

FRAKTALER

Den vakreste delen av matematikken

Innhold

Forord.....	5
Del I	7
En introduksjon (hva er fraktaler).....	8
Cantor-settet	9
Von Koch-kurven.....	10
Pytagoreiske trær	12
Irrasjonale tall	14
Sierpinski-trekanten.....	15
Forandring av reglene i kaos-spillet	17
Rotasjoner i kaos-spillet	20
IFS-fraktaler.....	23
Sammenhengen mellom kaos-spillet og IFS-fraktaler.....	24
Dragekurver.....	30
Konstruksjon av dragekurven.....	30
Den generelle dragekurven.....	32
Komplekse tall	33
Mer om addisjon av komplekse tall	34
Mer om multiplikasjon av komplekse tall.....	34
Mandelbrot-settet.....	37
Julia-settet.....	42
Del II	47
Georg Ferdinand Ludwig Philipp Cantor	48
Niels Fabian Helge von Koch	51
Waclaw Sierpinski.....	53
Gaston Maurice Julia.....	56
Benoit Mandelbrot	57
Tidslinje	59
Del III	60
Dataprogrammer	61
Program 1 – Kaos.bas	61

Program 2 – Mandel.bas	62
Program 3 – Julia.bas	63
Program 4 – Koch.bas.....	64
Program 5 – Bregne.bas	65
Program 6 – Pythagor.bas	67
Program 7 – Gendragn.bas.....	69
Program 8 – Determ.bas	71
Program 9 – Rotasjon.bas	72
Ordforklaringer	74
Kilder.....	76
Stikkordregister	78

Big fleas have little fleas
on their backs to bite them,
and little fleas have lesser fleas,
and so ad infinitum.

Jonathan Swift, 1733

Forord

Fraktaler er etter min mening den vakreste delen av matematikken. De færreste tenker på matematikk som noe spesielt vakkert (selv om det kan være morsomt å holde på med), men fraktaler er virkelig et veldig godt eksempel på at matematikk er vakkert.

Denne årsoppgaven er foreløpig siste del av et arbeid med fraktaler som jeg begynte på for mange år siden. Det begynte med et program, `fern.bas` (fern = bregne) som fulgte med bladet PC Plus i mars 1994. Endelsen `.bas` betyr at det er et Basic-program. Fra samme blad fikk jeg også `mandel.bas` (forkortelse for Mandelbrot), men det tok fryktelig lang tid å kjøre programmet ferdig! Grunnen er at det skal mange beregninger til, og datamaskinen min hadde bare en 25 MHz prosessor. Men på den tida hadde jeg ikke de matematikkunnskapene jeg har i dag, og jeg visste heller ikke mye om verken bregner eller Mandelbrot. Bladet er nå sporløst forsvunnet.

Noen år senere støtte jeg på et bilde av en annen bregne som var mye mer naturtro. Jeg fikk interesse for temaet igjen. Den gang husket jeg ikke i detalj hvordan bregnen fra programmet så ut, og til min skuffelse kunne jeg konstatere at programmet mitt ikke lagde en så fin bregne. Da jeg fikk tilgang til internett, var det med ett mye lettere å få informasjon om bregner og andre fraktaler. Men hvordan de vakre mønstrene oppstod, stod ingen steder! Det er greit nok å se hva programmet foretar seg, men hvorfor det blir så fint resultat er det ikke lett å gi noen forklaring på.

I januar 2000 lånte jeg boka «Fractals Everywhere» etter å ha fått vite om den på internett, i håp om å endelig forstå hva som skjedde, og hvorfor bregnene oppstod. Det gjorde jeg imidlertid ikke. Da jeg bestemte meg for dette temaet i årsoppgaven, lånte jeg denne boka på nytt. Senere fant jeg ut at det var lettere å bli klok av det som stod i boka «Fractals for the Classroom», etter tips fra Mats (matematikklæreren).

Heldigvis fant jeg en god internettside om Mandelbrot-settet. Den var laget av Frode Gill, ved Høgskolen i Agder, som klart og tydelig forklarte matematikken som må til for å få datamaskinen til å vise bildet. Uten den kunnskapen hadde jeg ikke vært i stand til å forstå teorien som ligger bak. På siden fant jeg også et Pascal-program som tegnet Mandelbrot-settet. Jeg kjente til programmering i Delphi, som er en slags Windows-variant av Pascal, så jeg laget et Delphi-program som i tillegg til å vise fraktalen, også kunne zoome inn.

Jeg har valgt å dele denne oppgaven inn i tre deler. I Del I tar jeg for meg det faglige stoffet. I Del II står biografier til betydningsfulle matematikere som har jobbet med fraktaler. Jeg tok med disse for å sette det hele i et mer historisk perspektiv, og for å gi leseren mer enn bare faglig kunnskap. Bakerst, i Del III, har jeg samlet programmer som får datamaskinen til å tegne fraktaler i tillegg til ordforklaringer, kildehenvisninger og stikkordregister.

Ord som er satt i *kursiv* blir forklart nærmere i ordforklaringene på side 74. Dette gjelder bare ord i Del I.

Som praktisk del har jeg valgt å tegne en blomkål, som er et eksempel på en naturlig ting med fraktalegenskaper.

Jeg vil takke Mats Fallenius, Hanne Gilbo, Tone Sølberg, Bente Kari Gulbrandsen og Gunnar Andreas Berg for god hjelp med arbeidet.

Diktet på side 4 er hentet fra kilde **[14]**.

Håper du får glede av å lese oppgaven!

Moss, 1. mars 2003
Kristoffer Andreas Berg

Del I

Faglig del

En introduksjon (hva er fraktaler)

Ordet «fraktal» kommer fra fransk og er avledet av latin «fractus», perfektum partisipp av «frangere» (bryte). Kilde [20].

En beskrivelse som passer for fraktaler, er at de er bilder der motivet «gjentar seg selv» i mindre og mindre størrelse. En fraktal kan forstørres uendelig mange ganger, men bildet blir ikke mindre detaljert. Von Koch-kurven er et eksempel på det. Fraktaler kan uttrykkes som en matematisk formel.

Noen fraktaler likner på ting som finnes i naturen, for eksempel bregner og snøkrystaller.

Fraktaler er en relativt ny grein innen matematikken. Von Koch-kurven ble for eksempel oppdaget i 1904. Mandelbrot-settet ble oppdaget rundt 1970 og fraktale bregner ble oppdaget i 1980-årene.

Det finnes mange typer fraktaler. Her er noen:

- ◆ Cantor-settet
- ◆ Von Koch-kurven
- ◆ Pytagoreiske trær
- ◆ Bregner, Sierpinski-trekanten og andre *IFS-fraktaler*
- ◆ Dragekurver
- ◆ Mandelbrot-settet
- ◆ Julia-settet

Cantor-settet

En av de enkleste fraktalene ble laget av tyskeren Georg Cantor. Bildet under viser Cantor-settet.

Når du skal lage den, gjør du slik:

1. tegn et linjestykke
2. del linjestykket i tre like deler
3. fjern den midtre delen
4. del alle de nye linjestykkene du får i tre

Trinnene 3–4 kan gjentas så lenge du vil.



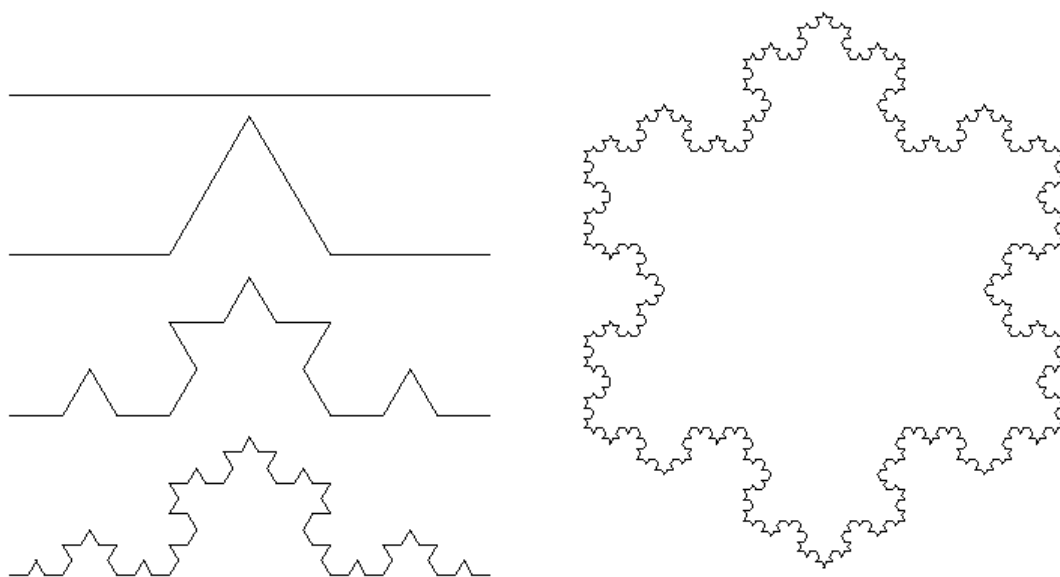
Dette bildet, kjent som Cantor-Dust (dust = støv), har jeg konstruert i *AutoSketch*. Jeg har stoppet etter fem trinn, men egentlig fortsetter det i det uendelige.

Von Koch-kurven

En annen fraktal er von Koch-kurven. Den ble først laget av svensken Helge von Koch i 1904. Du kan lage den på denne måten:

1. Tegn et rett linjestykke
2. Del det i tre
3. Fjern den midtre delen
4. Erstatt den med en likesidet trekant uten grunnlinje
5. Gjenta trinn 2–5 med hvert av de fire nye linjestykkene.

Hvis du gjentar delingene i det uendelige, avtar bildet aldri i detaljrikhet. Når vi setter sammen tre slike kurver til en «likesidet trekant», får vi noe som kan likne en snøkrystall.



Disse bildene er laget ved hjelp av et programmet på side 64.

Til sammenlikning, tar jeg også med et bilde av en ekte snøkrystall. Kilde [15]



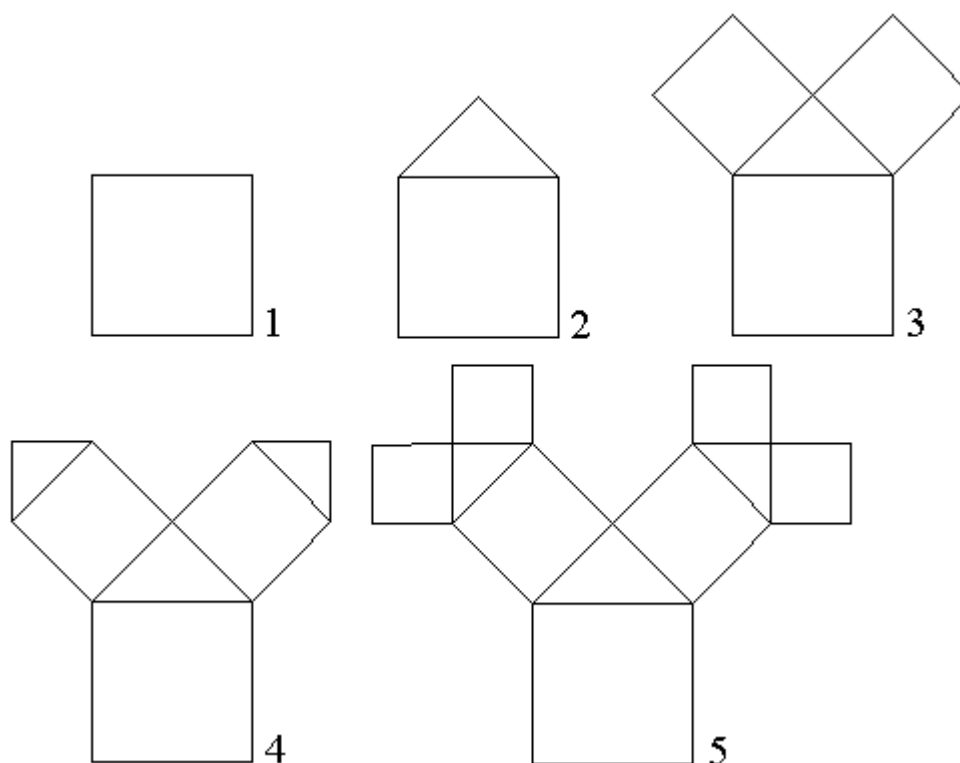
Pytagoreiske trær

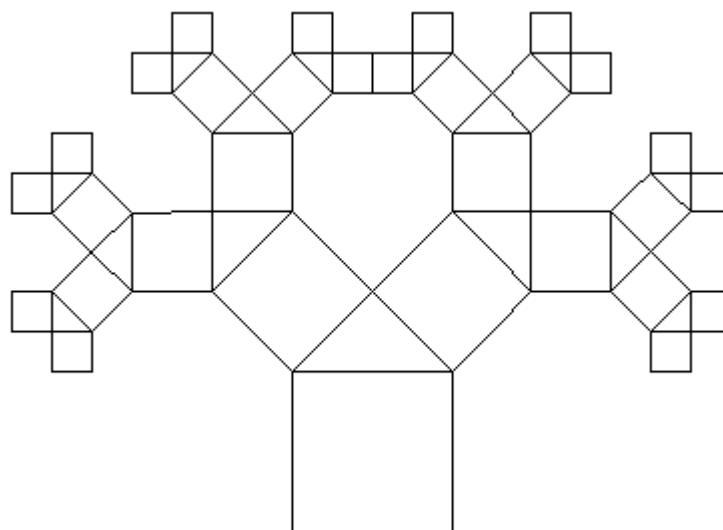
Pythagoras, som døde i begynnelsen av det femte århundret f. Kr., var kjent blant sine samtidige, og senere også av Aristoteles, som grunnlegger av et religiøst brorskap i Sør-Italia, der pytagoreerne spilte en viktig politisk rolle i det sjette århundret f. Kr.. Forbindelsen mellom navnet hans og *den pytagoreiske læresetningen* er imidlertid av nyere dato og ganske feilaktig. Setningen var faktisk kjent lenge før Pythagoras levde. En annen viktig oppdagelse, tilskrevet Pythagoras, eller iallfall skolen hans, er at siden og diagonalen i et kvadrat er inkommensurable, dvs. at forholdet mellom diagonalen og siden ikke er lik forholdet mellom to heltall.

Slik kan du lage et pytagoreisk tre:

1. Tegn et kvadrat.
2. Tilføy en rettvinklet trekant hvor hypotenusen faller sammen med en av sidene i kvadratet.
3. Tilføy kvadrater på de ledige sidene i trekanten.
4. Tilføy to rettvinklede trekanter.
5. Tilføy fire kvadrater.
6. Tilføy fire rettvinklede trekanter.
7. Tilføy åtte kvadrater.

Denne figuren viser bare de fem første trinnene, men viser klart og tydelig prinsippet.

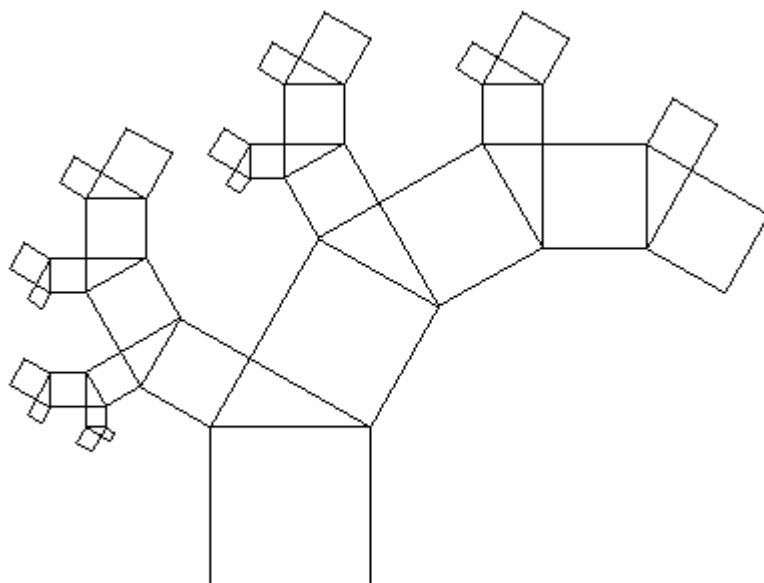




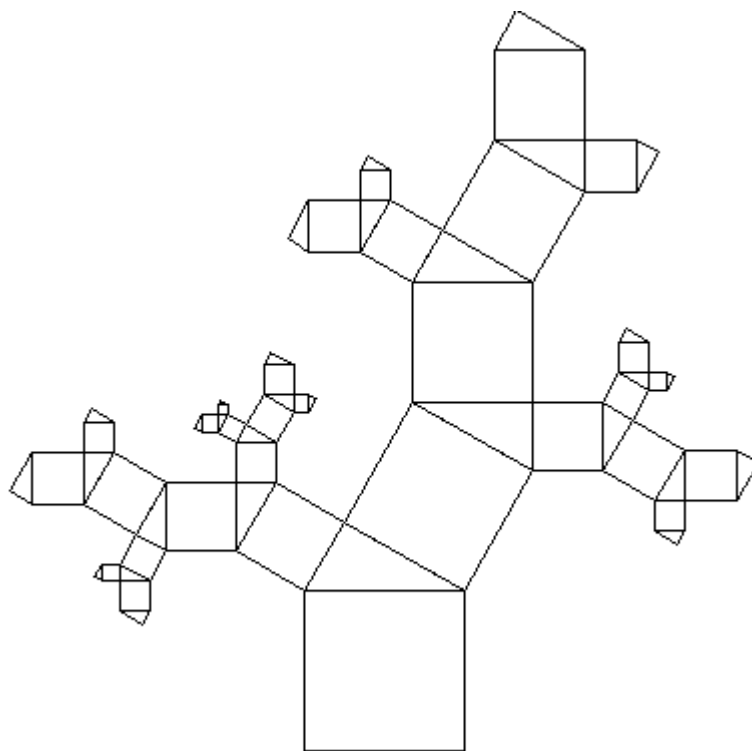
Et pytagoreisk tre er et eksempel på en enkel fraktal som du kan tegne selv.

Det er ikke nødvendig at firkantene er kvadratiske, og trekantene trenger ikke å være rettvinklede.

Slik kan det se ut hvis vi bruker trekanter som ikke er likebeinte:



Vi kan også gjøre den variasjonen at vi speilvender trekantene for hver ny grein. Da kan vi få et resultat som vist på figuren. Kilde [2]



Du finner et program som lager pytagoreiske trær på side 67.

Irrasjonale tall

Oppdagelsen av at forholdet mellom diagonalen og siden ikke er lik forholdet mellom to heltall, skapte behovet for å utvide tallsystemet til *irrasjonale tall*. Lengden av diagonalen i et enhetskvadrat (et kvadrat med sidelengde 1), $\sqrt{2}$, er irrasjonal. Her kommer en begrunnelse. Anta at p og q er positive heltall, hvor $\sqrt{2} = p/q$. Vi kan også anta at p og q ikke har noen felles divisor. Dermed blir $p^2 = 2q^2$, altså må p^2 være et partall. Men dette medfører at p selv må være et partall, siden kvadratet av et oddetall også er et oddetall. Derfor blir $p = 2r$. Tallet r er et naturlig tall, og brukes bare for å vise at $2r$ da blir et partall. Men da betyr $p^2 = 2q^2$ at $4r^2 = 2q^2$ eller $2r^2 = q^2$, som betyr at q også må være et partall. Men dette motsier antagelsen om at p og q ikke har noen felles divisor. Derfor er $\sqrt{2}$ irrasjonal. Dette beviset finnes i Euklids tiende bok rundt 300 f. Kr.

Kilde [16]

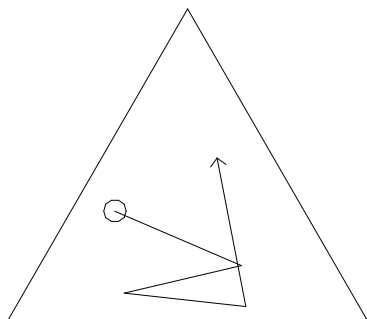
Sierpinski-trekanten

Sierpinski-trekanten ble først konstruert av Waclaw Sierpinski. Den kan beregnes på to måter.

Den deterministiske algoritmen: Lag en trekant. Del den i fire like trekanter ved å forbinde midtpunktene av sidene med linjestykker. Fjern trekanten i midten. Fortsett med å dele de tre trekantene som ligger i hjørnene. Gjenta delingen i det uendelige. Se programmet på side 71.

