

UiO : Matematisk institutt

Det matematisk-naturvitenskapelige fakultet

# Selvsimilære fraktaler og flatefyllende kurver

En analyse av selvsimilære systemer

**Ellisiv Wetjen Sigernes**

Masteroppgave, våren 2022





Denne masteroppgaven er levert inn som en del av programspesialiseringen *Matematikk* under *Lektorprogrammet* ved Universitetet i Oslo. Oppgaven er normert til 30 studiepoeng.

Forsiden viser et utsnitt av rotsystemet til den eksepsjonelle liegruppen  $E_8$ , projisert ned i planet. Liegrupper ble oppfunnet av den norske matematikeren Sophus Lie (1842–1899) for å uttrykke symmetriene til differensiallikninger og spiller i dag en sentral rolle i flere deler av matematikken.

---

# Sammendrag

---

Denne oppgaven har som mål å drøfte hvordan selvsimilaritet kan benyttes som et redskap i framstillingen og studiet av kurver med «patologiske» egenskaper, i all hovedsak fraktaler og flatefyllende kurver. For å kunne drøfte dette starter oppgaven med definisjon en av metriske rom før den går videre til å diskutere konvergens, kompletthet, kompakthet og kontinuitet. Dette leder til sist frem til Banach's fikspunktteorem.

Videre vil Hausdorff-metrikken introduseres og vises komplett som et steg for å knytte fraktalene og Banach's fikspunktteorem sammen. I studiet av fraktalene vil det i hovedsak fokuseres på selvtilnærte fraktaler, som studeres og beskrives gjennom to ulike måter å fremstille itererte funksjonssystemer (IFS). I kapittel 4 vil fraktaler som Sierpinski-trekanten og Koch-kurven studeres ved hjelp av en tilfeldighets-algoritme for IFS. Dette vil videre legge grunnlaget for å diskutere flatefyllende kurver i kapittel 5. Denne gangen ved hjelp av en deterministisk tilnærming av IFS.

---

# Abstract

---

The main purpose of this thesis is to examine how self-similarity can be used as a tool in outlining and studying curves with “pathological” properties, such as fractals and space-filling curves. The thesis starts with a formal definition of a metric space before it discusses topics as convergence, completeness, compactness, and continuity, before stating the Banach’s fixpoint theorem.

Furthermore, the Hausdorff metric will be introduced and its completeness will be discussed, this as a way of implementing Banach’s fixpoint theorem to the fractal space. In the study of fractals the thesis will mainly focus on self-similar fractals. To analyze the self-similar fractals, the thesis will apply two “ways of talking about” Iterated functions system (IFS). Chapter 4 discuss fractals such as the Sierpinski triangle and the Koch curve using a random algorithm for the IFS. This establishes the foundation for chapter 5 where self-similarity will be the topic. Here a deterministic approach to the IFS will be introduced.

---

# Forord

---

Med denne masteroppgaven markeres min tid på skolebenken og på universitetet som over – iallfall for denne gang! De siste årene har jeg hatt gleden av å studere matematikk på Lektorprogrammet i realfag. Dette munner nå ut i denne masteroppgaven i matematikk. Gjennom prosessen for å skrive masteroppgaven min har jeg hatt mange dyktige og flinke folk rundt meg, som alle fortjener en stor takk.

Jeg ønsker først og fremst å rette en stor takk til min svært tålmodige veileder Tom L. Lindstrøm som gang på gang har beroliget meg om at vi er på rett kurs og at denne oppgaven kommer i mål. Videre ønsker jeg også å rette en stor takk til Sissel Knatterød ved Matematisk-Naturvitenskapelig fakultet for godt støtte og mange gode samtaler når jeg har trengt det som mest.

Takk rettes også til gode kollegaer på Hartvig Nissen Videregående skole som har trodd på meg og bidratt med pedagogisk påfyll i en periode preget av svært mye matematikk-faglig input. Jeg ønsker også å takke mine medstudenter gjennom lærerike, spennende og krevende år: Vi klarte det!

Sist, men ikke minst rettes en stor og hjertelig takk til familien. Takk for all støtte, kjærlighet og omtanke, og ikke minst takk for at dere alltid har troen. En spesiell takk rettes til mamma, som i tillegg til mor og hobbypsykolog også har hjulpet med korrekturlesning av oppgaven, Tusen takk.

Til Bestemor, jeg vet du følger stolt med fra himmelen.

Ellisiv Wetjen Sigernes  
Oslo, 27.05.2022

---

# Innhold

---

<b>Sammendrag</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Forord</b>	<b>iii</b>
<b>Innhold</b>	<b>iv</b>
<b>Figurer</b>	<b>v</b>
<b>Tabeller</b>	<b>vi</b>
<b>1 Introduksjon</b>	<b>1</b>
1.1 Historisk perspektiv . . . . .	1
1.2 Oppgavens formål . . . . .	2
1.3 Disposisjon . . . . .	3
<b>2 Metriske rom</b>	<b>4</b>
2.1 Hva er et metrisk rom? . . . . .	4
2.2 Konvergens . . . . .	5
2.3 Kompletthet . . . . .	6
2.4 Kompakthet . . . . .	9
2.5 Kontinuitet . . . . .	10
<b>3 Hausdorff-metrikken</b>	<b>11</b>
3.1 Hva er Hausdorff-metrikken? . . . . .	11
3.2 Kompletthet av Hausdorff-metrikken . . . . .	12
<b>4 Selvsimilære systemer</b>	<b>16</b>
4.1 Fraktaler . . . . .	16
4.2 Selvsimilære fraktaler . . . . .	16
4.3 Itererte funksjonssystemer: Iteariv tilfeldighets-algoritme . . . . .	18
4.4 Eksempler på IFS for fraktaler . . . . .	19
<b>5 Flatefyllende kurver</b>	<b>30</b>
5.1 Flatefyllende kurver . . . . .	30
5.2 Itererte funksjonssystemer: Deterministisk algoritme . . . . .	31
5.3 Hilbert's flatefyllende kurve . . . . .	33

5.4	Flatefyllende kurve, ett eksempel . . . . .	36
<b>6</b>	<b>Oppsummering</b>	<b>39</b>
<b>A</b>	<b>Samling av Python koder</b>	<b>41</b>
A.1	Python koder for Sierpinski-trekant . . . . .	41
A.2	Python kode for Koch-kurven . . . . .	43
A.3	Python kode for Sierpinskis femkant . . . . .	44
A.4	Python kode for Hilberts flatefyllende kurve . . . . .	44
	<b>Bibliografi</b>	<b>48</b>

---

## Figurer

---

4.1	Sierpinski-trekanten, N=1 . . . . .	20
4.2	Sierpinski-trekanten, N=2 . . . . .	20
4.3	Python plott, Sierpinski-trekanten, N=1000 . . . . .	22
4.4	Python plott, Sierpinski-trekanten, N=10 000 . . . . .	22
4.5	Koch-kurven, utgangspunkt . . . . .	22
4.6	Koch-kurven, N=1 . . . . .	23
4.7	Koch-kurven, N=2 . . . . .	23
4.8	Python plott, koch-kurven, N=1000 . . . . .	24
4.9	Python plott, Koch-kurven, N=10 000 . . . . .	24
4.10	Sierpinskis femkant, utgangspunkt . . . . .	25
4.11	Sierpinskis femkant, 1 iterasjon . . . . .	25
4.12	Sierpinskis femkant, avstander til utregning . . . . .	26
4.13	Sierpinskis femkant, avbildning $\omega_2$ . . . . .	27
4.14	Sierpinskis femkant, avbildning $\omega_3$ . . . . .	27
4.15	Sierpinskis femkant, avbildning $\omega_4$ . . . . .	27
4.16	Sierpinskis femkant, avbildning $\omega_5$ . . . . .	27
4.17	Python plott, Sierpinskis femkant, N=10 000 . . . . .	29
4.18	Python plott, Sierpinskis femkant, N=100 000 . . . . .	29
5.1	Sierpinski-trekant plottet ved deterministisk kode, n=5 . . . . .	32
5.2	Sierpinski-trekant plottet ved deterministisk kode, n=10 . . . . .	32
5.3	Sierpinski-trekant plottet ved deterministisk kode, n=10, forstørret ca. 5-ganger . . . . .	32
5.4	Sierpinski-trekant plottet ved deterministisk kode, n=10, forstørret ca. 100 ganger . . . . .	32
5.5	Utgangspunkt Hilbert kurve . . . . .	33
5.6	Generert Hilberts kurve, n=1 . . . . .	33

5.7	Generert Hilberts kurve, n=2, usammenhengdene . . . . .	33
5.8	Generert Hilberts kurve, n=2, sammenhengende . . . . .	33
5.9	Hilbert kurven, n=1 . . . . .	35
5.10	Hilbert kurven, n=2 . . . . .	35
5.11	Hilbert kurven, n=3 . . . . .	35
5.12	Hilbert kurven, n=4 . . . . .	35
5.13	Hilbert kurven, n=5 . . . . .	35
5.14	Hilbert kurven, n=6 . . . . .	35
5.15	Alternativ Hilbert kurve, n=0 . . . . .	36
5.16	Alternativ Hilbert kurve, n=1 . . . . .	36
5.17	Plott av alternativ Hilbert kurve, n=1 . . . . .	37
5.18	Plott av alternativ Hilbert kurve, n=3 . . . . .	37
5.19	Plott av alternativ Hilbert kurve, n=6 . . . . .	37
5.20	Plott av alternativ Hilbert kurve, n=9 . . . . .	37
5.21	Plott av alternativ Hilbert kurve forstørret ca. 5 ganger, n=9 . . .	38
5.22	Plott av alternativ Hilbert kurve forstørret ca. 100 ganger, n=9 . .	38

---

## Tabeller

---

5.1	Tabell for IFS . . . . .	31
5.2	Tabell for Sierpinski-trekant . . . . .	31
5.3	Tabell for Hilbert kurven . . . . .	34
5.4	Tabell for Flatefyllende kurve . . . . .	36



# KAPITTEL 1

---

## Introduksjon

---

Matematikens historie strekker seg mange titusen år tilbake i tid. Mye tyder på at de første sporene av matematikken stammer tilbake til Ishago-beinet, et aritmetisk telleredskap benyttet i det som i dag er Kongo.

Det som ofte refereres til som den første matematikeren derimot er Tales fra Milet født rundt 624 f.kr. Gjennom sitt arbeid oppdaget Tales at en trekant innskrevet i en sirkel, der den ene sidekanten i trekanten tilsvarer diameteren i sirkelen, alltid vil være rettvinklet.

Fra Tales matematiske verden og frem til i dag har matematikken utviklet seg i rekordfart, og nye problemstillinger har oppstått, og noen også løst. Blant nye problemstillinger og tema i matematikken finner vi fraktaler og flatefyllende kurver. For *er det egentlig mulig å konstruere en kontinuerlig kurve som dekker alle punkt på en flate?* og *hva er egentlig et fraktal?*

Gjennom de neste kapitlene vil fraktalenes verden og flatefyllende kurver, samt spørsmålene fra siste avsnitt diskuteres. For å innlede fraktalenes verden og flatefyllende kurver starter kapitlet med et historisk perspektiv, hensikten med dette er å gi leseren grunnsteiner og perspektiver som en støtte til teorien i de kommende kapitlene. Kapitlet avsluttes med en kort disposisjon som beskriver oppsettet og rekkefølgen på resterende teori.

### 1.1 Historisk perspektiv

Lenge ble det kun snakket om funksjoner som skapte deriverbare kurver. Spoles det noen 1000 år frem i tid fra Milet og Ishago-benet, nærmere bestemt 1872, kom Karl Weierstrass opp med en ny type kurve. Denne kurven var allesteds kontinuerlig men ingensteds deriverbar. Selv om ordet *fraktal* for første gang kom til syne rundt 1975, og selv da med en vag definisjon, kan det likevel argumenteres for at Weierstrass kurve var starten på fraktaler.

I 1878 diskuterte matematikeren Georg Cantor tellbarhet i en artikkel publisert i *Crelle's Journal*. Gjennom sitt arbeid bidrar Cantor med viktig teori både innenfor læren om fraktaler og de flatefyllende kurvene. Gjennom artikkelen demonstrerte Cantor at enhver endelig-dimensjonal glatt mangfoldighet, uavhengig av dimensjon, har samme kardinalitet. Dette impliserer at intervallet

$[0, 1]$  kan avbildes bijektivt på kvadratet  $[0, 1]^2$ .

Dette fanger Eugen Nettos oppmerksomhet og i 1879 fulgte han opp Cantors kardinaliteter med å bevise at en slik avbildning må være diskontinuerlig. Spørsmålet nå ble om en kurve  $[0, 1]$  kan avbildes surjektivt på kvadratet  $[0, 1]^2$  og da være kontinuerlig?

Noen år senere, i 1883, introduserte Cantor også en funksjon  $\Psi$  der den deriverte var null, med unntak av noen punkter  $\{z\}$ . Mengden  $\{z\}$  utgjør, det nå anerkjente fraktalet, Cantor mengden. Cantor mengden har egenskaper som gjør at om man forstørrer inn på en del av mengden, så vil den nye forstørrede delen inneholde hele mengden igjen. Et slik fraktal vil senere bli kategorisert som et selvsimilært fraktal.

Giuseppe Peano regnes av mange som den flatefyllende kurvens far. I 1890 svarte han på spørsmålet om en flatefyllende kontinuerlig kurve var mulig hvis kravet om bijektivitet ble droppet. Dette gjorde han med å konstruere akkurat en slik kurve og dermed kom han David Hilbert i forkjøpet som beskrev en liknende kurve i 1891.

Tidlig 1900-tall tok matematikeren Helge Von Koch et steg videre fra Cantor mengden, da han introduserte Koch-kurven. Koch-kurven diskuteres nærmere i kapittel 4. Ved å kombinere tre av Koch-kurvene, skapes Kochs sneflak. Med sneflaket viste Koch en alternativ måte å snakke om ikke-deriverbarhet, samt knyttet koblingen mellom ikke-deriverbarheten i analyse og geometri.

I 1918 introduserte Felix Hausdorff konseptet Hausdorff dimensjonen, som viste seg å være svært nyttig i fagfeltene geometri og topologi. Hausdorff dimensjonen er ofte knyttet til fraktaler og kalles også gjerne for fraktaldimensjonen. Hausdorff dimensjonen vil ikke bli diskutert ytterligere gjennom denne oppgaven, selv om flere av figurene som diskuteres har noen svært interessante dimensjoner. Hausdorff dimensjonen legger også grunnlaget for Benoit Mandelbrots definisjon av fraktaler, denne definisjonen er sitert i kapittel 4.1.

## 1.2 Oppgavens formål

Gjennom denne masteroppgaven er det ønskelig å undersøke hvordan selvsimilaritet kan benyttes for å studere flatefyllende kurver.

Dette gjøres ved å studere utvalgte selvsimilære fraktaler som gjennom Hausdorff-metrikken, Banach's fikspunktteorem og itererte funksjonssystemer fremstiller (affine) kontraksjonsavbildninger som kan benyttes for å studere de utvalgte fraktalenes egenskaper.

Til sist vil metodene benyttet for å fremstille selvsimilære fraktaler, danne grunnlaget for grafisk fremstilling av flatefyllende kurver.

## 1.3 Disposisjon

Med utgangspunkt i oppgavens formål, er derfor resterende av oppgaven, i grove trekk, strukturert som følgende:

**Kapittel 2** gir en kort innføring i metriske rom. Her defineres det hva som legges til grunn for et metrisk rom, samt at egenskaper som konvergens, kompletthet og kontinuitet diskuteres.

**Kapittel 3** omhandler Hausdorff-metrikken, fraktalenes hjem. Her diskuteres det hvorvidt Hausdorff-metrikken faktisk er en metrikk, samt at det gjennom 3.2 bevises at denne metrikken er komplett. Dette som et viktig steg for å blant annet kunne benytte Banach's fikspunktteorem for å videre studere fraktaler.

**Kapittel 4** diskuterer fraktaler. Gjennom kapitlet vil de selvsimilære fraktaler spille størst rolle. For å videre kunne beskrive og diskutere fraktaler innføres itererte funksjonssystemer (IFS). Gjennom IFS beskrives kontraksjonsavbildningene som sammen skaper grunnlaget for å diskutere selvsimilære fraktaler. I kapittel 4.4 gis det en rekke eksempler på hvordan slike selvsimilære fraktaler kan fremstilles ved hjelp av IFS, samt hvordan Python kan brukes for å illustrere de.

**Kapittel 5** starter med å introdusere begrepet flatefyllende kurver. Videre sammenlignes og analyseres disse som selvsimilære fraktaler, ved hjelp av et IFS.

**Kapittel 6** gir en kort oppsummering og et overblikk over oppgaven.

**Tillegg A** samler alle Pythonkodene etter hvilket fraktal/kurve den hører under. Tillegget samler kodene referert til gjennom teksten, samt kodene som tidligere er gjengitt i sin helhet utover i oppgaven. Tillegget skiller også mellom deterministiske koder og koder basert på tilfeldighets-algoritmen.

