

Fraktale Landschaften¹

Christian Pfligersdorffer
Matrikelnr. 9720764

Christian Spielberger
Matrikelnr. 9820979

8. Juli 2002

¹Diese Arbeit entstand im Rahmen des Proseminars Computergraphik II (Sommersemester 2002) an der Universität Salzburg unter der Leitung von Univ.-Prof. Dr. Martin Held.

Zusammenfassung

Ziel dieser Arbeit ist es, einige Methoden zur Erzeugung fraktaler Landschaften vorzustellen. Wir untersuchen die mathematischen Hintergründe und stellen uns die Frage, wie man Objekte der Natur mathematisch beschreiben kann.

Eine geometrische Beschreibung eignet sich aufgrund der hohen Komplexität zumeist nicht, jedoch können natürliche Objekte wie Bäume, Sträucher, Landschaften, Planeten, Küstenlinien, Flüsse, Seen und vieles mehr mit einer gewissen Realitätsnähe durch fraktale Objekte angenähert werden. Eine umfangreiche Zusammenstellung über Chaos und Fraktale findet man in [4]. Einige Algorithmen zum Erstellen von fraktalen Pflanzen, Landschaften und Planeten findet man in [5].

1 Selbstähnlichkeit

Benoit B. Mandelbrot leitete das Wort „Fraktal“ vom lateinischen Verb *frangere* und vom zugehörigen Adjektiv *fractal* ab. Das Verb bedeutet brechen, unregelmäßige Fragmente bilden. Doch Phänomene der Natur, wie zum Beispiel die *Brownsche Bewegung* sind schon viel länger bekannt, nur nicht unter der Bezeichnung *Fraktale*.

Grundlegend für alle Fraktale ist das Prinzip der Selbstähnlichkeit. Betrachtet man ein fraktales Objekt und einen bestimmten Teil des Objekts mit einer passenden Skalierung, so erscheinen die beiden Bilder dem Betrachter identisch (oder in einer etwas weiter gefaßten Anschauung zumindest „ähnlich“). Dieses Phänomen kann man in der Natur oft beobachten.

Ein Baumstamm teilt sich in mehrere große Äste in ähnlicher Weise, wie sich Äste wieder in kleinere Äste aufteilen – bis zu den kleinsten Ästchen. Ein weiteres Beispiel dafür sind Flussdeltas.

Eine Bergwand weist von der Ferne betrachtet eine ähnliche Struktur auf, wie eine Felswand dieses Berges, und noch näher betrachtet, eine, ähnlich der Oberfläche des Gesteins.

Der Mond ist übersät mit Kratern. Egal mit welcher Skalierung wir die Mondoberfläche betrachten – wir sehen große Krater genauso wie mittelgroße und ganz kleine.

1.1 Die Koch-Kurve

Dies ist eine sehr einfach aufgebaute fraktale Kurve. Helge von Koch, ein schwedischer Mathematiker, veröffentlichte 1904 seine *Koch-Kurve*.

Wir starten mit einer waagrechten Strecke, teilen diese in drei gleichgroße Teile und ersetzen das mittlere Stück durch die linke und rechte Seite eines gleichseitigen Dreiecks. So erhalten wir vier Strecken, wie man in Schritt 1, Abbildung 1 sieht. Nun wenden wir dieses Verfahren in rekursiver Weise auf jede der vier Teilstrecken an. Abbildung 1 soll die Konstruktion verdeutlichen. Abbildung 2 zeigt die Kochkurve nach 11 Iterationsschritten.

Durch das rekursive Konstruktionsverfahren ergibt sich die Selbstähnlichkeit. Aus diesem Grund sind die meisten Fraktale durch ein rekursives Schema definiert.

1.2 Das Sierpinski-Dreieck

Ein weiteres Beispiel ist das *Sierpinski-Dreieck* (Abb. 3). In einem gleichseitigen Dreieck werden die Halbierungspunkte der Seiten miteinander verbunden und das dadurch entstandene, in der Mitte liegende Dreieck entfernt. Abbildung 4 zeigt das Sierpinski-Dreieck nach 10 Iterationsschritten.

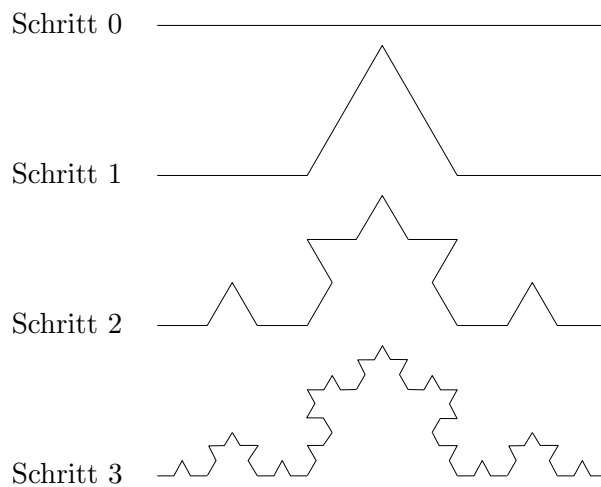


Abbildung 1: Konstruktion der *Koch*-Kurve

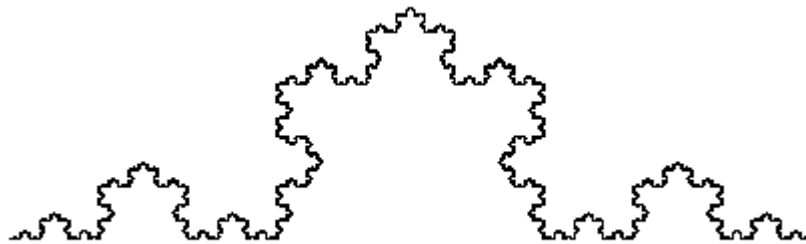


Abbildung 2: *Koch*-Kurve

2 Fraktale Dimension

Um den Grad der Komplexität eines fraktalen Objekts zu charakterisieren, führen wir den Begriff der fraktalen Dimension ein. Wir wollen ein Maß dafür angeben, wie stark zerklüftet eine Linie, Oberfläche, ein Baum usw. ist. Möchte man beispielsweise die Länge der Küste Englands messen, so hängt das Messergebnis von der verwendeten Maßeinheit ab. Umso feiner und genauer die Messung, desto Länger wird die Küstenlinie erscheinen. Die fraktale Dimension soll dabei ein Maß angeben, wie stark die Länge mit der Feinheit der Messung ansteigt.

