



Studieretningsprojekt – 2019/20

Elev: _____

Fag:

Matematik A

Vejleder:

Område/emne: Fraktaler**Opgaveformulering**

Hvilken betydning har arbejdet med fraktaler i det 20. århundrede haft for udviklingen matematik-faget?

Du skal i denne opgave gennem udvalgte nedslag beskrive, hvordan teorien om fraktaler har udviklet sig i det 20. århundrede.

Du skal her komme ind på

- Tidlige fraktaler som Kochs snefnug og Sierpinskis trekant.
- Gaston Julias arbejde med de såkaldte Julia-mængder.
- Benoit Mandelbrots beskrivelse af den britiske kystlinje ved hjælp af forskellige fraktaler.
- Mitchell Feigenbaums figentræer.

Du skal for hver af disse matematikere give eksempler på, hvordan man kan arbejde eksperimenterende og aksiomatisk-deduktivt med de typer af fraktaler, som de arbejdede med.

Som en del af opgaven skal du forklare, hvordan man gennem programmering kan tegne fraktaler fra hver af perioderne.

Opgavens omfang: 15-25 sider à 2400 tegn inkl. mellemrum – men ekskl. forside, indholdsfortegnelse, fodnoter, litteraturliste og eventuelle bilag.

Denne side skal indsættes som forsiden i besvarelsen
Besvarelsen afleveres i **Netprøver.dk** senest **fredag d. 3. april kl. 15:15**



Resumé

Denne opgave undersøger betydningen af arbejdet med fraktaler for matematikkens udvikling i det 20. århundrede. Fraktaler fra forskellige tidsperioder var med til at udvikle en ny geometrisk teori, fraktalteori. Til at udvikle den nye fraktalteori benyttes den aksiomatiske deduktive metode til at slutte gældende sætninger om fraktalerne: Kochs snefnug, sierpinski-trekanten, Julia-mængder og feigenbaumdiagrammer. De gældende sætninger bryder med den normale, førhen værende og til tider i dag dominerende klassiske geometri, samt vores intuition om funktioner. Samtidig bliver her brugt en mere eksperimentel tilgang som programmering og eksperimentel matematik til at slutte konklusioner om fraktalerne og fraktalteoriens udvikling og dermed også udviklingen af matematik-faget til et mere eksperimenterende fag. Til sidst vurderes fraktalteoriens betydning for matematikkens udvikling i 1900'tallet. Her konkluderes, at arbejdet med fraktalerne har været med til at udvide emnet geometri, samt ændre måden man arbejder i matematikken på ved nu at gå eksperimentelt til værk via programmering og derunder simulering.



Indhold

Indledning.....	3
Fraktaler.....	4
Tidlige fraktaler.....	4
Sierpinski-trekanten	4
Kochs snefnug.....	8
De tidlige fraktalers betydning for fraktalteoriens- og matematikkens udvikling	11
Julia-mængder	11
Hvor lang er den britiske kystlinje?	16
De logistiske fremskrivningsfunktioner	20
Figentræer for andre funktioner	24
Fraktalernes udvikling, og deres betydning for matematikkens udvikling.....	25
Konklusion	27
Litteraturliste	28
Bøger	28
Links.....	28
Bilag	32

Omfang: 49544 anslag

Ca. 20,6 normalsider.



Indledning

Geometrien er en af de ældste matematiske discipliner¹ og er helt uudværlig. Men geometrien har ændret sig igennem årene og specielt efter computerens opfinden, der muliggjorde, at en ny og spændende gren på geometristammen for alvor kunne undersøges. Her refereres til fraktalgeometrien med dens farverige objekter, uendelige snørklede mønstre og gentagende geometriske objekter, som har fascineret mange siden omkring 1980.² Fraktalgeometrien havde og har stadigvæk mysterier, og lige netop derfor er det interessant og at dykke ned i denne kaotiske og dynamiske verden, hvor matematikkens normale metoder nogle gange ikke rækker langt nok til at kunne beskrive den.

Derfor vil denne opgave undersøge betydningen af arbejdet med fraktaler for udviklingen af matematik-faget. Besvarelsen vil begrænse sig til udvalgte nedslag, hvilke er: Kocks snefnug, sierpinski-trekanten, Gaston Julias arbejde med Julia-mængder, og Mitchell Feigenbaums figentræer, som alle er fra det 20. århundrede. Samtidig vil her blive fokuseret på, hvordan man gennem programmering kan tegne de omtalte fraktaler, da dette vil underbygge betydning af fraktalteoriens betydning for matematik-fagets udvikling. I opgaven betragtes regneregler for komplekse tal, matricer samt normale kommandoer til tegning i Maple som værende kendt viden.

For at undersøge betydning af arbejdet med fraktaler for udviklingen af matematik-faget opbygges opgaven historisk kronologisk så det førstnævnte fraktal behandles først. Opgaven er opbygget sådan, da teorien, de forskellige matematikere opstiller, afhænger og bygger videre på de tidligere matematiske slutninger. Undervejs, mens udviklingen beskrives, vil her forklares programmer til at tegne fraktalerne. De fremstillede programmer vil bruges til at forklare den nye eksperimenterende tilgang til matematik-faget. Her er valgt ikke at forklare samt programmere mange fraktaler, da dette vil optage en del plads i opgaven og dermed forhindre opgaven i at gå i dybden med hver af fraktalerne og udvalgt teori, matematikerne arbejdede aksiomatiske deduktivt med. Derfor er her heller ikke valgt at fokusere på flere tidligere fraktaler end Kocks Snefnug og sierpinski-trekanten. Til slut vurderes betydningen af arbejdet med fraktaler for udviklingen af matematik-faget. Hermed arbejdes der både *om* og *i* matematikken, da matematik-fagets udvikling undersøges, og i denne undersøgelse opstilles sætninger, som bevises.

¹ Den store danske, Geometri, 2017

² Den store danske, 2017, Fraktal



Fraktaler

Benoit B. Mandelbrot definerede først begrebet fraktal i 1980'erne³ flere år efter, at de først fraktaler blev beskrevet. Dog er definitionen meget løs, da alle fraktaler ikke behøver at opfylde de samme kriterier for, at blive defineret som et fraktal, hvilket understreger, den kompleksitet fraktaler har. De fælles egenskaber fraktaler deler er, som følger:^{4,5} a) fremkommer ved iteration, b) indeholder detaljer på vilkårligt små skalaer, c) har ikke en heltallig dimension, d) er enten eksakt eller statistisk selvtilnærmende,⁶ e) er ikke differentiable, f) er rekursiv⁷ og g) er for irregulær til at kunne beskrives med klassiske geometriske termer. Her vil i denne opgave skelnes mellem to typer fraktaler. Geometriske fraktaler,⁸ som fremkommer ved at ændre på en geometrisk figur på samme måde igen og igen, og algebraiske fraktaler⁹, som fremkommer ved at iterere en iterationsfunktion. Der findes flere måder at inddele fraktaler på, og yderligere inddelinger vil benyttes i opgaven. Her startes med at definere begrebet iteration.¹⁰

Definition 1

Et iterativt system er en talfølge z_0, z_1, z_2, \dots , kaldet en bane. z_0 er begyndelsesværdien. Her gælder, at hvert element z kun afhænger af det forrige element ved en givet iterationsfunktion f $z_{n+1} = f(z_n)$, hvor $n \geq 0$.

Tidlige fraktaler

For at undersøge fraktalteoriens udvikling og dens betydning for matematikkens udvikling, starter opgaven med to nedslag i de tidligste fraktaler, sierpinski-trekanten og Kochs snefnug. Disse fraktaler medførte nogle nye og spændende konklusioner, som er bemærkelsesværdige ift. til den klassiske geometri, hvilket vil tydeliggøres i de følgende afsnit.

Sierpinski-trekanten

I omkring år 1900 arbejdede den polske matematiker Waclaw Sierpinski med det geometriske

³ Den store danske, 2017, Fraktal

⁴ Jonas Bargum, 2010, Fraktaler generelt.

⁵ Wikipedia, 2019, fraktal

⁶ Betyder at samme mønstre kan findes på mindre skalaer.

⁷ Betyder at den refererer til sig selv

⁸ Jesper Frandsen, 1990, s. 61-66

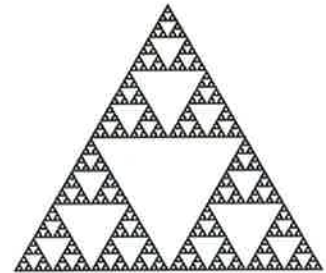
⁹ Forex, Historie om forekomst

¹⁰ Jesper Frandsen, 1990, s. 43



objekt sierpinski-trekanten, som kan ses på reverseret form på figur 1. Dog har mønstret været kendt i århundrede før. For at kunne studere Sierpinskis trekant tegnes den først. Sierpinskis trekant dannes via følgende geometriske iterationer.¹¹

- I. Tegn en ligesidet trekant (0. generation, figur 3)
- II. Del den ligesidede trekant i fire lige store trekanter, og fjern den midterste trekant (1. generation, figur 4)
- III. Del de nye trekanter i fire og fjern den midterste (2. generation)
- IV. Denne itereres til det uendelige (efter n iterationer, kaldes objektet n. generation).



Figur 1: Sierpinski-trekant, 7. generation. Fremkaldt med bilag 1

```
> transform:=proc(T,s,x,y)      #1
return                          #2
[s*T[1][1] + x, s*T[1][2] + y], #3
[s*T[2][1] + x, s*T[2][2] + y], #4
[s*T[3][1] + x, s*T[3][2] + y]]; #5
end:                             #6
> Sierpinski := proc(n)         #7
local L, M, i, k;              #8
L := [[0, 0], [1, 0], [0.5, 1]]; #9
for i to n do M := [];         #10
for k to nops(L) do M := [op(M), #11
transform(L[k], 0.5, 0, 0),    #12
transform(L[k], 0.5, 0.5, 0), #13
transform(L[k], 0.5, 0.25, 0.5)]; #14
end do;                         #15
L := M;                          #16
end do;                           #17
end proc:                          #18
> L := Sierpinski(3);           #19
> plots[polygonplot](L, axes = none, #20
color = black);                 #21
```

Figur 2: Program i Maple til fremstilling af sierpinski-trekanten

Med denne simple fremgangsmåde kan fraktalet programmeres i Maple¹². På figur 2 ses programkoderne, som tilsammen danner sierpinski-trekanten¹³. Programkoderne vil blive gennemgået meget detaljeret, for at de senere programkoder i denne opgave ikke behøves at gennemgås i samme detaljeringsgrad.¹⁴ I programmet er der tilføjet # efterfulgt af et tal. Alt efter et # på en linje "overses" af programmet og har derfor ingen betydning andet end, at det gør det nemmere at referere til enkelte linjer. Når man starter en programmering i Maple, skal man definere, hvad processen skal kaldes, her Sierpinski. Man vil gerne

have programmet til at udføre en proces, så n'te generation tegnes, hvilket skrives i Maple således: " := proc(n)". Herefter skrives, hvad programmet skal gøre, når n oplyses. Til at starte med defineres L, M, I og k til at være lokale operatører, hvilket betyder, at de kan indgå som variable i alle såkaldte for-loops,¹⁵. Her slutes af ved at skrive et semikolon, hvilket betyder, at

¹¹ Nigel Lesmoir-Gordon, Will Rood og Ralph Edney, 2009, s., s. 38-39

¹² Forklaringer på tegnsætning i Maple kan findes i Maple. Hvis, man fx ikke ved hvad "od" gør, skriver man på hvilket som helst linje "?od" og trykker enter. Så popper et vindue frem, hvor Maple forklarer det ønskede. Dette er benyttet.

¹³ Betten, Anton, 2008, s. 80- 85

¹⁴ Alt basal viden om programmering er fundet her: Maplesoft, *Programming Guide*.

¹⁵ Eller i en in-line hvor variablene også være lokale



programmet skal vise udregningen, samt at linjen er færdig. Dette bruges hele vejen igennem programmet. Herefter defineres en liste L, som indeholder lister med punkterne $a(0,0)$, $b(1,0)$ og $c(0.5,1)$. Hvis disse punkter forenes, dannes lige nøjagtigt en ligesidet trekant, hvilket er 0. generation sierpinski-trekanten. Man ønsker at udføre n-antal iterationer på objektet, og derfor skrives der i linje 10 "for i to n do M: = [];". Der skal laves en liste M, som indeholder alle de nye trekanters hjørnepunkter efter iterationerne. Den er i starten tom, men fyldes undervejs, som programmet kører. Derfor stod der "M: = [];" på linje 10. I den næste linje, 11, står der "for k to nops(L) do...", hvilket betyder, at for hver gang 1 går op i antallet af trekanter i L, skal det, som står efterfølgende, udføres. Nops(L) angiver antallet af enheder i L, hvilket er antal trekanter.