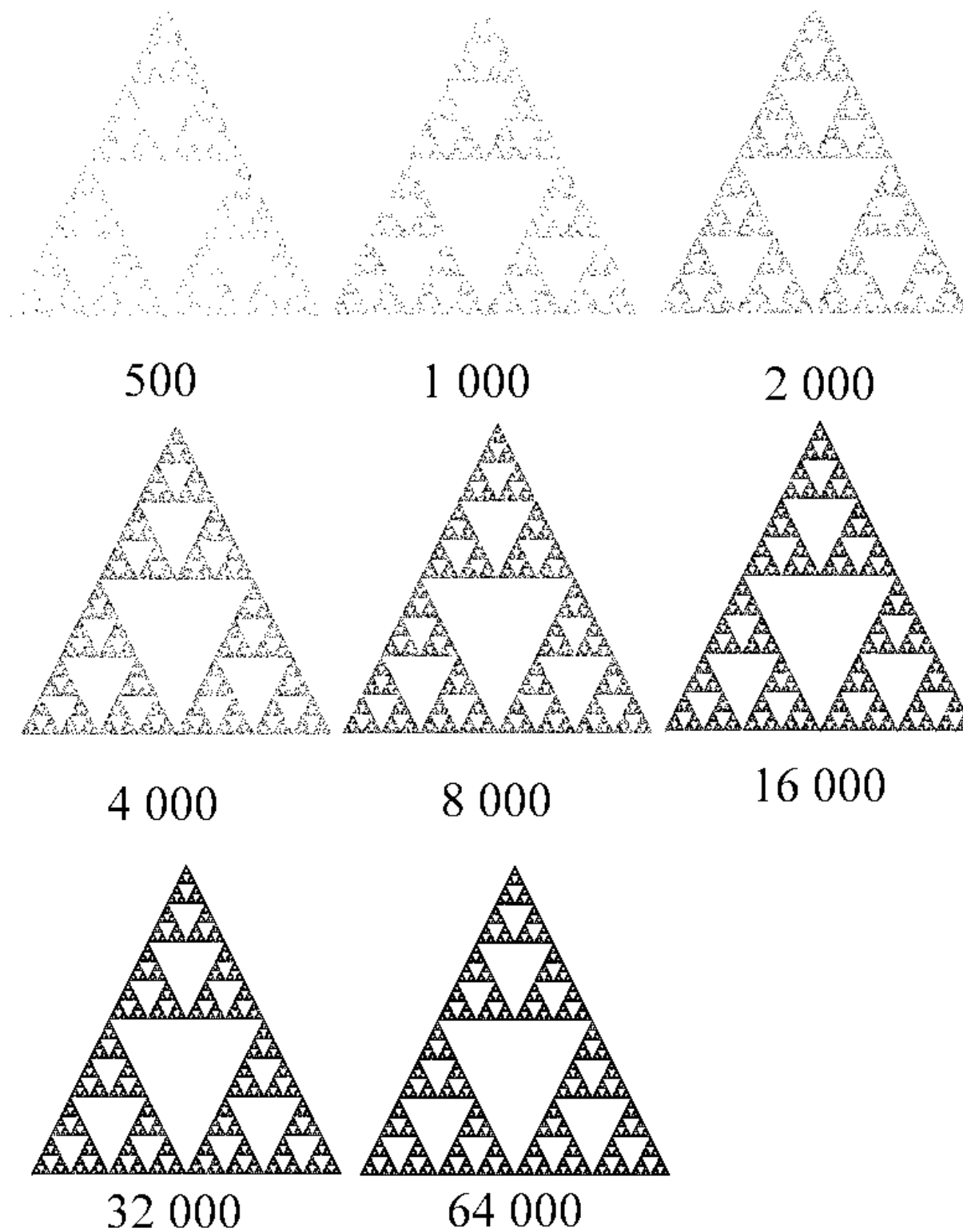


Kaos-spillet: Velg tre punkter og marker dem som prikker. Dette blir hjørnene i trekanten. Tegn et punkt (det kan godt være utenfor trekanten). Velg tilfeldig ett av hjørnene og marker et punkt halvveis mellom hjørnet og det siste punktet du tegnet. Punktene må markeres som prikker, ikke kryss. Velg så et nytt hjørne (det kan godt være det samme hjørnet som sist) og gjenta prosessen. Det var Michael Fielding Barnsley som introduserte ordet «kaos-spill», uten at det behøver å bety at det var han som først oppdaget det. Se programmet på side 61.



Denne figuren viser de første fem mulige punktene. Utgangspunktet er markert med en sirkel. Pilhodet viser det femte punktet. Du skal egentlig ikke tegne strekene, men jeg har gjort det her for å markere rekkefølgen av punktene, og for å gjøre det lettere å se hvilket hjørne hvert enkelt punkt beveger seg halvveis mot.

Når du tegner en million punkter, får du et veldig detaljert bilde. (Når jeg sier «du», mener jeg ikke at du skal sitte i ukevis. Du kan jo bruke en datamaskin.)

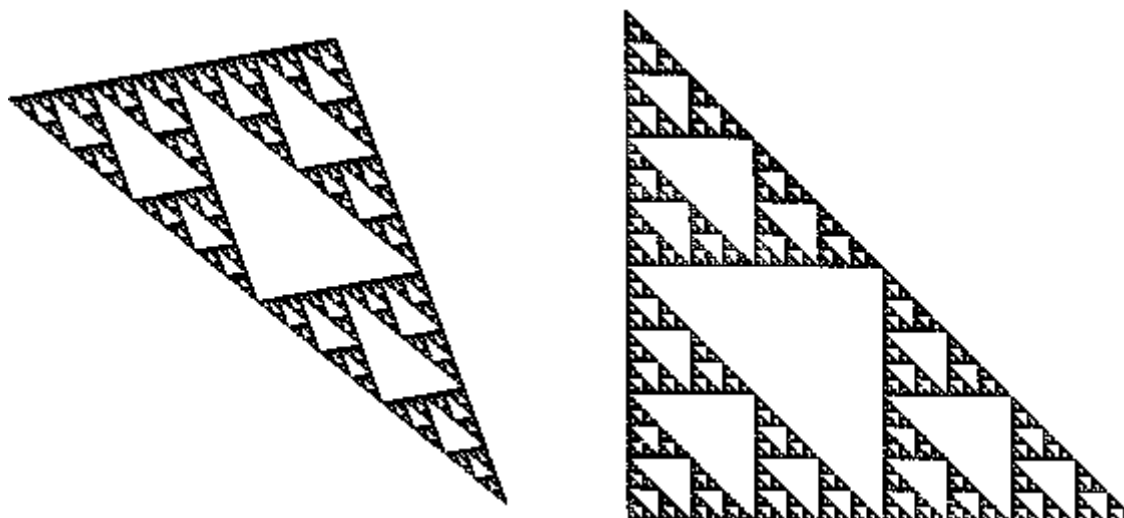


Denne bildeserien viser resultatet av kaos-spillet. Antall fullførte *iterasjoner* står under hvert bilde. Et program som kan brukes finner du på side 61. Kilde [25]

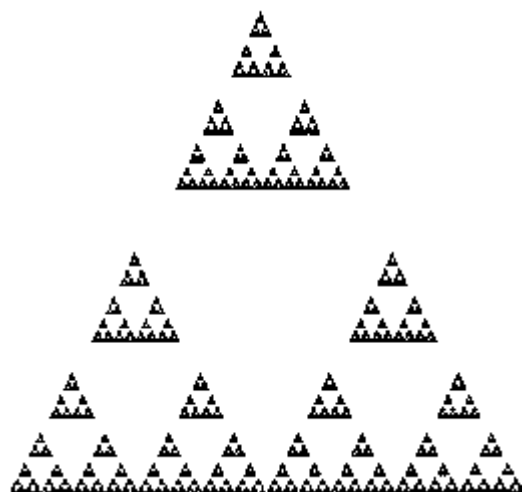
Forandring av reglene i kaos-spillet

Hvis vi velger hjørnepunktene til en likesidet trekant og spiller kaos-spillet, ved å flytte halvveis mot et tilfeldig hjørne hver gang, oppstår Sierpinski-trekanten S . Denne figuren består av 3 selv-liknende kopier av seg selv, og hver kopi er halvparten så stor som S (altså med skaleringsfaktor $1/2$).

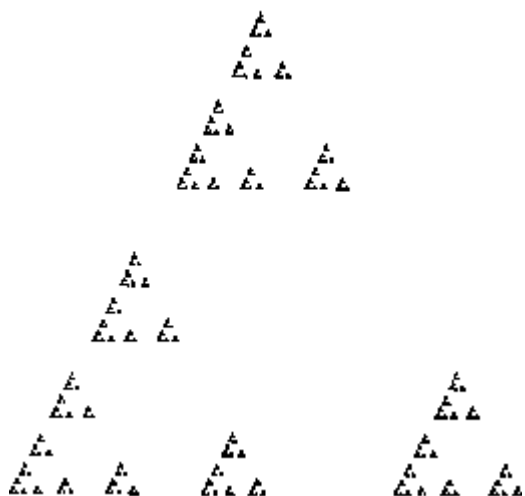
Vi burde egentlig kalt S den likesidede Sierpinski-trekanten, siden det er andre Sierpinski-trekanter. Hvis vi begynner med hjørnepunktene i en rettvinklet trekant eller en trekant med en stump vinkel, vil ulike fraktaler oppstå. Se figuren. Imidlertid består hvert av disse bildene av 3 selv-liknende kopier, som hver har en skaleringsfaktor på $1/2$.



Vi kan variere reglene i kaos-spillet. Vi prøver med hjørnepunktene til en likesidet trekant. Vi begynner med å bevege oss fra et tilfeldig valgt punkt mot et av trekantens hjørner. Hvilket av dem, kan vi bestemme med f. eks. en terning. Når vi går mot et av de to nedre hjørnene, halverer vi avstanden til hjørnet og avsetter et punkt. Når vi går mot topphjørnet, avsettes punktet på $2/3$ av avstanden (altså slik at avstanden fra det nye punktet til toppunktet er $1/3$ av avstanden fra det opprinnelige punktet til toppunktet). Spiller vi kaos-spillet med disse reglene, vil vi få et resultat som vist på neste bilde. Bildet består av 3 selv-liknende biter, hvorav 2 er nøyaktig halvparten så store som hele bildet, mens den andre er $1/3$ av størrelsen.

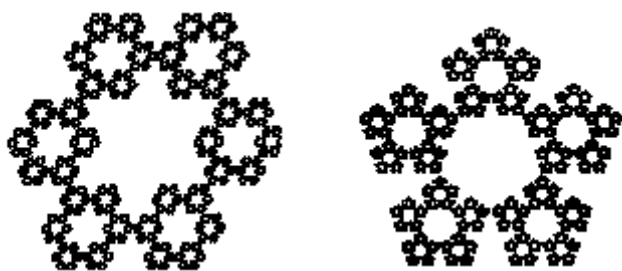


Kaos-spillet med 3 hjørner og skaleringsfaktorer $1/2$, $1/3$ og $1/2$.



Her brukte vi 3 hjørner og skaleringsfaktorer $1/2$, $1/3$ og $1/4$ for å lage bildet.

Vi kan også variere antall hjørner. Vi velger hjørnepunktene til en regulær sekskant, og flytter $1/3$ av avstanden mot det tilhørende hjørne ved hvert terningkast. Da får vi et bilde som vist på neste figur. Dette er et naturlig kaos-spill, siden vi kan nummerere hjørnene fra 1 til 6, og bruke et tilfeldig tall til å bestemme hvilket hjørne vi skal flytte mot. Vi kaller dette bildet Sierpinski-sekskanten. Den består av 6 selv-liknende bilder, hver med skaleringsfaktor $1/3$.

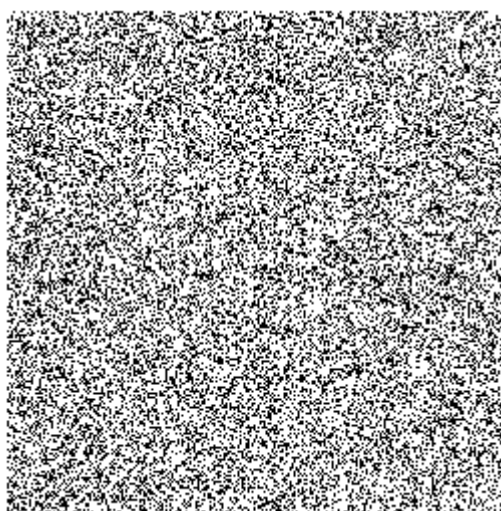


Sierpinski-sekskant og -femkant

Det finnes også en annen berømt fraktal gjemt inni Sierpinski-sekskanten. De indre grensene av figuren viser von Koch-kurven.

Hvis vi velger hjørnene i en regulær femkant og går $3/8$ mot et tilfeldig hjørne, oppstår Sierpinski-femkanten. Her ser vi 5 selv-liknende kopier, hver på $1/3$ av originalens størrelse.

Nå blir det interessant å gjette hvilke figurer som vil oppstå hvis vi bruker punktene på hjørnene av et kvadrat og en skaleringsfaktor på $1/2$. Mange vil nå forvente et Sierpinski-kvadrat. Feil! Punktene fyller ut hele kvadratet når vi spiller med disse reglene. Og det er ikke noen overraskelse, for et kvadrat er et selv-liknende bilde som kan deles i 4 selv-liknende bilder, hver med skaleringsfaktor $1/2$. Nok en gang kan vi avlese spillets regler ut av bildet.



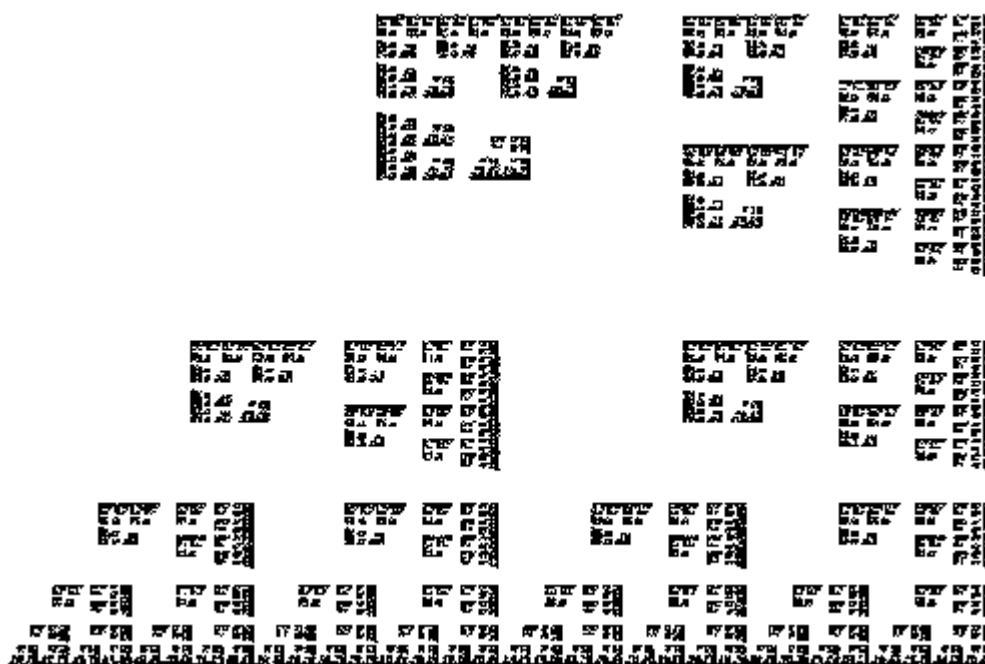
Kilde [3]

Rotasjoner i kaos-spillet

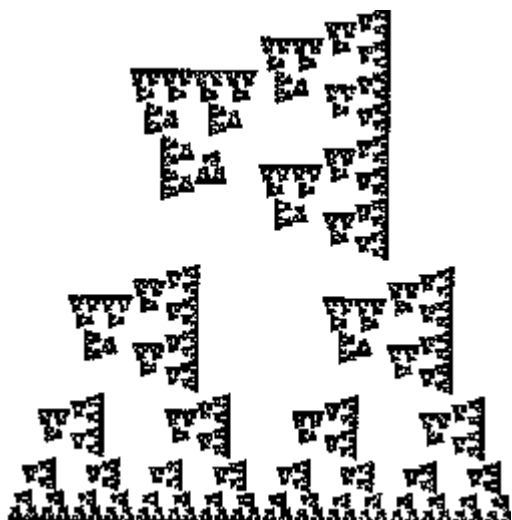
Dette kapitlet om Rotasjoner i kaos-spillet har jeg hentet fra

<http://math.bu.edu/DYSYS/chaos-game/node8.html>

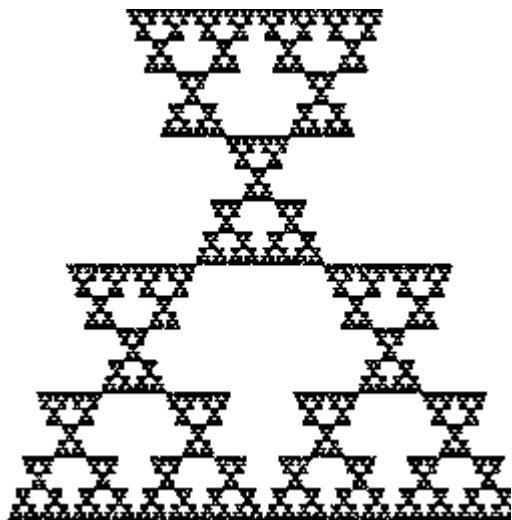
Jeg har oversatt det til norsk, men har kommet til den konklusjon at hvis man følger opplysningene i teksten til punkt og prikke, vil man ikke få et resultat som ser ut som figuren på neste side. For å utforske temaet, laget jeg et program i QBasic som fulgte oppskriften nevnt nedenfor. Programmet (Rotasjon.bas) står på side 72. Den riktige figuren ser slik ut.



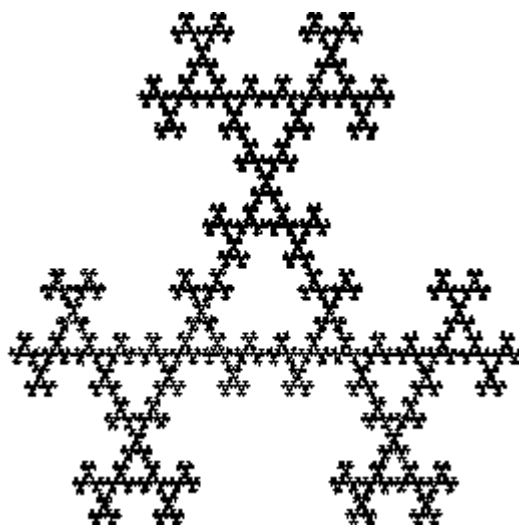
En annen, mer komplisert utgave av kaos-spillet oppstår når vi tillater rotasjoner i tillegg til forflytninger mot bestemte hjørnepunkter. For eksempel, anta at vi starter med hjørnene i en likesidet trekant. Som før flytter vi halvveis mot det valgte hjørnet. Men når det øverste hjørnet blir valgt, flytter vi først halvveis mot hjørnet, og så roterer vi punktet 90 grader om dette hjørnet i retning mot klokka. Vi kunne egentlig like godt rotere først, og så flytte halvveis mot hjørnet. Hvis vi spiller kaos-spillet med disse reglene, får vi et resultat som vist på neste figur. Denne figuren består av 3 selv-liknende deler, og hver er halvparten så stor som hele figuren, men den øverste delen er rotert 90 grader mot klokka. Det er ikke så lett å forutsi hva slags figur vi får når vi spiller kaos-spillet med disse nye reglene, men med en gang vi ser den, er det ingen spørsmål om hvilket spill vi brukte for å lage den.



Kaos-spillet med 3 hjørner, skaleringsfaktor $1/2$, og en 90 graders rotasjon om det øverste hjørnet



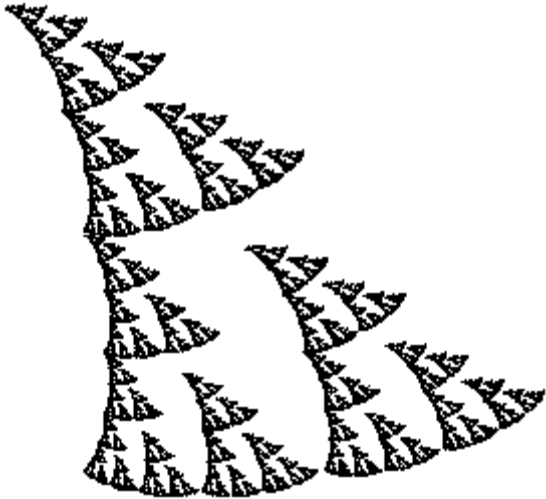
Kaos-spillet med 3 hjørner, skaleringsfaktor $1/2$, og en 180 graders rotasjon om det øverste hjørnet



Dette bildet ble laget med 180-graders rotasjoner om alle hjørnene.

Med tilgang til godt datautstyr, kan du lage en fraktal-animasjon. Dette kan gjøres ved å bruke kaos-spillet til å lage noen fraktaler med liten forandring av parameterne (rotasjoner og skaleringsfaktorer) fra et bilde til neste bilde.

Kilde [4]



Dette er en forblåst Sierpinski-trekant. Jeg laget den ved å bruke kaos-spillet, med rotasjoner på 10 grader om det øvre og det høyre hjørnet og -10 grader om det venstre.

IFS-fraktaler

Når vi skal lage en IFS-fraktal, har vi to variabler, x og y . Når vi starter, lar vi $x = 0$ og $y = 0$. Så trenger vi et tilfeldig tall i området $[0, 1]$. Avhengig av hva tallet blir, tar vi seks konstanter, a , b , c , d , e , og f , og beregner verdiene x_1 og y_1 på denne måten:

$$x_1 = ax + by + e$$

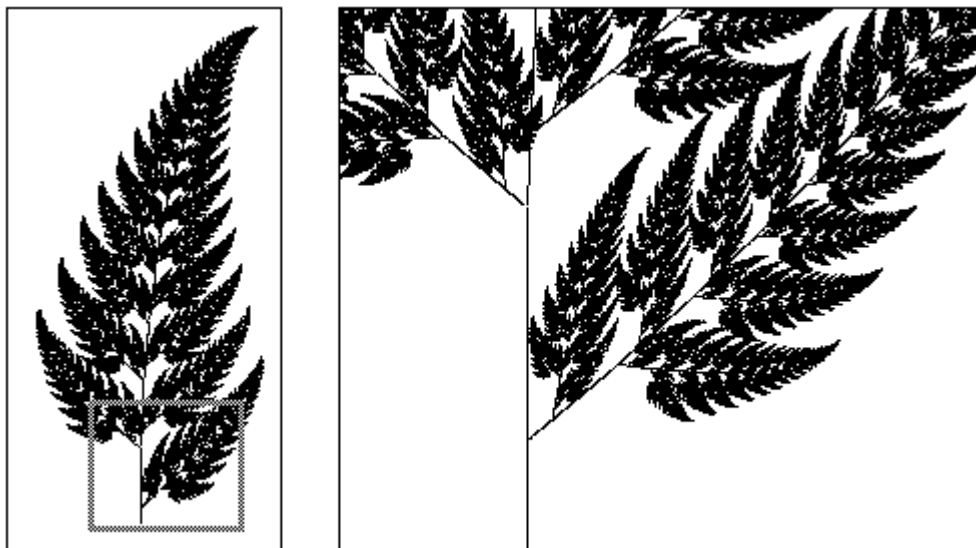
$$y_1 = cx + dy + f$$

Punktet (x, y) plottes i et koordinatsystem. Så lar vi $x = x_1$ og $y = y_1$, og gjentar prosessen ved å få et nytt tilfeldig tall. Hver beregning av x og y kalles en *iterasjon*.