## KAPITTEL 2

---

# Metriske rom

---

Metriske rom kan på mange måter kalles for fraktalenes hjem, for det er stort sett akkurat her, i de metriske rom, de befinner seg. Hensikten med følgende delkapittel er å gi en innføring i metriske rom. Dette som grunnlag for å videre kunne diskutere teori rundt selvsimilære systemer samt flatefyllende kurver.

### 2.1 Hva er et metrisk rom?

Et rom i matematikken defineres som en struktur, en måte å strukturere ulike objekter med felles sett regler, aksiomer og egenskaper. Barnsley ([Bar93]) forklarer rom som følgende:

**Definisjon 2.1.1** ([Bar93, s. 5]). Et rom  $X$  er en mengde. Punktene i rommet er elementene i mengden.

Eksempelvis vil  $\mathbb{R}$  kunne betegnes som et rom, bestående av de reelle tallene i tallsystemet.

Metriske rom inneholder også en reell funksjon som gir mulighet for distanssmål. Dette gjør metriske rom til en sentral del av og et nyttig verktøy i den geometriske verden. Et eksempel på et metrisk rom kan være de reelle tallene  $\mathbb{R}$  med metrikk  $|x - y|$ .

For planet, kan et metrisk rom foreksempel være  $\mathbb{R}^2$  med standard avstandsmål/metrikk  $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Et rom, eksempelvis  $\mathbb{R}^2$ , kan også være metrisk med andre metrikker. Et eksempel kan være metrikken  $|y_1 - x_1| + |y_2 - x_2|$  som betegner et avstandsmål mellom to punkter  $x = (x_1, x_2)$  og  $y = (y_1, y_2)$  i et plan der bevegelser kun skjer vannrett eller loddrett. Avstandsmålet  $d(x, y) = |y_1 - x_1| + |y_2 - x_2|$  kalles ofte "Manhattan metrikken" og er en metrikk i rommet  $X = \mathbb{R}^2$ .

Barnsley (1993) gir følgende definisjon av metriske rom

**Definisjon 2.1.2** ([Bar93, s. 10]). Et metrisk rom  $(X, d)$  er et rom  $X$  sammen med en reell funksjon  $d : X \times X \rightarrow \mathbb{R}$  som gir oss avstanden mellom par av punkter  $x, y \in X$ . Vi krever at funksjonen  $d$  følger følgende aksiomer:

1.  $d(x, y) = d(y, x)$  for alle  $x, y \in X$
2.  $0 < d(x, y) < \infty$ , for alle  $x, y \in X$ ,  $x \neq y$



3.  $d(x, x) = 0$  for alle  $x \in X$
4.  $d(x, y) \leq d(x, z) + d(z, y)$  for alle  $x, z, y \in X$  (Senere referert til som trekantulikheten).

En slik funksjon  $d$ , kalles en metrikk.

Dermed er metrikken ved definisjon alltid ikke-negativ, symmetrisk og oppfyller trekantulikheten.

Metrikkene fra eksemplene over, eksempelvis  $\mathbb{R}$  med metrikk  $|x - y|$ , må dermed følge de fire aksiomene for å kunne være metrikker.

## 2.2 Konvergens

Når en rekke, følge eller annen matematisk konstruksjon nærmer seg en grense, heter det at den konvergerer. Eksempelvis vil følgen  $a_n = \frac{n}{n+1}$  konvergere mot 1. Dette kan lett vises ved å sette  $n$  til et større og større tall slik at kvotienten  $a_n$  får en verdi nærmere og nærmere 1. For å forenkle konvergens tas det først utgangspunkt i konvergente følger i  $\mathbb{R}^d$ .

**Definisjon 2.2.1** ([HL15, s. 414]). Følgen  $a_n \in \mathbb{R}^d$  konvergerer mot  $a$ , hvis det for enhver  $\epsilon > 0$  eksisterer et heltall  $N$  slik at for alle  $n \geq N$  er  $|a_n - a| < \epsilon$ .

På lik linje med følgen  $a_n = \frac{n}{n+1}$  kan også en følge  $x_n$  i det metriske rommet  $X$  konvergere. For å synliggjøre dette tas det utgangspunkt i en metrikk som forklart i definisjon 2.1.2.

Metrikkene  $d(x_n, a)$  vil skape en følge av avstandsmål fra ulike punkter  $x_n$  til punktet  $a$ . Ved definisjon vil en slik følge bestå av ikke-negative tall. Siden metrikken betegner avstandsmål vil følgen  $d(x_n, a)$  konvergere mot 0, jo nærmere punktene  $x_n$  og  $a$  er hverandre. Dette gir utgangspunktet for følgende generalisering av definisjon 2.2.1.

**Definisjon 2.2.2** ([Bar93, s. 17]). En følge  $(x_n)_{n=1}^{\infty}$  av punkter i det metriske rommet  $(X, d)$  konvergerer mot et punkt  $a \in X$  hvis det, for enhver  $\epsilon > 0$ , eksisterer et heltall  $N > 0$  slik at for enhver  $n \geq N$  er  $d(x_n, a) < \epsilon$ .

På lik linje med generelle konvergente følger, har også konvergente følger i det metriske rom høyst en grenseverdi. Videre kan man med enkelhet vise at enhver konvergent rekke i det metriske rommet  $X$  vil være en Cauchy-følge.

**Definisjon 2.2.3** ([Bar93, s. 16–18]). En følge  $(x_n)_{n=1}^{\infty}$  av punkter i det metriske rommet  $(X, d)$  er en Cauchy-følge dersom det, for enhver  $\epsilon > 0$ , eksisterer et heltall  $N > 0$  slik at  $d(x_n, x_m) < \epsilon$  når  $n, m \geq N$ .

Definisjon 2.2.1 gir opphavet til følgende teorem:

**Teorem 2.2.4** ([Lin17, s. 37]). *En følge  $(a_n) \in \mathbb{R}^m$  konvergerer hvis og bare hvis den er en Cauchy-følge.*

Teorem 2.2.4 viser den tette sammenhengen mellom konvergente følger og Cauchy-følger i  $\mathbb{R}^d$ . Det er derfor av interesse å utvide teoremet til å også gjelde følger i metriske rom. For å innlede dette tas det utgangspunkt i følgende:

**Setning 2.2.5.** Hvis en følge  $(x_n)_{n=1}^\infty \in (X, d)$  konvergerer, så er den en Cauchy-følge.

*Bevis:*

Antar at  $(x_n)_{n=1}^\infty \in (X, d)$  konvergerer mot et punkt  $a \in X$ . Gitt en  $\epsilon > 0$ , så eksisterer en  $N \in \mathbb{N}$  slik at  $d(x_n, a) < \frac{\epsilon}{2}$  for alle  $n \geq N$ . Hvis  $n, k \geq N$  så er

$$d(x_n, x_k) \leq d(x_n, a) + d(a, x_k) < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.$$

Dermed må  $(x_n)_{n=1}^\infty \in (X, d)$  være en Cauchy-følge, og alle konvergente følger i  $(X, d)$  er Cauchy følger.

Å vise at en Cauchy-følge er konvergent lar seg ikke gjøre for generelle metriske rom, selv om dette kan vises å stemme eksempelvis for metriske rom i  $\mathbb{R}^m$  slik som teorem 2.2.4 foreslår.

Cauchy-følger er et nyttig verktøy for å vise konvergens for metrikker i metriske rom. Cauchy-følger spiller også en viktig rolle i å vise kontinuitet og kompletthet for metriske rom.

## 2.3 Kompletthet

Tall, tall-linjer og derivasjon er begreper som tidlig kommer til syne i skolematematikken. Allerede i barneskolen introduseres tall-linjen, og etter endt matematikk 1T bør begrepet derivasjon være kjent [Utd20]. Alt dette bygger på prinsippet kompletthet.

Prinsippet kompletthet er svært viktig innenfor matematikken, og tilnærmet essensielt innen analyse. Dermed er det svært ønskelig med redskaper for å også vise at ulike metriske rom er komplette.

Allerede i noen av de første matematikk-kursene på universitetsnivå lærer man at de reelle tallene  $\mathbb{R}$  er komplette og dermed gir opphav til en rekke egenskaper, blant annet gjennom kompletthetsprinsippet [Lin12, s. 94].

Kompletthetsprinsippet gir følgende:

**Teorem 2.3.1** ([Lin12, s. 94]). *Enhver ikke-tom, oppad begrenset delmengde av  $\mathbb{R}$  har en minste øvre skranke.*

En minste øvre skranke betegnes ofte med forkortelsen *Sup*, slik at den minste øvre skranken for delmengde  $A$  er  $Sup(A)$ . En øvre skranke tilsvarer et tall  $c$  som er større enn alle elementene i delmengden  $A$ . En minste øvre skranke tilsvarer det laveste verdien  $c$  kan ha, og fremdeles være en øvre skranke. Nettopp et slikt tall forteller kompletthetsprinsippet at eksisterer.

Kompletthet vil også kunne gi en rekke forenklinger i videre regneproseser i det metriske rom, dermed er det ønskelig med en formell definisjon av dette. For å forklare kompletthet i et metrisk rom  $(X, d)$ , tas det dermed utgangspunkt i Cauchy-følger.

**Definisjon 2.3.2** ([Bar93, s. 18]). Et metrisk rom  $(X, d)$  er komplett hvis enhver Cauchy-følge  $(x_n)_{n=1}^{\infty}$  i  $X$  har en grense  $x \in X$ .

Med andre ord er et metrisk rom komplett hvis alle Cauchy-følger konvergerer. Også enkelte delmengder av komplette metriske rom kan vises komplette. Eksempelvis enkelte rom i  $\mathbb{R}^m$  er komplette, dette gjelder blant annet  $m = 2$  med en Euklidsk metrikk.

Først gis det en kort definisjon av begrepene *omegn*, *randpunkt*, *lukket* og *åpen* mengde. Dette benyttes videre både i redegjørelsen for kompletthet og kompaktthet.

En omegn til et punkt  $A$  kan forklares som en ball med senter i  $A$ . Med andre ord kan en omegn tolkes som en delmengde der alle elementene ligger relativt nære punktet  $A$ . I boken *Spaces* gir T. Lindstrøm følgende eksempel på en omegn:

Et åpent sett som inneholder  $x$  er en omegn for  $x$ .

Et randpunkt for en mengde  $A$  vil være et punkt som ligger på randen av mengden. Nærmere forklart

**Definisjon 2.3.3** ([HL15, s. 412]). Et punkt  $a$  defineres som et randpunkt i mengden  $A$  hvis alle omegner av  $a$  inneholder minst ett punkt  $b \in A$  og ett punkt  $c \notin A$ .

Videre er

**Definisjon 2.3.4** ([HL15, s. 413]). En mengde er åpen hvis den ikke inneholder noen av sine randpunkter. En mengde er lukket hvis den inneholder alle sine randpunkter.

Anta at  $S \subset X$  er en delmengde av det komplette metriske rommet  $(X, d)$ . Antar at  $\{s_n\}$  er en Cauchy-følge i  $S$ . Siden  $S$  er en delmengde av  $X$ , og  $X$  er komplett, må også  $\{s_n\}$  være Cauchy-følger i  $X$ . Hvis  $\{s_n\}$  er Cauchy-følger i  $X$  må de ved definisjon 2.3.1 konvergere mot et punkt  $a$ . Dermed vil  $S \subset X$  også være et komplett rom hvis  $a \in S$ . Dette oppfylles om  $S$  er en lukket følge.

Dette gir opphavet til følgende teorem

**Teorem 2.3.5** ([Bar93, s. 20]). La  $(X, d)$  være et komplett metrisk rom og la  $S \subset X$ . Da er  $S$  komplett hvis  $S$  er lukket.

Altså vil enhver delmengde av et komplett metrisk rom også være komplett, om delmengden er lukket.

Enkelte situasjoner og systemer i matematikken kan benytte kompletthet for å finne tilnærmede løsninger. Ta utgangspunkt i det metriske rommet  $(X, d)$  med iterativt funksjonssystem  $f : X \rightarrow X$  som avbilder  $X$  på seg selv. Her kan en repeterende rekke av tilnærminger finne en løsning for systemet, dette skjer ved at det utfra initialbetingelsene  $x_0$  konstrueres en følge  $x_0, x_1, x_2, \dots$  der  $x_1 = f(x_0), x_2 = f(x_1), \dots$

Ved å gjentatte ganger tilnærme  $x_n = f(x_{n-1})$  vil det til slutt stabilisere seg i en slags likevekststilstand der  $f(a) = a$ . Et slikt punkt  $a$  kalles for et fikspunkt.

**Definisjon 2.3.6** ([Lin17, s. 68]). En funksjon  $f : X \rightarrow X$  kalles en sammentrekning med kontraksjonsfaktor  $s$  for  $f$ , hvis det eksisterer positivt tall  $s < 1$  slik at  $d(f(x), f(y)) \leq sd(x, y)$  for alle  $x, y \in X$ .

Dette leder opp til *Banach's fikspunktteorem* (teorem 2.3.7) som senere vil vise seg som nyttig i analysen av selvsimilære fraktaler.

**Teorem 2.3.7** ([Lin17, s. 68]). (*Banach's fikspunktteorem*): Anta at  $(X, d)$  er et komplett metrisk rom, og at funksjonen  $f : X \rightarrow X$  er en sammentrekning. Da har  $f$  et unikt fikspunkt  $a$ , slik at uavhengig av initialbetingelse  $x_0 \in X$  vil følgen

$$x_0, x_1 = f(x_0), x_2 = f(f(x_0)), \dots, x_n = f^{on}(x_0) \text{ konvergerer mot } a.$$

*Bevis: Banach's fikspunktteorem*

For å bevise Banach's fikspunktteorem, må følgende vises:

1. At  $f$  har ikke mer enn et fikspunkt
2.  $\{x_n\}$  er en Cauchy-følge
3. at  $f$  har et fikspunkt.

(1.) Bevises ved motsigelse. Anta at  $f$  har to fikspunkt  $a, b$  og at  $s$  er slik teoremet krever, en sammentrekning. Da gir definisjon 2.3.6 at  $d(a, b) = d(f(a), f(b)) \leq sd(a, b)$ , der  $s < 1$  er et positivt tall. Dermed er  $d(a, b) = d(f(a), f(b)) \leq sd(a, b)$  kun mulig om  $a = b$  slik at  $d(a, b) = 0$ . Dermed kan  $f$  kun ha et fikspunkt.

(2.) For å bevise at det eksisterer et fikspunkt, tas det utgangspunkt i  $x_0 \in X$  der  $x_0$  er startpunkt for følgende følge

$$x_0, x_1 = f(x_0), x_2 = f(f(x_0)), \dots, x_n = f^{on}(x_0), \dots$$

Siden  $(X, d)$  er et komplett metrisk rom, gir definisjon 2.3.2 at alle Cauchy-følger i  $X$  konvergerer. Hvis vi da kan vise at følgen  $\{x_n\}$  er en Cauchy-følge, vi dermed  $x_n$  konvergere mot et grensepunkt  $a$ . Dermed vil iterasjonen av følgen  $x_{n+1} = f(x_n)$  utvikle seg mot  $a = f(a)$  når  $n \rightarrow \infty$ . Dermed må  $a$  være et fikspunkt på  $f$ .

(3.) For å undersøke om  $\{x_n\}$  er en Cauchy-følge velges to elementer i følgen,  $\{x_n\}$  og  $\{x_{n+k}\}$ . Siden  $d$  er en metrikk må trekantulikheten holde, slik at:

$$d(x_n, x_{n+k}) \leq d(x_n, x_{n+1}) + d(x_{n+1}, x_{n+2}) + \dots + d(x_{n+k-1}, x_{n+k})$$

Som igjen vil tilsvare

$$d(x_n, x_{n+1}) + d(x_{n+1}, x_{n+2}) + \dots + d(x_{n+k-1}, x_{n+k}) = d(f^{on}(x_0), f^{on}(x_1)) + d(f^{on+1}(x_0), f^{on+1}(x_1)) + \dots + d(f^{on+k-1}(x_0), f^{on+k-1}(x_1))$$

Ved definisjon 2.3.5 må følgende holde for kontraksjonsavbildninger:  $d(f(x), f(y)) \leq sd(x, y)$ . Dermed må:

$$d(f^{on}(x_0), f^{on}(x_1)) + d(f^{on+1}(x_0), f^{on+1}(x_1)) + \dots + d(f^{on+k-1}(x_0), f^{on+k-1}(x_1)) \leq s^n d(x_0, x_1) + s^{n+1} d(x_0, x_1) + \dots + s^{n+k-1} d(x_0, x_1)$$



Ved hjelp av geometriske rekker kan det igjen skrives om til følgende:

$$\frac{s^n(1-s^k)}{1-s}d(x_0, x_1) \leq \frac{s^n}{1-s}d(x_0, x_1)$$

dermed er  $d(x_n, x_{n+k}) \leq \frac{s^n}{1-s}d(x_0, x_1)$  slik at ved å velge en  $n$  stor nok vil

$$d(x_n, x_{n+k}) \leq \frac{s^n}{1-s}d(x_0, x_1) \leq \epsilon$$

for en  $\epsilon > 0$ . Dermed er  $\{x_n\}$  en Cauchy-følge, og Banach's fikspunktteorem holder.