Til det efterfølgende har der været brug for et program til, hvor de nye trekanter laves. Programmet, "transform" fungerer på den måde, at man skal opgive en liste T med trekanter, som skal skaleres og forskydes ad første- og andenaksen. Dette gøres ved at oplyse en skaleringsfaktor s, og hvor langt den skal rykkes hen ad førsteaksen ("x") og op ad andenaksen ("y"). Med "retur[...]", hvis kanter er hele vejen rundt om programmets indre, fortælles det, at efter proceduren skal den oplyse informationerne fra programmet i en liste. I linje 3 står her: "[s * T[1][1] + x, s * T[1][2] + y]", Her fortæller T[1][1], at det er første-koodinatet i første liste T[1], som skal skaleres med faktoren s og rykkes x hen ad førsteaksen. T[1] er første hjørnepunkt i en trekant. Det samme gør sig gældende for anden-koodinatet ("T[2]") i første punkt ("T[1]") i y-retningen. Samme princip udføres i de næste to linje blot med 2. og 3. punkt i listen, hvis elementer, punkter, danner en trekant. Herefter endes programmet ved at skrive "end".

Nu omtales sierpinski-programmet igen, for her ses, at programmet "transform" benyttes i de næste tre trin (trin 12-14). Til at forklare hvorfor netop de værdier er valgt benyttes bilag 2. Der ses en liste T, som indeholder hjørnepunkter P, Q og R, har samme koordinater som trekanten, der arbejdes med i programmet. Med transform-programmet transformeres punkter. Der ses, at ved alle transformationerne skaleres alle punkterne ned med faktoren 0,5, hvilket betyder at sidelængderne af trekanterne halveres. Dette passer i overensstemmelse med at sidelængderne i sierpinski-trekanten halveres for hver iteration. Nu skal man forstille sig, at man har tre trekanter, som står oven i hinanden. Trekanterne skal forskydes, så sierpinski-trekanten dannes. Ved den første transformation dannes trekanten, som skal blive stående det samme sted, og forskydes derfor hverken hen ad første- eller andenaksen. I anden transformation rykkes den nye trekant 0,5



hen ad førsteaksen, så trekanten fra første transformation deler et hjørne med denne trekant. Ved den sidste transformation da rykkes trekanten 0,25 hen ad førsteaksen, og 0,75 op ad andenaksen. Dermed deler den 3. transformerede trekant et hjørnepunkt med de to andre transformerede trekanter. Sådanne transformationer udføres for alle trekanter i L. Anden For-loop på linje 11 stoppes ved at skrive "end do". Så defineres $L := M$, da L-listen, hvis lister transformeres, således kan lagres i M. Herefter stoppes første for-loop på linje 10, og hele proceduren stoppes med "end proc". Herefter kan programmet tegnes med Maples normale kommandoer for plots. For at plote punkterne kaldes listen sierpinski(n) et tilfældigt navn, som benyttes i kommandoen til at tegne fraktalet. Med dette program er følgende generationer af sierpinski-trekanten fremkaldt.



Figur 3: 0. generation, fremkaldt med bilag 3

Figur 4: 1. generation, fremkaldt med bilag 4

Figur 5: 2. generation, fremkaldt med bilag 5

Figur 6: 7 generation, fremkaldt med bilag 1, med akseindstillingerne $0 \leq x \leq 0.5$ og $0 \leq y \leq 0.5$

Dog kan programkoden ikke tegne 0. generation grundet linje 10, da "i" ikke kan gå op til 0. Derfor er figur 3 tegnet separat. Men med programkoden kan man undersøge en reverseret udgave af fraktalet på en mindre skala og bekræfte den selvsimilære egenskab, den besidder, som ikke kan bevises. Dermed er programmering med til eksperimentelt at udforske fraktalet. På figur 6, ses 7. generation sierpinski-trekant i et mindre grafvindue, hvor den selvsimilære egenskab observeres.¹⁶ Sierpinski-trekanten fremkommer altså ved geometriske iterationer. Ud fra overvejelser, observationer af fraktalets opførsel og udseende efter de første iterationer kunne Sierpinski konkludere og bevise sætning 1 og 2 for sierpinski-trekanten. Dog tegnede han fraktalet i hånden. Her gælder følgende sætning:¹⁷

Sætning 1

Arealet (S) af sierpinski trekanten i n . generation kan bestemmes med følgende formel

$S_n = \left(\frac{3}{4}\right)^n \cdot A$, hvor A er arealet af 0. generations trekant. Her gælder at når $n \rightarrow \infty$, da går $\left(\frac{3}{4}\right)^n \rightarrow 0$, hvilket betyder at arealet af sierpinski trekanten er nul.

¹⁶ Her er i mente, det er en reverseret form af fraktalet her undersøges. Fremkalde fraktaler vil altid være reverseret, da man ikke kan udføre en iteration uendeligt.

¹⁷ Jesper Frandsen, 1990, s. 61-62



Bevis¹⁸

Lad arealet af trekanten i 0. generation, som ses på figur 3, være A . Efter første iteration vil arealet være $\frac{3}{4}$ mindre. Arealet af fraktalet i 1. generation, vist på figur 4, er derfor $\frac{3}{4} \cdot A$. Efter næste iteration er igen $\frac{3}{4}$ af arealet fjernet, og derfor er arealet af 2. generation, som ses på figur 5, $\left(\frac{3}{4}\right)^2 \cdot A$. Ved at undersøge sierpinski-trekantens første iterationer kan man bestemme arealet efter den n 'te iteration ved følgende formel: $\left(\frac{3}{4}\right)^n \cdot A$. Her gælder at $\frac{3}{4} < 1$, så når $n \rightarrow \infty$, da går $\left(\frac{3}{4}\right)^n \rightarrow 0$. Derfor kan man konkludere, at arealet af sierpinski-trekanten er 0.

■

En anden interessant egenskab kan konkluderes om sierpinski-trekantens kantlængde.¹⁹

Sætning 2

Kantlængden (K) af n . generation af sierpinski-trekanten kan bestemmes med $K = \left(\frac{3}{2}\right)^n \cdot 3p$,

hvor p er sidelængden af 0. generations sider. Når $n \rightarrow \infty$ da går

$\left(\frac{3}{2}\right)^n \rightarrow \infty$, hvilket betyder at kantlængden af sierpinski-trekanten er uendeligt.

Beviset for denne sætning gennemføres ikke, da fremgangsmåden minder meget om sætning 1's bevis. Beviset kan findes i bilag 6.

Men ud fra den aksiomatiske deduktive metode, kan der konkluderes, at sierpinski-trekanten har en uendelig lang kantlængde, men intet areal. Eksperimentelt, med computerens hjælp, kan Sierpinskis trekant undersøges på mindre skala, hvor selvsimilaritet hurtigt observeres. Samtidig tegnes fraktalet meget præcist og hurtigt. Lignende konklusioner kan sluttes om Kochs snefnug.

Kochs snefnug

Den svenske matematiker Niels Fabian Helge Von Koch fremlagde hans arbejde med Kochs snefnug i en artikel²⁰ i 1904.²¹ Konstruktionen af Kochs snefnug er meget enkel:²²

¹⁸ Ibid

¹⁹ Ibid

²⁰ Navnet på artiklen: Sur une courbe continue sans tangente, obtenue par une construction géométrique élémentaire

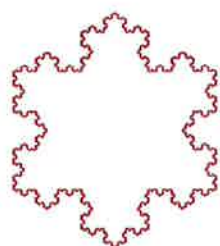
²¹ Miriam Ciavarella. *Some Fractals with Maple*. Maple, 2009

²² Jesper Frandsen, 1990, s. 61-62

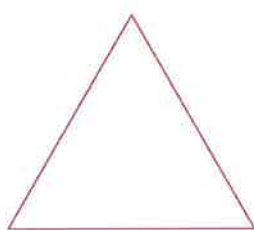


- I. Tegn en ligesidet trekant (0. generation, figur 9)
- II. Del hver af siderne i tre lige store længder, og konstruer en ligesidet trekant på det midterste linjestykke, og fjern det midterste linjestykke (1. generation)
- III. Herefter deles de nye sidelængder i tre, hvorefter nye trekantedes tegnes på samme måde. Denne proces kan fortsætte i det uendelige (efter n'te iteration kaldes den n. generation)

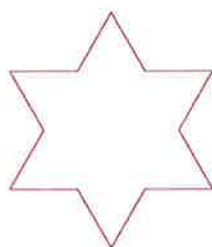
Kochs snefnug er den kurve, som fremkommer ved iterationen. En reversibel form af Kochs snefnug ses på figur 8. Med den simple fremgangsmåde ovenfor kan man opskrive et program til at tegne fraktalet, men dette vil først blive gjort under afsnittet: *Hvor lang er den britiske vestkyst?* Nedenstående generationer af Kochs snefnug er frembragt med programmet. Med programmet kan man eksperimentelt undersøge fraktalet på mindre skalaer og dermed konkludere på dens selvsimilære egenskab. Figur 12 vises fraktalet på en mindre skala, hvor den selvsimilære egenskab er værd at bemærke med det forbehold, at der undersøges en reverseret form af Kochs snefnug.



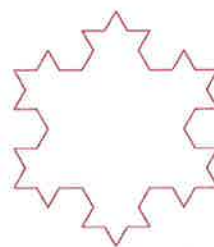
Figur 8: 6. generation Kochs snefnug



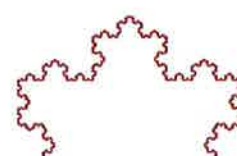
Figur 9: 0. generation Kochs snefnug



Figur 10: 2. generation Kochs snefnug



Figur 11: 2. generation Kochs snefnug



Figur 12: 6. generation Kochs snefnug, med akseindstillingerne $0,25 \leq x \leq 0,75$ og $0,65 \leq y \leq 0,9$

Koch undersøgte fraktalet i hånden, og ved at fastlægge nogle grundlæggende observationer efter få iterationer beviste han de følgende to sætninger. Her gælder følgende sætning.²³

Sætning 3

Omkredsen (L) af Kochs snefnug i den n . generation bestemmes med formlen: $L_n = \frac{4}{3} \cdot L_{n-1}$.
Omkredsen af en Kochs snekurve er uendeligt stort, da $L_n \rightarrow \infty$, når $n \rightarrow \infty$

Beviset for denne sætning vises ikke, dog kan beviset findes i bilag 7. At omkredsen er uendelig af et geometrisk objekt, er overraskende, men følgende sætning²⁴ vil gøre det mere interessant.

Sætning 4

Arealet (A) af Kochs snefnug kan bestemmes med formlen $A = \frac{2\sqrt{3} \cdot s^2}{5}$, hvor s er sidelængden af 0. generations Kochs snefnug.

²³ Ibid.

²⁴ Ibid.