De forskjellige verdiene til konstantene a , b , c , d , e , og f , er på forhånd bestemt. For å lage en bregne, kan vi bruke disse verdiene:

	a	b	c	d	e	f	p
1	0	0	0	0,16	0	0	0,01
2	0,85	0,04	-0,04	0,85	0	1,6	0,85
3	0,2	-0,26	0,23	0,22	0	1,6	0,07
4	-0,15	0,28	0,26	0,24	0	0,44	0,07

Tallene i den siste kolonnen angir som en sannsynlighet hvor ofte raden skal brukes. Den første raden skal brukes i 1 % av tilfellene (1 % = 0,01). Den andre raden skal brukes i 85 % av tilfellene, siden 85 % = 0,85. Den tredje raden skal brukes i 7 % av tilfellene, og det skal også den fjerde raden.



Dette er bregnen som skjuler seg bak tabellen ovenfor. Bregnen går under navnet Black Spleenwort-bregnen, og den kalles også Barnsleys bregne (eng. Barnsley's Fern). Det høyre bildet er et forstørret bilde av det venstre. Hvert eneste bregneblad består av små bregner, som igjen er oppbygd av bitte små bregner selv. På side 65 finner du et program som kan brukes. Kilde [17] og [13]

Sammenhengen mellom kaos-spillet og IFS-fraktaler

Reglene i det opprinnelige kaos-spillet (uten rotasjoner) kan på en enkel måte overføres til en konstanttabell for en IFS-fraktal. Slik ser reglene for en Sierpinski-trekant ut, når de er skrevet på IFS-form:

	a	b	c	d	e	f	p
1	0,5	0	0	0,5	0	0	0,33
2	0,5	0	0	0,5	0,5	0	0,33
3	0,5	0	0	0,5	0,25	0,433	0,34

Hvis vi for eksempel har punktet $(0,4, 0,2)$ og skal flytte det halvveis mot punktet $(1, 0)$, altså det nedre høyre hjørnet, må vi beregne gjennomsnittet av x -koordinatene og y -koordinatene hver for seg. Den nye x -koordinaten blir $(0,4 + 1) / 2$. Det kan også skrives $0,4 / 2 + 1 / 2$. Du ser at vi må dele x -koordinaten på 2 og så legge til halvparten av x -koordinaten til hjørnet. Vi følger et tilsvarende resonnement når det gjelder y -koordinaten.

Den nye x -koordinaten blir $x_1 = 0,5 \cdot x + 0,5 \cdot 1$. Siden vi vil ha uttrykket på formen

$$x_1 = ax + by + e$$

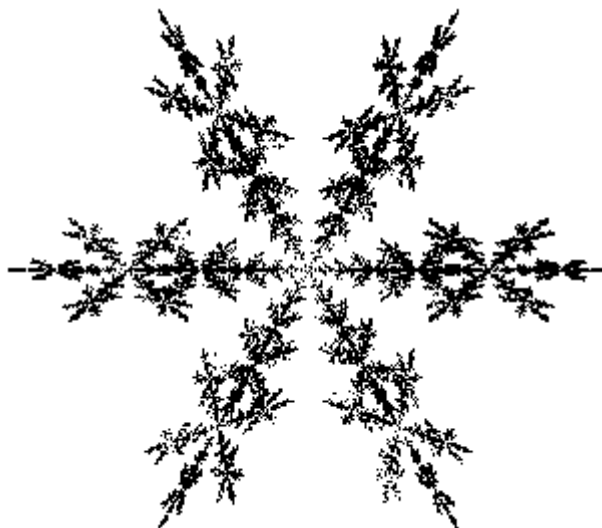
ser vi at $a = 0,5$, $b = 0$ og $e = 0,5$. På samme måte finner vi at $c = 0$, $d = 0,5$ og $f = 0$.

Disse tallene finner vi igjen i linje 2 i tabellen ovenfor.

Dette er forklaringen på hvordan vi kom fram til tallene i tabellen ovenfor. Tallet 0,433 er halvparten av 0,866, som er en tilnærming for $\frac{1}{2}\sqrt{3}$. Dette er høyden i en likesidet trekant med sidelengde 1.

Denne figuren viser en fraktal snøkrystall (som ikke er von Koch-kurven).

Bildet består av 50 000 beregnede punkter (iterasjoner).

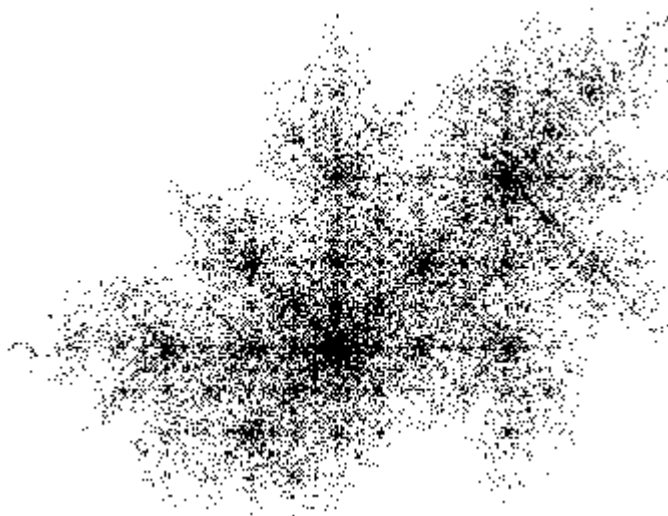


IFS-koden for denne fraktalen er

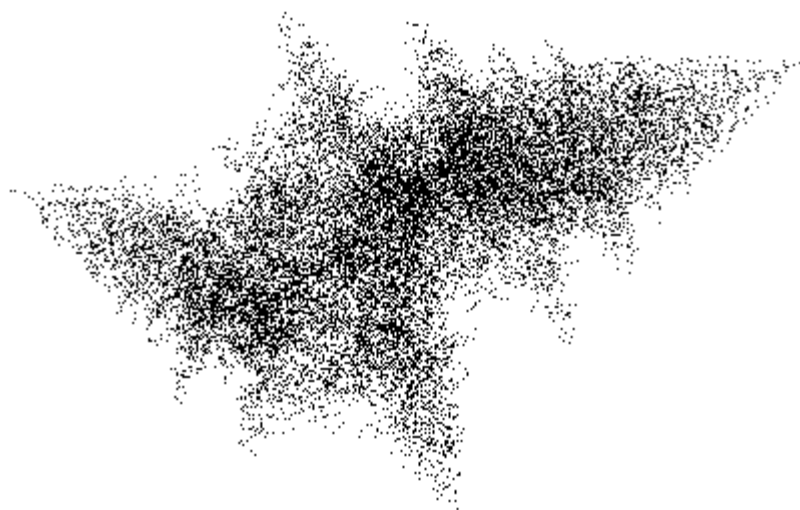
	a	b	c	d	e	f	p
1	0,6	0	0	0,6	0	0	0,05
2	0,4	0	0	0,2	0,6	0	0,15
3	0,5	-0,866	0,866	0,5	0	0	0,80

Kilde [11]

De to neste fraktalene kan likne på galakser. Begge «galaksene» er laget med 20 000 *iterasjoner*.



	a	b	c	d	e	f	p
1	-0,5	0,5	-0,5	-0,5	1	1	0,33
2	-0,5	-0,5	-0,5	0,5	0	0	0,33
3	0,5	-0,5	0,5	0,5	0	0	0,34



	a	b	c	d	e	f	p
1	0,5	0,5	0,5	-0,5	0	1	0,33
2	0,5	0,5	0,5	-0,5	1	1	0,33
3	-0,5	-0,5	0,5	-0,5	0,5	0	0,34

Her er noen IFS-fraktaler og deres tilhørende IFS-koder.



	a	b	c	d	e	f	p
1	0	0	0	0,25	0	-0,14	0,02
2	0,85	0,02	-0,02	0,83	0	1	0,84
3	0,09	-0,28	0,3	0,11	0	0,6	0,07
4	-0,09	0,28	0,3	0,09	0	0,6	0,07

Kilde [14]



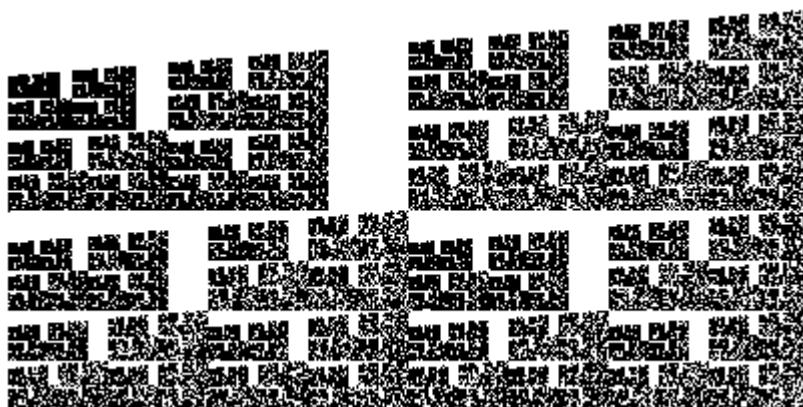
	a	b	c	d	e	f	p
1	0	0	0	0,2	4,85	0,65	0,05
2	-0,143	0,4	0,025	0,333	5,25	0,75	0,1
3	0,25	-0,4	0,04	0,167	3,95	1,75	0,1
4	0,85	0,04	-0,02	0,85	0,7	1,5	0,75

Kilde [24]



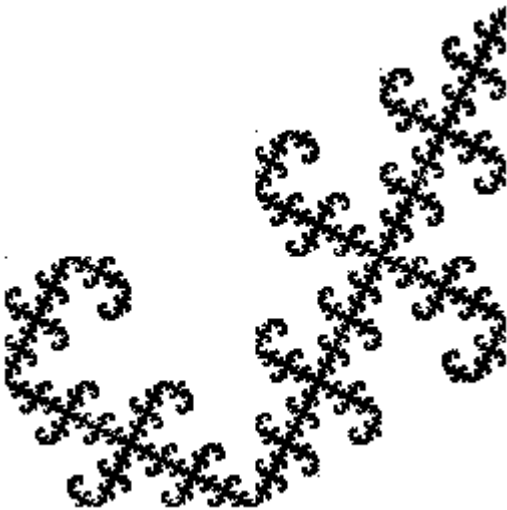
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>p</i>
1	0	0	0	0,32	0,5	0	0,02
2	-0,134	0,263	0,246	0,224	0,57	0,36	0,15
3	0,17	-0,215	0,222	0,176	0,408	0,893	0,13
4	0,781	0,034	-0,032	0,739	0,1075	2,70	0,7

På en internettside fant jeg noen tall som skulle lage denne bregnen. Men tallene laget ingen bregne, så jeg måtte prøve meg fram og forandre tallene for å komme fram til noe som liknet en bregne.



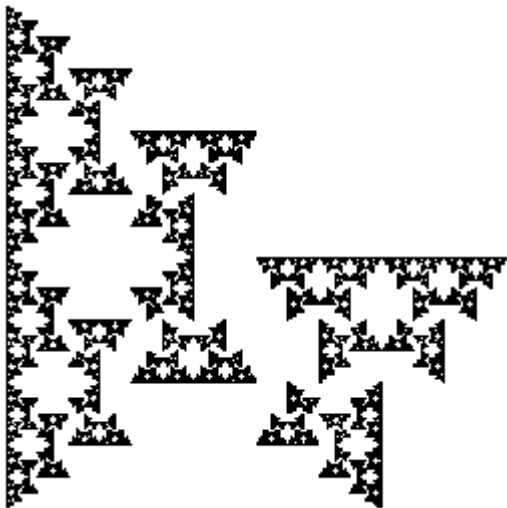
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>p</i>
1	0,5	0	0	0,5	0	0	0,25
2	0,5	0	0	0,5	2	0	0,25
3	0,4	0	0	0,4	0	1	0,25
4	0,5	0	0	0,5	2	1	0,25

Kilde [17]



	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>p</i>
1	0	0,5	−0,5	0	0	0,5	0,33
2	−0,5	0	0	−0,5	1	0,5	0,33
3	0,5	0	0	0,5	0,5	0,5	0,34

Kilde [12]

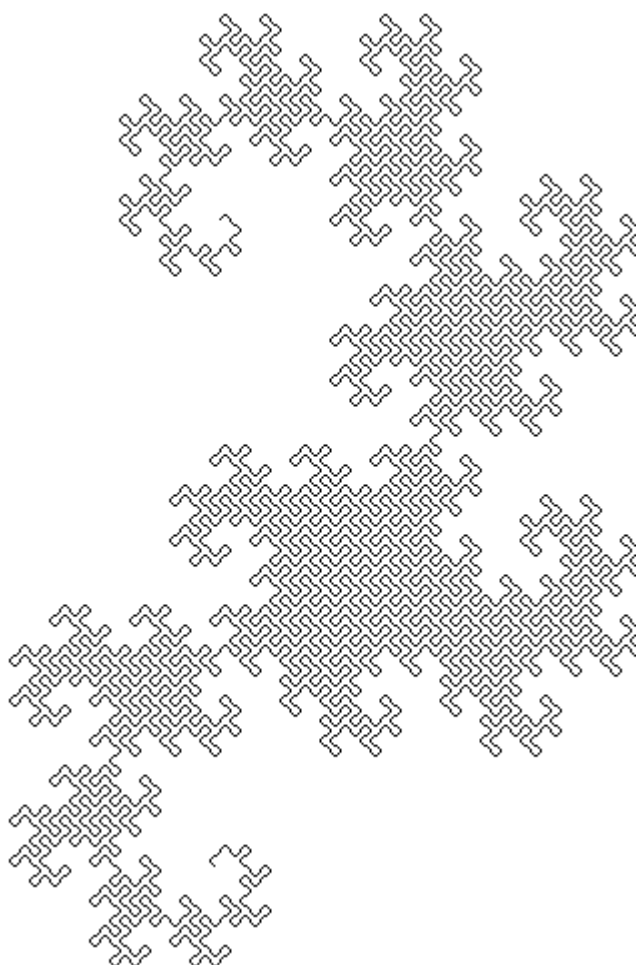


	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>p</i>
1	0,5	0	0	−0,5	0	0,5	0,33
2	0	0,5	−0,5	0	0,5	0,5	0,33
3	0,5	0	0	0,5	0	0,5	0,34

Kilde [12]

Dragekurver

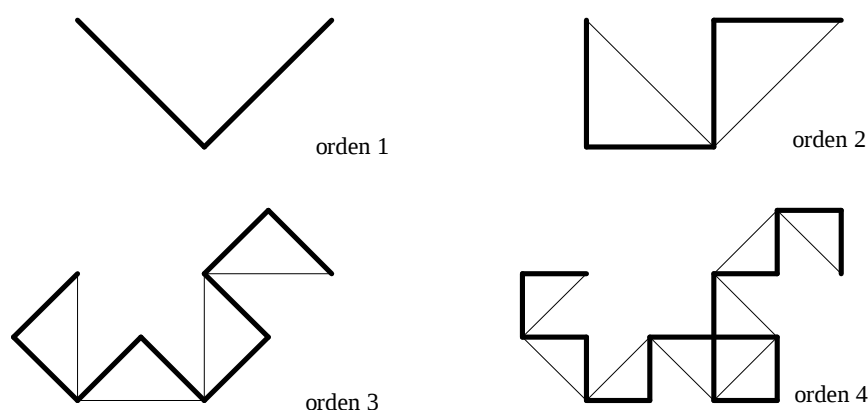
Martin Gardners spalte «Mathematical Games» i Scientific American, nå (1983) Douglas Hofstadters spalte «Metamagical Themas», var en skattkiste med ideer for personer med data som hobby. For noen år siden beskrev Gardner et datamaskin-utskrevet eksempel på noe han kalte en dragekurve (se figuren). Hvis du bruker fantasien, kan du se likheten mellom den og den klassiske orientalske dragen – det var slik den fikk navnet sitt. Donald Knuth, en matematiker og dataekspert, har gjort et stort arbeid med å dokumentere dragekurven og forholdet til tallteorien. Knuth ble så imponert av dragekurven at han gjenskapte mønsteret i keramiske fliser på gårdsplassen sin.



Her kommer en beskrivelse av teorien bak dragekurven, måter å konstruere den på, og forholdet til det bredere generaliserte dragemønsteret.

Konstruksjon av dragekurven

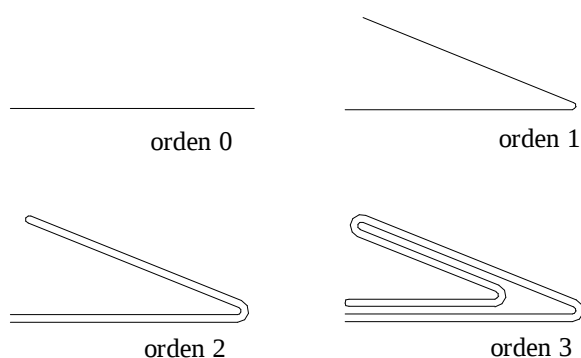
Dragekurver kan lages i forskjellige størrelser (ordener). Det er mange måter å konstruere disse kurvene på. De to første metodene jeg beskriver, vil hjelpe deg til å visualisere prosessen. De to siste er algoritmer som passer for dataprogrammering.



En måte å definere en dragekurve geometrisk er vist i figuren over. Start med en rett vinkel. Dette er en kurve av orden 1. Fortsett med å erstatte hver av linjene i vinkelen med mindre rette vinkler for å lage en kurve av orden 2. For å lage større dragekurver, kan du bruke den samme framgangsmåten. Ved hvert trinn, erstatt n rette linjer med n rette vinkler for å lage en dragekurve av orden $n + 1$.

En annen måte å lage enkle dragekurver av lave ordener på, er å brette en tynn papirstrimmel flere ganger. Denne ideen var grunnlaget for fysikeren John E. Heighways oppdagelse av dragekurven. Se for deg en papirstrimmel – en dragekurve av orden 0. Brett den en gang på midten, og du har en dragekurve av orden 1. Ved å brette strimmelen på midten n ganger, alltid i samme retning, vil du skape en dragekurve av orden n (se figuren under). Fordi denne operasjonen deler papiret i $2n$ deler, vil en utbrettet strimmel av orden n ha $2^n - 1$ rettvinklede bretter. (Et gammelt ord sier at, uavhengig av tykkelse og lengde, kan ingen papirstrimmel brettes mer enn sju ganger, men for denne drøftingen ignorerer vi denne fysiske begrensningen.)

Etter at du har brettet ferdig, kan du brette ut strimmelen, så brettene danner rette vinkler. Den utbrettede strimmelen vil ha et mønster med venstre- og høyrevendinger i tilsynelatende tilfeldige retninger.



Dragesekvensen blir lettere å beskrive når vi betegner en venstrebredd med V, en høyrebredd med H og gir strengen (bokstavsekvensen med V-er og H-er) navnet S. Hvis vi velger V som den første breddingen, blir strengene av laveste ordener beskrevet som følger:

$$S(1) = V$$

$$S(2) = VVH$$

$$S(3) = VVHVVHH$$

$$S(4) = VVHVVHHVHVHHVHH$$

Konstruksjon av dragesekvenser med en datamaskin dreier seg om å manipulere V- og H-strengene. Legg merke til at $S(n + 1)$ er et overordnet sett til sekvensen $S(n)$. Mønsteret som oppstår, kan bli *ekstrapolert* med en av de følgende rekursive (gjentakende) teknikkene:

Algoritme 1: Legg en V til strengen $S(n)$. Legg så til strengen oppnådd ved å invertere tegnet i midten av $S(n)$ (hvis tegnet er en H, så la det bli en V og omvendt). På denne måten blir $S(3)$ sammensatt av $VVH + V + VHH$, siden $S(2)$ er VVH .

Algoritme 2: Legg en V til $S(n)$, og legg så til den inverterte og reverserte sekvensen $\overline{S(n)}$. For eksempel, fordi $S(3) = VVHVVHH$, blir den inverterte strengen $HHVHHVV$. Ved å snu rekkefølgen på bokstavene slik at den første blir sist, og den nest første blir nest sist, får vi strengen $VVHHVHH$. Altså blir $S(4)$ da $S(3) + V + \overline{S(3)}$ eller $VVHVVHH + V + VVHHVHH$. Resultatet kan sammenliknes med å plassere to drager av orden n snute mot snute. Dette produserer en drage av orden $n + 1$. Symbolsk likner denne algoritmen mest på å brette en papirstrimmel, og er den metoden som er brukt i Basic-programmet som lager dragekurver. Dragene som blir laget med en av disse algoritmene har en del interessante egenskaper: linja som tegner en drage vil aldri krysse seg selv, det vil aldri skje at flere enn to vinkler møtes i samme punkt, og ingen del av linja vil noensinne bli tegnet to ganger.

Den generelle dragekurven

Inntil nå har vi bare omtalt den «klassiske» dragekurven, den figuren som oppstår når papirstrimmelen hele tida brettes til venstre. Det er lett å se at brette til høyre skaper et speilbilde av den klassiske dragen («hodet» blir «halen» og omvendt). Vi kan også veksle på retningene vi bretter.