Nun gibt es bereits eine Handvoll für Fraktale vorgeschlagener Dimensionsbegriffe, wie zum Beispiel:

- Selbstähnlichkeitsdimension (self-similarity dimension)

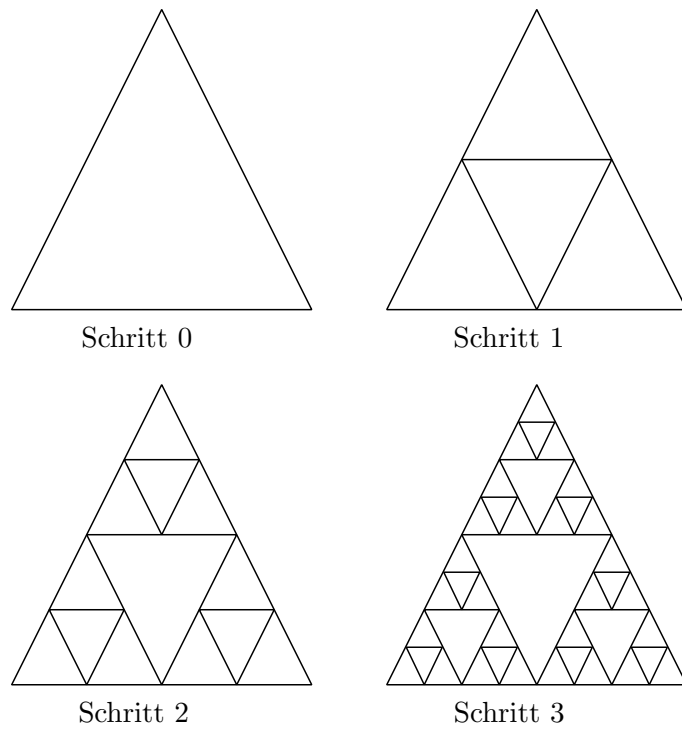


Abbildung 3: Konstruktion des *Sierpinski*-Dreiecks

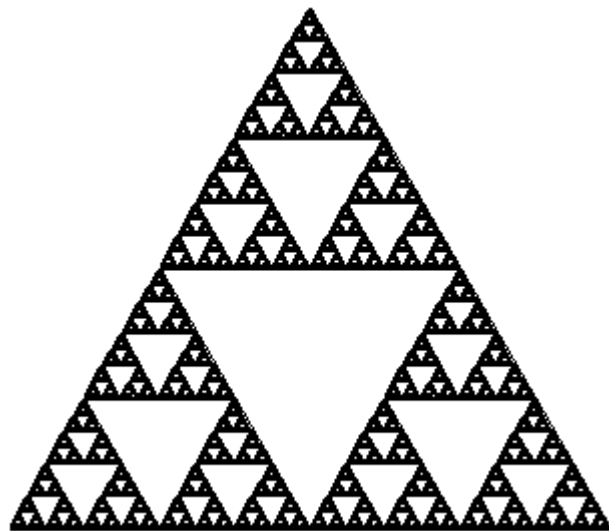


Abbildung 4: *Sierpinski*-Dreieck

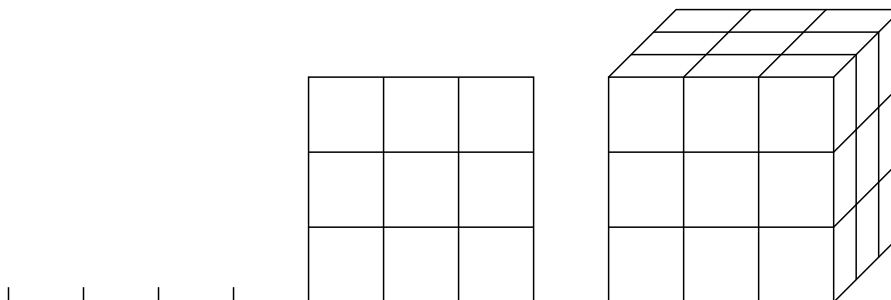


Abbildung 5: Selbstähnlichkeit von Strecke, Quadrat und Würfel

- Kompassdimension (compass dimension)
- Box-Dimension (box-counting dimension)

Diese sind spezielle Formen der Mandelbrotschen *fraktalen Dimension*.

2.1 Selbstähnlichkeitsdimension

Eine Struktur heißt selbstähnlich, wenn sie in beliebig kleine Teile aufgeteilt werden kann, wobei jedes Teilstück eine kleine Kopie des Ganzen ist [4].

Skalieren wir eine Strecke mit dem Faktor $s = 1/3$, so erhalten wir $a = 3$ Teile. Machen wir das selbe für ein Quadrat, so erhalten wir $a = 9$ Teile und für einen Würfel mit $s = 1/3$ bekommen wir $a = 27$ Teile (Abb. 5).

Es wird schnell klar, dass die Anzahl a der entstehenden Teile die folgende Gleichung erfüllt:

$$a = \frac{1}{s^D}, \quad (1)$$

wobei $D = 1$ für die Strecke, $D = 2$ für das Quadrat und $D = 3$ für den Würfel ist.

Betrachten wir nocheinmal die Koch-Kurve, so sehen wir, dass bei Skalierung $1/3$ hier 4 Teile entstehen. Formen wir Gleichung 1 um, erhalten wir

$$D = \frac{\log a}{\log 1/s}$$

und es ergibt sich somit für die Koch-Kurve eine fraktale Dimension von

$$D = \frac{\log 4}{\log 3} \approx 1.2619.$$

Weitere bekannte Fraktale mit ihren Dimensionen [4]:

Cantormenge	$\log 2 / \log 3 \approx 0.6309$
Sierpinski-Gasket	$\log 3 / \log 2 \approx 1.585$
Sierpinski-Teppich	$\log 8 / \log 3 \approx 1.8928$

