

Fraktale

Thomas Peters
Thomas' Mathe-Seiten
www.mathe-seiten.de

12. September 2004

Mit diesem Artikel werden wir in das weite Feld der Fraktale einsteigen. Er beginnt mit den nötigen Bemerkungen zum Dimensionsbegriff. Dann werden Grundlagen über die Iteration von Funktionen besprochen. An dieses Kapitel schließt sich direkt eine Behandlung der logistischen Gleichung an. Nun wird zu den komplexen Zahlen übergegangen, d. h. Julia- und Mandelbrot-Mengen werden untersucht. Dann folgen eine Reihe weiterer Kapitel zur Vertiefung. Ein weiterer wichtiger Punkt ist das Kapitel über IFS. Darauf folgt eine ausführliche Behandlung diverser fraktaler Kurven und ihrer Darstellung als Lindenmayer-System.

Inhaltsverzeichnis

Abbildungsverzeichnis	5
1 Eine kurze Geschichte der Dimension	8
1.1 Die Euklidische Dimension	8
1.2 Die Topologische Dimension	9
1.3 Die Fraktale Dimension	9
1.4 Die Ähnlichkeitsdimension	10
2 Fixpunkte	11
3 Wachstumsprozesse	13
3.1 Exponentielles Wachstum	13
3.2 Logistisches Wachstum	13
4 Die logistische Gleichung	15
4.1 Periodenverdopplung	16
4.2 Bifurkations-Diagramme	19
4.3 Ausblicke	19
4.4 Aufgaben	21
5 Julia-Mengen	23
6 Die Mandelbrot-Menge	28
6.1 Die Struktur der Mandelbrot-Menge	28
6.2 Eine fraktale Reise	31
6.3 π und die Mandelbrot-Menge	39
6.4 Verallgemeinerungen	40
7 Quaternionen	42
8 Eine kurze Geschichte der Farbe I	45
8.1 Die Iterationszahl	46
8.2 Dekomposition	46
8.3 Biomorph	47
8.4 Diverse Verfahren (äußere Punkte)	47

8.5	Diverse Verfahren (innere Punkte)	48
8.6	3D-Darstellung	48
8.7	Die Lake-Transformation	50
8.8	Filter	50
9	Das Curlicue-Fraktal	52
10	Das Newton-Fraktal	54
10.1	Reelle Funktionen	54
10.2	Komplexe Funktionen	55
11	Iterierte Funktionssysteme	59
12	Fraktale Kurven	66
12.1	Der Cantor-Staub	67
12.2	Die Koch-Kurve	68
12.3	Das Sierpinski-Dreieck	70
12.4	Die Drachenkurve	73
12.5	Die Peano-Kurve	75
13	Lindenmayer-Systeme	76
13.1	Das Fibonacci-L-System	76
13.2	Das Thue-Morse-L-System	77
13.3	Graphische Interpretation	78
14	Zellularautomaten	82
14.1	Binäre eindimensionale Automaten	82
14.2	Binäre zweidimensionale Automaten	83
14.3	Die Ameisensimulation	83
15	Eine kurze Geschichte der Farbe II	87
15.1	Mandelcloud	87
15.2	Das Newton-Fraktal	87
15.3	Iterierte Funktionssysteme	87
15.4	Lindenmayer-Systeme	89
	Index	90

