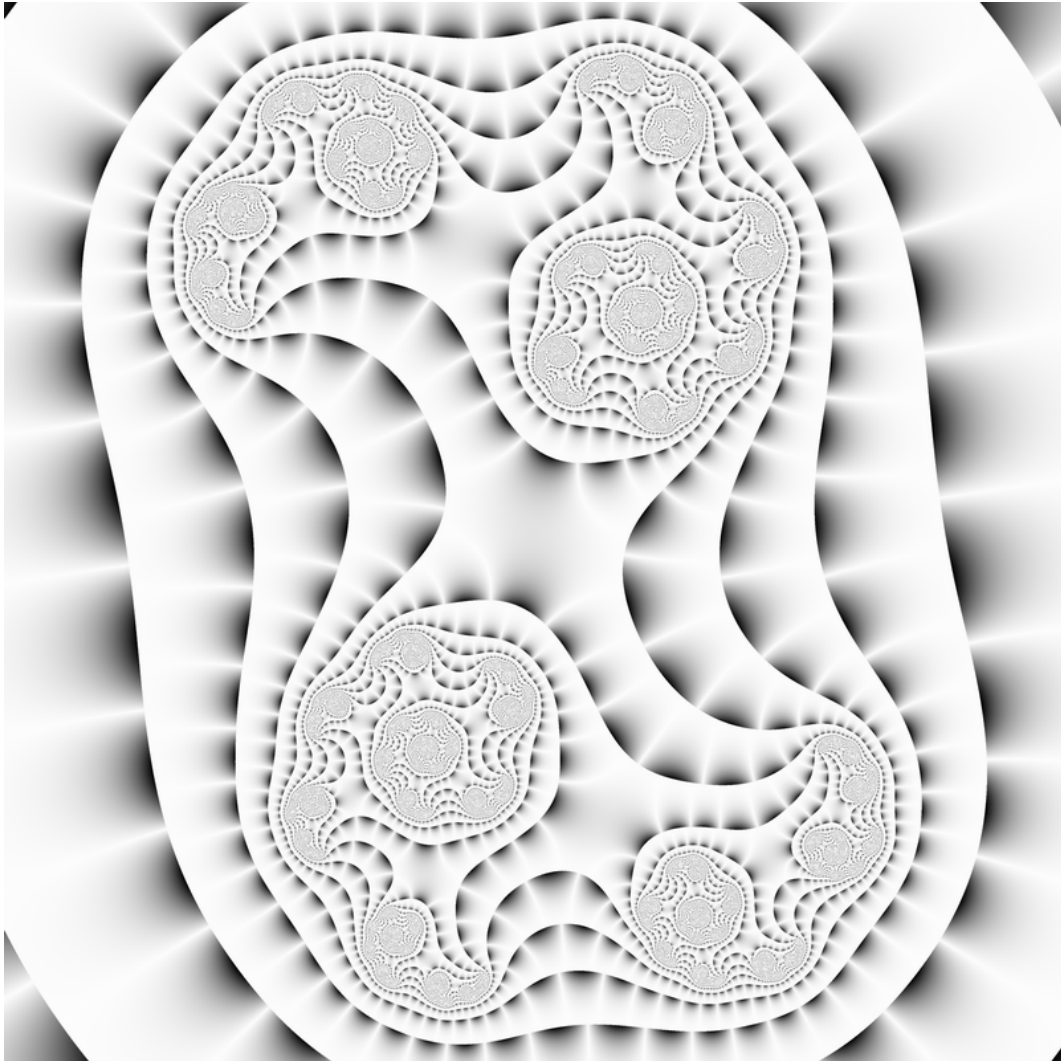


Figure 3.6: The error in the smooth iteration count due to the approximation 3.1.

# Chapter 4

## Branching Average Colorings

This chapter introduces the Triangle Inequality Average and Curvature Average colorings. Their properties, similarities and differences are analyzed and, based on this analysis, a definition of *branching average colorings* is presented. Section 4.5 introduces a new branching average coloring called *the Stripe Average*. Its use is exemplified in Appendix A.

All colorings named above are examples of *average colorings*. Before continuing to the discussion of these colorings, the average colorings must be defined. Also, two different interpolation methods for average colorings are presented.

### 4.1 Average Colorings

Given a truncated orbit  $O_T(z_0)$ , introduce the notation  $Z_i^m = \{z_{i-m}, z_{i-m+1}, \dots, z_i\}$ . The notation  $Z_i^m(z_0) = Z_i^m$  is used equivalently when the point  $z_0$  used to calculate the iterates is important. The constant  $m$  is the number of iterates in  $Z_i^m$  preceding  $z_i$ . *Average colorings* calculate an average of a function  $t : \mathbb{C}^{m+1} \mapsto \mathbb{R}$  evaluated at all points  $Z_i^m(z_0)$  contained in a truncated orbit  $O_T(z_0)$ . For a point  $z_0 \in L_i \cup B_i$ , denote this average sum  $S_i$ . Formally,

$$S_i(O_T(z_0)) = \frac{1}{i-m} \sum_{n=m+1}^i t_n \quad (4.1)$$

where  $t_n = t(Z_n^m(z_0))$ . The function  $t$  is called *the addend function* and it is assumed to be a continuous function of elements in  $Z_n^m$ .

Each element  $t(Z_i^m)$  of the average sum requires  $m$  iterations that precede  $z_i$ . When applying average colorings to a fractal, the iteration levels  $L_0, \dots, L_m$  are



excluded from the coloring calculations because the required number of preceding iterations are not available.

### 4.1.1 Using Smooth Iteration Count for Interpolation

Let  $i > m + 1$ . Consider a point  $z_0 \in B_i$ ,  $z_0 = r_0 e^{i\theta_0}$ , and another point  $\hat{z}_0 \in L_{i-1}$  such that  $\hat{z}_0 = (r_0 + \varepsilon) e^{i\theta_0}$ , where  $\varepsilon$  is small and nonnegative. The corresponding average sums are

$$S_i(O_T(z_0)) = \frac{1}{i - m} \sum_{n=m+1}^i t(Z_n^m(z_0))$$

for  $z_0$  and

$$S_{i-1}(O_T(\hat{z}_0)) = \frac{1}{i - m - 1} \sum_{n=m+1}^{i-1} t(Z_n^m(\hat{z}_0))$$

for  $\hat{z}_0$ . For all elements  $\hat{z}_n \in O_T(\hat{z}_0)$ ,  $\hat{z}_n \rightarrow z_n$  as  $\varepsilon \downarrow 0$ . It follows from the continuity of  $t$  that also  $t(Z_n^m(\hat{z}_n)) \rightarrow t(Z_n^m(z_0))$ . The sum  $S_{i-1}$  can thus be calculated for  $O_T(z_0)$  as  $\varepsilon \downarrow 0$ .

A discontinuity appears for the average sums between the points  $z_0$  and  $\hat{z}_0$  because, in general,  $S_i \neq S_{i-1}$ . This discontinuity can be eliminated by interpolating between  $S_{i-1}$  and  $S_i$  in  $L_i \cup B_i$ . The decimal part  $d$  of the smooth iteration count is 0 at  $B_{i-1}$  and converges to 1 in the neighborhood of  $B_i$  that intersects  $L_i$ . Thus  $d$  can be used as the interpolation coefficient.

Using linear interpolation, define for  $z \in L_i \cup B_i$  a coloring function

$$u_L(O_T(z)) = dS_i + (1 - d)S_{i-1}. \quad (4.2)$$

For  $z_0 \in B_i$  as above,  $d = 0$  and  $u_L = S_{i-1}$ . For  $\hat{z}_0 \in L_{i-1}$ ,  $d \rightarrow 1$  and  $u_L \rightarrow S_{i-1}$  as  $\varepsilon \downarrow 0$ . That is,  $u_L$  is continuous. However, linear interpolation normally results in discontinuity of the derivative at the control points  $S_i$ . This is illustrated in Figure 4.1.

Smoothness can be achieved by using some suitable type of *splines* instead of linear interpolation. A good choice are the Catmull-Rom splines [4, p. 324-325]. For  $z \in L_i \cup B_i$ , define

$$u_S(O_T(z)) = H_0(d)S_i + H_1(d)S_{i-1} + H_2(d)S_{i-2} + H_3(d)S_{i-3}. \quad (4.3)$$

The polynomials  $H_0 - H_3$  are listed in Table 4.1. The polynomials are chosen so that at  $d = 0$ ,

$$\frac{du_S}{dd} = \frac{1}{2}(S_{i-1} - S_{i-3})$$

and at  $d = 1$ ,

$$\frac{du_S}{dd} = \frac{1}{2}(S_i - S_{i-2}).$$

That is, the tangent of the graph of  $u$  at  $d = 0$  (or, equivalently, at  $d = 1$ ) is parallel to the line segment connecting the previous and the next control points.

Notation	Polynomial
$H_0$	$\frac{1}{2}(-d^2 + d^3)$
$H_1$	$\frac{1}{2}(d + 4d^2 - 3d^3)$
$H_2$	$\frac{1}{2}(2 - 5d^2 + 3d^3)$
$H_3$	$\frac{1}{2}(-d + 2d^2 - d^3)$

Table 4.1: The Catmull-Rom spline weighting polynomials.

The coloring 4.3 interpolates between the points  $S_{i-2}$  at  $d = 0$  and  $S_{i-1}$  at  $d = 1$ , whereas the points  $S_{i-3}$  and  $S_i$  are needed for specifying the derivatives at  $d = 0$  and  $d = 1$ , respectively. The coloring  $u_S$  is smooth also in the neighborhood of  $B_i$ . It is here called *smooth* interpolation due to its smoothness. Linear and smooth interpolation are illustrated in Figure 4.1.

The above discussion of average colorings is summarized by the following definition.

**Definition 4.1** (Average Coloring). *Given an addend function  $t : \mathbb{C}^m \mapsto \mathbb{R}$  and average sums specified by equation 4.1, an average coloring is given either by 4.2 or 4.3.*

For some fractals, the lack of smoothness for linear interpolation may create artifacts similar to those in Figure 3.3. For fractals generated with various values of  $c$  in system 2.3, these artifacts are subtle and almost indistinguishable (cf. Figure 4.3).

For some other fractals, however, the artifacts are clearly visible. Figure 4.2 compares the linear and smooth interpolation methods applied to a detail of a fractal set that clearly shows the artifacts. The Triangle Inequality Average coloring was used. The fractal is calculated in analogy with the example fractal, but system 2.3 is replaced by the system

$$z_{n+1} = \frac{z_n^2}{1 + dz_n} + c$$

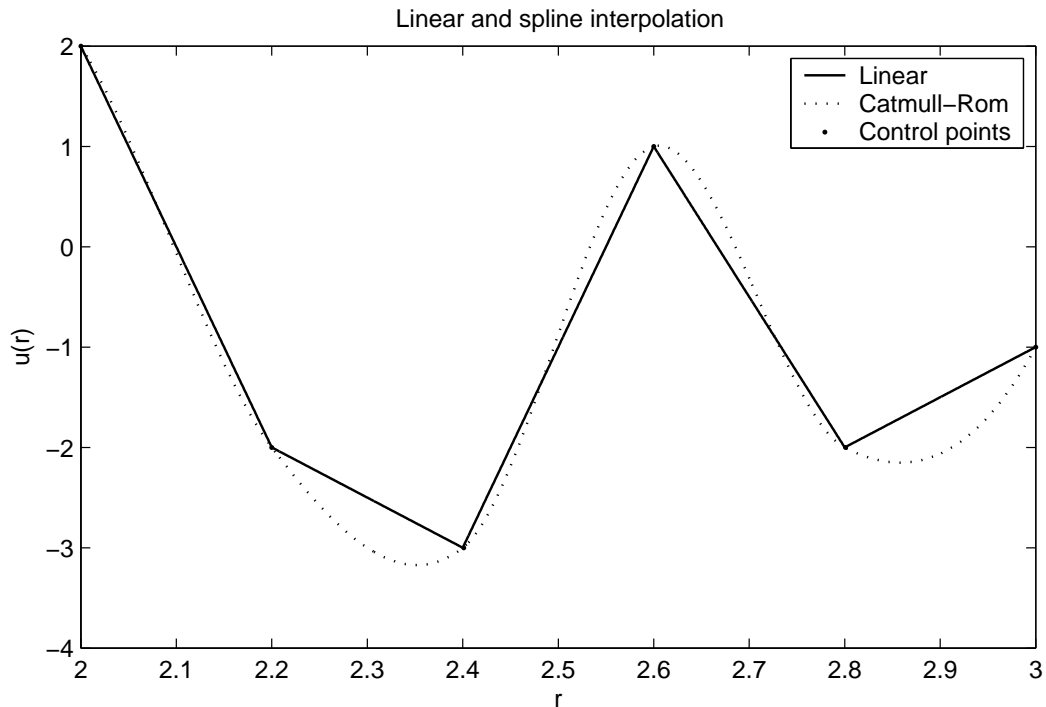


Figure 4.1: Comparison of linear and Catmull-Rom spline interpolation methods on an arbitrarily chosen set of control points.

where  $d = -0.2013 + 0.5638i$  and  $c = 1.4686 + 1.265i$ . Using smooth interpolation removes the artifacts. On the other hand, the structure of the coloring becomes blurrier because the average sums are interpolated over four iteration levels instead of two.