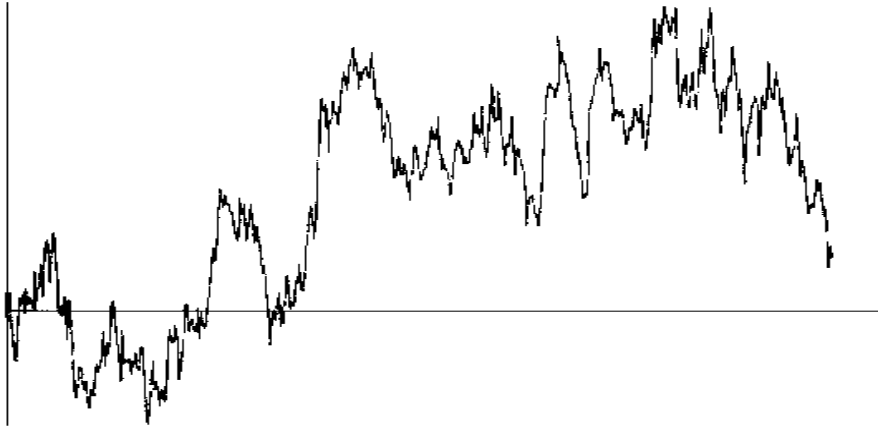


Abbildung 6: Brownsche Bewegung

3 Fraktale Brownsche Bewegung

Der Botaniker Robert Brown entdeckte 1827, dass kleinste Teilchen, unter dem Mikroskop betrachtet, heftige Zickzackbewegungen vollführen. Dieses Phänomen wurde später als nach ihm als *Brownsche Bewegung* benannt. Brownsche Bewegung ist die Basis für die Erstellung fraktaler Objekte wie etwa Landschaften. Betrachten wir zunächst eine eindimensionale Bewegung. Die horizontale Achse in Abbildung 6 ist die Zeitachse und die senkrechte Achse stellt die Position des Teilchens dar.

3.1 Selbstähnlichkeit der Brownschen Bewegung

Eine zentrale Eigenschaft von Fraktalen ist, wie wir schon wissen, die Selbstähnlichkeit. Bei einer Vergrößerung der fraktalen Kurve darf sich ihr Charakter, also ihre Dimension nicht verändern.

Die Brownsche Bewegung ist ein Zufallsprozess $X(t)$, wobei die Änderungen normalverteilt sind und

$$\text{var}(X(t_2) - X(t_1)) \propto |t_2 - t_1|^{2H}, \quad (2)$$

ist mit $H = 1/2$. Diese Bedingung sichert die Selbstähnlichkeit der Kurve $X(t)$. Die Verallgemeinerung für Parameter $0 < H < 1$ wird *fraktale Brownsche Bewegung* genannt. Man sagt auch, die Änderungen von X sind statistisch selbstähnlich mit Parameter H , also

$$X(t) - X(t_0) \quad \text{und} \quad \frac{X(rt) - X(t_0)}{r^H}$$

sind statistisch identisch verteilt für alle t_0 , t und $r > 0$. Setzen wir nun $t_0 = 0$ und $X(t_0) = 0$, dann erhalten wir:

$$X(t) \quad \text{und} \quad \frac{X(rt)}{r^H}$$

sind statistisch identisch verteilt. Der Parameter r gibt dabei die Skalierung der Zeitachse an. Um die Charakteristik der Kurve zu erhalten, muss man also bei Skalierung der Zeitachse mit r die Amplitude von $X(rt)$ mit $1/r^H$ skalieren.

Der Parameter H wird auch *Hurst-Exponent* genannt und hängt mit der fraktalen Dimension zusammen. Sei nun E die euklidische Dimension der Brownschen Bewegung, also in unserem Fall $E = 1$, so gilt

$$H = E + 1 - D,$$

wobei D die fraktale Dimension ist.

3.2 Mittelpunktverschiebungs-Algorithmus

Eine naheliegende und weit verbreitete Methode, um fraktale Brownsche Bewegung zu erzeugen, ist die *zufällige Mittelpunktverschiebung*. Wie viele der Algorithmen, die Fraktale erzeugen, funktioniert auch dieser rekursiv. Wir wollen eine fraktale Brownsche Bewegung

$$X : [0, 1] \rightarrow \mathbb{R}$$

definieren. Dazu legen wir zunächst $X(0) = 0$ und $X(1) = 0$ fest. Dann teilen wir die Zeitachse in der Mitte und verschieben $X(\frac{1}{2})$ ausgehend vom Mittelwert von $X(0)$ und $X(1)$, in diesem Fall 0, um einen zufälligen Wert nach oben bzw. nach unten. Wir fahren dann in rekursiver Weise für beide Teilintervalle fort (Algorithmus 1).

Bei jedem Aufruf wird das Intervall $[l, r]$ übergeben, für das der Mittelpunkt verschoben werden soll. Zusätzlich wird die aktuelle Rekursionstiefe d übergeben. Wenn d ein zuvor festgelegtes Maximum d_{max} überschreitet, wird die Rekursion abgebrochen.

Wir starten den Algorithmus 1 mit dem Aufruf `midpoint(0, 1, 1)` und nachdem er terminiert ist, interpolieren wir die bis jetzt undefinierten Werte linear.

Es stellt sich nun die Frage, wie man den Zufallswert D_d in Abhängigkeit von der Rekursionstiefe berechnen soll. Damit eine fraktale Brownsche Bewegung entsteht, muss Bedingung (2) erhalten sein.

Es gibt also eine Konstante c mit

$$\text{var}(X(t_2) - X(t_1)) = c|t_2 - t_1|^{2H}. \quad (3)$$

Algorithm 1 Mittelpunktverschiebungs-Algorithmus; `midpoint`(l, r, d)

$$m := \frac{l+r}{2}$$

berechne Zufallswert D_d

$$X(m) := \frac{X(l) + X(r)}{2} + D_d$$

if $d \leq d_{max}$ **then**
 `midpoint`($l, m, d+1$)
 `midpoint`($m, r, d+1$)
end if

Wir nehmen weiters an, dass

$$X(0) = 0 \quad \text{und} \quad X(1) \sim N(0, \sigma^2).$$

Für $t = 1$ gilt nun

$$\text{var}(X(1) - X(0)) = \text{var}(X(1)) = \sigma^2.$$

Andererseits ist aber nach Gleichung (3)

$$\text{var}(X(1) - X(0)) = c|1 - 0|^{2H}.$$

Daraus folgt, dass $c = \sigma^2$ ist. Wir erhalten

$$\text{var}(X(t_2) - X(t_1)) = \sigma^2 |t_2 - t_1|^{2H}. \quad (4)$$

Das soll nun unsere Selbstähnlichkeitsbedingung für die fraktale Brownsche Bewegung sein. Es ist naheliegend, für die Zufallsvariable D_d eine Normalverteilung mit einer bestimmten Varianz δ_d^2 zu wählen.

$$D_d \sim N(0, \delta_d^2)$$

Im Folgenden analysieren wir die beiden ersten Schritte des Algorithmus und bestimmen dafür das δ_d .