Bevis²⁵

Arealet af en ligesidet trekant med sidelængden s kan beregnes med formlen²⁶: $T = \frac{\sqrt{3}}{4} \cdot s^2$

Først opskrives arealet af kurven, hvilket gøres ved at lægge alle trekanternes arealer, kurven består af, sammen. Ud fra de fremkaldte generationer af Kochs snefnug ses, at efter n antal iterationer er der $3 \cdot 4^{n-1}$ antal nye trekanter.

$$A = 3 \cdot 0 \frac{\sqrt{3}}{4} \cdot s^2 + 3 \cdot 1 \frac{\sqrt{3}}{4} \cdot \left(\frac{s}{3}\right)^2 + 3 \cdot 4 \frac{\sqrt{3}}{4} \cdot \left(\frac{s}{3^2}\right)^2 + 3 \cdot 4^2 \frac{\sqrt{3}}{4} \cdot \left(\frac{s}{3^3}\right)^2 + \dots$$

"..." betyder her, at det fortsætter i uendelighed. Dette udtryk omskrives ved at trække $\frac{\sqrt{3}}{4} \cdot s^2$ uden for en parentes.

$$A = \frac{\sqrt{3}}{4} \cdot s^2 \left(1 + 3 \cdot \left(\frac{1}{3}\right)^2 + 3 \cdot 4 \cdot \left(\frac{1}{3^2}\right)^2 + 3 \cdot 4^2 \cdot \left(\frac{1}{3^3}\right)^2 + \dots \right)$$

Herefter multipliceres udtrykket inden for parentesen med fire, mens det uden for parentesen divideres med fire. Dermed er værdien i udtrykket ikke ændret.

$$\begin{aligned} & \frac{\sqrt{3}}{16} \cdot s^2 \left(4 + 3 \cdot 4 \cdot \left(\frac{1}{3}\right)^2 + 3 \cdot 4^2 \cdot \left(\frac{1}{3^2}\right)^2 + 3 \cdot 4^3 \cdot \left(\frac{1}{3^3}\right)^2 + \dots \right) \\ &= \frac{\sqrt{3}}{16} \cdot s^2 \left(4 + 3 \cdot 4 \cdot \frac{1}{9} + 3 \cdot 4^2 \cdot \left(\frac{1}{9}\right)^2 + 3 \cdot 4^3 \cdot \left(\frac{1}{9}\right)^3 + \dots \right) \\ &= \frac{\sqrt{3}}{16} \cdot s^2 \left(4 + 3 \cdot \frac{4}{9} + 3 \cdot \left(\frac{4}{9}\right)^2 + 3 \cdot \left(\frac{4}{9}\right)^3 + \dots \right) \end{aligned}$$

Dette udtryk kan forkortes. Til dette defineres $s = \frac{4}{9} + \left(\frac{4}{9}\right)^2 + \left(\frac{4}{9}\right)^3 + \dots$. Her undersøges $\frac{4}{9} \cdot s$

$$s \cdot \frac{4}{9} = \left(\frac{4}{9}\right)^2 + \left(\frac{4}{9}\right)^3 + \left(\frac{4}{9}\right)^4 + \dots$$

Og til dette trækkes $\frac{4}{9} \cdot s$ fra s

$$s - \frac{4}{9} \cdot s = \frac{4}{9} + \left(\frac{4}{9}\right)^2 + \left(\frac{4}{9}\right)^3 + \dots - \left(\left(\frac{4}{9}\right)^2 + \left(\frac{4}{9}\right)^3 + \left(\frac{4}{9}\right)^4 + \dots \right) = \frac{4}{9}$$

Så $s - \frac{4}{9} \cdot s = \frac{4}{9}$. Her gælder at $s = \frac{9}{5}s$, og dermed kan man sige at:

$$\frac{9}{5}s - \frac{4}{9} \cdot s = \frac{4}{9} = \frac{5}{9}s \Leftrightarrow s = \frac{4}{9} \cdot \frac{9}{5} = \frac{4}{5}$$

Efter denne undersøgelse, kan man omskrive det røde markerede fra tidligere til

$$3 \cdot \frac{4}{9} + 3 \cdot \left(\frac{4}{9}\right)^2 + 3 \cdot \left(\frac{4}{9}\right)^3 + \dots = 3 \cdot \left(\frac{4}{9} + \left(\frac{4}{9}\right)^2 + \left(\frac{4}{9}\right)^3 + \dots \right) = 3 \cdot \frac{4}{5}$$

Derfor gælder her:

²⁵ Khan Academy, 2011, part 1 og 2.

²⁶ Omregningen til denne formel findes i bilag 8



$$A = \frac{\sqrt{3}}{16} \cdot s^2 \left(4 + \frac{12}{5}\right) = \frac{\sqrt{3}}{16} \cdot s^2 \left(\frac{20}{5} + \frac{12}{5}\right) = \frac{\sqrt{3}}{16} \cdot s^2 \cdot \frac{32}{5} = \frac{\sqrt{3}}{1} \cdot s^2 \cdot \frac{2}{5} = \frac{2\sqrt{3} \cdot s^2}{5}$$

Aralet af Kochs snefnug, hvor der gælder, at generation 0.'s sidelængden er s , er derfor $\frac{2\sqrt{3} \cdot s^2}{5}$.

Man kan altså også om Kochs snefnug aksiomatisk deduktivt slutte egenskaber ved fraktalet, som at den har en uendelige omkreds, men begrænser et bestemt areal. Samtidig kan man på helt små skalaer eksperimentelt undersøge fraktalet og studere dens selvsimilære egenskaber.

De tidlige fraktalers betydning for fraktalteoriens- og matematikkens udvikling Sierpinski-trekanten og Kochs snefnug var nogle af de først beskrevet fraktaler, og dermed lagde de, de første sten i fraktalteorien. Deres simple konstruktion gjorde dem mulige at udforske, og slutte konklusioner ud fra. Her blev konkluderet at objekterne havde selvsimilære egenskaber, samt, at man havde med geometriske objekter at gøre, som ikke passer ind den klassiske geometri. Dermed var arbejdet med disse fraktaler altså med til at udfordre det matematiske område geometri. For eksempel har begge omtale fraktaler en uendelig lang kurvelængde, men et begrænset areal (Kochs snefnug) eller intet areal (sierpinski-trekant). Så allerede de tidlige fraktaler var med til at udvikle matematikkens geometri. I dag kan man undersøge fraktalernes selvsimilære egenskaber på helt små skalaer og danne en god approximation af fraktalet. Men fraktalernes betydning for udviklingen af matematikken er ikke færdig i starten i 1900'tallet.

Julia-mængder

Den franske matematiker Gaston Julia arbejdede i starten af 1900'tallet med de såkaldte Julia-mængder. G. Julia udgav i 1918 et værk om rationelle iterationer²⁷. Men Julia-mængderne var næsten glemt indtil slutningen af 1970'erne, hvor Mandelbrot skulle bruge dem.²⁸ Grunden til, at de næsten forglemmes, bliver tydeliggjort i denne opgave. Til dette vil udvalgt teori om Julia-mængderne først blive gennemgået, hvormed computerens betydning for fraktalerne er tydelig.

Julia-mængder fremkommer ved iterationsfunktionen $f(z) = z^2 + c$, hvor c er en konstant²⁹. Her vil i denne opgave blive fokuseret på Julia-mængderne i det komplekse talplan, hvilket betyder, at

²⁷ J J O'Connor og E F Robertson, 2008, Gaston Maurice Julia

²⁸ Jesper Frandsen, 1990, s. 58

²⁹ Kenneth Falconer, 2006, s. 215



c er et irrationelt tal.³⁰ z er begyndelsesværdien for iterationen.³¹ Mængden af begyndelsesværdier der går mod uendelig ved iteration, her gælder: $|z_n| \rightarrow \infty$, når $n \rightarrow \infty$, kaldes U mængden. Julia-mængden er defineret som mængden, der ved iteration er randen af U , men ikke er den del af U .³² c kan være et hvilket som helst tal i det komplekse plan, og hver ny c -værdi er en ny Julia-mængde.³³ For at forklare hvorfor dette er gældende, indføres følgende begreb.³⁴

Definition 2

Et fikspunkt z^* for en funktion f , er hvor der $f(z^*) = z^*$

Fikspunkter for funktionen $f(z) = z^2 + c$ må derfor kunne findes således³⁵:

$$f(z) = z \Leftrightarrow z^2 + c = z \Leftrightarrow z^2 - z + c = 0 \Leftrightarrow z = \frac{1 \pm \sqrt{1 - 4c}}{2}$$

Når en funktion har et fikspunkt, da er der startværdier, der ikke undslipper til det uendelige efter uendelig antal iterationer. I den komplekse talplan findes der to løsninger til ligningen ovenfor, og dermed to fikspunkter.³⁶ Fikspunkter samt baner kan være tiltrækkende eller frastødende, alt efter om en anden bane tiltrækkes eller frastøder fikspunktet.³⁷ Hvis fikspunkterne er tiltrækkende, er der baner med hver deres startværdi, som aldrig irriteres mod det uendelig, og derfor må ligge på randen til mængden U og dermed være en del af Julia-mængden J . Fikspunktet afhænger af c , og derfor findes der uendeligt mange Julia-mængder, da der for hver ny c -værdi er en ny Julia-mængde. For hver Julia-mængde skal man afgøre hvilke startværdier, der er en del af Julia-mængden, og dem der ikke er. For at forsimple beregningerne af Julia-mængderne er det derfor en god ide at frasortere de begyndelsesværdier z_0 , hvor der gælder, at dens bane går mod uendelig³⁸: $|z_n| \rightarrow \infty$, når $n \rightarrow \infty$, hvilket følgende sætning er belejlig med:³⁹

Sætning 5

For iteration af $f(z) = z^2 + c$ med begyndelsesværdien z_0 og n er antal iterationer, gælder der:

Hvis $|c| \leq 2$, og $|z_0| > 2$, gælder der at $|z_n| \rightarrow \infty$ når $n \rightarrow \infty$

Hvis $|c| > 2$, og $|z_0| \geq |c|$, gælder der at $|z_n| \rightarrow \infty$ når $n \rightarrow \infty$

³⁰ Jesper Frandsen, 1990, s. 79-87 og s. 91-102. c kan også være rationelt.

³¹ Søren Halse, Erik Laage-Petersen og Jens Peter Touborg, 2005, s. 139

³² Ibid.

³³ På vær $c = 0$ og $c = -2$, da funktionen da bliver differentiable over alt. Jesper Frandsen, 1990, s. 94

³⁴ Jesper Frandsen, 1990, s. 48

³⁵ Jesper Frandsen, 1990, s. 74-75. Her benyttes determinanten for et andengradspolynomium til omskrivelsen.

³⁶ Medmindre der kun findes en løsning til ligningen. (to fikspunkter er faldet sammen til en).

³⁷ Ibid, s. 50. Nærmere uddybelser senere i opgaven

³⁸ Jesper Frandsen, 1990, s. 72

³⁹ Ibid, s. 72

**Bevis⁴⁰**

Man ser på to tilfælde hvor der gælder at $|z_0| > c$ og $|z_0| > 2$. Med anvendelse af sætningen i bilag 9 får man for $|z| > c$ at:

$$|f(z)| \geq |z|^2 - |c| \geq |z|^2 - |z| = |z| \cdot (|z| - 1)$$

For at simplificere dette udtryk sættes $|z| - 1 = k$. Derfor kan man skrive:

$$|f(z)| \geq |z| \cdot k$$

Man ved, at $k > 1$, da $|z| > 2$. Dette må betyde at $f(z)$ ligger længere væk fra 0 end z . Hvis man iterere 1 gang, da må der gælde:

$$|f^2(z)| = |f(f(z))| \geq |f(z)| \cdot k \geq k^2|z|$$

Dette skrives generelt op, med anvendelse at $f^n(z) = z_n$, som:

$$|f^n(z)| \geq k^n|z| = |z_n| \geq k^n|z_0|$$

Man ved som sagt, at $k > 1$, hvilket må betyde at $k^n \rightarrow \infty$ og dermed også $|z_n| \rightarrow \infty$ når $n \rightarrow \infty$. $|z| - 1 = k$, og derfor afhænger ovenstående af z . Her undersøges dette problem ved at sige $|z_n| - 1 = k_n$, og $n = 0$ og $n = 1$. Da kan man opskrive:

$$|z_1| > |z_0| \Leftrightarrow |z_1| - 1 > |z_0| - 1 \Leftrightarrow k_1 > k_2 > 1.$$

Dette kan gøres for alle iterationer n , hvilket betyder at: $k_n > k_{n-1} > 1 \dots > k_1 > k_0$

Hvis k 'et i $|z_n| \geq k^n|z_0|$ sættes til at være k_0 , og dermed erstatter alle de andre k 'er, gøres højresiden endnu mindre ift., hvad den kunne have været. Dermed får højresiden i $|z_n| \geq k^n|z_0|$ endnu mindre betydning, og således er man ude over problemet med, at udtrykket afhænger af k .

■

Sætning 5 tager ikke højde for alt. Hvad hvis $|c| \leq 2$ og $|z_0| < 2$? For dette findes der ingen sætning,⁴¹ og derfor må man eksperimentelt undersøge disse Julia-mængder. Her skal man se efter, om banen for begyndelsespunktet z_0 engang kommer udenfor cirklen med centrum i 0 og

⁴⁰ Ibid, s. 72-73

⁴¹ Ibid, s. 73



radius på 2,⁴² for da siger sætning 5, at banen vil gå imod det uendelige. Sætningen kommer heller ikke ind på tilfældet $|c| > 2$ og $|z_0| < |c|$.⁴³ Man kan begrænse sine beregninger af Julia-mængder yderligere med følgende sætning⁴⁴.

Sætning 6

Hvis $|c| > 2$ vil den kritiske bane for $f(z) = z^2 + c$ gå mod uendelig

Bevis⁴⁵

Sætning i bilag 10 siger, at hvis $|z| \geq |c| > 2$, så vil banen med z som begyndelsesværdi gå mod uendelig. Her gælder, at $f(0) = c$, så hvis $|c| > 2$, da går den kritiske bane⁴⁶ mod uendelig. ■

Ved brug af sætning 6 kan man indsnævre sin undersøgelse af iterationsfunktioner, hvor c -værdier ligger på eller inden for en cirkel med radius 2 og centrum i O ⁴⁷. Her noteres, at sætningen kun fortæller, hvad der sker med funktioner med c -værdier uden for cirklen og ikke, at kritiske baner inden for cirklen ikke kan gå mod uendelig.

