3. Geïtereerde functiesystemen

In de ontwikkeling van allerhande toepassingen wordt de dag van vandaag gebruik gemaakt van geïtereerde functiesystemen. Bijvoorbeeld in het hedendaags multimediaal computertijdperk is het belangrijk om beelden te kunnen stockeren en te genereren met zo weinig mogelijk informatie.

De klassieke meetkunde stelt ons in staat Euclidische begrippen zoals bijvoorbeeld rechten en cirkels exact te beschrijven. Maar deze meetkunde biedt ons niet de middelen om wolken, varens, bomen, bergen, melkwegstelsels, ... te beschrijven. De fractaal-meetkunde is hiervan een mooie uitbreiding. De fractaal-meetkunde kan je beschouwen als een nieuwe taal. Eens je de taal spreekt, kun je de vorm van een wolk beschrijven zoals een architect een huis beschrijft.

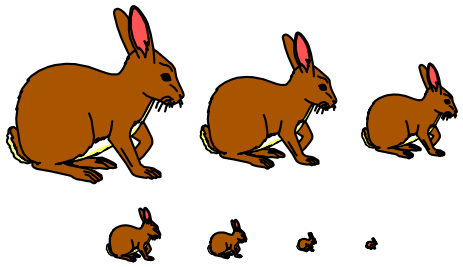
In de vorige hoofstukken bestudeerden we steeds dynmamische processen en hun evolutie in de tijd. Dit ging steeds gepaard met het meten van afstanden: absolute waarden in $\mathbb{R}$ en modulus in $\mathbb{C}$. Beide zijn concrete gevallen van het algemeen begrip metriek. In wat volgt zouden we om exact te werken een afstand moeten definiëren tussen figuren, maar dit zou ons echter te ver leiden.

In dit hoofdstuk introduceren we geïtereerde functiesystemen, een topic uit de fractaal-meetkunde. Maar eerst voeren we enkele wiskundige begrippen in.

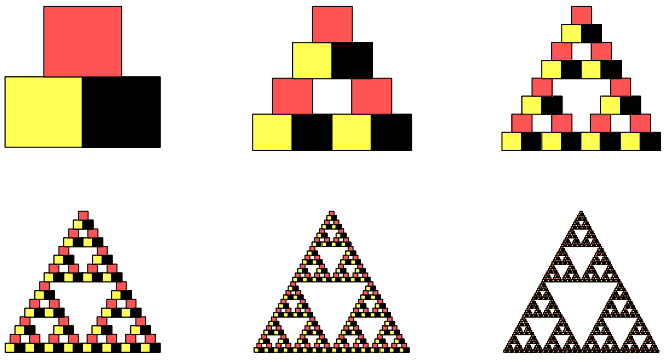
3.1 Contracties

Veronderstel dat we beschikken over een kopieermachine met een reductie-functie, zodat het resultaat van een reductie gelijkvormig is met het origineel.

Indien we van de reductie opnieuw een reductie nemen en van deze laatste reductie opnieuw een reductie, ..., zien we na enkele stappen dat de originele figuur gereduceerd zal worden tot een punt.



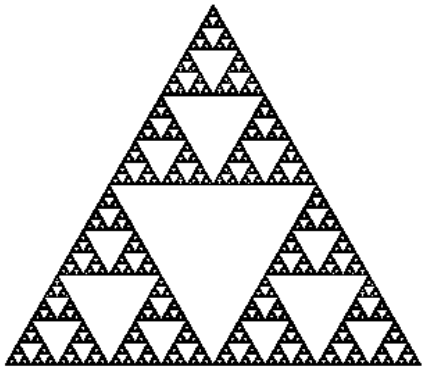
De reductie gebeurt met een lens-systeem. We breiden onze machine uit tot een kopieermachine met drie lens-systemen waarbij ieder systeem een reductie creëert en de reductie ergens op papier plaatst. Veronderstel dat ieder systeem een reductiefactor 0.5 heeft. In de volgende figuur zien wat er gebeurt met een rechthoek indien we iedere reductie telkens weer als origineel gebruiken.

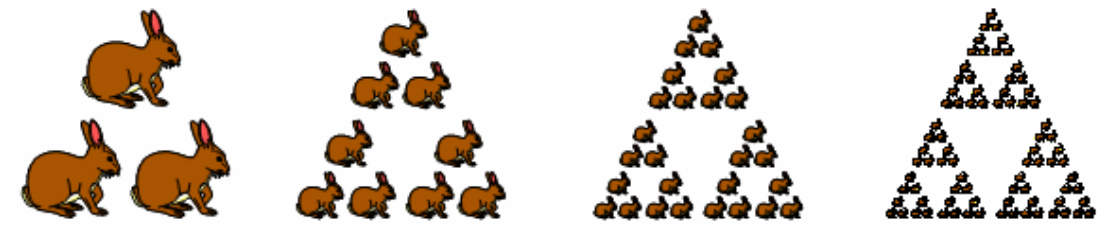


De lengte van de zijden van de vierkanten neemt zeer snel af, maar in dit geval wordt het beeld niet gereduceerd tot een punt.