Abbildungsverzeichnis

2.1	Einfache Spinnwebdiagramme.	12
3.1	Das logistische Wachstum für infinitesimale Zeitintervalle.	14
4.1	Spinnwebdiagramme für verschiedene r	16
4.2	Spinnwebdiagramm und 2. Iterierte.	17
4.3	Periode 8.	18
4.4	Chaos.	18
4.5	Bifurkationsdiagramm von $r = 0$ bis $r = 4$	20
4.6	Bifurkationsdiagramm von $r = 3$ bis $r = 4$	20
4.7	Periodisches Fenster.	21
4.8	Voriterationen bei Zufallszahl.	22
4.9	Voriterationen bei festgehaltenem Startwert.	22
4.10	Bifurkationsdiagramm der Gleichung $x_{n+1} = r \sin(\pi x_n)$	22
5.1	Julia-Menge für $c = -1$	24
5.2	Julia-Menge für $c = -0,1 + 0,7i$	24
5.3	Julia-Menge für $c = -0,1 - 0,7i$	25
5.4	Julia-Menge für $c = 0,360284 + 0,1i$	25
5.5	Julia-Menge für $c = i$	26
5.6	Julia-Menge für $c = 0,11301 + 0,67037i$	26
5.7	Julia-Menge für $c = -0,1011 + 0,9563i$	26
6.1	Die Mandelbrot-Menge.	29
6.2	Mandelbrotmenge und Bifurkationsdiagramm.	29
6.3	Äquipotentialkurven der Mandelbrotmenge.	30
6.4	Rumpf und Kopf der Mandelbrot-Menge.	32
6.5	Ausschnitt aus der Mandelbrot-Menge.	32
6.6	Entsprechende Julia-Menge.	33
6.7	$(-2,237251; 1,324524)$ $(1,110561; -1,186334)$	33
6.8	$(-0,9875; -0,025)$ $(-0,5875; -0,325)$	34
6.9	$(-0,9; -0,11625)$ $(-0,67; -0,28875)$	34
6.10	$(-0,841374; -0,226834)$ $(-0,825733; -0,238565)$	35
6.11	$(-0,835655; -0,229743)$ $(-0,832331; -0,232236)$	35
6.12	Mini-Mandelbrot-Menge.	36

6.13	„Beule“ am Rand der Mandelbrot-Menge.	36
6.14	Mini-Mandelbrot-Menge in der Nähe einer „Beule“.	37
6.15	Der obere Rand des „Rumpfes“.	37
6.16	Filamente.	38
6.17	Mini-Mandelbrot-Menge.	38
6.18	Mandelbrot-Menge zu $z_{n+1} = z_n^3 + c$	41
6.19	Mandelbrot-Menge zu $z_{n+1} = z_n^4 + c$	41
7.1	Quaternionen-Julia-Menge von $(-0,456426,0,510553,0,1224,-0,0924)$	44
7.2	Quaternionen-Julia-Menge von $(0,5857,0,4385,0,0)$	44
8.1	Iterationen: 100, 250, 500, 1000.	46
8.2	Färben nach der Iterationszahl.	46
8.3	Dekomposition.	47
8.4	Biomorph.	47
8.5	Diverse Verfahren.	48
8.6	Gefärbt nach $\arg(z)$	49
8.7	3D-Darstellung der Mandelbrot-Menge.	49
8.8	Dekomposition mit Lake-Transformation.	50
8.9	Äquipotentiallinien der Mandelbrot-Menge.	51
9.1	Der Goldene Schnitt.	52
9.2	Die Euler-Mascheroni-Konstante.	53
9.3	π	53
10.1	Beispiel zum reellen Newton-Verfahren.	55
10.2	Newton-Fraktal für $n = 3$	56
10.3	Newton-Fraktal für $n = 3$ (Ausschnitt).	56
10.4	Newton-Fraktal für $n = 7$	57
10.5	Newton-Fraktal für $n = 7$ (Ausschnitt).	57
10.6	Newton-Fraktal für $f(z) = z^5 - z^2 + 0,1$	58
11.1	Das Sierpinski-Dreieck.	60
11.2	Die ersten vier Iterationen des Sierpinski-IFS.	60
11.3	Die affinen Abbildungen des IFS-Farns.	62
11.4	Der IFS-Farn.	63
11.5	Ein IFS-Blatt.	64
11.6	Ein IFS-Baum.	64
11.7	Noch ein IFS-Baum.	65
12.1	Der Cantor-Staub.	67
12.2	Die Teufelstreppe.	68
12.3	Der Cantor-Vorhang.	69
12.4	Die Koch-Kurve.	69
12.5	Die Schneeflockenkurve.	70