Im ersten Schritt berechnen wir

$$X\left(\frac{1}{2}\right) = \frac{X(0) + X(1)}{2} + D_1.$$

Wir subtrahieren $X(0)$ von beiden Seiten.

$$\begin{aligned} X\left(\frac{1}{2}\right) - X(0) &= \frac{X(1) - X(0)}{2} + D_1 \\ \text{var}\left(X\left(\frac{1}{2}\right) - X(0)\right) &= \text{var}\left(\frac{X(1) - X(0)}{2} + D_1\right) \\ &= \left(\frac{1}{2}\right)^2 \text{var}(X(1) - X(0)) + \text{var}(D_1) \\ &= \left(\frac{1}{2}\right)^2 \sigma^2 + \delta_1^2 \end{aligned}$$

Andererseits soll aber nach Bedingung (4)

$$\text{var}\left(X\left(\frac{1}{2}\right) - X(0)\right) = \sigma^2 \left(\frac{1}{2}\right)^{2H}$$

sein. Damit erhalten wir

$$\begin{aligned} \delta_1^2 &= \sigma^2 \left(\frac{1}{2}\right)^{2H} - \left(\frac{1}{2}\right)^2 \sigma^2 = \sigma^2 \left(\frac{1}{2}\right)^{2H} \left(1 - \left(\frac{1}{2}\right)^{2-2H}\right) \\ &= \sigma^2 2^{-2H} (1 - 2^{2H-2}) \end{aligned}$$

wodurch die gesuchte Varianz δ_1^2 in Abhängigkeit von σ und dem Hurst-Exponenten H bestimmt ist.

Im zweiten Schritt erhalten wir analog zum ersten:

$$\begin{aligned} \text{var}\left(X\left(\frac{1}{4}\right) - X(0)\right) &= \text{var}\left(\frac{X\left(\frac{1}{2}\right) - X(0)}{2} + D_2\right) \\ &= \left(\frac{1}{2}\right)^2 \text{var}\left(X\left(\frac{1}{2}\right) - X(0)\right) + \text{var}(D_2). \end{aligned}$$

Nun soll aber gelten

$$\begin{aligned} \text{var}\left(X\left(\frac{1}{4}\right) - X(0)\right) &= \sigma^2 \left(\frac{1}{4}\right)^{2H} \quad \text{und} \\ \text{var}\left(X\left(\frac{1}{2}\right) - X(0)\right) &= \sigma^2 \left(\frac{1}{2}\right)^{2H}. \end{aligned}$$

Somit erhalten wir eine Gleichung, durch die δ_2 bestimmt ist.

$$\begin{aligned}\sigma^2 \left(\frac{1}{4}\right)^{2H} &= \left(\frac{1}{2}\right)^2 \sigma^2 \left(\frac{1}{2}\right)^{2H} + \delta_2^2 \\ \delta_2^2 &= \sigma^2 \left(\frac{1}{2}\right)^{2 \cdot 2H} - \left(\frac{1}{2}\right)^2 \sigma^2 \left(\frac{1}{2}\right)^{2H} \\ &= \sigma^2 \left(\frac{1}{2}\right)^{2 \cdot 2H} \left(1 - \left(\frac{1}{2}\right)^{2-2H}\right) \\ &= \sigma^2 2^{-2 \cdot 2H} (1 - 2^{2H-2})\end{aligned}$$

Unsere Vermutung für den n -ten Schritt lautet

$$\delta_n^2 = \sigma^2 2^{-n \cdot 2H} (1 - 2^{2H-2}). \quad (5)$$

Diesen Zusammenhang kann man zum Beispiel mit vollständiger Induktion beweisen.

Jetzt haben wir fast alles, was wir für die Erzeugung einer fraktalen Brownschen Bewegung mittels Mittelpunktverschiebungs-Algorithmus brauchen. Eine Kleinigkeit fehlt noch: Wir benötigen normalverteilte Zufallszahlen.

3.3 Erzeugung von normalverteilten Zufallszahlen

Im Computer haben wir einen Pseudozufallsgenerator, der gleichverteilte Werte liefert. Um daraus eine Normalverteilung realisieren zu können, bedienen wir uns des zentralen Grenzwertsatzes von Lindeberg-Lévy.

Seien X_1, X_2, X_3, \dots gleichverteilte, unabhängige Zufallsvariablen mit

$$X_i \sim G_{0,m},$$

also mit Dichtefunktion

$$f_{X_i}(x) = \begin{cases} \frac{1}{m} & \text{für } 0 \leq x \leq m, \\ 0 & \text{sonst.} \end{cases}$$

Wir bezeichnen den Erwartungswert der X_i ($i \geq 1$) mit μ und die Varianz mit σ^2 . Sei weiters

$$Y_k = X_1 + X_2 + \dots + X_k,$$

dann folgt zunächst

$$\begin{aligned}\mathbb{E}(Y_k) &= k \mu \\ \text{var}(Y_k) &= k \sigma^2\end{aligned}$$

für den Erwartungswert und die Varianz von Y_k , weil die X_i unabhängig sind. Wir normieren nun Y_k .

$$Y_k^* = \frac{Y_k - E(Y_k)}{\sqrt{\text{var}(Y_k)}} = \frac{\sum_{i=1}^k X_i - k\mu}{\sigma\sqrt{k}}$$

Nach dem zentralen Grenzwertsatz von Lindeberg-Lévy gilt nun für $k \rightarrow \infty$:

$$Y_k^* \longrightarrow Y \sim N(0, 1)$$

Von der Gleichverteilung wissen wir

$$\begin{aligned} E(X_i) &= \mu = \frac{m}{2} \quad \text{und} \\ \text{var}(X_i) &= \sigma^2 = \frac{m^2}{12}. \end{aligned}$$

Daraus erhalten wir die Vereinfachung

$$Y_k^* = \frac{\sum_{i=1}^k X_i - k\frac{m}{2}}{m\sqrt{\frac{k}{12}}} = a \sum_{i=1}^k X_i - b$$

mit

$$a = \frac{1}{m} \sqrt{\frac{12}{k}} \quad \text{und} \quad b = \sqrt{3k}.$$

Geschickterweise wählen wir $k = 12$ und erhalten dadurch $a = 1/m$ und $b = 6$. Das liefert uns eine ganz brauchbare Approximation für eine Normalverteilung. Wir können also für eine Implementierung eines Pseudozufallsgenerators mit normierter Normalverteilung die folgende Formel verwenden:

$$Y_k^* = \frac{1}{m} \sum_{i=1}^{12} X_i - 6$$

4 Fraktale Landschaften

Wir werden nun den Mittelpunktverschiebungs-Algorithmus in geeigneter Weise modifizieren, um fraktale Landschaften zu erzeugen.