Sætning 5 og 6 er anvendelig i programmering af Julia-mængder. Begge sætninger er anvendt i programmet for Julia-mængden $f(z) = z^2 - 0,801 - 0,155 \cdot i$. For Julia-mængden gælder der, at $|c| < 2$. Sætning 5 anvendes ved, at her kun undersøges startværdier, hvis bane ikke går mod uendelig. Programkoden ses på figur 13.⁴⁸

```
> restart: with(plots): #1
> Julia1:=proc(x,y); #2
local c, z, m; #3
c:=evalf(x+y*I); #4
z:=c; #5
for m from 0 to 100 while abs(z)<2 do #6
z:=z^2-0.801-0.155*I #7
od; #8
m #9
end proc: #10
> #11
> plot3d(0,-2..2,-1.2..1.2, #11
orientation=[-90, 0],grid=[640,350], #12
style=patchnogrid, color=Julia1); #13
```

Figur 13: Programkode til Julia-mængden $f(z) = z^2 - 0.801 - 0.155 \cdot i$ i Maple

⁴² Læs om cirkelafbildning, iterationer med $f(z) = z^2$ i bogen *komplekse tal og fraktaler* af Jesper Frandsen for en dybere forståelse af cirkels betydning.

⁴³ Kan læses om i bogen *komplekse tal og fraktaler* af Jesper Frandsen kapitel 11.

⁴⁴ Jesper Frandsen, 1990, s. 94-96

⁴⁵ Ibid, s. 95-96

⁴⁶ Den kritiske bane har enden nul eller c som begyndelsepunkt

⁴⁷ Ibid, s. 96

⁴⁸ Alexander F. Walz, 1999, Mandelbrot.



Her kaldes programmet Julia1. Til dette program udnyttes den automatiske farvelægning ved 3D-plot i Maple. Til at programmere benyttes nogle lokale operatører c , x og m . Først defineres et punkt c er i den komplekse talplan som skal angives med decimaler, hvilket "evalf" sørger for. Derefter sættes $z = c$, for således kan der regnes med z som et komplekst tal med decimaler. Herefter skal funktionen $f(z)$ itereres hundrede gange, hvilket skrives i linje 5 som "for m from 0 to 100 [...] do", så længe at modulus af z er mindre end 2, hvilket skrives som "while $abs(z)=2$ ". Dette skrives



Figur 14: Reverseret tegning af Julia-mængden:
 $f(z) = z^2 - 0,801 - 0,155 \cdot i$

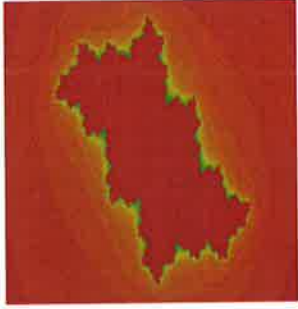

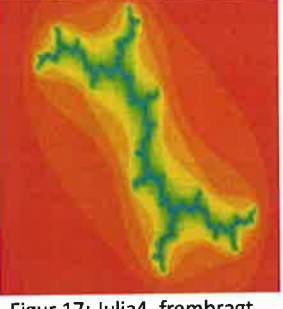
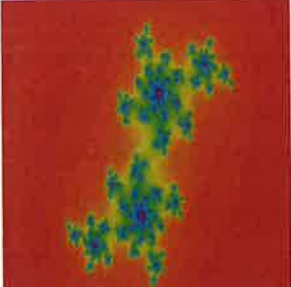
grundet sætning 5. Herefter stoppes for-loopet, som startede i linje 6 ved at skrive "od" i linje 8. "m" bruges til at tegne Julia-mængden, da m er de punkter, som ved iteration ikke kom udenfor cirklen med centrum i O , og radius 2. Til at tegne programmet anvendes plot3d i Maple hvor det passende akseinterval til Julia-mængden vælges. Her sættes z -aksen til 0, da Julia-mængderne er i 2D. Derudover anvendes "orientation = [-90,0]", så man ser ned på x - y -planen. "Grid" angiver, hvor mange prikker der skal være i feltet, og stilen "patchngrid" anvendes, så tegningen hænger sammen, da standartindstillingen er prikker. Til sidst angives der, at det er Julia1, der skal farves. Her fremkommer Julia-mængden på figur 14. Julia-mængden er overgangen fra det indkredsede røde til det flerfarvede. Programmet skal man dog have forbehold over for, da man ikke med sikkerhed kan sige, at en bane ikke går mod uendelig, når iterationen stoppes. De mest præcise Julia-mængder frembringes ved at udføre flest mulige iterationer.

Med computeren blev det muligt at visualisere Julia-mængderne, og G. Julias teori om Julia-mængderne kunne dermed bedre forstås. Ved at have undersøgt mange Julia-mængder eksperimentelt inddeles de i fire grupper alt efter udseende, hvilket fremgår af tabellen nedenfor. Hver kategori har forskellige karakteristika, som man eksperimentelt har fundet frem til.⁴⁹ Her er tegnet en fra hver kategori.⁵⁰

⁴⁹ Fremgangsmåden for dette er her ikke lagt vægt på, da det vil opkræve meget mere plads end denne opgave må. Dog kan her læses lidt mere om det i bogen: *komplekse tal og fraktaler* af Jesper Frandsen, s. 91-102

⁵⁰ Jesper Frandsen, 1990, s. 93.



<p>Julia-mængder med enkle lukkede kurver, der deler planen i tre dele. F.eks. $f(z) = z^2 - 0,195 + 0,59 \cdot i$.</p> 	<p>Julia-mængder der deler planen i tre dele dog mindre simpelt. F.eks. $f(z) = z^2 - 0,6686866 - 0,3506868 \cdot i$</p> 	<p>Julia-mængder hvor $c = i$, omkredser ingen punkter, og ligner en tråd. F.eks. $f(z) = z^2 + i$.</p> 	<p>Julia-mængder som består af en sky af punkter kaldes støvfraktal. F.eks. $f(z) = z^2 + 0,11 - 0,67 \cdot i$.</p> 
<p>Figur 15: Julia2, frembragt med 40 iterationer. Programkode i bilag 11</p>	<p>Figur 16: Julia3, frembragt med 200 iterationer. Programkode i bilag 12</p>	<p>Figur 17: Julia4, frembragt med 30 iterationer. Programkode i bilag 13</p>	<p>Figur 18: Julia5, frembragt med 60 iterationer. Programkode i bilag 14</p>

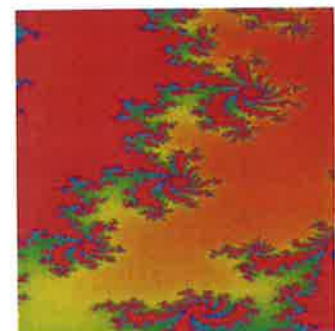
Med programmerne kan man udforske fraktalerne på mindre skalaer. Der er ingen sætninger for, hvordan disse udformer sig, og derfor er man nødt til eksperimentelt at undersøge dem. På figur 20 ses en forstørrelse af Julia3.

Her er i denne opgave lagt fokus på, hvordan udvalgte dele af G. Julias teori om Julia-mængder, fundet ved aksiomatisk deduktive beviser, kan bruges til at programmere Julia-mængder,⁵¹ hvorved fraktalerne kan undersøges eksperimentelt, og deres udseende både som en hel mængde og på mindre skalaer kan bestemmes.

Fraktalteorien blev udvidet til også at indeholde algebraiske fraktaler, hvor nye begreber som fikspunkter har stor betydning for deres eksistens og udseende. Ingen sætninger, fundet med den aksiomatiske deduktive metode, beskriver, hvordan en Julia-mængde ser ud, og derfor er matematikken nødsaget til eksperiment at undersøge Julia-mængderne, hvilket er imod den normale opfattelse af, hvordan sætninger sluttes i matematik.

Hvor lang er den britiske kystlinje?

Dette spørgsmål dykkede Benoit B. Mandelbrot ned i med artiklen "How long is the coast of



Figur 19: Skaleret Julia3: Frembragt med 200 iterationer. Programkode: bilag 12

⁵¹. Man kunne i dette afsnit have gået i dybden med fikspunkter, tiltrækkende og frastødende periode-n punkter, samt deres betydning fra fraktalernes udseende. Af pladsmangel er dette ikke medtaget. Læs mere om det i *Komplekse tal og fraktaler*, af Jesper Frandsen, s. 94.



Britain?" i 1967.⁵² Her genoptog han Lewis Fry Richardson's problematik om at måle længden af den britiske kystlinje⁵³. Den britiske kystlinje er irregulær og har selvsimilære egenskaber, som ses ved at skalere kystlinjen. Man kan blive ved med at skalere kystlinjen mindre, ligesom de geometriske fraktaler i afsnittet om de tidlige fraktaler, hvis omkreds var uendelig. Her opstår problematikken: den britiske kystlinje har en uendelig omkreds.⁵⁴ Dette er ikke anvendeligt, når kysters længder skal sammenlignes, da alle kyster i så fald må være uendeligt lange. Derfor argumenterede Mandelbrot for, man måtte bruge en anden faktor end længdemål til at sammenligne kystlinjers kurver, nemlig den fraktale dimension.⁵⁵ Her udfordrer fraktalteorien endnu engang den klassiske Euklids geometri, hvor begrebet dimension før var defineret ud fra. Her er et punkt 0. dimension, en linje 1. dimension, en firkant 2. dimension og en terning 3. dimension.⁵⁶ Hvis disse tildeles en enhed f.eks. cm, da svarer dimensionen til eksponenten,⁵⁷ altså henholdsvis cm^1 , cm^2 og cm^3 . Fraktaler er hverken et punkt, en linje, firkant eller terning. Dog har selvsimilære geometriske objekter noget til fælles med dimensioner. Hvis man deler en linje, et kvadrat eller terning i lige mange mindre stykker, da er forholdet mellem hele figuren og et stykke af den $1:n$. Hvis k er antallet af nye figurer med længden $1/n$, da må der gælde for alle tre tilfælde, hvor d er dimensionen:⁵⁸

$$n^d = k \Leftrightarrow d \cdot \log(n) = \log(k) \Leftrightarrow d = \frac{\log(k)}{\log(n)}$$

På samme måde kan et selvsimilær fraktal skaleres ned med en faktor n , hvorved k antal nye kopierer af fraktalet dannes. Derfor gælder følgende definition.⁵⁹

Definition 3

En selvsimilær fraktals fraktale dimension, D , kan bestemmes ved

$$D = \frac{\log(\text{antal nye dele i næste trin})}{\log(\text{scarlingsfaktor fra lille til stor del})}$$

Mandelbrot forsøgte sig frem med forskellige fraktaler og bestemte deres dimension. Herunder

⁵² Mandelbrot, B. B, How long is the coast of Britain?, 1967

⁵³ Ibid

⁵⁴ Ibid

⁵⁵ Mandelbrot, B. B, How long is the coast of Britain? 1967

⁵⁶ Jesper Frandsen, 1990, s. 66-67

⁵⁷ Ibid, s. 67.