12.6 Die Entstehung des Sierpinski-Dreiecks.	70
12.7 Das Sierpinski-Dreieck als Julia-Menge.	72
12.8 Die Sierpinski-Pyramide.	72
12.9 Der Menger-Teppich.	73
12.10 Der Menger-Schwamm.	73
12.11 Die Entwicklung der Drachenkurve.	74
12.12 Die Drachenkurve.	74
12.13 Die Peano-Kurve.	75
13.1 Die Evolution einer botanischen Form.	79
13.2 Das fertige Bild.	80
13.3 $F \mapsto FF - [-F + F + F] + [+F - F - F]$	80
13.4 $F \mapsto F[+FF][-FF]F[+FF][-FF]F$	81
14.1 Binärer Zellularautomat.	83
14.2 Noch ein binärer Zellularautomat.	83
14.3 Läufer im Game of Life.	84
14.4 2 Ameisen, Regel 10010011.	84
14.5 10 Ameisen, Regel 111000.	85
14.6 10 Ameisen, Regel 110.	85
14.7 10 Ameisen, Regel 111100000.	86
15.1 Die Mandelbrot-Menge als Mandelcloud.	88
15.2 Einzelne Orbits durch Mandelcloud.	88
15.3 Attraktor & Iterationen mod 2 und Iterationen.	88
15.4 Das Newton-Fraktal in 3D.	89
15.5 Die affinen Abbildungen des Farns.	89

1 Eine kurze Geschichte der Dimension

Es gibt unzählige (verschiedene) Definitionen der Dimension. In diesem Kapitel werden die wichtigsten Definitionen erläutert und Beziehungen zwischen ihnen hergestellt. Außerdem wird auf Probleme hingewiesen, die sich beim Studium fraktaler Kurven ergeben.

1.1 Die Euklidische Dimension

Der intuitivste Zugang zum Dimensionsbegriff besteht wohl darin, sie als die minimale Anzahl von Parametern zu definieren, die benötigt wird, um ein geometrisches Objekt vollständig zu beschreiben. Bei einer Geraden benötigt man einen Parameter (die x -Koordinate), sie hat die euklidische Dimension $D_E = 1$. Für eine Ebene braucht man zwei Parameter, z. B. x - und y -Koordinate oder Polarkoordinaten r, θ , also ist sie zweidimensional.

Diese Vorstellung wird von der Linearen Algebra unterstützt, denn dort ist die Dimension eines Vektorraums gleich der Anzahl der Basisvektoren, was genau der minimalen Anzahl der Koordinatenachsen entspricht.

Es gibt nun zwei Probleme, die diese Dimensionsvorstellung beinhaltet. Das erste Problem ist die eineindeutige Abbildung der Punkte einer Geraden auf die Punkte einer Ebene, die von Cantor gefunden wurde. Das widerspricht völlig der Vorstellung, dass eine Ebene mehr Punkte enthalten muss als eine Gerade. Der Beweis ist eine Abwandlung von Cantors Diagonalverfahren.

Wie beschränken uns wieder auf das Intervall $[0; 1]$ bzw. $[0; 1]^2$. Dann können Punkte des Einheitsquadrats mit den Koordinaten

$$(0, a_1 a_2 a_3 \dots; 0, b_1 b_2 b_3 \dots)$$

eineindeutig auf einen Punkt der Einheitsstrecke mit den Koordinaten

$$0, a_1 b_1 a_2 b_2 a_3 b_3 \dots$$

abgebildet werden. Es existiert also eine umkehrbare Abbildung

$$f: \mathbb{R} \longrightarrow \mathbb{R}^2.$$

Wichtig ist, zu erkennen, dass eine eineindeutige Transformation die Dimension ändern kann!