Hvis vi går et trinn videre, kan det bevises at å brette papiret i tilfeldige retninger, vil lage et mønster med de samme drage-liknende egenskapene.

Kilde [23]

Komplekse tall

Før vi går i gang med Mandelbrot- og Julia-settene, bør du kjenne til komplekse tall. Vi starter med en rask gjennomgang av tallmengder.

N Naturlige tall er alle hele tall større enn null.

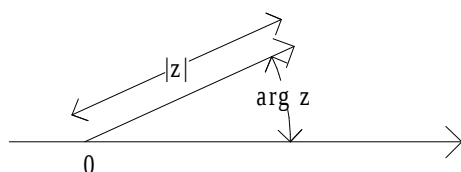
Z Hele tall er alle positive hele tall, negative hele tall og null.

Q Rasjonale tall er alle tall som kan skrives som en brøk med hele tall i telleren og nevneren.

R Reelle tall er alle rasjonale og irrasjonale tall.

Både rasjonale og irrasjonale tall kan plasseres på tallinja. Det finnes likevel en tallmengde til, i tillegg til de fire vi har sett på. Tallene kalles komplekse tall, og betegnes med $\mathbb{C}$. Disse tallene ligger ikke på tallinja, men utenfor!

Vi tenker oss tallinja plassert i planet, og en *vektor* som starter i 0 på tallinja. Vektoren kan beskrives ved lengden $|z|$ og en vinkel φ . Vi kaller $|z|$ absoluttverdien av z . Vinkelen φ kalles argumentet til z og skrives $\arg z$. Den regnes i positiv omløpsretning (mot klokka) fra den positive delen av tallinja og måles i *radianer*.



Komplekse tall adderes som vektorer, og multipliseres ved å multiplisere absoluttverdiene og addere argumentene. Disse to setningene utleder vi senere.

De komplekse tallene innbefatter på en naturlig måte de reelle tallene. For eksempel kan vi tenke på -1 som et komplekst tall med absoluttverdi 1 og argument π .

Det komplekse tallet som har absoluttverdi 1 og argument $\pi/2$ spiller en så viktig rolle at vi gir det et eget symbol, nemlig i . Siden i^2 har absoluttverdi $1 \cdot 1 = 1$ og argument $\pi/2 + \pi/2 = \pi$, så har vi $i^2 = -1$.

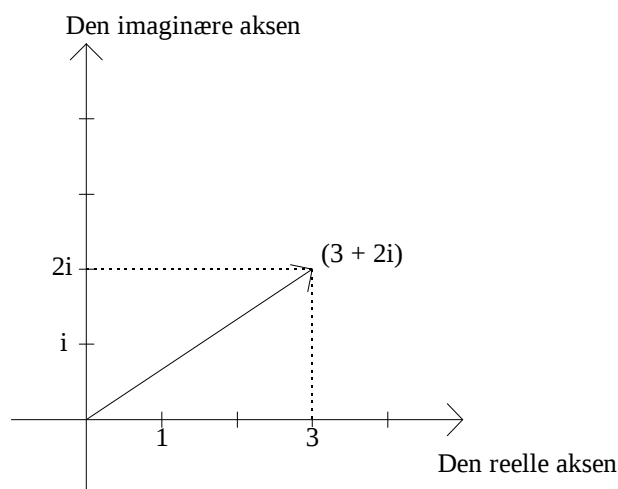
Når vi utvider tallbegrepet til komplekse tall, lar derfor likningen $x^2 + 1 = 0$ seg løse. En løsning er i , og en annen er $-i$. Vi kaller i den *imaginære* enhet.

Vi har definert komplekse tall som vektorer i planet og som starter i 0. Slike vektorer er entydig bestemt av sitt endepunkt. Komplekse tall kan vi derfor godt tenke på som punkter i planet, akkurat som vi kan tenke på reelle tall som punkter på tallinja. Ethvert komplekst tall z kan vi skrive på formen $z = a + bi$ der a og b er reelle tall og i er den imaginære enhet. Dette er en vanlig skrivemåte for komplekse tall. Som vist på neste figur, kan a og b oppfattes som koordinatene til vektoren (eller punktet) z i forhold til et *ortonormert koordinatsystem*.

Vi kaller a realdelen og bi imaginærdelen til z . Vi skriver

$$\operatorname{Re} z = a$$

$$\operatorname{Im} z = bi$$



Kilde [19]

Resten av kapitlet har jeg laget selv.

Når vi skal multiplisere to tall på formen $(a + bi)$ med hverandre, multipliserer vi ut parentesene på samme måte som vi gjør med vanlige tall.

Eksempel

$$\begin{aligned}
 (3 + 5i)(2 + 7i) &= 3 \cdot 2 + 3 \cdot 7i + 5i \cdot 2 + 5i \cdot 7i \\
 &= 6 + 21i + 10i + 35i^2 \\
 &= 6 + 31i - 35 \\
 &= -29 + 31i
 \end{aligned}$$

Kvadrering av et komplekst tall kan gjøres ved hjelp av første kvadratsetning, som vanligvis skrives $(a + b)^2 = a^2 + 2ab + b^2$. I dette tilfellet får vi $(a + bi)^2 = a^2 + 2abi + (bi)^2 = a^2 - b^2 + 2abi$.

Mer om addisjon av komplekse tall

Når vi skal addere de to komplekse tallene $a + bi$ og $c + di$, adderer vi realdelene og imaginærdelene hver for seg og får $a + c + (b + d)i$. Vektorer adderes ved å addere x - og y -komponentene (som tilsvarer realdelen og imaginærdelen til et komplekst tall) hver for seg. Siden metoden for å addere komplekse tall er lik den for å addere vektorer, ser vi at komplekse tall kan adderes som vektorer.

Mer om multiplikasjon av komplekse tall

Vi tenker oss to komplekse tall, $z = a + bi$ og $y = c + di$. Når vi multipliserer ut parentesene, finner vi at produktet blir $(a + bi)(c + di) = ac - bd + (ad + bc)i$.

Vektoren til det komplekse tallet z , x -aksen og linja fra vektorens endepunkt til x -aksen danner en rettvinklet trekant. Tangens til argumentet er imaginærdelen delt på realdelen. Ut fra definisjonen av sinus og cosinus, finner vi

$$a = |z| \cdot \cos \arg z$$

og

$$b = |z| \cdot \sin \arg z$$

Altså får vi realdelen ved å multiplisere absoluttverdien med cosinus til argumentet, og imaginærdelen ved å multiplisere absoluttverdien med sinus til argumentet.

Nå ser vi på tangens til argumentet til produktet av de to komplekse tallene.

$$\tan \arg(a + bi)(c + di) = \frac{ad + bc}{ac - bd}$$

Nå kan vi bytte ut a med $|z| \cdot \cos \arg z$ og b med $|z| \cdot \sin \arg z$.

Vi kan også bytte ut c med $|y| \cdot \cos \arg y$ og d med $|y| \cdot \sin \arg y$. Da får vi

$$\tan \arg zy = \frac{|z| \cdot \cos \arg z \cdot |y| \cdot \sin \arg y + |z| \cdot \sin \arg z \cdot |y| \cdot \cos \arg y}{|z| \cdot \cos \arg z \cdot |y| \cdot \cos \arg y - |z| \cdot \sin \arg z \cdot |y| \cdot \sin \arg y}$$

Vi forkorter brøken med $|z| \cdot |y|$ og får

$$\tan \arg zy = \frac{\cos \arg z \cdot \sin \arg y + \sin \arg z \cdot \cos \arg y}{\cos \arg z \cdot \cos \arg y - \sin \arg z \cdot \sin \arg y}$$

Vi kjenner igjen formlene for sinus og cosinus til summen av to vinkler, og får

$$\tan \arg zy = \frac{\sin \arg(z + y)}{\cos \arg(z + y)} = \tan \arg(z + y)$$

Nå gjenstår å vise at absoluttverdien til produktet er lik produktet av absoluttverdien til tallene.

Absoluttverdien til et komplekst tall er lengden av vektoren fra origo til punktet (a, b) , og kan beregnes ved hjelp av den pytagoreiske læresetningen. Vi får $|a + bi| = \sqrt{a^2 + b^2}$.

Absoluttverdien til produktet blir

$$\begin{aligned} |zy| &= |ac + adi + bci - bd| = |ac - bd + (ad + bc)i| \\ &= \sqrt{(ac - bd)^2 + (ad + bc)^2} \\ &= \sqrt{a^2c^2 - 2abcd + b^2d^2 + a^2d^2 + 2abcd + b^2c^2} \\ &= \sqrt{a^2c^2 + a^2d^2 + b^2c^2 + b^2d^2} \end{aligned}$$

Produktet av absoluttverdiene blir

$$|x| \cdot |y| = \sqrt{a^2 + b^2} \cdot \sqrt{c^2 + d^2} = \sqrt{(a^2 + b^2)(c^2 + d^2)} = \sqrt{a^2 c^2 + a^2 d^2 + b^2 c^2 + b^2 d^2}$$

Vi ser at de to uttrykkene er like. Vi har dermed bevist regelen for multiplikasjon av komplekse tall.

Mandelbrot-settet

Mandelbrot-settet beregnes etter formelen $Z_{(n+1)} = (Z_{(n)})^2 + C$. Funksjonen Z og konstanten C er komplekse tall. I Mandelbrot-settet er $Z_{(0)} = 0$, og C er et punkt i det komplekse planet. Det komplekse planet visualiseres med et koordinatsystem med en reell akse a og en imaginær akse b . Da representerer hvert punkt i koordinatsystemet et tall på formen $a + bi$, hvor i er den *imaginære* enhet, nemlig $\sqrt{-1}$.

Når settet skal beregnes, må hver *piksel* beregnes for seg. Vi bruker formelen ovenfor, og setter realdelen til C lik a -koordinaten til pikselen, og imaginærdelen til C lik b -koordinaten til pikselen.

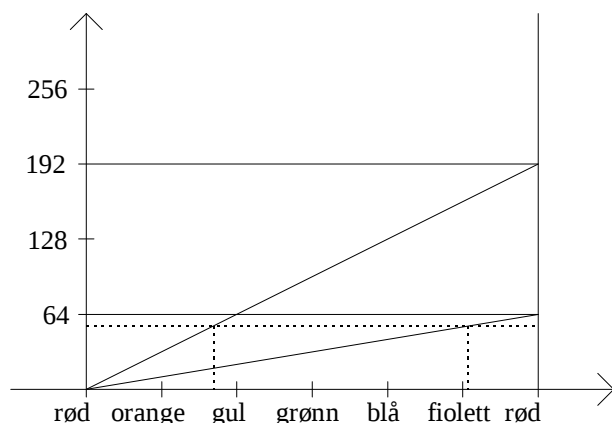
Fargen til pikselen blir bestemt av hvor mange *iterasjoner* som må til for at verdien skal gå mot uendelig. Men når går verdien mot uendelig?

Den dårlige nyheten: Du kan ikke undersøke om den er uendelig.

Den gode nyheten: Hvis avstanden til origo overstiger 2, vet du at den (en eller annen gang) vil bli uendelig.

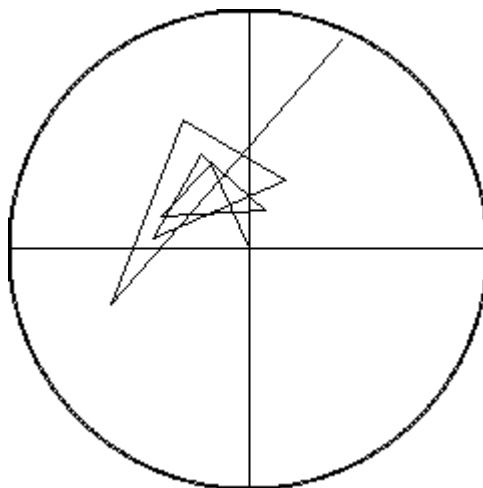
For noen verdier av C vil verdien aldri bli større enn 2. Derfor må vi sette en grense for hvor lenge vi skal holde på før vi går videre til neste piksel. En piksel der beregningene blir avbrutt, blir svart.

Kilde [10]

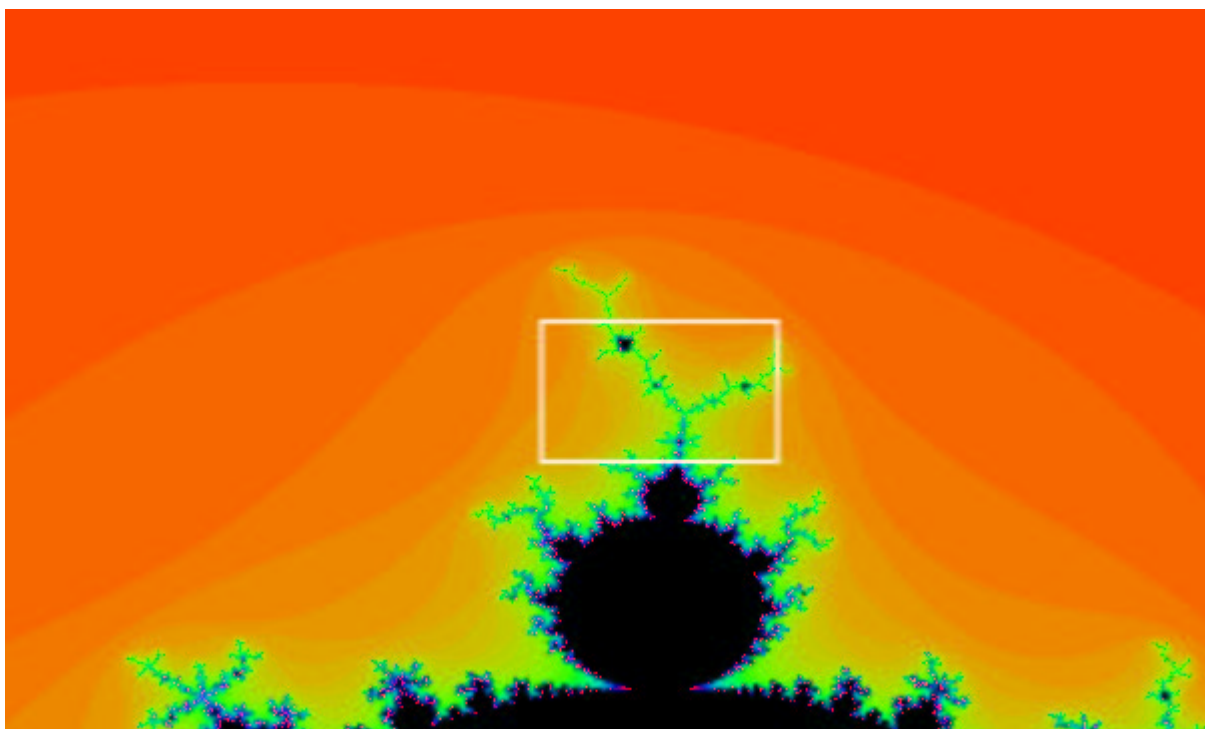
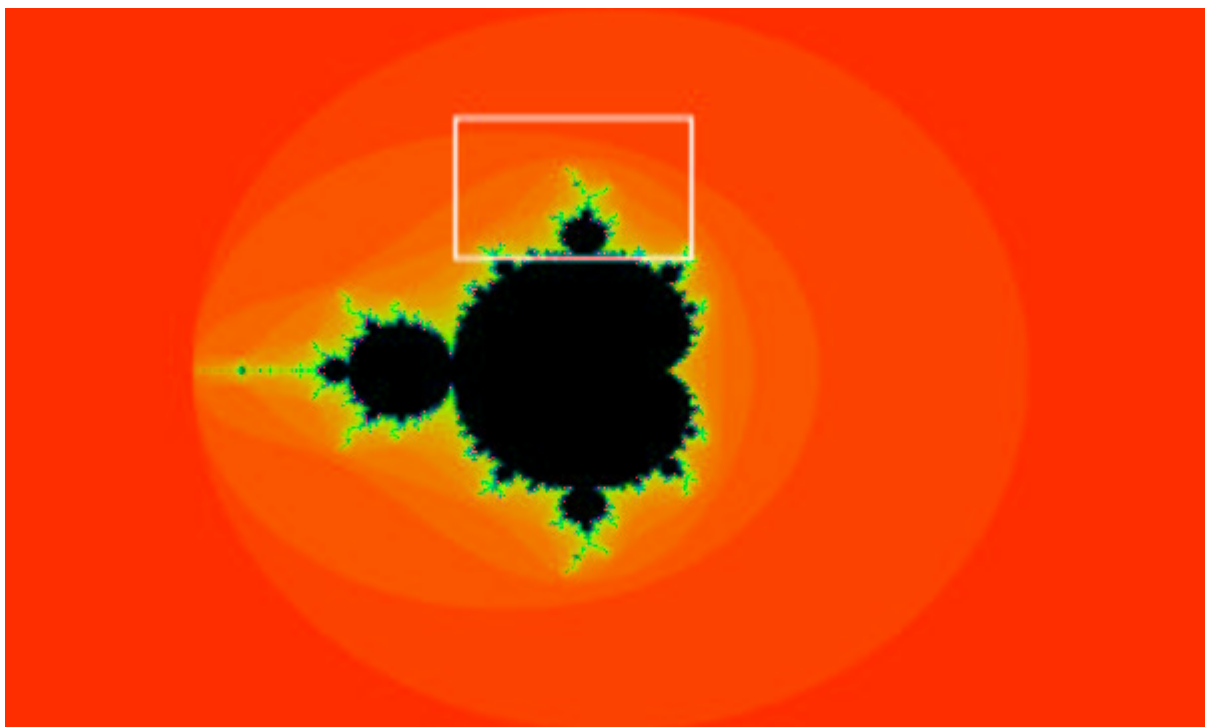


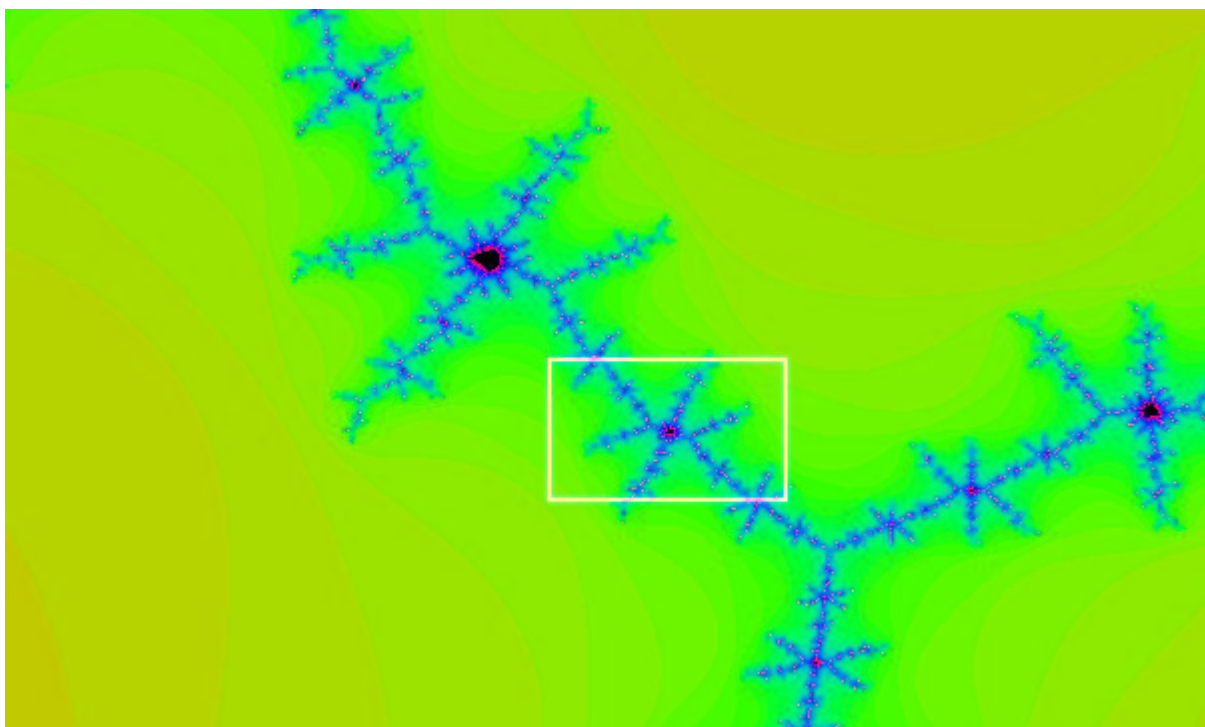
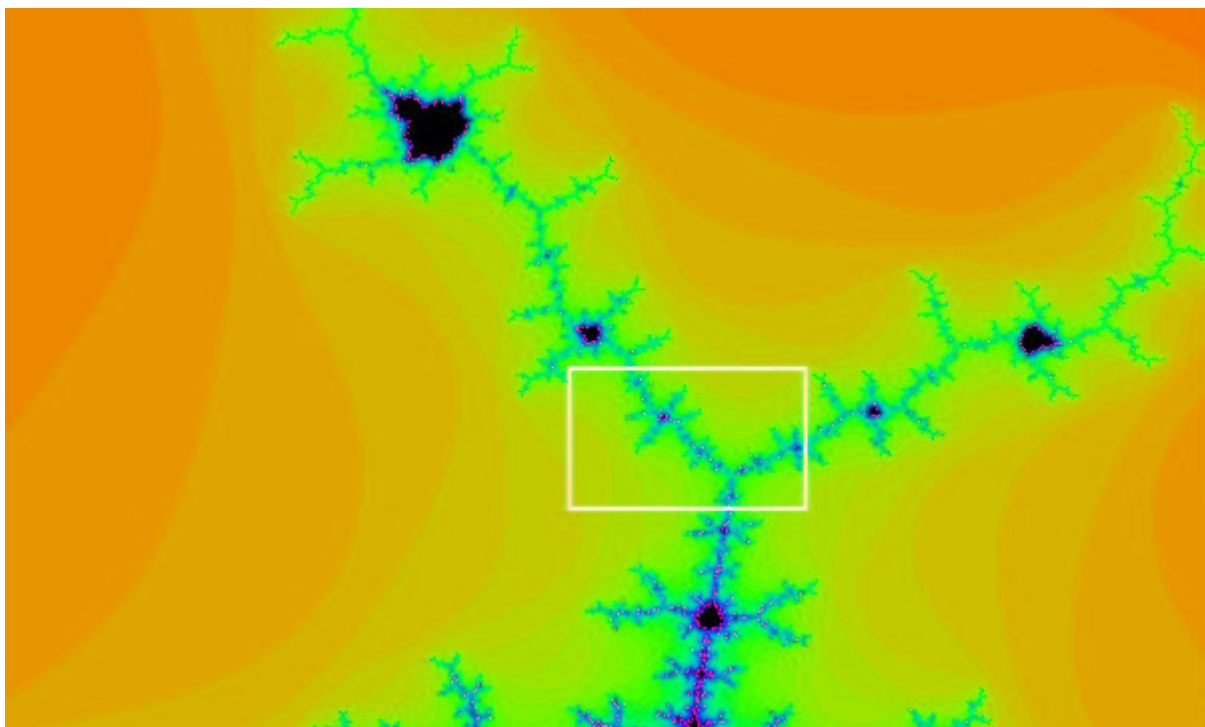
Denne figuren illustrerer hvordan fargen til punktet blir valgt. Langs y-aksen i koordinatsystemet står antall fullførte iterasjoner før avstanden til origo blir større enn 2. Den nederste skrå streken viser hvordan fargen blir valgt hvis maksimum antall iterasjoner er satt til 64. Den vannrette stiplede linja skjærer y-aksen tilsvarende antall fullførte iterasjoner før avstanden blir større enn 2. (Linja er her tilfeldig valgt for å illustrere prinsippet.) I skjæringspunktet med den nederste skrå streken tegner vi en loddrett strek, og på x-aksen leser vi av hvilken farge som skal brukes. Men hvis vi øker maksimum antall iterasjoner til 192, må vi tegne en loddrett strek fra skjæringspunktet med den øverste skrå streken til x-aksen, og da får vi en helt annen farge.