## 2.4 Kompakthet

I jakten på kompakthet må vi tilbake til læren om følger og delfølger. Delfølger er gitt ved deler av en følge, det blir en slags underkategori av den opprinnelige følgen. Eksempelvis vil  $\frac{1}{1+2n}$  være en delfølge av  $\frac{1}{1+n}$ . Delfølgen inneholder med andre ord et utvalg av elementer fra den opprinnelige følgen.

Videre er en mengde begrenset hvis følgende gjelder:

**Definisjon 2.4.1** ([Bar93, s. 20]). La  $(X, d)$  være et metrisk rom og  $S \subset X$ . Da er  $S$  begrenset hvis det eksisterer et punkt  $a \in X$  og et tall  $R > 0$  slik at  $d(a, x) < R$  for alle  $x \in X$ .

Eksempelvis vil alle Cauchy-følger i det metriske rommet være begrenset.

*Bevis alle Cauchy-følger er begrenset*

Antar at følgen  $(x_n)_{n=1}^{\infty}$  er en Cauchy-følge velger å skrive følgen  $(x_n)_{n=1}^{\infty}$  slik at  $(x_n)_{n=1}^{\infty} = (x_1, x_2, \dots, x_{\infty})$ .

Ved definisjon 2.2.3, må det for enhver  $\epsilon > 0$ , for eksempel  $\epsilon = 1$ , eksistere et heltall  $N > 0$  slik at  $d(x_n, x_m) < 1$  for alle  $n, m \geq N$ . Følgelig må det dermed eksistere en  $M = \text{Max}(1, d(x_1, x_N), \dots, d(x_{N-1}, x_N))$  slik at  $d(x_n, x_m) \leq 1$ . Dermed har Cauchy-følgen en øvre skranke  $\epsilon$ . Slik at følgen må være oppad begrenset.

Videre er det også av interesse og definere begrepet totalt begrenset.

**Definisjon 2.4.2** ([Bar93, s. 20]). La  $(X, d)$  være et metrisk rom og  $S \subset X$ . Da er  $S$  totalt begrenset hvis det for enhver  $\epsilon > 0$ , eksisterer et endelig mengde av punkt  $(y_1, y_2, \dots, y_n) \subset S$  slik at når  $x \in X$  så er  $d(x, y_i) < \epsilon$  for en  $y_i \in (y_1, y_2, \dots, y_n)$ .

Ethvert totalt begrenset metrisk rom, vil også være begrenset, men ikke nødvendigvis vice versa. At en mengde er begrenset vil si at hele mengden befinner seg innenfor en radius  $r$ . Hvis en mengde er totalt begrenset vil det bety at du kan dekke hele mengden med et endelig antall små radiuser  $\epsilon$ . For alt arbeid i planet, vil begrenset og totalbegrenset være synonymt.

Med definisjonene for begrenset, totalt begrenset og lukket definert kan en nå gå videre til kompakthet.

**Definisjon 2.4.3** ([Bar93, s. 20]). La  $(X, d)$  være et metrisk rom og  $S \subset X$ . Da kalles  $S$  kompakt dersom enhver følge i  $S$  har en delfølge som konvergerer mot et punkt i  $S$ .

Definisjon 2.4.3 innleder med dette for følgende teorem.

**Teorem 2.4.4** ([Bar93, s. 20]). La  $(\mathbb{R}^n, d)$  være et metrisk rom og  $S \subset \mathbb{R}^n$ . Da er  $S$  kompakt hvis og bare hvis den er lukket og totalt begrenset.

Dermed er et kompakt metrisk rom også komplett. Videre gir ekstremalverdisetningen følgende for kompakte delmengder av  $(X, d)$

**Teorem 2.4.5** ([Lin17, s. 73]). (Ekstremalverdisetningen): Anta at  $K$  er en ikke-tom, kompakt delmengde av  $(X, d)$  og at  $f : K \rightarrow \mathbb{R}$  er en kontinuerlig funksjon. Da har  $f$  et maksimum og et minimumspunkt i  $K$ .

Med andre ord, for en ikke-tom, kompakt delmengde av  $(X, d)$  med en kontinuerlig funksjon  $f$ , eksisterer  $c, d \in K$  slik at  $f(d) \leq f(x) \leq f(c)$  for alle  $x \in X$ .

## 2.5 Kontinuitet

En velkjent definisjon og en fundamental del av kalkulus lyder som følger:

**Definisjon 2.5.1** ([Lin12, s. 212]). En funksjon  $\mathbb{R}^n \rightarrow \mathbb{R}^m$  er kontinuerlig i  $a$  hvis det for hver  $\epsilon > 0$  eksisterer en  $\delta > 0$  slik at  $\|F(x) - F(a)\| < \epsilon$  når  $\|x - a\| < \delta$ .

Denne definisjonen er også grunnlaget for kontinuitet i metriske rom. Med utgangspunkt i definisjon 2.5.1 defineres derfor følgende

**Definisjon 2.5.2** ([Lin17, s. 56]). Anta to metriske rom  $(X, d_X)$  og  $(Y, d_Y)$ . En funksjon  $F : X \rightarrow Y$  er kontinuerlig i et punkt  $a \in X$  hvis det for enhver  $\epsilon > 0$  eksisterer en  $\delta > 0$  slik at  $d_Y(F(x), F(a)) < \epsilon$  når  $d_X(x, a) < \delta$ .

Akkurat som i kalkulus, er en funksjon kontinuerlig om den er kontinuerlig i alle punkt.

Videre er det også av interesse å diskutere kontinuitet for avbildninger i metriske rom

**Lemma 2.5.3** ([Bar93, s. 79]). La  $f : X \rightarrow X$  være en sammentrekning på det metriske rommet  $(X, d)$ . Da er  $f$  kontinuerlig.

*Bevis sammentrekningen er kontinuerlig*

Fra definisjon 2.5.2 er en  $f : X \rightarrow X$  kontinuerlig hvis det for hver  $\epsilon > 0$  finnes en  $\delta > 0$  slik at  $d(f(y), f(x)) < \epsilon$ . Siden  $f$  er en kontraksjonsavbildning, vil det ved definisjon eksistere en kontraksjonsfaktor  $s > 0$  slik at  $d(f(y), f(x)) \leq sd(y, x)$ . For  $d(y, x) < \delta$  og  $\delta = \frac{\epsilon}{s}$  vil derfor  $d(f(y), f(x)) \leq sd(y, x) < \epsilon$  slik at kontraksjonsavbildningen  $f$  er kontinuerlig.

## KAPITTEL 3

---

# Hausdorff-metrikken

---

Gjennom å introdusere Hausdorff-metrikken er målet å legge grunnlaget for å kunne måle avstand mellom ulike delmengder i et metrisk rom, og på den måten kunne bevege oss videre til å snakke om fraktaler. Det vil også være relevant å bevise at denne metrikken er komplett for å senere kunne snakke om fraktaler i lys av Banach's fikspunktteorem.

### 3.1 Hva er Hausdorff-metrikken?

Hausdorff-metrikken, og viktigheten av denne, kommer tydelig frem når en definerer følgende for delmengder i det metriske rom:

**Definisjon 3.1.1** ([Bar93, s. 27]). La  $(X, d)$  være et komplett metrisk rom, da er  $\mathcal{H}(X)$  rommet bestående av de ikke-tomme kompakte delmengdene av  $X$ .

$\mathcal{H}(X)$  inneholder utvalgte delmengder av det metriske rommet  $X$  og gjør med dette  $\mathcal{H}(X)$  til et ideelt rom for fraktalet. For videre analyser av fraktaler er det derfor ønskelig å vise at  $(\mathcal{H}(X), h)$  kan være et komplett metrisk rom. Første steg på veien vil være å finne metrikken  $h$  for rommet  $\mathcal{H}(X)$ . For å kunne definere en slik metrik må det først avklares hvordan man kan finne avstanden fra et punkt, for eksempel  $x$ , til en mengde  $B$ .

**Definisjon 3.1.2** ([Bar93, s. 27]). For et komplett metrisk rom  $(X, d)$ , punktet  $x \in X$  og delmengden  $B \in \mathcal{H}(X)$ , er avstanden fra punktet  $x$  til mengden  $B$  gitt ved  $d(x, B) = \min\{d(x, y) : y \in B\}$ .

Videre vil metrikken  $h$  for rommet  $\mathcal{H}(X)$  defineres ved:

**Definisjon 3.1.3** ([Bar93, s. 29, 32]). For et komplett metrisk rom  $(X, d)$  er Hausdorff-metrikken i  $\mathcal{H}(X)$  gitt ved:  $h(A, B) = \max(d(A, B), d(B, A))$ , der  $d(A, B) = \max\{d(x, B) : x \in A\}$  og  $d(B, A) = \max\{d(x, A) : x \in B\}$ .

I definisjonen betegner  $d(A, B)$  avstanden mellom settet  $A \in \mathcal{H}(X)$  og  $B \in \mathcal{H}(X)$ , og  $d(B, A)$  vice versa.

For å vise at Hausdorff-metrikken  $h$  er en metrik, må det vises at den oppfyller aksiomene fra definisjon 2.1.2.

Dette vil si at  $h$  må bevises *ikke-negativ*, *symmetrisk* samt *oppfyller trekantulikheten*.

### 3.2. Kompletthet av Hausdorff-metrikken

*Symmetrisk:* Skal metrikken  $h$  være symmetrisk må  $h(A, B) = h(B, A)$  for alle  $A, B \in \mathcal{H}(X)$ . Ved definisjon er Hausdorff-metrikken den største av avstandene  $d(A, B)$  og  $d(B, A)$ , dermed vil den også ved definisjon være symmetrisk slik at

$$h(A, B) = h(B, A) = \max(d(A, B), d(B, A))$$

*Ikke-negativ:* For å vise dette må vi vise  $0 < h(A, B) < \infty$ , for alle  $A, B \in \mathcal{H}(X)$ ,  $A \neq B$ , samt at  $h(A, A) = 0$  for alle  $A \in \mathcal{H}(X)$ . At  $h(A, B) \geq 0$  og at  $h(A, A) = 0$ , følger direkte fra definisjonen av Hausdorff-metrikken.

Dermed gjenstår det å vise at  $h(A, B) > 0$  når  $A \neq B$ . Definisjon 3.1.2 gir at hvis  $a \in X$  og  $E \subset X$  så er  $d(a, E) = \min\{d(a, y) : y \in E\}$ .

Dermed hvis  $K \subset X$  og hvis  $K$  er kompakt så gir ekstremalverdiesteningen (Teorem 2.4.5) at det eksisterer et maksimumspunkt  $c$  og et minimumspunkt  $d$ , slik at  $d(a, d) \leq d(a, y) \leq d(a, c)$  for alle  $y \in K$ . Dermed må  $d(a, K) = d(a, d)$ . Med andre ord ligger punktet  $d \in K$  nærmest  $a$ . Dersom  $a \notin K$  er da  $d(a, K) = d(a, d) > 0$ . Videre hvis  $A, B \subset X$  og  $A \neq B$ , så gir teorem 3.1.3 at  $h(A, B) = \max(d(A, B), d(B, A))$  slik at  $h(A, B) > 0$ .

*Oppfyller trekantulikheten:* Siden argumentene vil være gyldig uavhengig av om  $h(A, B) = d(A, B)$  eller  $h(A, B) = d(B, A)$ , argumenteres det videre med utgangspunkt i at  $h(A, B) = d(A, B)$ . Velg  $a \in A$  slik at

$$d(a, b) = \sup\{d(x, B) : x \in A\} = d(A, B),$$

videre velges også en  $c \in C$  slik at

$$d(a, c) = \inf\{d(a, z) : z \in C\} \leq d(A, C).$$

til sist velges også en  $b \in B$  slik at

$$d(c, b) = \inf\{d(c, y) : y \in B\} \leq d(C, B).$$

Dermed vil  $h(A, B) = d(A, B) = d(a, b)$ . Siden  $d(a, b)$  er en metrikk i  $X$  vil

$$d(a, b) \leq d(a, c) + d(c, b) \text{ slik at}$$

$$h(A, B) \leq d(a, c) + d(c, b) \leq d(A, C) + d(C, B) = h(A, C) + h(C, B).$$

Dermed er  $h(A, B) \leq h(A, C) + h(C, B)$  og trekantulikheten er oppfylt.

Med dette må Hausdorff-metrikken,  $h(A, B) = \max(d(A, B), d(B, A))$ , være en metrikk.

### 3.2 Kompletthet av Hausdorff-metrikken

I kapittel 2.3 defineres et metrisk rom som komplett dersom alle Cauchy-følger konvergerer. Det er ønskelig å utvide dette, til å også kunne snakke om kompletthet av Hausdorff-metrikken. I dette delkapittelet ønsker en da å vise at, dersom  $(X, d)$  er komplett, så er også rommet  $(\mathcal{H}(X), h)$  et komplett metrisk rom. Teorien gjennom dette delkapittelet vil i all hovedsak være hentet fra [Lin22a].

Vi definerer følgende

**Definisjon 3.2.1** ([Bar93, s. 34]). La  $A \subset X$  og  $\epsilon \geq 0$ . Da er  $A + \epsilon = \{y \in X : d(x, y) \leq \epsilon, x \in A\}$ . Vi kaller  $A + \epsilon$  for utvidelsen av  $A$  med en ball med radius  $\epsilon$ .



### 3.2. Kompletthet av Hausdorff-metrikken

Ved hjelp av definisjonen over, kan følgende vises ekvivalent for metrikken  $h$ :

**Lemma 3.2.2.** For  $A, B \in \mathcal{H}(X)$  der  $(X, d)$  er et metrisk rom og  $\epsilon > 0$  så er  $h(A, B) \leq \epsilon$  hvis og bare hvis  $A \subset B + \epsilon$  og  $B \subset A + \epsilon$ .

*Bevis*  $h(A, B) \leq \epsilon \Leftrightarrow A \subset B + \epsilon$  og  $B \subset A + \epsilon$

Anta at  $d(A, B) \leq \epsilon$ , ved definisjon 3.1.2 vil  $d(A, B) = \max\{d(a, B) : a \in A\} \leq \epsilon$  slik at også  $d(a, B) \leq \epsilon$  for alle  $a \in A$ . Siden  $d(a, B) \leq \epsilon$  så må også  $a \in B + \epsilon$  og  $A \subset B + \epsilon$ , da  $a$  er et vilkårlig element i  $A$ . Dermed er  $d(A, B) \leq \epsilon \Leftrightarrow A \subset B + \epsilon$ .

Tilsvarende vil  $d(B, A) \leq \epsilon \Leftrightarrow B \subset A + \epsilon$ . Siden Hausdorff-metrikken er gitt i definisjon 3.1.2 ved  $h(A, B) = \max(d(A, B), d(B, A))$  der  $d(B, A) \leq \epsilon \Leftrightarrow B \subset A + \epsilon$  og  $d(A, B) \leq \epsilon \Leftrightarrow A \subset B + \epsilon$  må det  $h(A, B) \leq \epsilon$ .

Videre, la  $\{A_n\}$  være en Cauchy-følge i  $(\mathcal{H}(X), h)$ . Da vil det tilsvarende for enhver  $\epsilon > 0$  finnes en  $N \in \mathbb{N}$  slik at alle  $n, m \geq N$  gir at  $A_n \subset A_m + \epsilon$  og  $A_m \subset A_n + \epsilon$ .

Med disse forutsetningene er det ønskelig å vise at dersom  $\{A_n\}$  er en Cauchy-følge i  $\mathcal{H}(X)$  så finnes det en  $A \in \mathcal{H}(X)$  slik at  $A = \lim_{n \rightarrow \infty} A_n \in \mathcal{H}(X)$ . Skissen for beviset er i hovedtrekk følgende:

1. Vise at alle  $\{A_n\}_{n \in \mathbb{N}}$  er inneholdt i samme kompakte delmengde  $C$
2. Definere  $A$  og vise at denne er ikke-tom
3. Bevise at  $A$  er en lukket delmengde av  $C$  og dermed kompakt
4.  $\lim A_n = A$ .

For å vise dette er det ønskelig å vise at det eksisterer en kompakt mengde  $C$ , konstruert slik at den inneholder alle  $\{A_n\}_{n \in \mathbb{N}}$  i  $\mathcal{H}(X)$ . Dette vil forenkle prosessen, da enhver lukket delmengde av en kompakt mengde, er kompakt.

**Lemma 3.2.3.** Hvis  $\{A_n\}_{n \in \mathbb{N}}$  er en Cauchy-følge i  $\mathcal{H}(X)$  så er tillukningen  $C = \overline{\bigcup_{n \in \mathbb{N}} A_n}$  kompakt.

*Bevis*  $C$  er kompakt :

Fra Teorem 2.4.4 er en delmengde av et metrisk rom kompakt hvis den er lukket og totalt begrenset. Siden  $C$  er tillukningen gitt ved Cauchy-følgene  $\{A_n\}_{n \in \mathbb{N}}$  er den ved definisjon allerede lukket. Dermed gjenstår det å vise at den er totalt begrenset.