Theoretisch kan men dit proces oneindig vaak verder zetten. De fractaal die uit dit limietproces ontstaat noemt men de *Driehoek van Sierpinski.*

Een eigenaardige eigenschap van onze machine is dat, vertrekkende van de meeste figuren, het resultaat telkens de driehoek van Sierpinski is. De driehoek van Sierpinski noemt men de attractor van de machine.





Indien we de machine in een formeel wiskundig kader willen omschrijven, kunnen we ze beschouwen als een verzameling affiene transformaties.

In wat volgt werken we steeds in het vlak, genoteerd door $\mathbb{R}^2$.

Een affiene transformatie W in $\mathbb{R}^2$ is een afbeelding die als volgt gedefinieerd is:

$$\begin{pmatrix} x \\ y \end{pmatrix} \overset{W}{\longmapsto} \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \text{ met } a,b,c,d,e,f \in \mathbb{R}.$$

Hierbij bepaalt het gedeelte $\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$ een lineaire transformatie en $\begin{pmatrix} e \\ f \end{pmatrix}$ een translatie.

Een affiene transformatie die het origineel transformeert in een gelijkvormig beeld noemen we een gelijkvormigheid. In het geval van een gelijkvormigheid bestaat het lineaire gedeelte uit de samenstelling van een homothetie en een rotatie.

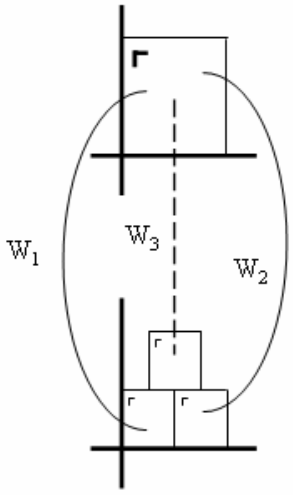
Een gelijkvormigheid heeft als voorschrift $\begin{pmatrix} x \\ y \end{pmatrix} \overset{W}{\longmapsto} r\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$ waarbij r de schaalfactor is en θ de rotatiehoek.

De machine die we ontwikkeld hebben, is opgebouwd uit drie affiene transformaties W_1, W_2 en W_3.

$$W_1 : \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$$

$$W_2 : \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix}$$

$$W_3 : \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{4} \\ \frac{1}{2} \end{pmatrix}$$



Een transformatie in $\mathbb{R}^2$ noemen we ruwweg een contractie als de punten van het beeld dichter bij elkaar liggen dan bij het origineel. In het geval van een gelijkvormigheid hebben we voor $0 < r < 1$ een contractie.

Meer specifiek noemen we een transformatie W in $\mathbb{R}^2$ een contractie indien er een $0 \le r < 1$ bestaat met de eigenschap: $d(W(z_1), W(z_2)) \le r\, d(z_1, z_2)$ waarbij $d(p_1, p_2)$ de afstand voorstelt tussen de punten p_1 en p_2.

r noemt men de contractiefactor.

Een contractieve gelijkvormigheid kan ook als volgt genoteerd worden:

$$\begin{pmatrix} x \\ y \end{pmatrix} \overset{W}{\longmapsto} r \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

waarbij we (x_0, y_0) kunnen afleiden uit de translatie (e, f) en vice versa.

De vraag die we ons stellen is wat er gebeurt als we een contractieve transformatie W itereren. M.a.w. voor $z \in \mathbb{R}^2$ beschouwen we de baan

$z \mapsto W(z) \mapsto W^{(2)}(z) \mapsto W^{(3)}(z) \mapsto \ldots.$

CONTRACTIESTELLING

Voor een contractie $W : \mathbb{R}^2 \to \mathbb{R}^2$ bestaat er juist één punt $z_0 \in \mathbb{R}^2$ met de eigenschap dat $W(z_0) = z_0$ en voor alle $z \in \mathbb{R}^2$ geldt : $W^{(n)}(z) \mapsto z_0$ voor $n \mapsto +\infty$.

