Examensarbete

## Fractals and Computer Graphics

Meritxell Joanpere Salvadó
LiTH - MAT - INT - A - 2011 / 01 - - SE

# Fractals and Computer Graphics 

Applied Mathematics, Linköpings Universitet<br>Meritxell Joanpere Salvadó<br>LiTH - MAT - INT - A - 2011 / $01-$ - SE

Level: A

Supervisor: Milagros Izquierdo,
Applied Mathematics, Linköpings Universitet
Examiner: Milagros Izquierdo,
Applied Mathematics, Linköpings Universitet
Linköpings: June 2011

## Abstract

Fractal geometry is a new branch of mathematics. This report presents the tools, methods and theory required to describe this geometry. The power of Iterated Function Systems (IFS) is introduced and applied to produce fractal images or approximate complex estructures found in nature.
The focus of this thesis is on how fractal geometry can be used in applications to computer graphics or to model natural objects.

Keywords: Affine Transformation, Möbius Transformation, Metric space, Metric Space of Fractals, IFS, Attractor, Collage Theorem, Fractal Dimension and Fractal Tops.

## Acknowledgements

I would like to thank my supervisor and examiner Milagros Izquierdo, at the division of applied mathematics, for giving me the opportunity to write my final degree thesis about Fractals, and for her excellent guidance and her constant feedback.
I also have to thank Aitor Villarreal for helping me with the $\mathrm{LA}_{\mathrm{E}} \mathrm{X}$ language and for his support over these months.
Finally, I would like to thank my family for their interest and support.

## Nomenclature

Most of the reoccurring abbreviations and symbols are described here.

## Symbols

| $d$ | Metric |
| :--- | :--- |
| $(\mathbb{X}, d)$ | Metric space |
| $d_{\mathbb{H}}$ | Hausdorff metric |
| $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$ | Space of fractals |
| $\Omega$ | Code space |
| $\sigma$ | Code |
| $\mathcal{A}$ | Alphabet |
| $l$ | Contractivity factor |
| $f_{n}$ | Contraction mappings |
| $p_{n}$ | Probabilities |
| $N$ | Cardinality (of an IFS) |
| $\mathcal{F}$ | IFS |
| $A$ | Attractor of the IFS |
| $D$ | Fractal dimension |
| $\phi, \varphi$ | Adress function |

## Abbreviations

IFS Iterated function system.
iff if and only if

## List of Figures

1.1 Some orbits of a logistic function ..... 5
1.2 Initial triangle. ..... 8
1.3 Original triangle and the result of applying $f_{1}$ ..... 9
1.4 Self-portrait. ..... 9
1.5 Möbius transformation $f(z)=\frac{1}{z}$ ..... 10
1.6 Möbius transformation $f(z)=\frac{z-i}{z+1}$ ..... 11
1.7 Self-portrait after the translation $f$. ..... 11
1.8 Self-portrait after the Möbius transformation $f(z)=\frac{1}{z}$ ..... 12
1.9 Original self-portrait and the self-portrait after the Möebius trans- formation (really small close to $(0,0))$. ..... 12
2.1 The Hausdorff distance between $A$ and $B$ is 1 . ..... 18
2.2 The Hausdorff distance between $A$ and $B$ is 10 . ..... 18
2.3 A Cauchy sequence of compact sets $A_{n}$. ..... 19
2.4 Addresses of points in the Cantor set. ..... 21
3.1 The Sierpinski triangle is self-similar. ..... 23
3.2 First 4 stages in Cantor set generation. ..... 24
3.3 Stage 0, 1, 2 and 9 of the Koch curve. ..... 24
3.4 Sierpinski triangle ..... 25
3.5 Stages 0,1 and 2 of the Sierpinski triangle. ..... 25
3.6 Stages 1, 2 and 3 of the self-portrait fractal. ..... 25
3.7 Firsts iterations of the Sierpinski Carpet. ..... 26
3.8 Sierpinski pentagon ..... 26
3.93 iterations of the Peano curve construction ..... 26
4.1 Addresses of points for the firsts two steps of the Sierpinski tri- angle transformation ..... 31
4.2 Addresses of some points of the Sierpinski triangle. ..... 32
4.3 Sierpinski triangle constructed with the Deterministic Algorithm. ..... 34
4.4 Fern constructed using the deterministic algorithm. ..... 35
4.5 Random Iteration Algorithm for the Sierpinski triangle. ..... 36
4.6 Random Iteration Algorithm for the Sierpinski triangle. ..... 36
4.7 The result of running the fern random algorithm of program 3.2.2 for $2.000,10.000$ and 25.000 iterations respectively. ..... 37
4.8 The result of running the modified random algorithm (with equal probabilities) for 25.000 iterations. ..... 38
4.9 Both pictures are the same attractor. Colors will help us to solve the problem. ..... 39
4.10 We can approximate the attractor with an IFS ..... 40
5.1 The self-similarity dimension of the Sierpinski triangle is $D=\frac{\ln 3}{\ln 2}$. ..... 42
5.2 The self-similarity dimension of the koch curve is $D=\frac{\ln 4}{\ln 3}$ ..... 42
5.3 Sierpiski triangle ..... 43
5.4 Koch curve ..... 44
5.5 Stage 0 in the construction of the Peano curve ..... 45
5.6 Stage 1 of the contruction of the Peano Curve ..... 45
5.7 Stages 2, 3 and 4 of the Peano curve ..... 46
5.8 Graph of the interpolation function. ..... 48
5.9 Members of the family of fractal interpolation functions corre- sponding to the set of data $\{(0,0),(1,1),(2,1),(3,2)\}$, such that each function has diferent dimension. ..... 50
5.10 FTSE 100 chart of Monday, June 62011. ..... 52
5.11 Part of the Norway's coastline. ..... 52
5.12 Cloud constructed using the Collage theorem ..... 53
5.13 Clouds generated using plasma fractal method compared with real clouds of Linköping ..... 53
6.1 Fractal top produced by colour-stealing. The colours were 'stolen' from the picture on the right ..... 57
6.2 Fractal top produced by colour-stealing. The colours were 'stolen' from the picture on the right ..... 58

## List of Tables

1.1 Various Orbits of $f_{4}(x)=4 x(1-x)$. ..... 4
4.1 IFS code for a Sierpinski triangle ..... 30
4.2 General IFS code ..... 30
4.3 Another IFS code for a Sierpinski triangle ..... 31
4.4 IFS code for a Fern ..... 31
4.5 IFS code for example 3.1.1 ..... 40
5.1 Dimension data for Euclidean d-cubes. ..... 41
5.2 IFS code for an interpolation function ..... 48
6.1 Another IFS code for a Sierpinski triangle ..... 57

## Contents

0 Introduction ..... 1
1 Transformations ..... 3
1.1 Logistic functions ..... 3
1.2 Linear and affine transformations ..... 6
1.2.1 Linear transformations ..... 6
1.2.2 Affine transformations ..... 6
1.3 Möbius transformations ..... 10
2 The metric space of fractals $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$ ..... 13
2.1 Metric spaces and its properties ..... 13
2.1.1 Metric spaces ..... 13
2.1.2 Cauchy sequences, limits and complete metric spaces ..... 14
2.1.3 Compact spaces ..... 15
2.1.4 Contraction mappings ..... 16
2.2 The metric space of fractals ..... 17
2.2.1 The completeness of the space of fractals ..... 18
2.2.2 Contraction mappings on the space of fractals ..... 19
2.3 Adresses and code spaces ..... 20
2.3.1 Metrics of code space ..... 21
3 What is a fractal? ..... 23
3.1 The Cantor Set ..... 24
3.2 Koch curve ..... 24
3.3 Sierpinski triangle ..... 24
3.4 Other examples ..... 25
3.4.1 Self-portrait fractal ..... 25
3.4.2 Sierpinski carpet and Sierpinski pentagon ..... 26
3.4.3 Peano curve ..... 26
4 Iterated Function Systems ..... 27
4.1 IFS ..... 27
4.2 IFS codes ..... 29
4.2.1 The addresses of points on fractals ..... 31
4.3 Two algorithms for computing fractals from IFS ..... 32
4.3.1 The Deterministic Algorithm ..... 33
4.3.2 The Random Iteration Algorithm ..... 33
4.4 Collage theorem ..... 38
5 Fractal dimension and its applications ..... 41
5.1 Fractal dimension ..... 41
5.1.1 Self-similarity dimension ..... 41
5.1.2 Box dimension ..... 42
5.2 Space-filling curves ..... 44
5.3 Fractal interpolation ..... 46
5.3.1 The fractal dimension of interpolation functions ..... 49
5.4 Applications of fractal dimension ..... 51
5.4.1 Fractals in Stock Market ..... 51
5.4.2 Fractals in nature ..... 52
6 Fractal tops ..... 55
6.1 Fractal tops ..... 55
6.2 Pictures of tops: colour-stealing ..... 56

## Chapter 0

## Introduction

This work has been written as the final thesis of the degree "Grau en Matemàtiques" of the Universitat Autònoma de Barcelona. This thesis has been done at Linköpings Universitet due to an Erasmus exchange program organizated between both universities, and has been supervised by Milagros Izquierdo.

Classical geometry provides a first approximation to the structure of physical objects. Fractal geometry is an extension of classical geometry, and can be used to make precise models of physical structures that classical geometry was not able to approximate, because actually mountains are not cones, clouds are not spheres or trees are not cylinders, as Mandelbrot said.

In 1975, Benoit Mandelbrot coined the term fractal when studying selfsimilarity. He also defined fractal dimension and provided fractal examples made with computer. Mandelbrot also defined a very well known fractal called Mandelbrot Set. The study of self-similar objects and similar functions began with Leibnitz in the 17th century and was intense at the end of the 19th century and beginning of 20th century by H. Koch (koch's curve), W.Sierpinski (Sierpinski triangle), G. Cantor (Cantor Set), H. Poincaré (attractor and dynamical systems) and G. Julia (Julia Set), among others. M. Barnsley has developed during the last two decades applications of fractals to computer graphics, for instance he defined the most well known algorithm to draw ferns.

The focus of this thesis is in building fractals models used in computer graphics to represent objects that appear in different areas: nature (forets, fern, clouds), stock market, biology, medical computing, etc. Despite the close relationship between fractals and dynamic systems, we center our attention only on the deformation properties of the spaces of fractals. That will allow us to approximate physical objects by fractals, beginning with one fractal and deforming and adjusting it to get the desired approximation. This work is a study of the so called Collage theorem and its applications on computer graphics, modelling and analysis of data. At the same time, the Collage theorem is a typical example of properties of complete metric spaces: approximation. In the examples in this thesis we use the Collage Theorem to approximate fractals to target images, natural profiles, landscapes, etc.

Chapter One deals with logistic functions and transformations, paying particular attention to affine transformations and Möbius transformations in $\mathbb{R}^{2}$.

Chapter Two introduces the basic topological ideas that are needed to describe the space of fractals $\mathbb{H}(\mathbb{X})$. The concepts introduced include metric spaces, openness, closedness, compactness, completeness, convergence and connectedness. Then the contraction mapping principle is explained. The principal goal of this chapter is to present the metric space of fractals $\mathbb{H}(\mathbb{X})$. Under the right conditions this space is complete and we can use approximation theory to find appropiate fractals.

Once we have defined the metric space of fractals, in Chapter Three we can define a fractal and give some examples of fractal objects. All the examples in this chapter will show one of their properties: the self-similarity. There are non self-similar fractals, like plasma fractals.

In Chapter Four, we learn how to generate fractals by means of simple transformations. We explain what is an iterated function system (IFS) and how it can define a fractal. We present two different algorithms to draw fractals, the Deterministic Algorithm and the Random Iteration Algorithm.
The Collage theorem is presented and will help us to find an IFS for a given compact subset of $\mathbb{R}^{2}$. This theorem allows us to find good fractals that can represent physical objects.

Chapter Five introduces the concept of fractal dimension. The fractal dimension of a set is a number that tells how densely is a set in the space it lies. We gives formulas to compute the fractal dimension of fractals. We also present some applications of fractal dimension and the Collage theorem to computer graphics, such as fractal interpolation or applications of fractals in stocks markets and nature.

Finally, in Chapter Six we introduce the new idea of fractal top and we use computer graphics to plot beautiful pictures of fractals tops using colourstealing. Colour-stealing is a new method that has potential applications in computer graphics and image compression. It consist in 'stealing' colours from an initial picture to 'paint' the new fractal.

## Chapter 1

## Transformations

In this chapter we introduce the chaotic behaviour of logistic functions. This chapter also deals with transformations, with particular attention to affine and Möbius transformations in $\mathbb{R}^{2}$.

We use the notation

$$
f: \mathbb{X} \rightarrow \mathbb{Y}
$$

to denote a function that acts on the space $\mathbb{X}$ to produce values in the space $\mathbb{Y}$. We also call $f: \mathbb{X} \rightarrow \mathbb{Y}$ a transformation from the space $\mathbb{X}$ to the space $\mathbb{Y}$.

Definition 1.0.1. Let $\mathbb{X}$ be a space. A transformation on $\mathbb{X}$ is a function $f: \mathbb{X} \rightarrow \mathbb{X}$, which assings exactly one point $f(x) \in \mathbb{X}$ to each point $x \in \mathbb{X}$.

We say that $f$ is injective (one-to-one) if $x, y \in \mathbb{X}$ with $f(x)=f(y)$ implies $x=y$. Function $f$ is called surjective (onto) if $f(\mathbb{X})=\mathbb{X}$. We say that $f$ is invertible if it is injective and surjective, in this case it is possible to define a transformation $f^{-1}: \mathbb{X} \rightarrow \mathbb{X}$, called the inverse of $f$.

### 1.1 Logistic functions

There is a close relationship between dynamical systems and fractals.
Definition 1.1.1. A dynamical system is a transformation $f: \mathbb{X} \rightarrow \mathbb{X}$ on a metric space $\mathbb{X}$. The orbit of a point $x_{0} \in \mathbb{X}$ under the dynamical system $\{\mathbb{X} ; f\}$ is the sequence of points $\left\{x_{n}=f^{n}\left(x_{0}\right): n=0,1,2, \ldots\right\}$.

The process of determining the long term behavior of orbits of a given dynamical system is known as orbit analysis.

An example of dynamical system are the logistic functions in the space $[0,1]$, that are functions of the form:

$$
f_{c}(x)=c x(1-x), \quad c>0
$$

Since each value of the parameter $c$ gives a distinct function, this is really a family of functions. Using subscripts to indicate time periods, we can write $x_{i+1}=L_{c}\left(x_{i}\right)$, an then rewrite the equation $x_{i+1}=c x_{i}\left(1-x_{i}\right)$.

The fixed points of the logistic function are the solutions of the equation $f_{c}(x)=x$, that is $c x(1-x)=x$. If we solve this quadratic equation we get that one solution is $x=0$ and the other is $x=\frac{c-1}{c}$. This last solution is called the nontrivial fixed point of a logistic function.

Example 1.1.1. Consider the logistic function in the space $\mathbb{X}=[0,1]$ where $c=4, f_{4}(x)=4 x(1-x)$. The process of iterating this function consists of computing a sequence, as follows:

$$
\begin{gathered}
x_{1}=f_{4}\left(x_{0}\right) \\
x_{2}=f_{4}\left(x_{1}\right)=f_{4}\left(f_{4}\left(x_{0}\right)\right) \\
\vdots \\
x_{n}=f_{4}^{n}\left(x_{0}\right)
\end{gathered}
$$

The orbit of the point $x_{0} \in \mathbb{X}$ under the dynamical system $\left\{\mathbb{X} ; f_{4}\right\}$ is the sequence of points $\left\{x_{n}: n=0,1,2, \ldots\right\}$. Applying $f_{4}$ to the endpoints of its domain gives $f_{4}(0)=0$ and $f_{4}(1)=0$, so all successive iterates $x_{i}$ for both $x_{0}=0$ and $x_{0}=1$ yield the value 0 . Thus, we say that 0 is a fixed point of the logistic function $f_{4}$. If we analyse the orbit of this logistic function, we see that in general there is no pattern for a given $x_{0}$, as illustrated in Table 1.1.

| $x_{0}$ | 0.25 | 0.4 | 0.49 | 0.5 | 0.75 |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $x_{1}$ | 0.75 | 0.96 | 1.00 | 1 | 0.75 |
| $x_{2}$ | 0.75 | 0.154 | 0.02 | 0 | 0.75 |
| $x_{3}$ | 0.75 | 0.52 | 0.006 | 0 | 0.75 |
| $x_{4}$ | 0.75 | 0.998 | 0.025 | 0 | 0.75 |
| $x_{5}$ | 0.75 | 0.006 | 0.099 | 0 | 0.75 |
| $x_{6}$ | 0.75 | 0.025 | 0.357 | 0 | 0.75 |
| $x_{7}$ | 0.75 | 0.099 | 0.918 | 0 | 0.75 |
| $x_{8}$ | 0.75 | 0.358 | 0.302 | 0 | 0.75 |
| $x_{9}$ | 0.75 | 0.919 | 0.843 | 0 | 0.75 |
| $x_{1} 0$ | 0.75 | 0.298 | 0.530 | 0 | 0.75 |

Table 1.1: Various Orbits of $f_{4}(x)=4 x(1-x)$.

In the particular case when $x_{0}=0.75$ the orbit converges to the nontrivial fixed point $\frac{4-1}{4}=0.75$, whereas the orbit of $x_{0}=0.5$ converges to the fixed point 0 .