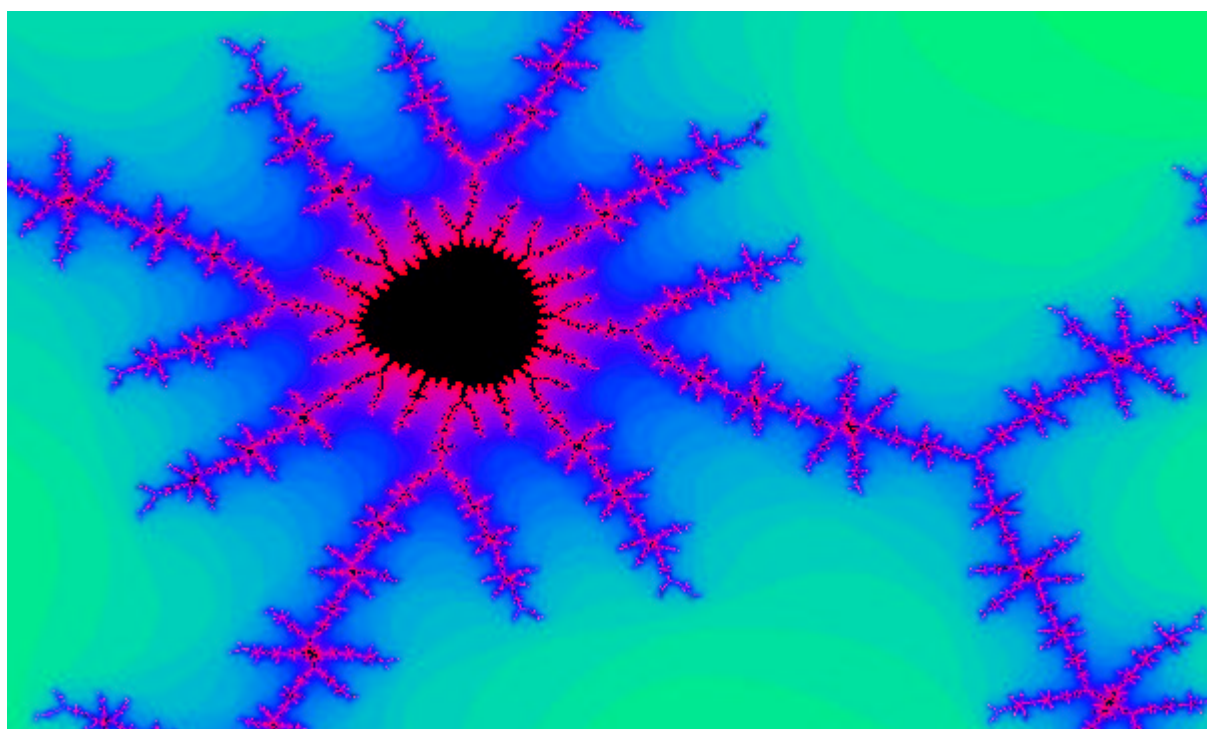
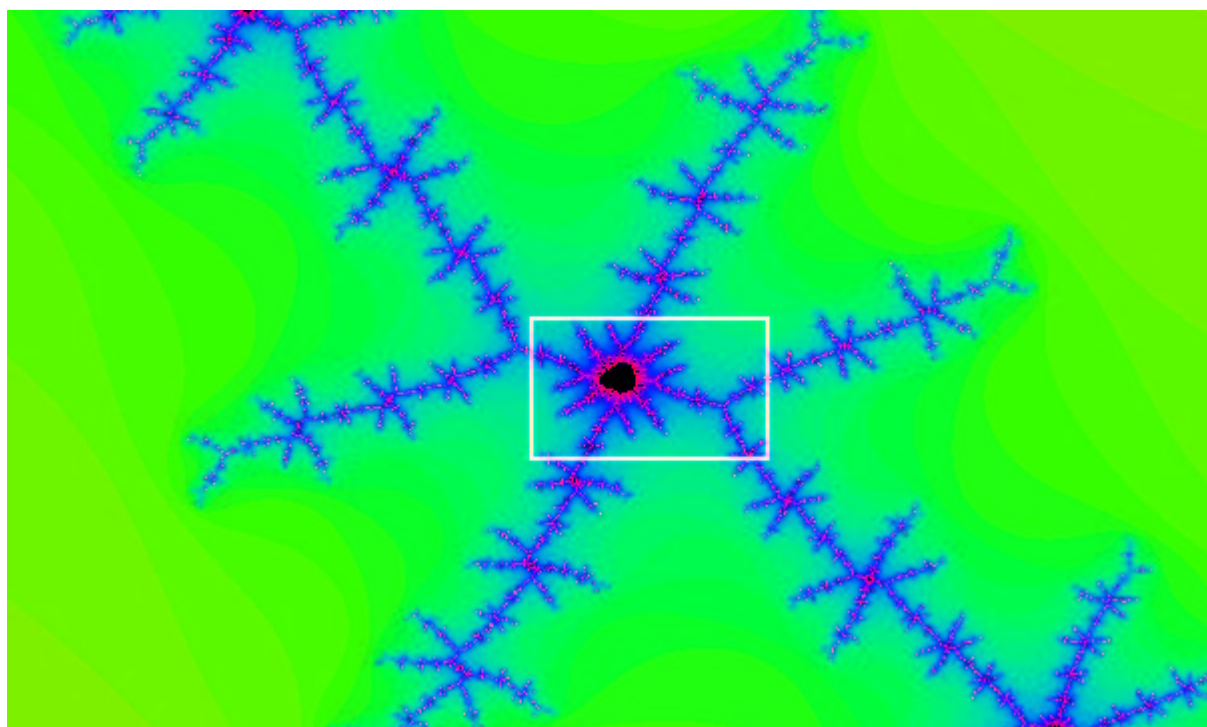
En visualisering av beregningene bak et Mandelbrot-sett ser du på neste figur. Sirkelen har en radius på 2, og de 9 første iterasjonene er vist som streker. Streken starter i punktet $0 + 0i$. Tallet kvadreres, og en konstant C (i dette tilfellet er $C = -0,309 + 0,711i$) legges til. Vi får et nytt komplekst tall, som kan ses på som et punkt i koordinatsystemet. Punktene forbindes med en strek. Det nye tallet kvadreres, den samme konstanten legges til, og vi får et nytt punkt å tegne en strek til. Etter 10 iterasjoner havner vi utenfor sirkelen, og da vil tallet (før eller senere) gå mot uendelig.



Vi kan (som tidligere nevnt) forstørre en fraktal så mye vi vil, og Mandelbrot er ikke noe unntak. På de neste sidene kommer et eksempel.







Julia-settet

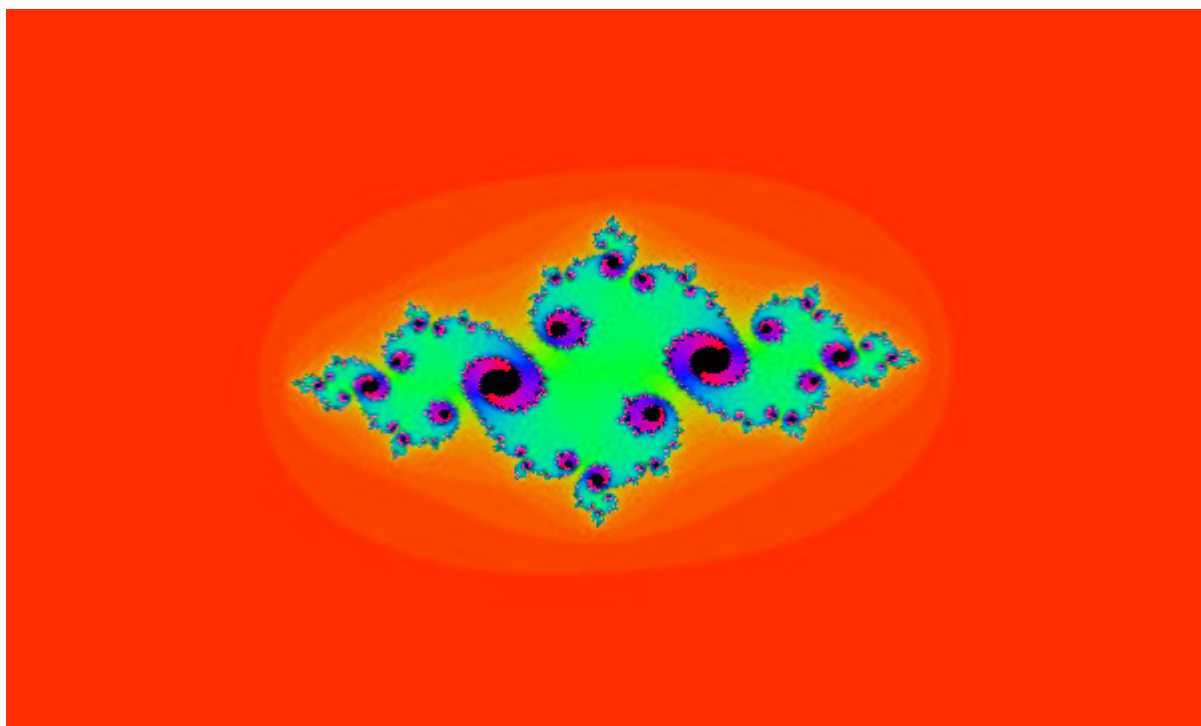
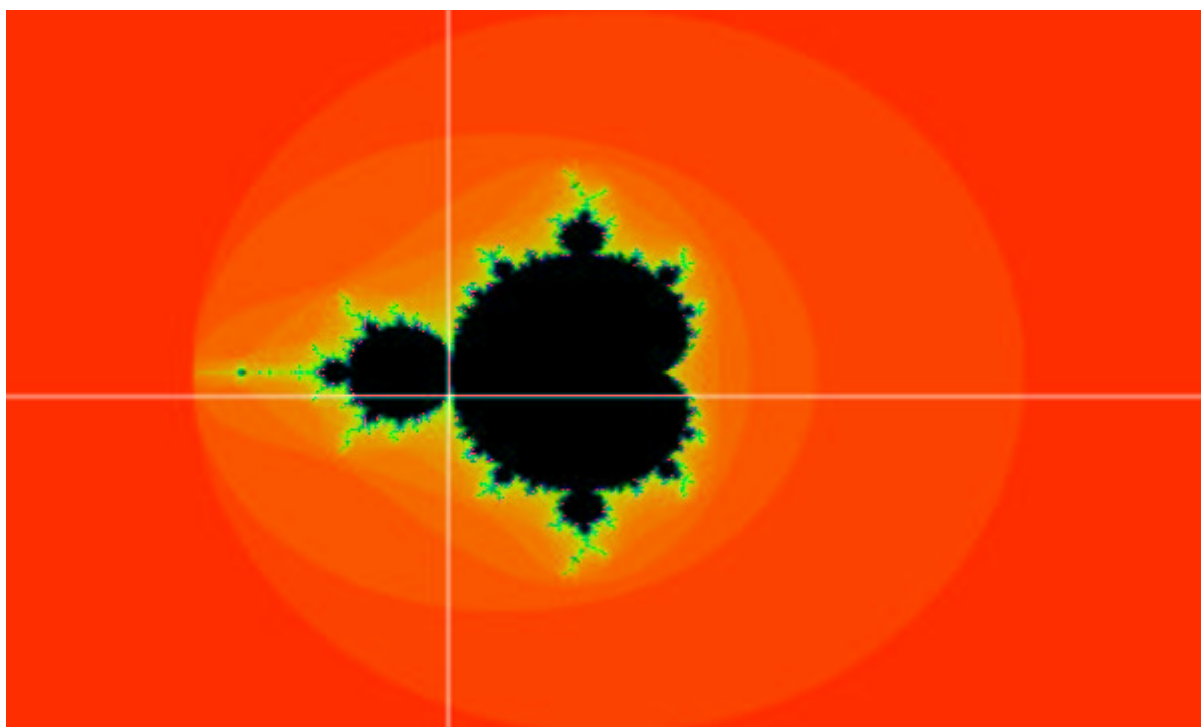
Teorien bak Julia-settet er stort sett den samme som den som ligger bak Mandelbrot-settet.

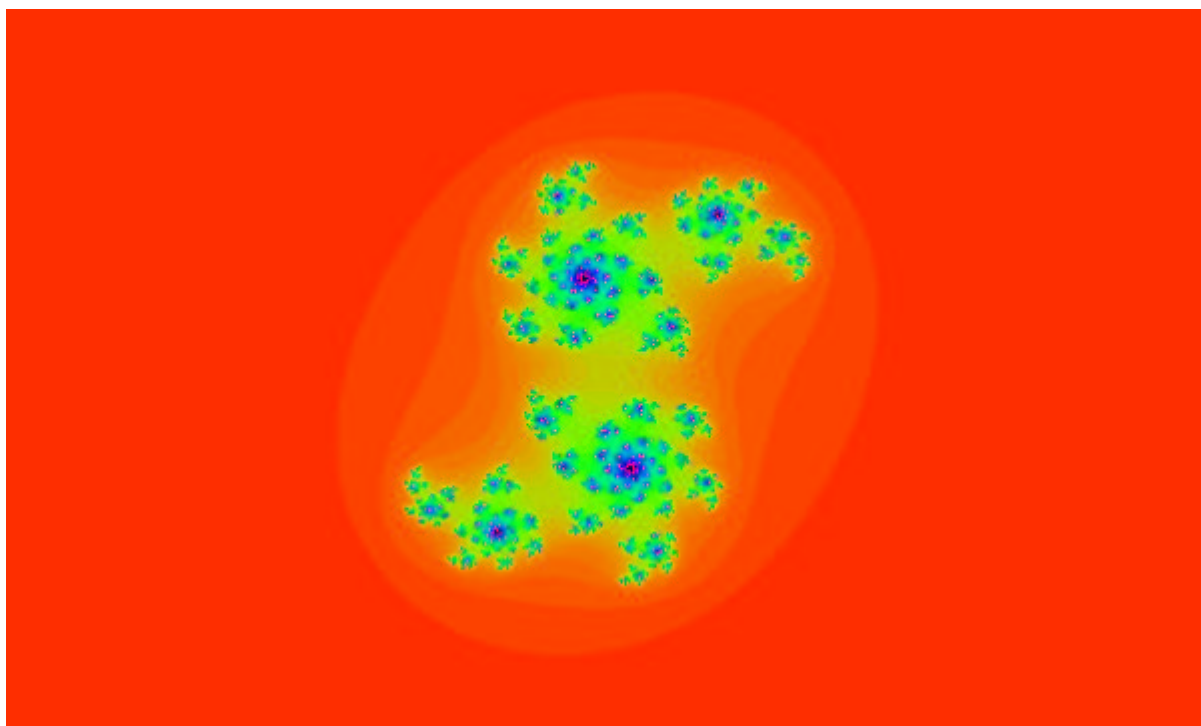
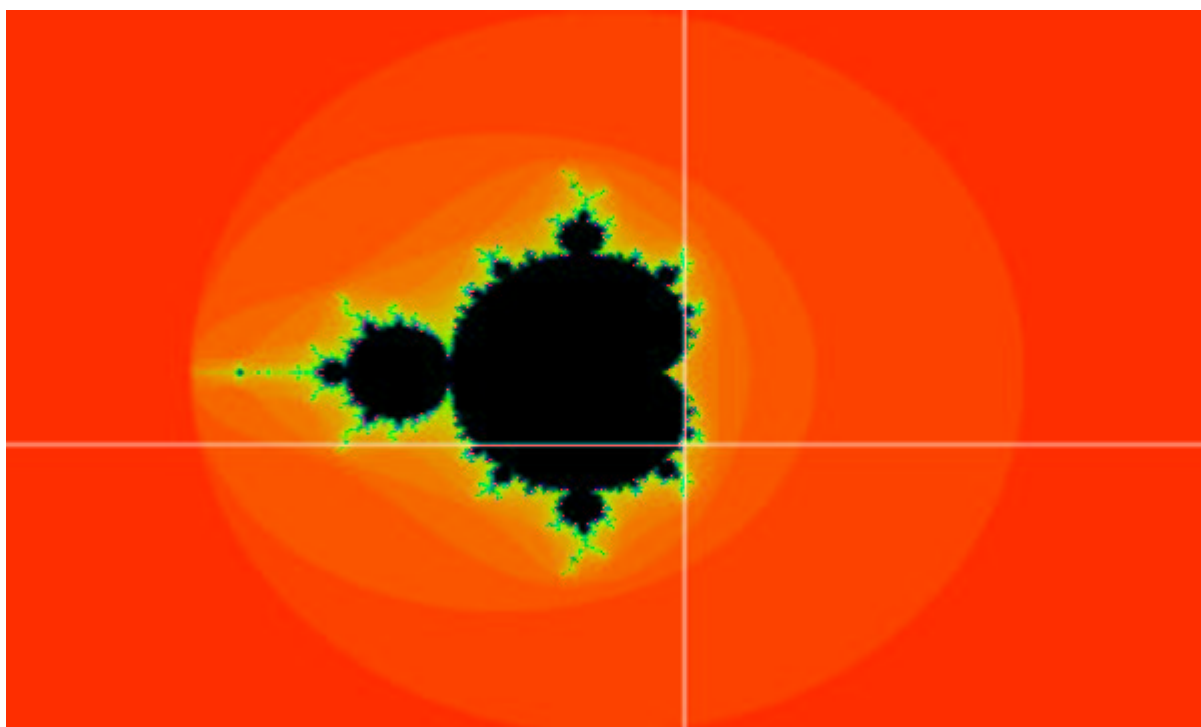
Vi får Julia-settet hvis vi lar $Z_{(0)}$ være et punkt i det komplekse planet, og lar C være konstant. Hvis vi lager et Julia-sett og lar C være et punkt i Mandelbrot-settet med fargeforandringer, får vi et spennende Julia-sett. Hvis vi lar C være 0, får vi bare en fylt svart ring med rødt utenfor.