Anta at  $C$  ikke er totalt begrenset, dette betyr at det eksisterer  $\epsilon > 0$  slik at  $C$  ikke kan dekkes av et endelig antall baller med radius  $\epsilon$ . Lemma 3.2.2 gir dermed at det må eksistere en følge  $\{x_n\} \in C$  slik at  $d(x_n, x_m) \geq \epsilon$ .

Videre siden  $\{A_n\}_{n \in \mathbb{N}}$  er en Cauchy-følge må det eksistere en  $N \in \mathbb{N}$  slik at  $C = \overline{\bigcup_{n \in \mathbb{N}} A_n} \subseteq \bigcup_{n=1}^N A_n + \frac{\epsilon}{3}$ .

### 3.2. Kompletthet av Hausdorff-metrikken

Dermed må det for hver  $\{x_n\}$  være mulig å velge en  $y_n \in \bigcup_{n=1}^N A_n$  slik at  $d(y_n, x_n) \leq \frac{\epsilon}{3}$ . Siden  $d(y_n, x_n)$  er en metrikk, må trekantulikheten gjelde. Dermed må  $d(x_n, x_m) \leq d(x_n, y_n) + d(y_n, y_m) + d(y_m, x_m)$ . Siden  $d(x_n, x_m)$  må være større en  $\epsilon$  og  $d(x_n, y_n), d(x_m, y_m) \leq \frac{\epsilon}{3}$  krever trekantulikheten at  $d(y_n, y_m) \geq \frac{\epsilon}{3}$ . Skal  $d(y_n, y_m) \geq \frac{\epsilon}{3}$  vil  $\{y_n\}$  ikke kunne ha noen konvergent delfølge. Dette gir en motsigelse siden  $y_n \in \bigcup_{n=1}^N A_n$  og dermed ligger i en kompakt mengde. Det er nå bevist at det må eksistere en følge  $\{x_n\} \in C$  slik at  $d(x_n, x_m) < \epsilon$  og  $C$  må dermed være totalbegrenset. Side  $C$  er både lukket og totalbegrenset er den ved teorem 2.4.4 også kompakt.

For å forenkle senere bevis defineres den potensielle grensen  $A = \lim_{n \rightarrow \infty} A_n \in \mathcal{H}(X)$  til  $\{a_n\} \in B_n$  der  $B_n = \bigcup_{m \geq n} A_m$ . Videre, la  $\mathcal{S}$  være mengden av alle følger  $\{x_n\}$  slik at  $x_n \in B_n$  for  $n \in \mathbb{N}$ .

**Lemma 3.2.4.** Hvis  $A = \{a \in X : \{a_n\} \in \mathcal{S} \text{ konvergerer til } a\}$  så er  $A \in \mathcal{H}(X)$ .

Definisjon 3.1.1 gir at  $A$  ligger i  $\mathcal{H}(X)$  hvis  $A$  er ikke-tom og kompakt.

*Bevis  $A$  er ikke tom :*

Ved definisjon er  $a_n \in \mathcal{S}$  slik at det holder å undersøke en av følgene. Ved Lemma 3.2.3 vil følgen også ligge i det kompakte settet  $C$  og dermed ha en delfølge  $\{a_{n_k}\}$  som også konvergerer til et punkt  $a \in X$ . Siden  $\mathcal{S}$  er mengden av alle følger  $\{x_n\}$  slik at  $x_n \in B_n$  for  $n \in \mathbb{N}$  vil også alle delfølger av  $\{a_n\}$  være inneholdt i  $\mathcal{S}$  og  $\{a_{n_k}\} \in \mathcal{S}$ . Hvis delfølgen  $\{a_{n_k}\}$  ligger i  $\mathcal{S}$  må  $a$  ligge i  $A$  slik at  $A$  er ikke-tom.

*Bevis  $A$  er kompakt:*

Teorem 2.4.4 gir at en delmengde av et komplett metrisk rom er kompakt om den er lukket og totalt begrenset. Siden  $A$  er en delmengde av  $C$ , som tidligere er bevist kompakt, må  $A$  være totalbegrenset. Dermed er det tilstrekkelig å vise at  $A$  er lukket.

Skal  $A$  være lukket må ethvert randpunkt til  $A$  være inneholdt i mengden  $A$ . Vi velger oss et punkt  $b \in \bar{A}$ . Hvis følgen  $\{b_m\} \in \mathcal{S}$  konvergerer mot  $b$ , vil  $b \in A$  være slik at ethvert randpunkt  $a$  til  $A$  er inneholdt i mengden  $A$ . Kan dette vises vil  $A$  være lukket og dermed kompakt. Siden  $a \in A$  eksisterer det en følge  $\{a_n\} \in \mathcal{S}$  som konvergerer mot  $a$ . Videre må også  $a \in A$  for enhver  $m \in \mathbb{N}$  nærme seg randpunktet. Velger derfor  $d(a, b) \leq \frac{1}{2^{m+1}}$ . Siden  $a$  er inneholdt i  $A$ , eksisterer en følge  $\{a_n\} \in \mathcal{S}$  som konvergerer mot  $a$  slik at det eksisterer en  $b_m \in B_m$  der  $d(a, b_m) \leq \frac{1}{2^{m+1}}$ . Siden avstanden  $d(a, b_m)$  blir mindre og mindre for økende  $m$ , vil  $a_n$  nærme seg  $B_m$  for en tilstrekkelig stor  $n$ . Trekantulikheten gir  $d(b, b_m) \leq d(b, a) + d(a, b_m) < \frac{1}{2^{m+1}} + \frac{1}{2^{m+1}} = \frac{1}{2^m}$ . Dermed flytter punktet  $b_m$  seg nærmere og nærmere  $b$ , slik at følgen  $\{b_m\} \in \mathcal{S}$  konvergerer mot  $b$ , og dermed er lukket.

### 3.2. Kompletthet av Hausdorff-metrikken

*Bevis*  $A \in \mathcal{H}(X)$ :

$A$  er bevist både kompakt og ikke-tom samt at er en delmengde av det komplette metriske rommet  $(X, d)$ , dermed må  $A \in \mathcal{H}(X)$ .

Til nå er det vist at punkt (1)-(3) stemmer, dermed er det kun punkt (4),  $\lim A_n = A$ , som gjenstår. Kan dette bevises vil derfor følgende setning gjelde:

**Lemma 3.2.5.** *La  $(X, d)$  være et komplett metrisk rom. Da er også  $(\mathcal{H}(X), h)$  et komplett metrisk rom.*

*Bevis*  $\lim A_n = A$ :

For å bevise at Cauchy-følgen  $\{A_n\}$  konvergerer mot  $A$ , må vi bevise at for enhver  $\epsilon > 0$  eksisterer en  $N \in \mathbb{N}$  slik at for alle  $n \in \mathbb{N}$  slik at  $A_n \subset A + \epsilon$  og  $A \subset A_n + \epsilon$ .

Siden  $\{A_n\}$  er en Cauchy-følge vil det for enhver  $\epsilon > 0$  finnes en  $N \in \mathbb{N}$  slik at alle  $n, m \geq N$  gir  $A_n \subset A_m + \epsilon$  og  $A_m \subset A_n + \epsilon$ .

Siden  $h(A_n, A_m) < \epsilon$  for alle  $n, m \geq N$  vil  $A_m \subseteq A_n + \epsilon$  slik at  $B_m = \bigcup_{k \geq m} A_k \subseteq A_n + \epsilon$ . Ved definisjonen av  $a \in A$  er grensen til følgerne gitt ved  $B_m$  og da må også  $a \in A_n + \epsilon$  slik at  $A \subset A_n + \epsilon$ .

Videre må  $A_n \subset A + \epsilon$  også vises.

La  $b$  være et punkt i  $A_n$ , da eksisterer et punkt  $a_m \in A_m$  for  $m \geq N$  slik at  $d(b, a_m) \leq \epsilon$ . Som tidligere vist, har  $a_m$  en delfølge  $a_{m_k} \in \mathcal{S}$  som konvergerer mot et punkt  $a$ . Siden delfølgen er inneholdt i  $S$ , så må  $a$  være inneholdt i  $A$ . Da vil også  $d(b, a) = \lim_{k \rightarrow \infty} d(b, a_{m_k}) \leq \epsilon$  slik at  $A_n \subset A + \epsilon$ .

Dermed for enhver  $\epsilon > 0$  eksisterer en  $N \in \mathbb{N}$  slik at for alle  $n \geq N$  er  $A_n \subset A + \epsilon$  og  $A \subset A_n + \epsilon$  og Cauchy-følgen  $\{A_n\}$  konvergerer mot  $A$ .

Oppsummert gir dette følgende teorem:

**Teorem 3.2.6** ([Bar93, s. 35]). *La  $(X, d)$  være et komplett metrisk rom. Da er også  $\mathcal{H}(X)$  et komplett metrisk rom. Videre, hvis  $\{A_n \in \mathcal{H}(X)\}_{n=1}^{\infty}$  er en Cauchy følge, så er  $A = \lim_{n \rightarrow \infty} A_n \in \mathcal{H}(X)$ .*

## KAPITTEL 4

---

# Selvsimilære systemer

---

Selvsimilaritet er et begrep ofte knyttet til Mandelbrot-mengden, Kochs sneflak og andre utvalgte fraktaler. Selvsimilære systemer er delvis eller helt lik på ulike deler av systemet. Kort fortalt, selvsimilære systemer er stykkevis lik seg selv. De selvsimilære fraktalene som skal studeres er i all hovedsak fikspunkter i  $(\mathcal{H}(x), h)$ .

### 4.1 Fraktaler

Historisk sett er fraktaler et vanskelig begrep å definere, Mandelbrot definerte følgende for fraktalet i 1995:

Et fraktal er en mengde hvor Hausdorff dimensjonen er større enn den topologiske dimensjonen.

Denne definisjonen er kritisert for å ekskludere enkelte mengder, som fremdeles er å regne som fraktaler. Etter dette er det ved flere anledninger forsøkt å få til en allmenn akseptert definisjon av fraktaler, men uten hell. Likevel er det noen egenskaper som kan gjenkjennes i fraktalet. Fraktaler har ofte irregulære detaljer, som igjen ofte er for irregulære til å beskrives av den generelle geometrien. Fraktaler har også ofte en eller annen form for selvsimilaritet.

Noen kjente fraktaler er Mandelbrot-mengden, Cantor mengden og Koch sneflak.

### 4.2 Selvsimilære fraktaler

For videre analyser kommer i all hovedsak de selvsimilære fraktalene til å være mest fremtredende. Dermed er det også naturlig at dette defineres for seg. Selvsimilære fraktaler oppstår når en rekke formbevarende avbildninger utføres. For å enklere kunne forklare og beskrive de selvsimilære fraktalene introduseres IFS (iterated function systems), itererte funksjonssystemer.

**Definisjon 4.2.1** ([Bar93, s. 80]). Et iterert funksjonssystem (IFS) er bestående av et komplett metrisk rom og et endelig sett sammentrekninger  $\omega_n = X \rightarrow X$  med tilhørende kontraksjonsfaktorer  $s_n$ , for  $n = 1, 2, \dots, N$ . Notasjonen for et slik system er  $\{X; \omega_n, n = 1, 2, \dots, N\}$  med kontraksjonsfaktor  $s = \max\{s_n : n = 1, 2, \dots, N\}$ .

Videre kan definisjonen utvides for å også omfatte Hausdorff-metrikken. Med utgangspunkt i en sammentrekning på det metriske rommet, er det mulig å fastslå følgende for  $(\mathcal{H}(X), h)$ :

**Lemma 4.2.2** ([Bar93, s. 79]). *La  $\omega : X \rightarrow X$  være en sammentrekning på det metriske rommet  $(X, d)$  med kontraksjonsfaktor  $s$ . Da vil  $\omega$  avbilde  $\mathcal{H}(X)$  på seg selv. Videre vil  $\omega : \mathcal{H}(X) \rightarrow \mathcal{H}(X)$ , definert ved  $\omega(B) = \{\omega(x) : x \in B\}$  for alle  $B \in \mathcal{H}(X)$ , være en sammentrekning på  $(\mathcal{H}(X), h)$  med kontraksjonsfaktor  $s$ .*

For å vise lemmaet vises det først at for  $B$ , en kompakt delmengde av  $X$ , så må også  $\omega(B)$  være en kompakt delmengde av  $X$ .

*Bevis  $\omega : \mathcal{H}(X) \rightarrow \mathcal{H}(X)$  når  $\omega : X \rightarrow X$  :*

Anta at  $\mathcal{S}$  er et ikke-tomt kompakt delmengde av  $(X, h)$  slik at  $\omega(\mathcal{S}) = \{\omega(x) : x \in \mathcal{S}\}$ . Siden  $\mathcal{S}$  er ikke-tomt, vil også  $\omega(\mathcal{S})$  være ikke-tomt. Lar  $\{y_n = \omega(x_n)\}$  være en uendelig rekke i  $\omega(\mathcal{S})$  slik at  $\{x_n\}$  også er en uendelig følge av punkt i  $\mathcal{S}$ . Siden  $\mathcal{S}$  er kompakt vil det ved definisjon 2.4.3 eksistere en delfølge av  $\{x_n\}$  som konvergerer mot et punkt  $a \in \mathcal{S}$ . Ved lemma 2.5.3 vil  $\omega$  være kontinuerlig. Dermed vil delmengden  $x_{N_n} \in x_n$  implisere at delmengden  $y_{N_n} \in y_n$  er gitt ved  $y_{N_n} = \omega(x_{N_n})$  der mengden konvergerer mot et punkt  $b = \omega(a) \in \omega(\mathcal{S})$ . Dermed må  $\omega(\mathcal{S})$  ved definisjon 2.4.3 være kompakt. Det er nå vist at  $\omega$  avbilder  $\mathcal{H}(X)$  på seg selv.

*Bevis  $\omega$  er en sammentrekning :*

Ta utgangspunkt i  $B, C \in \mathcal{H}(X)$ , fra før er det vist at  $\omega$  avbilder  $\mathcal{H}(X)$  på seg selv. Dermed vil følgende gjelde:

$$d(\omega(B), \omega(C)) = \max\{\min\{d(\omega(x), \omega(y)) : y \in C\} : x \in B\}$$

Antar videre at det eksisterer en kontraksjonsfaktor  $s$ , slik at

$$d(\omega(B), \omega(C)) = \max\{\min\{d(\omega(x), \omega(y)) : y \in C\} : x \in B\} \leq \max\{\min\{s * d(x, y) : y \in C\} : x \in B\} = s * d(B, C)$$

Tilsvarende kan  $d(\omega(C), \omega(B)) < s * d(C, B)$  slik at:

$$h(\omega(B), \omega(C)) \leq s * d(b, c) \vee d(C, B) \leq s * d(B, C). \text{ Dermed kan } \omega \text{ ved definisjon 2.3.6 kalles en sammentrekning.}$$

Det er også av interesse å kunne si noe om ulike kombinasjoner av sammentrekninger på  $\mathcal{H}(X)$ . Ta utgangspunkt i to sammentrekninger  $\omega_1$  og  $\omega_2$  på  $\mathcal{H}(X)$ , det er da ønskelig å forsøke og kombinere disse.

Velg to delmengder  $B, C \in \mathcal{H}$ , da vil unionen av de to avbildningene  $\omega_1$  og  $\omega_2$  på  $\mathcal{H}(X)$  være gitt ved  $h(\omega_1(B) \cup \omega_2(B), \omega_1(C) \cup \omega_2(C))$ .

Her er  $h$  Hausdorff-metrikken og  $\omega_1, \omega_2$  sammentrekninger slik at

$$h(\omega_1(B) \cup \omega_2(B), \omega_1(C) \cup \omega_2(C)) \leq h(\omega_1(B), \omega_1(C)) \vee h(\omega_2(B), \omega_2(C)) \leq s_1 h(B, C) \vee s_2 h(B, C) \leq sh(B, C).$$

Dermed er det vist at to sammentrekninger kan kombineres og argumentet kan lett utvides til flere sammentrekninger. Dermed har vi vist:

### 4.3. Itererte funksjonssystemer: Iteariv tilfeldighets-algoritme

**Lemma 4.2.3** ([Bar93, s. 80]). La  $(X, d)$  være et metrisk rom og  $\{\omega_n : n = 1, 2, \dots, N\}$  være sammentrekninger på  $(\mathcal{H}(X), h)$ . La kontraksjonsfaktoren for den enkelte avbildning  $\omega_n$  være  $s_n$ . Definer  $W : \mathcal{H}(X) \rightarrow \mathcal{H}(X)$  ved  $W(B) = \omega_1(B) \cup \omega_2(B) \cup \dots \cup \omega_n(B) = \cup_{n=1}^N \omega_n(B)$  for alle  $B \in \mathcal{H}(X)$ . Da er  $W$  en sammentrekning med kontraksjonsfaktor  $s = \max\{s_n : n = 1, 2\}$ .