The orbit of a initial point under a logistic function can also be constructed graphically using the algorithm below to generate a construction known as a web diagram.

## Algorithm (Orbit tracing for logistic function)

For a given iterated function $f: \mathbb{R} \rightarrow \mathbb{R}$, the plot consists of a diagonal $y=x$ line and a curve representing $y=f(x)$. To plot the behaviour of a value $x_{0}$, apply the following steps.

1. Given an integer $n$ and an initial variable $x_{0}$.
2. Find the point on the function curve with an x-coordinate of $x_{i}$. This has the coordinates $\left(x_{i}, f\left(x_{i}\right)\right)$.
3. Plot horizontally across from this point to the diagonal line. This has the coordinates $\left(f\left(x_{i}\right), f\left(x_{i}\right)\right)$.
4. Plot vertically from the point on the diagonal to the function curve. This has the coordinates $\left(f\left(x_{i}\right), f\left(f\left(x_{i}\right)\right)\right)$.
5. If $i+1=n$, stop. Otherwise, go to step 2 .


Figure 1.1: From left to right and from top to bottom we have the orbits of $x_{0}=0.25, x_{0}=0.4, x_{0}=0.49$, and $x_{0}=0.5$ respectively, for the logistic function when $c=4$.

In Figure 1.1 we have plotted (using the Maple) some orbits of the logistic function $f_{4}(x)=4 x(1-x)$, for $n=10$. We can observe graphically that the orbit of $x_{0}=0.25$ converges to the fixed point 0.75 and that the orbit of $x_{0}=0.5$ converges to 0 .

As a result, the behavior described by dynamical systems can become extremely complicated and unpredictable. In this cases, very slight differences in the values of these initial conditions may lead to vastly different results. This fact is known as the butterfly effect, because of the sentence "the presence or absence of a butterfly flapping its wings could lead to creation or absence of a hurricane".

### 1.2 Linear and affine transformations

### 1.2.1 Linear transformations

In mathematics, a linear transformation is a function between two vector spaces ${ }^{1}$ that preserves the operations of vector addition and scalar multiplication.

Definition 1.2.1. Let $\mathbb{V}$ and $\mathbb{W}$ be vector spaces over the same field $\mathbb{F}$. Then $f: \mathbb{V} \rightarrow \mathbb{W}$ is called a linear transformation iff

$$
f\left(\alpha x_{1}+\beta x_{2}\right)=\alpha f\left(x_{1}\right)+\beta f\left(x_{2}\right)
$$

for all $\alpha, \beta \in \mathbb{F}$ and all $x_{1}, x_{2} \in V$.
To any linear transformation $f:=\mathbb{R}^{2} \rightarrow \mathbb{R}^{2}$ there corresponds a unique matrix

$$
A:=\left(\begin{array}{ll}
a & b \\
c & d
\end{array}\right)
$$

such that

$$
f\binom{x}{y}=\left(\begin{array}{ll}
a & b \\
c & d
\end{array}\right)\binom{x}{y}=\binom{a x+b y}{c x+d y}
$$

for all $(x, y) \in \mathbb{R}^{2}$ and $a, b, c, d \in \mathbb{R}$. That is,

$$
f(x, y)=(a x+b y, c x+d y)
$$

### 1.2.2 Affine transformations

Definition 1.2.2. A transformation $f: \mathbb{R}^{2} \rightarrow \mathbb{R}^{2}$ of the form

$$
f\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)=\left(\begin{array}{lll}
a & b & e \\
c & d & f \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)
$$

where $a, b, c, d, e, f \in \mathbb{R}$, is called a two-dimensional affine transformation. An affine transformation consist of a lineal transformation followed by a translation.

The basic properties of affine transformations are that they
i) Map straight lines into straight lines.
ii) Preserve ratios of distances between points on straight lines.
iii) Map parallel straight lines into parallel straight lines, trinagles into triangles and interiors of triangles to interiors of triangles.

Definition 1.2.3. A translation is an affine transformation in which the linear part is the identity

$$
f\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)=\left(\begin{array}{lll}
1 & 0 & e \\
0 & 1 & f \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)
$$

where $e, f \in \mathbb{R}$.

[^0]Definition 1.2.4. A similarity with ratio $r$ is an affine transformation $f$ of the Euclidean plane such that for each pair of points $P$ and $Q$,

$$
d(f(P), f(Q))=r d(P, Q)
$$

for some nonzero real number $r>0$. A similarity with ratio $r$ has one of the following matrix representations:

$$
\begin{aligned}
f\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) & =\left(\begin{array}{ccc}
a & b & e \\
-b & a & f \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) \quad \text { (Direct) } \\
f\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) & =\left(\begin{array}{ccc}
a & b & e \\
b & -a & f \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) \quad \text { (Indirect) }
\end{aligned}
$$

where $a^{2}+b^{2}=r^{2}$ and $a, b, e, f \in \mathbb{R}$.
When $r=1$ the affine transformation is an isometry and it preserves the distance, that is $d(X, Y)=d(f(X), f(Y))$.
Example 1.2.1. This transformation is a direct similarity with ratio $r=\frac{1}{2}$.

$$
f\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)=\left(\begin{array}{ccc}
\frac{1}{2} & 0 & 0 \\
0 & \frac{1}{2} & 0 \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)
$$

Definition 1.2.5. A similarity $f: \mathbb{R}^{2} \rightarrow \mathbb{R}^{2}$ can also be express with one of these forms

$$
\begin{aligned}
f\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) & =\left(\begin{array}{ccc}
r \cos \theta & -r \sin \theta & e \\
r \sin \theta & r \cos \theta & f \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) \\
f\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) & =\left(\begin{array}{ccc}
r \cos \theta & r \sin \theta & e \\
r \sin \theta & -r \cos \theta & f \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)
\end{aligned}
$$

for some translation $e, f \in \mathbb{R}, \theta \in[0,2 \pi]$ and $r \neq 0$.
$\theta$ is called the rotation angle while $r$ is called the scaling factor or ratio.
Definition 1.2.6. The linear transformation

$$
f\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)=\left(\begin{array}{ccc}
\cos \theta & -\sin \theta & 0 \\
\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)
$$

is a rotation, where $\theta \in[0,2 \pi]$.

Definition 1.2.7. The linear transformation

$$
f\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)=\left(\begin{array}{ccc}
1 & 0 & 0 \\
0 & -1 & 0 \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)
$$

is a reflection.

Definition 1.2.8. A shear with axis $m$, denoted $S_{m}$, is an affinity that keeps $m$ pointwise invariant and maps every other point $P$ to a point $P^{\prime}$ so that the line $P P^{\prime}$ is parallel to $m$. The matrix representation of a shear with axis $x[0,1,0]$ is

$$
S_{m}\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)=\left(\begin{array}{lll}
1 & j & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)
$$

Definition 1.2.9. A strain with axis $m$, denoted $T_{m}$, keeps $m$ pointwise invariant and maps every other point $P$ to a point $P^{\prime}$ so that the line $P P^{\prime}$ is perpendicular to $m$. The matrix representation of a strain with axis $x[0,1,0]$ is

$$
T_{m}\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)=\left(\begin{array}{lll}
1 & 0 & 0 \\
0 & k & 0 \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)
$$

Theorem 1.2.1. Any affinity can be written as the product of a shear, a strain and a direct similarity.

Example 1.2.2 (Self-portrait). We begin with the triangle of vertices $(0,0)$, $(10,0)$ and $(5,9)$ in Figure 1.2. We will apply some affine transformations to this triangle to construct a self-portrait.


Figure 1.2: This is our initial triangle.
To contruct the "mouth" we will use the following affine transformation.

$$
f_{1}\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)=\left(\begin{array}{ccc}
\frac{1}{3} & 0 & \frac{10}{3} \\
0 & \frac{1}{6} & \frac{10}{6} \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)
$$

Then, we apply this affine transformation to the vertice of our initial triangle. We show how to do it for the first vertice $(0,0)$.

$$
f_{1}\left(\begin{array}{l}
0 \\
0 \\
1
\end{array}\right)=\left(\begin{array}{ccc}
\frac{1}{3} & 0 & \frac{10}{3} \\
0 & \frac{1}{6} & \frac{10}{6} \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
0 \\
0 \\
1
\end{array}\right)=\left(\begin{array}{c}
\frac{10}{3} \\
\frac{5}{3} \\
1
\end{array}\right)
$$

If we apply the affine transformation in the same way for the others two vertices of the triangle, we have that the vertices for the "mouth" are $\left(\frac{10}{3}, \frac{5}{3}\right),\left(\frac{20}{3}, \frac{5}{3}\right)$ and $\left(5, \frac{19}{6}\right)$.


Figure 1.3: Original triangle and the result of applying $f_{1}$.

Figure 1.3 shows both the original triangle and the result of applying the affine transformation $f_{1}$ to construct the "mouth".

Once we have the mouth, we have to construct the two eyes. The affine transformations to construct the left and the right eyes are, respectively:

$$
\begin{aligned}
f_{2}\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) & =\left(\begin{array}{ccc}
\frac{1}{10} & 0 & \frac{7}{2} \\
0 & -\frac{1}{30} & \frac{11}{2} \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) \\
f_{3}\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) & =\left(\begin{array}{ccc}
\frac{1}{10} & 0 & \frac{11}{2} \\
0 & -\frac{1}{30} & \frac{11}{2} \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)
\end{aligned}
$$

Note that $f_{2}, f_{3}$ have a reflection included. Applying this affine transformations to the vertices of the original triangle, we get the new vertices for the left and right eye. These are:

Left eye: $\left(\frac{7}{2}, \frac{11}{2}\right),\left(\frac{9}{2}, \frac{11}{2}\right)$ and $\left(4, \frac{26}{5}\right)$.
Right eye: $\left(\frac{11}{2}, \frac{11}{2}\right),\left(\frac{13}{2}, \frac{11}{2}\right)$ and $\left(6, \frac{26}{5}\right)$.
If we draw all transformations together, with the initial triangle (all filled) we get Figure 1.4.


Figure 1.4: Self-portrait constructet applying $f_{1}, f_{2}$ and $f_{3}$ to the initial triangle.

### 1.3 Möbius transformations

Definition 1.3.1. The set $\mathbb{C} \cup\{\infty\}$ is called the extended complex plane or the Riemann sphere and is denoted by $\hat{\mathbb{C}}$.

Definition 1.3.2. A transformation $f: \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ defined by

$$
f(z)=\frac{(a z+b)}{(c z+d)}
$$

where $a, b, c, d \in \hat{\mathbb{C}}$ and $a d-b c \neq 0$ is called a Möbius transformation on $\hat{\mathbb{C}}$.
Definition 1.3.3. Let $f$ be a Möbius transformation. If $c \neq 0$ we define $f\left(\frac{-d}{c}\right)=\infty$ and $f(\infty)=\frac{a}{c}$. If $c=0$ we define $f(\infty)=\infty$.

Möbius transformations have the property that map the set of all circles and straight lines onto the set of all circles and straight lines. In addition, they preserve angles and its orientation.

Theorem 1.3.1 (Fundamental theorem of Möbius transformations). Let $z_{1}, z_{2}, z_{3}$ and $w_{1}, w_{2}, w_{3}$ be two sets of distinct points in the extended complex plane $\hat{\mathbb{C}}=\mathbb{C} \cup\{\infty\}$. Then there exists a unique Möbius transformation that maps $z_{1}$ to $w_{1}, z_{2}$ to $w_{2}$ and $z_{3}$ to $w_{3}$.

Example 1.3.1. An example of a Möbius transformation is $f(z)=\frac{1}{z}$. As we can see in Figure 1.5, this transformation maps 0 to $\infty, \infty$ to 0 and 1 to 1. The unit circle $\{z \in \mathbb{C}:|z|=1\}$ is invariant as a set.


Figure 1.5: Möbius transformation $f(z)=\frac{1}{z}$

Example 1.3.2. Another example of a Möbius transformation is $f(z)=\frac{z-i}{z+1}$, shown in Figure 1.6, that takes the real line to the unit circle centered at the origin.

To draw Figures 1.5 and 1.6 we have used an applet ${ }^{2}$ that allows us to draw points, lines, and circles, and see what happens to them under a specific Möbius transformation.

Example 1.3.3. Any affine transformation is a Möbius transformation with the point at infinity fixed, i.e that maps $\infty$ to $\infty$.

[^1]

Figure 1.6: Möbius transformation $f(z)=\frac{z-i}{z+1}$

Remark 1.3.1. Any affine transformation is determined by the image of three non-colinear points.

Example 1.3.4 (Self-portrait). We want to apply the Möbius transformation $f(z)=\frac{1}{z}$ to our self-portrait constructed in the Example 1.2.2.
One vertice of the initial triangle is $(0,0)$ and we would have problems mapping the Möbius transformation, so, first of all, we have to apply a translation to the self-portrait.

The translation applied to the all the vertices of the self-portrait (initial triangle, mouth and eyes) is

$$
f\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)=\left(\begin{array}{lll}
1 & 0 & 3 \\
0 & 1 & 3 \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)
$$

So, now, the self-portrait has been moved 3 units to the right and 3 units up. The new vertices for the self-portrait are:
Initial triangle: $(3,3),(13,3)$ and $(8,12)$.
Mouth: $\left(\frac{19}{3}, \frac{14}{3}\right),\left(\frac{29}{3}, \frac{14}{3}\right)$ and ( $8, \frac{37}{6}$ ).
Left eye: $\left(\frac{131}{20}, \frac{42}{5}\right),\left(\frac{151}{20}, \frac{42}{5}\right)$ and $\left(\frac{141}{20}, \frac{81}{10}\right)$
Right eye: $\left(\frac{171}{20}, \frac{42}{5}\right),\left(\frac{191}{20}, \frac{42}{5}\right)$ and $\left(\frac{181}{20}, \frac{81}{10}\right)$


Figure 1.7: Self-portrait after the translation $f$.
Now we can apply the Möbius transformation $f(z)=\frac{1}{z}$ to the new self-
portrait. If we take $z=x+i y$ we can write the transformation like this

$$
f(z)=\frac{1}{z}=\frac{1}{x+i y}=\frac{x-i y}{(x+i y)(x-i y)}=\frac{x-i y}{x^{2}+y^{2}}
$$

So, we can apply the following transformation to all the new vertices of the self-portrait.

$$
f(x, y)=\left(\frac{x}{x^{2}+y^{2}},-\frac{y}{x^{2}+y^{2}}\right)
$$

The result of drawing the new self-portrait after the Möbius transformation $f(z)=\frac{1}{z}$ is shown in Figure 1.8.


Figure 1.8: Self-portrait after the Möbius transformation $f(z)=\frac{1}{z}$
In Figure 1.9 we have plotted the original self-portrait and the self-portrait after the Möbius transformation all in one. Notice that the self-portrait after the transformation is really small compared with the original. Like in Figure 1.5 , points are mapped close to the point $(0,0)$..


Figure 1.9: Original self-portrait and the self-portrait after the Möebius transformation (really small close to $(0,0)$ ).

## Chapter 2

## The metric space of fractals $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$

In this chapter we introduce metric spaces, with the focus of those properties that we will be used later, like the space of fractals: $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$ To know more about metric spaces see [6].

### 2.1 Metric spaces and its properties

In mathematics, a metric space is a set where a notion of distance (called a metric) between elements of the set is defined. See [6].

### 2.1.1 Metric spaces

Definition 2.1.1. A metric space $(\mathbb{X}, d)$ consists of a space $\mathbb{X}$ together with a metric or distance function $d: \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$ that measures the distance $d(x, y)$ between pairs of points $x, y \in \mathbb{X}$ and has the following properties:
(i) $d(x, y)=d(y, x) \forall x, y \in \mathbb{X}$
(ii) $0<d(x, y)<+\infty \forall x, y \in \mathbb{X}, x \neq y$
(iii) $d(x, x)=0 \forall x \in \mathbb{X}$
(iv) $d(x, y) \leq d(x, z)+d(z, y) \forall x, y, z \in \mathbb{X}$ (obeys the triangle inequality)

Example 2.1.1. One example of a metric space is $\left(\mathbb{R}^{2}, d_{\text {Euclidean }}\right)$, where

$$
d_{\text {Euclidian }}(x, y):=\sqrt{\left(x_{1}-y_{1}\right)^{2}+\left(x_{2}-y_{2}\right)^{2}}
$$

for all $x, y \in \mathbb{R}^{2}$.
Metric spaces of diverse types play a fundamental role in fractal geometry. They include familiar spaces like $\mathbb{R}, \mathbb{C}$, code spaces (see section 2.3 ) and many other examples.

We denote by

$$
f:\left(\mathbb{X}, d_{\mathbb{X}}\right) \rightarrow\left(\mathbb{Y} . d_{\mathbb{Y}}\right)
$$

a transformation between two metric spaces $\left(\mathbb{X}, d_{\mathbb{X}}\right)$ and $\left(\mathbb{Y}, d_{\mathbb{Y}}\right)$.