## 4.2 The Triangle Inequality Average Coloring

The Triangle Inequality Average coloring was originally developed by Kerry Mitchell. Figure 4.3 shows the Triangle Inequality Average coloring applied to the example fractal.

The original implementation used linear interpolation. It was designed for coloring the Mandelbrot set whereas a variation adapted for Julia sets is presented here. In the Julia set variant, the constant  $c$  in the discussion below denotes the seed of the Julia set ( $0.5 + 0.25i$  for the example fractal). In the Mandelbrot variant,  $c$  corresponds to the position of the pixel to be colored. However, both variants are based on the same idea and the two variations merely reflect the

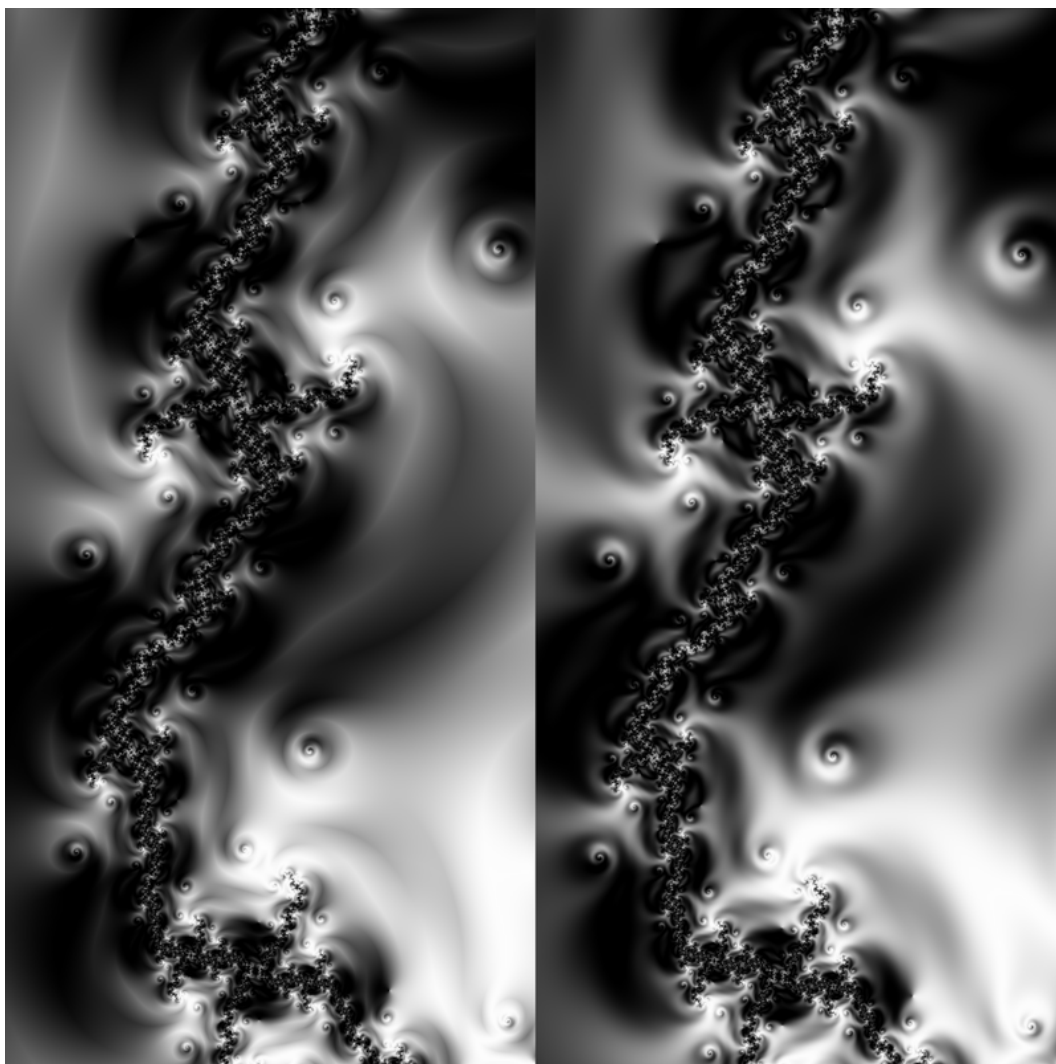


Figure 4.2: Comparison of linear (left) and smooth (right) interpolation.

differences in the calculation algorithm for Julia and Mandelbrot fractal images. Their visual appearance is very similar.

Consider the system 2.3. The  $n$ th iteration can be written

$$z_n = z_{n-1}^p + c.$$

Applying the triangle inequality to the right side of the equation gives

$$\left| |z_{n-1}^p| - |c| \right| \leq |z_{n-1}^p + c| \leq |z_{n-1}^p| + |c|,$$

or

$$\left| |z_{n-1}^p| - |c| \right| \leq |z_n| \leq |z_{n-1}^p| + |c|.$$

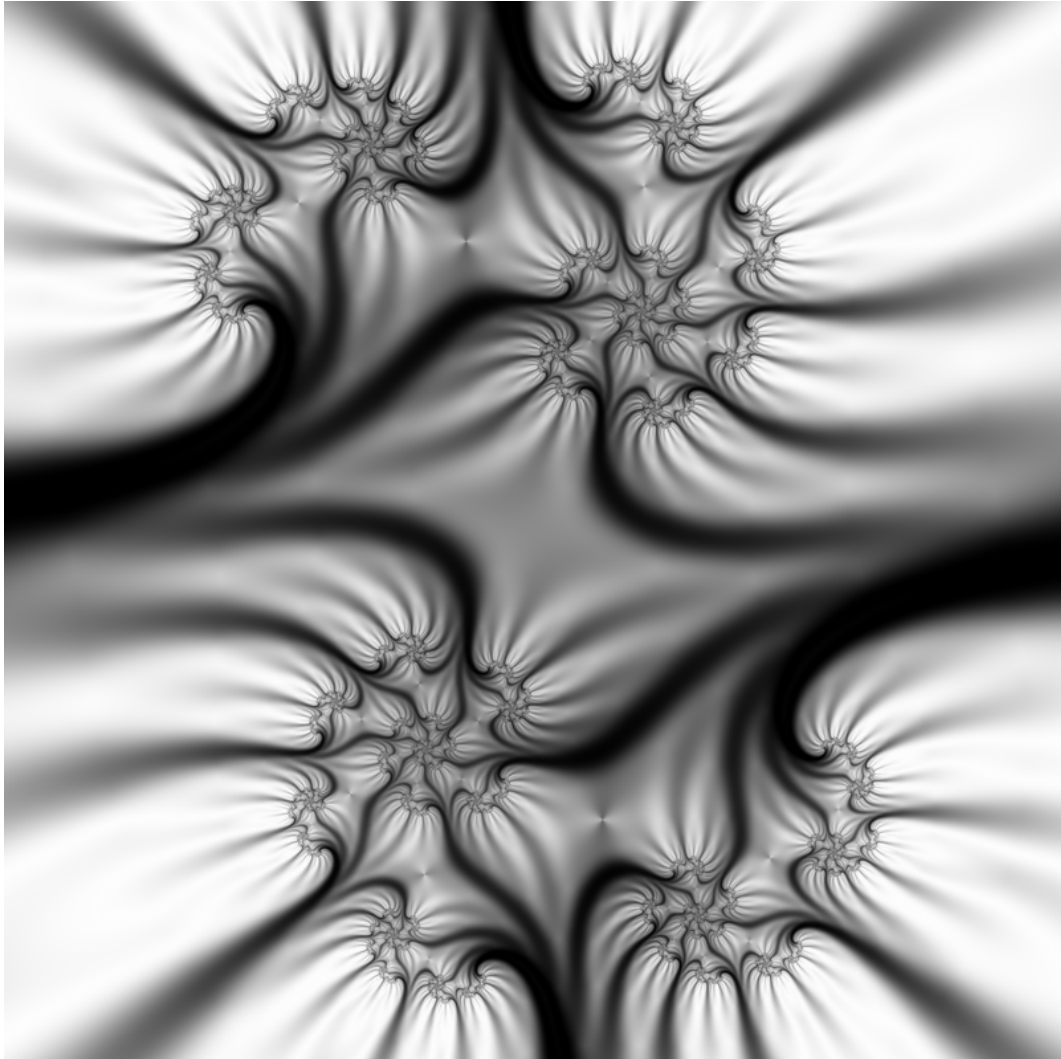


Figure 4.3: The example fractal colored with the Triangle Inequality Average coloring using linear interpolation.

Introduce the notation

$$m_n = ||z_{n-1}^p| - |c||$$

and

$$M_n = |z_{n-1}^p| + |c|$$

for the upper and lower bounds of  $|z_n|$ . Note that  $m_n = m_n(z_{n-1})$  and  $M_n = M_n(z_{n-1})$ . These bounds can be used to define a quotient in the interval  $[0, 1]$ .

**Definition 4.2** (Triangle Inequality Average Coloring). *Define the addend function  $t : \mathbb{C}^2 \mapsto \mathbb{R}$  as*

$$t(Z_n^1) = \frac{|z_n| - m_n}{M_n - m_n}. \quad (4.4)$$

Using this addend function, the Triangle Inequality Average coloring is given by Definition 4.1.

Consider the two cases when the term  $|z_n|$  reaches its upper and lower bounds.

1. If  $\arg z_{n-1}^p = -\arg c$ ,  $z_{n-1}^p$  and  $c$  point at opposite directions in the complex plane and the lower bound is reached. Formally,  $|z_{n-1}^p + c| = ||z_{n-1}^p| - |c||$ .
2. If  $\arg z_{n-1}^p = \arg c$ ,  $z_{n-1}^p$  and  $c$  are parallel and  $|z_{n-1}^p + c| = |z_{n-1}^p| + |c|$ .

Thus  $t(Z_n^1)$  varies as a function of  $\arg z_{n-1}^p$  between 0 when  $\arg z_{n-1}^p = -\arg c$  and 1 when  $\arg z_{n-1}^p = \arg c$ .

### 4.2.1 Behavior for Distant Starting Points

The following discussion describes the properties of an addend  $t_n$  under the assumption that the magnitude of the starting point  $z_0$  is large so that the simplified system 3.2 is accurate for the  $n-1$  first iterations. The addend function  $t$  of the Triangle Inequality Average coloring is based on the effect of  $c$  on the argument of iterates. Thus the system 2.3 is used for the  $n$ th iteration whereas the system 3.2 is used for the  $n-1$  first iterations.

Denote  $c = r_c e^{i\theta_c}$ . Assuming  $r_c \ll |z_0|$ , the difference in the bounds  $M_n$  and  $m_n$  can be simplified to

$$\begin{aligned} M_n - m_n &= |z_{n-1}^p| + r_c - ||z_{n-1}^p| - r_c| \\ &= 2r_c \end{aligned}$$

because  $|z_{n-1}^p| > r_c$ . The  $n$ th addend  $t_n$  can be expressed as

$$\begin{aligned} t_n &= \frac{r_n - m_n}{M_n - m_n} \\ &= \frac{r_n^2 - m_n^2}{2r_c(r_n + m_n)}. \end{aligned}$$

The square of  $r_n$  can be written

$$\begin{aligned} r_n^2 &= |z_n|^2 \\ &= (z_{n-1}^p + c)\overline{(z_{n-1}^p + c)} \\ &= (r_{n-1}^p e^{i\theta_{n-1}p} + r_c e^{i\theta_c})(r_{n-1}^p e^{-i\theta_{n-1}p} + r_c e^{-i\theta_c}) \\ &= r_{n-1}^{2p} + r_c^2 + r_{n-1}^p r_c (e^{i(\theta_{n-1}p - \theta_c)} + e^{-i(\theta_{n-1}p - \theta_c)}) \\ &= r_{n-1}^{2p} + r_c^2 + 2r_{n-1}^p r_c \cos(\theta_{n-1}p - \theta_c). \end{aligned} \tag{4.5}$$

In analogy with 3.3, the  $(n - 1)$ th iterate is  $z_{n-1} = z_0^{p^{n-1}}$ . It follows that  $r_{n-1} = r_0^{p^{n-1}}$  and  $\theta_{n-1} = p^{n-1}\theta_0$ . Substituting to equation 4.5 gives

$$r_n^2 = r_0^{2p^n} + r_c^2 + 2r_0^{p^n} r_c \cos(p^n \theta_0 - \theta_c).$$

Using the identities above, the  $n$ th addend  $t_n$  becomes

$$\begin{aligned} t_n &= \frac{r_n^2 - m_n^2}{2r_c(r_n + m_n)} \\ &= \frac{r_0^{2p^n} + r_c^2 + 2r_0^{p^n} r_c \cos(p^n \theta_0 - \theta_c) - (r_0^{2p^n} - 2r_0^{p^n} r_c + r_c^2)}{2r_c(r_n + m_n)} \\ &= \frac{2r_0^{p^n} r_c [\cos(p^n \theta_0 - \theta_c) + 1]}{2r_c(r_n + r_0^{p^n} - r_c)} \\ &= \frac{\cos(p^n \theta_0 - \theta_c) + 1}{|z_0^{p^n} + c|/r_0^{p^n} + 1 - r_c/r_0^{p^n}} \end{aligned} \quad (4.6)$$

$$\approx \frac{1}{2} \cos(p^n \theta_0 - \theta_c) + \frac{1}{2}. \quad (4.7)$$

In the last step the approximations  $|z_0^{p^n} + c|/r_0^{p^n} \approx 1$  and  $r_c/r_0^{p^n} \approx 0$  have been used.