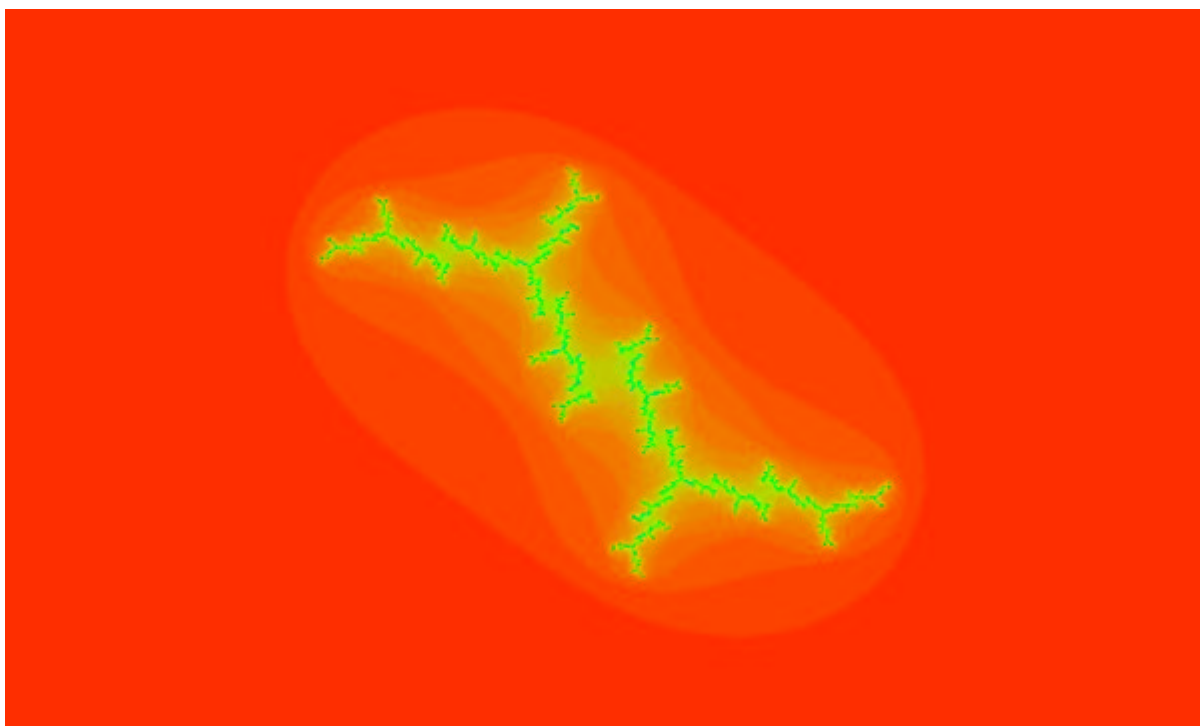
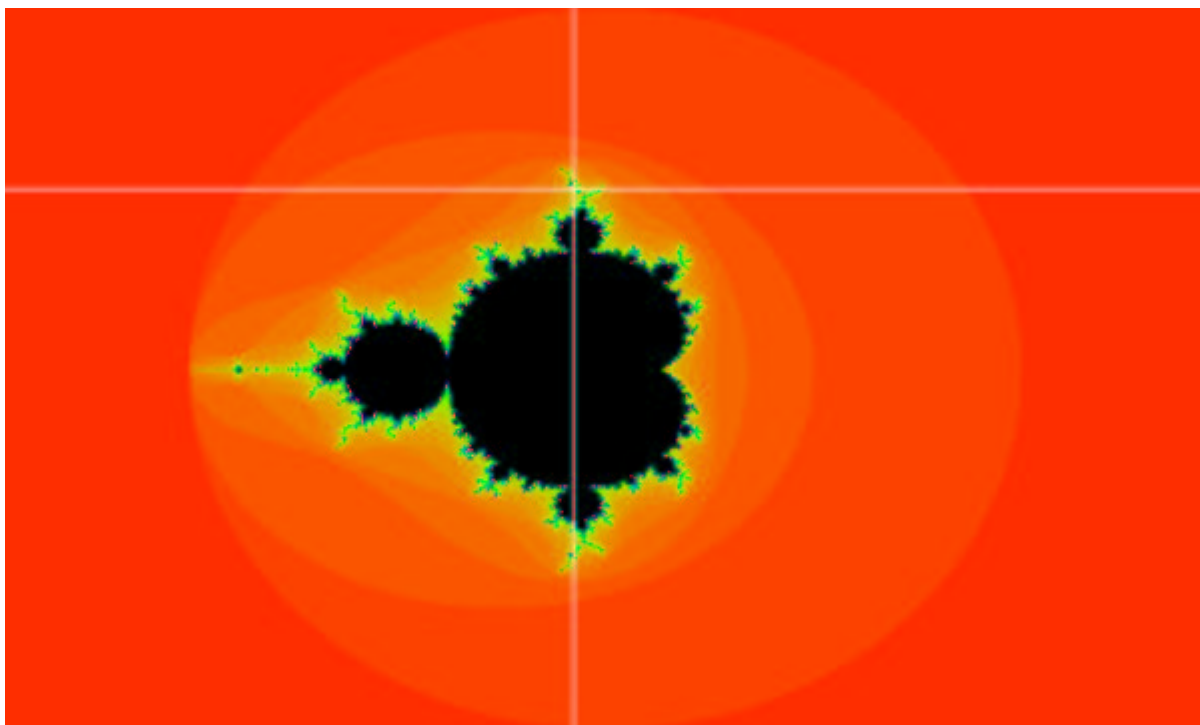
Når vi lager Mandelbrot-settet, er det mulig å eksperimentere med andre verdier enn 0 for $Z_{(0)}$.

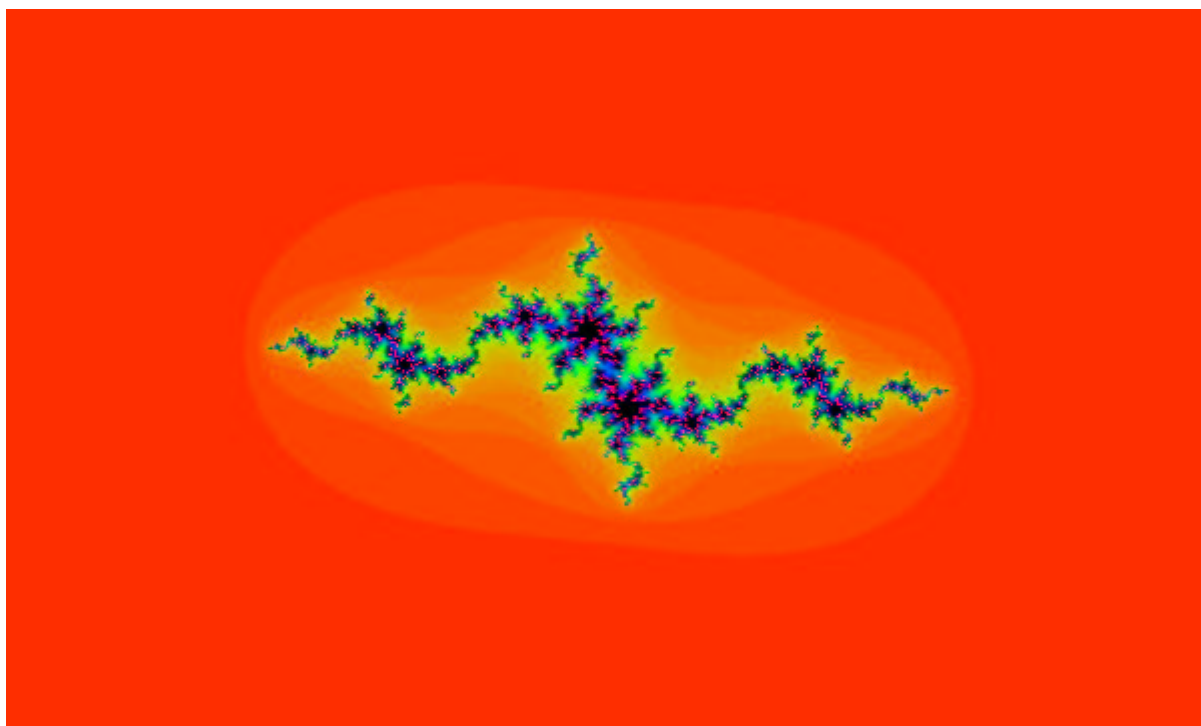
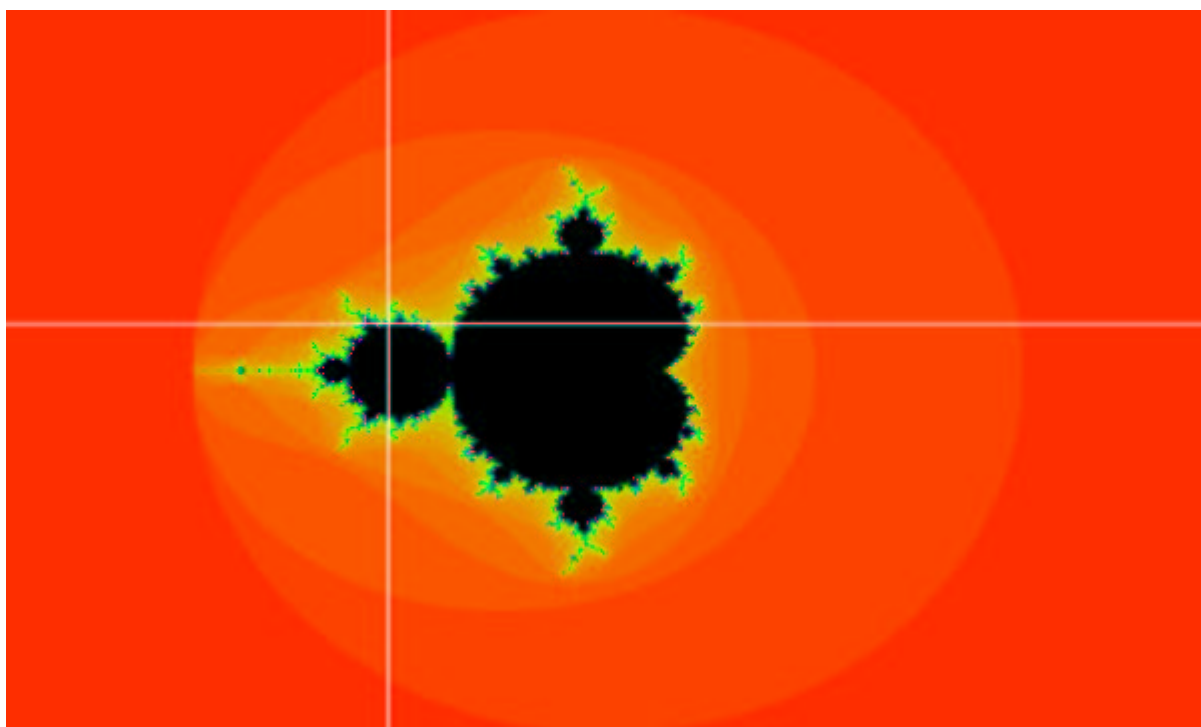
På de neste sidene kommer en serie bilder av Mandelbrot-settet, der et punkt er avmerket. Under hvert bilde er det et Julia-sett med konstanten C satt til a - og b -koordinaten til det avmerkede punktet i Mandelbrot-settet.

Kilde [10]









Del II

Biografier

Georg Ferdinand Ludwig Philipp Cantor



Født: 5. mars 1845 i St. Petersburg, Russland

Død: 6. januar 1918 i Halle, Tyskland

Georg Cantors far, Georg Waldemar Cantor, var kjøpmann. Han var født i Danmark og han var en mann med dyp kjærlighet til kultur og kunst. Georgs mor, Maria Anna Böhm, var russisk og veldig musikalsk. Det er sikkert at Georg arvet store musikalske og kunstneriske talenter fra foreldrene sine, siden han var en enestående fiolinist. Georg var oppdratt som protestant, som var farens religion, mens moren var romersk-katolsk.

Etter tidlig undervisning hjemme fra en privatlærer, begynte Cantor på skolen i St. Petersburg. Så, i 1856, flyttet familien til Tyskland. Cantor var da elleve år gammel.

Siden Cantors far hadde dårlig helse, flyttet de til Tyskland for å finne et varmere klima enn de strenge vintrene i St. Petersburg. Først bodde de i Wiesbaden, hvor han begynte på gymnaset, så flyttet de til Frankfurt. Cantor studerte ved Realschule in Darmstadt hvor han bodde som kostelev. Han tok eksamen i 1860 med en enestående karakterbok, som spesielt nevnte hans uvanlig gode matematikkferdigheter, særlig trigonometri. Etter å ha gått ved Höheren Gewerbeschule in Darmstadt fra 1860, begynte han på Polyteknisk skole i Zürich i 1862.

Studiene i Zürich ble avbrutt av farens død i 1863. Cantor flyttet til Universitetet i Berlin, der han ble venner med Herman Schwarz, en medstudent. Cantor gikk på forelesninger av Weierstrass, Kummer og Kronecker. Han tilbrakte sommersemestret i 1866 ved Universitetet i Göttingen, og vendte så hjem til Berlin for å fullføre avhandlingen sin om tallteori, *De aequationibus secundi gradus indeterminantis*, i 1867.

Mens Cantor var med i Matematisk Forening i Berlin, som president i 1864–1865, var han også med i en liten gruppe av unge matematikere som hadde ukentlige møter. Etter at han mottok doktoratet sitt i 1867, underviste Cantor på en pikeskole i Berlin. Så, i 1868, sluttet han seg til Schellbach Seminar for matematikklærere. I løpet av denne tida arbeidet han med sin dyktighets-attest, og, umiddelbart etter at han ble

ansatt ved Halle i 1869, presenterte han avhandlingen sin, også denne gangen om tallteori, og mottok sin attest.

Ved Halle vendte Cantors arbeid seg bort fra tallteori og mot analyse. Dette fordi Heine, en av hans seniorkolleger ved Halle, utfordret Cantor til å bevise det åpne problemet om det enestående ved framstillingen av en funksjon som ei trigonometrisk rekke. Dette var et vanskelig problem som mange matematikere nytteløst hadde angrepet, deriblant Heine selv, i tillegg til Dirichlet, Lipschitz og Riemann. Cantor løste problemet ved å bevise framstillingen i april 1870. Han gav ut ytterligere vitenskapelige artikler angående trigonometriske rekker, og alle viste innflytelsen av Weierstrass' undervisning.

Cantor ble utnevnt til Ekstraordinær Professor ved Halle i 1872, og i det året startet han sitt vennskap med Dedekind, som han hadde møtt på en ferie i Sveits. Cantor gav ut en avhandling om trigonometriske rekker i 1872, der han definerte irrasjonale tall uttrykt ved konvergente rekker av rasjonale tall.

I 1873 beviste Cantor at de rasjonale tallene er tellelige, dvs. at de kan settes i en-til-en-korrespondanse med de naturlige tallene. Han viste også at de algebraiske tallene, dvs. tall som røttene av polynomlikninger med heltallige koeffisienter, er tellelige. Det viste seg imidlertid at hans forsøk på å bestemme om de reelle var tellelige, ble vanskeligere. Han beviste i desember 1873 at de ikke var tellelige.

I 1874 giftet Cantor seg med Vally Guttmann. De drog på bryllupsreise til Interlaken i Sveits, der Cantor tilbrakte mye tid i matematiske diskusjoner med Dedekind. Cantor opprettholdt gjennom årene sin brevveksling med Dedekind, der de delte ideer og utvekslet meninger fram til 1882.

Cantor publiserte mellom 1879 og 1884 seks avhandlinger i *Mathematische Annalen*, som tok for seg den grunnleggende introduksjonen til sett-teorien. Det viste seg at utgivelsene hans ikke ble møtt med den imøtekommenhet han håpet. Etter bruddet med Dedekind fikk Cantor kontakt med Mittag-Leffler, som publiserte de samme seks avhandlingene. Heller ikke disse fikk særlig aksept i miljøet.

Etter dette fikk Cantor sitt første depressive anfall. En teori var en stund at Cantors depressive anfall ble forårsaket av hans bekymringer i matematikken. Vi vet i dag at Cantors matematiske bekymringer ble forstørret av depresjonen og ikke forårsaket dem.

I 1886 kjøpte Cantor et fint nytt hus i Händelstrasse, en gate oppkalt etter den tyske komponisten Händel. Han fikk seks barn.

Cantor publiserte i 1894 en ganske merkelig avhandling, som listet opp hvordan alle partall opp til 1 000 kunne skrives som summen av to primtall. Siden en bekreftelse av Goldbachs formodning opp til 10 000 hadde vært gjennomført 40 år tidligere, er det sannsynlig at avhandlingen sier mer om Cantors sinnstilstand enn den gjør om Goldbachs formodning. (Goldbachs formodning, som er en av matematikkens store uløste gåter, sier at hvert partall større enn eller lik 6 kan skrives som summen av 2 odde primtall og at hvert oddetall større enn eller lik 9 kan skrives som summen av 3 odde primtall.)

I oktober 1899 søkte han om å forlate undervisningen for vintersemesteret 1899–1900. Søknaden ble innvilget. Den 16. desember 1899 døde Cantors yngste sønn. Fra da av, og ut livet, kjempet han mot depresjoner. Han fortsatte å undervise, men stod over noen vintersemestre, nemlig 1902–1903, 1904–1905 og 1907–1908. Cantor oppholdt seg på flere sanatorier i de største depresjonsperiodene, fra 1899 og utover. Han fortsatte å arbeide, og gav ikke opp matematikken fullstendig. Han foreleste om paradoksene med sett-teori for en sammenkomst av *Deutsche Mathematiker-Vereinigung* i September 1903, og han ble med i Internasjonal Matematikerkongress i Heidelberg i august 1904.

Cantor trakk seg tilbake i 1913. De siste årene var han syk og hadde lite mat på grunn av krigstilstandene i Tyskland. En større markering av Cantors 70-årsdag ble avlyst på grunn av krigen, men en mindre markering ble holdt hjemme hos ham. I juni 1917 besøkte han sanatoriet for siste gang, og han skrev stadig til kona og spurte om lov til å komme hjem. Han døde av et hjerteattakk.

Kilde [5]

Niels Fabian Helge von Koch



Født: 25. januar 1870 i Stockholm, Sverige

Død: 11. mars 1924 i Danderyd, Stockholm, Sverige

Von Koch begynte på en god skole i Stockholm, og gjorde seg ferdig med studiene sine der i 1887. Så begynte han på Stockholms Universitet. Stockholms Universitet var det tredje universitetet i Sverige og ble planlagt allerede fra 1865. Da det åpnet i 1880 (kilde [22] sier 1878) ble Mittag-Leffler den første professoren i matematikk. Legg merke til at selv om vi her betegner det som dets nåværende navn, Stockholms Universitet, gikk det under navnet Stockholms Höghskola fram til 1950 (kilde [22] sier 1960).

Von Koch gikk en tid på Uppsalas Universitet fra 1888. Han var elev hos Mittag-Leffler ved Stockholms Universitet. Von Kochs første arbeid dreide seg om uendelig mange lineære likninger med uendelige mange ukjente.

Mellom 1893 og 1905 hadde von Koch flere utnevnelser som assisterende professor i matematikk. Von Koch ble utnevnt som leder av fakultetet for ren matematikk ved Kungliga Teknologiska Högskolan i Stockholm (KTH) i 1905, etter at Bendixson i 1905, en student under Mittag-Leffler, sa opp sitt professorat på KTH. I 1911 overtok von Koch Mittag-Lefflers stilling som matematikkprofessor på Stockholms Universitet.

Von Koch er berømt for sin kurve som framgår av hans publikasjon i 1906 om *Une méthode géométrique élémentaire pour l'étude de certaines questions de la théorie des courbes plane*. Kurven konstrueres ved å dele et linjestykke i tre like deler og erstatte den midtre delen med en likesidet trekant uten grunnlinje. Gjenta utskiftningen på alle (nå 4) segmentene. Gjenta prosessen i det uendelige. Dette gir en kontinuerlig kurve med uendelig lengde som ikke er deriverbar i noe punkt. Hvis man starter med en likesidet trekant og anvender denne konstruksjonen, får man til slutt von Kochs snøkrystall. Du kan lese mer om dette på side 10.

Snøkrystallen er en kontinuerlig kurve som ikke har en tangent i noe punkt. Hans publikasjon fra 1906 består i hovedsak av bevis for dette faktum. Han viser også der at det er to funksjoner f og g som begge ikke er deriverbare i noe punkt slik at snøkrystallens kurve blir

$$x = f(t), y = g(t) \text{ der } -1 \leq t \leq 1.$$

Kilde [7]

Waclaw Sierpinski



Født: 14. mars 1882 i Warszawa, Polen

Død: 21. oktober 1969 i Warszawa, Polen

Faren var lege, og Sierpinski gikk på en skole i Warszawa der hans talent for matematikk raskt ble oppdaget av hans første matematikklærer. Det var på den tida vanskelig å få utdannelse i Polen på grunn av Russlands okkupasjon. Russerne ønsket å holde analfabetismen i Polen så høy som mulig, og hadde påtvunget Polen det russiske språk og kultur i all skoleundervisning mellom 1869 og 1874.

Til tross for disse vanskelighetene begynte Sierpinski i 1899 på Universitetet i Warszawa (tsarens universitet) på fakultetet for matematikk og fysikk. De ansatte var russere, forelesningene var på russisk og Sierpinskis første lærer i matematikk, russeren Voronoy, fikk således stor betydning for ham. I 1903 fikk Sierpinski en gullmedalje for den beste avhandlingen om Voronoys bidrag til tallteorien.

Femti år etter at Sierpinski hadde sluttet ved universitetet, fortalte han litt om problemene som student

... vi måtte ta en årlig leksjon i det russiske språket. ... Hver student satte sin ære i å få det dårligste resultatet. ... Jeg svarte ikke på et eneste spørsmål. ... Jeg fikk dårlige karakterer. ... Jeg bestod alle prøver. ... En lektor foreslo at jeg skulle ta eksamen om igjen, for ellers ville jeg ikke få god nok karakter til å studere videre. Jeg nektet, og sa at det ville være første gang at en som hadde toppkarakterer i alle fag, ikke ville komme inn på matematikkvitenskapen på grunn av dårlig karakter i russisk.

Sierpinski kom likevel inn da læreren hans endret språkkarakteren til «god».

Han ble ferdig utdannet i 1904 og arbeidet en stund som skolelærer i matematikk og fysikk på en pikeskole i Warszawa. Da skolen stengte, bestemte han seg for å dra til Krakow for å ta doktorgraden. Der gikk han på forelesningene til Zaremba, og studerte i tillegg astronomi og filosofi. Etter doktorgraden begynte han på Universitetet i Lvov i 1908.

I 1907 ble han interessert i sett-teorien da han kom over et teorem som sa at punkter i planet kunne angis med en enkel koordinat. Han skrev til Banachiewicz i Göttingen og spurte hvordan et sånt resultat var mulig. Han fikk bare ett ord til svar: «Cantor». Sierpinski begynte å studere sett-teori i 1909, og var den første som holdt forelesninger bare om dette.

Gjennom livet opprettholdt Sierpinski en enorm produksjonskapasitet med hensyn til å gi ut forskningsresultater og bøker.