In Analogie zum eindimensionalen Fall starten wir mit einem Quadrat der Seitenlänge 1 (Abb. 7).

Das Hinabsteigen von Rekursionstiefe $i - 1$ nach i geschieht hier in zwei Schritten:

1. Wir berechnen zunächst die Höhe des Mittelpunktes durch Interpolation der vier Nachbarpunkte plus einen Zufallswert $D_{M,i}$, den wir in ähnlicher Weise wie im eindimensionalen erzeugen.

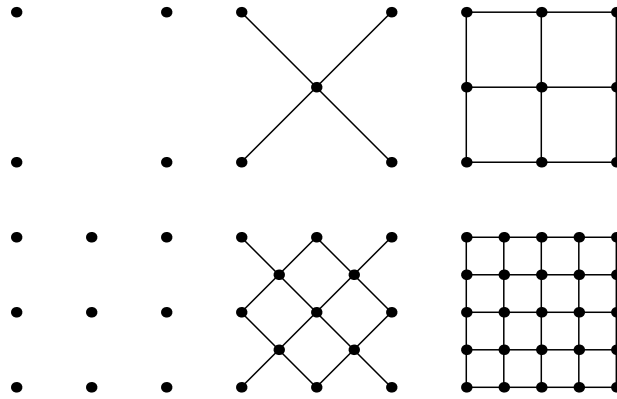


Abbildung 7: Erzeugung fraktaler Landschaften durch Mittelpunktverschiebung

- Wir berechnen die Höhe der verbleibenden, dazwischenliegenden vier Punkte p_1, p_2, p_3, p_4 .

Diese haben, sofern man sich nicht gerade am Rand des ursprünglichen Quadrats befindet, ebenfalls je vier Nachbarpunkte. Aus diesen interpolieren wir die Höhe für p_j und addieren einen Zufallswert $D_{j,i}$ ($1 \leq j \leq 4$).

Auf der xy -Ebene wird das ursprüngliche Quadrat am Mittelpunkt in vier Quadrate mit halber Seitenlänge aufgeteilt. Für jedes dieser Quadrate führen wir das Verfahren rekursiv fort.

Entscheidend für die Selbstähnlichkeit ist wiederum, dass die Zufallswerte $D_{M,i}$ und $D_{j,i}$ ($1 \leq j \leq 4$) normalverteilt sind mit Varianz δ_i^2 . Wir berechnen die Varianz δ_i^2 aus Gleichung 5. Sie nimmt also mit zunehmender Rekursionstiefe ab.

Achtung! Der Hurst-Exponent H unterscheidet sich hier vom eindimensionalen Fall, weil wir für die euklidische Dimension $E = 2$ ansetzen müssen. Also ist

$$H = E + 1 - D = 3 - D,$$

wobei D die fraktale Dimension bezeichnet.

5 Perlin Noise

In diesem Abschnitt stellen wir ein weiteres Verfahren vor, das in der Computergrafik vielseitige Anwendung erfährt. Perlin Noise wird zur Generierung von Landschaften, Oberflächentexturen (z.B. Holz, Marmor – siehe [1]) und

dergleichen verwendet, kann aber auch eingesetzt werden, um Computerzeichnungen aussehen zu lassen, als wären Sie von Menschenhand gefertigt (also leicht verwackelt und nicht exakt – siehe [2]).

Die breitgefächerte Anwendbarkeit ist auf das sehr allgemein gehaltene Konzept zurückzuführen, das sich auch leicht auf höhere Dimensionen (als die hier vorgestellte Dimension 1) ausweiten läßt.

5.1 Noise-Funktionen

Eine Noise-Funktion ist eine Abbildung, die jeder natürlichen Zahl einen Zufallswert (in bestimmten Schranken) zuordnet. Wir modifizieren diese ganz allgemeine Definition auf die Weise, dass wir das Einheitsintervall in 2^n Intervalle unterteilen und jedem der so entstandenen $2^n + 1$ Teilungspunkte einen zufälligen Wert zuordnen.

$$F_n := 2^n$$

nennen wir dabei die Frequenz der zugehörigen Noise-Funktion f_n . Bisher wurde noch nichts über den Wertebereich von f_n , also das Intervall, aus dem die Zufallszahlen stammen, gesagt. Wir bezeichnen die Länge dieses Intervalls als Amplitude A_n und wollen, dass die Amplitude für höhere Frequenzen immer kleiner wird.

Die Folge der A_n soll also für $n \rightarrow \infty$ gegen Null streben. Um ein qualitatives Maß dafür zu haben, wie schnell dieses „gegen-Null-Streben“ vonstatten geht, verwenden wir einen Parameter a , den wir Persistence nennen. Ist solch ein $a \in [0, 1)$ gegeben, dann setzen wir:

$$A_n := a^n$$

und haben damit eine mit Sicherheit konvergente geometrische Folge mit Grenzwert Null, wie gewünscht. Nun wissen wir alles über Wertebereich und Grundmenge der Noise-Funktion f_n .

$$f_n : \left\{ \frac{0}{2^n}, \frac{1}{2^n}, \dots, \frac{2^n}{2^n} \right\} \rightarrow \left[-\frac{a^n}{2}, \frac{a^n}{2} \right]$$

Mithilfe eines (Pseudo-)Zufallszahlengenerators können wir bereits Noise-Funktionen generieren, beispielsweise für $n = 2$ und $a = 0.8$ (d.h. wir erhalten fünf Punkte mit Abszissenwerten zwischen -0.32 und 0.32 – siehe Abb. 8, linkes Bild).

5.2 Interpolierte Noise-Funktion

Der folgende Schritt besteht darin, die Noise-Funktion zu interpolieren.

$$I_{f_n} : [0, 1] \rightarrow \mathbb{R}$$

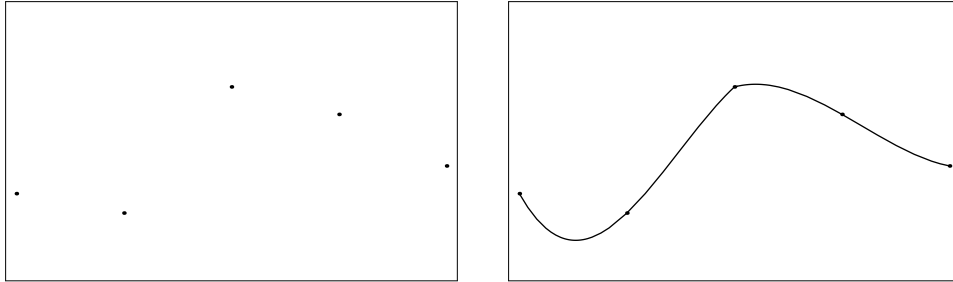


Abbildung 8: Interpolation der zufällig erzeugten Abszissen.