Expressions 4.6 and 4.7 encapsulate the characteristic properties of the Triangle Inequality Average coloring. Firstly, 4.6 shows that the addend  $t_n$  is only weakly dependent of the magnitude  $r_0$  of the initial point. Secondly,  $t_n$  varies sinusoidally as a function of the initial angle  $\theta_0$ . Each iteration increases its frequency by a factor  $p$ . Section 4.4.1 explains how these properties lead to the branching appearance that is characteristic to the Triangle Inequality Average coloring.

Figure 4.4 shows  $t(Z_1^1) \dots t(Z_4^1)$  and their mean value as a function of  $r_1$  and  $\theta_1$ . Note that  $r_1$  and  $\theta_1$  correspond to  $r_0$  and  $\theta_0$  in the above equations because evaluating  $t$  requires two iterates. The iterates  $z_1 \dots z_4$  are calculated for the example fractal. The curves are in accordance with the above observations based on expressions 4.6 and 4.7.

## 4.2.2 Behavior in the Neighborhood of Origin

**Theorem 4.1.** *The Triangle Inequality Average coloring is discontinuous at points that are mapped to 0.*

*Proof.* Assume that  $z_n = \varepsilon e^{i\theta}$ , where  $\varepsilon$  is small, is an arbitrary iterate that has been mapped to the neighborhood of the origin. The corresponding upper and

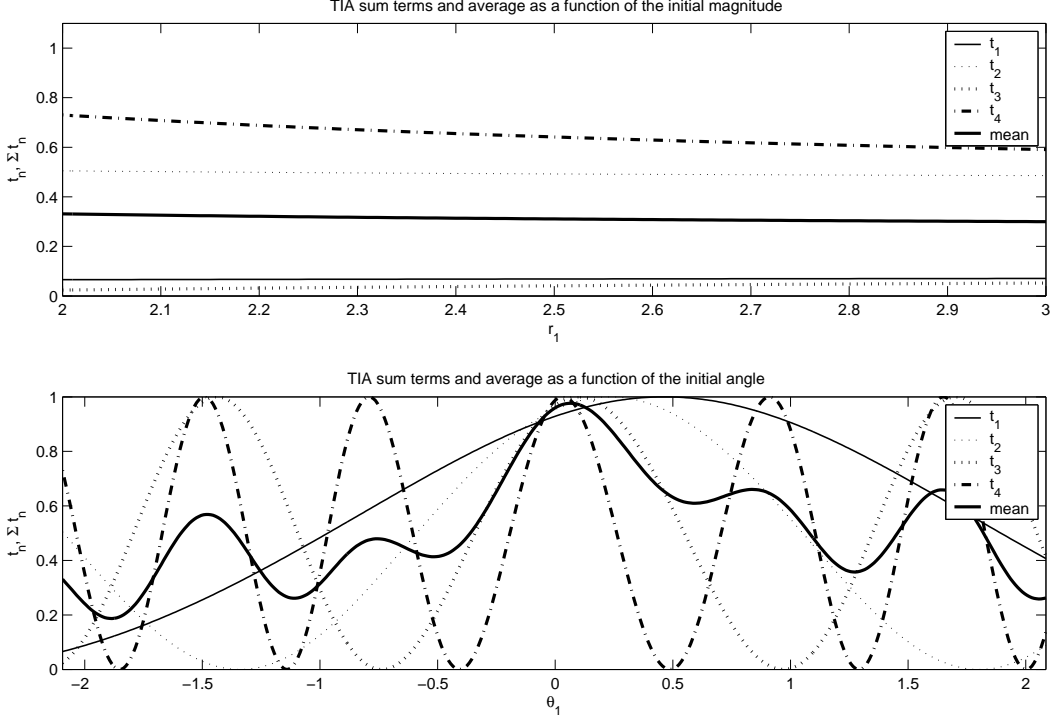


Figure 4.4: The dependence of the addends  $t_1 \dots t_4$  on  $r_1$  and  $\theta_2$  in the Triangle Inequality Average coloring. System 2.3 was used in the calculations.

lower bounds are

$$M_{n+1} = \varepsilon^p + r_c$$

and

$$m_{n+1} = |\varepsilon^p - r_c| = r_c - \varepsilon^p.$$

The denominator in expression 4.4 is

$$M_{n+1} - m_{n+1} = 2\varepsilon^p.$$

Using equation 4.5, the addend  $t_{n+1}$  can be written

$$\begin{aligned} t_{n+1} &= \frac{r_{n+1} - m_{n+1}}{M_{n+1} - m_{n+1}} \\ &= \frac{r_{n+1}^2 - m_{n+1}^2}{2\varepsilon^p(r_{n+1} + m_{n+1})} \\ &= \frac{\varepsilon^{2p} + r_c^2 + 2\varepsilon^p r_c \cos(p\theta - \theta_c) - (\varepsilon^{2p} - 2\varepsilon^p r_c + r_c^2)}{2\varepsilon^p(r_{n+1} + m_{n+1})} \\ &= \frac{2\varepsilon^p r_c [\cos(p\theta - \theta_c) + 1]}{2\varepsilon^p(r_{n+1} + m_{n+1})} \\ &= \frac{r_c [\cos(p\theta - \theta_c) + 1]}{(r_{n+1} + r_c - \varepsilon^p)} \end{aligned}$$



The magnitude of the  $(n + 1)$ th iterate is

$$r_{n+1} = |\varepsilon e^{i\theta} + c| \rightarrow r_c$$

as  $\varepsilon \rightarrow 0$ . Thus

$$t_{n+1} \rightarrow \frac{r_c[\cos(p\theta - \theta_c) + 1]}{2r_c} = \frac{1}{2} \cos(p\theta - \theta_c) + \frac{1}{2}.$$

This limit is dependent on the angle  $\theta$  so  $t_{n+1}$  is discontinuous at  $z_n = 0$ .  $\square$

Figure 4.5 shows the discontinuities of the Triangle Inequality Average marked with circles.

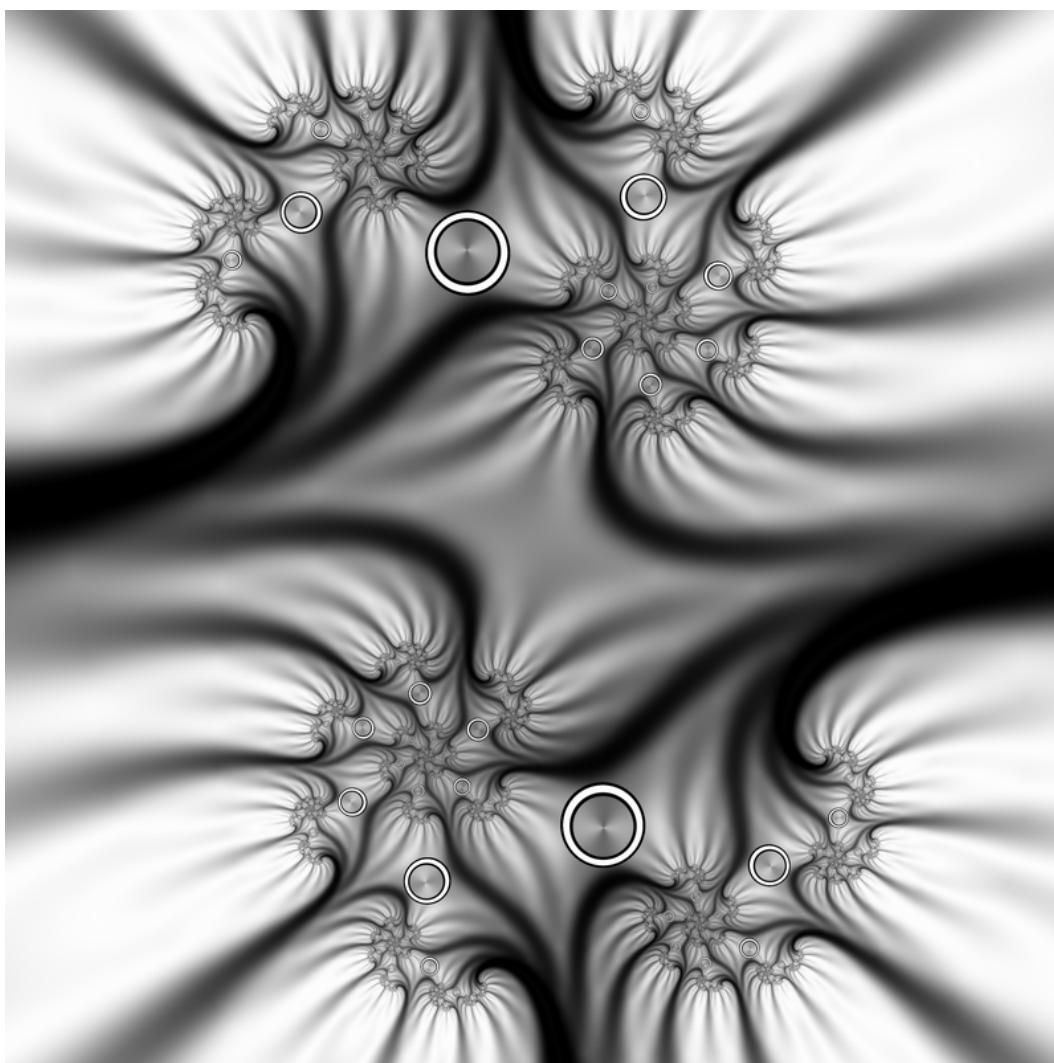


Figure 4.5: The Triangle Inequality Average coloring applied to the example fractal. The discontinuous points that are mapped to 0 are marked with circles.

### 4.3 The Curvature Average Coloring

The Curvature Average coloring was originally developed by Damien M. Jones for Ultra Fractal in 1999. It is illustrated in Figure 4.6. As the figure and the following analysis reveal, it is both visually and mathematically very similar to the Triangle Inequality Average coloring.

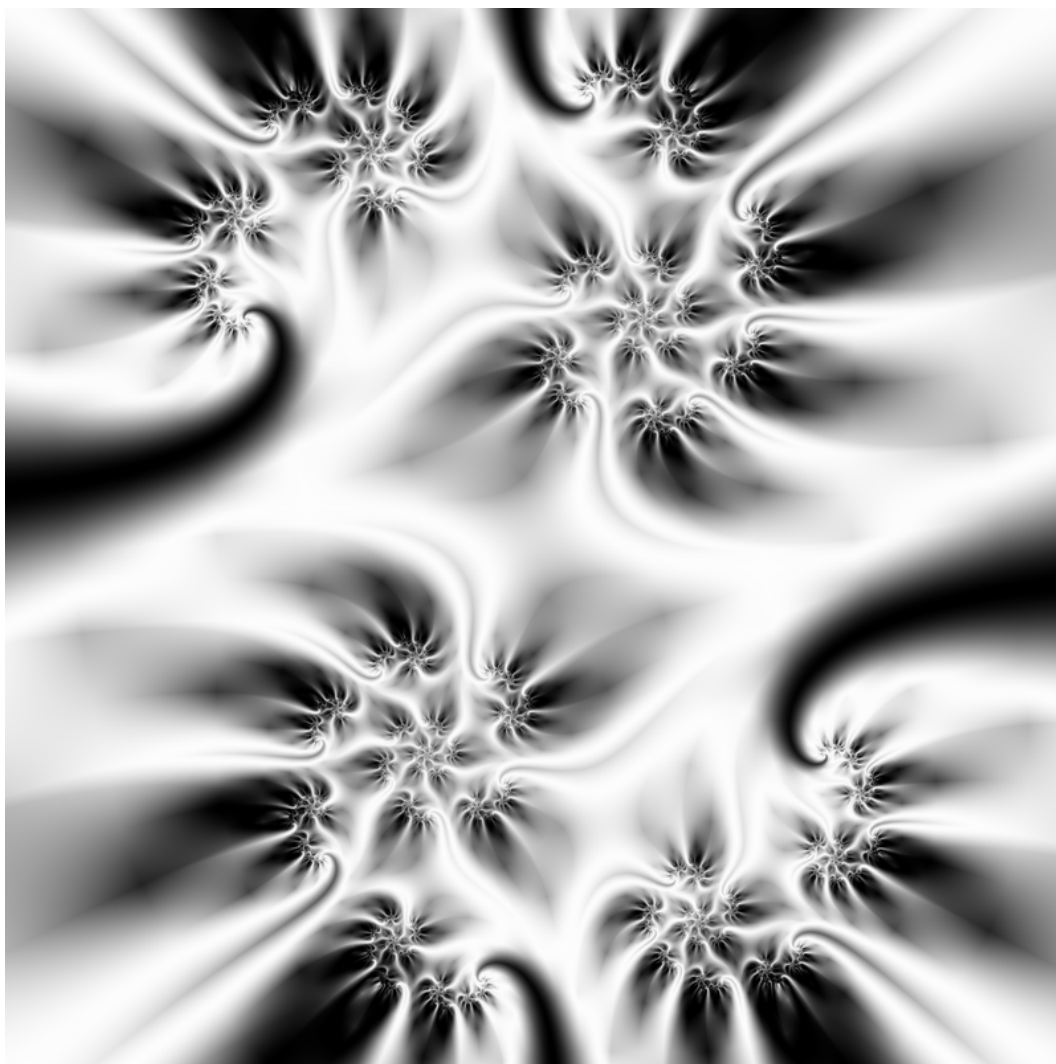


Figure 4.6: The example fractal colored with the Curvature Average coloring using linear interpolation.