Da første verdenskrig begynte i 1914, var Sierpinski og familien i Russland. På den tida prøvde regjeringene i Østerrike og Russland å gjøre bruk av det polske problemet som et politisk våpen. Sierpinski ble internert i Viatkan, men fikk, ved hjelp av Egorov og Lutzin, anledning til å bli overført til Moskva. Resten av krigsårene tilbrakte han der, hvor han i samarbeid med Lutzin begynte studiet av analytiske sett. I løpet av denne perioden, utga han det første eksemplet på et absolutt normalt tall, som er et tall hvis sifre forekommer med lik frekvens, uansett hvilket tallsystem det er skrevet i.

Etter første verdenskrig dro Sierpinski tilbake til Lvov. Like etter ble han tilbudt en stilling på Universitetet i Warszawa som han takket ja til. I 1919 ble han utnevnt til professor og tilbrakte resten av livet der.

I 1920 startet han, sammen med en tidligere student av seg, matematikkmagasinet *Fundamenta Mathematica*. Sierpinski redigerte bladet, som spesialiserte seg på sett-teori.

I denne perioden arbeidet Sierpinski mest på området med sett-teorier. Han studerte Sierpinski-kurven som beskriver en lukket vei som inneholder ethvert indre punkt i en firkant. Kurvens lengde er uendelig, mens området avgrenset av den, er $5/12$ av kvadratet.

Sierpinski var også sterkt involvert i utviklingen av faget matematikk i Polen. Han ble beæret med å bli valgt inn i det polske akademi i 1921, og ble dekanus på universitetets fakultet samme år. I 1928 ble han nestleder for Warszawas Vitenskapelige Selskap, og samme år ble han valgt til leder av Polens Matematiske Selskap.

I 1939 endret livet i Warszawa seg dramatisk på grunn av andre verdenskrig. Sierpinski fortsatte å arbeide «underground» samtidig som han opprettholdt sin offisielle stilling på kontoret. Utgivelsene hans fortsatte ved at han klarte å sende papirene til Italia. Hver av utgivelsene ble avsluttet med følgende tekst:

Bevisene for disse teoremene vil komme i Fundamenta Mathematica

som alle forstod betydde

Polen vil overleve

Etter opprøret i 1944 brente nazistene huset hans, og ødela biblioteket med bøkene og alle personlige brev. Flere av hans studenter som døde under krigen ble nevnt i Sierpinskis senere forelesninger. Mer enn halvparten av Polens matematikere ble drept.

Sierpinski gav ut 724 avhandlinger og 50 bøker. Han trakk seg tilbake i 1960, men fortsatte å holde seminarer fram til 1967. Han mottok så mange æresbevisninger at det er umulig å nevne alle.

Rotkiewicz, en student av Sierpinski, beskrev ham slik:

... hadde en sjeldent god helse og positiv væremåte. ... Kunne arbeide under alle forhold. ... Han likte ikke at noen korrigerende avhandlingene hans, da tilføyde han bare følgende setning: «Mr X har bemerket at...". Han var kreativ og likte kreativ matematikk. Han var den største og mest produktive polske matematiker.

Kilde [8]

Gaston Maurice Julia



Født: 3. februar 1893 i Sidi Bel Abbès, Algerie

Død: 19. mars 1978 i Paris, Frankrike

Han var bare 25 år gammel da han gav ut sitt 199 siders mesterstykke, *Mémoire sur l'iteration des fonctions rationnelles*, noe som gjorde ham berømt i de matematiske sentre på den tid.

Som soldat i første verdenskrig, ble Julia alvorlig såret i et angrep på den franske front. Angrepet ble iverksatt for å feire keiserens fødselsdag. Mange på begge sider ble skadet, blant andre Julia, som mistet nesa og måtte gå med en lærstropp over ansiktet resten av livet. Innimellom flere smertefulle operasjoner fortsatte han sine matematiske undersøkelser på sykehuset.

Han ble senere en respektert professor på École Polytechnique i Paris. Han utgav i 1918 et fantastisk stykke arbeid, *Mémoire sur l'iteration des fonctions rationnelles*, *Journal de Math. Pure et Appl.*, som behandlet iterasjon av rasjonale funksjoner. Han mottok Grand Prix fra Vitenskapsakademiet.

I 1925 ble det organisert seminarer i Berlin for å studere arbeidene hans, og blant deltakerne var Brauer, Hopf og Reidemeister. H. Cremer skrev en artikkel om arbeidene til Julia, der han tok for seg visualiseringen av et Julia-sett.

Til tross for at han var berømt på 1920-tallet, ble arbeidene hans i praksis glemt, helt til B. Mandelbrot bragte dem fram i lyset i 1970-årene ved hjelp av sine grunnleggende datamaskin-eksperimenter.

Kilde [6]

Benoit Mandelbrot



Født: 20. november 1924 i Warszawa, Polen

Benoit Mandelbrot har mye av æren for den nåværende interessen for fraktalgeometri. Han viste hvorledes fraktaler kan oppstå på mange forskjellige steder, både i matematikken og rundt omkring i naturen.

Han ble født i Polen i 1924 i en familie med akademiske tradisjoner. Hans far livnærte seg imidlertid av å kjøpe og selge klær, mens moren var lege. Som ung gutt ble Mandelbrot kjent med matematikken gjennom sine to onkler. Familien emigrerte til Frankrike i 1936. Onkelen, som het Szolem Mandelbrojt, var professor i matematikk ved Collège de France. Han tok ansvaret for guttens utdannelse. Onkelens innflytelse var både positiv og negativ, da han var en stor beundrer av Hardys matematiske filosofi. Dette gjorde at Mandelbrot fikk en motreaksjon til ren matematikk. Mandelbrot forstod først senere hvordan Hardys dypfølte pasifisme hadde påvirket ham selv til å frykte at den anvendte matematikken i gale hender kunne bli brukt i ond hensikt i krigstider.

Mandelbrot gikk på Lycée Rolin i Paris da andre verdenskrig brøt ut, og familien flyttet til Tulle i Midt-Frankrike. Dette var en vanskelig tid for ham, han fryktet flere ganger for sitt liv, og av den grunn måtte han gå ukonvensjonelle veier for å fortsette studiene. Dette oppmuntret ham til kritisk og selvstendig tenkning, og var medvirkende til at han utviklet sin geometriske tilnærming til matematikken. Hans fantastiske geometriske intuisjon og visjoner gav ham på sin side en unik innsikt i matematiske problemer.

Etter å ha studert i Lyon, var han student ved École Normale i Paris bare en dag. Han begynte i 1944 på École Polytechnique under Paul Lévy. Han drog deretter til USA til California Institute of Technology, og fortsatte ved Institute for Advanced Study i Princeton.

Han vendte tilbake til Frankrike i 1955, der han arbeidet ved Centre National de la Recherche Scientifique. Han giftet seg så med Ailette Kagan, og de drog sammen tilbake til USA, der han begynte sine langvarige studier og et fruktbart samarbeid med IBM som professor ved deres berømte laboratorier i Yorktown Heights i staten New York. IBM gav Mandelbrot anledning til å utforske sitt store idé-repertoar, og gav ham muligheten til å forske videre, en mulighet som ingen universitetsstilling kunne tilbuds.

I 1945 hadde Mandelbrots onkel vist ham Julias mesterstykke fra 1918, men det likte ikke Mandelbrot. Han opplevde at onkelens forhold til matematikk var helt motsatt av hans eget. Isteden valgte Mandelbrot sin egen kurs videre framover, noe som igjen brakte ham tilbake til Julias arbeider i 1970 etter mange ulike vitenskaper, en kurs som kan karakteriseres som meget individualistisk eller nomadisk. Mandelbrots beslutning om å bidra på så mange ulike vitenskapelige områder, ble fattet meget bevisst da han var yngre, og det er bemerkelsesverdig hvordan han klarte å oppfylle denne ambisjonen med slike fantastiske resultater på så mange områder.

Ved hjelp av datagrafikk, arbeidet Mandelbrot ved IBMs Watson Research Center, og klarte å vise hvordan Julias arbeid var en kilde til noen av de mest fantastiske fraktalene vi kjenner i dag. For å klare dette, måtte han utvikle ikke bare nye matematiske ideer, men også noen av de første dataprogrammene for å skrive ut grafikk.

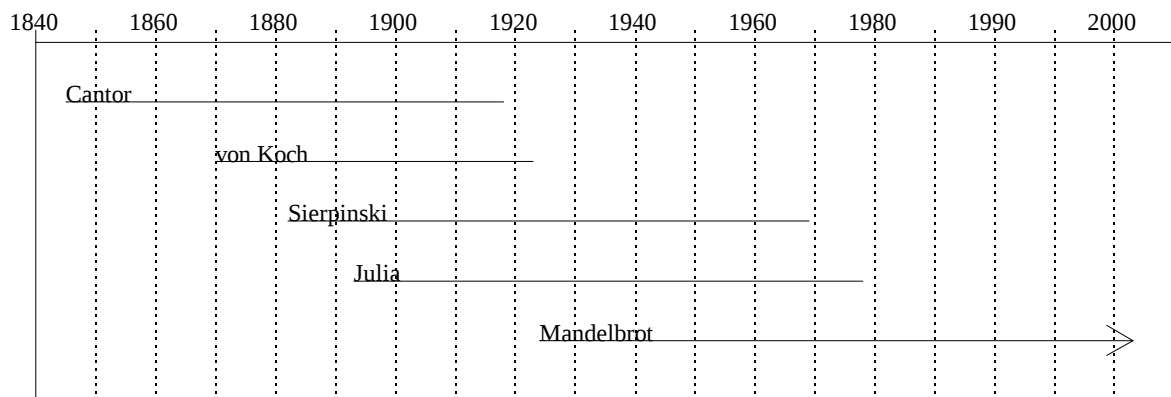
Mandelbrot-settet består av punkter i det komplekse planet. Les mer om dette på side 37. Han gav ut en bok om dette i 1975, som heter *Les objets fractals, form, hasard et dimension*. En mer komplett versjon ble utgitt i 1982, nemlig *The fractal geometry of nature*.

Mandelbrot var professor i praktisk matematikk ved Harvard University. Han var også ingeniørprofessor ved Yale, professor i matematikk ved École Polytechnique, professor i økonomi ved Harvard og professor i fysiologi ved Einstein College of Medicine.

Han har mottatt mengder av æresbevisninger og priser som anerkjennelse for sin bemerkelsesverdige innsats. I 1980 fikk han Barnard Medal for Meritorious Service to Science. I 1986 fikk han Franklin-medaljen. I 1987 ble han æret med Alexander von Humboldt-prisen, og han fikk Steinmetz-medaljen i 1988. I tillegg til mange andre medaljer, mottok han også Nevada-medaljen i 1991 og Wolf-prisen for fysikk i 1993.

Kilde [9]

Tidslinje



Del III

Tillegg

Dataprogrammer

For å gjøre det lett å forstå hvordan programmene fungerer, har jeg valgt programmeringsspråket Basic for programmene jeg viser her. Jeg har også laget noen Delphi-programmer.

Program 1 – Kaos.bas

```
'KAOS.BAS
'-----

'Programmet bruker kaos-spillet til å tegne Sierpinski-trekanten
'Trykk på en tast for å stoppe beregningene

'Kristoffer Andreas Berg, 5. februar 2003

RANDOMIZE TIMER
SCREEN 9          'velger skjermmodus med 640 x 350 pikslers grafikk
WINDOW (0, 1)-(1, 0) 'innstiller koordinatsystemet

oldx = 0
oldy = 0

n = 0             'setter iterasjonstelleren til null
DO
n = n + 1         'øker iterasjonstelleren med 1
r = INT(RND * 3)  'velger et tilfeldig hjørne
SELECT CASE r
CASE 0            'flytter halvveis mot det venstre hjørnet
  x = oldx / 2
  y = oldy / 2
CASE 1            'flytter halvveis mot topphjørnet
  x = (oldx + .5) / 2
  y = (oldy + 1) / 2
CASE 2            'flytter halvveis mot det høyre hjørnet
  x = (oldx + 1) / 2
  y = oldy / 2
END SELECT

PSET (x, y)       'plotter punktet (x, y)

oldx = x
oldy = y

LOOP WHILE INKEY$ = "" 'gjentar prosessen helt til en tast trykkes ned

PRINT "Antall iterasjoner:"
PRINT n
```

Programmet har jeg laget selv, basert på framgangsmåten beskrevet i kilde [25]

Program 2 – Mandel.bas

```
'MANDEL.BAS
'-----

'Programmet viser Mandelbrot-settet

'Kristoffer Andreas Berg, 5. februar 2003

RANDOMIZE TIMER

SCREEN 9                'velger skjermmodus med 640 x 350 pikslers grafikk

FOR x = 145 TO 494
  FOR y = 0 TO 349

    Ca = (x - 320) / 87.5 'konverterer fra skjermkoordinater
    Cb = (y - 175) / 87.5 'til bildekoordinater
    olda = 0
    oldb = 0
    i = 0                'setter iterasjonstelleren til null

    DO
      a = olda ^ 2 - oldb ^ 2 + Ca 'beregner den nye realdelen
      b = 2 * olda * oldb + Cb     'beregner den nye imaginærdelen
      i = i + 1                    'øker iterasjonstelleren med 1
      l = a ^ 2 + b ^ 2             'beregner kvadratet av lengden til origo
      olda = a
      oldb = b
    LOOP UNTIL l > 4 OR i > 15     'avbryter beregningen for pikselen hvis
                                   'lengden til origo er større enn 2 eller
                                   'hvis antall iterasjoner overstiger 15

    PSET (x, y), i                'gir pikselen (x, y) fargen i

  NEXT y
NEXT x
```

Programmet har jeg laget selv, basert på framgangsmåten beskrevet i kilde **[10]**

Program 3 – Julia.bas

```
'JULIA.BAS
'-----

'Programmet viser et Julia-sett

'Kristoffer Andreas Berg, 5. februar 2003

RANDOMIZE TIMER
SCREEN 9                'velger skjermmodus med 640 x 350 pikslers grafikk

PRINT "Oppgi realdel og"
PRINT "imaginærdel for"
PRINT "den komplekse"
PRINT "konstanten C"
PRINT

PRINT "Realdel?"
INPUT "", Ca

PRINT "Imaginærdel?"
INPUT "", Cb

FOR x = 145 TO 494
  FOR y = 0 TO 349

    olda = (x - 320) / 87.5 'konverterer fra skjermkoordinater
    oldb = (y - 175) / 87.5 'til bildekoordinater
    i = 0                  'setter iterasjonstelleren til null

    DO
      a = olda ^ 2 - oldb ^ 2 + Ca 'beregner den nye realdelen
      b = 2 * olda * oldb + Cb    'beregner den nye imaginærdelen
      i = i + 1                   'øker iterasjonstelleren med 1
      l = a ^ 2 + b ^ 2           'beregner kvadratet av lengden til origo

      olda = a
      oldb = b

    LOOP UNTIL l > 4 OR i > 15    'avbryter beregningen for pikselen hvis
                                'lengden til origo overstiger 2 eller hvis
                                'antall iterasjoner er større enn 15

    PSET (x, y), i               'gir pikselen (x, y) fargen i

  NEXT y
NEXT x
```

Programmet har jeg laget selv, basert på framgangsmåten beskrevet i kilde [10]

Program 4 – Koch.bas

'Programmet tegner von Kochs snøkrystall

DECLARE SUB snow (x!, y!, a!, n!)

pi = ATN(1) * 4

SCREEN 9

WINDOW (-3, -9 / 4 - 1)-(3, 9 / 4 - 1)

CLS

INPUT "Dybde"; d

CALL snow(-1.5, 0, 0, 0)

CALL snow(1.5, 0, -2 * pi / 3, 0)

CALL snow(0, -SQR(27) / 2, 2 * pi / 3, 0)

SUB snow (x, y, a, n)

SHARED pi, d

r = 1 / 3 ^ n

IF n > d - 1 THEN LINE (x, y)-(x + r * 3 * COS(a), y + r * 3 * SIN(a)): EXIT SUB

CALL snow(x, y, a, n + 1)

CALL snow(x + r * COS(a), y + r * SIN(a), a + pi / 3, n + 1)

CALL snow(x + r * COS(a + pi / 6) * SQR(3), y + r * SIN(a + pi / 6) * SQR(3), a - pi / 3, n + 1)

CALL snow(x + 2 * r * COS(a), y + 2 * r * SIN(a), a, n + 1)

END SUB

Dette programmet fant jeg på internett, men jeg kan ikke med sikkerhet gjengi kilden.

Program 5 – Bregne.bas

Du kan selv modifisere konstantene a , b , c , d , e , f og p som brukes i programmet. De lagres i tabellen kalt a . Da kan du lage en fraktal du selv velger.

```
'BREGNE.BAS
'-----

'Programmet tegner Black Spleenwort-bregnen, også kjent som Barnsleys bregne
'Trykk på en tast for å stoppe beregningene

'Kristoffer Andreas Berg, 5. februar 2003

RANDOMIZE TIMER
SCREEN 9
WINDOW (-6.2, 10.5)-(8.8, 0)

DIM a(7, 4)

'Leser inn konstanter

'a
a(1, 1) = 0
a(1, 2) = .85
a(1, 3) = .2
a(1, 4) = -.15
'b
a(2, 1) = 0
a(2, 2) = .04
a(2, 3) = -.26
a(2, 4) = .28
'c
a(3, 1) = 0
a(3, 2) = -.04
a(3, 3) = .23
a(3, 4) = .26
'd
a(4, 1) = .16
a(4, 2) = .85
a(4, 3) = .22
a(4, 4) = .24
'e
a(5, 1) = 0
a(5, 2) = 0
a(5, 3) = 0
a(5, 4) = 0
'f
a(6, 1) = 0
a(6, 2) = 1.6
a(6, 3) = 1.6
a(6, 4) = .44
'p
a(7, 1) = .01
a(7, 2) = .85
a(7, 3) = .07
a(7, 4) = .07

n = 0      'setter iterasjonstelleren til null

oldx = 0
oldy = 0
```

DO

```
n = n + 1  'øker iterasjonstelleren med 1
r = RND    'velger et tilfeldig tall

SELECT CASE r
  CASE 0 TO a(7, 1)
    v = 1
  CASE a(7, 1) TO a(7, 1) + a(7, 2)
    v = 2
  CASE a(7, 1) + a(7, 2) TO a(7, 1) + a(7, 2) + a(7, 3)
    v = 3
  CASE a(7, 1) + a(7, 2) + a(7, 3) TO 1
    v = 4
END SELECT

x = a(1, v) * oldx + a(2, v) * oldy + a(5, v)  'beregner ny x-verdi
y = a(3, v) * oldx + a(4, v) * oldy + a(6, v)  'beregner ny y-verdi

oldx = x
oldy = y
PSET (x, y), 2  'plotter punktet (x, y)

LOOP UNTIL INKEY$ <> ""

PRINT "Antall iterasjoner:"
PRINT n
```

Programmet har jeg laget selv, basert på informasjon i kilde [17]

Program 6 – Pythagor.bas

```
'Pythagor.bas
'-----

'Programmet tegner et pytagoreisk tre

DIM x2(1000), y2(1000), u2(1000), v2(1000), s(1000), double(100)
SCREEN 9
INPUT "vinkel"; angle
INPUT "orden"; order

f = angle * ATN(1) / 45
WINDOW (-3, -2)-(5, 4)

outt = 0
coss = COS(f)
sinn = SIN(f)

a1 = -COS(f) * SIN(f)
a2 = COS(f) ^ 2
b1 = a1 + a2
b2 = -a1 + a2
c1 = b2
c2 = 1 - b1
d1 = 1 - a1
d2 = 1 - a2
x1 = 0
y1 = 0
u1 = 1
v1 = 0
q = 0
j = 1
s(0) = 1

LINE (0, 0)-(1, 0)
LINE (1, 0)-(1, -1)
LINE (1, -1)-(0, -1)
LINE (0, -1)-(0, 0)

WHILE (NOT outt)
    m = q + j
    x = u1 - x1
    y = v1 - y1

    xa = x1 + a1 * x - a2 * y
    ya = y1 + a2 * x + a1 * y
    xb = x1 + b1 * x - b2 * y
    yb = y1 + b2 * x + b1 * y

    x2(m) = x1 + c1 * x - c2 * y
    y2(m) = y1 + c2 * x + c1 * y
    u2(m) = x1 + d1 * x - d2 * y
    v2(m) = y1 + d2 * x + d1 * y

    sinn = x * x + y * y
    IF (sinn < .01) THEN COLOR 2 ELSE COLOR 6
    s(m) = 1

    LINE (x1, y1)-(xa, ya)
    LINE (xa, ya)-(xb, yb)
```

```

LINE (xb, yb)-(u1, v1)
LINE (u1, v1)-(u2(m), v2(m))
LINE (u2(m), v2(m))-(x2(m), y2(m))
LINE (x2(m), y2(m))-(x1, y1)

x1 = xa
y1 = ya
u1 = xb
v1 = yb
IF (m = order OR sinn < eps) THEN
  k = 1
  WHILE (s(m - k) = 0)
    k = k + 1
  WEND
  IF (m = k) THEN
    outt = -1
  ELSE
    q = m - k
    x1 = x2(q)
    y1 = y2(q)
    u1 = u2(q)
    v1 = v2(q)
    s(q) = s(q) - 1
    j = 0
  END IF
END IF
j = j + 1
WEND

```

Programmet har jeg tatt fra kilde [1], men jeg har tilpasset det Basic (det var opprinnelig i Java).