Tevens geldt voor ieder begrensd deel A van $\mathbb{R}^2$ dat $W^{(n)}(A) \mapsto z_0$, voor $n \mapsto +\infty$.

z_0 noemen we het fixpunt of vast punt van de contractie W.

Merk op dat in tweede uitdrukking voor een gelijkvormigheid met $0 < r < 1$, $z_0 = (x_0, y_0)$ het vast punt is.

3.2 Geïtereerde functiesystemen (IFS)

Een IFS, W, bestaat uit een eindig aantal contractieve transformaties, $W_1, W_2, \ldots, W_n$, die we als volgt laten inwerken op begrensde delen van $\mathbb{R}^2$:

$A \longrightarrow W_1(A) \cup W_2(A) \cup \ldots \cup W_n(A)$.

We kunnen ook hier W itereren. Stellen we $W(A) = B$ dan geldt:

$W^{(2)}(A) = W(B) = W_1(B) \cup W_2(B) \cup \ldots \cup W_n(B)$. En verder $W^{(3)}(A)$, $W^{(4)}(A)$, $W^{(5)}(A)$,.....

Weerom stellen we ons de vraag waarnaar $W^{(n)}(A)$ zal streven als we n naar oneindig laten streven. De volgende stelling geeft hierop een antwoord.

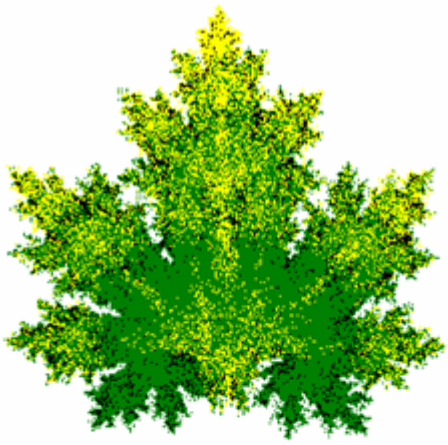
CONTRACTIESTELLING VOOR IFS

Voor een IFS $W : \mathbb{R}^2 \to \mathbb{R}^2$ bestaat er juist één gesloten en begrensde $F \subset \mathbb{R}^2$ zodat voor elke begrensde $D \subset \mathbb{R}^2$ geldt dat $W^{(n)}(D) \to F$, voor $n \to +\infty$.

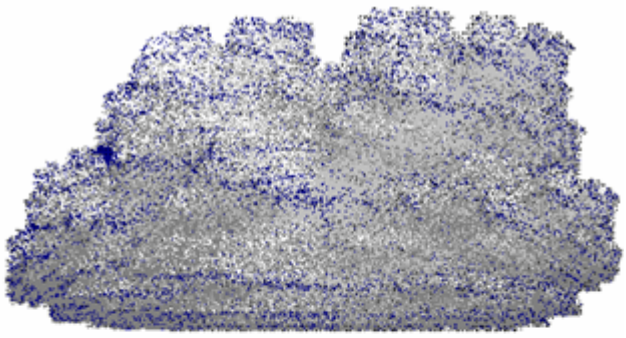
De verzameling F noemt men de attractor van het IFS. Voor de attractor F van een IFS geldt $F = W(F)$. M.a.w. de attractor kan beschouwd worden als het vast punt van het IFS.

Vaak zijn de W_i's, zoals in het geval van de driehoek van Sierpinski, gelijkvormigheden zodat F en $W_i(F)$ gelijkvormig zijn. In dit geval noemt men F zelfgelijkvormig, een basiseigenschap van fractalen.

Zelfs met geïtereerde functiesystemen bestaande uit een klein aantal contracties, al dan niet gelijkvormig, kunnen we heel wat mooie plaatjes genereren.

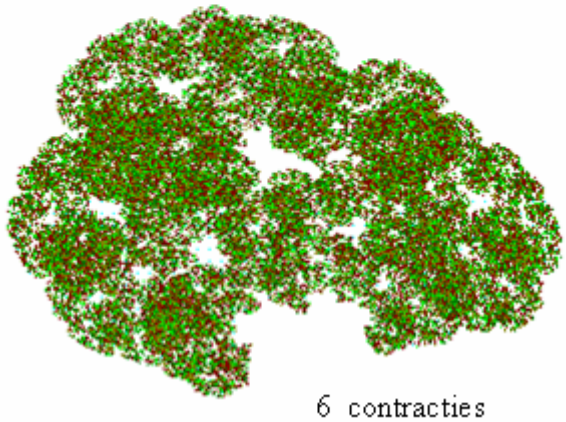


4 contracties



6 contracties

4 contracties



6 contracties

De laaste figuur lijkt op een kruin van een boom, maar dan zonder takken en zonder stam. In wat volgt formuleren we een resultaat dat ook de takken en de stam genereert.