The coloring is based on the idea of approximating the curvature of a curve defined by discrete points. The curve is in this case defined by the points of a

truncated orbit  $O_T(z_0)$ . The approximation used is

$$t(Z_n^2) = \left| \arg_{(-\pi, \pi]} \frac{z_n - z_{n-1}}{z_{n-1} - z_{n-2}} \right|.$$

where  $\arg_{(-\pi, \pi]} z$  denotes the argument of  $z$  in the interval  $(-\pi, \pi]$ . This also defines the averaging function in the sum 4.1.

**Definition 4.3** (Curvature Average Coloring). *Define the addend function  $t : \mathbb{C}^3 \mapsto \mathbb{R}$  as*

$$t(Z_n^2) = \left| \arg_{(-\pi, \pi]} \frac{z_n - z_{n-1}}{z_{n-1} - z_{n-2}} \right|. \quad (4.8)$$

*Using this addend function, the Curvature Average coloring is given by Definition 4.1.*

Unlike the Triangle Inequality Average coloring, the Curvature Average is continuous at points that are mapped to 0.

### 4.3.1 Behavior for Distant Starting Points

Suppose  $r_0$  is large and the simplified system 3.2 can be used instead of 2.3. Substituting  $z_n = z_0^{p^n}$  to 4.8 gives

$$\begin{aligned} t(Z_n^2) &= \left| \arg_{(-\pi, \pi]} \frac{z_n - z_{n-1}}{z_{n-1} - z_{n-2}} \right| \\ &= \left| \arg_{(-\pi, \pi]} \frac{z_0^{p^n} - z_0^{p^{n-1}}}{z_0^{p^{n-1}} - z_0^{p^{n-2}}} \right| \\ &= \left| \arg_{(-\pi, \pi]} \frac{z_0^{p^{n-1}} (z_0^{p^{n-1}(p-1)} - 1)}{z_0^{p^{n-2}} (z_0^{p^{n-2}(p-1)} - 1)} \right| \\ &= \left| \arg_{(-\pi, \pi]} \frac{z_0^{p^{n-2}(p-1)} (z_0^{p^{n-1}(p-1)} - 1)}{z_0^{p^{n-2}(p-1)} - 1} \right| \end{aligned}$$

Assuming  $z_0^{p^{n-1}(p-1)} > z_0^{p^{n-2}(p-1)} \gg 1$  gives

$$\begin{aligned} t(Z_n^2) &\approx \left| \arg_{(-\pi, \pi]} z_0^{p^{n-1}(p-1)} \right| \\ &= \left| -\pi + \theta_0 p^{n-1}(p-1) - 2\pi \left\lfloor \frac{\theta_0 p^{n-1}(p-1)}{2\pi} \right\rfloor \right|. \end{aligned} \quad (4.9)$$

where  $\lfloor a \rfloor$  denotes the largest integer less than or equal to  $a$ .

The visual appearance of the Curvature Average coloring is in accordance with equation 4.9. Figure 4.7 shows  $t(Z_2^2) \dots t(Z_5^2)$  and their mean value as a

function of  $r_0$  and  $\theta_0$ . The iterates are generated for the example fractal. The addends are clearly almost independent of  $r_0$ . The approximative expression 4.9 to  $t(Z_n^2)$  defines a continuous and periodic function that corresponds to the absolute value of a triangle wave function (cf. Figure 4.7). Its frequency becomes  $p$  times larger at each iteration. Thus the Curvature Average coloring exhibits the similar properties as the Triangle Inequality Average that lead to branching.

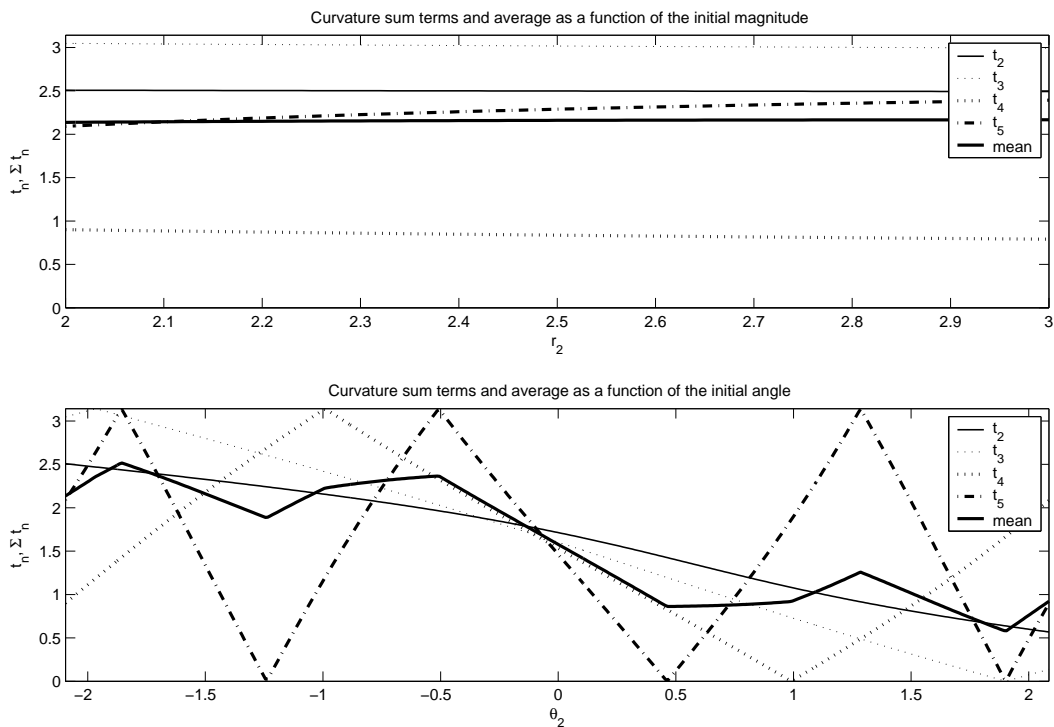


Figure 4.7: The addends  $t(Z_2^2) \dots t(Z_5^2)$  as functions of  $r_2$  and  $\theta_2$  in the Curvature Average coloring. System 2.3 was used in the calculations.

## 4.4 Similarities and Differences

Although the Triangle Inequality Average and Curvature Average colorings are based on different approaches, they appear to have visually similar structure. This is due to the fact that the addends exhibit similar behavior for distant iterates, as can be noted by comparing equations 4.7 and 4.9.

As an iterate of an orbit escapes outside the area where the system 2.3 behaves chaotically, the following iterates diverge at an increasing rate. The conclusions

made above for distant starting points are valid for these iterates. The corresponding addends appear to give the average sum its characteristic appearance.

This claim is supported by Figure 4.8. It shows the Triangle Inequality Average and Curvature Average colorings with the addends corresponding to the last 5 iterations excluded from the average sum. In contrast to the distant iterates, the remaining addends do not exhibit regular behavior described by the simplified system 3.2. Thus the characteristic branching structure is not present.

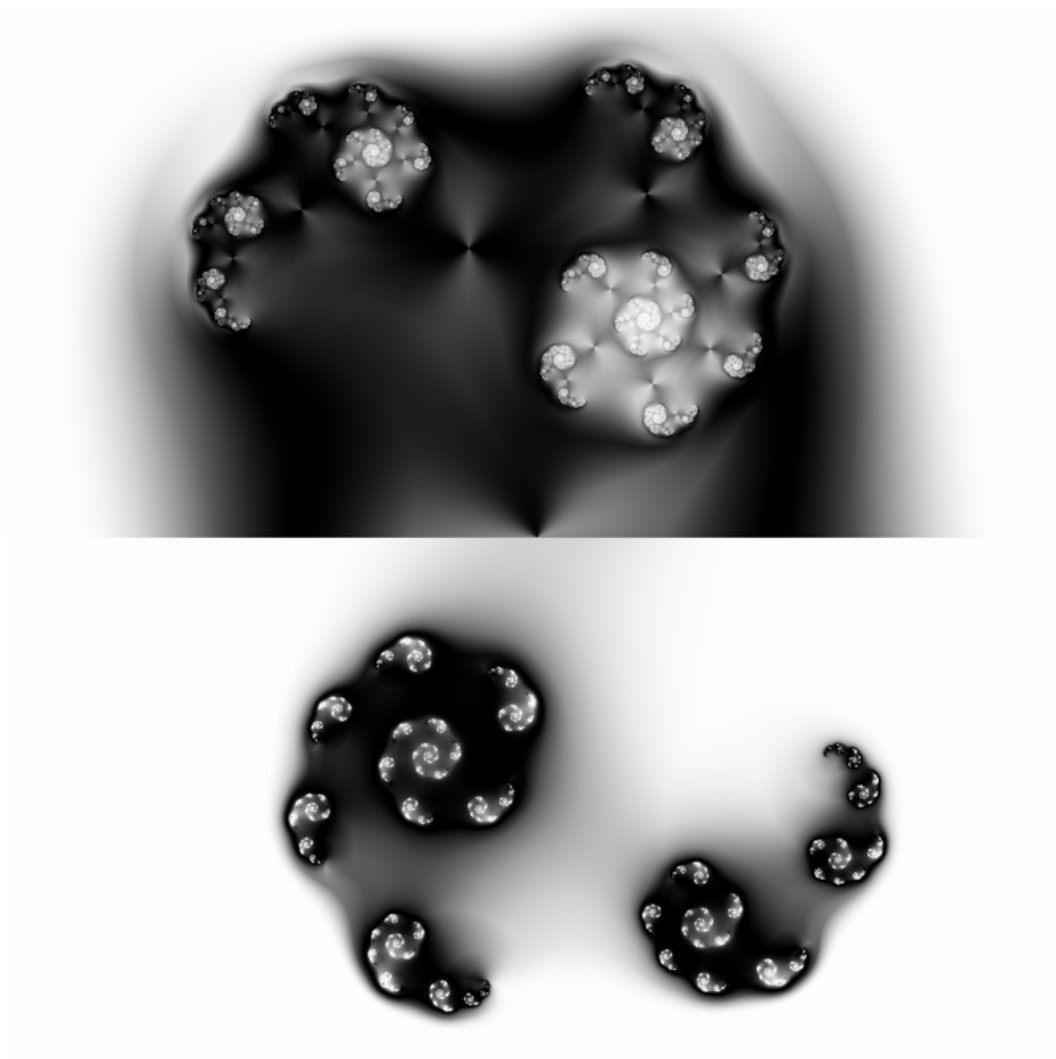


Figure 4.8: The Triangle Inequality Average (top) and Curvature Average colorings (bottom) with the last 5 addends excluded from the average sums.

The addends for these colorings are periodic functions and each iteration increases the frequency by a factor  $p$ . The periodic function is the cosine function

for the Triangle Inequality Average and the absolute value of a triangle wave function for Curvature Average. The difference in smoothness with respect to the initial angle  $\theta_0$  emerge from these functions – the cosine function is continuous and smooth whereas the absolute value of a triangle function is continuous but not smooth at its maximum and minimum points. The points where the Curvature Average is not smooth are clearly visible as edges in Figure 4.6.

#### 4.4.1 Emergence of Branches

The following definition summarizes the properties that lead to the branching appearance characteristic to the Triangle Inequality Average and Curvature Average colorings (cf. Figure 4.3 and Figure 4.6).

**Definition 4.4** (Branching Coloring). *An average coloring function is said to be branching if its addends  $t_n = t(Z_n^m)$  fulfill the following properties when applied to the simplified system 2.3.*

1. *For  $k = n, n-1, \dots, n-m$ , the derivatives  $\frac{dt_n}{dr_k}$  of an addend  $t_n$  with respect to the magnitudes  $r_k$  are small.*
2. *Addends are periodic functions of the argument of iterates.*
3. *Each iteration increases the period by an integer factor  $p$ .*

Property 1 implies that the addends vary more noticeably with respect to the argument of iterates than their magnitude, and lines perpendicular to the iteration boundaries appear at the iteration levels. Property 2 guarantees that the lines are evenly distributed around the iteration levels. Property 3 causes the number of lines to increase by a factor of  $p$  from  $L_{i-1}$  to  $L_i$ . Interpolation of the average sums between two iteration boundaries create an impression that the lines divide into  $p$  branches.

For the example fractal,  $p = 2$  so each iteration divides the branches into two. This period doubling is a consequence of Property 3 above. Figure 4.9 illustrates that the number of lines is doubled between each iteration boundary when moving towards the fractal set. The figure shows the average sums of the Triangle Inequality Average coloring applied to the example fractal without interpolation.