Definition 2.1.2. Two metrics $d$ and $\tilde{d}$ are equivalent if and only if there exists a finite positive constant $C$ such that

$$
\frac{1}{C} d(x, y) \leq \tilde{d}(x, y) \leq C d(x, y) \text { for all } x, y \in \mathbb{X}
$$

Definition 2.1.3. Two metric spaces $\left(\mathbb{X}, d_{\mathbb{X}}\right)$ and $\left(\mathbb{Y}, d_{\mathbb{Y}}\right)$ are equivalent if there is a function $f:\left(\mathbb{X}, d_{\mathbb{X}}\right) \rightarrow\left(\mathbb{Y}, d_{\mathbb{Y}}\right)$ (called a metric transformation) which is injective and surjective (i.e it is invertible), and the metric $d_{\mathrm{X}}$ is equivalent to the metric $d$ given by

$$
\tilde{d}(x, y)=d_{\mathbb{Y}}(f(x), f(y)) \text { for all } x, y \in \mathbb{X} .
$$

Every metric space is a topological space in a natural manner, and therefore all definitions and theorems about general topological spaces also apply to metric spaces.
Definition 2.1.4. Let $S \subset \mathbb{X}$ be a subset of a metric space $(\mathbb{X}, d)$. $S$ is open if for each $x \in S$ there is an $\epsilon>0$ such that $B(x, \epsilon)=\{y \in \mathbb{X}: d(x, y)<\epsilon\} \subset S$. $B(x, \epsilon)$ is called the open ball of radius $\epsilon$ centred at $x$.

Definition 2.1.5. The complement of an open set is called closed. A closed set can be defined as a set which contains all its accumulation points.

Definition 2.1.6. If $(\mathbb{X}, d)$ is a metric space and $x \in \mathbb{X}$, a neighbourhood of $x$ is a set $V$, which contains an open set $S$ containing $x$.
Definition 2.1.7. Let $\mathbb{X}$ be a topological space. Then $\mathbb{X}$ is said to be connected iff the only two subsets of $\mathbb{X}$ that are both open and closed are $\mathbb{X}$ and $\emptyset$.
A subset $S \subset \mathbb{X}$ is said to be connected iff the space $S$ with the relative topology is connected. $S$ is said to be disconnected iff it is not connected.
Definition 2.1.8. Let $\mathbb{X}$ be a topological space. Let $S \subset \mathbb{X}$. Then $S$ is said to be pathwise connected iff whenever, $x, y \in S$ there is a continuous function $f:[0,1] \subset \mathbb{R} \rightarrow S$ such that $x, y \in f([0,1])$.

### 2.1.2 Cauchy sequences, limits and complete metric spaces

In this section we define Cauchy sequences, limits, completeness and continuity. These important concepts are related to the construction and existence of various types of fractals.
Definition 2.1.9. Let $(\mathbb{X}, d)$ be a metric space. Then a sequence of points $\left\{x_{n}\right\}_{n=1}^{\infty} \subset \mathbb{X}$ is said to be a Cauchy sequence iff given any $\epsilon>0$ there is a positive integer $N>0$ such that

$$
d\left(x_{n}, x_{m}\right)<\epsilon \text { whenever } n, m>N
$$

In other words, we can find points as near as wanted by going long enough in the sequence. However, just because a sequence of points moves closer together as one goes along the sequences, we must not infer that they are approaching a point.
Definition 2.1.10. A point $x \in \mathbb{X}$ is said to be an accumulation point of a set $S \subset \mathbb{X}$ if every neighbourhood of $x$ contains infinitely many points of $S$.

Definition 2.1.11. A sequence of points $\left\{x_{n}\right\}_{n=1}^{\infty}$ in a metric space ( $\mathbb{X}, d$ ) is said to converge to a point $x \in \mathbb{X}$ iff given any $\epsilon>0$ there is a positive integer $N>0$ such that

$$
d\left(x_{n}, x\right)<\epsilon \quad \text { whenever } \quad n>N
$$

In this case x is called the limit of $\left\{x_{n}\right\}_{n=1}^{\infty}$, and we write

$$
\lim _{n \rightarrow \infty} x_{n}=x
$$

Theorem 2.1.1. If a sequence of points $\left\{x_{n}\right\}_{n=1}^{\infty}$ in a metric space $(\mathbb{X}, d)$ converge to a point $x \in \mathbb{X}$, then $\left\{x_{n}\right\}_{n=1}^{\infty}$ is a Cauchy sequence.

The converse of this theorem is not true. For example, $\left\{x_{n}=\frac{1}{n}: n=\right.$ $1,2, \ldots\}$ is a Cauchy sequence in the metric space $\left((0,1), d_{\text {Euclidean }}\right)$ but it has no limit in the space. So we make the following definition:

Definition 2.1.12. A metric space $(\mathbb{X}, d)$ is said to be complete iff whenever $\left\{x_{n}\right\}_{n=1}^{\infty}$ is a Cauchy sequence it converges to a point $x \in \mathbb{X}$.

In other words, there actually exists, in the space, a point $x$ to which the Cauchy sequence is converging. This point x is of course the limit of the sequence.

Example 2.1.2. The sequence $\left\{x_{n}=\frac{1}{n}: n=1,2, \ldots\right\}$ converges to 0 in the metric space $[0,1]$. We say that 0 is an accumulation point.

Example 2.1.3. The spaces $\left(\mathbb{R}^{n}, d_{\text {Euclidean }}\right)$ for $n=1,2,3, \ldots$ are complete, but the spaces $\left((0,1), d_{\text {Euclidean }}\right)$ and $\left(B:=\left\{(x, y) \in \mathbb{R}^{2}: x^{2}+y^{2}<1\right\}, d_{\text {Euclidean }}\right)$ are not complete.

Definition 2.1.13. Let $(\mathbb{X}, d)$ and $(\mathbb{Y}, d)$ be metric spaces. Then the function

$$
f:(\mathbb{X}, d) \rightarrow(\mathbb{Y}, d)
$$

is said to be a continuous at a point $x$ iff, given any $\epsilon>0$, there is a $\delta>0$ such that

$$
d(f(x), f(y))<\epsilon \text { whenever } d(x, y)<\delta \text { with } x, y \in \mathbb{X}
$$

We say that $f: \mathbb{X} \rightarrow \mathbb{Y}$ is continuous iff it is continuous at every point $x \in \mathbb{X}$.

### 2.1.3 Compact spaces

Many fractal objects that we will present are construct by a sequence of compact sets. So, we need to define compactness and provide ways of knowing when a set is compact.
Definition 2.1.14. Let $S \subset \mathbb{X}$ be a subset of a metric space $(\mathbb{X}, d)$. $S$ is compact if every infinite sequence $\left\{x_{n}\right\}_{n=1}^{\infty}$ in $S$ contains a subsequence having a limit in $S$.

An equivalent definition of compactness is given here.
Definition 2.1.15. Let $S \subset \mathbb{X}$ be a subset of a metric space $(\mathbb{X}, d)$. $S$ is compact iff for any family $\left\{U_{i}\right\}_{i \in \mathcal{I}}$ of open sets of $\mathbb{X}$ such that $S \subseteq \bigcup_{i \in \mathcal{I}} U_{i}$, there is a finit subfamily $U_{i_{1}}, \ldots, U_{i_{n}}$ that covers $S$.

Definition 2.1.16. Let $S \subset \mathbb{X}$ be a subset of a metric space $(\mathbb{X}, d)$ ). $S$ is bounded if there is a point $a \in \mathbb{X}$ and a number $R>0$ so that

$$
d(a, x)<R \quad \forall x \in \mathbb{X}
$$

Theorem 2.1.2. Let $\mathbb{X}$ be a subspace of $\mathbb{R}^{n}$ with the natural topology. Then the following three properties are equivalent:
(i) $\mathbb{X}$ is compact.
(ii) $\mathbb{X}$ is closed and bounded.
(iii) Each infinite subset of $\mathbb{X}$ has at least one accumulation point in $\mathbb{X}$.

Definition 2.1.17. A metric space ( $\mathbb{X}, d$ ) is said to be a totally bounded iff, for given $\epsilon>0$, there is a finite set of points $\left\{x_{1}, x_{2}, \ldots, x_{L}\right\}$ such that

$$
\mathbb{X}=\bigcup\left\{\left(B\left(x_{l}, \epsilon\right): l=1,2, \ldots, L\right\}\right.
$$

where $B\left(x_{l}, \epsilon\right)$ is the open ball of radius $\epsilon$ centred at $x_{l}$.
Theorem 2.1.3. Let $(\mathbb{X}, d)$ be a complete metric space. Then $\mathbb{X}$ is compact iff it is totally bounded.

### 2.1.4 Contraction mappings

We begin defining a contraction mapping, also called contractive transformation.
Definition 2.1.18. A transformation $f: \mathbb{X} \rightarrow \mathbb{X}$ on a metric space $(\mathbb{X}, d)$ is called contractive or a contraction mapping if there is a constant $0 \leq l<1$ such that

$$
d(f(x), f(y)) \leq l d(x, y) \quad \forall x, y \in \mathbb{X}
$$

Such number $l$ is called a contractivity factor (or ratio) for $f$.
Example 2.1.4. A similarity with ratio $r<1$ is a contractive function.
The following theorem will be used to construct fractal sets.
Theorem 2.1.4 (Contraction mapping theorem). Let $\mathbb{X}$ be a complete metric space. Let $f: \mathbb{X} \rightarrow \mathbb{X}$ be a contraction mapping with contraction factor $l$. Then $f$ has a unique fixed point $a \in \mathbb{X}$. Moreover, if $x_{0}$ is any point in $\mathbb{X}$ and we have $x_{n}=f\left(x_{n-1}\right)$ for $n=1,2,3, \ldots$ then

$$
d\left(x_{0}, a\right) \leq \frac{d\left(x_{0}, x_{1}\right)}{1-l}
$$

and

$$
\lim _{n \rightarrow \infty} x_{n}=a .
$$

Proof
The proof of this theorem starts by showing that $\left\{x_{n}\right\}_{n=0}^{\infty}$ is a Cauchy sequence. Let $a \in \mathbb{X}$ be the limit of this sequence. Now for the continuity of $\mathrm{f} a=f(a)$.

Lemma 2.1.1. Let $f: \mathbb{X} \rightarrow \mathbb{X}$ be a contraction mapping on the metric space $(\mathbb{X}, d)$. Then $f$ is continuous.

Lemma 2.1.2. Let $(\mathbb{X}, d)$ be a complete metric space. Let $f: \mathbb{X} \rightarrow \mathbb{X}$ be a contraction mapping with contractivity factor $0 \leq l \leq 1$, and let the fixed point of $f$ be $a \in \mathbb{X}$. Then

$$
d(x, a) \leq \frac{d(x, f(x))}{1-l} \text { for all } x \in \mathbb{X}
$$

Lemma 2.1.3. Let $\left(P, d_{p}\right)$ be a metric space and $(\mathbb{X}, d)$ be a complete metric space. Let $f: P \times \mathbb{X} \rightarrow \mathbb{X}$ be a family of contraction mappings on $\mathbb{X}$ with contractivity factor $0 \leq l \leq 1$. That is, for each $p \in P, f(p, \cdot)$ is a contraction mapping on $\mathbb{X}$. For each fixed $x \in \mathbb{X}$ let $f$ be continuous on $P$. Then the fixed point of $f$ depends continuously on $p$. That is, $a: P \rightarrow \mathbb{X}$ is continuous.

### 2.2 The metric space of fractals

Let $(\mathbb{X}, d)$ be a metric space such that $\mathbb{R}^{2}$ or $\mathbb{C}$. We will describe the space $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$ of fractals on the space $\mathbb{X} . \mathbb{H}(\mathbb{X})$ is the space of nonempty compact sets of $\mathbb{X}$.

Definition 2.2.1. Let $(\mathbb{X}, d)$ be a complete metric space. Then $\mathbb{H}(\mathbb{X})$ denotes the space whose points are compact subsets of $\mathbb{X}$, other than the empty set.

Definition 2.2.2. Let $(\mathbb{X}, d)$ be a complete metric space and $\mathbb{H}(\mathbb{X})$ denote the space of nonempty compact subsets of $\mathbb{X}$. Then the distance from a point $x \in \mathbb{X}$ to $B \in \mathbb{H}(\mathbb{X})$ is defined by

$$
\mathcal{D}_{B}(x):=\min \{d(x, b): b \in B\}
$$

We refer to $\mathcal{D}_{B}(x)$ as the shortest-distance function from $x$ to the set B.
Now we are going to define the distance from one set to another, that is the distance in $\mathbb{H}(\mathbb{X})$.

Definition 2.2.3. Let $(\mathbb{X}, d)$ be a metric space and $\mathbb{H}(\mathbb{X})$ the space of nonempty compact subsets of $\mathbb{X}$. The distance from $A \in \mathbb{H}(\mathbb{X})$ to $B \in \mathbb{H}(\mathbb{X})$ is defined by

$$
\mathcal{D}_{B}(A):=\max \left\{\mathcal{D}_{B}(a): a \in A\right\}
$$

for all $A, B \in \mathbb{H}(\mathbb{X})$.
Finally we can define the Hausdorff metric.
Theorem 2.2.1. Let $(\mathbb{X}, d)$ be a metric space and $\mathbb{H}(\mathbb{X})$ denote the nonempty compact subsets of $\mathbb{X}$. Let

$$
d_{\mathbb{H}(\mathbb{X})}:=\max \left\{\mathcal{D}_{B}(A), \mathcal{D}_{A}(B)\right\} \quad \text { for all } \quad A, B \in \mathbb{H}(\mathbb{X})
$$

Then $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}(\mathbb{X})}\right)$ is a metric space.
Definition 2.2.4. The metric $d_{\mathbb{H}}=d_{\mathbb{H}(\mathbb{X})}$ is called the Hausdorff metric. The quantity $d_{\mathbb{H}}(A, B)$ is called the Hausdorff distance between the points $A, B \in$ $\mathbb{H}(\mathbb{X})$.


Figure 2.1: The Hausdorff distance between $A$ and $B$ is 1 .

Example 2.2.1. In the following example we compute the Hausdorff distance between $A, B \in \mathbb{H}(\mathbb{X})$, illustrated in Figure 2.1. $A$ is the unit circle and $B$ the circle of radius 2 , both centered at $(0,0)$.
The distance from $A$ to $B$ is $\mathcal{D}_{B}(A):=\max \left\{\mathcal{D}_{B}(a): a \in A\right\}=0$. The distance from $B$ to $A$ is $\mathcal{D}_{A}(B):=\max \left\{\mathcal{D}_{A}(b): b \in B\right\}=1$.
So, the Hausdorff distance between $A$ and $B$ is the maximum of the distances above

$$
d_{\mathbb{H}(\mathbb{X})}(A, B):=\max \left\{\mathcal{D}_{B}(A), \mathcal{D}_{A}(B)\right\}=\max \{0,1\}=1
$$

Example 2.2.2. In this other example, the Hausdorff distance between the two rectangles $A$ and $B$ (see Figure ??) is

$$
d_{\mathbb{H}(\mathbb{X})}(A, B):=\max \left\{\mathcal{D}_{B}(A), \mathcal{D}_{A}(B)\right\}=\max \{5,10\}=10
$$



Figure 2.2: The Hausdorff distance between $A$ and $B$ is 10 .

### 2.2.1 The completeness of the space of fractals

Our principal goal is to establish that the space of fractals $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}(\mathbb{X})}\right)$ is a complete metric space.

Theorem 2.2.2 (Extension lemma). Let $(\mathbb{X}, d)$ be a complete metric space and let $\left\{A_{n} \in \mathbb{H}(\mathbb{X})\right\}_{n=1}^{\infty}$ be a Cauchy sequence in $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$. Consider the Cauchy sequence $\left\{x_{n_{j}} \in A_{n_{j}}\right\}_{j=1}^{\infty}$ in $(\mathbb{X}, d)$, where $\left\{n_{j}\right\}_{j=1}^{\infty}$ is an increasing sequence of positive integers. Then there exists a Cauchy sequence $\left\{x_{n} \in A_{n}\right\}_{n=1}^{\infty}$ in $(\mathbb{X}, d)$ for which $\left\{x_{n_{j}} \in A_{n_{j}}\right\}_{j=1}^{\infty}$ is a subsequence.

The following result provides a general condition under which $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$ is complete and a characterization of the limits of Cauchy sequences in $\mathbb{H}(\mathbb{X})$.


Figure 2.3: A Cauchy sequence of compact sets $A_{n}$ in the space $\mathbb{H}\left(\mathbb{R}^{2}\right)$ converging to a fern set.

Theorem 2.2.3 (The completeness of the space of fractals). Let ( $\mathbb{X}, d$ ) be a complete metric space. Then $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$ is a complete metric space. Moreover, if $\left\{A_{n} \in \mathbb{H}(\mathbb{X})\right\}_{n=1}^{\infty}$ is a Cauchy sequence then

$$
A:=\lim _{n \rightarrow \infty} A_{n}
$$

can be characterized as
$A=\left\{x \in \mathbb{X}:\right.$ there is a Cauchy sequence $\left\{x_{n} \in A_{n}\right\}_{n=1}^{\infty}$ that converges to $\left.x\right\}$.
One of the properties of the space $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{X}}\right)$ is that is pathwise connected. This is used in the applications to computer graphics, to find the attractors.

### 2.2.2 Contraction mappings on the space of fractals

Let $(\mathbb{X}, d)$ be a metric space and let $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$ denote the corresponding space of nonempty compact subsets, with the Hausdorff metric $d_{\mathbb{H}}$.
The following lemma tells us how to construct a contraction mapping on $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$, from a contraction mapping on the metric space ( $\mathbb{X}, d$ ).