3.3 IFS met Condensatie

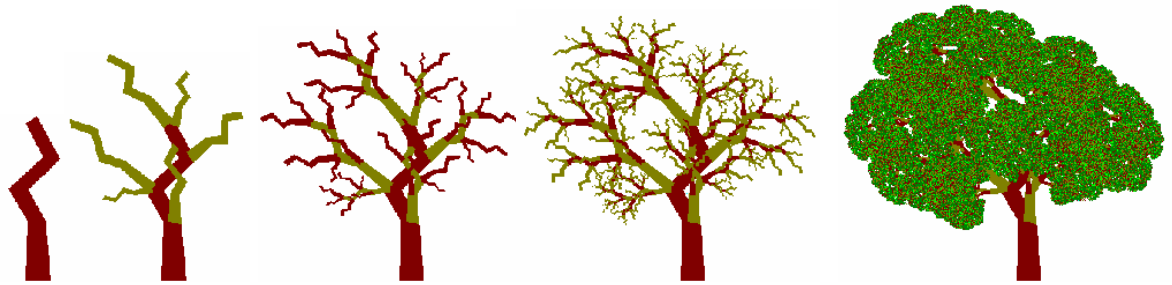
Deze variant van geïtereerde functiesystemen, geïtereerde functiesystemen met condensatie, bestaat uit een IFS, $(W_1,\ldots,W_n)$, en een gesloten begrensd deel K van $\mathbb{R}^2$.

Voor deze variant stellen we voor een begrensd deel A van $\mathbb{R}^2$:

$W(A) = W_1(A) \cup \ldots \cup W_2(A) \cup K$.

Weerom bezit W één gesloten begrensde attractor F met $W(F) = F$.

In het volgende voorbeeld zorgt de stam voor condensatie en de ontwikkeling van takken en bladeren.



In het voorschrift van een affiene transformatie $\begin{pmatrix} x \\ y \end{pmatrix} \overset{w}{\mapsto} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$ vinden we de parametes $a, b, c, d, e, f \in \mathbb{R}$.

Indien we deze parameters voor een IFS op een continue manier veranderen, zal ook de attractor op een continue manier mee veranderen. Het achter elkaar plaatsen van deze verschillende plaatjes kan leiden tot bewegende beelden.

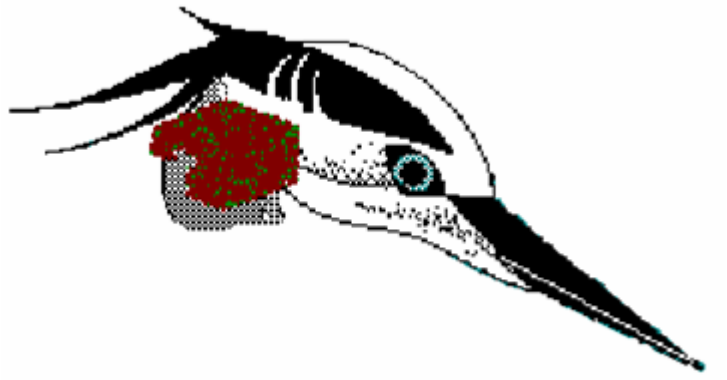
3.4 Beeldverwerking

Het nabootsen van landschappen was één van de eerste toepassingen van fractalen. Vertrekkende van een bestaande figuur is het mogelijk fractalen te vinden die deze figuur met iedere gewenste graad van precisie kunnen weergeven. Omdat fractalen zelf zeer compact kunnen worden beschreven, hebben we hiermee de mogelijkheid de originele afbeelding op te slaan in sterk gecomprimeerde vorm.

Michael Barnsley, *A better way to compress images*, BYTE, 1988.

Verfijnde beeldverwerkingssystemen die gebruik maken van fractalen zijn zeer duur en maken zelfs gebruik van speciale hardware voor het herkennen van patronen en gebruiken bibliotheken met voorgedefinieerde fractalen die bij de patronen passen.

We eindigen met een redelijk eenvoudig voorbeeld ter illustratie van deze werkwijze.
Om onderstaande bitmap op te slaan hebben we 69 074 bytes nodig.



Met een IFS met condensatie, 693 bytes, kunnen we het resultaat rechts onderaan genereren. De figuur links toont de eerste iteratie van het IFS, bestaande uit 17 contracties. De kleuren kunnen eenvoudig bijgesteld worden maar een nauwkeurige benadering vergt meer tijd om het geschikte IFS te vinden, maar vereist daarom niet veel meer bytes.