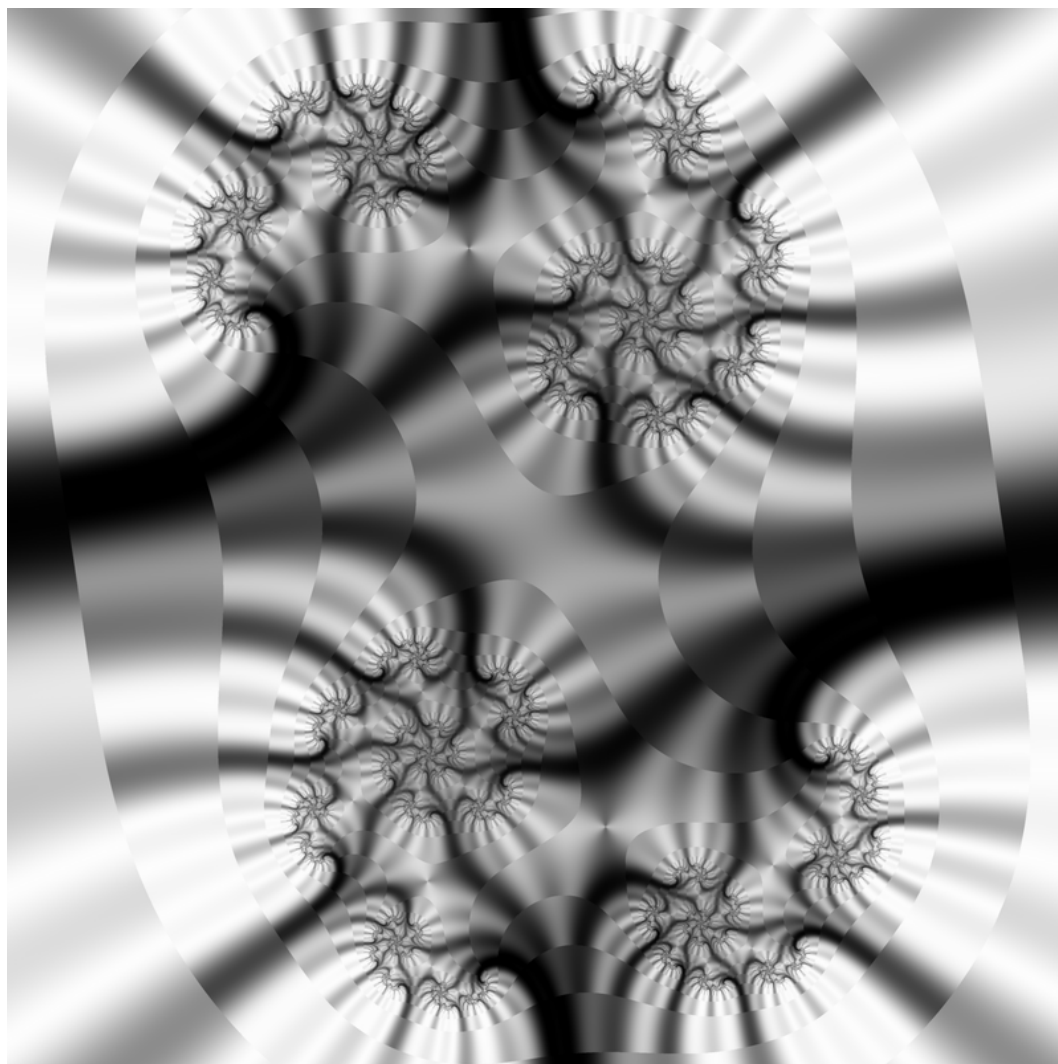


Figure 4.9: The average sums of the Triangle Inequality Average coloring applied to the example fractal without interpolation.

#### 4.4.2 Continuity and Smoothness

Except for points that are mapped to 0, the Triangle Inequality Average coloring is continuous and smooth in the iteration levels. Ignoring the effect of the small discontinuity in the decimal part of the smooth iteration count, it is also continuous but not smooth at the iteration boundaries when using linear interpolation. Using smooth interpolation guarantees smoothness also at the iteration boundaries.

The same conclusions also hold for the Curvature Average coloring except for the lines that emerge at the extreme points of the triangle waves. On these lines

the coloring is continuous but not smooth.

## 4.5 The Stripe Average Coloring

A branching coloring similar to the Triangle Inequality Average and Curvature Average can be achieved with a more explicit approach.

**Definition 4.5** (Stripe Average Coloring). *Define the addend function  $t : \mathbb{C} \mapsto \mathbb{R}$  as*

$$t(z_n) = \frac{1}{2} \sin(s \arg z_n) + \frac{1}{2}. \quad (4.10)$$

*The constant  $s$  is called the stripe density. Using this addend function, the Stripe Average coloring is given by Definition 4.1.*

Using the approximative identity  $\theta_n = p^n \theta_0$ , the addend function becomes

$$t(z_n) = \frac{1}{2} \sin(sp^n \theta_0) + \frac{1}{2}.$$

For  $s = 1$ , the addend function is equivalent to 4.7 after a translation by  $\theta_c$  with respect to  $\theta_0$ . Consequently, the Stripe Average coloring fulfills properties 1-3 of Definition 4.4. Its behavior is equivalent to that of the Triangle Inequality Average for distant iterates.

### 4.5.1 The Stripe Density Parameter

As its name indicates, the stripe density parameter  $s$  can be used to adjust the density of stripes. Large values increase the variation frequency with respect to  $\theta_0$  and thus result in denser stripes. Non-integer values result in discontinuity. Figure 4.10 shows the Stripe Average coloring using  $s = 3$  (top) and  $s = 10$  (bottom) applied to the example fractal. Comparison with Figure 4.3 shows that the branching structure is similar but stripes are denser in Figure 4.10. Bailout  $M = 10^{20}$  was used in both images.

It is noteworthy that increasing the bailout value has a similar effect of creating denser stripes. An increase to a power of  $p$  in the bailout adds one iteration and a corresponding addend to every average sum. This addend has a frequency that is  $p$  times larger than the frequency of the preceding addend. However, numerical accuracy sets a limit to how large bailout values can be used. Thus a higher stripe density can be achieved even for small bailouts by choosing the value of  $s$  appropriately.



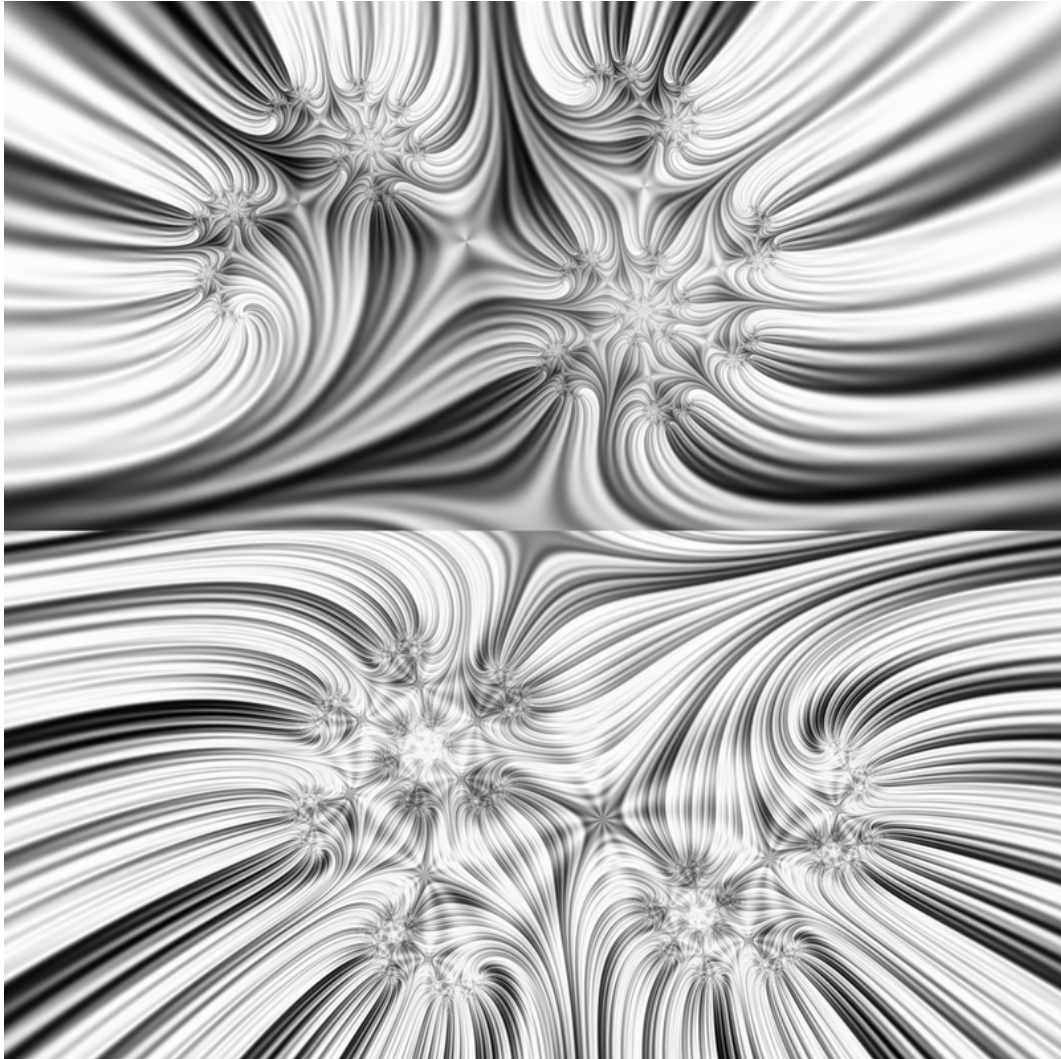


Figure 4.10: The Stripe Average coloring, using the stripe factors  $s = 3$  (top) and  $s = 10$  (bottom), applied to the example fractal.

### 4.5.2 Behavior in the Neighborhood of Origin

For  $z_n = \varepsilon e^{i\theta}$ , the expression 4.10 becomes

$$t(z_n) = \frac{1}{2} \sin(s\theta) + \frac{1}{2}$$

This is independent of  $\varepsilon$  but dependent on  $\theta$ . Thus the Stripe Average coloring is discontinuous at points that are mapped to 0. The discontinuities are clearly identifiable in Figure 4.10.

Figure 4.11 shows the Stripe Average applied to the example fractal with the 6 last addends excluded from the average sum. In analogy with the Triangle

Inequality Average and Curvature Average colorings, the branching structure disappears as the addends corresponding to the distant iterates are excluded.

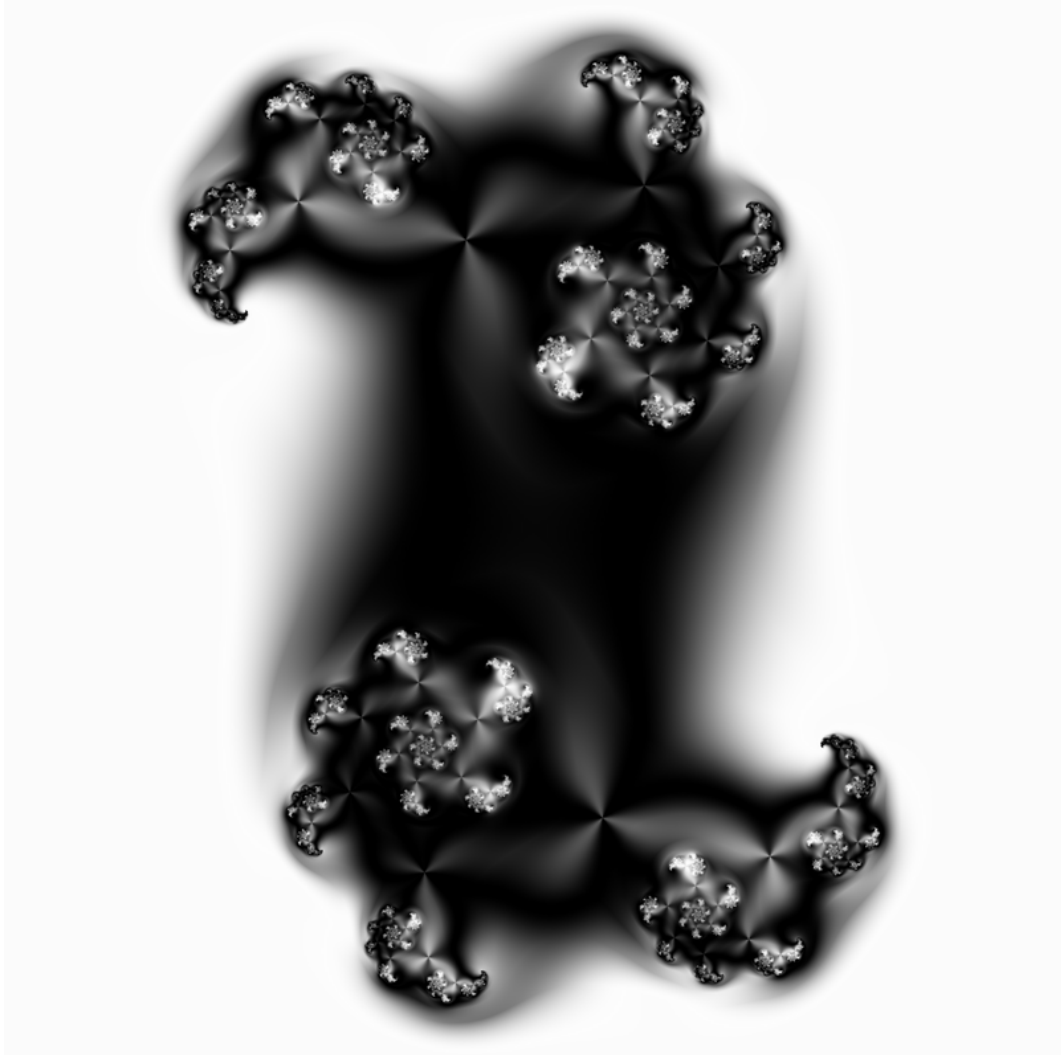


Figure 4.11: The Stripe Average coloring, using  $s = 3$ , applied to the example fractal. Six last addends were excluded from the average sums.