Lemma 2.2.1. Let $f: \mathbb{X} \rightarrow \mathbb{X}$ be a contraction mapping on the metric space $(\mathbb{X}, d)$ with contractivity factor $l$. Then $f:=\mathbb{H}(\mathbb{X}) \rightarrow \mathbb{H}(\mathbb{X})$ defined by

$$
f(B)=\{f(x): x \in B\} \quad \forall B \in \mathbb{H}(\mathbb{X})
$$

is a contraction mapping on $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$ with contractivity factor $l$.
We also can combine mappings on $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$ to produce new contraction mappings on $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$. The following lemma provides us a method to do it.

Lemma 2.2.2. Let $(\mathbb{X}, d)$ be a metric space. Let $\left\{f_{n}: n=1,2, \ldots, N\right\}$ be contraction mappings on $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$. Let the contractivity factor for $f_{n}$ be denoted by $l_{n}$ for each $n$. Define $\mathcal{F}: \mathbb{H}(\mathbb{X}) \rightarrow \mathbb{H}(\mathbb{X})$ by

$$
\mathcal{F}=f_{1}(B) \cup f_{2}(B) \cup \cdots \cup f_{N}(B)=\bigcup_{n=1}^{N} f_{n}(B), \text { for each } B \in \mathbb{H}(\mathbb{X})
$$

Then $\mathcal{F}$ is a contraction mapping with contractivity factor $l=\max \left\{l_{n}: n=\right.$ $1,2, \ldots, N\}$.

Lemma 2.2.3. Let $(\mathbb{X}, d)$ be a metric space and suppose we have continuous transformations $f_{n}: \mathbb{X} \rightarrow \mathbb{X}$, for $n=1,2, \ldots, N$ depending continuously on a parameter $p \in P$, where $\left(P, d_{p}\right)$ is a compact metric space. That is $f_{n}(p, x)$ depends continuously on $p$ for fixed $x \in \mathbb{X}$. Then the transformation $\mathcal{F}: \mathbb{H}(\mathbb{X}) \rightarrow$ $\mathbb{H}(\mathbb{X})$ defined by

$$
\mathcal{F}(p, B)=\bigcup_{n=1}^{N} f_{n}(p, B) \quad \forall B \in \mathbb{H}(\mathbb{X})
$$

is also continuous in $p$. That is, $\mathcal{F}(p, B)$ is continuous in $p$ for each $B \in \mathbb{H}(\mathbb{X})$, in the metric space $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$.

### 2.3 Adresses and code spaces

In this section we describe how the points of a space may be organized by addresses. Addresses are elements of certain types of spaces called code spaces. When a space consists of many points, as in the cases of $\mathbb{R}$ and $\mathbb{R}^{2}$, it is often convenient to have addresses for the points in the space. An address of a point is a way to indentify the point.

Example 2.3.1. For example, the address of a point $x \in \mathbb{R}$ may be its decimal expansion. Points in $\mathbb{R}^{2}$ may be addressed by ordered pairs of decimal expansions.

We shall introduce some useful spaces of addresses, namely code spaces. These spaces will be needed later to represent sets of points on fractals, in chapter 4.2.1.

An address is made from an alphabet of symbols. An alphabet $\mathcal{A}$ consists of a nonempty finite set of symbols as $\{1,2, \ldots, N\}$ or $\{0,1, \ldots, N\}$, where each symbol is distinct. The number of symbols in the alphabet is $|\mathcal{A}|$.

Let $\Omega_{\mathcal{A}}^{\prime}$ denote the set of all finite strings made of symbols from the alphabet $\mathcal{A}$. The set $\Omega_{\mathcal{A}}^{\prime}$ includes the empty string $\emptyset$. That is, $\Omega_{\mathcal{A}}^{\prime}$ consists of all expressions of the form

$$
\sigma=\sigma_{1} \sigma_{2} \cdots \sigma_{K}
$$

where $\sigma_{n} \in \mathcal{A}$ for all $1 \leq n \leq k$.
Examples of points in $\Omega_{[1,2,3]}^{\prime}$ are 1111111, 123, 123113 or 2.
A more interesting space for us, which we denote by $\Omega_{\mathcal{A}}$, consists of all infinite strings of symbols from the alphabet $\mathcal{A}$. That is, $\sigma \in \Omega_{\mathcal{A}}$ if and only if it can be written

$$
\sigma=\sigma_{1} \sigma_{2} \cdots \sigma_{n} \cdots
$$

where $\sigma_{n} \in \mathcal{A}$ for all $n \in\{1,2, \ldots\}$.
An example of a point in $\Omega_{[1,2]}$ is $\sigma=121121121111 \cdots$. An example of a point in $\Omega_{[1,2,3]}$ is $\sigma=\overline{2}=22222222222222 \cdots$.
Definition 2.3.1. Let $\varphi: \Omega \rightarrow \mathbb{X}$ be a function from $\Omega=\Omega_{\mathcal{A}}^{\prime} \cup \Omega_{\mathcal{A}}$ onto a space $\mathbb{X}$. Then $\varphi$ is called an address function for $\mathbb{X}$, and points in $\Omega$ are called addresses. $\Omega$ is called a code space. Any point $\sigma \in \Omega$ such that $\varphi(\sigma)=x$ is called an address of $x \in \mathbb{X}$. The set of all addresses of $x \in \mathbb{X}$ is $\varphi^{-1}(\{x\})$.

Example 2.3.2. The Cantor set is an example of the code space $\Omega_{[0,1]}$.

| 0 |  | 1 |  |
| :---: | :---: | :---: | :---: |
| 00 | 01 | 10 | 11 |
| 000001 | 010011 | 100101 | 110111 |
| -- | -- | -- -- | -- -- |

Figure 2.4: Addresses of points in the Cantor set.

### 2.3.1 Metrics of code space

We give two examples of metric for any code space $\Omega=\Omega_{\mathcal{A}}^{\prime} \cup \Omega_{\mathcal{A}}$.
A simple metric on $\Omega_{\mathcal{A}}$ is defined by $d_{\Omega}(\sigma, \sigma)=0$ for all $\sigma \in \Omega_{\mathcal{A}}$, and

$$
d_{\Omega}(\sigma, \omega):=\frac{1}{2^{m}} \text { if } \sigma \neq \omega,
$$

for $\sigma=\sigma_{1} \sigma_{2} \sigma_{3} \cdots$ and $\omega=\omega_{1} \omega_{2} \omega_{3} \cdots \in \Omega_{A}$, where $m$ is the smallest positive integer such that $\sigma_{m} \neq \omega_{m}$.

We can extend $d_{\Omega}$ to $\Omega_{\mathcal{A}}^{\prime} \cup \Omega_{\mathcal{A}}$ by adding a symbol, which we will call $Z$, to the alphabet $\mathcal{A}$ to make a new alphabet $\tilde{\mathcal{A}}=\mathcal{A} \cup\{Z\}$. Then we embed $\Omega_{\mathcal{A}}^{\prime} \cup \Omega_{\mathcal{A}}$ in $\Omega_{\tilde{\mathcal{A}}}$ via the function $\varepsilon: \Omega_{\mathcal{A}}^{\prime} \cup \Omega_{\mathcal{A}} \rightarrow \Omega_{\tilde{\mathcal{A}}}$ defined by

$$
\begin{gathered}
\varepsilon=\sigma Z Z Z Z Z Z Z \cdots=\sigma \bar{Z} \text { if } \sigma \in \Omega_{\mathcal{A}}^{\prime} \\
\varepsilon(\sigma)=\sigma \text { if } \sigma \in \Omega_{\mathcal{A}}
\end{gathered}
$$

and we define

$$
d_{\Omega}(\sigma, \omega)=d_{\Omega}(\varepsilon(\sigma), \varepsilon(\omega)) \text { for all } \sigma, \omega \in \Omega_{\mathcal{A}}^{\prime} \cup \Omega_{\mathcal{A}}
$$

There is another metric that we can define on $\Omega_{\mathcal{A}}^{\prime} \cup \Omega_{\mathcal{A}}$. It depends on the number of elements $|\mathcal{A}|$ in the alphabet $\mathcal{A}$, so we denote it by $d_{|\mathcal{A}|}$. Assume that $\mathcal{A}=\{0,1, \ldots, N-1\}$ and the number of elements of the alphabet is $|\mathcal{A}|=N$. This metric is defined on $\Omega_{\mathcal{A}}$ by

$$
d_{|\mathcal{A}|}=\left|\sum_{n=1}^{\infty} \frac{\sigma_{n}-\omega_{n}}{(|\mathcal{A}|+1)^{n}}\right| \text { for all } \sigma, \omega \in \Omega_{\mathcal{A}} .
$$

Finally we extend $d_{|\mathcal{A}|}$ to the space $\Omega_{\mathcal{A}}^{\prime} \cup \Omega_{\mathcal{A}}$ using the same construction as above and defining $\xi:=\Omega_{\mathcal{A}}^{\prime} \rightarrow[0,1]$ such that

$$
\xi\left(\sigma_{1} \sigma_{2} \cdots \sigma_{n}\right)=0 . \sigma_{1} \sigma_{2} \cdots \sigma_{n} \bar{Z}
$$

that is,

$$
\xi(\sigma)=\sum_{n=1}^{m} \frac{\sigma_{n}}{(N+1)^{n}}+\frac{1}{(N+1)^{m}}
$$

for all $\sigma=\sigma_{1} \sigma_{2} \cdots \sigma_{n} \in \Omega_{\mathcal{A}^{\prime}}$.
We define

$$
d_{|\mathcal{A}|}(\sigma, \omega)=|\xi(\sigma)-\xi(\omega)|=d_{\text {Euclidean }}(\xi(\sigma), \xi(\omega)) \text { for all } \sigma, \omega \in \Omega_{\mathcal{A}}^{\prime} \cup \Omega_{\mathcal{A}} .
$$

Theorem 2.3.1. Both $\left(\Omega_{\mathcal{A}}^{\prime} \cup \Omega_{\mathcal{A}}, d_{\Omega}\right)$ and $\left(\Omega_{\mathcal{A}}^{\prime} \cup \Omega_{\mathcal{A}}, d_{|\mathcal{A}|}\right)$ are metric spaces.
The code space has the properties of a metric space.
Theorem 2.3.2. The metric spaces $\left(\Omega_{\mathcal{A}} \cup \Omega_{\mathcal{A}}^{\prime}, d_{\Omega}\right)$ and $\left(\Omega_{\mathcal{A}}^{\prime} \cup \Omega_{\mathcal{A}}, d_{|\mathcal{A}|}\right)$ are complete.

Theorem 2.3.3. The code space $\Omega=\Omega_{\mathcal{A}} \cup \Omega_{\mathcal{A}}^{\prime}$ is compact.

## Chapter 3

## What is a fractal?

In this chapter we give a definition of fractal, and introduce one of its properties, the self-similarity. Finally we present some examples of fractal objects.

Once we have defined the space of fractals $\left(\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)\right.$, we can define a fractal.

Definition 3.0.2. Let $(X, d)$ be a metric space. We say that a fractal is a subset of $\left(\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)\right.$. In particular, is a fixed point of a contractive function on $\left(\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)\right.$.

A fractal is a geometric object that is repeated at ever smaller scales to produce irregular shapes that cannot be represented by classical geometry. We say that they are self-similar.

An object is said to be self-similar if it looks "roughly" the same on any scale. The Sierpinski triangle in Figure 3.1 is an example of a self-similar fractal. If we zoom the red triangle we see that it is similar to the first one. This occurs in all scales.


Figure 3.1: The Sierpinski triangle is self-similar.
In chapter 4 we will see that a fractal is invariant under certain tranformations of $X$.

In the following subsections we are going to introduce specific examples of fractals.

### 3.1 The Cantor Set

The cantor set is generated by beginning with a segment (usually of length 1 ) and removing the open middle third of this segment. The process of removing the open middle third of each remaining segment is repeated for each of the new segments.
Figure 3.2 shows the first five stages in this generation.


Figure 3.2: First 4 stages in Cantor set generation.

### 3.2 Koch curve

The Koch curve is another well known fractal. To construct it begin with a straight line. Divide it into three equal segments and replace the middle segment by the two sides of an equilateral triangle of the same length as the segment being removed. Now repeat the same construction for each of the new four segments. Continue these interations.



Figure 3.3: Stage 0, 1, 2 and 9 of the Koch curve.

### 3.3 Sierpinski triangle

Without a doubt, Sierpinski's Triangle is at the same time one of the most interesting fractals and one of the most simple to construct.


Figure 3.4: Sierpinski triangle

One simple way of generating the Sierpinski Triangle in Figure 3.4 is to begin with a triangle. Connect the midpoints of each side to form four separate triangles, and cut out the triangle in the center. For each of the three remaining triangles, perform this same act. Iterate infinitely. The firsts iterations of the sierpinski triangle are presented in Figure 3.5.


Figure 3.5: Stages 0,1 and 2 of the Sierpinski triangle.

### 3.4 Other examples

In this subsection we show other examples of fractals.

### 3.4.1 Self-portrait fractal

Here we have a fractal constructed applying repeatedly the affine transformations seen in section 1.2.


Figure 3.6: Stages 1, 2 and 3 of the self-portrait fractal.

### 3.4.2 Sierpinski carpet and Sierpinski pentagon

The Sierpinksi carpet is a generalization of the Cantor set to two dimensions. The construction of the Sierpinski carpet begins with a square. The square is cut into 9 congruent subsquares in a 3 -by- 3 grid, and the central subsquare is removed. The same procedure is then applied recursively to the remaining 8 subsquares, and infinitum.


Figure 3.7: Firsts iterations of the Sierpinski Carpet.
The sierpinski pentagon, a fractal with 5 -fold simmetry, is formed starting with a pentagon and using similar rules that in the sierpinski triangle but for pentagons.


Figure 3.8: Sierpinski pentagon

### 3.4.3 Peano curve

The Peano curve is created by iterations of a curve. The limit of the Peano curve is a space-filling curve, whose range contains the entire 2-dimensional unit square. In Figure 3.9 we can see the firts 3 iterations of the curve. We will explore more this curve in chapter 5.2.




Figure 3.9: 3 iterations of the Peano curve construction.

## Chapter 4

## Iterated Function Systems

A fractal set generally contains infinitely many points whose organization is so complicated that it is not possible to describe the set by specifying directly where each point in it lies. Instead, the set may be defined by the 'relations between the pieces'. [Barnsley]

Iterated function systems provide a convenient framework for the description, classification and expression of fractals. Two algorithms, the Random Iteration Algorithm and the Deterministic Algorithm, for computing pictures of fractals, are presented. Finally, the collage theorem characterises an iterated function system whose attractor is close to a given set. All the results here can be found in [1] [2].

### 4.1 IFS

So far the examples of fractals we have seen are all strictly self-similar, that is, each can be tiled with congruent tiles where the tiles can be mapped onto the original using similarities with the same scaling-factor; or inversely, the original object can be mapped onto the individual tiles using similarities with a common scaling factor.

In general, modelize such complicated objects require involved algorithms, but one can develope quite simple algorithms by studying the relations between parts of a fractal that allow us to use relative small sets of affine transformations.

The set of Sierpinski transformations is an example of an iterated function system (IFS) consisting of three similarities of ratio $r=\frac{1}{2}$. Since $r<1$, the transformations are contractive, that is, the transformations decrease the distance between points making image points closer together than their corresponding pre-images. When the three transformations are iterated as a system they form the Sierpinski triangle.
In general, an iterated function system consists of affine transformations, this allowing direction specific scaling factors as well as changes in angles. We formalize these ideas in the following definitions.

Definition 4.1.1. An iterated function system consists of a complete metric space $(\mathbb{X}, d)$ together with a finite set of contraction mappings (see section 2.1.4) $f_{n}: \mathbb{X}$ for $n=1,2, \ldots, N$, where $N \geq 1$. The abbreviation "IFS" is used for
"iterated function system". It may be denoted by

$$
\left\{\mathbb{X} ; f_{1}, f_{2}, \ldots, f_{N}\right\} \quad \text { or } \quad\left\{\mathbb{X} ; f_{n}, n=1,2, \ldots, N\right\} .
$$

Moreover, if $\left\{f_{1}, f_{2}, \ldots, f_{N}\right\}$, is a finite sequence of strictly contractive transformations, $f_{n}: \mathbb{X} \rightarrow \mathbb{X}$, for $n=1,2, \ldots, N$, then $\left\{\mathbb{X} ; f_{1}, f_{2}, \ldots, f_{N}\right\}$ is called a strictly contractive IFS or a hyperbolic IFS.

We say that a transformation $f_{n}: \mathbb{X} \rightarrow \mathbb{X}$ is strictly contractive if and only if there exists a number $l_{n} \in[0,1)$ such that

$$
d\left(f_{n}(x), f_{n}(y)\right) \leq l_{n} d(x, y)
$$

for all $x, y \in \mathbb{X}$. The number $l_{n}$ is called a contractivity factor for $f_{n}$ and the number

$$
l=\max \left\{l_{1}, l_{2}, \ldots, l_{N},\right\}
$$

is called a contractivity factor for the IFS.
We use such terminology as 'the $\operatorname{IFS}\left\{\mathbb{X} ; f_{1}, f_{2}, \ldots, f_{N}\right\}$ and 'Let $\mathcal{F}$ denote and IFS'.

The following theorem is the cornerstone of the theory of fractals. The theorem gives us the algorithm to create a fractal using contractive affine transformations.