Oppsummert vil teorien presentert rundt selvsimilære fraktaler, fikspunkt og Hausdorff-metrikken, lede til følgende teorem:

**Teorem 4.2.4** ([Bar93, s. 81]). La  $\{X; \omega_n, n = 1, 2, \dots, N\}$  være et iterert funksjonssystem med kontraksjonsfaktor  $s$ . Da er avbildningen  $W : \mathcal{H}(X) \rightarrow \mathcal{H}(X)$ , gitt ved  $W(B) = \cup_{n=1}^N \omega_n(B), \forall B \in \mathcal{H}(X)$ , en sammentrekning på det komplette metriske rommet  $(\mathcal{H}(X), h)$  med kontraksjonsfaktor  $s$ .

Videre er det unike fikspunktet,  $A \in \mathcal{H}(X)$ , gitt ved

$$A = \lim_{n \rightarrow \infty} W^{\circ n}(B), \forall B \in \mathcal{H}(X),$$

slik at følgende sammenheng gjelder:  $A = W(A) = \cup_{n=1}^N \omega_n(A)$ .

Her skaper fikspunktet  $A \in \mathcal{H}(X)$  et endelig sett av avbildninger på det metriske rommet  $X$ . Dette settet av avbildninger,  $A = W(A) = \cup_{n=1}^N \omega_n(A)$ , skaper et selvsimilært fraktal i  $X$ .

For resten av oppgaven kommer selvsimilære fraktaler til å være synonymt med fikspunktet for itererte funksjonssystemer, altså der noe avbilder på seg selv. Som nevnt i kapittel 4.1 er definisjonen av fraktaler i seg selv et omstridt tema i matematikken, dette gjelder også definisjonen av selvsimilære fraktaler. Dermed vil definisjonen valgt benyttet for resterende av oppgaven også være omstridt, og kanskje i overkant inkluderende.

Et eksempel på at det er en svært inkluderende definisjon vil være et system som avbilder kvadrater på seg selv. For et slik system vil avbildningen av kvadratet skape et fikspunkt, slik at det vil være mulig å skape et iterert funksjonssystem som oppfyller definisjonen. På den annen side vil man kunne argumentere med at et slikt system bestående av kvadrater ikke vil se særlig fraktale ut.

### 4.3 Itererte funksjonssystemer: Iteariv tilfeldighets-algoritme

For å fremstille de selvsimilære fraktalene er det ønskelig med en algoritme som kan implementeres i en kode. Dette gjør det betydelig enklere å grafisk fremstille, såkalt plote, fraktalet. Gjennom en kode er det også betydelig enklere å endre startpunkt, antall iterasjoner osv, for å leke med ulike utgaver av de ulike fraktalene.

I all hovedsak kommer en iteariv tilfeldighets-algoritme til å benyttes når de første selvsimilære fraktalene fremstilles i neste delkapittel. All kode er skrevet i kodespråket Python.



Fra Teorem 4.2.4 vet vi at fikspunktet  $A = \lim_{n \rightarrow \infty} W^{\circ n}(B), \forall B \in \mathcal{H}(X)$  gir et endelig sett med avbildninger som skaper det selvtilnærmede fraktalet. I all hovedsak velger vi å studere fraktaler på formen  $\{\mathbb{R}^2; \omega_n : n = 1, \dots, N\}$ , der avbildningene  $\omega$  kan uttrykkes ved

$$\omega_i \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix}. \quad (4.1)$$

Dette er bedre kjent som affine avbildninger. For  $n$  avbildninger vil dermed det totale bildet være

$$\omega_1 \cup \omega_2 \cup \dots \cup \omega_n. \quad (4.2)$$

For å kunne benytte oss av en iteariv tilfeldighets-algoritme, må hver av avbildningene også tilegnes en sannsynlighet. Siden metoden baserer seg på tilfeldighet, der algoritmen hele tiden vandrer rundt på de ulike avbildningene, er det nærliggende å bestemme sannsynlighetene ut fra størrelsen på de ulike avbildningene. Dette for å strebe etter at algoritmen skal besøke områder med lik størrelse, omtrent like ofte.

For avbildningene  $\omega_i$  (Se ligning 4.1) så vil determinanten  $a_i d_i - b_i c_i$  være avgjørende for skaleringen til  $\omega_i$ , slik at den da også bør påvirke sannsynligheten. For  $n$  avbildninger, der sannsynlighetene til sammen skal summeres til 1, kan derfor sannsynlighetene tilnærmes ved følgende formel:

$$p_i \approx \frac{|a_i d_i - b_i c_i|}{\sum_{i=1}^N |a_i d_i - b_i c_i|}. \quad (4.3)$$

Dette forutsetter at avbildningene ikke overlapper betraktelig. Dette legger videre grunnlag for følgende definisjon:

**Definisjon 4.3.1** ([Bar93, s. 87–88]). La  $\{X; \omega_1, \omega_2, \dots, \omega_N\}$  være et iterert funksjonssystem, med sannsynligheter  $p_i > 0$  for den enkelte avbildning  $\omega_i, i = 1, 2, \dots, N$  der  $\sum_{i=1}^N p_i = 1$ . Velg et startpunkt  $x_0 \in X$  og  $x_n \in \{\omega_1(x_{n-1}), \omega_2(x_{n-1}), \dots, \omega_n(x_{n-1})\}, n = 1, 2, 3, \dots$  slik at  $x_n = \omega_i(x_{n-1})$  har sannsynlighet  $p_i$  og skaper følgende følge  $\{x_n : n = 0, 1, 2, 3, \dots\} \subset X$ .

## 4.4 Eksempler på IFS for fraktaler

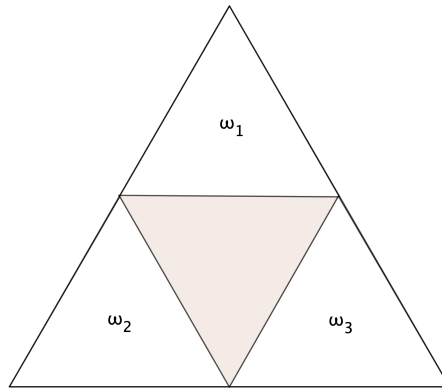
Under kommer noen eksempler på selvtilnærmede fraktaler dannet av itererte funksjonssystemer. Felles for fraktalene er at de dannes ved et iterasjonssystem som avbilder en figur, eller deler av en figur, gjentatte ganger i seg selv.

### Eksempel 1: Sierpinski-trekanten

Sierpinski-trekanten dannes ved en likesidet trekant som avbildes på seg selv. Trekanten består av tre avbildninger  $\omega_1, \omega_2, \omega_3$ , der hver av avbildningene skaper en ny likesidet trekant, men med halv sidelengde av den originale.

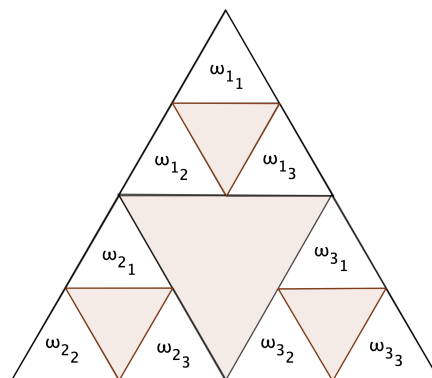
Avbildningene vil skape trekanter av lik størrelse, men plassere dem i ulike hjørner av den opprinnelige trekanten.

Figurene benyttet for å illustrere mål og avbildninger videre er tegnet ved hjelp av GeoGebra.



Figur 4.1: Sierpinski-trekanten etter en iterasjon.

Videre kan avbildningene igjen utføres på de tre nye likesidede trekantene.



Figur 4.2: Sierpinski-trekanten etter to iterasjoner.

Slik kan de tre avbildningene fortsette i  $N$  antall iterasjoner og med det skape mindre og mindre likesidede trekanter. Dette iterasjonssystemet skaper Sierpinski-trekanten, gitt ved fraktalet og fikspunktet  $A = \cup_{n=1}^3 \omega_n(A)$ .

De ulike avbildningene er klart selvsimilære, da hver av trekantene er en skalert og forskjøvet utgave av de andre.

#### 4.4. Eksempler på IFS for fraktaler

Sierpinski-trekanten er fikspunktet for iterert funksjonssystem, som forklart i Teorem 4.2.3, med avbildninger:

$$\omega_1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0,5 & 0 \\ 0 & 0,5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4.4)$$

$$\omega_2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0,5 & 0 \\ 0 & 0,5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix} \quad (4.5)$$

$$\omega_3 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0,5 & 0 \\ 0 & 0,5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{4} \\ \frac{\sqrt{3}}{4} \end{bmatrix}. \quad (4.6)$$

Videre kan dermed sannsynlighetene for hver av avbildningene skrives:

$$p_1 \approx \frac{|0,5 * 0,5 - 0|}{|0,25| + |0,25| + |0,25|} = \frac{0,25}{0,75} = \frac{1}{3} \quad (4.7)$$

$$p_2 \approx \frac{|0,5 * 0,5 - 0|}{|0,25| + |0,25| + |0,25|} = \frac{0,25}{0,75} = \frac{1}{3} \quad (4.8)$$

$$p_3 \approx \frac{|0,5 * 0,5 - 0|}{|0,25| + |0,25| + |0,25|} = \frac{0,25}{0,75} = \frac{1}{3}. \quad (4.9)$$

Sannsynlighetene  $p_1, p_2, p_3$  fremstår i tråd med målet om at like store overflater skal besøkes et likt antall ganger. Dette fordi hver av de tre avbildningene vil skape tre like store trekanter. Dermed bør  $p_1 = p_2 = p_3 = \frac{1}{3}$ , slik som likning 4.7 -4.9 foreslår.

Med denne informasjonen kan vi nå studere de tre avbildningene  $\omega_1, \omega_2, \omega_3$  med sannsynligheter  $p_1, p_2, p_3$  ved hjelp av Python. Følgende kode implementeres:

---

```
import numpy as np
import random as random

def Sierpinski(a,b,N):                                     #startpunkt(a,b) og N iterasjoner
    x_1=np.zeros(N+1)
    x_2= np.zeros(N+1)
    x_1[0] = a
    x_2[0] = b
    for n in range(N):
        z= random.random()                               # tilfeldig variabel med verdi mellom 0 og 1
        if z <= 1/3.:                                    # hvis z <= 1/3
            x_1[n+1] = x_1[n]/2.
            x_2[n+1] = x_2[n]/2.
        elif z <= 2/3. :                                 # hvis 1/3 > z <= 2/3
            x_1[n+1] = x_1[n]/2. + 1/2.
            x_2[n+1] = x_2[n]/2.
        else:                                           # hvis z > 2/3
            x_1[n+1] = x_1[n]/2. + 1/4.
            x_2[n+1] = x_2[n]/2. + np.sqrt(3)/4.
    return x_1,x_2
```

---

#### 4.4. Eksempler på IFS for fraktaler

Ved å kjøre koden for startpunkt i origo, og med 1000 og 10 000 kjøring, illustreres det hvordan sannsynlighet kan benyttes for å plote tilfeldighetsalgoritmen for iterasjon funksjonssystemet for Sierpinski-trekanten.

Følgende kodelinjer benyttes i tillegg til koden ovenfor, for å plote Sierpinski-trekant.

```
import matplotlib.pyplot as plt
x,y = Sierpinski(0,0,1000)
plt.plot(x,y, '.', markersize=0.5)
plt.title('Sierpinski-trekanten', fontsize=15)
plt.show()
```

Tilsvarende kodesnutter er benyttet for å plote resterende av figurene også.

Ved å kjøre koden kom følgende figurer til syne:



Figur 4.3: Plott,  $N = 1000$ .



Figur 4.4: Plott,  $N = 10\ 000$ .

Det observeres at origo er et godt startpunkt for figuren, samt at den må kjøres for et tilstrekkelig antall  $N$ , for at illustrasjonen skal bli klar.

#### Eksempel 2: Koch-kurven

Koch-kurven dannes ved hjelp av fire avbildninger  $\omega_1, \omega_2, \omega_3, \omega_4$  og oppstår ved utgangspunkt i en linje.

Figur 4.5: Koch-kurven før iterasjoner.

Videre skalerer avbildningene linjestykket slik at startpunktet flyttes  $1/3$  inn på den opprinnelige linja, før det videre roteres 60 grader "opp", deretter flyttes den til enden av det nye linjestykket, roteres 60 grader tilbake. Deretter fjernes bunnen i den nyskapede likesidede trekanten.

Deretter gjentas prosessen for de fire nye linjestykkene som har oppstått. På denne måten, skapes en figur som kan benyttes for å studere Koch-kurven. Se figur 4.6 og 4.7



Figur 4.6: Koch-kurven etter en iterasjon.



Figur 4.7: Koch-kurven etter to iterasjon.

Iterasjonssystemets fikspunkt  $A = \cup_{n=1}^4 \omega_n(A)$  skaper med dette det selv-similære fraktalet kjent som Koch-kurven. De ulike avbildningene er også her, i likhet med for Sierpinski-trekanten, klart selv-similære.

Vi studerer avbildningene nærmere. Kurven er altså skapt ved fikspunktet for et iterert funksjonssystem som i teorem 4.2.3.

$$\omega_1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4.10)$$

$$\omega_2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{6} & -\frac{\sqrt{3}}{6} \\ \frac{\sqrt{3}}{6} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{3} \\ 0 \end{bmatrix} \quad (4.11)$$

$$\omega_3 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{6} & \frac{\sqrt{3}}{6} \\ -\frac{\sqrt{3}}{6} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{6} \end{bmatrix} \quad (4.12)$$

$$\omega_4 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{2}{3} \\ 0 \end{bmatrix}. \quad (4.13)$$

Avbildningene svarer videre til følgende sannsynligheter:

$$p_1 \approx \frac{|\frac{1}{3} * \frac{1}{3} - 0|}{|\frac{1}{9}| + |\frac{1}{9}| + |\frac{1}{9}| + |\frac{1}{9}|} = \frac{\frac{1}{9}}{\frac{4}{9}} = \frac{1}{4} \quad (4.14)$$

$$p_2 \approx \frac{|\frac{1}{6} * \frac{1}{6} - \frac{-\sqrt{3}}{6} * \frac{\sqrt{3}}{6}|}{|\frac{1}{9}| + |\frac{1}{9}| + |\frac{1}{9}| + |\frac{1}{9}|} = \frac{\frac{1}{9}}{\frac{4}{9}} = \frac{1}{4} \quad (4.15)$$

$$p_3 \approx \frac{|\frac{1}{6} * \frac{1}{6} - \frac{\sqrt{3}}{6} * \frac{-\sqrt{3}}{6}|}{|\frac{1}{9}| + |\frac{1}{9}| + |\frac{1}{9}| + |\frac{1}{9}|} = \frac{\frac{1}{9}}{\frac{4}{9}} = \frac{1}{4} \quad (4.16)$$

$$p_4 \approx \frac{|\frac{1}{3} * \frac{1}{3} - 0|}{|\frac{1}{9}| + |\frac{1}{9}| + |\frac{1}{9}| + |\frac{1}{9}|} = \frac{\frac{1}{9}}{\frac{4}{9}} = \frac{1}{4}. \quad (4.17)$$

Vi kan plote deler av Koch-kurven ved å benytte avbildningene og sannsynlighetene som basis for følgende kode i Python:

---

```

import numpy as np
import matplotlib.pyplot as plt
import random as random

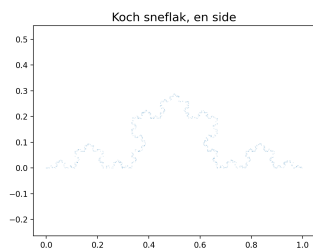
def koch(a,b,N):
    #startpunkt(a,b) og N iterasjoner
    x_1=np.zeros(N+1)
    x_2= np.zeros(N+1)
    x_1[0] = a
    x_2[0] = b
    for n in range(N):
        z= random.random() #tilfeldig variable med verdi mellom 0 og 1
        if z <= 1/4.: #hvis z <= 1/4
            x_1[n+1] = x_1[n]/3.
            x_2[n+1] = x_2[n]/3.
        elif z <= 2/4. : #hvis 1/4 > z <= 2/4
            x_1[n+1] = x_1[n]/6. - (np.sqrt(3)*x_2[n]/6.) + 1/3.
            x_2[n+1] = (np.sqrt(3)*x_1[n]/6.) + x_2[n]/6.
        elif z <= 3/4. : #hvis 2/4 > z <= 3/4
            x_1[n+1] = x_1[n]/6. + (np.sqrt(3)*x_2[n]/6. )+ 1/2.
            x_2[n+1] = -(np.sqrt(3)*x_1[n]/6.) + (x_2[n]/6.) +np.sqrt(3)/6.
        else: #hvis z > 3/4
            x_1[n+1] = (x_1[n]/3.) + 2/3.
            x_2[n+1] = x_2[n]/3.

    return x_1,x_2

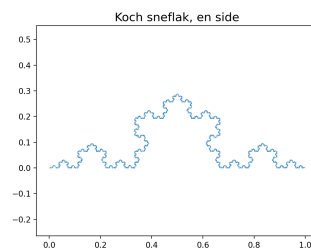
```

---

Ved å kjøre koden og plote for henholdsvis 1000 og 10 000 iterasjoner, kan vi også illustrere Koch-kurven ved hjelp av en tilfeldighets-algoritme. Startpunkt er valgt i origo. Se figur 4.8 og 4.9 for illustrasjon av kurven.