# Kapitel 5

## Svensk sammanfattning

Målsättningen med detta arbete var att studera matematiska egenskaper hos några algoritmer som används för att färglägga fraktalbilder.

Betrakta funktionen  $f(z) = z^p + c$  som definierar ett *dynamiskt system*  $z_n = z_{n-1}^p + c$ . Beroende på värdena av  $z_0$  och  $c$  kan iterationerna  $z_1 = f(z_0), z_2 = f(z_1), \dots$  sträva till oändligheten eller oskillera periodiskt mellan ett ändligt antal punkter.

Den klassiska algoritmen för beräkning av divergenta fraktaler kan utvidgas på följande sätt genom att införa index-, palett-, och färgläggningsfunktioner. För en punkt  $z_0 \in \mathbb{C}$ , beräkna iterationerna  $z_1 = f(z_0), z_2 = f(z_1), \dots, z_N = f(z_{N-1})$  tills  $|z_N| > M$ . Konstanten  $M$  definierar mängden  $|z_n| \leq M$  i  $\mathbb{C}$  där fraktalen beräknas. Om villkoret  $|z_n| > M$  inte uppfylls för någon iteration  $z_1, z_2, \dots, z_{N_{\max}}$  så betraktas punkten höra till fraktalmängden. Konstanten  $N_{\max}$  definierar maximala antalet iterationer. Detta arbete koncentrerar sig på områden där  $|z_n| > M$  uppfylls.

Färgläggningsfunktionen avbildar mängden av iterationerna  $\{z_0, z_1, \dots, z_N\}$  på ett reellt tal. Detta tal avbildas sedan av indexfunktionen och palettfunktionen till den slutliga RGB-färgen. Dessa operationer utförs för varje pixel. *Iterationsgränser* förekommer mellan punkter som kräver olika antal iterationer för att avbildas utanför området  $|z_n| \leq M$ .

Förutom att kräva så lite beräkningsoperationer som möjligt är det ofta önskvärt att färgläggningen är glatt, dvs. att den är kontinuerligt deriverbar i  $\mathbb{C}$ . Sådana parametrar som ingår i färgläggningsformeln och kan användas för att justera färgläggningens utseende är också nyttiga.

Färgläggningsfunktionen för den klassiska iterationsräknaren kan definieras som

$$u = kN$$

där  $N$  betecknar antalet iterationer som behövs för att  $|z_N| > M$  och  $k$  är en positiv konstant. Färgläggningens värde ändras diskontinuerligt vid iterationsgränserna och hålls konstant mellan dem.

Den kontinuerliga iterationsräknaren är en kontinuerlig men inte glatt färgläggning och den ger klart synliga gränserna där iterationsräknarens värde ändras. Den glatta iterationsräknaren är glatt överallt. Dessa färgläggningar antar värden lika med iterationsräknaren på iterationsgränserna men varierar kontinuerligt mellan dem. Iterationsgränserna kan inte urskiljas från en fraktal som färglagts med den glatta iterationsräknaren. En jämförelse med den kontinuerliga iterationsräknaren visar varför glatthet är en viktig egenskap hos färgläggningar.

Den glatta iterationsräknaren är dock glatt endast för systemet  $f(z) = z^p$  medan den visar diskontinuiteter på iterationsgränserna för det kaotiska systemet  $f(z) = z^p + c$ ,  $c \neq 0$ . En övre gräns för diskontinuitetens storlek kan beräknas med formeln

$$\Delta_{\max} = \frac{1}{\ln p} \ln \frac{\ln M^p}{\ln(M^p - |c|)}. \quad (5.1)$$

I praktiken är diskontinuiteten obetydligt liten för  $M = 10$  och blir ännu mindre för större värden av  $M$ .

Medeltalfärgläggningar är en familj av färgläggningar som utnyttjar medelvärdessummorna

$$S_i(z_0, z_1, \dots, z_i) = \frac{1}{i - m} \sum_{n=m+1}^i t_n. \quad (5.2)$$

Addenderna  $t_n$  beräknas med en så kallad addendfunktion  $t : \mathbb{C}^{m+1} \mapsto \mathbb{R}$  som är en funktion av  $z_n$  och eventuellt ett antal föregående iterationer  $z_{n-1}, z_{n-2}, \dots, z_{n-m}$ . Addendfunktionens egenskaper bestämmer färgläggningens utseende.

Medeltalfärgläggningarna beräknar summorna  $S_{i-1}$  och  $S_i$  och interpolerar mellan dessa. Decimaldelen av den glatta iterationsräknaren används som interpolationsvariabel. Linjär interpolation ger vanligen goda och nästan glatta resultat, annars kan man använda Catmull-Roms kurvor för att få en glatt färgläggning.

Triangelolikhet- och kurvatur-medelvärdesfärgläggningarna är exempel på medelvärdesfärgläggningar. Addendfunktionen för den sistnämnda är

$$t(z_n, z_{n-1}, z_{n-2}) = \left| \arg_{(-\pi, \pi]} \frac{z_n - z_{n-1}}{z_{n-1} - z_{n-2}} \right| \quad (5.3)$$

där  $\arg_{(-\pi, \pi]} z$  betecknar argumentet av  $z$  på intervallet  $(-\pi, \pi]$ . För triangelolikhet-medelvärdesfärgläggningen är addendfunktionen

$$t(z_n, z_{n-1}) = \frac{|z_n| - m_n}{M_n - m_n} \quad (5.4)$$

där  $m_n = ||z_{n-1}^p| - |c||$  och  $M_n = |z_{n-1}^p| + |c|$ .

Dessa färgläggningar är kontinuerliga. Om Catmull-Roms kurvor används för interpolation så är triangelolikhet-färgläggningen också glatt. Kurvaturfärgläggningens färgläggningsfunktion innehåller absolutbelopp och är därför inte glatt.

Dessa färgläggningar har en likadan förgrenande struktur. Denna struktur följer ur följande egenskaper hos addendfunktionen.

1. Addendernas derivator  $\frac{dt_n}{dr_k}$  med avseende på iterationernas storlek  $r_k$  är små.
2. Addenderna är periodiska funktioner av iterationernas argument.
3. Varje iteration ökar på perioden med en heltalsfaktor  $p$ .

En medelvärdesfärgläggning som uppfyller dessa egenskaper kallas *förgrenande medelvärdesfärgläggning*. Egenskap 1 medför att färgläggningen varierar tydligare vinkelrätt mot iterationsgränserna och bildar linjer mellan dem. Från egenskap 2 följer att linjerna är jämnt fördelade. Egenskap 3 medför att antalet linjer ökar  $p$ -faldigt. På grund av interpoleringen ser linjerna ut att förgrenas till  $p$  grenar mellan varje iterationsgräns.

För stora värden av  $z_n$  är triangelolikhet-medeltalfärgläggningens addendfunktion approximativt

$$\frac{1}{2} \sin(\arg z_n) + \frac{1}{2}. \quad (5.5)$$

Genom att införa den så kallade randparametern  $s$  fås addendfunktionen

$$t(z_n) = \frac{1}{2} \sin(s \arg z_n) + \frac{1}{2}.$$

Denna addendfunktion definierar *rand-medelvärdesfärgläggningen*. Randparametern  $s$  bestämmer randtätheten hos denna färgläggning.

# Appendix A

## Using Colorings on a Fractal Image

This Appendix illustrates the construction process of a fractal image. It also exemplifies some possible uses of the Smooth Iteration Count and Stripe Average colorings, and shows that the colorings discussed can be applied to other systems than the system  $z_{n+1} = z_n^p + c$  used in earlier examples.

The image is constructed from multiple *layers* each containing an independent fractal image. Layers are combined together using *merge modes* that define how they interact visually [6]. The merge modes used in this example are described in Table A.1.

A fractal set called Rudy's Cubic Mandelbrot, developed by Mark Townsend, was chosen to be used for the image. It is defined by the system

$$z_{n+1} = z_n^3 + dz_n + c$$

where  $d$  is a constant. Both  $z_0$  and  $c$  are set to correspond to the position of the pixel to be calculated. The bailout value  $10^{10}$  and the exponent  $p = 3$  was used for calculating the smooth iteration count.

The image was created using Ultra Fractal version 4.03. At the moment of writing, the Smooth Iteration Count and Stripe Average colorings are available in the Ultra Fractal Formula Database in the file `jh.ucl`. The Ultra Fractal parameter file for the image is available in `jh.upr`.

## A.1 Intermediate Images

Figures A.1 - A.9 present a sequence of intermediate images and associated notes.

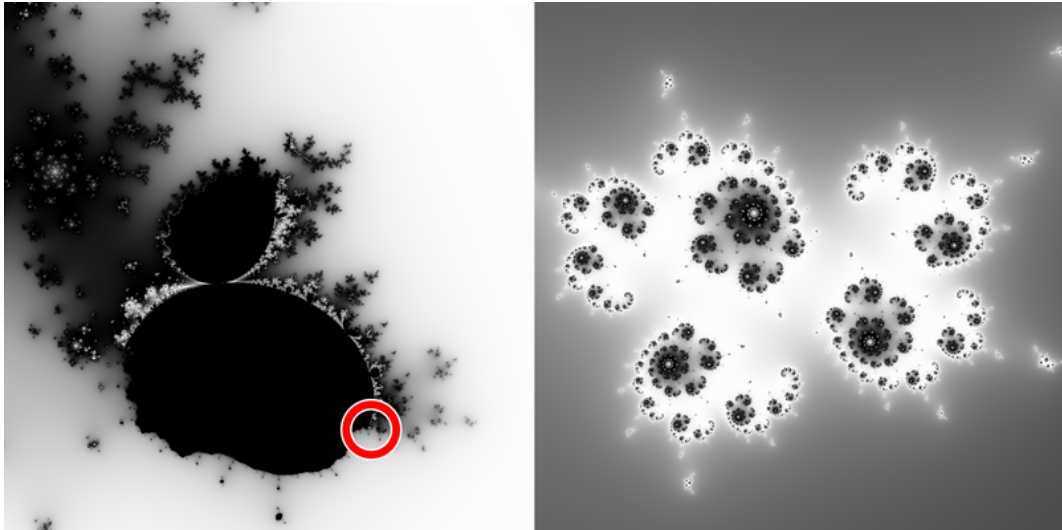


Figure A.1: After trying various values, the constant  $d$  was set to  $-0.7198 + 0.9111i$ . Interesting shapes shown on the right side of the figure were found in the location indicated on the left.

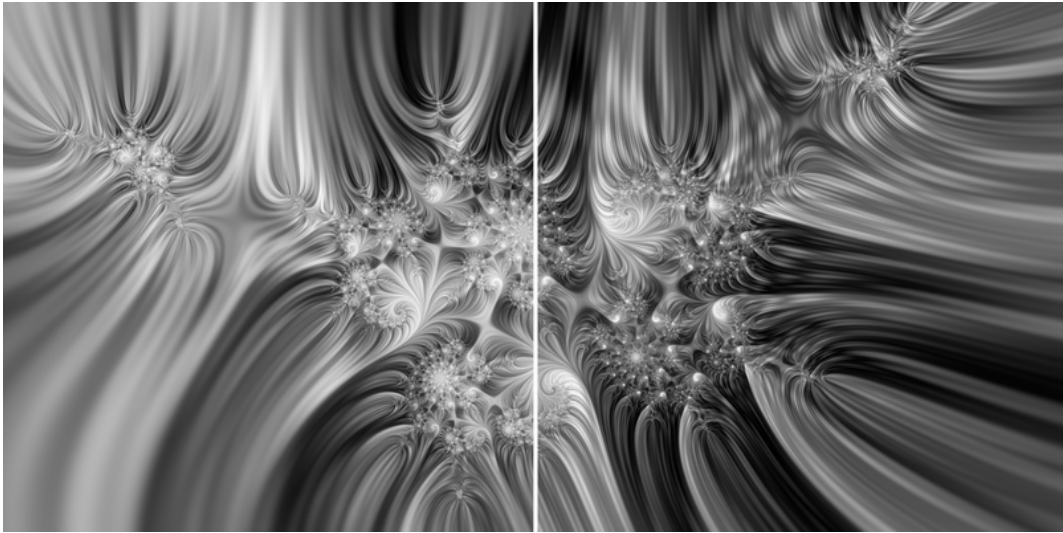


Figure A.2: Three layers using the Stripe Average coloring with stripe densities  $s = 2$ ,  $s = 4$  and  $s = 8$  were created. The opacities of the layers were set to 100%, 50% and 33% so that  $1/3$  of the colors of the final result comes from each layer. This averaging technique has been proposed by Kerry Mitchell [7].