Theorem 4.1.1. Let $\left\{\mathbb{X} ; f_{n}, n=1,2, \ldots, N.\right\}$ be a hyperbolic iterated function system with contractivity factor $l$. Then the transformation $\mathcal{F}: \mathbb{X}(\mathbb{H}) \rightarrow \mathbb{H}(\mathbb{X})$ defined by

$$
\mathcal{F}(B)=f_{1}(B) \cup f_{2}(B) \cup \ldots \cup f_{N}(B)=\bigcup_{n=1}^{N} f_{n}(B)
$$

for all $B \in \mathbb{H}(\mathbb{X})$, is a contraction mapping on the complete metric space $\left(\mathbb{H}(\mathbb{X}), d_{\mathbb{H}}\right)$ with contractivity factor $l$. That is

$$
h(\mathcal{F}(B), \mathcal{F}(C)) \leq l \cdot h(B, C)
$$

for all $B, C \in \mathbb{H}(\mathbb{X})$.
Its unique fixed point, $A \in \mathbb{H}(\mathbb{X})$ obeys the self-referential equation

$$
A=f_{1}(A) \cup f_{2}(A) \cup \ldots \cup f_{N}(A)=\bigcup_{n=1}^{N} f_{n}(A)
$$

and is given by $A=\lim _{n \rightarrow \infty} \mathcal{F}^{n}(B)$ for any $B \in \mathbb{H}(\mathbb{X})$.
Definition 4.1.2. The fixed point $A \in \mathbb{H}(\mathbb{X})$ described in the theorem is called the attractor of the IFS.

The following theorem establish the continuous dependence of the attractor of a hyperbolic IFS on parameters in the maps of the IFS.

Theorem 4.1.2. Let $(\mathbb{X}, d)$ be a metric space. Let $\left\{\mathbb{X} ; f_{n}, n=1,2, \ldots, N.\right\}$ be a hyperbolic iterated function system with contractivity factor $l$. For $n=$ $1,2, \ldots, N$, let $f_{n}$ depend continuously on a parameter $p \in P$, where $P$ is a compact metric space. Then the attractor $A(p) \in \mathbb{H}(\mathbb{X})$ depends continuously on $p \in P$, with respect to the Hausdorff metric $d_{\mathbb{H}}$.

Theorem 4.1.2 says that small changes in the parameters will lead to small changes on the attractor. This is very important because we can continuously control the attractor of an IFS, by varying parameters on the transformations. We will use it in the applications to computer graphics (collage theorem or fractal interpolation), to find the attractors that we want.

Example 4.1.1. The set of Sierpinski transformations is an example of an iterated function system (IFS) consisting of three similarities of ratio $r=\frac{1}{2}$. In this case the iterated function system consist of the complete metric space $\mathbb{R}^{2}$ together with a finite set of contraction mappings $f_{n}: \mathbb{R}^{2} \rightarrow \mathbb{R}^{2}$ for $n=1,2,3$. Here we have three contraction mappings to generate the fractal:

$$
\begin{gathered}
f_{1}(x, y)=\left(\frac{x}{2}, \frac{y}{2}\right) \\
f_{2}(x, y)=\left(\frac{x}{2}, \frac{y}{2}+1\right) \\
f_{3}(x, y)=\left(\frac{x}{2}+1, \frac{y}{2}+1\right)
\end{gathered}
$$

In this case, the contractivity factor for the IFS is

$$
l=\max \left\{l_{1}, l_{2}, l_{3}\right\}=\max \left\{\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right\}=\frac{1}{2}
$$

The attractor of this IFS is the Sierpinski triangle in Figure 3.1.
Example 4.1.2. The Iterated Function System for the self-portrait fractal (Figure 3.6) consists of the following three tranformations:

$$
\begin{aligned}
& f_{1}\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)=\left(\begin{array}{ccc}
\frac{1}{3} & 0 & \frac{10}{3} \\
0 & \frac{1}{6} & \frac{5}{3} \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) \\
& f_{2}=\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)\left(\begin{array}{ccc}
\frac{1}{10} & 0 & \frac{7}{2} \\
0 & -\frac{1}{30} & \frac{11}{2} \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) \\
& f_{3}=\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)\left(\begin{array}{ccc}
\frac{1}{10} & 0 & \frac{11}{2} \\
0 & \frac{-1}{30} & \frac{11}{2} \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)
\end{aligned}
$$

### 4.2 IFS codes

Here we describe the notation used to implement IFS, called IFS codes.
For simplicity we restrict attention to hyperbolic IFS of the form $\left\{\mathbb{R}^{2}: f_{n}: n=\right.$ $1,2, \ldots, N\}$, where each mapping is an affine transformation.
As we have seen in chapter 1 each affine transformation is given by a matrix.
We are going to illustrate the IFS described in Example 4.1.1, whose attractor is a Sierpinski triangle, in the matrix form:

$$
f_{1}\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)=\left(\begin{array}{ccc}
0.5 & 0 & 0 \\
0 & 0.5 & 0 \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)
$$

$$
\begin{aligned}
f_{2}\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) & =\left(\begin{array}{ccc}
0.5 & 0 & 0 \\
0 & 0.5 & 1 \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) \\
f_{3}\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) & =\left(\begin{array}{ccc}
0.5 & 0 & 1 \\
0 & 0.5 & 1 \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)
\end{aligned}
$$

Table 4.1 is another way of representing the same iterated function system presented in example 4.1.1.


Table 4.1: IFS code for a Sierpinski triangle

So, in general we can represent each mapping transformation using this matrix form

$$
f_{n}\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)=\left(\begin{array}{ccc}
\mathfrak{a}_{\mathfrak{n}} & \mathfrak{b}_{\mathfrak{n}} & \mathfrak{e}_{\mathfrak{n}} \\
\mathfrak{c}_{\mathfrak{n}} & \mathfrak{o}_{\mathfrak{n}} & \mathfrak{f}_{\mathfrak{n}} \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) \text { for } n=1,2, \ldots, N .
$$

A tidier way of representing a general iterated function system is given in table 4.2.

| $n$ | $\mathfrak{a}$ | $\mathfrak{b}$ | $\mathfrak{c}$ | $\mathfrak{d}$ | $\mathfrak{e}$ | $\mathfrak{f}$ | $p$ |
| ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: |
| 1 | $\mathfrak{a}_{1}$ | $\mathfrak{b}_{1}$ | $\mathfrak{c}_{1}$ | $\mathfrak{d}_{1}$ | $\mathfrak{e}_{1}$ | $\mathfrak{f}_{1}$ | $p_{1}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| n | $\mathfrak{a}_{\mathfrak{n}}$ | $\mathfrak{b}_{\mathfrak{n}}$ | $\mathfrak{c}_{\mathfrak{n}}$ | $\mathfrak{o}_{\mathfrak{n}}$ | $\mathfrak{e}_{\mathfrak{n}}$ | $\mathfrak{f}_{\mathfrak{n}}$ | $p_{n}$ |

Table 4.2: General IFS code

Table 4.2 also provides a number $p_{n}$ associated with $f_{n}$ for $n=1,2,3$. These numbers are the probabilities of using the function $f_{n}$. In the more general case of the IFS $\left\{\mathbb{X}: f_{n}: n=1,2, \ldots, N\right\}$ there would be $N$ such numbers $\left\{p_{n}: n=1,2, \ldots, N\right\}$ which obey

$$
p_{1}+p_{2}+\ldots+p_{N}=1 \text { and } p_{n}>0 \text { for } n=1,2, \ldots, N .
$$

These probabilities play an important role in the computation of images of the attractor of an IFS using the Random Iteration Algorithm (Section 4.3.2). They play no role in the Deterministic Algorithm.
Other IFS codes are given in Tables 4.3 and 4.4.

|  | $n$ | $\mathfrak{a}$ | $\mathfrak{b}$ | $\mathfrak{c}$ | $\mathfrak{d}$ | $\mathfrak{e}$ | $\mathfrak{f}$ |
| ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: |
| $\square$ | 1 | $\frac{1}{2}$ | 0 | 0 | $\frac{1}{2}$ | 0 | 0 |
|  | 2 | $\frac{1}{2}$ | 0 | 0 | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 |
| $\frac{1}{3}$ |  |  |  |  |  |  |  |
| $\square$ | 3 | $\frac{1}{2}$ | 0 | 0 | $\frac{1}{2}$ | $\frac{1}{4}$ | $\frac{\sqrt{3}}{4}$ |
|  | $\frac{1}{3}$ |  |  |  |  |  |  |



Table 4.3: Another IFS code for a Sierpinski triangle

|  | $n$ | $\mathfrak{a}$ | $\mathfrak{b}$ | $\mathfrak{c}$ | $\mathfrak{d}$ | $\mathfrak{e}$ | $\mathfrak{f}$ |
| ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: |
| $\square$ | 1 | 0 | 0 | 0 | 0.16 | 0 | 0 |
| $\square$ | 2 | 0.85 | 0.04 | -0.04 | 0.85 | 0 | 1.6 |
| $\square$ | 3 | 0.2 | -0.26 | 0.23 | 0.22 | 0 | 1.6 |
| $\square$ | 4 | -0.15 | 0.28 | 0.26 | 0.24 | 0 | 0.44 |
| $\square$ |  |  |  |  |  |  |  |

Table 4.4: IFS code for a Fern

### 4.2.1 The addresses of points on fractals

We begin by considering the concept of the addresses of points on the attractor of a hyperbolic IFS. Consider the IFS of Table 4.1 whose attractor $A$, is a Sierpinski triangle with vertices at $(0,0),(0,1)$ and $(1,1)$.

We can address points on $A$ according to the sequences of transformations which lead to them, how we can see in Figure 4.1, for the firsts two steps of the Sierpinski triangle transformation.


Figure 4.1: Addresses of points for the firsts two steps of the Sierpinski triangle transformation.

There are points in $A$ which have two addresses. One example is the point that lies in the set $f_{1}(A) \cap f_{3}(A)$. The address of this point can be $311111 \ldots$ or $1333333 \ldots$, as illustrated in Figure 4.2.

On the other hand, some points on the Sierpinski triangle have only one address, such as the three vertices. The proportion of points with multiple addresses is 'small'. In such cases we say that the IFS is just-touching. If there is a unique address to every point of $A$ we say that the IFS is totally disconnected. When it appears that the proportion of points with multiple addresses is large, the IFS is overlapping.

## Continuous transformations from code space to fractals

Definition 4.2.1. Let $\left\{\mathbb{X} ; f_{1}, f_{2}, \ldots, f_{N}\right\}$ be a hyperbolic IFS. The code space associated with the IFS, $\left(\Omega, d_{|\mathcal{A}|}\right)$, is defined to be the code space on $N$ symbols


Figure 4.2: Addresses of some points of the Sierpinski triangle.
$\{1,2, \ldots, N\}$, with the metric $d_{|\mathcal{A}|}$ described in 2.3.1.
Theorem 4.2.1. Let $(\mathbb{X}, d)$ be a complete metric space. Let $\left\{\mathbb{X} ; f_{1}, f_{2}, \ldots, f_{N}\right\}$ be a hyperbolic IFS. Let A denote the attractor of the IFS and let $\left(\Omega, d_{|\mathcal{A}|}\right)$ denote the code space associated with the IFS. There exists a continuous transformation

$$
\phi: \Omega_{\{1,2, \ldots, N\}} \rightarrow A
$$

defined by

$$
\phi(\sigma)=\lim _{n \rightarrow \infty} f_{\sigma_{1} \sigma_{2} \ldots \sigma_{n}} \quad \text { for } \sigma=\sigma_{1} \sigma_{2} \ldots \sigma_{n} \in \Omega_{\{1,2, \ldots, N\}}
$$

for any $x \in \mathbb{X}$, where $f_{\sigma_{1} \sigma_{2} \ldots \sigma_{n}}(x)=f_{\sigma_{1}} \circ f_{\sigma_{2}} \circ \ldots \circ f_{\sigma_{n}}(x)$.
The function $\phi: \Omega \rightarrow A$ provided is continuous and surjective.
Definition 4.2.2. Let $\phi: \Omega \rightarrow A$ be the continuous function from code space onto the attractor of the IFS. An address of a point $x \in A$ is any member of the set

$$
\phi^{-1}(x)=\{\sigma \in \Omega: \phi(\sigma)=x\}
$$

This set is called the set of addresses of $x \in A$.
In Figure 4.2 we find examples of addresses.

### 4.3 Two algorithms for computing fractals from IFS

In this section we provide two algorithms for rendering pictures of attractors of an IFS. The algorithms presented are the Deterministic Algorithm and the Random Iteration Algorithm. Both are based in Theorem 4.1.1 and Theorem 2.2.3.

### 4.3.1 The Deterministic Algorithm

Let $\mathcal{F}=\left\{\mathbb{X} ; f_{1}, f_{2}, \ldots, f_{N}\right\}$ be a hyperbolic IFS. We choose a compact set $A_{0} \in \mathbb{R}^{2}$. Then we compute successively $A_{n}$ for $n=1,2, \ldots$ according to

$$
\begin{gathered}
A_{1}=\mathcal{F}\left(A_{0}\right)=\bigcup_{j=1}^{N} f_{j}\left(A_{0}\right) \\
A_{2}=\mathcal{F}^{2}\left(A_{0}\right)=\bigcup_{j=1}^{N} f_{j}\left(A_{1}\right) \\
\vdots \\
A_{n}=\mathcal{F}^{n}\left(A_{0}\right)=\bigcup_{j=1}^{N} f_{j}\left(A_{n-1}\right)
\end{gathered}
$$

Thus construct a sequence $\left\{A_{n}: n=0,1,2,3, \ldots\right\} \in \mathbb{H}(\mathbb{X})$. Then by Theorem 4.1.1 the sequence $\left\{A_{n}\right\}$ converges to the attractor of the IFS in the Hausdorff metric of $\mathbb{H}(\mathbb{X})$.

We have used the IFS Construction Kit ${ }^{1}$ [7] to run the Deterministic Algorithm. The algorithm takes an initial compact set $A_{0} \in \mathbb{H}(\mathbb{X})$ (the red square in Figure 4.3) and apply the function $\mathcal{F}\left(A_{n}\right)=f_{1}\left(A_{n}\right) \cup f_{2}\left(A_{n}\right) \cup \ldots \cup f_{N}\left(A_{n}\right)$ where $f_{1}, f_{2}, \ldots, f_{N}$ are the functions on the IFS (of the table 4.3 for the Sierpinski triangle). Then it plots the new set $\mathcal{F}\left(A_{0}\right)$. The next iteration plots $\mathcal{F}^{2}\left(A_{0}\right)=\mathcal{F}\left(\mathcal{F}\left(A_{0}\right)\right)$. Continued iteration produces the sequence of sets $A_{0}, \mathcal{F}\left(A_{0}\right), \mathcal{F}^{2}\left(A_{0}\right), \mathcal{F}^{3}\left(A_{0}\right) \ldots$ that converges to the attractor.

Figure 4.4 is the result of running the Deterministic Algorithm for IFS code in Table 4.4 starting from a circle as the initial array.

### 4.3.2 The Random Iteration Algorithm

The Random Iteration Algorithm is a method of creating a fractal, using a polygon and an initial point selected at random inside it. This algorithm is sometimes called the "chaos game" due to the role of the probabilities in the algorithm.

Let $\left\{\mathcal{F} ; f_{1}, f_{2}, \ldots, f_{N}\right\}$ be a hyperbolic IFS, where probability $p_{n}$ has been assigned to $f_{n}$ for $n=1,2, \ldots, N$, with

$$
\sum_{n=1}^{N} p_{n}=1
$$

Let $\Omega_{\{1,2, \ldots, N\}}$ be the code space associated with the IFS, and $\sigma=\sigma_{1}, \sigma_{2}, \ldots, \sigma_{l} \in$ $\Omega_{\{1,2, \ldots, N\}}$. Choose $x_{0} \in \mathbb{X}$ to be the initial point.

[^2]

Figure 4.3: The result of running the Deterministic Algorithm with various values of N, for the IFS code in Table 4.3, whose attractor is the Sierpinski Triangle. Shown, from left to right and top to bottom are the sets $\mathcal{F}^{n}\left(A_{0}\right)$ for $n=0,1,2, \ldots, 8$.

Then, for $l=1,2,3, \ldots$ do

$$
\begin{gathered}
x_{1}=f_{\sigma_{1}}\left(x_{0}\right) \\
x_{2}=f_{\sigma_{2}}\left(x_{1}\right) \\
\vdots \\
x_{l}=f_{\sigma_{l}}\left(x_{l-1}\right)
\end{gathered}
$$

where $\sigma_{l}$ are chosen according to the probabilities $p_{n}$.
Thus construct a sequence $\left\{x_{l}: l=0,1,2,3, \ldots\right\} \in \mathbb{X}$. Each point $x_{l}$ is a combination of the points $f_{1}(l-1), f_{2}(l-1), \ldots, f_{N}(l-1)$ with weights the probabilities $p_{n}$. The attractor A for the fractal constructed using the random iteration algorithm is

$$
\lim _{l \rightarrow \infty} f_{\sigma_{1} \sigma_{2} \ldots \sigma_{n}}\left(x_{0}\right)
$$

Random Iterated Algorithms have the advantages, when compared with deterministic iteration, of low memory requirement and high accuracy, the iterated


Figure 4.4: Fern constructed using the deterministic algorithm. The initial comptact set $A_{0}$ is a circle. Shown, from left to right and top to bottom are the sets $\mathcal{F}^{n}\left(A_{0}\right)$ for $n=0,1,2, \ldots, 8$.
point can be kept at a precision much higher than the resolution of the attractor. [2]

We illustrate the implementation of the algorithm. The following program computes and plots $n$ points on the attractor corresponding to the IFS code in Table 4.1. The program is written in Maple and it plots a fractal image constructed by the iterated function scheme discussed by Michael Barnsley in his 1993 book Fractals Everywhere [1].