⁵⁸ Ibid

⁵⁹ Ibid



undersøgte han f.eks. *A Smother Koch Island*,⁶⁰ hvis dimension er $D=1,1291$ og *The Arrowhead Curve*,⁶¹ hvis dimension er 1,5849 og mange flere⁶². Han undersøgte også Kochs snefnug, som er omtalt tidligere. Fraktalet animeres her, da den har en bestemt rolle i at kunne beskrive Storbritanniens kystlinje. Programkoden til en animation af Kochs snefnug ses på figur 20.⁶³

På linje 1 defineres et regnestykke, for at de to programmer nedenfor beregnes hurtigere. Dette gøres ved at angive udtrykket med apostrof. Herefter startes det første program "side", hvilket har til formål at beregne punkterne til de nye koordinater, for at siden AB efter n iterationer viser n . generation af den pågældende side på Kochs snefnug. I

```

> `1/sqrt(3)` := 1.0/sqrt(3.0); #1
> Side := proc(A, B, n) #2
  local n1, A1, A2, B1, B2, C, D, E; #3
  if n <= 0 then return B end if; #4
  n1 := n-1; A1 := A[1]; A2 := A[2]; B1 := B[1]; B2 := B[2]; #5
  C := [(2.0*A1+B1)/3.0, (2.0*A2+B2)/3.0]; #6
  D := [(A1+2.0*B1)/3.0, (A2+2.0*B2)/3.0]; #7
  E := [((A1+B1)-`1/sqrt(3)`*(B2-A2))/2.0, #8
        ((A2+B2)+`1/sqrt(3)`*(B1-A1))/2.0]; #9
  Side(A, C, n1), Side(C, E, n1), #11
  Side(E, D, n1), Side(D, B, n1) #12
end proc; #13
> Snowflake := proc(nmax) #14
  local A, B, C, n; #15
  A := [0.0,0.0]; B := [0.5, sqrt(3.0)/2.0]; C := [1.0,0.0]; #16
  seq(plot([A, Side(A, B, n), Side(B, C, n), #17
            Side(C, A, n)]), n = 0..nmax); #18
  plots[display]({%}, insequence = true, #19
                 axes = NONE, scaling = CONSTRAINED, #20
                 title = "Koch Snowflake"); #21
end proc; #22
> Snowflake(6) #23

```

Figur 20: programkode til animation af Kochs snefnug

programmet konstrueres en sekvens af toppunkter, som er repræsenteret som lister bestående af xy -koordinater. Der benyttes igen lokale operatører. I linje 4 står der, at hvis $n \leq 0$, da skal programmet returnere til B, hvilket også giver god mening, da der ingen iteration skal ske når $n \leq 0$. I linje 5 defineres første- og andenkoordinaterne for punkterne A og B hver for sig, så man kan regne med dem separat. I bilag 15, vises udregninger for hvorfor linje 6-9 skrives op som de gør, da dette ellers ville tage plads fra andre dele af opgaven. Punkterne C, D og E blev her defineret. I linje 11-12 samles de nye punkter i sider. I bilag 15 ses også en figur af en side af Kochs snefnug efter 1. iteration, hvor man da kan se at siderne skal side sammen som AC, CE, DE og DB. Her ses i programmet at $n1 := n - 1$, fordi første generation i programmet er den generation som i denne opgave kaldes 0. Dette program skal altså udføres $n-1$ gange for at danne n . generation af en af siderne i Kochs snefnug. Herefter stoppes programmet med "end proc".

⁶⁰ Mandelbrot, Benoit B., 1977, s. 46

⁶¹ Ibid, s. 56

⁶² Se i bogen *Fractals form, chance, and dimension* af Benoit B Mandelbrot for flere eksempler.

⁶³ Francis Wright, *Computing with Maple*, s. 255-259



Herefter benyttes programmet Snowflake, hvilket tegner den n. generation. Her benyttes igen lokale operatorer. 0. Generation's hjørnepunkter er opgivet som koordinaterne A, B, og C, som ses i linje 16. Sammen med disse plottes nu resultaterne, som fremstilles i programmet "side" for alle siderne i trekanten, altså AB, AC og BC. Dette skrives ved, at sekvensen fra programmet "side" hentes over i Snowflakeprogrammet, hvilket sker i linje 23. Herefter benyttes plotindstillinger til at animere, nemlig "*plots[display]*", og det, den skal animere, er den udvikling, som står ovenfor, så for ikke at skrive det hele igen, anvendes at % betyder, at værdierne ovenfor hentes ned og benyttes. Ved nu at benytte normale indstillinger for plots i Maple animeres op til 6. generation af Kochs snefnug ved at skrive Snowflake(6). Animationen skal manuelt sættes i gang på play i Maple

Ud fra animationen kan man se at Kochs snefnug får yderligere fire sider pr. i forvejen eksisterende side pr. iteration. Siderne er alle 1/3 af sidelængden af den forrige. Dermed kan Kochs snefnugs fraktale dimension bestemmes således:

$$D = \frac{\log(3)}{\log(4)} = 1,261859507$$

Kochs snefnugs fraktale dimension er derfor 1,26. Mandelbrot beregnede på baggrund af Richardsons resultater fra 1961 Storbritanniens vestkysts fraktaldimension til at være 1,25.⁶⁴ Kochs snefnug er altså det fraktal, som bedst beskriver Storbritanniens vestkysts uregelmæssigheder. Så i stedet for at spørge hvor lang Storbritanniens vestkyst er, kan man i stedet spørge, hvor kurvet den er.⁶⁵

Det nye i fraktalteorien i dette nedslag er altså, at fraktaler kan tildeles en ikke helttalig dimension, den fraktale dimension, og dermed havde fraktaler endnu en gang en betydning for matematikkens udvikling. Begrebet dimension blev omdefineret, og fraktaler fik en betydning for beskrivelsen af den ellers ubeskrivelige Britiske vestkyst. Fraktalerne muliggjorde en besvarelse af et problem, som ellers så uløseligt ud.

Feigenbaumdiagram

I 1975 arbejdede Mitchell Feigenbaum med ikke-lineære iterationsfunktioner.⁶⁶ Disse iterationsformer er meget interessante, som man allerede kan konkludere med de tidligere

⁶⁴ Mandelbrot, B. B, How long is the coast of Britain? 1967

⁶⁵ Nigel Lesmoir-Gordon, Will Rood og Ralph Edney, 2009, s. 37

⁶⁶ Jesper Frandsen, 1990, s. 88



omtalte Julia-mængder. Feigenbaumdiagrammer åbne som Julia-mængderne op for forskning i kaosteori.

De logistiske fremskrivningsfunktioner

Det mest kendte feigenbaumdiagram, er skabt med iterationsformlen:⁶⁷

$$x_{n+1} = a \cdot x_n \cdot (1 - x_n) = ax - ax^2$$

Her er x den uafhængelige variabel, mens x_{n+1} er den afhængelige, og a er en fast reel parameter.⁶⁸ For nogle a -værdier fremkommer ikke pæne typiske logistiske kurver, og det er disse a -værdier, der er interessante.⁶⁹ For at undersøge dette ses her nærmere på selve formelen. Først findes funktionens fikspunkter. Her antages det, at a er positiv, da det er disse funktioner, som er interessante. Fikspunktet findes ved at løse følgende ligning:⁷⁰

$$f(x) = x \Leftrightarrow x = a \cdot x \cdot (1 - x)$$

Her ses, at der altid er fikspunktet $x = 0$, hvilket er uafhængige af, hvilken værdi a tillægger sig. Ved at omskrive sætningen, kan det andet fikspunkt x^* findes.

$$x^* = ax^* \cdot (1 - x^*) = ax^* - ax^{*2} \Leftrightarrow \frac{x^*}{a} = x^* - x^{*2} \Leftrightarrow \frac{1}{a} = 1 - x^* \Leftrightarrow x^* = 1 - \frac{1}{a}$$

Fikspunkter kan, som tidligere skrevet, være enten tiltrækkende eller frastødende, hvilket er grundigere defineret her:⁷¹

Definition 3

Et fikspunkt x^* siges at være tiltrækkende, hvis $|f'(x^*)| < 1$. Et fikspunkt x^* siges at være frastødende, hvis $|f'(x^*)| > 1$. Et fikspunkt x^* siges at være neutralt, hvis $|f'(x^*)| = 1$

I forlængelse af fixpunkter er følgende sætning interessant for feigenbaumdiagrammer.⁷²

Sætning 7

Hvis et fikspunkt x^* for f er tiltrækkende, så vil en hver bane konvergere mod fikspunktet x^* , hvis startværdigen vælges tæt på x^*

Bevis⁷³

⁶⁷ Bjørn Felsager og Jonny Schulz, 1990, s. 7

⁶⁸ Ibid, s. 7

⁶⁹ Ibid

⁷⁰ Ibid, s. 20

⁷¹ Ibid, s. 16

⁷² Ibid

⁷³ Ibid, s. 16-17



Her antages, at $f'(x^*) > 0$. Ved brug af definition 3, da gælder det, at $0 < f'(x^*) < 1$. Her vælges et tal k mellem $f'(x^*)$ og 1, og ved brug af definitionen af en differentialkvotient gælder der:

$$\frac{f(x) - f(x^*)}{x - x^*} \rightarrow f'(x^*) \text{ når } x \rightarrow x^*$$

Derfor kan man vælge et symmetrisk interval I omkring x^* , så der for $x \in I$ gælder:

$$0 < \frac{f(x) - f(x^*)}{x - x^*} < k$$

I intervallet I gælder det:

$$|f(x) - f(x^*)| < k \cdot |x - x^*|$$

Hvis iterationen starter med begyndelsesværdien x_0 , som tilhører I , da gælder det:

$$|x_1 - x^*| = |f(x_0) - f(x^*)| < k^1 \cdot |x_0 - x^*|$$

Her ses, at x_1 ligger tættere på x^* end x_0 . Hvis man iterer n antal gange, gælder det generelt:

$$|x_n - x^*| < k^n \cdot |x_0 - x^*|$$

Da $0 < k < 1$ gælder det, at $|x_n - x^*| \rightarrow 0$, når $n \rightarrow \infty$. Hermed er sætningen bevist i tilfældet, hvor $f'(x^*) > 0$. ■

For de logistiske fremskrivningsfunktioner vil her undersøges, hvornår det er gældende, at den har tiltrækkende eller frastødende fikspunkter for senere at kunne beskrive feigenbaumdiagrammet.

Her undersøges først, hvis $a \leq 1$, da er nul det eneste fikspunkt⁷⁴. Her afgøres hurtigt ved brug af definition 3, at hvis $a < 1$, da er fikspunktet $x = 0$ tiltrækkende, men hvis $a > 1$, da er fikspunktet $x = 0$ frastødende. Dette kan konkluderes, netop fordi a er tangenthældningen i $x = 0$.⁷⁵

Hvis $a > 1$, findes der som sagt endnu et fikspunkt, som her kaldes x^* , hvis forskrift var:

$x^* = 1 - \frac{1}{a}$. For at afgøre om fikspunktet er tiltrækkende eller frastødende⁷⁶ benyttes definition 9.

Derfor differentieres $f(x)$.

$$f'(x) = (a \cdot x \cdot (1 - x))' = (ax - ax^2)' = a - 2ax$$

Her indsættes fikspunktet $x^* = 1 - \frac{1}{a}$ i den differentierede ligning:

$$f'(x^*) = a - 2a \cdot \left(1 - \frac{1}{a}\right) = a - 2a + 2 = 2 - a$$

Her kan man nu bestemme, hvilke a -værdier, hvor fikspunktet x^* , er tiltrækkende:

⁷⁴ Ibid, s. 19

⁷⁵ Ibid.

⁷⁶ Ibid, s. 20



$$|f'(x^*)| < 1 \Leftrightarrow |2 - a| < 1 \Leftrightarrow 1 < a < 3$$

Ud fra dette kan man slutte, at når $0 < a < 3$, da er der et fikspunkt til en givet x -værdi ud over fikspunktet $x = 0$.⁷⁷ Disse a -værdier er ikke så interessante, ift. $3 < a < 4$, for her optræder der periode- n punkter.⁷⁸ Disse periode-punkter⁷⁹ skal findes blandt fixpunkterne for den sammensatte funktion, hvilket tager lang tid at regne i hånden, hvorved computeren er behjælpelig. I et (a, x) -koordinatsystem, som Mitchell Feigenbaum studerede, opdagede han overraskende egenskaber ved den simple iterationsfunktion.⁸⁰ Denne graf fortæller til hvilke parameterværdier a , der er et tiltrækkende fikspunkt eller periodiske punkter. Et program⁸¹ til at tegne et figentræ for de logistiske fremskrivningsfunktioner ses på figur 21.

```
> Lfigentræ := proc(ned, øvre, pas)
  local k, start, a, b, s;
  s := {};
  a := ned;
  while a <= øvre do
    start := 0.1;
    for k to 50 do
      start := a*start*(1 - start);
    end do;
    for k to 100 do
      start := a*start*(1 - start);
      s := s union {[a, evalf(start)]};
    end do; a := a + pas; end do;
    plot([op(s)], 'a' = ned .. øvre,
    style = POINT, symbol = POINT, color=red);
  end proc;

Lfigentræ(0, 4, 0.1)
```

#1 Programmet kaldes Lfigentræ, og for at
#2 programmet kan køre, skal den bruge hvilken a -
#3 værdi, den skal starte ved: "ned" og hvilken,
#4 den skal slutte ved "øvre", samt et tal, "pas".
#5 Herefter er her defineret hvilke lokale
#6 operatorer, der vil benyttes. Så oprettes et set
#7 kaldet s til matematiske udtryk, hvilket her er
#8 punkterne, som generes i programmet. Dette
#9 er skrevet således: " $s := \{\}$ ". Herefter defineres
#10 a til at være den nedre a -værdi. Herefter skal
#11 programmet, des længe a er mindre end den
#12 øvre grænse, beregne det efterfølgende. Dette
#13 er kodet således " $while a <= øvre do$ ". Efterfølgende defineres startværdien for x til at være
#14 $0,1$, hvilket er værdien der itereres med. Så igangsættes iterationen ved at sige " $for k to 50 do$ ".
#15 Iterationsfunktionen er skrevet således: $start = a * start * (1 - start)$, hvilket er den logistiske
#16 fremskrivningsfunktion, som skal laves et Feigenbaumdiagram for. Herefter stoppes proceduren
#17 med " $end do$ " i linje 9 uden data dateres. Så starter en lignende iteration, som gentages 100
#18 gange, som gemmes, da der i programmet herefter er skrevet " $s := s union \{[a, evalf(start)]\};$ ".