Diese Transformation ist jedoch nicht stetig. Peano hat jedoch eine Transformation gefunden, die eine Strecke stetig auf eine Fläche abbildet, und zwar ebenfalls eineindeutig. Diese *flächenfüllende Kurve* wird ausführlich im Kapitel über fraktale Kurven besprochen. Das Problem ist nun, dass die Dimension nicht mehr die minimale Anzahl der Parameter sein kann!

1.2 Die Topologische Dimension

Die *topologische Dimension* wählt einen rekursiven Zugang zur Bestimmung der Dimension.

Die leere Menge soll die Dimension -1 haben. Dann hat ein Objekt die Dimension $n + 1$, wenn man jeden beliebigen Teil dieses Objekts mit Objekten der Dimension n vom Rest trennen kann und n die kleinste solche Dimension ist.

- Um eine abzählbare Menge (die kein Kontinuum ist) zu separieren, benötigt man nichts. Die Dimension ist also $D_T = -1 + 1 = 0$.
- Eine Linie kann durch einen Punkt unterteilt werden. Also ist ihre Dimension $D_T = 0 + 1 = 1$.
- Eine Ebene wird durch eine Gerade zerteilt: $D_T = 1 + 1 = 2$.
- Ein Würfel kann schließlich durch eine Ebene zerschnitten werden: $D_T = 2 + 1 = 3$.

Bei den Standardfiguren ist die Bestimmung der topologischen Dimension kein Problem. Schwieriger sieht es beispielsweise mit einem „ö“-förmigen Objekt (der Kreis des „ö“ sei gefüllt) aus. Hier muss man zwischen *globaler* und *lokaler* Dimension unterscheiden.

Denn betrachtet man das „ö“ als Ganzes, so benötigt man nichts, um es zu separieren — seine topologische Dimension wäre 0. Um das der Anschauung widersprechende Ergebnis zu korrigieren, führt man die lokale Dimension ein. Man betrachtet die „ö“-Punkte getrennt vom „ö“-Kreis. Die Punkte haben die Dimension 0. Um die Kreisscheibe zu unterteilen, benötigt man jedoch eine Gerade, daher ist ihre lokale Dimension 2. Damit das gesamte „ö“ die topologische Dimension 2 erhält, definieren wir sie als das Maximum der lokalen Dimensionen.

1.3 Die Fraktale Dimension

Synonyme für *fraktale Dimension* sind *Box-Counting-Dimension*, *Kapazitätsdimension* und *Hausdorff(-Besicovitch)-Dimension*.

Die Bestimmung der fraktalen Dimension erfolgt folgendermaßen: Man bedeckt das Objekt mit Kreisscheiben vom Durchmesser ε . Die Anzahl der Kreisscheiben sei $N(\varepsilon)$. Dabei ist das Objekt stets so zu bedecken, dass $N(\varepsilon)$ minimal wird (man könnte sagen, die dichteste Variante). Dann ist die fraktale Dimension

$$D_H = - \lim_{\varepsilon \rightarrow 0} \log_{\varepsilon} N(\varepsilon),$$

falls der Grenzwert existiert. Der Index H soll an Hausdorff-Dimension erinnern, welcher der gebräuchlichste Name für die fraktale Dimension ist.

Als Beispiel berechnen wir die Hausdorff-Dimension der Einheitsstrecke. Diese kann von einer Kreisscheibe des Durchmessers 1 bedeckt werden, von 2 Scheiben mit Durchmesser $1/2$, 3 Scheiben mit Durchmesser $1/3$ usw. Wir erhalten also:

$$N(\varepsilon) \cdot \varepsilon = 1 \Rightarrow N(\varepsilon) = \frac{1}{\varepsilon}$$

Daraus folgt: $D_H = 1$. Die Hausdorff-Dimension stimmt also für eine Strecke mit der topologischen Dimension überein.