## PROGRAM 4.3.1.

## restart;

fractal: $=\operatorname{proc}(\mathrm{n})$
local Mat1, Mat2, Mat3,Vector1, Vector2, Vector3, Prob1, Prob2,
Prob3,P, prob, counter, fractalplot,starttime, endtime;
Mat1: $=\operatorname{linalg}[$ matrix $]([[0.5,0.0],[0.0,0.5]])$;
Mat2: $=\operatorname{linalg}[$ matrix $]([[0.5,0.0],[0.0,0.5]])$;
Mat3: $=\operatorname{linalg}[$ matrix $]([[0.5,0.0],[0.0,0.5]]) ;$
Vector $1:=\operatorname{linalg}[$ vector $]([0,0])$;
Vector2: $=$ linalg $[$ vector $]([0,1])$;
Vector3: $=\operatorname{linalg}[$ vector $]([1,1])$;
Prob1:=1/3;


Figure 4.5: Random iteration algorithm for the Sierpinski triangle for 100 and 200 points. Black point is $x_{0}$.

```
Prob2:=1/3;
Prob3:=1/3;
P:=linalg[vector]([0,0])
writedata("fractaldata", [[P[1],P[2]]], [float,float]);
starttime:=time():
for counter from 1 to n do
    prob:=rand()/10^12;
    if prob < Prob1 then P:=evalm(Mat1&*P+Vector1)
        elif prob < Prob1+Prob2 then P:=evalm(Mat2&*P+Vector2)
            else P:=evalm(Mat3&*P+Vector3);
fi;
writedata[APPEND]("fractaldata", [[P[1],P[2]]], [float,float]);
od;fractalplot:=readdata("fractaldata",2);
print(plot(fractalplot, style=point, scaling=constrained,
axes= none, color=green, title=cat(n," iterations")));
fremove("fractaldata");
end:
```



Figure 4.6: This Sierpinski triangle is the result of running program 4.3.1 presented above for $2.000,10.000$ and 25.000 iterations respectively.

The mathematics underlying this code is the following iteration scheme. Pick a position vector in the plane and apply an affine transformation. Plot the resulting point. Apply to the new point a possibly different affine transformation, depending on the probabilities. Repeat. In the given example, there are three different affine transformations involved, and the one that is picked at a given step is randomized; each transformation has a the same probability ( $p_{n}=\frac{1}{3}$ ) of being chosen at any particular step.

The final plot is thus a set of points in the plane, and because of the randomness, a different set each time the procedure is executed. The surprise is that for a large number of iterations, the final picture always looks the same.

The result of running this program is presented in Figure 4.6. We have run
the program for $n=100,200,2.000,10.000,25.000$ points.
We can construct a Fern using a similar algorithm. In this case we need 4 transformations and we have to modify the others variables according to the Table 4.4 (IFS code for a Fern). Here the probabilities for choosing the different transformations are not the same. The following algortihm is the random iteration algorithm to draw a fern. If we run the random iteration algorithm for a Fern we obtain Figure 4.7

## PROGRAM 4.3.2.

restart;
fractal: $=\operatorname{proc}(\mathrm{n})$
local Mat1, Mat2, Mat3,Vector1, Vector2, Vector3, Prob1, Prob2,
Prob3,P, prob, counter, fractalplot,starttime, endtime;
Mat1: = linalg [matrix] ( ([0.0,0.0], [0.0,0.16]]);
Mat2: = linalg $[$ matrix $]([[0.85,0.04],[-0.04,0.85]]) ;$
Mat3: $=\operatorname{linalg}[$ matrix $]([[0.2,-0.26],[0.23,0.22]]) ;$
Mat4: $=\operatorname{linalg}[$ matrix $]([[-0.15,0.28],[0.26,0.24]]) ;$
Vector $1:=\operatorname{linalg}[$ vector $]([0,0])$;
Vector $2:=\operatorname{linalg}[$ vector $]([0,1.6])$;
Vector3: $=\operatorname{linalg}[$ vector $]([0,1.6])$;
Vector4: = linalg[vector] $([0,0.44])$;
Prob1:=0.01;
Prob2:=0.85;
Prob3:=0.07;
Prob4:=0.07;
$\mathrm{P}:=$ linalg[vector] $([0,0])$;
writedata("fractaldata", [[P[1],P[2]]], [float,float]);
starttime: $=$ time ():
for counter from 1 to n do
prob: $=$ rand ()$/ 10 \wedge 12$;
if prob $<$ Prob1 then $\mathrm{P}:=$ evalm(Mat1\&*P+Vector1)
elif prob $<$ Prob1+Prob2 then $\mathrm{P}:=$ evalm(Mat2\&*P+Vector2)
else $\mathrm{P}:=$ evalm(Mat3\&*P + Vector3);
fi;
writedata[APPEND]("fractaldata", [[P[1],P[2]]], [float,float]);
od;fractalplot:=readdata(" fractaldata", 2);
print(plot(fractalplot, style=point, scaling=constrained,
axes $=$ none, color $=$ green, title $=\operatorname{cat}(\mathrm{n}, "$ iterations")));
fremove("fractaldata");
end:


Figure 4.7: The result of running the fern random algorithm of program 3.2.2 for $2.000,10.000$ and 25.000 iterations respectively.

Probabilities play an important role in Random Iteration Algorithm. If we modify the probabilities $p_{n}$, the final attractor may vary considerably. For example, in program 4.3.2 we can change probabilites for these new ones:
Prob1:=0.25;
Prob2:=0.25;

Prob3: $=0.25$;
Prob4:=0.25;
If we run the modified random algorithm, where all probabilities are equal, we obtain the attractor of the figure 4.8.


Figure 4.8: The result of running the modified random algorithm (with equal probabilities) for 25.000 iterations.

We observe that when all probabilites are equal, the stem (as well as the central part of the fern) of the fern is wider than the stem of the fern in Figure 4.7, where the probability for this part of the fern was very small.

### 4.4 Collage theorem

When we want to find an IFS whose attractor is equal to a given compact target set $T \subset \mathbb{R}^{2}$. Sometimes we can simply spot a set of contractive transformations $f_{1}, f_{2}, \ldots, f_{N}$ taking $\mathbb{R}^{2}$ into itself, such that

$$
T=f_{1}(T) \cup f_{2}(T) \cup \ldots \cup f_{N}(T)
$$

If this equation holds, the unique solution $T$ of this equation is the attractor of the IFS $\left\{\mathbb{R}^{2} ; f_{1}, f_{2}, \ldots, f_{N}\right\}$. But in computer graphics modelling or image approximation is not always possible to find and IFS such that this equation holds. However, we may search an IFS that makes this equation approximately true. That is, we may try to make $T$ out of transformations of itself.

Michael Barnsley [1, 2] used an IFS consisting of four transformations to generate the fern that has become another "icon" of fractal geometry. He described a method for finding an IFS to generate a target image in his Collage Theorem.

According to Barnsley, the theorem tells us that to find an IFS whose attractor is "close to" or "looks like" a given set, one must find a set of transformations (contraction mappings on a suitable space within which the given set lies) such that the union, or collage, of the images of the given set under the transformations is near to the given set. Nearness is measured using the Hausdorff metric in $\mathbb{H}(\mathbb{X})$. The Collage theorem gives an upper bound to the distance between
the attractor of the resulting IFS and T.
Methods of finding and modifying these transformations to fit the set are, for instance, keyboard manipulation of numerical entries in the transformation matrices or onscreen dragging of transformation images that induces computer calculation of the corresponding matrix entries.
Iterated function systems are dense in $\mathbb{H}(\mathbb{X})$, we can approximate any fractal by an hyperbolic IFS.
Theorem 4.4.1. (The collage theorem) Let $(\mathbb{X}, d)$ be a complete metric space. Let $T \in \mathbb{H}(\mathbb{X})$ be given and let $\epsilon \geq 0$ be given. Suppose that a hyperbolic IFS $\mathcal{F}=\left\{\mathbb{X} ; f_{1}, f_{2}, \ldots, f_{N}\right\}$ of contractivity factor $0 \leq l<1$ can be found such that

$$
d_{\mathbb{H}}(T, \mathcal{F}(T)) \leq \epsilon
$$

where $d_{\mathbb{H}}$ denotes the Hausdorff metric. Then

$$
d_{\mathbb{H}}(T, A) \leq \frac{\epsilon}{1-l}
$$

where $A$ is the set attractor of the IFS.
The collage theorem is closely related to Lemma 2.1.2. In fact, is a particular case of the Lemma when $d(x, f(x))=\epsilon$.
Example 4.4.1. We will use the Collage Theorem to help find a hyperbolic IFS of the form $\left\{\mathbb{R}^{2}, f_{1}, f_{2}, f_{3}\right\}$, where $f_{1}, f_{2}$ and $f_{3}$ are transformations in $\mathbb{R}^{2}$, whose attractor is represented in Figure 4.9.


Figure 4.9: Both pictures are the same attractor. Colors will help us to solve the problem.

Solution: We can view the fractal as part of a square with vertices $(0,0)$, $(0,1),(1,0)$ and $(1,1)$. We can develope a quite simple algorithm by studying the relations between parts of the fractal. It's easy to see that there are three smaller replicas of the big picture (green, blue and orange). So, the IFS will consist of three similarity transformations with the same ratio $r=\frac{1}{2}$.
If we look at the green region, we see that we have a similarity with ratio $\frac{1}{2}$, so can define the function

$$
f_{1}(x, y)=\left(\frac{x}{2}, \frac{y}{2}\right)
$$

To define $f_{2}$ we look at the blue region. Here we have a similarity with ratio $r=\frac{1}{2}$ and a translation with vector $\left(\frac{1}{2}, 0\right)$. So, the function is

$$
f_{2}(x, y)=\left(\frac{x}{2}+\frac{1}{2}, \frac{y}{2}\right)
$$

For the orange region we have the similarity with ratio $\frac{1}{2}$, a translation with vector $\left(1, \frac{1}{2}\right)$ and a rotation of 90 degrees to the left. So we can define last function

$$
f_{3}(x, y)=\left(-\frac{y}{2}+1, \frac{x}{2}+\frac{1}{2}\right)
$$

Hence, the iterated function system we are looking for is

$$
\mathcal{F}=\left\{\mathbb{R}^{2},\left(\frac{x}{2}, \frac{y}{2}\right),\left(\frac{x}{2}+\frac{1}{2}, \frac{y}{2}\right),\left(-\frac{y}{2}+1, \frac{x}{2}+\frac{1}{2}\right)\right\}
$$

We can express this iterated function system in the Table 4.5.


Table 4.5: IFS code for example 3.1.1

Example 4.4.2. Let $T \in \mathbb{H}(\mathbb{X})$ be the picture on Figure 4.10. Let $\epsilon=\frac{1}{2 \sqrt{2}}$ be given. Let

$$
\mathcal{F}=\left\{\mathbb{R}^{2},\left(\frac{x}{2}, \frac{y}{2}\right),\left(\frac{x}{2}+\frac{1}{2}, \frac{y}{2}\right),\left(-\frac{y}{2}+1, \frac{x}{2}+\frac{1}{2}\right)\right\}
$$

be the hiperbolic IFS found, with contractivity factor $l=\frac{1}{2}$, such that $d_{\mathbb{H}}(T, \mathcal{F}(T)) \leq$ $\frac{1}{2 \sqrt{2}}$. Then

$$
d_{\mathbb{H}}(T, A) \leq \frac{\epsilon}{1-l}=\frac{1}{\sqrt{2}}
$$



Figure 4.10: We can approximate the attractor with an IFS

The collage theorem is an expression of the general principle that the attractor of a hyperbolic IFS depends continuosly on its defining parameters, such as the coefficients in an IFS code. (Theorem 4.1.2).
The flexibility and adjustability of such objects have meany applications in computer graphics, biological modelling and many other situations where we want to construct and adjust fractal models in order to appoximate given information.

## Chapter 5

## Fractal dimension and its applications

In this chapter we introduce the concept of fractal dimension. The fractal dimension of a set is a number which tells how densely the set occupies the metric space in which it lies [1]. We also present applications of the collage theorem, such as the fractal interpolation.

### 5.1 Fractal dimension

### 5.1.1 Self-similarity dimension

To yield self-similarity dimension to fractals, it is helpful to consider how segments, squares, and cubes can be tiled using the same magnification factor for each tile, such that the new objects are similar to the original.
The table below shows the continuation of this procedure. We arribe to an equation that relates an object's dimension $D$, the magnification factor $s$ and the number of tiles $N$.

| Original object | Dimension <br> $(D)$ of the <br> object | Number of tiles $(N)$ after <br> magnification factor $s=2$ | Picture |
| :---: | :--- | :--- | :--- |
| Segment | 1 | $2=2^{1}$ | $\longmapsto$ |
| Square | 2 | $4=2^{2}$ | $\square$ |
| Cube | 3 | $8=2^{3}$ |  |
| 4-cube | 4 | $16=2^{4}$ |  |
| d-cube | $d$ | $N=2^{d}$ |  |

Table 5.1: Dimension data for Euclidean d-cubes.

As we can see in Table 5.1, the equation relating the dimension $D$ of a d-cube, the number of tiles $N$ the and the magnification factor $s$ is $N=s^{D}$.

Definition 5.1.1. Given a self-similar set where $N$ is the number of tiles and $s$ the magnification factor, the self-similarity dimension of the set is given by $N=s^{D}$. Solving this equation, we can express $D$ as

$$
D=\frac{\ln N}{\ln s}
$$

The magnification factor $s$ is the inverse of the scaling factor $l$ presented in chapter 2. So, we have that $s=\frac{1}{l}$.

Example 5.1.1. The self-similarity dimension of the Sierpinski Triangle can be found using the equation $D=\frac{\ln N}{\ln s}$. In this case, the scaling factor is $l=\frac{1}{2}$ so, the magnification factor is $s=2$. The number of new tiles is $N=3$. So we have that the self-similarity dimension of this fractal is

$$
D=\frac{\ln 3}{\ln 2} \approx 1.58496
$$



Figure 5.1: The self-similarity dimension of the Sierpinski triangle is $D=\frac{\ln 3}{\ln 2}$.

Example 5.1.2. In the case of the Koch curve, the magnification factor is $s=3$ (because the scaling factor is $l=\frac{1}{3}$ ) and the number of tiles congruent to the original after the interation is $N=4$. So, using the equation $D=\frac{\ln N}{\ln s}$, we have that the self-similarity dimension for the koch curve is

$$
D=\frac{\ln 4}{\ln 3} \approx 1.26185
$$



Figure 5.2: The self-similarity dimension of the koch curve is $D=\frac{\ln 4}{\ln 3}$

### 5.1.2 Box dimension

The self-similarity dimension applies only to sets that are strictly self-similar. So, we need to define a more generalized dimension that can be applied to sets that are only "approximately" self-similar, including natural fractals like coastlines. This generalized dimension is called box dimension.

Definition 5.1.2. Let $A \in \mathbb{H}(\mathbb{X})$ where $(\mathbb{X}, d)$ is a metric space. For each $\epsilon>0$, let $N(A, \epsilon)$ denote the smallest number of closed balls of radius $\epsilon>0$ needed to cover A.
If

$$
D=\lim _{\epsilon \rightarrow 0}\left\{\frac{\ln (N(A, \epsilon))}{\ln \left(\frac{1}{\epsilon}\right)}\right\} \quad \text { exists }
$$

then $D$ is called the fractal dimension of A.
We will say that "A has fractal dimension $D$ ".
The following theoremes simplify the process of calculating the fractal dimension, because it allows us to change the variable $\epsilon$ for a discrete variable.

Theorem 5.1.1 (The Box counting theorem). Let $A \in \mathbb{H}(\mathbb{X})$ where $(\mathbb{X}, d)$ is a metric space. Let $\epsilon_{n}=C r^{n}$ for real numbers $0<r<1, C>0$ and $n=1,2,3, \ldots$.
If

$$
D=\lim _{n \rightarrow \infty}\left\{\frac{\ln \left(N\left(A, \epsilon_{n}\right)\right)}{\ln \left(\frac{1}{\epsilon_{n}}\right)}\right\} \quad \text { exists }
$$

then $A$ has fractal dimension $D$.
Next theorem is a very used particular case of the Box counting theorem when $r=\frac{1}{2}$.
Theorem 5.1.2. Let $A \in \mathbb{H}(\mathbb{X})$ where the Euclidean metric is used. Cover $\mathbb{R}^{n}$ by closed just-touching square boxes of side length $\left(\frac{1}{2^{n}}\right)$. Let $N_{n}(A)$ denote the number of boxes of side length $\left(\frac{1}{2^{n}}\right)$ which intersect the attractor. If

$$
D=\lim _{n \rightarrow \infty}\left\{\frac{\ln \left(N_{n}(A)\right)}{\ln \left(2^{n}\right)}\right\} \quad \text { exists, }
$$

then $A$ has fractal dimension $D$.
Example 5.1.3. Consider the attractor $A$ of the Sierpinski triangle, in Figure 5.3 , as a subset of $\left(\mathbb{R}^{2}\right.$, Euclidean $)$.