Figur 4.8: Plott, N = 1000.



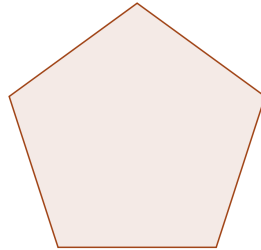
Figur 4.9: Plott, N = 10 000.

### Eksempel 3: Sierpinski femkant

Akkurat som ved Sierpinski-trekanten, dannes Sierpinski femkant av en figur som gjentatt avbilder på seg selv. Femkanten dannes av fem avbildninger, der hver av avbildningene skaper en ny femkant i hvert sitt hjørne av den originale femkanten.

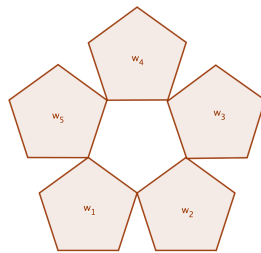


Utgangspunktet for kurven er en regulær femkant, illustrert i figur 4.10.



Figur 4.10: Sierpinskis femkant før iterasjoner.

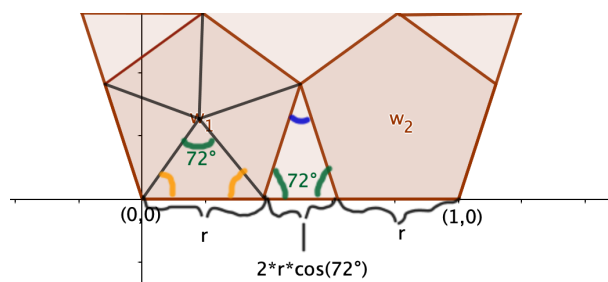
Deretter vil de fem avbildningene skalere femkanten ned i fem nye regulære femkanter i hvert sitt hjørne av den originale femkanten, før den originale femkanten fjernes. Slik som i figur 4.11.



Figur 4.11: Sierpinskis femkant , 1 iterasjon.

Der avbildningene for  $\omega_n$  frem til nå har vært mer åpenbare, vil det nå kreve noe mer regning å finne disse. Ta utgangspunkt i en regulær femkant med side lengde 1, og et startpunkt i origo. Det er tydelig at det er fem avbildninger for systemet (en for hvert hjørne).

Alle avbildningene skalerer størrelsen av den nye femkanten. Dermed kan en naturlig start for å sette opp algoritmen være å finne skaleringsfaktoren. For å finne skaleringen tas det utgangspunkt i nederste del av figuren. Anta at den nye lengden av den skalerte femkanten er  $r$ , da vil enkel trigonometri gi følgende vinkler og avstander:



Figur 4.12: Sierpinskis femkant, lengder og vinkler.

Siden lengden av den originale femkanten er 1, vil

$$r + r + 2r * \cos(72^\circ) = 1 \quad (4.18)$$

$$r(2 + 2\cos(72 * \frac{\pi}{180})) = 1 \quad (4.19)$$

$$r = \frac{1}{2 + 2\cos(72 * \frac{\pi}{180})} \quad (4.20)$$

Altså vil lengden av den nye skalerte femkanten være gitt ved

$$r = \frac{1}{(2+2\cos(72*\frac{\pi}{180}))} \approx 0,382.$$

I tillegg til å skalere, vil noen av avbildningene forskyve femkanten.

*Utregning for algoritmen til  $\omega_1$ :*

Den første avbildningen  $\omega_1$  vil kun skalere femkanten og ikke forskyve den, dermed vil algoritmen kunne se ut som følgende:

$$\omega_1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \quad (4.21)$$

For å skape de resterende avbildningene må femkanten også forskyves. Dette gjøres ved hjelp av vektorer, som illustrert i figur 4.13 - 4.16. For å finne forskyvningsvektoren benyttes cosinus for å finne lengden av hver enkelt vektor i  $x$  retning, og sinus for å finne lengden av hver enkelt vektor i  $y$ -retning.

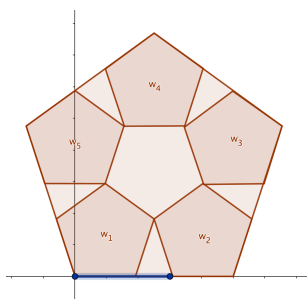
Deretter kombineres disse sammen, og tilnærmes for å kunne gi forskyvningsvektoren.

*Utregning for algoritmen til  $\omega_2$ :*

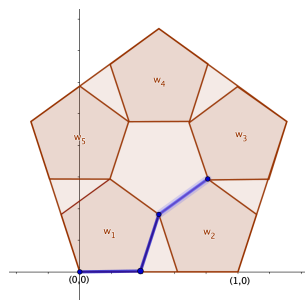
Når algoritmen for  $\omega_2$  settes opp, er det ønskelig med en avbildning av startpunktet (origo) og på tilsvarende hjørne, bare i den nye regulære og skalerte femkanten skapt av  $\omega_2$ . Vektoren benyttet i avbildningen er uthevet i blått i figur 4.13.

Vektoren i figur 4.13 forskyver origo kun i  $x$ -retning. Punktet forskyves med en lengde  $r + 2r * \cos(72 * \frac{\pi}{180}) \approx 0,618$ , se figur 4.12 for lengde av sidene.

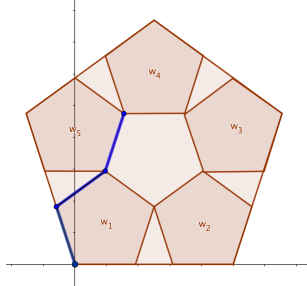
Dermed vil origo forskyves med  $\begin{bmatrix} 0,618 \\ 0 \end{bmatrix}$ .



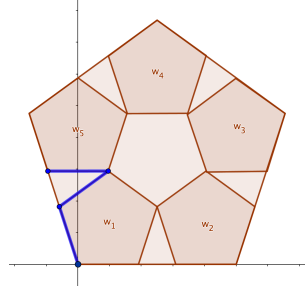
Figur 4.13: Sierpinski's femkant, avbildning  $\omega_2$ .



Figur 4.14: Sierpinski's femkant, avbildning  $\omega_3$ .



Figur 4.15: Sierpinski's femkant, avbildning  $\omega_4$ .



Figur 4.16: Sierpinski's femkant, avbildning  $\omega_5$ .

Siden alle avbildningene også skaleres vil  $\omega_2$  kunne se ut som følgende:

$$\omega_2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0,618 \\ 0 \end{bmatrix}. \quad (4.22)$$

*Utrekning for algoritmen til  $\omega_3$ :*

$\omega_3$  konstrueres på lik måte som  $\omega_2$ , men nå ved hjelp av tre vektorer. Figur 4.14 viser vektorene. I motsetning til  $\omega_2$  vil det nå være en forskyvning både i  $x$  og  $y$  retning.

Den første vektoren starter i origo og beveger seg med en lengde  $r$  langs  $x$ -aksen, det betyr at punktet vil forskyves med  $\begin{bmatrix} r \\ 0 \end{bmatrix}$ .

Neste vektor forskyver videre punktet både i  $x$ - og  $y$ -retning. Ved å benytte oss av cosinus og sinus setningen og den kjente lengden  $r$  samt vinkelen kan vi konstruere en rettvinklet trekant med hypotenus  $r$ . Dette gir oss hvor mye punktet forskyves i hver retning. Dermed forskyves punktet med  $\begin{bmatrix} r * \cos(72^\circ) \\ r * \sin(72^\circ) \end{bmatrix}$ .

Den siste forskyvningen vil forflytte punktet ytterligere både i  $x$  og  $y$  retning. Denne gangen vil vi ta utgangspunkt i den siste vinkelen i den likesidede trekanten mellom femkantene. Siden vinkelsummen i en trekant er  $180^\circ$ , og de to andre vinklene er  $72^\circ$  vil den resterende vinkelen være  $36^\circ$ .

#### 4.4. Eksempler på IFS for fraktaler

---

Forskyvningen skapes på tilsvarende måte og gir med dette oss  $\begin{bmatrix} r * \cos(36^\circ) \\ r * \sin(36^\circ) \end{bmatrix}$ .

Ved å kombinere alle forskyvningene, vil forskyvningen som skal benyttes i  $\omega_3$  fremkomme:

$$\begin{bmatrix} r \\ 0 \end{bmatrix} + \begin{bmatrix} r * \cos(72^\circ) \\ r * \sin(72^\circ) \end{bmatrix} + \begin{bmatrix} r * \cos(36^\circ) \\ r * \sin(36^\circ) \end{bmatrix} \approx \begin{bmatrix} 0,809 \\ 0,588 \end{bmatrix}. \quad (4.23)$$

Dermed vil  $\omega_3$  kunne se ut som følgende:

$$\omega_3 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0,809 \\ 0,588 \end{bmatrix}. \quad (4.24)$$

På tilsvarende måte kan vi konstruere de to resterende avbildningene  $\omega_4$  og  $\omega_5$ . Se figur 4.15 og 4.16 for alternative ruter for forskyvningene.

Dermed kan avbildningene for Sierpinskis femkant være gitt ved følgende:

$$\omega_1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4.25)$$

$$\omega_2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0,618 \\ 0 \end{bmatrix} \quad (4.26)$$

$$\omega_3 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0,809 \\ 0,588 \end{bmatrix} \quad (4.27)$$

$$\omega_4 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0,309 \\ 0,951 \end{bmatrix} \quad (4.28)$$

$$\omega_5 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -0,191 \\ 0,58 \end{bmatrix}. \quad (4.29)$$

Videre kan vi finne sannsynligheten for de ulike avbildningene på tilsvarende måte som for Sierpinski-trekanten og Koch-kurven. Da vil avbildningene tilsvare en uniform sannsynlighet  $p_1, \dots, p_5 = \frac{1}{5}$ .

Avbildningene og sannsynligheten kan videre benyttes for å konstruere kurven ved hjelp av følgende Python program:

---

```
import numpy as np
import matplotlib.pyplot as plt
import random as random

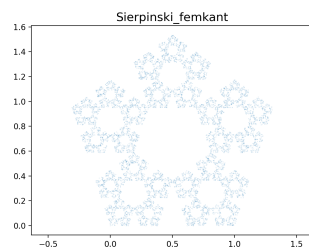
def Sirp_femkant(a,b,N):
    x=np.zeros(N+1)
    y= np.zeros(N+1)
    x[0] = a
    y[0] = b
    s = 1./(2+2*np.cos(72*np.pi/180)) #skaleringsfaktor for femkanten
```

#### 4.4. Eksempler på IFS for fraktaler

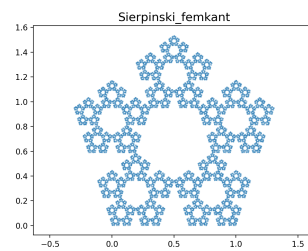
```
for n in range(N):
    z= random.random()
    if z <= 1/5.:
        x[n+1] = x[n]*s
        y[n+1] = y[n]*s
    elif z <= 2/5. :
        x[n+1] = x[n]*s + 0.618
        y[n+1] = y[n]*s
    elif z <= 3/5. :
        x[n+1] = x[n]*s + 0.809
        y[n+1] = y[n]*s + 0.588
    elif z <= 4/5. :
        x[n+1] = x[n]*s + 0.309
        y[n+1] = y[n]*s + 0.951
    else:
        x[n+1] = x[n]*s - 0.191
        y[n+1] = y[n]*s + 0.588

return x,y
```

Gjennom å kjøre koden for  $N$  antall iterasjoner kommer Sierpinski femkanten til syne. Velger startpunkt i origo og deretter henholdsvis 10 000 og 100 000 iterasjoner.



Figur 4.17: Plott,  $N = 10\ 000$ .



Figur 4.18: Plott,  $N = 100\ 000$ .

Det observeres at et tilstrekkelig antall iterasjoner  $N$  må gjennomføres for å få et klart bilde.

## KAPITTEL 5

---

# Flatefyllende kurver

---

Flatefyllende kurver var sent 1800-tall og tidlig på 1900-tallet et av matematikkens store spørsmål. I 1878 oppdaget George Cantor blant annet at et intervall  $[1, 0]$  kunne avbildes bijektivt på et kvadrat  $[1, 0]^2$ . Neste steg nå ville bli å avgjøre om en slik kurve kunne være kontinuert. Kravet om bijektivitet ble droppet og det nye spørsmålet ble som følgende:

Er det mulig å konstruere en kurve som går gjennom alle punkt i et to-dimensjonalt område med positivt areal? ([Sag94, s. 1])

### 5.1 Flatefyllende kurver

For å forklare flatefyllende kurver, samt besvare spørsmålet over, må begrepet defineres. Flatefyllende kurver er sådan en god betegnelse på fenomenet, da det er akkurat det det er; En kurve som fyller en flate. Men hva menes med flate og hva menes med kurve? For kapitlet defineres følgende:

**Definisjon 5.1.1** ([Sag94, s. 4]). Hvis avbildningen  $f : \mathcal{I} \rightarrow \mathbb{R}^n$  er kontinuert, kalles avbildningen  $\omega(\mathcal{I})$  en kurve.

$\mathcal{I}$  vil her betegne intervallet  $[1, 0]$ , slik at kurvens startpunkt er  $\omega(0)$  og kurvens sluttspunkt er  $\omega(1)$ . Videre vil kapittel 5 kun ta for seg kurver som fyller et kvadrat, slik at flaten vår vil være arealet av et kvadrat gitt ved  $\mathcal{Q} = [1, 0]^2$ .

Når det nå er definert hva en kurve er, og hvilken flate vi arbeider i, kan det defineres hva det vil si at noe er en flatefyllende kurve.

**Definisjon 5.1.2** ([Sag94, s. 5]). Hvis avbildningen  $f : \mathcal{I} \rightarrow \mathbb{R}^n$  er kontinuert,  $n \geq 2$  og avbildningen gir et positivt areal. Så er avbildningen  $\omega(\mathcal{I})$  en flatefyllende kurve.

Det er flere matematikere som har konstruert egne varianter av flatefyllende kurver og det er ønskelig å studere noen av dem. På lik linje som for de selvsimilære fraktalene oppstår de flatefyllende kurvene når en rekke formbevarende avbildninger utføres. Dermed kan flatefyllende kurver tolkes som selvsimilære fraktaler i seg selv.

For å analysere, beskrive og tolke selvsimilære fraktaler i kapittel 4.4 ble det benyttet IFS. For å videre analysere de flatefyllende kurvene, er det derfor naturlig å fortsette med å benytte seg av dette. Der det i kapittel 4.4 ble



## 5.2. Itererte funksjonssystemer: Deterministisk algoritme

benyttet tilfeldighets-algoritme, vil denne nå kunne skape en del kaos i plottet vårt. Dette fordi algoritmen kun avbilder punktvis, og ikke tar hensyn til rekkefølge. For å kunne benytte IFS for å beskrive dette må det dermed først gis en ny måte å beskrive fraktaler ved hjelp av IFS på.

### 5.2 Itererte funksjonssystemer: Deterministisk algoritme

Slik som i kapittel 4.3, der vi så på itererte funksjonssystemer på formen

$$\omega_i \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix} \quad (5.1)$$

skal også dette kapitlet studere tilsvarende (affine) avbildninger. I kapittel 4.3 ble disse uttrykt ved hjelp av en tilfeldighets-algoritme som tok utgangspunkt i et startpunkt, og dermed avbildet det  $n$  antall ganger i ulike retninger, basert på et tilfeldig tall.

Nå er det ønskelig å ta utgangspunkt i et kompakt sett for å så, stegvis nærme seg fikspunktet ved å øke antall iterasjoner for  $A_{n+1} = \cup_{j=1}^n \omega_j(A_n)$ . Dette skaper følgen  $\{A_n\}$  som ved teorem 4.2.4 vil konvergere mot et unikt fikspunkt. Vi kan systematisere dataene eksempelvis gjennom en tabell.

$\omega$	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>
1	$a_1$	$b_1$	$c_1$	$d_1$	$e_1$	$f_1$
2	$a_2$	$b_2$	$c_2$	$d_2$	$e_2$	$f_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n$	$a_n$	$b_n$	$c_n$	$d_n$	$e_n$	$f_n$

Tabell 5.1: Tabellen viser hvordan vi kan systematisere data for et IFS.

### Sierpinski-trekanten ved deterministisk algoritme

Sierpinski-trekanten ble diskutert med utgangspunkt i tilfeldighets-algoritmen i kapittel 4.4. Med utgangspunkt i samme tankegang i konstruksjonen av avbildningene får vi følgende tabell for IFS til Sierpinski-trekanten.