Figur 21: programkode til fremkaldelse af figentræet for de logistiske fremskrivningsfunktioner

er kodet således " $while a <= øvre do$ ". Efterfølgende defineres startværdien for x til at være $0,1$, hvilket er værdien der itereres med. Så igangsættes iterationen ved at sige " $for k to 50 do$ ". Iterationsfunktionen er skrevet således: $start = a * start * (1 - start)$, hvilket er den logistiske fremskrivningsfunktion, som skal laves et Feigenbaumdiagram for. Herefter stoppes proceduren med " $end do$ " i linje 9 uden data dateres. Så starter en lignende iteration, som gentages 100 gange, som gemmes, da der i programmet herefter er skrevet " $s := s union \{[a, evalf(start)]\};$ ".

⁷⁷ Ibid.

⁷⁸ Ibid, s. 20-21

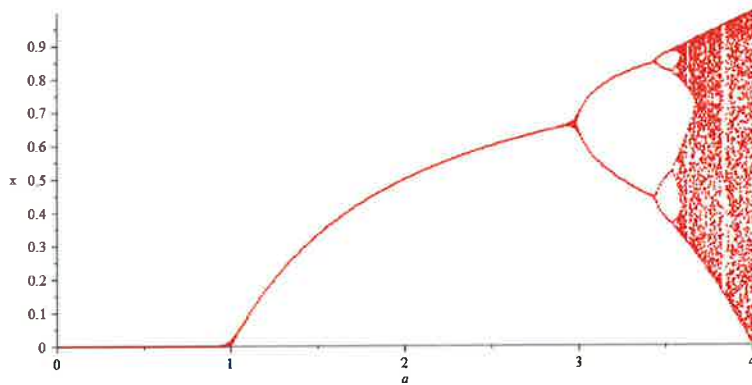
⁷⁹ Et periodisk punkt er hvor z efter et endeligt antal iterationer igen bliver tallet z .

⁸⁰ Ibid, s. 22

⁸¹ Anton Betten, 2008. Lecture 5: Sierpinski Triangle



Her defineres listen s til at skulle indeholde punktet s , som er en del af (union) punktet med a som første koordinat og de tilhørende x -koordinater som fremkom ved anden omgang iteration. Herefter stoppes iterationen og a sættes lig med a sammenlagt med tallet pas i linje 13. "Pas" bestemmer, hvor mange decimaler a skal have og dermed hvor stort mellemrum, der skal være mellem a -værdierne i programmet. Således kan man justere hvor detaljeret, programmet skal være. Derefter starter iterationen forfra fra "for k to 50 do" indtil, alle a -værdier er løbet igennem programmet. Programmet stopper, når a er lig "øvre", hvilket er signaleret med "end do". Når programmet er færdigt, plottes sættet s med punkterne. Til dette er her bestemt, at førsteaksens interval er fra nedre a -værdi til øvre, som er skrevet således "' a ' = ned.. øvre,". Her benyttes stilen punktplot, og symbolet er punker. Farven på punkterne er angivet til rød. Til sidst endes programmet ved at skrive "end proc". Herefter indtastes hvilket a -interval, man vil se grafen i, samt pas i kommandoen i LFigentræ og fraktalet tegnes. Ved programmet fremkommer figur 22.



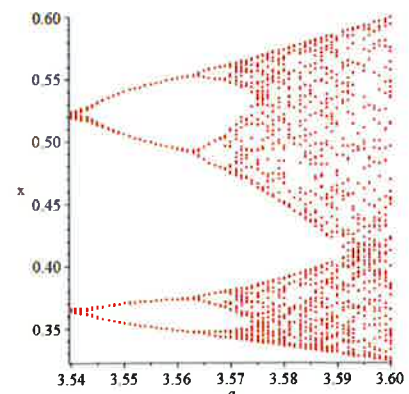
Figur 22: Feigenbaumdiagram for den logistiske fremskrivningsfunktioner, hvilken fremkaldes af programkoden fra figur 21

Med programmet kan man, som Feigenbaum gjorde, gå på opdagelse i figentræet. Stammen af figentræet består af fikspunkter og tiltrækkende periodiske baner, da stammen er i intervallet $0 < a < 3$.⁸² Ved a -værdien 3 deles stammen i to og periode-2 punkter opstår. Når periode-2 punkterne bliver frastødende, opstår der periode-4 punkter, som er

⁸² Ibid, s. 23-24.

⁸³ Ibid, s. 26

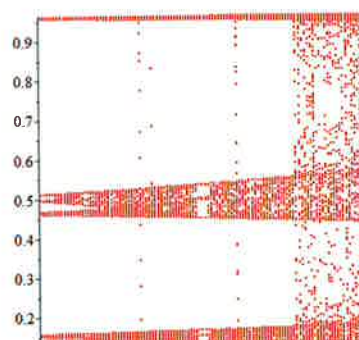
Grunden til, at her itereres først 50 gange uden, at punkterne gemmes, er for at iterationen stabiliserer sig, hvis man ved a -værdien har et fikspunkt eller periodiske punkter. Dermed vil kun x -værdier ved a -værdi med tiltrækkende fikspunkter eller tiltrækkende periodiske blive afsat.⁸²



Figur 23: Skaleret feigenbaumdiagram for den logistiske fremskrivningsfunktioner, frembragt med bilag 16 kode. $0,54 \leq a \leq 3,6$ og $0,324 \leq x \leq 0,6$.



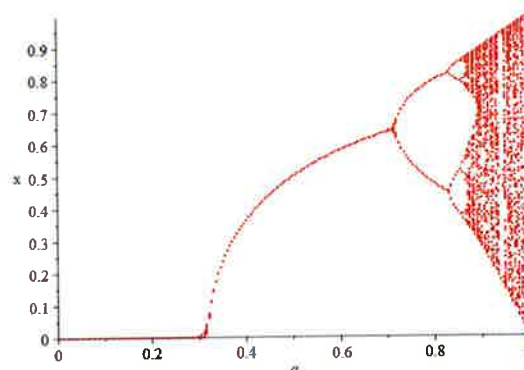
tiltrækkende.⁸⁴ Denne periodefordobling fortsætter i uendelighed, og derfor kan man udforske fraktalet selvtilnærmende egenskaber, hvilket er gjort på figur 23. Men ved de a -værdier, hvor der ikke er en eller flere tiltrækkende periodiske punkter, vandrer punktet rundt for hver iteration i en vandret linje i feigenbaumdiagrammet uden at falde ind i et mønster (kaos), hvilket kan ses på figur 24.⁸⁵ Dette optræder første gang ved den 1. uegentlige attraktor,⁸⁶ man møder i figentræet, Feigenbaum parameteren $a = 3,569 \dots$ ⁸⁷, hvilket er, hvor de uendeligt mange periode-fordoblinger er gennemgået og er blevet frastødende. Det er her kaos opstår. Der findes mange flere af disse uendelige attraktorer i feigenbaumdiagrammet, som ikke vil uddybes her, og mange flere spændende ting i feigenbaumdiagrammet at dykke ned i.⁸⁸



Figur 24: Feigenbaumdiagram for den logistiske fremskrivningsfunktioner. $0,3854 \leq a \leq 3,858$

Figentræer for andre funktioner

Her gælder, at der kan tegnes feigenbaumdiagrammer for andre end for den logistiske forskrift. Der skal dog kunne opstå periodepunkter, da disse er vejen til kaos.⁸⁹ Et Feigenbaumdiagram for funktionen $f(x) = r \cdot \sin(\pi \cdot x)$, ses på figur 25. Her gælder for alle Feigenbaumdiagrammer uanset funktion, at mellem hver periodefordobling er forholdet den såkaldte Feigenbaum konstant $4,669.201.609$ ⁹⁰



Figur 25: Feigenbaumdiagram for $f(x) = r \cdot \sin(\pi \cdot x)$, for $0 \leq a \leq 1$, hvilken fremkaldes af programkoden som ses i bilag 17

Mitchell Feigenbaum arbejdede med ikke-lineære iterationsfunktioner, og specielt fraktalerne, feigenbaumdiagrammerne, gav helt nye konklusioner om fraktalgeometrien og kaos.

⁸⁴ Ibid, s. 20-21

⁸⁵ Ibid, s. 24

⁸⁶ En uendelig attraktor: er en a -værdi der tiltrækker en startværdi, hvor iterationen vil se forvirrende ud og vil bevæge sig kompliceret rundt på attraktoren i et tilfældigt mønster. (Bjørn Felsager og Jonny Schulz, 1990, s. 26-27)

⁸⁷ Bjørn Felsager og Jonny Schulz, 1990, s. 27

⁸⁸ Læs mere om dette i *Figentræer og Mandelbrød* af Bjørn Felsager og Jonny Schulz, s. 26-32

⁸⁹ Bjørn Felsager og Jonny Schulz, 1990, s. 28

⁹⁰ Mogens Esrom Larsen, 2017, Feigenbaums tal.



Feigenbaumdiagrammerne var nogle af de første programmer, hvor man fik en datamaskine til at regne på noget uden et system - kaos⁹¹. Altså har den eksperimenterende tilgang igen været med til at udvikle matematikken. Feigenbaum forenede matematisk analyse og eksperimenterende matematik til at opnå hans resultater, hvilket var helt nyt og var derfor med til at udvikle matematikken.⁹² I den eksperimentelle undersøgelse af fraktalerne, kan man f.eks. studere uendelige periodefordobling i feigenbaumdiagrammerne, samt undersøge utrolig mange andre egenskaber i træernes kroner, hvor kaos opstår.⁹³ Samtidigt tilgås fraktalerne også aksiomatisk deduktivt ved at udfører beviser, for at beskrive fraktalet mere grundigt.

Fraktalernes udvikling, og deres betydning for matematikkens udvikling.

Som det fremgår af opgaven, har de udvalgte fraktaler alle været med til at udvikle fraktalteorien og dermed også matematik-faget. I dette afsnit vil arbejdet med fraktalers betydning for matematikkens udvikling blive undersøgt mere generelt. Fraktalgeometrien læner sig op ad mange emner inden for matematik-faget og var og er med til at udvide flere af emnerne.

Det mest oplagte emne, hvor fraktaler har været med til at udvikle matematik-faget, er geometri. Op til 1900'tallet havde matematik kun omgivet sig med funktioner, som var differentiable, og opfattelsen var, at alle funktioner ville uden tvivl producere sådan en kurve.⁹⁴ Men Kochs kurve, kaotisk data osv. er ikke differentiable kurver, så det var ikke så mærkeligt, at disse kurver blev kaldt monstre.⁹⁵ Dog blev denne opfattelse vendt i det 20. århundrede, og med denne forandring fulgte udvikling af matematik-faget. Her udvikles gennem det 20. århundrede et nyt emne under geometri, fraktalgeometrien. Denne gren af geometrien adskiller sig fx ved at der opstår geometri på en uendelig lille skala, som Kochs snefnug og Mandelbrot-mængden er eksempler på. Den nye geometri var nødvendig, fordi objekterne var for irregulære til at kunne blive beskrevet med den klassiske Euklids geometri. Der var behov for et nyt begrebsark, hvor selvsimilær, iteration, fraktaldimension, dynamiske systemer og kaos spiller centrale roller.

⁹¹ Jonny Hyman, This equation will change how you see the world (the logistic map), time: 5.25-5.34

⁹² Jesper Frandsen, 1990, s. 126

⁹³ Her gennemgås ikke en dybere forklaring, men læs mere i bogen *Figentræer og Mandelbrød* af Bjørn Felsing og Jonny Schulz.