Figure 5.3: Sierpiski triangle.
We see that

$$
\begin{gathered}
N_{1}(A)=3 \\
N_{2}(A)=9 \\
N_{3}(A)=27 \\
N_{4}(A)=81
\end{gathered}
$$

$$
N_{n}(A)=3^{n} \quad \text { for } \quad n=1,2,3, \ldots
$$

So, by the particular case of the Box counting theorem, we have that the fractal dimension of the attractor of Sierpinski triangle is

$$
D(T)=\lim _{n \rightarrow \infty}\left\{\frac{\ln \left(N_{n}(A)\right)}{\ln \left(2^{n}\right)}\right\}=\lim _{n \rightarrow \infty}\left\{\frac{\ln \left(3^{n}\right)}{\ln \left(2^{n}\right)}\right\}=\frac{\ln 3}{\ln 2}
$$

Example 5.1.4. Let $A \in(\mathbb{H}(\mathbb{X})$ be the attractor of the koch curve in Figure 5.4. In this case we have that $\epsilon_{n}=\left(\frac{1}{3}\right)^{n}$, for $n=1,2,3, \ldots$. The number of boxes of side lenght $\left(\frac{1}{2^{n}}\right)$ are:

$$
\begin{gathered}
N_{1}(A)=4 \\
N_{2}(A)=16 \\
N_{3}(A)=64 \\
\vdots \\
N_{n}(A)=4^{n} \text { for } n=1,2,3, \ldots
\end{gathered}
$$

So, using the Box Couting theorem, we have that A has fractal dimension

$$
D(T)=\lim _{n \rightarrow \infty}\left\{\frac{\ln \left(N_{n}(A)\right)}{\ln \left(3^{n}\right)}\right\}=\lim _{n \rightarrow \infty}\left\{\frac{\ln \left(4^{n}\right)}{\ln \left(3^{n}\right)}\right\}=\frac{\ln 4}{\ln 3}
$$



Figure 5.4: Koch curve

One can see in Examples 5.1.3 and 5.1.4, that the fact fractal $A$ has fractional dimension $D$ means that the "density" of the fractal in the plane is bigger than the Euclidean dimension 1 and smaller than the Euclidean dimension 2.

### 5.2 Space-filling curves

We say that a space-filling curve is a curve whose range contains the entire 2-dimensional unit square. Space-filling curves are special cases of fractal constructions, which fractal dimension is 2 .

Space-filling curves in the 2-dimensional plane are commonly called Peano curves, because Giuseppe Peano was the first to discover one. Peano discovered a dense curve that passes through every point of the unit square. His purpose was to construct a continuous mapping from the unit interval onto the unit square.

Example 5.2.1. To construct an example of a Peano curve, we start with a square of vertices $(0,0),(0,1),(1,0)$ and $(1,1)$.
Stage 0 in this procediment is the function $y=x$, as we can see in Figure 5.5.


Figure 5.5: Stage 0 in the construction of the Peano curve
In stage 1 (Figure 5.6) we draw the following functions:

$$
y=x ; y=x-\frac{2}{3} ; y=x+\frac{2}{3}
$$

and

$$
y=-x+\frac{2}{3} ; y=-x+\frac{4}{3}
$$



Figure 5.6: Stage 1 of the contruction of the Peano Curve
In general, in the n-stage, we draw the following functions:

$$
y_{k}=x+p_{k} \quad \text { where } p_{k}=\frac{-3^{n}+1+2 k}{3^{n}} \text { with } k=0, \ldots, 3^{n}-1
$$

and

$$
y_{l}=-x+p_{l} \quad \text { where } p_{l}=\frac{2 j}{3^{n}} \text { with } j=1, \ldots, 3^{n}-1
$$

We have run the following program with Maple. We have plotted (for $n=$ $0,1,2,3,4$.) each stage of the construction of the Peano curve.

## PROGRAM 5.2.1.

restart;
$\mathrm{n}:=1 ; \%$ Change this $n$ to draw the diferent iterations.
$\mathrm{A}:=\operatorname{seq}\left(x+\left(-1 * 3^{n}+1+2 * i\right) / 3^{n}, i=0 . .3^{n}-1\right)$ :
$\mathrm{B}:=\operatorname{seq}\left(-x+2 * i / 3^{n}, i=1 . .3^{n}-1\right)$ :
$\operatorname{plot}([\mathrm{A}, \mathrm{B}], \mathrm{x}=0 . .1, \mathrm{y}=0 . .1)$;


Figure 5.7: Result of running the Program 5.2 .1 for $n=2,3,4$ respectively. (Stages 2, 3 and 4 of the Peano curve)

The self-similarity dimension of Peano curve is

$$
D=\frac{\ln N}{\ln l}=\frac{\ln 9}{\ln 3}=2
$$

where $N=9$ are the number of new tiles and $l=\frac{1}{3}$ the scaling factor. The dimension is 2, so it leads to a filled square, and the curve is a space-filling curve.

### 5.3 Fractal interpolation

In this section we introduce fractal interpolation functions. Using this new technology one can make complicated curves. It is shown how geometrically complex graphs of continuous functions can be constructed to pass through specified data points. The graphs of these functions can be used to approximate image components such as the profiles of mountains, the tops of clouds, horizons over forets, tumours, etc. Fractal interpolation is a consequence of the Collage theorem and theorem 4.1.2.

Definition 5.3.1. A set of data is a set of points of the form $\left\{\left(x_{i}, F_{i}\right) \in \mathbb{R}^{2}\right.$ : $i=0,1,2, \ldots, N\}$ where

$$
x_{0}<x_{1}<x_{2}<\cdots<x_{N}
$$

An interpolation function corresponding to this set of data is a continuous function $f:\left[x_{0}, x_{N}\right] \rightarrow \mathbb{R}$ such that

$$
f\left(x_{i}\right)=F_{i} \text { for } i=0,1,2, \ldots, N .
$$

The points $\left(x_{i}, F_{i}\right) \in \mathbb{R}^{2}$ are called the interpolation points. We say that he function $f$ interpolates the data; and that $f$ passes through the interpolation points.

Let a set of data $\left\{\left(x_{i}, F_{i}\right): i=0,1,2, \ldots, N\right\}$ be given. We explain how to construct an IFS such that its attractor is the graph of continuous interpolation function $f:\left[x_{0}, x_{N}\right] \rightarrow \mathbb{R}$ which interpolates the data. We consider an IFS of the form $\mathbb{R}^{2} ; f_{n}, n=1,2, \ldots, N$, where the contractive mappings have this special form

$$
f_{n}\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right)=\left(\begin{array}{ccc}
\mathfrak{a}_{\mathfrak{n}} & 0 & \mathfrak{e}_{\mathfrak{n}} \\
\mathfrak{c}_{\mathfrak{n}} & \mathfrak{d}_{\mathfrak{n}} & \mathfrak{f}_{\mathfrak{n}} \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
x \\
y \\
1
\end{array}\right) \text { for } n=1,2, \ldots, N .
$$

The transformations are constrained by data according to

$$
f_{n}\left(\begin{array}{c}
x_{0} \\
F_{0} \\
1
\end{array}\right)=\left(\begin{array}{c}
x_{n-1} \\
F_{n-1} \\
1
\end{array}\right) \quad \text { and } \quad f_{n}\left(\begin{array}{c}
x_{N} \\
F_{N} \\
1
\end{array}\right)=\left(\begin{array}{c}
x_{n} \\
F_{n} \\
1
\end{array}\right) \quad \text { for } \quad n=1,2, \ldots, N
$$

Let $n \in\{1,2,3, \ldots, N\}$. The transformations $f_{n}$ depends of five numbers $\mathfrak{a}_{\mathfrak{n}}, \mathfrak{c}_{\mathfrak{n}}, \mathfrak{o}_{\mathfrak{n}}, \mathfrak{e}_{\mathfrak{n}}$ and $\mathfrak{f}_{\mathfrak{n}}$, which obey the four linear equations

$$
\begin{gathered}
\mathfrak{a}_{\mathfrak{n}} x_{0}+\mathfrak{e}_{\mathfrak{n}}=x_{n-1} \\
\mathfrak{a}_{\mathfrak{n}} x_{N}+\mathfrak{e}_{\mathfrak{n}}=x_{n} \\
\mathfrak{c}_{\mathfrak{n}} x_{0}+\mathfrak{o}_{\mathfrak{n}} F_{0}+\mathfrak{f}_{\mathfrak{n}}=F_{n-1} \\
\mathfrak{c}_{\mathfrak{n}} x_{N}+\mathfrak{d}_{\mathfrak{n}} F_{N}+\mathfrak{f}_{\mathfrak{n}}=F_{n}
\end{gathered}
$$

So, if we have 5 parameters and 4 linear equations, it follows that there is one free parameter. We choose this free parameter to be $\mathfrak{d}_{\mathfrak{n}}$. We call $\mathfrak{d}_{\mathfrak{n}}$ the vertical scaling factor in the transformation $f_{n}$. If $\mathfrak{d}_{\mathfrak{n}}$ is any real number, we can solve the above equations in terms of $\mathfrak{d}_{\mathfrak{n}}$. The solutions are

$$
\begin{gathered}
\mathfrak{a}_{\mathfrak{n}}=\frac{\left(x_{n}-x_{n-1}\right)}{\left(x_{N}-x_{0}\right)} \\
\mathfrak{c}_{\mathfrak{n}}=\frac{\left(F_{n}-F_{n-1}\right)}{\left(x_{N}-x_{0}\right)}-\frac{\mathfrak{o}_{\mathfrak{n}}\left(F_{N}-F_{0}\right)}{\left(x_{N}-x_{0}\right)} \\
\mathfrak{e}_{\mathfrak{n}}=\frac{\left(x_{N} x_{n-1}-x_{0} x_{n}\right)}{\left(x_{N}-x_{0}\right)} \\
\mathfrak{f}_{\mathfrak{n}}=\frac{\left(x_{N} F_{n-1}-x_{0} F_{n}\right)}{\left(x_{N}-x_{0}\right)}-\frac{\mathfrak{o}_{\mathfrak{n}}\left(x_{N} F_{0}-x_{0} F_{N}\right)}{\left(x_{N}-x_{0}\right)}
\end{gathered}
$$

So, we have defined an IFS $\left\{\mathbb{R}^{2} ; f_{n}, n=1,2, \ldots, N\right\}$ which attractor is a continuous interpolation function where $\mathfrak{d}_{\mathfrak{n}}$ is the vertical scaling factor. In the following subsection we will see that modifing this free parameter we can approximate the fractal dimension of the interpolation function.

Program 5.3.1, written in Maple, computes the solutions above to find the parameters of the transformations of an IFS, for the particular case when $N=3$. There are four interpolation points, that can be modified. The scaling factors can also be changed.

## PROGRAM 5.3.1.

## restart;

$\mathrm{x} 0:=0: \mathrm{F} 0:=0: \%$ you can change the interpolation points
$\mathrm{x} 1:=30: \mathrm{F} 1:=40$ :
$\mathrm{x} 2:=60:$ F2: $=30$ :
$\mathrm{x} 3:=100:$ F3: $=50$ :
$\mathrm{d} 1:=0.3: \mathrm{d} 2:=0.3: \mathrm{d} 3:=0.3: \%$ scaling factors can be changed
$\mathrm{b}:=0$ :
$\mathrm{a} 1:=(\mathrm{x} 1-\mathrm{x} 0) /(\mathrm{x} 3-\mathrm{x} 0)$ :
$\mathrm{c} 1:=(\mathrm{F} 1-\mathrm{F} 0) /(\mathrm{x} 3-\mathrm{x} 0)-\mathrm{d} 1 *(\mathrm{~F} 3-\mathrm{F} 0) /(\mathrm{x} 3-\mathrm{x} 0):$
$\mathrm{e} 1:=\left(\mathrm{x} 3^{*} \mathrm{x} 0-\mathrm{x} 0{ }^{*} \mathrm{x} 1\right) /(\mathrm{x} 3-\mathrm{x} 0)$ :
$\mathrm{f} 1:=\left(\mathrm{x} 3^{*} \mathrm{~F} 0-\mathrm{x} 0^{*} \mathrm{~F} 1\right) /(\mathrm{x} 3-\mathrm{x} 0)-\mathrm{d} 1^{*}\left(\mathrm{x} 3^{*} \mathrm{~F} 0-\mathrm{x} 0 * \mathrm{~F} 3\right) /(\mathrm{x} 3-\mathrm{x} 0)$ :
$\mathrm{a} 2:=(\mathrm{x} 2-\mathrm{x} 1) /(\mathrm{x} 3-\mathrm{x} 0)$ :
$\mathrm{c} 2:=(\mathrm{F} 2-\mathrm{F} 1) /(\mathrm{x} 3-\mathrm{x} 0)-\mathrm{d} 2^{*}(\mathrm{~F} 3-\mathrm{F} 0) /(\mathrm{x} 3-\mathrm{x} 0):$

```
e2:=(x3*x1-x0*x2)/(x3-x0):
f2:=(x3*F1-x0*F2)/(x3-x0)-d2*(x3*F0-x0*F3)/(x3-x0):
a3:=(x3-x2)/(x3-x0):
c3:=(F3-F2)/(x3-x0)-d2*(F3-F0)/(x3-x0):
e3:=(x3*x2-x0*x3)/(x3-x0):
f3:=(x3*F2-x0*F3)/(x3-x0)-d2*(x3*F0-x0*F3)/(x3-x0):
a1; b; c1; d1; e1; f1;
a2; b; c2; d2; e2; f2;
a3; b; c3; d3; e3; f3;
```

Example 5.3.1. We have used program 5.3.1 to find the IFS which attractor interpolates the following data set

$$
\{(0,0),(30,40),(60,30),(100,50)\}
$$

The program gives us the parameters of the IFS. This parameters are shown in the following table

|  | $n$ | $\mathfrak{a}$ | $\mathfrak{b}$ | c | $\mathfrak{d}$ | $\mathfrak{e}$ | f | $p$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\square$ | 1 | $\frac{3}{10}$ | 0 | $\frac{1}{4}$ | $\frac{3}{10}$ | 0 | 0 | $\frac{1}{3}$ |
| $\square$ | 2 | $\frac{3}{10}$ | 0 | $-\frac{1}{4}$ | $\frac{3}{10}$ | 30 | 40 | $\frac{1}{3}$ |
| $\square$ | 3 | $\frac{2}{5}$ | 0 | 0.05 | $\frac{3}{10}$ | 60 | 30 | $\frac{1}{3}$ |

Table 5.2: IFS code for an interpolation function
The attractor of the IFS code of Table 5.2 is shown in Figure 5.8. The IFS has a unique attractor which is the graph of a function which passes through the interpolation points $\{(0,0),(30,40),(60,30),(100,50)\}$. In this case, we have chosen $\mathfrak{d}_{\mathfrak{n}}=0.3$ for $n=1,2,3$.


Figure 5.8: Graph of the interpolation function.

Hyperbolic IFS with attractor are a way of finding interpolation functions. In the next two theorems we first define an IFS from the points $\left(x_{i}, F_{i}\right)$ and secondly we see that the attractor of the IFS denotes the required interpolation function.

Theorem 5.3.1. Let $N$ be a positive integer greater thant one. Let $\mathbb{R}^{2} ; f_{n}, n=$ $1,2, \ldots, N$ denote the IFS associated with the data set $\left\{\left(x_{i}, F_{i}\right): i=0,1, \ldots, N\right\}$.

Let the vertical scaling factor $\mathfrak{d}_{\mathfrak{n}}$ obey $0 \leq \mathfrak{d}_{\mathfrak{n}}<1$ for $n=1,2, \ldots, N$. Then there is a metric $d$ on $\mathbb{R}^{2}$, equivalent to the Euclidean metric, such that the IFS is hyperbolic with respect to $d$. In particular, there is a unique nonempty compact set

$$
G=\bigcup_{n=1}^{N} f_{n}(G)
$$

Theorem 5.3.2. Let $N$ be a positive integer greater than one. Let Let $\mathbb{R}^{2} ; f_{n}, n=$ $1,2, \ldots, N$ denote the IFS associated with the data set $\left\{\left(x_{i}, F_{i}\right): i=0,1, \ldots, N\right\}$. Let the vertical scaling factor $\mathfrak{d}_{\mathfrak{n}}$ obey $0 \leq \mathfrak{d}_{\mathfrak{n}}<1$ for $n=1,2, \ldots, N$. Let $G$ denote the attractor of the IFS. Then $G$ is the graph of a continuous function $f:\left[x_{0}, x_{N}\right] \rightarrow \mathbb{R}$ which interpolates the data $\left\{\left(x_{i}, F_{i}\right): i=0,1, \ldots, N\right\}$, that is

$$
G=\left\{(x, f(x)): x \in\left[x_{0}, x_{N}\right]\right\}
$$

where

$$
f\left(x_{i}\right)=F_{i} \text { for } i=0,1,2, \ldots, N .
$$

Definition 5.3.2. The function $f(x)$ whose graph is the attractor of an IFS as described in the above theorems, is called a fractal interpolation function corresponding to the data $\left\{\left(x_{i}, F_{i}\right): i=0,1, \ldots, N\right\}$.

The theory of hyperbolic IFS is applied to fractal interpolation functions. We can use IFS algorithms of chapter 4 to compute fractal interpolation functions. The Collage theorem is used to find fractal interpolations functions which approximate the given data.

### 5.3.1 The fractal dimension of interpolation functions

The following theorem tells us the fractal dimension of fractal interpolation functions.