Stripe density  $s = 1$  did not give enough structure whereas  $s = 16$  resulted in too dense stripes. The figure shows the three layers averaged with  $s = 16$  on the right and  $s = 8$  on the left.

The color densities and palettes were adjusted to give a pleasant variation of dark and light areas. This requires that the dark and light spots of the averaged layers partially coincide.



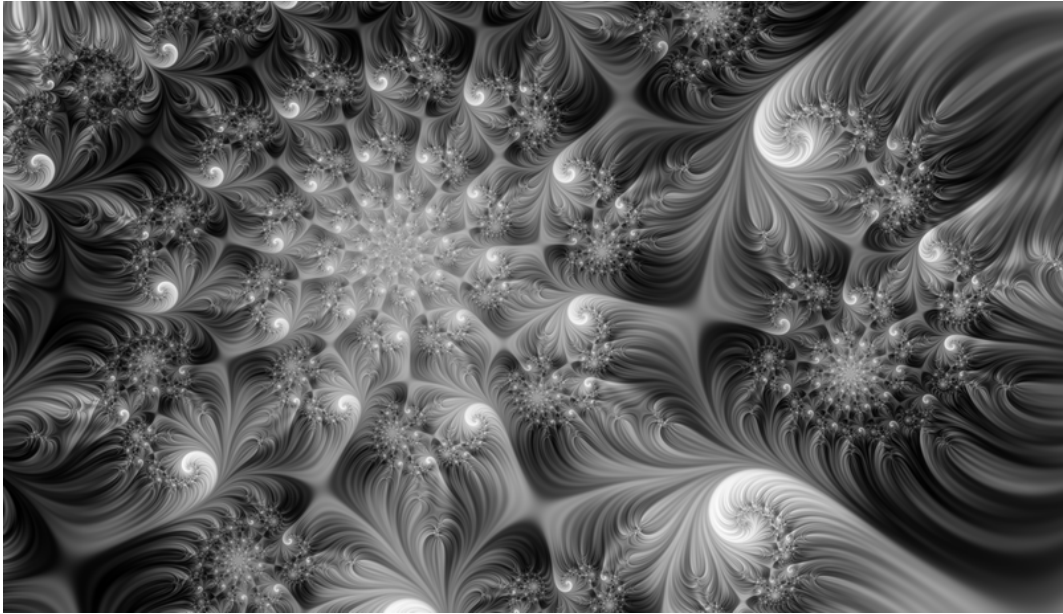


Figure A.3: After some searching, an interesting spiral shape was found.

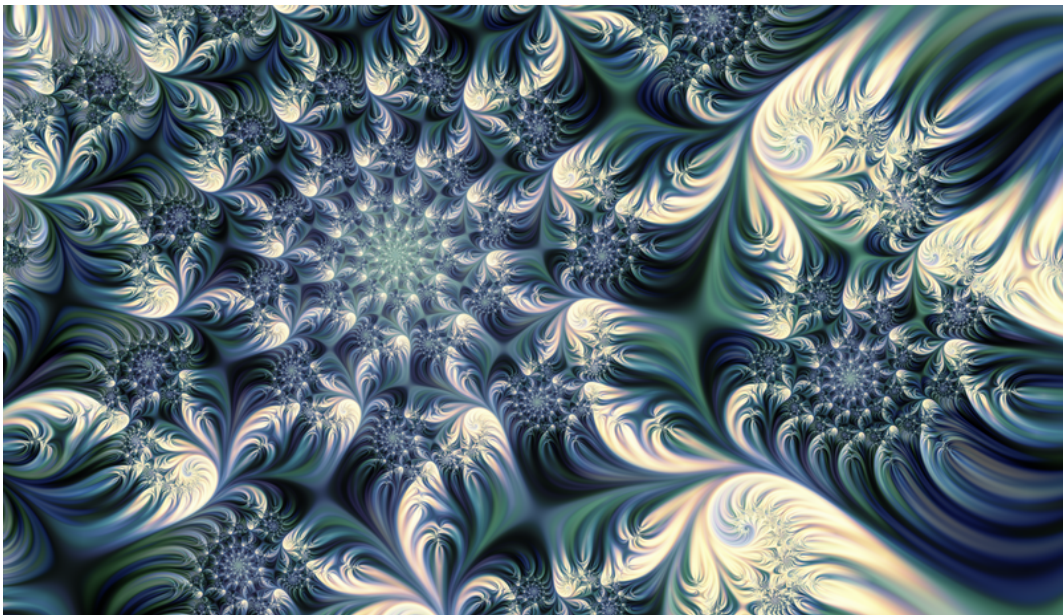


Figure A.4: Colors were added to the averaged layers. Analogous green and blue hues were chosen for each of the three layers in order to create an interesting variation of color. A new, partially transparent Stripe Average layer was added to highlight parts of the image with a color lighter than the background.

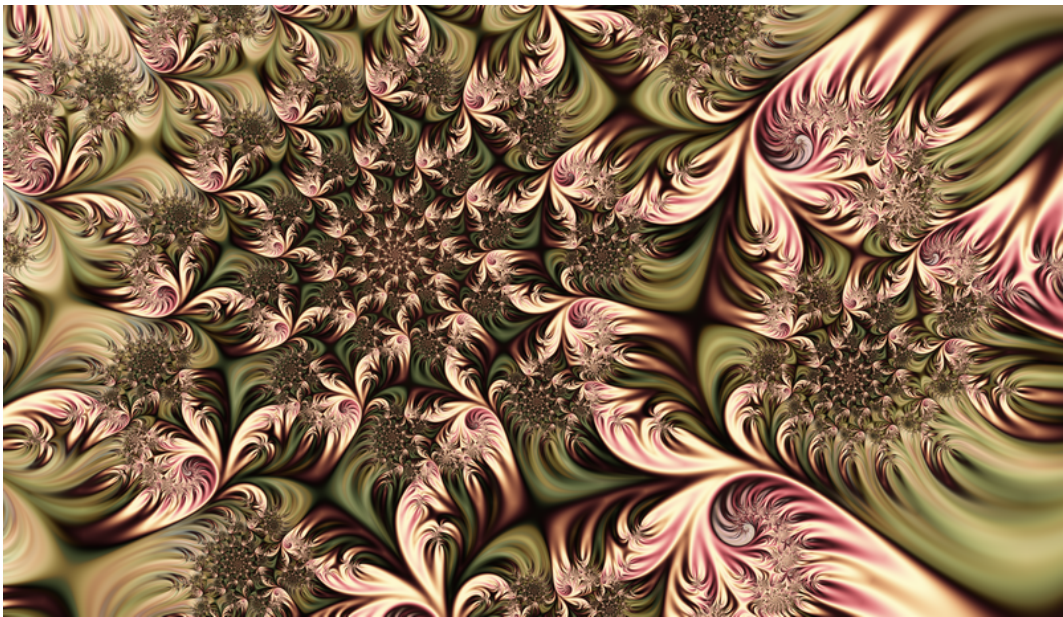


Figure A.5: The colors of the Stripe Average layers were further adjusted. The black color was replaced by transparency on the two topmost averaged layers. This made the final result lighter and removed gray artifacts from the background that emerged from averaging white and black.

A layer using the Smooth Iteration Count coloring was added using the Hue merge mode. However, the layer was removed as it did not improve the appearance of the image.



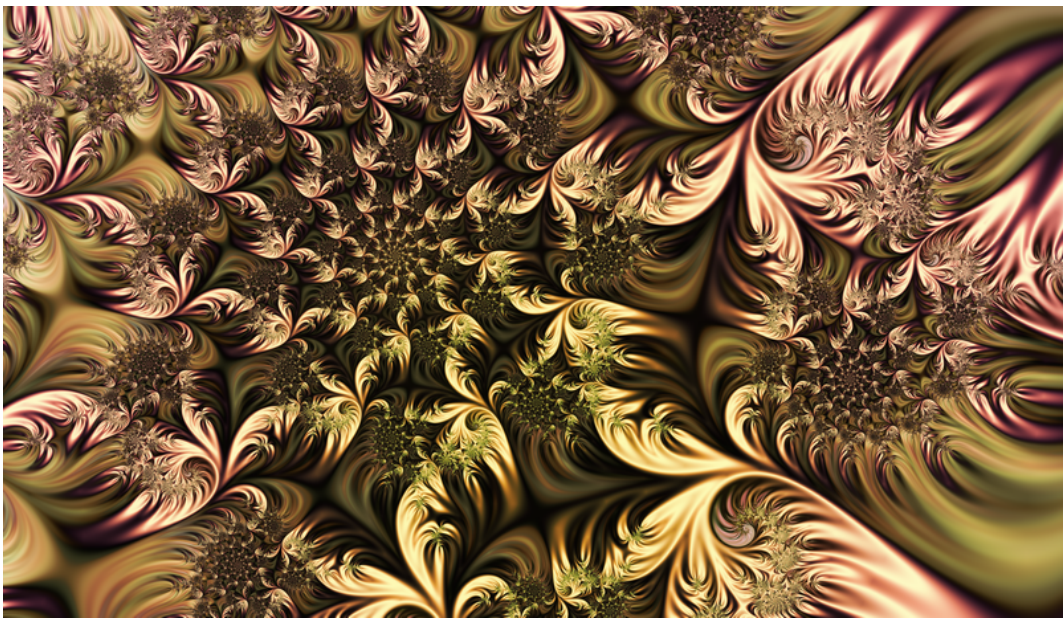


Figure A.6: Two layers with a smooth disk-shaped color transition were added using the overlay merge mode. The first of the layers darkens the background tints. The second one, masked to only affect the highlight layer, transforms the highlights to create more variation in color.

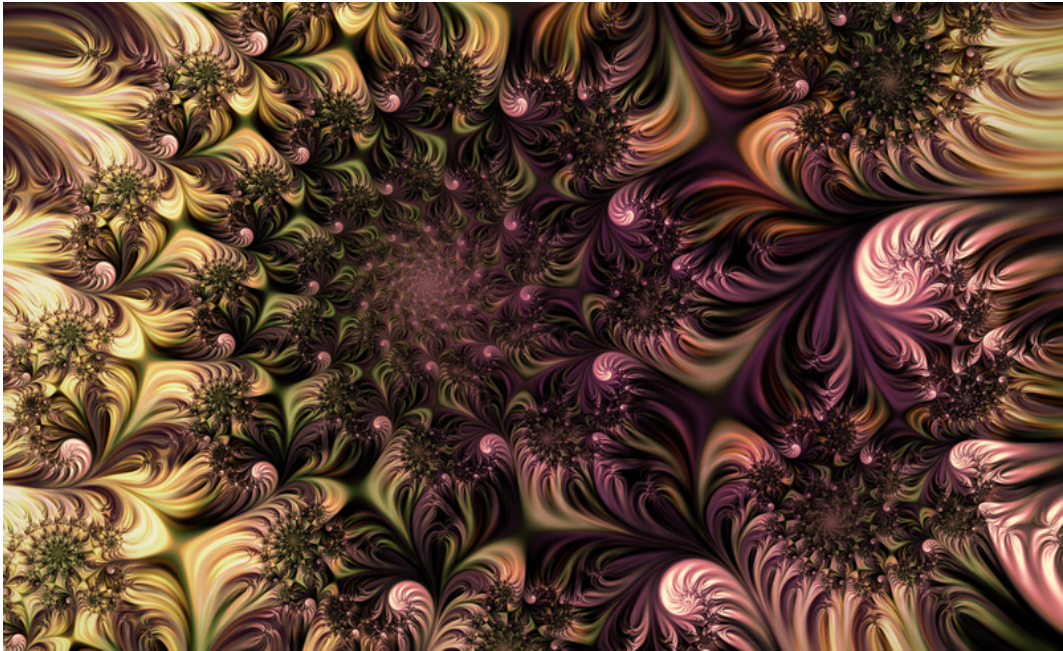


Figure A.7: The size, position and color of the disk overlays were further adjusted. The size of the highlights was reduced. Minor adjustments in color were also made to other layers.