Program 7 – Gendrag.n.bas

Dette programmet, som opprinnelig var skrevet i TRS-80 Disk Basic, har jeg redigert slik at det kjører i QBasic.

```
1  REM **
2  REM **  GENDRAGN
3  REM **  written by Dan Rollins
4  REM **  4/29/82
5  REM **
6  REM **  Creates a "generalized" or "classical" dragon curve.
7  REM **  Use with DISKPLOTT subroutines or modify for
8  REM **  high-res video plotting.
9  REM **
10 CLEAR : DEFINT A-Z '** clear less if Out of Memory
30 RANDOMIZE TIMER: SCREEN 12: CLS : PRINT "Generalized Dragon Curve generator"
40 PRINT "programmed by Dan Rollins"
50 PRINT : INPUT "size (order of dragon curve)"; N
60 PRINT : PRINT "input Direction Reversal Sequence"
70 PRINT "(string of Ls and Rs ... Example: LLRRLRRR )"
80 PRINT "enter 'G' to generate a random sequence"
90 PRINT "default='L' (the classical dragon curve)"
100 lr$ = "L": INPUT lr$: lr$ = UCASE$(lr$): IF lr$ <> "G" THEN 140
110 lr$ = "": FOR j = 1 TO N
120 IF RND < .5 THEN lr$ = lr$ + "L" ELSE lr$ = lr$ + "R"
130 NEXT: PRINT "direction reversal sequence: "; lr$
140 se$ = lr$: l = LEN(lr$): last = 2 ^ N - 1: DIM d(last + 1)
150 sc = 2: INPUT "scale (1 - ??, default=2)"; sc: IF sc < 1 GOTO 150
160 PRINT : qa$ = "Y"
170 CLS : PRINT : PRINT "*** generating the dragon sequence ***"
180 sp = 1: ip = 1: IF LEFT$(lr$, 1) = "L" THEN d(1) = 1 ELSE d(1) = -1
190 FOR j = 2 TO N
200   ip = ip + 1: IF ip > l THEN ip = 1
210   IF MID$(lr$, ip, 1) = "L" THEN in = 1 ELSE in = -1
220   d(sp + 1) = in '** add DRS fold
230   FOR k = 1 TO sp
240     d(2 * sp + 2 - k) = -d(k) '** invert prior folds
250   NEXT
260   sp = sp * 2 + 1
270 NEXT
280 dx(2) = 1: dx(4) = -1: dy(1) = -1: dy(3) = 1
290 IF qa$ = "Y" GOTO 320
300 sx = 248: sy = 360: d = 1: INPUT "starting point (X,Y)"; sx, sy: INPUT "starting
   direction: (1)=North, (2)=West, (3)=South, (4)=East"; d
310 IF sx < 0 OR sx > 496 OR sy < 0 OR sy > p9 OR d < 1 OR d > 4 THEN 300 ELSE 460
320 sx = 0: sy = 0: d = 1: sd = 1: xh = -1: xl = 1: yh = -1: yl = 1
330 PRINT : PRINT "*** Centering ***"
340 FOR j = 1 TO last
350   sx = sx + dx(d): sy = sy + dy(d)
360   IF sx < xl THEN xl = sx ELSE IF sx > xh THEN xh = sx
370   IF sy < yl THEN yl = sy ELSE IF sy > yh THEN yh = sy
380   d = d - d(j): IF d < 1 THEN d = 4 ELSE IF d > 4 THEN d = 1
390 NEXT
400 xh = xh * (4 + sc): xl = xl * (4 + sc): yl = yl * (4 + sc): yh = yh * (4 + sc)
410 xd = xh - xl: yd = yh - yl
420 IF yd > 496 AND xd > 496 THEN PRINT USING "PLOT WON'T FIT ! it's #### wide,####
   high"; xd, yd: INPUT "(D)EFINE START OR (N)EW DRAGON"; qa$: IF qa$ = "D" GOTO 300
   ELSE IF qa$ = "N" THEN RUN ELSE 420
430 sx = (496 - xd) / 2 - xl: sy = (p9 - yd) / 2 - yl
440 IF xd > yd AND p9 > 496 THEN sd = 2: sx = (496 - yd) / 2 + yh: sy = (p9 - xd) / 2 -
   xl
```

```

450 '**
451 '** following code interprets the dragon sequence,
452 '** plotting the folds
453 '**
460 x = sx: y = sy: d = sd: CLS
470 FOR j = 1 TO last
480   FOR k = 1 TO sc
490     x = x + dx(d): y = y + dy(d): PSET (x, y + 240)** plot x,y
500   NEXT
510   d1 = d
520   d = d - d(j): IF d < 1 THEN d = 4 ELSE IF d > 4 THEN d = 1
530   FOR k = 1 TO 2
540     x = x + dx(d1) + dx(d): y = y + dy(d1) + dy(d)
550     PSET (x, y + 240)
560   NEXT
561 FOR DE = 1 TO 5000: NEXT
580 NEXT
581 LOCATE 1, 1: PRINT se$
610 I$ = "": WHILE I$ = "": I$ = INKEY$: WEND: END

```

Programmet er hentet fra kilde [23]

Program 8 – Determ.bas

Programmet bruker den deterministiske algoritmen til å lage en Sierpinski-trekant.

```
'Eksempel på den deterministiske algoritmen
'-----
'
'Fra sidene 86 - 87 Fractals Everywhere av Michael Barnsley (1993)
'
SCREEN 12
DIM s(100, 100), t(100, 100)

a(1) = .5: b(1) = 0: c(1) = 0: d(1) = .5: e(1) = 1: f(1) = 1
a(2) = .5: b(2) = 0: c(2) = 0: d(2) = .5: e(2) = 50: f(2) = 1
a(3) = .5: b(3) = 0: c(3) = 0: d(3) = .5: e(3) = 25: f(3) = 50

FOR i = 1 TO 100
  FOR j = INT(i / 2) TO 100 - i / 2
    t(j, i) = 1
    PSET (j, 101 - i)
  NEXT j
NEXT i

DO
  FOR i = 1 TO 100
    FOR j = 1 TO 100
      IF t(i, j) = 1 THEN
        s(a(1) * i + b(1) * j + e(1), c(1) * i + d(1) * j + f(1)) = 1
        s(a(2) * i + b(2) * j + e(2), c(2) * i + d(2) * j + f(2)) = 1
        s(a(3) * i + b(3) * j + e(3), c(3) * i + d(3) * j + f(3)) = 1
      END IF
    NEXT j
  NEXT i
  CLS
  FOR i = 1 TO 100
    FOR j = 1 TO 100
      t(i, j) = s(i, j)
      s(i, j) = 0
      IF t(i, j) = 1 THEN PSET (i, 101 - j)
    NEXT j
  NEXT i
LOOP WHILE INKEY$ = ""
```

Programmet er hentet fra kilde [17]

Program 9 – Rotasjon.bas

```
'ROTASJON.BAS
'-----

'Programmet følger reglene i kaos-spillet, men hvis det øverste hjørnet
'i den likesidede trekanten blir valgt, blir punktet dreid 90 grader om
'hjørnet i positiv omløpsretning (mot klokka)

'Trykk på en tast for å stoppe beregningene

'Kristoffer Andreas Berg, 14. februar 2003

DEF fnarcsin (x) 'funksjonen beregner invers sinus
IF x = 1 THEN fnarcsin = ATN(1) * 2
IF x = -1 THEN fnarcsin = -ATN(1) * 2
IF -1 < x AND x < 1 THEN fnarcsin = ATN(x / SQR(1 - x ^ 2))
END DEF

RANDOMIZE TIMER
SCREEN 12 'velger 640 x 480 pikslers grafikk
WINDOW (-.1, 1.6)-(1.1, -.1) 'innstiller koordinatsystemet

DIM a(7, 4)

'Leser inn konstanter

'r
a(1, 1) = .5
a(1, 2) = .5
a(1, 3) = .5
a(1, 4) = 0
's
a(2, 1) = .5
a(2, 2) = .5
a(2, 3) = .5
a(2, 4) = 0
'e
a(5, 1) = 0
a(5, 2) = .5
a(5, 3) = 1
a(5, 4) = 0
'f
a(6, 1) = 0
a(6, 2) = 1
a(6, 3) = 0
a(6, 4) = 0
'p
a(7, 1) = .33
a(7, 2) = .33
a(7, 3) = .34
a(7, 4) = 0

g = ATN(1) / 45

t = ATN(1) * 2

oldx = 0
oldy = 0
x = 0
y = 0
```

```

n = 0      'setter iterasjonstelleren til null
DO
    n = n + 1  'øker iterasjonstelleren med 1
    r = RND    'velger et tilfeldig tall

    v = 4
    IF r < a(7, 1) + a(7, 2) + a(7, 3) THEN v = 3
    IF r < a(7, 1) + a(7, 2) THEN v = 2
    IF r < a(7, 1) THEN v = 1

    x0 = a(5, v) 'x-koordinaten til det valgte hjørnet
    y0 = a(6, v) 'y-koordinaten til det valgte hjørnet
    dx = oldx - x0 'x-komponenten til avstanden fra punktet til hjørnet
    dy = oldy - y0 'y-komponenten til avstanden fra punktet til hjørnet
    lng = SQR(dx ^ 2 + dy ^ 2) + .000001 'lengden fra punktet til hjørnet
                                           '(jeg legger til en milliondel for
                                           'å unngå divisjon med null)
    vinkel = fnarcsin(dy / lng) 'vinkelen fra den positive delen av x-aksen
                                'til punktet, med toppunkt i hjørnet
    IF dx < 0 THEN vinkel = -vinkel + ATN(1) * 4 'korrigerer vinkelen hvis
                                                'den er i 2. eller 3. kvadrant
    IF v = 2 THEN q = t ELSE q = 0 'hvis det øverste hjørnet ble valgt, så
                                'la vinkelen punktet skal dreies være
                                '90 grader
    x = x0 + lng * a(1, v) * COS(vinkel + q) 'flytter til punktet oppnådd
    y = y0 + lng * a(2, v) * SIN(vinkel + q) 'ved å flytte halvveis mot hjørnet
                                           'og eventuelt dreie om hjørnet

    oldx = x
    oldy = y
    PSET (x, y) 'plotter punktet (x, y)

LOOP UNTIL INKEY$ <> ""
PRINT "Antall iterasjoner:"
PRINT n

```

Dette programmet har jeg laget selv.

Ordforklaringer

IFS-fraktaler

IFS er et akronym for Iterated Function System. Du har to variable x og y . IFS-fraktaler lages ved hjelp av en formel som, basert på verdiene av x og y , lager to nye verdier x_1 og y_1 . Så settes $x = x_1$ og $y = y_1$, og prosessen gjentas. Det betyr at resultatet av en beregningen brukes som grunnlag for neste beregning. Hver gjentakelse kalles en iterasjon.

AutoSketch

Et data-assistert konstruksjonsprogram (DAK-program) hvor brukeren kan konstruere tegninger bestående av linjer, sirkler, buer, ellipser og elliptiske buer. Målene oppgis helt nøyaktig, og derfor blir tegningen akkurat slik du vil ha den.

Den pytagoreiske læresetningen

En setning som sier at i en rettvinklet trekant, er kvadratet av hypotenusen (den lange siden) lik summen av kvadratene av katetene (sidene som danner den rette vinkelen). Det finnes massevis av bevis for denne setningen, men vi tar ikke med noen her.

Irrasjonale tall

Tall som ikke kan skrives som en brøk med hele tall i telleren og nevneren. Irrasjonale tall er desimaltall med uendelig mange desimaler. Det finnes ikke noe mønster som gjentar seg i desimalene til irrasjonale tall, de er altså ikke periodiske. Eksempler på irrasjonale tall er $\sqrt{2}$ og π .

Iterasjon

Gjentatt beregning. Når en IFS-fraktal skal beregnes, er hver beregning av x og y en iterasjon. I Mandelbrot- og Julia-settene kalles hver beregning av funksjonen for en iterasjon. Vi kan også se på beregningen av et punkt i kaos-spillet som en iterasjon.

Ekstrapolere

Beregne verdier som ligger utenfor en serie verdier man kjenner til.

Vektor

En vektor beskriver en størrelse med retning. Bevegelse er et eksempel på noe som må angis som en vektor. I matematikk skriver vi en vektor som en pil. Lengden til pila viser størrelsen til vektoren. Vektorer adderes ved å flytte den ene vektoren (uten å forandre retningen) og legge den slik at den begynner der den andre vektoren slutter. Summen blir en vektor som begynner der den første vektoren begynner, og slutter der den andre vektoren slutter.

Radian

Radianer kalles også absolutt vinkelmål. En vinkel i radianer måles ved å dele buelengden på radius i sirkelen (en sirkel som har samme radius som buen). Siden omkretsen O av en sirkel med radius r er $O = 2\pi r$, blir hele sirkelen delt inn i 2π radianer.

Imaginær

Et imaginært tall er et tall som multiplisert med seg selv gir et negativt reelt tall. Tallet som gir -1 når det kvadreres, kaller vi den imaginære enhet og betegner det med i .

Ortonormert koordinatsystem

Et koordinatsystem der aksene står vinkelrett på hverandre og enhetene langs de to aksene er like.

Piksel

Den beste norske oversettelsen for «piksel» er «bildepunkt». Ordet piksel kommer fra ordet «pixel», som er en sammenslåing av to ord, PICTure og ELe ment, som på godt amerikansk blir «pixel». En piksel er den minste enheten i et digitalt bilde, og kan ikke vise mer enn én farge om gangen. Dersom du i et bildeprogram zoomer så langt du kan inn på et digitalt bilde, vil de se pikslene tydelig. På dataskjermer er piksler kvadratiske.

Kilder

Internettider

- [1] <http://hale.nrsd.net/fractals/trees.html>
- [2] http://www.cevis.uni-bremen.de/fractals/nsfpe/Pythagorean_Trees/PT.html
- [3] <http://math.bu.edu/DYSYS/chaos-game/node7.html>
- [4] <http://math.bu.edu/DYSYS/chaos-game/node8.html>
- [5] <http://www-history.mcs.st-and.ac.uk/Mathematicians/Cantor.html>
- [6] <http://www-history.mcs.st-and.ac.uk/Mathematicians/Julia.html>
- [7] <http://www-history.mcs.st-and.ac.uk/Mathematicians/Koch.html>
- [8] <http://www-history.mcs.st-and.ac.uk/Mathematicians/Sierpinski.html>
- [9] <http://www-history.mcs.st-and.ac.uk/Mathematicians/Mandelbrot.html>
- [10] <http://home.hia.no/~fgill/>
- [11] <http://classes.yale.edu/99-00/math190a/Snow2.html>
- [12] <http://classes.yale.edu/99-00/math190a/IFSExercAns.html>
- [13] <http://classes.yale.edu/99-00/math190a/FernIFS.html>
- [14] <http://www.home.aone.net.au/byzantium/ferns/fractal.html>
- [15] http://www.apollon.uio.no/2002_4/artikler/snokrystaller.shtml

Bøker

- [16] Heinz-Otto Peitgen, Hartmut Jürgens og Dietmar Saupe: Fractals for the Classroom, Springer-Verlag 1992
- [17] Michael Fielding Barnsley: Fractals Everywhere (Second Edition), Academic Press 1993
- [18] Karl Erik Sandvold og Stein Øgrim: Matematikk på grunnkurs, Gyldendal Norsk Forlag A/S 1994
- [19] Tor Gulliksen: Matematikk i praksis, Universitetsforlaget AS 1981
- [20] Tor Guttu: Norsk Illustrert Ordbok, Kunnskapsforlaget 1993
- [21] Irvng Adler: A New Look At Arithmetic, The New American Library 1964
- [22] Aschehougs konversasjonsleksikon, fjerde utgave 1961

Tidsskrifter

- [23] Byte, desember 1983
- [24] PC Plus, mars 1994

Andre

[25] Göteborgs Universitet, Gymnasielärarprogrammet, vt-2000, Fraktaler – fraktal geometri – en kort introduktion om dimensionsbegreppet och IFS – Arbetet utförd av Puskar Mersiha, 2000-05-01

Stikkordregister

A

absoluttverdi	33
Alexander von Humboldt-prisen	59
analytisk sett	54
argument	33
Aristoteles	12
AutoSketch.....	9; 75

B

Banachiewicz	54
Barnard Medal for Meritorious Service to Science	59
Barnsleys bregne	66
Basic	62
Bendixson	51
Berg, Gunnar Andreas.....	6
Berlin.....	48; 57
Black Spleenwort-bregnen.....	66
Brauer	57
bregne	8; 23
Bregne.bas.....	66
bregneblad.....	23
Böhm, Maria Anna.....	48

C

California Institute of Technology	59
Cantor, Georg Ferdinand Ludwig Philipp	9; 48; 54
Cantor, Georg Waldemar.....	48
Cantor-Dust.....	9
Cantor-settet.....	8; 9
Centre National de la Recherche Scientific	59
Collège de France.....	58
cosinus	35
Cremer, H.	57

D

DAK-program	75
Dedekind.....	49
Delphi	5
deterministiske algoritmen, den	15
Deutsche Mathematiker-Vereinigung	50
Dirichlet.....	49
dragekurve, konstruksjon av	30
dragekurver	8; 30

E

École Normale.....	58
École Polytechnique	57; 58; 59
Egorov	54
Einstein College of Medicine.....	59
ekstrapolere.....	32; 75
en-til-en-korrespondanse	49
Euklid	14

F

Fallenius, Mats	6
fargeforandringer.....	42
fargen til et punkt	37
forflytninger	20
Fractals Everywhere.....	5
Fractals for the Classroom	5
fraktaler.....	8
fraktalgeometri	58
Frankfurt	48
Franklin-medaljen	59
Fundamenta Mathematica.....	54

G

galakser	26
Gendrag.bas.....	70
Gilbo, Hanne	6
Gill, Frode	5
Goldbachs formodning.....	49
Gulbrandsen, Bente Kari.....	6
Guttman, Vally	49
Göttingen.....	48; 54

H

Halle.....	48; 49
Hardy.....	58
Harvard University	59
Heighways, John E.	31
Heine.....	49
Hofstadter, Douglas	30
Hopf	57
Höheren Gewerbeschule in Darmstadt	48

I

IBM.....	59
IFS-fraktaler	8; 23; 75
imaginær	76
imaginær akse	37
imaginærdel.....	33; 37
imaginære enhet, den.....	33
inkommensurabel	12
Institute for Advanced Study.....	59
Interlaken	49
Internasjonal Matematikerkongress i Heidelberg.....	50
iterasjon.....	16; 23; 25; 37; 75
Iterated Function System.....	75

J

Julia, Gaston Maurice	57; 59
Julia.bas.....	64
Julia-settet	8; 33; 42; 57; 64

K

Kagan, Ailette	59
Kaos.bas	62
kaos-spillet	15; 62
kaos-spillet, rotasjoner i	20
kilder	77
Knuth, Donald	30
Koch, Niels Fabian Helge von	10; 51
Koch.bas	65
koeffisient, heltallig	49
komplekse planet, det	37; 42
komplekse tall, addisjon av	34
komplekse tall, kvadrering av	34
komplekse tall, multiplikasjon av	34
konstanttabell	24
koordinatsystem	23; 37; 76
koordinatsystem, ortonormert	33; 76
Krakow	53
Kronecker	48
Kummer	48
Kungliga Teknologiska Högskolan i Stockholm	51

L

Lévy, Paul	58
likesidet trekant, høyden i en	24
Lipschitz	49
Lutzin	54
Lvov	54
Lycée Rolin	58
Lyon	58

M

Mandel.bas	63
Mandelbrojt, Szolem	58
Mandelbrot, Benoit	57; 58
Mandelbrot-settet	8; 33; 37; 42; 59; 63
Matematisk Forening i Berlin	49
Mathematical Games	30
Mathematische Annalen	49
Metamagical Themas	30
Mittag-Leffler	49; 51

N

Nevada-medaljen	59
New York	59

O

oddetall	50
orden	30
orientalsk drage	30

P

papirstrimmel	31
Paris	57
partall	49

Pascal	5
PC Plus	5
piksel	37; 76
Polens Matematiske Selskap	54
polynomlikning	49
Polyteknisk skole i Zürich	48
primtall	50
Princeton	59
pytagoreisk tre	8; 12; 68
pytagoreiske læresetningen, den	12; 75
Pythagor.bas	68
Pythagoras	12

Q

QBasic	10; 70
--------------	--------

R

radianer	33; 75
rasjonale funksjoner, iterasjon av	57
realdel	33; 37
Realschule in Darmstadt	48
reell akse	37
Reidemeister	57
rekursiv teknikk	32
Riemann	49
rotasjoner	20
Rotkiewicz	56

S

Schellbach Seminar	49
Schwarz, Herman	48
Scientific American	30
sekvens, reversert	32
selv-liknende	17
sett-teori	49; 54
Sidi Bel Abbès	57
Sierpinski, Waclaw	15; 53
Sierpinski-femkanten	19
Sierpinski-kurven	54
Sierpinski-kvadrat	19
Sierpinski-sekskanten	18; 19
Sierpinski-trekant, forblåst	22
Sierpinski-trekanten	8; 15; 17; 24; 62
sinus	35
skaleringsfaktor	17
snøkrystall, bilde av en	10
snøkrystall, fraktal	25
snøkrystall, von Kochs	52; 65
St. Petersburg	48
Steinmetz-medaljen	59
Stockholm	51
Stockholms Högskola	51
Stockholms Universitet	51
streng	31
streng, invertert	32
Swift, Jonathan	4

Sølberg, Tone6

T

tall, absolutt normalt54
tall, algebraisk49
tall, hele33
tall, irrasjonale 14; 33; 49; 75
tall, komplekse33
tall, naturlige33
tall, rasjonale33
tall, reelle33
tallinje33
tallmengder33
tangens35
tegn, invertere32
The fractal geometry of nature59
TRS-80 Disk Basic70
Tulle58

U

Universitetet i Warszawa53; 54
Uppsalas Universitet51

V

vektor 33; 75
Viatkan 54
vinkelmål, absolutt 75
von Koch-kurven 8; 10; 19
Voronoy 53

W

Warszawa 53; 58
Warszawas Vitenskapelige Selskap 54
Watson Research Center 59
Weierstrass 48; 49
Wiesbaden 48
Wolf-prise 59

Y

Yale 59
Yorktown Heights 59

Z

Zaremba 54