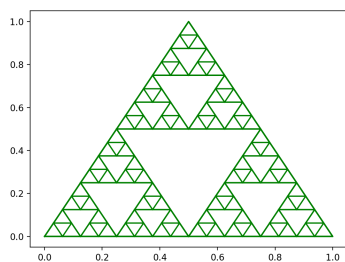
$\omega$	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>
1	0,5	0	0	0,5	0	0
2	0,5	0	0	0,5	0,5	0
3	0,5	0	0	0,5	0,25	0,5

Tabell 5.2: Tabellen viser hvordan vi kan systematisere data for det IFS til Sierpinski-trekant.

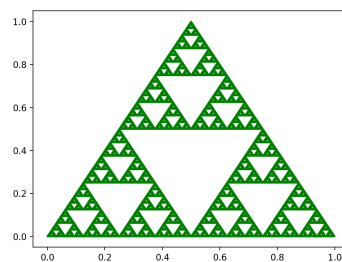
For å nå programmere illustrasjonen for Sierpinski-trekanten benyttes en rekursiv programmeringsmetode, altså en metode som kaller på seg selv.

## 5.2. Itererte funksjonssystemer: Deterministisk algoritme

Først legges tabellen i programmet, før startfiguren bestemmes. I dette tilfellet er startfiguren valgt til å være en likesidet trekant med hjørner i punktene  $(0, 0)$ ,  $(1, 0)$ ,  $(\frac{1}{2}, 1)$ . Videre utføres avbildningene  $\omega_j$  slik at sluttproduktet  $A$ , blir gitt ved unionen av alle avbildningene. Python koden i sin helhet finnes i tilleggskapittel A.1, under delkapitlet *deterministisk, rekursiv metode*. Ved å kjøre koden for henholdsvis  $N = 5$  og  $N = 10$  iterasjoner, får vi følgende plott (se figur 5.1 og 5.2):



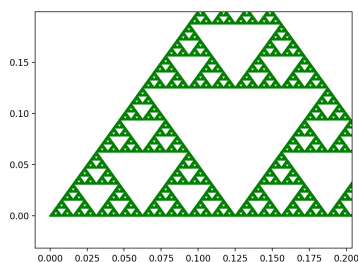
Figur 5.1: Figuren viser resultat av deterministisk kode for Sierpinski-trekant.  $N=5$ .



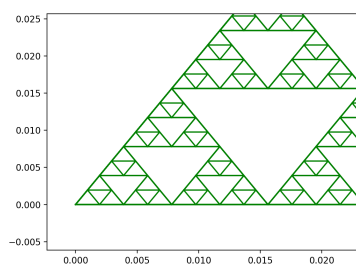
Figur 5.2: Figuren viser resultat av deterministisk kode for Sierpinski-trekant.  $N=10$ .

Sammenligner vi figurene 5.1 og 5.2 med figur 4.3 og 4.4 observeres det at plottet for den deterministiske koden fremstår mer detaljert enn plottet fra tilfældighetsalgoritmen. Dette blir særlig synlig for lave  $N$  verdier. Det fremstår også en tettere sammenheng mellom antall iterasjoner  $N$ , og forventet figur ut fra skissene gitt i figur 4.1 og 4.2.

Avbildningen fremstår også mer detaljert når vi forstørrer delen av figuren, slik at det selvsimilære aspektet ved Sierpinski-trekanten kommer enda bedre til syne.



Figur 5.3: Figuren viser resultat av deterministisk kode for Sierpinski-trekant, her forstørret ca 5-ganger.



Figur 5.4: Figuren viser resultat av deterministisk kode for Sierpinski-trekant, her forstørret med ca.100 ganger.

Forstørringene flytter seg tettere inn på trekantene med hjørnet i origo. Figur 5.3 og 5.4 viser tydelig at det for 5 ganger forstørring, fremdeles er vanskelig å se forskjell på hovedfiguren og forstørringen.

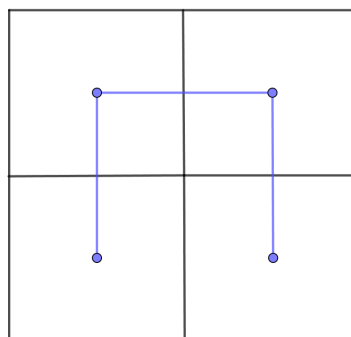
### 5.3 Hilbert's flatefyllede kurve

Matematikeren David Hilbert (1862 - 1943) var blant de første som oppdaget et iterasjonssystem for konstruksjon av flatefyllede kurver. Systemet tar utgangspunkt i et kvadrat  $Q$  og et linjestykke  $\mathcal{I}$ . Systemets utgangspunkt er illustrert i figur 5.5. Videre deles kvadratet opp i fire kongruente deler og linjestykket avbildes mellom disse (se 5.6).

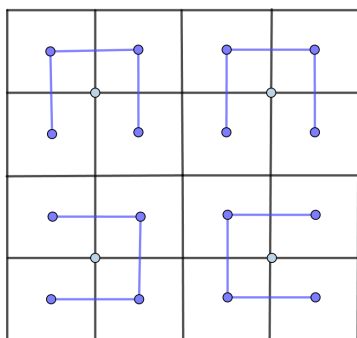
Kurven konstrueres videre ved å dele hver av de nye kvadratene inn i fire nye kongruente deler, for å igjen avbilde linjestykkene på de nye delene (se figur 5.7). Legg merke til at de to nederste linjestykke-figurene er rotert med 90 grader. Deretter kobles delene sammen som illustrert i figur 5.8.



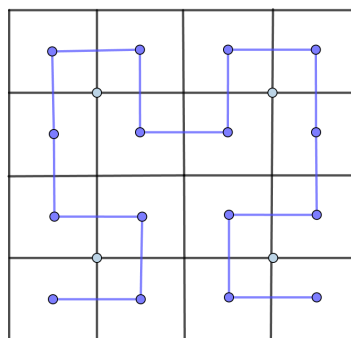
Figur 5.5: Utgangspunkt Hilberts kurve.



Figur 5.6: Generering av Hilberts kurve,  $n=0$ .



Figur 5.7: Hilberts kurve,  $n=1$ , før sammenkobling.



Figur 5.8: Generering av Hilberts kurve,  $n=1$ .

Tanken bak iterasjonssystemet er bygget på følgende:  
Hvis intervallet  $\mathcal{I}$  kan avbildes kontinuerlig på et kvadrat  $Q$ , kan vi dele  $\mathcal{I}$  inn i 4 kongruente delintervaller og  $Q$  i fire kongruente del-kvadrater der hvert av intervallene kan avbildes kontinuerlig på et av del-kvadratene.

Hilbert kurven gjennomløper hver av de ulike delene med en jevn fart og i løpet av intervallet  $[0, 1]$ . Dette vil si at for lik tid  $t$  vil kurven for 5.6, 5.8 og kurver med høyere  $n$ , befinne seg på nesten samme plass på kurven.

Denne prosessen kan igjen gjentas for mindre og mindre intervaller og delkvadrater. Hilbert kurven oppstår for et uendelig antall iterasjoner, og vil på dette punktet kunne kategoriseres som flatefyllede. Med andre ord finner vi Hilberts flatefyllede kurve for  $r(t) = \lim_{n \rightarrow \infty} r_n(t)$ .

Sagan (Sag94) definerer i sin bok Hilbert kurven slik:

**Definisjon 5.3.1** ([Sag94, s. 11]). Enhver  $t \in \mathcal{I}$  er unikt bestemt av en følge av lukkede intervaller der lengden av intervallene går mot null. Til hver av følgene tilhører en unik følge av lukkede kvadrater, der diagonalene går mot et unikt punkt i  $\mathcal{Q}$ . avbindningen  $f_h(t)$ . Da kaller vi  $f_s h^*(\mathcal{I})$  for Hilbert kurven.

Det er ønskelig å vise at Hilbert kurven er en flatefyllede kurve. Skal dette være tilfellet må Hilbert kurven tilsvare kvadratet, altså må  $f_s h^*(\mathcal{I}) = \mathcal{Q}$ .

**Setning 5.3.2** ([Sag94, s. 12]).  $f_s h^*(\mathcal{I}) = \mathcal{Q}$  slik at Hilbert kurven er en flatefyllede kurve

På samme måte som i kapittel 4, kan de flatefyllede kurvene beskrives ved hjelp av itererte funksjonssystemer. Det kan også argumenteres for at iterasjonene som skaper Hilbert kurven er selvsimilære fraktal i seg selv.

For å konstruere den flatefyllede kurven, er det ønskelig med en funksjon  $f : [0, 1] \rightarrow \mathbb{R}^2$  slik at  $f([0, 1]) = A$ , der  $A$  er kompakt delmengde av  $\mathbb{R}^2$ . For Hilberts flatefyllede kurve vil  $A$  representere kvadratet  $\mathcal{Q}$ , som beskrevet over.

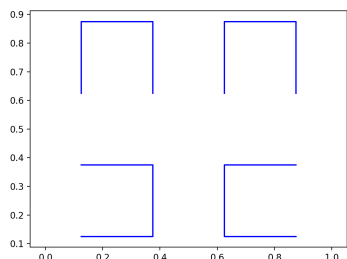
For å beskrive Hilberts flatefyllede kurve ser vi på systemet  $\{X; \omega_n, n = 1, 2, \dots, 4\}$  der hver av avbildningene er similituder med kontraksjonsfaktor 0.5, dette da avbildningene halveres for hver iterasjon (se figur 5.2 og 5.3). Med samme fremgangsmåte som for de selvsimilære fraktalene fra kapittel 4.4, gir Hilbert kurven følgende tabell for det IFS:

$\omega$	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>
1	0	0,5	0,5	0	0	0
2	0,5	0	0	0,5	0	0,5
3	0,5	0	0	0,5	0,5	0,5
4	0	-0,5	-0,5	0	1	0,5

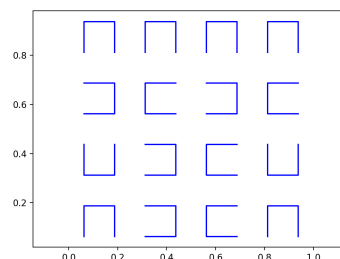
Tabell 5.3: Tabellen viser hvordan vi kan systematisere data for det IFS til Hilbert kurven.

Denne tabellen, sammen med formel 4.1 gir utgangspunkt for en Python-kode som plotter kurven deterministisk for en gitt  $N$ . Koden i sin helhet finner du i tilleggskapittel A.4. Kjører vi koden for henholdsvis  $N = 1$  og  $N = 2$  får vi følgende.

### 5.3. Hilbert's flatefyllende kurve



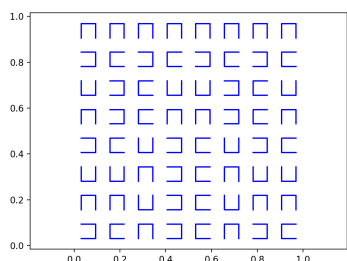
Figur 5.9: Figuren viser Hilberts kurve etter 1 iterasjon.



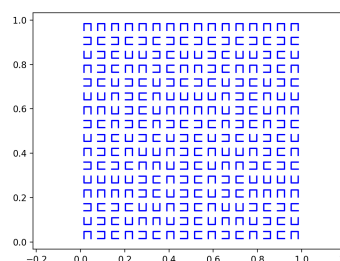
Figur 5.10: Figuren viser Hilberts kurve etter 2 iterasjoner.

Plottet i 5.9 samsvarer med skissen i 5.7, men mangler sammenkoblingene som gjør den til en kurve, slik som illustrert i 5.8.

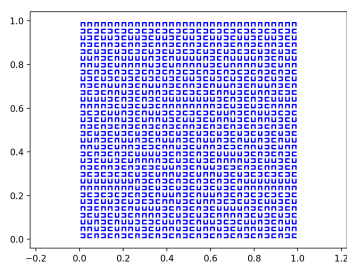
Ved å kjøre koden for  $n = 3$ ,  $n = 4$ ,  $n = 5$  og  $n = 6$  også, kommer det til syne at kurven skapt av koden dekker stadig mer av kvadratets overflate.



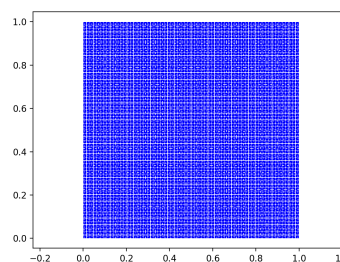
Figur 5.11: Figuren viser Hilberts kurve etter 3 iterasjoner.



Figur 5.12: Figuren viser Hilberts kurve etter 4 iterasjoner.



Figur 5.13: Figuren viser Hilberts kurve etter 5 iterasjoner.



Figur 5.14: Figuren viser Hilberts kurve etter 6 iterasjoner.

Allerede for  $n=6$  er store deler av kvadratets overflate dekket.

Med utviklingen fra figur 5.9 (1 iterasjon) til figur 5.14 (6 iterasjoner) ser vi at et større og større areal av kvadratet  $[0, 1]^2$  er dekket, og således virker til å konvergere mot en flatefyllende kurve.

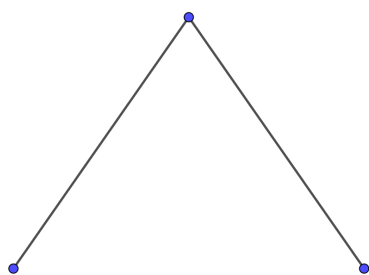
## 5.4 Flatefyllende kurve, ett eksempel

Legg merke til at tabellen for denne flatefyllende kurven er tilsvarende som for Hilberts kurve. Dette da de samme avbildningene  $\omega$  benyttes her, men med et ulikt utgangspunkt.

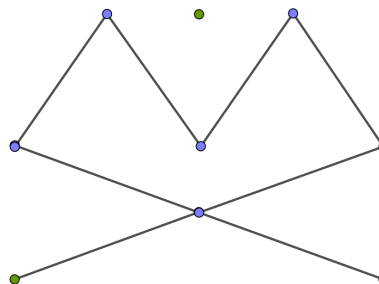
$\omega$	a	b	c	d	e	f
1	0	0,5	0,5	0	0	0
2	0,5	0	0	0,5	0	0,5
3	0,5	0	0	0,5	0,5	0,5
4	0	-0,5	-0,5	0	1	0,5

Tabell 5.4: Tabellen viser hvordan vi kan systematisere data for det IFS til en flatefyllende kurve.

Der det for Hilberts flatefyllende kurve ble tatt utgangspunkt i et slags kvadrat (se figur 5.6) vil utgangspunktet for kurven nå se ut som en slags likebeint trekant (se figur 5.15).



Figur 5.15: Utgangspunkt en annen versjon av Hilberts kurve,  $n=0$ .



Figur 5.16: Utgangspunkt en annen versjon av Hilberts kurve,  $n=1$ .

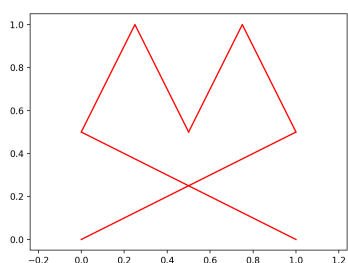
Ved å kjøre iterasjonen vil avbildningene roteres og forflyttes akkurat slik som diskutert i kapittel 5.3. Figur 5.16 viser hvordan denne varianten av Hilberts kurve kan se ut etter en iterasjon, her er punktene etter startfiguren markert i grønt.



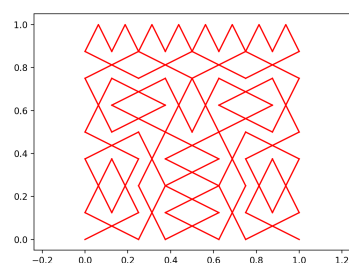
## 5.4. Flatefyllende kurve, ett eksempel

I motsetning til Hilberts originale kurve, vil ikke denne kurven gi noen utfordringer med at bitene må kobles sammen. Dette da trekantene er av en slik størrelse og legger seg i et slikt mønster, at de er sammenhengende av seg selv. Dette gjør at denne kurven er langt enklere å fremstille i sin helhet enn den originale Hilbert kurven.

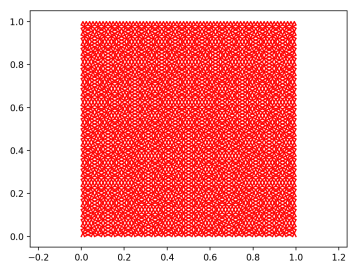
Ved å benytte tilsvarende kode som for den originale Hilbert kurven, men med ett nytt utgangspunkt kan vi derfor fremstille følgende figurer ved hjelp av Python (se figur 5.17-5.20):



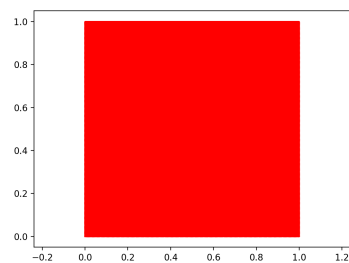
Figur 5.17: Plott av alternativ Hilbert kurve,  $n=1$ .



Figur 5.18: Plott av alternativ Hilbert kurve,  $n=3$ .