⁹⁴ Holly Trochet, A History of Fractal Geometry, 2009

⁹⁵ Ibid



Dynamiske systemer og kaos er to matematiske områder, som har en rolle i fraktalgeometrien og omvendt. Både i Julia-mængden, Mandelbrot-mængden⁹⁶ og i feigenbaumdiagrammer optræder der kaos. Kaos i de omtalte fraktaler opstår, når iterationener er kaotiske, hvilket er, når to nærliggende begyndelsesværdier ikke giver to baner, som ligner hinanden, hvilket éns intuition siger, de skal. De vil derimod tilsyneladende bevæge sig helt uafhængigt af hinanden⁹⁷. Netop derfor er der stor følsomhed for startværdier, når der er tale om kaos. Overvej blot, at man kun undersøger Julia-mængder med c -værdier, hvis modulo er mindre eller lig med 2, og hvor stor mangfoldighed i udseendet af Julia-mængder er inden for dette lille interval. Men kaos har andre anvendeligheder end at fremstille æstetiske fraktaler som Mandelbrot, Julia-mængder og feigenbaumdiagrammer. Flere dynamiske systemer undersøges med 'Kaos' forskning som f.eks. elektriske kredsløb med selvsvingninger og turbulens.⁹⁸ Altså har fraktalerne haft den betydning for matematik-faget, at her nu er flere muligheder for at arbejde *med* matematikken.

Mandelbrots arbejde tog udgangspunkt i naturens geometri, som er for kompleks til at kunne beskrives med den klassiske geometri.⁹⁹ Hans arbejde med fraktalgeometrien ligger altså selv op til at skulle anvendes til beskrivelse af virkeligheden. F.eks. kystlinjer, som omtales tidligere i opgaven, men også fx skyer, lyn og træer kan fraktalerne være med til at beskrive. I biologisk sammenhæng benyttes fraktaler også som simulering af flere fysiologiske processer i menneskets krop, fx hjertets rytme, som har vist sig at svinge i et tydeligt fraktalt mønster.¹⁰⁰

For at ovenstående er muligt, er computeren helt essentiel.¹⁰¹ Helt banebrydende var Feigenbaums forening af den analyserende matematik og eksperimentelle matematik i fremkaldelse af feigenbaumdiagrammerne. Her forandres matematikken helt, da man førhen ikke havde fremstillet eksperimentelt data i matematikken. Dette benyttes nu i dag i et bredere omfang fx til simulationerne, som omtales ovenfor, men også i fx meteorologi. Matematik-faget ændrede sig, da computeren blev en del af faget, da man i stedet for udelukkende at gå aksiomatisk deduktivt til værks også eksperimenterer for at finde frem til løsninger. Man kan sige, at matematik generelt udvikler sig, mens man prøver at finde svaret på nye matematiske problemer.

⁹⁶ Jonas Lindstrøm Jensen, Fraktaler Mandelbrots Mængde

⁹⁷ Hodgkin, Luke, 2005 s.247

⁹⁸ Jesper Frandsen, 1990, s. 127

⁹⁹ IBM, Fractal Geometry.

¹⁰⁰ Ibid

¹⁰¹ Hodgkin, Luke, 2005 s.247



Konklusion

I gennem udvalgte nedslag i fraktaler og udvalgt teori ses, at fraktalteorien har udviklet sig i det 20. århundrede. De tidlige geometriske fraktaler sierpinski-trekanten og Kochs snefnug lagde de første sten til forståelse af fraktaler og deres komplekse struktur. F.eks. er Kochs snefnug et objekt med uendelig omkreds men et bestemt areal, mens sierpinski-trekanten har en uendelig kantlængde og intet areal. Dette stod i kontrast til den klassiske Euklid geometri og var derfor med til at udvikle matematik-faget. Julia var med til at udvide fraktalteorien til også at indeholde algebraiske fraktaler, da han undersøgte den simple iterationsfunktion $f(x) = z^2 + c$, hvor $c \in \mathbb{C}$. Han undersøgte funktionen rent teoretisk og konkluderede derudfra om f.eks. fikspunkter, og hvorfor netop sætning 5 og 6 er sande. Julia-teorien fik først rigtig betydning i 1980'erne, hvor Mandelbrot fik brug for dem, og først her blev det muligt at undersøge de geometriske figurer eksperimentelt ved at programmere dem. Herefter kunne man inddele dem efter udseende. Der er ingen sætninger for, hvordan en Juliamængde ser ud, og derfor er man nødt til eksperimentelt at undersøge dem. Normalt eksperimenterede man sig ikke frem til resultater og derfor var Juliamængderne med til at udvikle matematik-faget. Mandelbrot udvidede også fraktalteorien ved at definere fraktaldimensionen, som kan benyttes til at beskrive naturlige kurver som kystlinjer. Her er det interessante, at fraktaler ikke har en heltallig dimension og derfor strider imod den normale opfattelse af dimension. Det sidste nedslag, der blev fokuseret på, var feigenbaumdiagrammer, som også var med til at udvikle teorien om fraktaler, fx ved en yderligere forståelse for fixpunkter og deres evne til at være tiltrækkende samt kaos, som opstår ved periodedeling. Matematik-faget ændrede sig med Feigenbaums arbejde, netop fordi han kobede eksperimentel- og analytisk matematik for at fremstille feigenbaumdiagrammer og åbnede op for en helt ny matematik, hvor computeren spiller en stor rolle i at simulere og programmere, og ud fra dette konkludere. Alle fraktalerne, som blev undersøgt i denne opgave, kan undersøges på mindre skalaer via programmering, hvor selvsimilære egenskaber, kaos og andre interessante ting kan undersøges eksperimentelt. Altså havde arbejdet med fraktaler en betydning for udviklingen af matematikfaget i det 20. århundrede.



Litteraturliste

Bøger

Bjørn Felsager og Jonny Schulz. *Figentræer og Manelbrød - et emnehæfte til datalogisk matematik*. Matematiklærerforeningen 1990.

Falconer, Kenneth. *Fractal Geometry Mathematical Foundations and Applications second edition*. Wiley, 2006

Frandsen, Jesper. *Komplekse tal og fraktaler*. Systime, 1. udgave andet oplæg, 1995.

Hodgkin, Luke. *A History of Mathematics. From Mesopotamia to Modernity*, OUP 2005 s.246-248

Mandelbrot, Benoit B., *Fractals form, chance, and dimension*. W. H. Freeman and company, 1977.

Nigel Lesmoir-Gordon, Will Rood og Ralph Edney. *Fractals a graphic guide*. Icon Books Ltd, 2009.

Søren Halse, Erik Laage-Petersen og Jens Peter Touborg, Gyldendals Minilex. *Matematik*. Gyldendal, 1. udgave. 1 oplæg, 2005. s. 139-140

Links

Betten, Anton, *Mathematical Algorithms using Matlab, Maple and C*. 12 marts 2008. Lecture 5: Sierpinski Triangle. s. 80- 85. Besøgt: 27 marts 2020.

https://qmplus.qmul.ac.uk/pluginfile.php/294139/mod_resource/content/1/CH1%20-%20Mathematical%20Algorithms.pdf

Ciavarella, Miriam. *Some Fractals with Maple*. Maple, 2009. Besøgt: 25 marts 2020

<https://www.maplesoft.com/applications/view.aspx?sid=97624&view=html>

Forex Online Magazine, *Fraktaler - Hvad Du Har Brug For At Vide*. Besøgt: 30. marts 2020.

<https://da.vivisoltk.com/1061-forex-fractals-what-you-need-to-know.html>



Hansen, Vagn Lundsgaard. *Geometri*. Den Store Danske, Gyldendal, 2017. Besøgt: 30. marts 2020, [http://denstoredanske.dk/It, teknik og naturvidenskab/Matematik og statistik/Geometri/geometri](http://denstoredanske.dk/It,_teknik_og_naturvidenskab/Matematik_og_statistik/Geometri/geometri)

Hoggard, John. How Long is the Coast of Great Britain? 1997. Besøgt: 27. marts 2020. <http://www.aiecon.org/staff/shc/course/annga/RR/main/How%20Long%20is%20the%20Coast%20of%20Great%20Britain.htm>

Hyman, Jonny. This equation will change how you see the world (the logistic map), YouTube. Besøgt: 3. Marts 2020. <https://www.youtube.com/watch?v=ovJcsL7vyrk>

IBM, Fractal Geometry. Besøgt: 1 april 2020 <https://www.ibm.com/ibm/history/ibm100/us/en/icons/fractal/>

Jensen, Jonas Lindstrøm. Fraktaler Mandelbrots Mængde Institut For Matematiske Fag - Århus Universitet. Besøgt: 3. april 2020. <https://docplayer.dk/5766998-Fraktaler-mandelbrots-maengde.html>

J. J. O'Connor og E. F. Robertson. Gaston Maurice Julia. School of Mathematics and Statistics, University of St Andrews, Scotland, 2008. Besøgt: 28. marts 2020. <http://mathshistory.st-andrews.ac.uk/Biographies/Julia.html>

Jonas Bargum, sidst redigeret d. 30-11-2010. Besøgt d. 30 marts 2020. <https://jonasbargum.weebly.com/fraktaler-generelt.html>



Khan Academy. *Area of Koch snowflake (part 1) - advanced Perimeter, area, and volume*, 5. okt. 2011. Besøgt: 12 marts 2020. https://www.youtube.com/watch?v=64bH_27Ehoc

Khan Academy. *Area of Koch snowflake (part 2) - advanced Perimeter, area, and volume*. 5. okt. 2011. Besøgt: 12 marts 2020. <https://www.youtube.com/watch?v=vBIR2xNAGmo&t=66s>

Mandelbrot, B. B. *How long is the coast of Britain? Statistical self-similarity and fractional dimension*, Science: 156, 1967, 636-638. Besøgt: 20. marts 2020.
https://users.math.yale.edu/~bbm3/web_pdfs/howLongIsTheCoastOfBritain.pdf

Maplesoft. Programming Guide. Besøgt alle SRP'ens dage: ca. 1. jan - 3. april.
<https://www.maplesoft.com/support/help/Maple/view.aspx?path=ProgrammingGuide/Chapter01>

Mogens Esrom Larsen. *Feigenbaums tal*. Den Store Danske, Gyldendal, 2017. Besøgt: 31. marts 2020
[http://denstoredanske.dk/It, teknik og naturvidenskab/Matematik og statistik/Analyse, vektor- og matrixregning og funktionsteori/Feigenbaums tal](http://denstoredanske.dk/It,_teknik_og_naturvidenskab/Matematik_og_statistik/Analyse,_vektor-og_matrixregning_og_funktionsteori/Feigenbaums_tal)

Mogens Esrom Larsen. *Fraktal* i *Den Store Danske*, Gyldendal, 2017. Besøgt: 30. marts 2020.
[http://denstoredanske.dk/It, teknik og naturvidenskab/Matematik og statistik/Geometri/frakta](http://denstoredanske.dk/It,_teknik_og_naturvidenskab/Matematik_og_statistik/Geometri/frakta)
!

Trochet, Holly. *A History of Fractal Geometry*. School of Mathematics and Statistics University of St Andrews, Scotland, 2009. Besøgt: 3. april 2020 <http://mathshistory.st-andrews.ac.uk/HistTopics/fractals.html>



Walz, Alexander F. *Mandelbrot*. MAPLE V FRACTALS, 1999. Besøgt 4. marts 2020.