Grundsätzlich sind dazu sehr viele Verfahren möglich, angefangen bei linearer Interpolation über kubische Splines bis hin zum Interpolationspolynom. Letzteres brauchen wir von vornherein nicht unter Betracht zu ziehen, da der Grad des Interpolationspolynoms gerade der Frequenz entspricht und damit für hohe n extrem aufwendig wird.

Für welches Interpolationsverfahren man sich letztlich entscheidet, wird abhängig sein von der Applikation, der verfügbaren Rechenleistung, der geforderten Glätte des Endproduktes, etc.

Wir haben für die Abbildungen in dieser Arbeit die unter Mathematica standardmäßig eingestellte kubische Interpolation verwendet. Als Beispiel interpolieren wir die im vorhergehenden Abschnitt erzeugte Noise-Funktion mit $n = 2$ und $a = 0.8$ (siehe Abb. 8).

An diesem Punkt ließe sich eine Vielzahl von praktikablen Varianten bzw. Zusätzen angeben. So könnte man vor der Interpolation noch einen glättenden Mittelwertfilter über die Punktemenge lassen, oder man verzichtet gleich auf exakte Interpolation und wendet stattdessen ein Approximationsverfahren an, etc.

5.3 Das Endergebnis

Setzen wir die Bestandteile zusammen: eine Perlin Noise-Funktion f ist die Summe (über alle n) von interpolierten Noise-Funktionen I_{f_n} .

$$f = \sum_{n=1}^{\infty} I_{f_n}$$

Freilich ist die unendliche Summe nur Teil des theoretischen Konzepts, aber durch die beständige Abnahme der Amplituden der I_{f_n} läßt sich leicht irgendein N finden, sodass der Reihenrest $\sum_{n>N} I_{f_n}$ außerhalb des wahrnehmbaren Bereichs liegt (also beispielweise lediglich Änderungen im Subpixelbereich erwirkt). Die Größe von N hängt dabei stark vom Persistence-Wert ab!

Abb. 9 zeigt die ersten acht Bestandteile der in Abb. 10 zusammengesetzten Perlin Noise-Funktion, wobei für die Persistence $a = 0.6$ verwendet wurde.

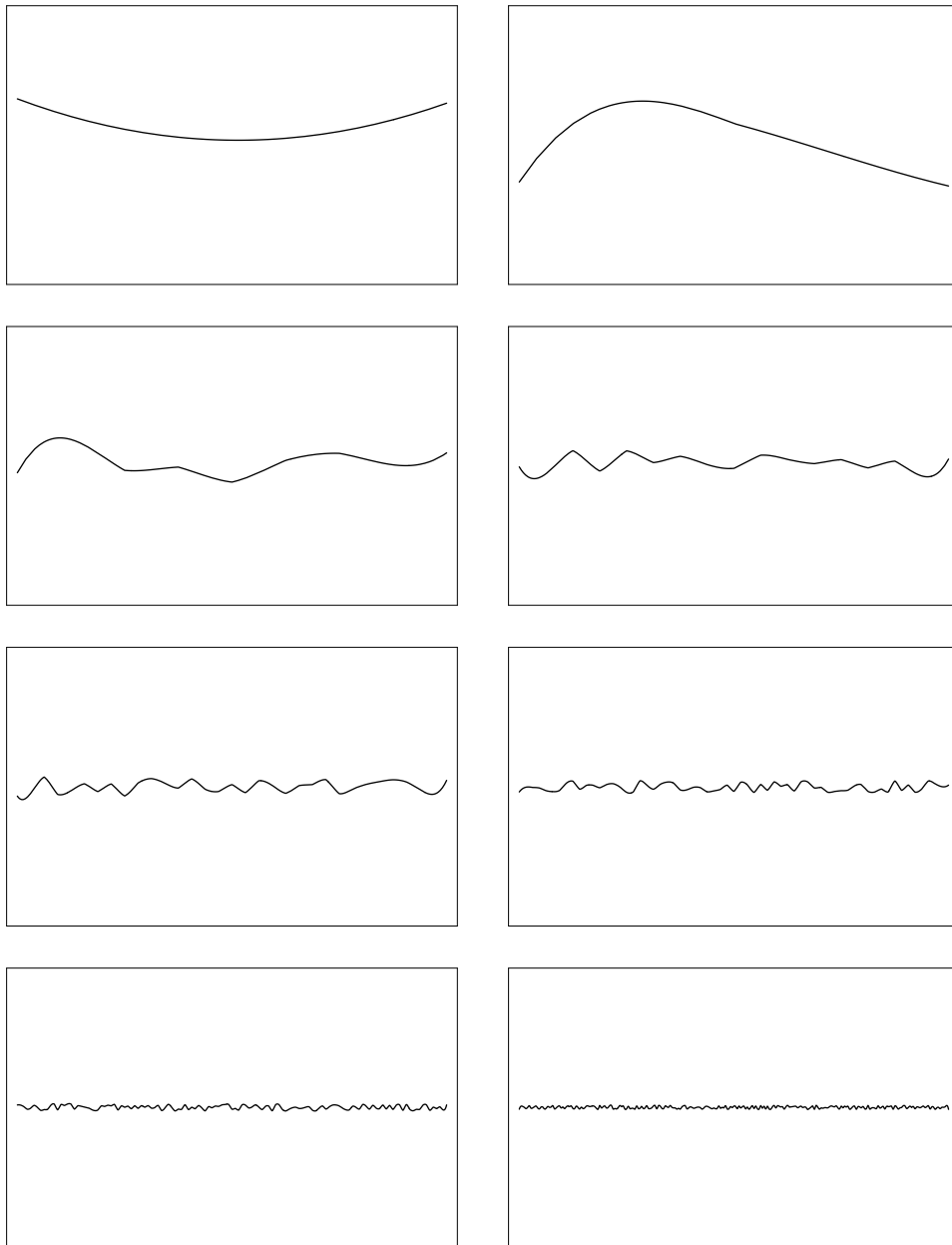


Abbildung 9: Acht interpolierte Noise-Funktionen addieren sich...