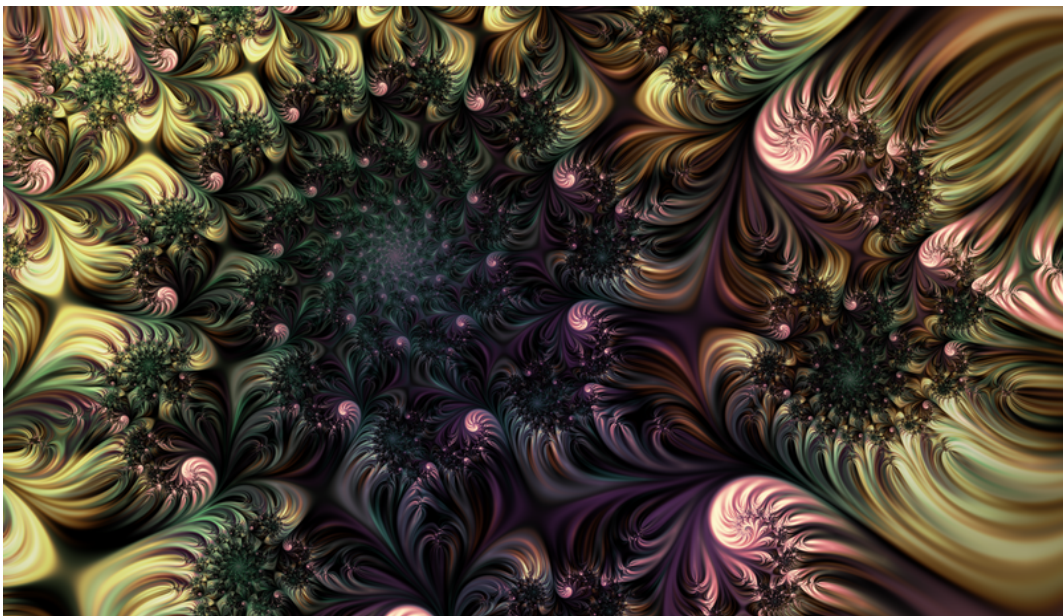


Figure A.8: Another color scheme was made as well, with cold blue and green hues in the spiral. However, the author found the warm hues of Figure A.7 more appealing.





Figure A.9: Two layers using the Smooth Iteration Count were added to highlight the fractal structure. The first layer uses the hard light merge mode with opacity 50% to add color to the background. The second layer uses the overlay merge mode to lighten the fractal structure highlights.

Merge mode	Description
Normal	Directly returns colors from the layer.
Multiply	Multiplies the layer with the underlying layers. The result is always a darker color, thus darkening the underlying layers.
Screen	Multiplies the inverse of the layer with the inverse of the underlying layers, and inverts that again. The result is always a lighter color, thus brightening the underlying layers. Screen is the inverse of Multiply.
Overlay	Multiplies or screens the colors, depending on the color in the underlying layers. Creates color blending effects between the layer and the underlying layers.
Hue	Returns the hue of the layer, and the saturation and luminance of the underlying layers. Colors the underlying layers with the hue of the layer.
Hard Light	Multiplies or screens the colors, depending on the color in the layer. Emphasizes the dark and light regions in the layer, while the areas with medium brightness become transparent.

Table A.1: Description of layer merge modes [8].

## A.2 Conclusion

Tables A.2 - A.4 show the layer contents and their cumulative effect to the resulting image. Layer names, opacities and merge modes used to combine each layer to the underlying cumulative image are given above the corresponding layer images. The merge modes are described in Table A.1.

The Stripe Average coloring is well suited for the layer averaging technique – the result has more interesting structure than a single layer colored using the Stripe Average. This technique also offers more freedom to adjust the final result.

Whereas the Stripe Average coloring has its characteristic striped appearance, the soft color transitions created by the Smooth iteration Count coloring direct the attention of the viewer to the fractal structure. Combined with the use of transparency, it is useful for highlighting the fractal structure without affecting other areas of the image.

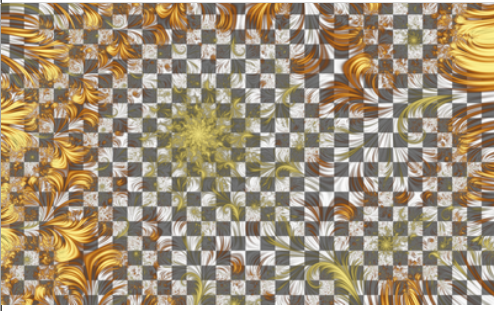
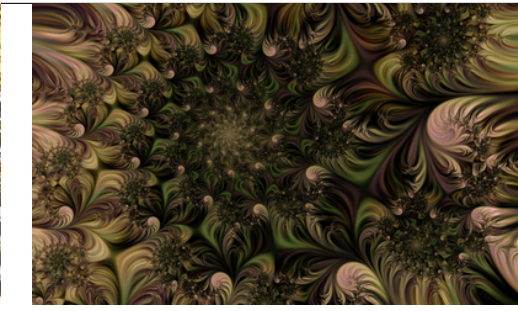
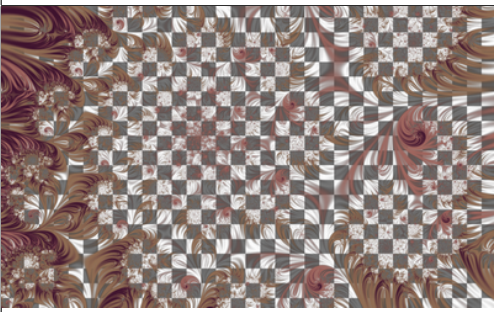
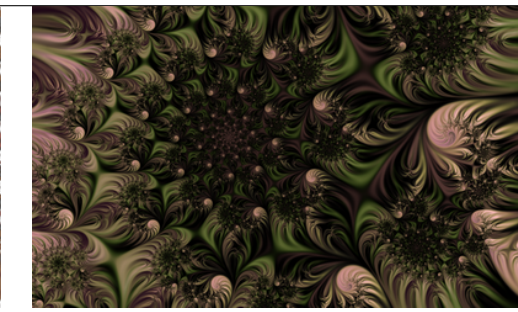
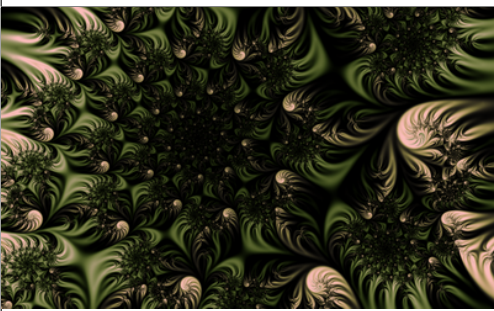
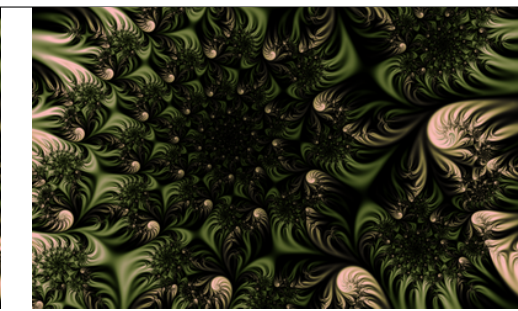
Layer Image	Cumulative Image
3. Stripe Average $s = 8$ (Normal, 33%)	
	
2. Stripe Average $s = 4$ (Normal, 50%)	
	
1. Stripe Average $s = 2$ (Normal, 100%)	
	

Table A.2: Layers 1-3 of the image. Layer name and number is shown above the corresponding images. Merge mode and opacity are given in brackets. The checkerboard pattern in the images indicates transparent areas.




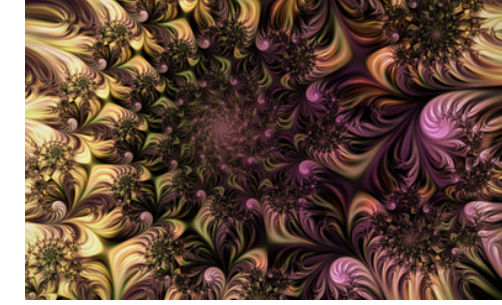

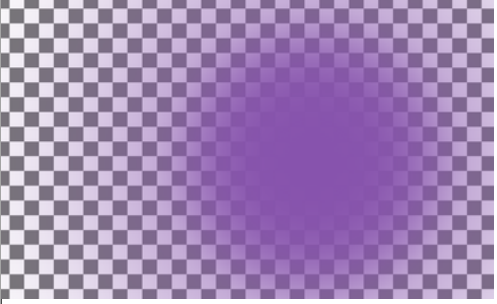
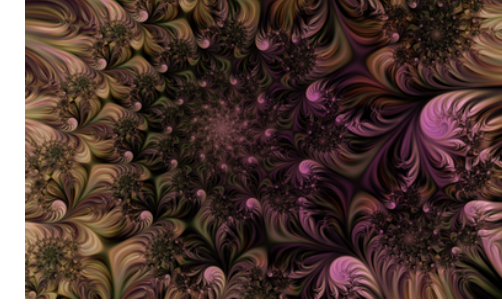
Layer Image	Cumulative Image
6. Lighten background overlay disk (Overlay, 100%)	
	
5. Highlights mask (Mask for layer 6)	
	
4. Violet overlay disk (Overlay, 100%)	
	

Table A.3: Layers 4-6.

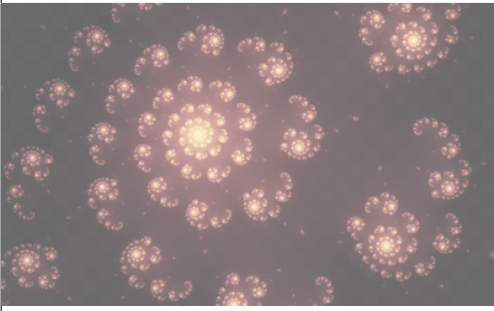
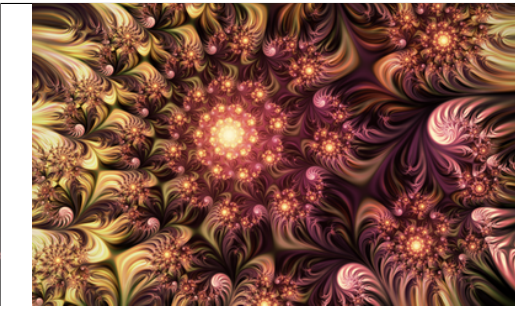
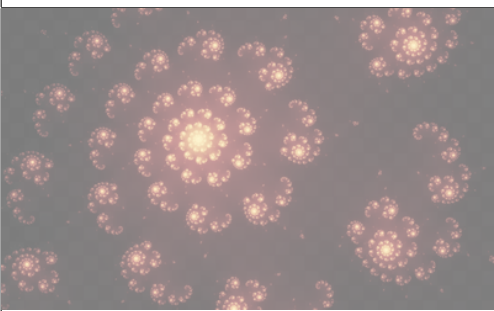
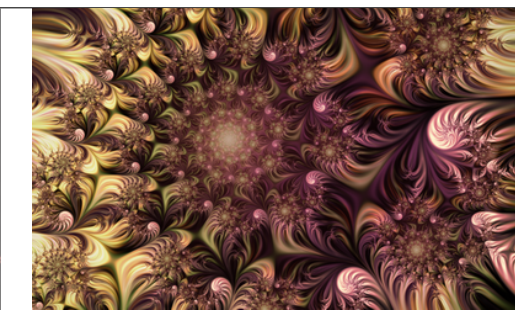
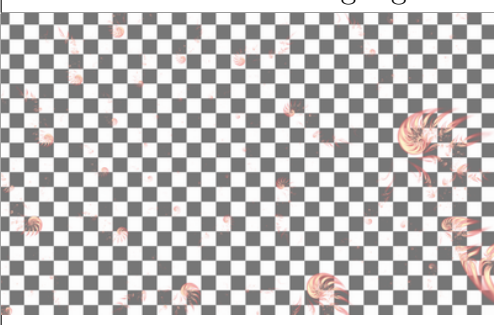
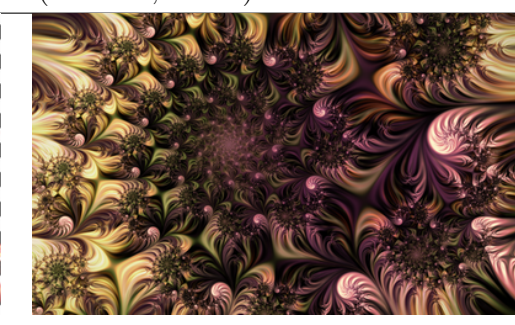
Layer Image	Cumulative Image
9. Highlight spiral overlay (Overlay, 100%)	
	
8. Highlight spiral background (Hard Light, 50%)	
	
7. Highlight knots (Normal, 100%)	
	

Table A.4: Layers 7-9.

# Bibliography

- [1] M. F. Barnsley, *Fractals Everywhere*, Academic Press Inc., London, 1988.
- [2] J. Barrallo, D. M. Jones, *Coloring Algorithms for Dynamical Systems in the Complex Plane*, Visual Mathematics, vol. 1, No. 4, 1999 (April 19th, 2007).  
<http://www.mi.sanu.ac.yu/vismath/javier/index.html>
- [3] E. L. Devaney, *An Introduction to Chaotic Dynamical Systems*, Second Edition, Westview Press, Colorado, 2003.
- [4] D. Hearn, M. P. Baker, *Computer Graphics C Version*, Second edition, Prentice Hall, New Jersey, 1997.
- [5] J. H. Hubbard, B. B. Hubbard, *Vector Calculus, Linear Algebra, and Differential Forms*, Prentice Hall, New Jersey, 1999.
- [6] A. Kelley, *Layering techniques in fractal art*, Computers & Graphics 24 2000, p. 611-616.
- [7] J. Parke, *Working with Ultra Fractal*, course material for lesson 6, 2005 (May 22nd, 2007).  
<http://www.visual-arts-academy.com>
- [8] F. Slijkerman, *Ultra Fractal Help*, 2005 (May 17th, 2007).  
<http://www.ultrafractal.com/help/>