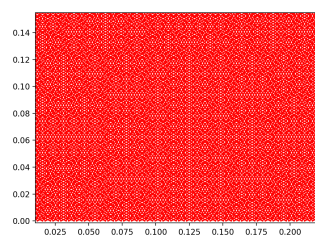
Figur 5.19: Plott av alternativ Hilbert kurve,  $n=6$ .



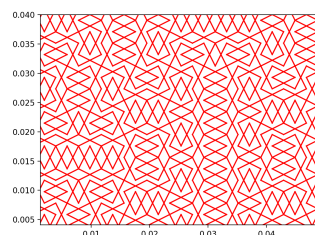
Figur 5.20: Plott av alternativ Hilbert kurve,  $n=9$ .

I likhet med Hilberts originale kurve, ser vi nå også at et større og større areal av kvadratet  $[0, 1]^2$  blir dekket av kurven etterhvert som  $n$  øker. For  $n=6$  har kurven tilsynelatende kun små deler av kvadratet som ikke er dekt av kurven. For  $n=9$  er kvadratet tilsynelatende helt dekt, men forstørret vi deler av kvadratet blir det tydelig at dette ikke er tilfellet. På figur 5.21 og 5.22 er kvadratet forstørret med henholdsvis ca. 5- og 100 ganger.

#### 5.4. Flatefyllende kurve, ett eksempel



Figur 5.21: Plott av alternativ Hilbert kurve forstørret ca. 5 ganger,  $n=9$ .



Figur 5.22: Plott av alternativ Hilbert kurve forstørret ca. 100 ganger,  $n=9$ .

Det blir da tydelig at  $n=9$  ikke er tilstrekkelig for flatefyllende kurve, men funnene i figur 5.17-5.22 støtter opp under teorien om at  $n \rightarrow \infty$  vil gi en flatefyllende kurve.

## KAPITTEL 6

---

# Oppsummering

---

Til slutt gis en kort oppsummering av oppgavens hovedmomenter:

Fraktaler og flatefyllede kurver har vært og er to svært debatterte områder i matematikken. Selv den dag i dag eksisterer det ikke enighet om en formell definisjon av fraktaler, kun noen egenskaper som ofte er knyttet til disse.

Fraktalene beskrives best ved hjelp av rommet  $(\mathcal{H}(X), h)$ , som tillater å knytte delmengder av metriske rom sammen. Her betegner  $\mathcal{H}(X)$  utvalgte delmengder av et metrisk rom  $X$  og  $h$  Hausdorff-metrikken.

Fraktalene diskutert i oppgaven er alle et fikspunkt skapt av unionen av avbildninger  $\omega$ , disse diskuteres i lys av Banach's fikspunktteorem som gir at nettopp et slikt unik fikspunkt eksisterer. Banach's fikspunktteorem forutsetter at vi er i et komplett metrisk rom, med en sammentrekning  $f : X \rightarrow X$ . Dermed bevises det at  $(\mathcal{H}(X), h)$  er komplett, samt at det tas utgangspunkt i en sammentrekning  $\omega : \mathcal{H}(X) \rightarrow \mathcal{H}(X)$  når vi diskuterer fraktalene.

Flatefyllede kurver kan i likhet med fraktalene også beskrives gjennom en union av avbildninger  $\omega$ . Dermed ble det naturlig å analysere disse på tilsvarende måte. Ved å ta utgangspunkt i et affint system kunne kurvene analyseres ved hjelp av IFS og Python. Flatefyllede kurver er kort forklart kurver som fyller et areal, i dette tilfellet arealet  $[0, 1]^2$ . Ved å programmere for et stadig økende antall iterasjoner ble det synliggjort at Hilbert kurven, både med kvadrat og trekant som startfigur, kunne nærme seg en flatefyllede kurve for et uendelig antall iterasjoner.

---

## **Appendikser**

---

# TILLEGG A

---

## Samling av Python koder

---

Tillegg A inneholder Python kodene benyttet gjennom oppgaven. Dette inkluderer både kodene som alt er vist i de respektive kapitlene, og de som det kun er referert til.

### A.1 Python koder for Sierpinski-trekant

Sierpinski-trekanten kan leses mer om i kapittel 4.4, hvor det utledes hvor hvordan man kan fremstille trekanten ved hjelp av tilfeldighets-algoritmen. Eller i kapittel 5.2 hvor den benyttes som eksempel på hvordan man kan programmere selvsimilære fraktaler deterministisk.

#### Programmert med tilfeldighets-algoritme

---

```
import numpy as np
import random as random

def Sierpinski(a,b,N):                                     #startpunkt(a,b) og N iterasjoner
    x_1=np.zeros(N+1)
    x_2= np.zeros(N+1)
    x_1[0] = a
    x_2[0] = b
    for n in range(N):
        z= random.random()                               # tilfeldig variabel med verdi mellom 0 og 1
        if z <= 1/3.:                                   # hvis z <= 1/3
            x_1[n+1] = x_1[n]/2.
            x_2[n+1] = x_2[n]/2.
        elif z <= 2/3. :                                 # hvis 1/3 > z <= 2/3
            x_1[n+1] = x_1[n]/2. + 1/2.
            x_2[n+1] = x_2[n]/2.
        else:                                           # hvis z > 2/3
            x_1[n+1] = x_1[n]/2. + 1/4.
            x_2[n+1] = x_2[n]/2. + np.sqrt(3)/4.
    return x_1,x_2
```

---

## Deterministisk, rekursiv metode

---

```

import numpy as np
import matplotlib.pyplot as plt

#tabell for sierpinski trekantem
a = np.array([0.5, 0.5, 0.5 ])
b = np.array([0,0,0 ])
c = np.array([0,0,0 ])
d = np.array([0.5,0.5,0.5])
e = np.array([0,0.5,0.25 ])
f = np.array([0,0,np.sqrt(3)/4. ])

#start f lge
x_start = np.array([0,1,0.5,0])
y_start = np.array([0,0,1,0])
plt.plot(x_start,y_start,'g')

#rekursiv metode for avbildninger
def w1(x,y, teller, N):
    teller += 1
    if teller == N:
        return

    x_1 = x*a[0] + y*b[0] + e[0]
    y_1 = x*c[0] + y*d[0] + f[0]

    plt.plot(x_1,y_1,'g')

    w1(x_1,y_1, teller,N)
    w2(x_1,y_1, teller, N)
    w3(x_1,y_1, teller, N)

    return

def w2(x,y, teller, N):
    teller += 1
    if teller == N:
        return

    x_2 = x*a[1] + y*b[1] + e[1]
    y_2 = x*c[1] + y*d[1] + f[1]

    plt.plot(x_2,y_2,'g')

    w1(x_2,y_2, teller, N)
    w2(x_2,y_2, teller, N)
    w3(x_2,y_2, teller, N)

    return

def w3(x,y, teller, N):
    teller += 1
    if teller == N:
        return

    x_3 = x*a[2] + y*b[2] + e[2]
    y_3 = x*c[2] + y*d[2] + 1/2.*f[2]
    plt.plot(x_3,y_3,'g')

    w1(x_3,y_3, teller, N)

```

```
w2(x_3,y_3, teller, N)
w3(x_3,y_3, teller, N)

return

N = 5 #antall iterasjoner

w1(x_start,y_start,0, N)
w2(x_start,y_start,0, N)
w3(x_start,y_start,0, N)

plt.show()
```

---

## A.2 Python kode for Koch-kurven

Mer om Koch-kurven, og hvordan den kan fremstilles ved bruk av en tilfeldighets-algoritme, beskrives i kapittel 4.4

---

```
import numpy as np
import matplotlib.pyplot as plt
import random as random

def koch(a,b,N): #startpunkt(a,b) og N iterasjoner
    x_1=np.zeros(N+1)
    x_2= np.zeros(N+1)
    x_1[0] = a
    x_2[0] = b
    for n in range(N):
        z= random.random() #tilfeldig variable med verdi mellom 0 og 1
        if z <= 1/4.: #hvis z <= 1/4
            x_1[n+1] = x_1[n]/3.
            x_2[n+1] = x_2[n]/3.
        elif z <= 2/4. : #hvis 1/4 > z <= 2/4
            x_1[n+1] = x_1[n]/6. - (np.sqrt(3)*x_2[n]/6.) + 1/3.
            x_2[n+1] = (np.sqrt(3)*x_1[n]/6.) + x_2[n]/6.
        elif z <= 3/4. : #hvis 2/4 >z <= 3/4
            x_1[n+1] = x_1[n]/6. + (np.sqrt(3)*x_2[n]/6. )+ 1/2.
            x_2[n+1] = -(np.sqrt(3)*x_1[n]/6.) + (x_2[n]/6.) +np.sqrt(3)/6.
        else: #hvis z > 3/4
            x_1[n+1] = (x_1[n]/3.) + 2/3.
            x_2[n+1] = x_2[n]/3.

    return x_1,x_2
```

---

### A.3 Python kode for Sierpinski's femkant

Sierpinski's femkant fremstilles og beskrives i kapittel 4.4

---

```
import numpy as np
import matplotlib.pyplot as plt
import random as random

def Sirp_femkant(a,b,N):
    x=np.zeros(N+1)
    y= np.zeros(N+1)
    x[0] = a
    y[0] = b
    s = 1./(2+2*np.cos(72*np.pi/180)) #skaleringsfaktor for femkanten
    for n in range(N):
        z= random.random()
        if z <= 1/5.:
            x[n+1] = x[n]*s
            y[n+1] = y[n]*s
        elif z <= 2/5. :
            x[n+1] = x[n]*s + 0.618
            y[n+1] = y[n]*s
        elif z <= 3/5. :
            x[n+1] = x[n]*s + 0.809
            y[n+1] = y[n]*s + 0.588
        elif z <= 4/5. :
            x[n+1] = x[n]*s + 0.309
            y[n+1] = y[n]*s + 0.951
        else:
            x[n+1] = x[n]*s - 0.191
            y[n+1] = y[n]*s + 0.588

    return x,y
```

---

### A.4 Python kode for Hilberts flatefylende kurve

Det kan leses mer om Hilberts flatefylende kurve i kapittel 5.3 og 5.4. Kodene for de to versjonene for Hilberts kurve er nærmest identiske, men med ulikt utgangspunkt.

#### Hilberts originale kurve, usammenhengende

---

```
import numpy as np
import matplotlib.pyplot as plt

#tabell hilbert-kurven
a = np.array([0, 0.5, 0.5, 0 ])
b = np.array([0.5,0,0, (-0.5) ])
c = np.array([0.5,0,0, (-0.5) ])
d = np.array([0,0.5,0.5,0])
e = np.array([0,0,0.5, 1 ])
f = np.array([0,0.5,0.5, 0.5 ])

#Start f lge
x = np.array([0.25,0.25,0.75,0.75])
y = np.array([0.25,0.75,0.75,0.25])

#rekursiv metode for avbildninger
```



#### A.4. Python kode for Hilberts flatefyllende kurve

---

```
def w1(x,y, count, N):
    count += 1

    x_1 = x*a[0] + y*b[0] + e[0]
    y_1 = x*c[0] + y*d[0] + f[0]

    if count == N:
        plt.plot(x_1,y_1, 'b')
        return

    w1(x_1,y_1, count, N)
    w3(x_1,y_1, count, N)
    w4(x_1,y_1, count, N)
    w2(x_1,y_1, count, N)

def w2(x,y, count, N):
    count += 1

    x_2 = x*a[1] + y*b[1] + e[1]
    y_2 = x*c[1] + y*d[1] + f[1]

    if count == N:
        plt.plot(x_2,y_2, 'b')
        return

    w1(x_2,y_2, count, N)
    w3(x_2,y_2, count, N)
    w4(x_2,y_2, count, N)
    w2(x_2,y_2, count, N)
    return

def w3(x,y, count, N):
    count += 1

    x_3 = x*a[2] + y*b[2] + e[2]
    y_3 = x*c[2] + y*d[2] + f[2]

    if count == N:
        plt.plot(x_3,y_3, 'b')
        return

    w2(x_3,y_3, count, N)
    w1(x_3,y_3, count, N)
    w4(x_3,y_3, count, N)
    w3(x_3,y_3, count, N)
    return

def w4(x,y, count, N):
    count += 1

    x_4 = x*a[3] + y*b[3] + e[3]
    y_4 = x*c[3] + y*d[3] + f[3]

    if count == N:
        plt.plot(x_4,y_4, 'b')
        return

    w2(x_4,y_4, count, N)
    w3(x_4,y_4, count, N)
    w1(x_4,y_4, count, N)
    w4(x_4,y_4, count, N)
```

#### A.4. Python kode for Hilberts flatefyllede kurve

---

```
    return

N = 5 #antal iterasjoner
w1(x,y,0,N)
w2(x,y,0,N)
w3(x,y,0, N)
w4(x,y,0,N)

plt.axis('equal') #gir aksene lik lengde.
plt.show()
```

---

#### Alternativ Hilbert kurve, sammenhengende

---

```
import numpy as np
import matplotlib.pyplot as plt

#tabell hilbert-kurven
a = np.array([0, 0.5, 0.5, 0 ])
b = np.array([0.5,0,0, (-0.5) ])
c = np.array([0.5,0,0, (-0.5) ])
d = np.array([0,0.5,0.5,0])
e = np.array([0,0,0.5, 1 ])
f = np.array([0,0.5,0.5, 0.5 ])

#Start f lge
x = np.array([0,0.5,1])
y = np.array([0,1,0])

#rekursiv metode for avbildninger
def w1(x,y,count,N):
    count += 1

    x_1 = x*a[0] + y*b[0] + e[0]
    y_1 = x*c[0] + y*d[0] + f[0]

    if count == N:
        plt.plot(x_1,y_1,'r')
        return

    w1(x_1,y_1, count, N)
    w3(x_1,y_1, count, N)
    w4(x_1,y_1, count, N)
    w2(x_1,y_1, count, N)

def w2(x,y, count, N):
    count += 1

    x_2 = x*a[1] + y*b[1] + e[1]
    y_2 = x*c[1] + y*d[1] + f[1]

    if count == N:
        plt.plot(x_2,y_2, 'r')
        return

    w1(x_2,y_2, count, N)
    w3(x_2,y_2, count, N)
```

#### A.4. Python kode for Hilberts flatefyllede kurve

---

```
w4(x_2,y_2, count, N)
w2(x_2,y_2, count, N)
return

def w3(x,y, count, N):
    count += 1

    x_3 = x*a[2] + y*b[2] + e[2]
    y_3 = x*c[2] + y*d[2] + f[2]

    if count == N:
        plt.plot(x_3,y_3, 'r')
        return

    w2(x_3,y_3,count, N)
    w1(x_3,y_3, count, N)
    w4(x_3,y_3, count, N)
    w3(x_3,y_3, count, N)
    return

def w4(x,y, count, N):
    count += 1

    x_4 = x*a[3] + y*b[3] + e[3]
    y_4 = x*c[3] + y*d[3] + f[3]

    if count == N:
        plt.plot(x_4,y_4, 'r')
        return

    w2(x_4,y_4, count, N)
    w3(x_4,y_4, count, N)
    w1(x_4,y_4, count, N)
    w4(x_4,y_4, count, N)
    return

N = 9 #antal iterasjoner
w1(x,y,0,N)
w2(x,y,0,N)
w3(x,y,0, N)
w4(x,y,0,N)

plt.axis('equal') #gir aksene lik lengde.
plt.show()
```

---

---

## Bibliografi

---

- [Bar93] Barnsley, M. F. *Fractals everywhere*. eng. 2nd ed. Boston: Academic Press, 1993.
- [Gmb22] GmbH, G. *GeoGebra*. Versjon 5.0.679.0-d. 29. mar. 2022.
- [HL15] Hveberg, K. og Lindstrøm, T. *Flervariabel analyse med linær algebra*. Gyldendal Norsk Forlag AS, 2015.
- [Lin12] Lindstrøm, T. *Kalkulus*. Universitetsforlaget As, 2012.
- [Lin17] Lindstrøm, T. *Spaces: An Introduction to Real Analysis*. Providence: American Mathematical Society, 2017.
- [Lin22a] Lindstrøm, T. «Completeness of the Hausdorff metric». unpublished. 2022.
- [Lin22b] Lindstrøm, T. «Sannsynlighetene  $p_i$ ». unpublished. 2022.
- [OR98] O'Connor, J. og Robertson, E. F. *MT MacTutor: Georg Ferdinand Ludwig Philipp Cantor*. 1998. URL: <https://mathshistory.st-andrews.ac.uk/Biographies/Cantor/> (sjekket 30.04.2022).
- [Sag94] Sagan, H. *Space-Filling Curves*. Springer New York, 1994.
- [Tro09] Trochet, H. *MT MacTutor: A History of Fractal Geometry*. 2009. URL: <https://mathshistory.st-andrews.ac.uk/HistTopics/fractals/> (sjekket 28.04.2022).
- [Utd20] Utdanningsdirektoratet. *kompetansemål etter matematikk 1T*. 2020. URL: <https://www.udir.no/lk20/mat09-01/kompetansemaal-og-vurdering/kv42?lang=nob> (sjekket 27.04.2022).