Theorem 5.3.3. Let $N$ be a positive integer greater than one. Let $\left\{\left(x_{i}, F_{i}\right): i=\right.$ $0,1, \ldots, N\}$ be a set of data. Let $\left\{\mathbb{R}^{2} ; f_{n}, n=1,2, \ldots, N\right\}$ be an IFS associated with the data. Let $G$ denote the attractor of the IFS, so that $G$ is the graph of a fractal interpolation function associated with the data. If

$$
\sum_{n=1}^{N}\left|\mathfrak{o}_{\mathfrak{n}}\right|>1
$$

and the interpolation points do not all lie on a single straight line, the the fractal dimension of $G$ is the unique real solution $D$ of

$$
\sum_{n=1}^{N}\left|\mathfrak{o}_{\mathfrak{n}}\right| \mathfrak{a}_{\mathfrak{n}}{ }^{D-1}=1
$$

Otherwise, the fractal dimension of $G$ is one.
Example 5.3.2. Consider the set of data $\{(0,0),(1,1),(2,1),(3,2)\}$. In this case, the interpolation points are equally spaced. It follows that $\mathfrak{a}_{\mathfrak{n}}=\frac{1}{N}$, where
$N=3$. Hence, if condition (1) in Theorem 5.3.3 holds, then the fractal dimension $D$ of the interpolation function is

$$
\sum_{n=1}^{N}\left|\mathfrak{o}_{\mathfrak{n}}\right| \mathfrak{a}_{\mathfrak{n}}^{D-1}=\sum_{n=1}^{N}\left|\mathfrak{d}_{\mathfrak{n}}\right|\left(\frac{1}{3}\right)^{D-1}=1
$$

If we isolate the summation we have

$$
\sum_{n=1}^{3}\left|\mathfrak{o}_{\mathfrak{n}}\right|=3^{D-1}
$$

If we then apply logaritms and solve the equation we get that

$$
D=1+\frac{\log \left(\sum_{n=1}^{3}\left|\mathfrak{o}_{\mathfrak{n}}\right|\right)}{\log 3}
$$

We observe that varying the scaling factor $\mathfrak{d}_{\mathfrak{n}}$ for every $n=1,2, \ldots, N$ we can approximate the fractal dimension of the interpolation function.

We have run the program 5.3.1 with our interpolation points, to get the IFS code for diferent values of the scaling factor $\mathfrak{d}_{\mathfrak{n}}$. So, we have fractal interpolation functions corresponding to the set of data $\{(0,0),(1,1),(2,1),(3,2)\}$ with diferent fractal dimension. Figure 5.9 shows these fractal interpolation functions.


Figure 5.9: Members of the family of fractal interpolation functions corresponding to the set of data $\{(0,0),(1,1),(2,1),(3,2)\}$, such that each function has diferent dimension.

We have varied the scaling factor $\mathfrak{d}_{\mathfrak{n}}$ of each transformation to obtain the different fractal interpolation functions in Figure 5.9.

1. For the first fractal interpolation function (top left) we have that $\mathfrak{d}_{1}=0$, $\mathfrak{d}_{2}=0$ and $\mathfrak{d}_{3}=0$, so the fractal dimension is $D=1$.
2. For the second function (top right) $\mathfrak{d}_{1}=\frac{1}{3}, \mathfrak{d}_{2}=\frac{1}{3}$ and $\mathfrak{d}_{3}=\frac{1}{3}$, so the fractal dimension is also $D=1$.
3. For the third interpolation function (bottom left), we have chosen $\mathfrak{d}_{1}=0.4$, $\mathfrak{d}_{2}=0.4$ and $\mathfrak{D}_{3}=0.4$, so the fractal dimension is $D \approx 1.1659$.
4. For our last interpolation function (bottom right), we have that $\mathfrak{d}_{1}=0.5$, $\mathfrak{d}_{2}=0.5$ and $\mathfrak{d}_{3}=0.5$, so in this case the fractal dimension of the function is $D \approx 1.369$.

Observation: In this four examples we have chosen $\mathfrak{d}_{1}=\mathfrak{d}_{2}=\mathfrak{d}_{3}$, but this is not necessary. Each $\mathfrak{d}_{\mathfrak{n}}$ can be diferent for $n=1,2,3$.

### 5.4 Applications of fractal dimension

### 5.4.1 Fractals in Stock Market

Benoit Mandelbrot, a mathematician known as the father of fractal geometry, began to apply his knowledge of fractals to explain stock markets [5].

Looking to stock markets, we see that they have turbulence. Some days the change in markets is very small, and some days it moves in a huge leap. Only fractals can discribe this kind of random change.

Economists in the 1970s and 1980s proposed Gaussian models to analyze the market behaviour. But Mandelbrot explained that there are far more market bubbles and market crashes than these models suggested. He argues that fractal techniques may provide a more powerful way to analyse risk, and how fractal techniques might be applied to financial data to provide better estimation for risk and volatility.

The FTSE 100 Index is a share index of the 100 most highly capitalised UK companies listed on the London Stock Exchange. Figure $5.10{ }^{1}$ shows the changing volatility of FTSE 100 Index, as the magnitude of price varied wildly during the day. This chart is from Monday, 6th of June 2011.

It is possible to use fractal notions to study and build models of how stock markets work. For example, computing the fractal dimension of data charts it is possible to know if it has been a day of high turbulence and volatility. Fractal dimension of data chart usually varies from 1.15 to 1.4. If the fractal dimension of a data chart is high, this means that this day (or period) has high turbulences and volatility. On that chart we will observe the line jumping up and down.

Unlike the fractals we have seen so far, a stock market is not an exact selfsimilar geometric object.

[^3]

Figure 5.10: FTSE 100 chart of Monday, June 62011.

### 5.4.2 Fractals in nature

Approximate fractals are easily found in nature. These objects display selfsimilar structure over an extended, but finite, scale range. Examples include clouds, mountain, snow flakes, broccoli or systems of blood vessels and pulmonary vessels. Ferns are fractal in nature and can be modeled on a computer by using a recursive algorithm, as we have done in chapter 4 .

## Coastlines

Even coastlines may be loosely considered fractal in nature. We are going to shown the example of the coast of Norway. Norway comprises the western part of Scandinavia in Northern Europe. It has a rugged coastline, broken by huge fjords and thousands of islands.


Figure 5.11: Part of the Norway's coastline.
Figure $5.11^{2}$ shows part of the coast of Norway. The rougher an object is, the higher its fractal dimension. The coast of Norway is very rough, so using the

[^4]box dimension, we get that the fractal dimension of the coast is approximately $D=1.52$. Coastlines are not an exact self-similar geometric object.

## Clouds

Clouds look very irregular in shape, indeed, clouds are fractal in shape just like most other objects in nature.
We have used Collage theorem to find an IFS whose attractor looks like a cloud, varying the parameters of the IFS to get what we want. Figure 5.12 is the result of iterating this IFS with three transformation functions. This fractal has been plotted using a similar technique that the one in Section 5.3 but for overlapping IFS.


Figure 5.12: We have used the Collage theorem to construct a fractal that looks like a cloud.

Although this can be a method to approximate fractals, normally self-similar fractals are too regular to be realistic. To make fractals more realistic, we use a different type of self-similarity called Brownian self-similar. In the Brownian self-similarity, although each line is composed of smaller lines (as in the selfsimilarty that we knew), the lines are random instead of being fixed.
Brownian self-similarity is found in plasma fractals. Plasma fractals are very useful in creating realistic landscapes and fractals in nature. Unlike most other fractals, they have a random element in them, which gives them Brownian selfsimilarity. Due to their randomness, plasma fractals closely resemble nature. Because of this, we have used them to make plasma fractals look like clouds, as we can see in Figure 5.13.


Figure 5.13: Clouds generated using plasma fractal method compared with real clouds of Linköping.

We have used a free software called XFractint to draw this plasma fractal. We can control how fragmented the clouds are by changing parameters.

## Chapter 6

## Fractal tops

One application of tops is to modelling new families of synthetic pictures in computer graphics by composing fractals. A top is "the inverse" of an adressing function for a fractal (attractor of an IFS) that is surjective but, in general, not bijective. With fractals tops we address each point in the attractor of a IFS in a unique way.

### 6.1 Fractal tops

We recall from chapter 4 how to get fractals as the attractor A of an IFS. Let an hyperbolic iterated function system (IFS) be denoted

$$
\mathcal{F}:=\left\{\mathbb{X}, f_{1}, f_{2}, \ldots, f_{N}\right\}
$$

As we have seen this consists of a finite sequence of one-to-one contractive mappings

$$
f_{n}: \mathbb{X} \rightarrow \mathbb{X}, n=0,1,2, \ldots, N-1
$$

acting on the compact metric space $(\mathbb{X}, d)$ with metric $d$. So that for some $0 \leq l<1$ we have $d\left(f_{n}(x), f_{n}(y)\right) \leq l \cdot d(x, y)$ for all $x, y \in \mathbb{X}$.

Let $A$ denote the attractor of the IFS, that is $A \subset X$ is the unique non-empty compact set such that

$$
A=\bigcup_{n} f_{n}(A) \text { given by theorem 4.1.1 }
$$

Let the associated code space be denoted by $\Omega=\Omega_{\{1,2, \ldots, N\}}$. We have seen in chapter 4.2.1 that there exist a continuous transformation

$$
\phi: \Omega_{\{1,2, \ldots, N\}} \rightarrow A
$$

from the code space $\Omega_{\{1,2, \ldots, N\}}$ onto the set attractor $A$ of the hyperboloic IFS $\mathcal{F}:=\left\{\mathbb{X}, f_{1}, f_{2}, \ldots, f_{N}\right\}$.
This transformation is defined by

$$
\phi(\sigma)=\lim _{n \rightarrow \infty} f_{\sigma_{1} \sigma_{2} \ldots \sigma_{n}} \quad \text { for } \sigma=\sigma_{1} \sigma_{2} \ldots \sigma_{n} \in \Omega_{\{1,2, \ldots, N\}}
$$

for any $x \in \mathbb{X}$, where $f_{\sigma_{1} \sigma_{2} \ldots \sigma_{n}}(x)=f_{\sigma_{1}} \circ f_{\sigma_{2}} \circ \ldots \circ f_{\sigma_{n}}(x)$.

Now notice that the set of addresses of a point $x \in A$, defined to be $\phi^{-1}(x)$, is both bounded above and closed, hence it is compact. So it must posses a unique largest element (the top). We denote this element by $\tau(x)$.

Definition 6.1.1. Let $\mathcal{F}$ be a hyperbolic IFS with set attractor $A$ and code space function $\phi: \Omega \rightarrow A$. Then the tops function of $\mathcal{F}$ is $\tau: A \rightarrow \Omega$, defined by

$$
\tau(x)=\max \{\sigma \in \Omega: \phi(\sigma)=x\}
$$

The set of points $G_{\tau}:=\{(x, \tau(x)): x \in A\}$ is called the graph of the top of the IFS or simply the fractal top of the IFS.

### 6.2 Pictures of tops: colour-stealing

Here we introduce the application of tops to computer graphics. The basic idea of the colour-stealing is that we start with two iterated function systems (IFSs) and an input image. Then we run the random iteration algorithm, applying the same random choice to each IFS simultaneously. One of the IFSs produces a sequence of points that lie on the input image. The other IFS produces a sequence of points that are coloured according to the colour value of the other IFS, read off from the input picture.

Here we are interested in picture functions of the form $\mathfrak{B}: D_{\mathfrak{B}} \subset \mathbb{R}^{2} \rightarrow \mathcal{C}$, where $\mathcal{C}$ is a colour space, for example $\mathcal{C}=[0,255]^{3} \subset \mathbb{R}^{3}$. For colour-stealing applications we may choose

$$
D_{\mathfrak{B}}=\square:=\left\{(x, y) \in \mathbb{R}^{2}: 0 \leq x, y \leq 1\right\}
$$

Let two hyperbolic IFSs be

$$
\mathcal{F}_{D}:=\left\{\square ; f_{1}, f_{2}, \ldots, f_{N}\right\} \quad \text { and } \quad \mathcal{F}_{C}:=\left\{\square ; \tilde{f}_{1}, \tilde{f}_{2}, \ldots, \tilde{f_{N}}\right\}
$$

The index ' $D$ ' stands for 'drawing' and the index ${ }^{\prime} C^{\prime}$ for colouring.
Let a picture function

$$
\mathfrak{B}_{C}:=\square \rightarrow \mathcal{C}
$$

be given. Let $A$ denote the attractor of the IFS $\mathcal{F}_{D}$ and let $\tilde{A}$ denote the attractor of the IFS $\mathcal{F}_{C}$.
Let

$$
\tau_{D}: A \rightarrow \Omega
$$

denote the tops function for $\mathcal{F}_{D}$ and

$$
\phi_{C}: \Omega \rightarrow \tilde{A}
$$

denote the addressing function for $\mathcal{F}_{C}$. Then we define a new picture function $\mathfrak{B}_{D}: A \rightarrow \mathfrak{C}$ by

$$
\mathfrak{B}_{D}=\mathfrak{B}_{C} \circ \phi_{C} \circ \tau_{D}
$$

This is the unique picture function defined by the IFS's $\mathcal{F}_{D}, \mathcal{F}_{C}$ and the picture $\mathfrak{B}_{C}$. We say that $\mathfrak{B}_{D}$ has been produced by tops plus colour-stealing.

An example of the colour-stealing method is shown in Figure 6.1. This fractal
has been drawn using IFS Construction Kit. The method used by the software is based on the idea explained above.

We choose a colorful input image and a coloring IFS $\left(\mathcal{F}_{C}\right)$ with the same number of functions as the drawing $\operatorname{IFS}\left(\mathcal{F}_{D}\right)$ being used to generate the fractal. Each time a random function from the drawing IFS is chosen to plot the next point in that iteration, the corresponding function in the coloring IFS is used to plot the next point for that IFS. The coloring IFS is drawn on top of the input image. The point computed for the drawing IFS is plotted with the same color as the point determined by the coloring IFS. The drawing IFS "steals" the color from the image underneath the coloring IFS.

The fractal image on Figure 6.1 is a fractal fern colored based on the image of purple flowers. In this case $\mathcal{F}_{\mathcal{D}}$ and $\mathcal{F}_{\mathcal{C}}$ are based in the same IFS code in Table 4.4, whose attractor is a fractal fern.


Figure 6.1: Fractal top produced by colour-stealing. The colours were 'stolen' from the picture on the right.

On Figure 6.2 we have construct a new fractal, using the IFS code in Table 4.4 for the $\mathcal{F}_{\mathcal{D}}$ and a Sierpinski triangle to colour the image. As we need the same number of transformations in each IFS, we have added a new transformation with probability 0 in the IFS of Table 4.3, so that the final attractor $A$ remains the Sierpinski triangle, but now we are able to use the colour-stealing method. So, for the coloring IFS $\mathcal{F}_{\mathcal{C}}$ we have used the IFS code in Table 6.1.
Thus, the final fractal is a fern colored using a Sierpinski triangle that 'stoles' colours from the initial picture.

|  | $n$ | $\mathfrak{a}$ | $\mathfrak{b}$ | $\mathfrak{c}$ | $\mathfrak{d}$ | $\mathfrak{e}$ | $\mathfrak{f}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\square$ | 1 | $\frac{1}{2}$ | 0 | 0 | $\frac{1}{2}$ | 0 | 0 |
| $\square$ | 2 | $\frac{1}{2}$ | 0 | 0 | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 |
| $\frac{1}{3}$ |  |  |  |  |  |  |  |
| $\square$ | 3 | $\frac{1}{2}$ | 0 | 0 | $\frac{1}{2}$ | $\frac{1}{4}$ | $\frac{\sqrt{3}}{4}$ |
| $\square$ | 4 | $\frac{1}{2}$ | 0 | 0 | $\frac{1}{2}$ | 0 | 0 |



Table 6.1: Another IFS code for a Sierpinski triangle


Figure 6.2: Fractal top produced by colour-stealing. The colours were 'stolen' from the picture on the right.

Observe that in both Figures 6.1 and 6.2 the coloring IFS is drawn on the initial picture.

## Bibliography

[1] Barnsley, Michael. (1988), Fractals everywhere, Boston MA, Academic Press.
[2] Barnsley, Michael. (2006), Superfractals, Cambrige University Press.
[3] Beardon, Alan F. (2005), Algebra and Geometry, Cambrige University Press.
[4] Cederberg, Judith N. (2005), A course in modern geometries, Second Edition. Springer.
[5] Mandelbrot, Benoit and Hudson, Richard. (2004), The (Mis)behavior of Markets: A Fractal View of Risk, Ruin, and Reward., Basic Books.
[6] Rudin, Walter. (1964), Principles of Mathematical Analysis, Second Edition. McGraw-Hill, New York.
[7] IFS Construction Kit webpage. http://ecademy.agnesscott.edu/~lriddle/ifskit/

## Copyright

The publishers will keep this document online on the Internet - or its possible replacement - for a period of 25 years from the date of publication barring exceptional circumstances. The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility. According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement. For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

Upphovsrätt
(c) 2011, Meritxell Joanpere Salvadó


[^0]:    ${ }^{1}$ See for example, chapter 7 in [3].

[^1]:    ${ }^{2}$ This applet can be found on http://www.math.ucla.edu/ tao/java/Mobius.html

[^2]:    ${ }^{1}$ IFS Construction Kit is a free software to design and draw fractals based on iterated function systems.

[^3]:    ${ }^{1}$ This Figure has been taken from finance.yahoo.com

[^4]:    ${ }^{2}$ This picture is from http://photojournal.jpl.nasa.gov/catalog/PIA03424