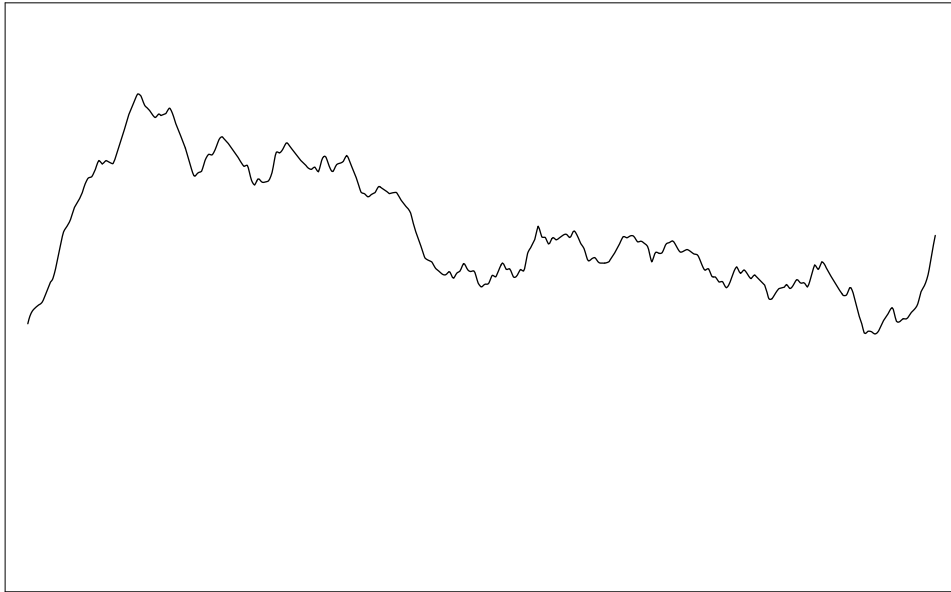


Abbildung 10: ... zu dieser Perlin Noise-Funktion. ($a = 0.6$)

Die Abb. 11 und 12 zeigen Ergebnisse, die mit größerem bzw. kleinerem Persistence-Wert erzielt wurden.

Perlin Noise gehört zur Obergruppe der fraktalen Brownschen Bewegung.

6 Fraktale Planetenoberflächen

Dieser Abschnitt stellt ein Verfahren vor, mit dem sich mehr oder weniger realistisch wirkende Ansichten ganzer Planeten erzeugen lassen. Der Beobachter hält sich also nicht auf der Oberfläche eines Planeten auf, um sich Landschaft, Vegetation etc. anzusehen, sondern er befindet sich im Welt-raum.

Ein Anwendungsfeld für Algorithmen dieser Art sind Science Fiction-Filme oder -Serien (wie etwa *Star Trek*), in denen Raumschiffe an Planeten vorüberfliegen, in deren Orbit kreisen etc.

Ausgangspunkt für das Verfahren ist eine Kugel mit einheitlichem Intensitätswert (die Graustufe eines Punktes auf der Kugeloberfläche kann später als Höhe oder als Index einer Farbpalette interpretiert werden).

Wir wählen einen Vektor zufällig aus und sehen ihn als Normalvektor einer Ebene durch den Ursprung an. Nun erhöhen wir die Intensitätswerte aller Punkte der Kugeloberfläche auf der einen Seite dieser Ebene und verringern die der anderen (um einen kleinen Betrag). Dieses Verfahren wiederholen wir sooft, bis die Linienstruktur auf der Oberfläche der Kugel verschwunden ist.

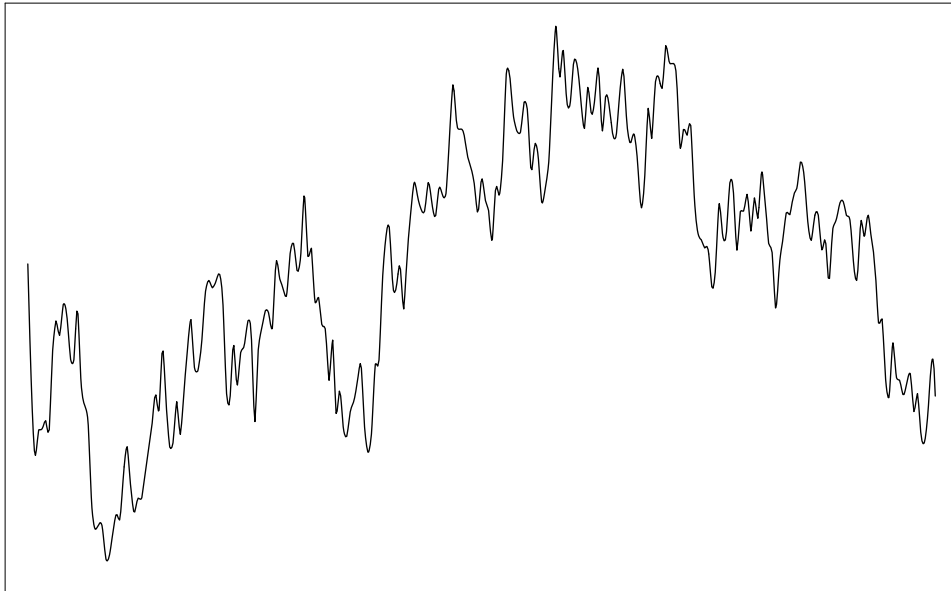


Abbildung 11: Ein höherer Wert für Persistence äußert sich in stärkerer „Zerklüftung“ der Perlin Noise-Funktion. ($a = 0.8$)