<http://ftp.informatik.rwth-aachen.de/maple/mfrmand.htm>

Wikipedia, 2019. Besøgt: 1. april 2020. <https://da.wikipedia.org/wiki/Fraktal>

Wright, Francis. *Computing with Maple*, Chapman and Hall/CRC, 2002. Besøgt: 31 marts 2020.

https://books.google.dk/books?id=rt9xKPLCFcgC&pg=PA255&lpg=PA255&dq=koch%27+snowflake+maple+programming&source=bl&ots=ju23N1_z1l&sig=ACfU3U1pEnSw1ZSU_ul4VYcjpCC9LjB3mA&hl=en&sa=X&ved=2ahUKEwjU4K64-cHoAhUJtYsKHZSiDBcQ6AEwDnoECAkQAQ#v=onepage&q=koch'%20snowflake%20maple%20programming&f=false



Bilag

Bilag 1: Programkode til fremkaldelse af 7. generation sierpinski-trekant.

```

> transform:=proc(T,s,x,y)
  return[
    [s*T[1][1] + x, s*T[1][2] + y],
    [s*T[2][1] + x, s*T[2][2] + y],
    [s*T[3][1] + x, s*T[3][2] + y]];
end:
> Sierpinski := proc(n)
  local L, M, i, k;
  L := [[[0, 0], [1, 0], [0.5, 1]]];
  for i to n do M := [];
  for k to nops(L) do M := [op(M),
    transform(L[k], 0.5, 0, 0),
    transform(L[k], 0.5, 0.5, 0),
    transform(L[k], 0.5, 0.25, 0.5)];
  end do;
  L := M;
  end do;
end proc:
> L := Sierpinski(7);
> plots[polygonplot](L, axes = none,
  color = black);

```

Bilag 2: Bruger programmet *transform* på listen T, hvilket består af tre punkter, som udgør en ligesidet trekant, hvis punkterne forbindes

```

> T := PQR:                T := [[0, 0], [1, 0], [0.5, 1]]
> transform(T, 0.5, 0, 0)  [[0., 0.], [0.5, 0.], [0.25, 0.5]]
> transform(T, 0.5, 0.5, 0):  [[0.5, 0.], [1.0, 0.], [0.75, 0.5]]
> transform(T, 0.5, 0.25, 0.5):  [[0.25, 0.5], [0.75, 0.5], [0.50, 1.0]]

```

Bilag 3: Fremgangsmåde til fremkaldelse af 0. generation sierpinski-trekant

```

> P:=[0,0]:
> Q:=[1,0]:
> R:=[0.5,1]:
> PQR:=[P,Q,R]:
> plots[polygonplot]({PQR}, axes = none, color = black, scaling = constrained)

```

Bilag 4: Programkode til fremkaldelse af 1. generation sierpinski-trekant



```

> transform:=proc(T,s,x,y)
  return[
    [s*T[1][1] + x, s*T[1][2] + y],
    [s*T[2][1] + x, s*T[2][2] + y],
    [s*T[3][1] + x, s*T[3][2] + y]];
end:
> Sierpinski := proc(n)
  local L, M, i, k;
  L := [[0, 0], [1, 0], [0.5, 1]];
  for i to n do M := [];
  for k to nops(L) do M := [op(M),
    transform(L[k], 0.5, 0, 0),
    transform(L[k], 0.5, 0.5, 0),
    transform(L[k], 0.5, 0.25, 0.5)];
  end do;
  L := M;
end do;
end proc:
> L := Sierpinski(1);
> plots[polygonplot](L, axes = none,
  color = black);

```

Bilag 5: Programkode til fremkaldelse af 2. generation sierpinski-trekant

```

> transform:=proc(T,s,x,y)
  return[
    [s*T[1][1] + x, s*T[1][2] + y],
    [s*T[2][1] + x, s*T[2][2] + y],
    [s*T[3][1] + x, s*T[3][2] + y]];
end:
> Sierpinski := proc(n)
  local L, M, i, k;
  L := [[0, 0], [1, 0], [0.5, 1]];
  for i to n do M := [];
  for k to nops(L) do M := [op(M),
    transform(L[k], 0.5, 0, 0),
    transform(L[k], 0.5, 0.5, 0),
    transform(L[k], 0.5, 0.25, 0.5)];
  end do;
  L := M;
end do;
end proc:
> L := Sierpinski(2);
> plots[polygonplot](L, axes = none,
  color = black);

```

Bilag 6: Bevis for sætning 2¹⁰²

Lad sidelænden af den ligedannede trekant være p . Dermed er generation 0's kantlængde $3 \cdot p$, 0. generation ses på figur 4. Efter næste iteration, da er der tre ligedannede trekanter med sidelængden $\frac{1}{2} \cdot h$. Derfor er 1. generations kantlængde $3 \cdot \frac{1}{2} \cdot 3 \cdot p$. Efter næste iteration, da er antallet af trekanter tre gange så mange, hvilke alle må have sidelængderne $\left(\frac{1}{2}\right)^2 \cdot p$. Derfor er 2. generations kantlængde $3 \cdot 3 \cdot \left(\frac{1}{2}\right)^2 \cdot 3 \cdot p$. Så ved at betragte de første generationer af sierpinski-

¹⁰² Jesper Frandsen, 1990, s. 61-62



trekanten kan man finde kantlængden K , efter n iterationer med formelen: $K = \left(\frac{3}{2}\right)^n \cdot 3p$. Her bemærkes at $\frac{3}{2} > 0$. Hvis man lader $n \rightarrow \infty$ da går $\left(\frac{3}{2}\right)^n \rightarrow \infty$, hvilket betyder at kantlængden af sierpinski-trekanten er uendeligt.

Bilag 7: Bevis for sætning 3¹⁰³

Omkredsen bestemmes ved at multiplicerer sidelængden s med antallet af sider. Sidelængden af den første trekant er s , og derfor er omkredsen af trekanten i 0. generation $3 \cdot s = L_0$. Efter en iteration, da kommer der yderligere fire sider, hvis sidelængder må være $\frac{s}{3}$, da man deler hver side i tre. Dermed er sidelængden efter første iteration: $L_1 = 3 \cdot 4 \cdot \frac{s}{3} = 3 \cdot s \cdot \frac{4}{3} = \frac{4}{3} \cdot L_0$. Efter anden iteration, dannes 2. generation Kochs snefnug, som ses på figur 11, hvilken må have omkredsen: $L_2 = 3 \cdot 4^2 \cdot \frac{s}{3^2} = 3 \cdot s \cdot \left(\frac{4}{3}\right)^2 = \frac{4}{3} \cdot L_1$. Dette kan man fortsætte med i uendelighed, og derfor må der generelt gælde at $L_n = \frac{4}{3} \cdot L_{n-1}$, hvor n er n antal iterationer. Hvis man lader $n \rightarrow \infty$, da ses at omkredsen bliver $\frac{4}{3}$ gange større for hver iteration. Da $\frac{4}{3} > 1$, da må det betyde, at efter et uendelige antal iterationer, da er omkredsen af Kochs snefnug uendeligt.

Bilag 8:

Arealet (T) af en trekant kan bestemmes med følgende formel:

$$T = \frac{1}{2} \cdot ab \cdot \sin(C)$$

Hvor a og b er længden af kateterne som danner vinkel C i en trekant.

I en ligesidet trekant er alle sidelængderne lig hinanden, og alle vinklerne er $60^\circ = \frac{\pi}{3}$. Derfor må arealet af lige dannet trekant, med sidelængden s , være:

$$T = \frac{1}{2} \cdot s \cdot s \cdot \sin\left(\frac{\pi}{3}\right) = \frac{1}{2} \cdot s^2 \cdot \frac{\sqrt{3}}{2} = \frac{\sqrt{3}}{4} \cdot s^2$$

Derfor kan arealet af en ligesidet trekant med sidelængden s , bestemmes med følgende formel:

¹⁰³ Ibid.



$$T = \frac{\sqrt{3}}{4} \cdot s^2$$

Bilag 9: Sætninger som bruges i bevis¹⁰⁴

Sætning

For $f(z) = z^2 + c$ gælder der at $|f(z)| \geq |z|^2 - |c|$

Bilag 10: Sætninger som bruges i bevis¹⁰⁵

Sætning

Hvis en iterationsfunktion $f(z) = z^2 + c$ har en tiltrækkende periodisk bane, vil den specielt tiltrække den kritiske bane, hvilken har enden nul eller c som begyndelsespunkt.

Bilag 11: Programkode til fremkaldelse af Julia2

```
> restart: with(plots):
> Julia2:=proc(x,y);
  local c, z, m;
  c:=evalf(x+y*I);
  z:=c;
  for m from 0 to 40 while abs(z)<2 do
    z:=z^2-0.195+0.59*I
  od;
  m
end proc:
>
> plot3d(0,-2..2,-1.2..1.2, orientation=[-90, 0], grid=[640,350], style=patchnogrid, color=Julia2);
```

Bilag 12: Programkode til fremkaldelse af Julia3

```
> Julia3:=proc(x,y);
  local c, z, m;
  c:=evalf(x+y*I);
  z:=c;
  for m from 0 to 200 while abs(z)<2 do
    z:=z^2-0.6686866-0.3506868*I
  od;
  m
end proc:
> plot3d(0, -0.3 .. -0.2, -0.2 .. 0, orientation = [-90, 0], grid = [640, 350], style = patchnogrid, color = Julia3)
```

¹⁰⁴ lbis, s. 71

¹⁰⁵ lbis, s. 71



Bilag 13: Programkode til fremkaldelse af Julia4

```

> restart: with(plots):
> Julia4:=proc(x,y);
  local c, z, m;
  c:=evalf(x+y*I);
  z:=c;
  for m from 0 to 30 while abs(z)<2 do
    z:=z^2+I
  od;
  m
end proc:
> plot3d(0,-2.2..2.2,-1.4..1.4, orientation=[-90, 0],grid=[640,350], style=patchnograd, color=Julia4);

```

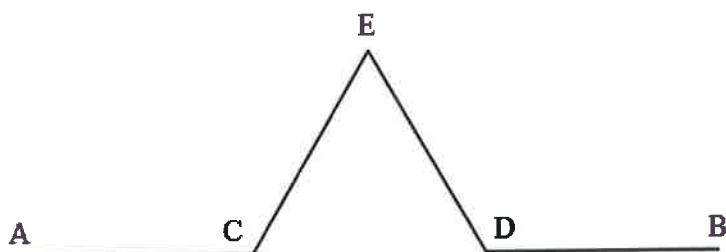
Bilag 14: Programkode til fremkaldelse af Julia4

```

> restart: with(plots):
> Julia5:=proc(x,y);
  local c, z, m;
  c:=evalf(x+y*I);
  z:=c;
  for m from 0 to 60 while abs(z)<2 do
    z:=z^2+0.11-0.67*I
  od;
  m
end proc:
> plot3d(0,-2.2..2.2,-1.4..1.4, orientation=[-90, 0],grid=[640,350], style=patchnograd, color=Julia5);

```

Bilag 15: En side på Kochs snefnug. 1. Generation, samt forklaring af udtryk I programkoden. Kilde: Wright, Francis. Computing with Maple, Chapman and Hall/CRC,2002





```

> Først repræsenteres punkterne A og B som 2-dimensionelle vektorer
> A := vector(2): B := vector(2):
> Punkter C må kunne findes ved at ligge en tredjedel af sidelænden AB til A, så derfor må der gælde:
C := evalm(A + 1/3*(B-A));


$$C := \begin{bmatrix} \frac{2A_1}{3} + \frac{B_1}{3} & \frac{2A_2}{3} + \frac{B_2}{3} \end{bmatrix}$$

> Punkt D må tilsvarende kunne findes bare ved at ligge 2/3 af sidelængden til A, så derfor må der gælde
D := evalm(A + 2/3*(B-A));


$$D := \begin{bmatrix} \frac{A_1}{3} + \frac{2B_1}{3} & \frac{A_2}{3} + \frac{2B_2}{3} \end{bmatrix}$$

> Punktet E må være drejet Pi/3, hvilket svarer til 60°, hvilket er vinklen i en ligesidet trekant. Derfor
defineres en matrix R til at rotere punkt E Pi/3
R := matrix([[cos(Pi/3), -sin(Pi/3)], [sin(Pi/3), cos(Pi/3)]]);


$$R := \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix}$$

> Ud fra figuren ovenfor kan man konstruere punktet E lægge 1/3 af vektor AB, som skal roteres Pi/3, til vektor
C. Derfor gælder her:
E := evalm(C + 1/3*R*(B-A));


$$E := \begin{bmatrix} \frac{B}{6} - \frac{A}{6} + C & -\frac{(\frac{B}{3} - \frac{A}{3})\sqrt{3}}{2} \\ \frac{(\frac{B}{3} - \frac{A}{3})\sqrt{3}}{2} & \frac{B}{6} - \frac{A}{6} + C \end{bmatrix}$$


```

Bilag 16: Program til et feigenbaumdiagram for de logistiske funktionsforskrifter hvor $0,54 \leq a \leq 3,6$

```

>
> Lfigentræ := proc(ned, øvre, pas)
local k, start, a, b, s;
s := {};
a := ned;
while a <= øvre do
start := 0.1;
for k to 50 do
start := a*start*(1 - start);
end do;
for k to 100 do
start := a*start*(1 - start);
s := s union {[a, evalf(start)]};
end do; a := a + pas; end do;
plot([op(s)], 'a' = ned .. øvre,
style = POINT, symbol = POINT, color=red);
end proc;

Lfigentræ(3.54,3.6, 0.001)

```

Bilag 17: Feigenbaumdiagram for $f(x) = r \cdot \sin(\pi \cdot x)$, for $0 \leq a \leq 1$.



```
> Sfigentræ := proc(debut, fin, pas)
  local k, itere, a, b, s;
  s := {};
  a := debut;
  while a <= fin do
    itere := 0.1;
    for k to 50 do
      itere := a*sin(Pi*itere);
    end do;
    for k to 100 do
      itere := a*sin(Pi*itere);
    end do; a := a + pas; end do;
  s := s union {[a, evalf(itere, 4)]};
  plot([op(s)], 'a' = debut .. fin, style = POINT, symbol = POINT, color=red);
end proc;

Sfigentræ(0, 1, 0.005)
```