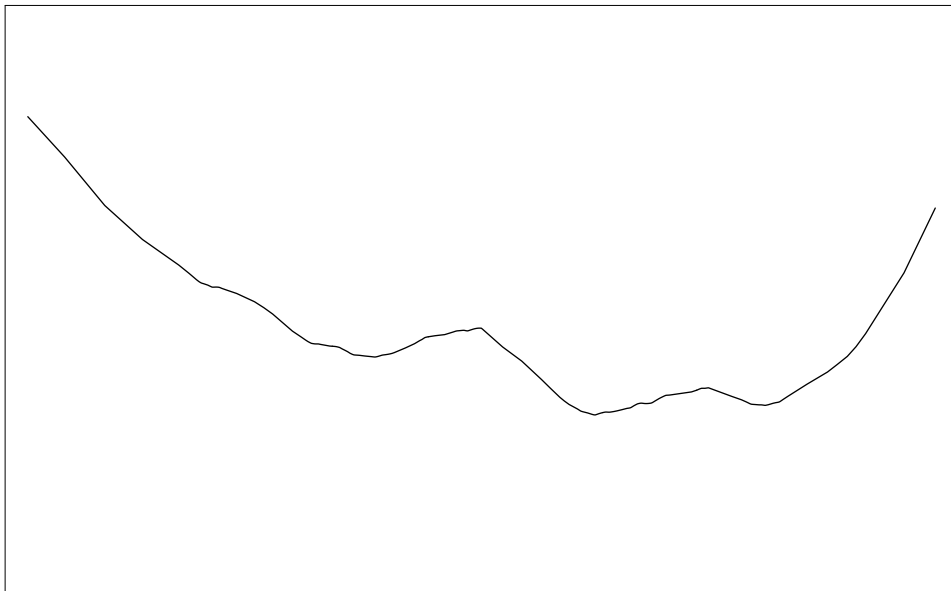


Abbildung 12: Dementsprechend liefert ein geringerer Wert Formen, die durchaus dem Querschnitt einer Landschaft gleichen. ($a = 0.4$)

Wir bemerken, dass eine Approximation der Kugeloberfläche mittels Polygonen zu keinerlei Erschwernis des Verfahrens führt. Die Überprüfung, ob ein Punkt p auf jener Seite der Ebene liegt, in die der zufällig gewählte Normalvektor n zeigt, reduziert sich auf die Abfrage

$$p \cdot n > 0$$

Alle Punkte, für die das nicht zutrifft, liegen auf der anderen Seite (bzw. im Schnittkreis der Kugeloberfläche mit der Ebene).

Die beiden nachfolgenden Seiten dienen der Illustration des Verfahrens. Die Abbildungen sind [3] entnommen und verändert.

Literatur

- [1] http://freespace.virgin.net/hugo.elias/models/m_perlin.htm.
- [2] <http://mrl.nyu.edu/~meyer/projects/etchapad/>.
- [3] <http://astronomy.swin.edu.au/~pbourke/terrain/planets/>.
- [4] Peitgen Heinz-Otto, Jürgens Hartmut, and Saupe Dietmar. *Chaos and Fractals*. Springer, 1992.
- [5] Scholl Reinhard and Olaf Pfeiffer. *Natur als Fraktale Grafik*. Markt und Technik, 1991.

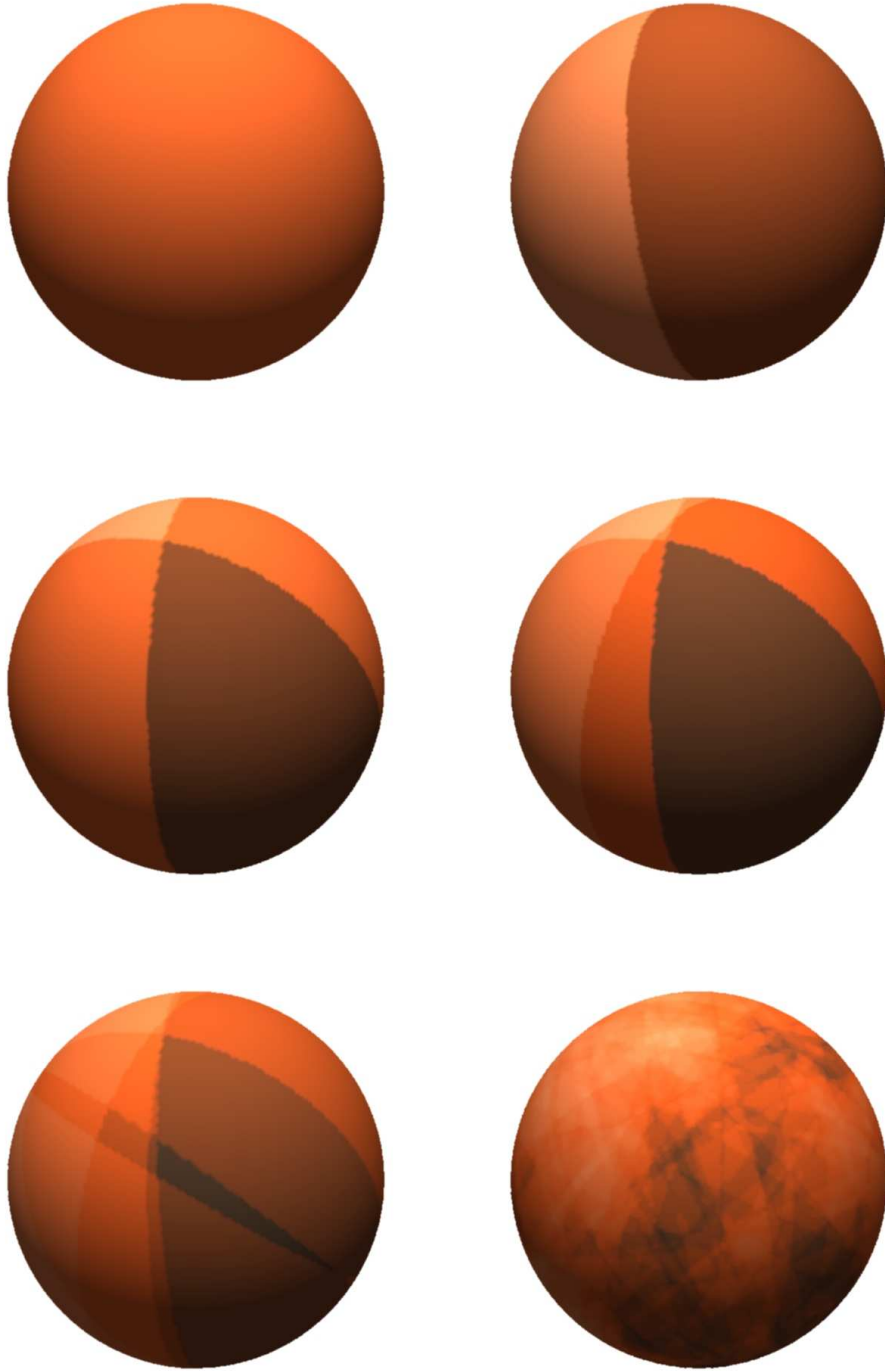


Abbildung 13: Die Genese eines fiktiven Planeten.



Abbildung 14: Die Oberfläche nach der 1000. Iteration.