Bei der Untersuchung fraktaler Kurven fällt auf, dass die Hausdorff-Dimension auch eine gebrochene Zahl sein kann. So liegt z. B. die Dimension des Cantor-Staubs zwischen Punkt und Gerade, die der Koch-Kurve zwischen Gerade und Fläche. Insbesondere heißt eine Kurve dann *fraktal*, wenn ihre Hausdorff-Dimension die topologische Dimension übersteigt (in den genannten Fällen 0 bzw. 1). Ein Extrembeispiel ist die schon erwähnte Peano-Kurve, welche die Hausdorff-Dimension 2 besitzt!

1.4 Die Ähnlichkeitsdimension

Die *Ähnlichkeitsdimension* kann als ein Spezialfall der Hausdorff-Dimension angesehen werden. Sie ist bei *selbstähnlichen* Objekten von Nutzen, d. h. wenn Teile des Objekts aussehen wie das Objekt als Ganzes.

Die Hausdorff-Dimension kann für beliebige Objekte bestimmt werden, allerdings muss der Grenzwert streng ermittelt werden, was u.U. sehr schwierig ist. Das ist bei selbstähnlichen Objekten kein Problem, da der Ausdruck $\log_\varepsilon N(\varepsilon)$ konstant ist. d. h., es genügt die Betrachtung einer Größenordnung, um den Grenzwert ermitteln zu können. Beispiele hierzu werden im Kapitel über fraktale Kurven angeführt.

2 Fixpunkte

Gegeben sei eine reelle Zahl x und eine Funktion f , die *iterativ* auf x angewendet werden soll. Die n -te Iterierte von x bezeichnet man dann als $f^n(x)$. $f^2(x)$ ist also gleich $f(f(x))$.

Die Folge $x, f(x), f(f(x)), \dots$ wird als *Orbit* von x bezeichnet. Der Orbit von x kann verschiedene Formen annehmen, die an (trivialen) Beispielen erläutert werden sollen.

- *Fixpunkt*

Eine Zahl x heißt Fixpunkt, wenn der Orbit von x konstant ist. Beispiel: Die Funktion $f(x) = x^2$ hat genau 2 Fixpunkte, nämlich $x = 0$ und $x = 1$. In diesem Beispiel heißt $x = -1$ *Vorfixpunkt*, da er nach einer Iteration in einen Fixpunkt übergeht.

Die Fixpunkte einer Funktion sind stets Lösungen der Gleichung $f(x) = x$.

- *Zyklische Wiederholung*

Es kann vorkommen, dass ein Orbit zyklisch zwischen mehreren Punkten hin- und her-springt. Den kleinsten solchen Zyklus nennt man die *Periode* des Orbits. Beispiel: Der Orbit von $x = -1$ ist zyklisch mit der Periode 2 unter der Funktion $f(x) = x^2 - 1$.

- *Chaotisches Verhalten*

Zeigt der Orbit keinerlei Muster und zeitliche Vorhersagbarkeit, so heißt er chaotisch. Chaos ist streng von *Zufall* zu unterscheiden, da letzterer *stochastischer* Natur ist. Ein chaotisches System jedoch, gefüttert mit denselben Daten, erzeugt stets wieder exakt dasselbe chaotische Verhalten. Das Chaos ist also reproduzierbar. Um diese Eigenschaft zu unterstreichen, spricht man auch von *deterministischem* Chaos.

Ein nützliches Hilfsmittel zur Untersuchung von Orbits ist die *Graphische Iteration*. Dabei plottet man zunächst die Funktion über x . Dann geht man vom *Startwert* x_0 vertikal nach oben, bis zum Funktionsgraphen. Von dort geht man horizontal bis zur Winkelhalbierenden und erhält $f(x_0) = x_1$. Als nächstes sucht man wieder vertikal den Funktionsgraphen usw. Die so entstehenden Diagramme nennt man *Spinnwebdiagramme*.

Um mit diesem Verfahren vertraut zu werden, untersuchen wir die einfache lineare Funktion $f(x) = a \cdot x + b$ und unterscheiden vier Fälle:

1. $a > 1$
2. $0 < a < 1$
3. $-1 < a < 0$
4. $a < -1$

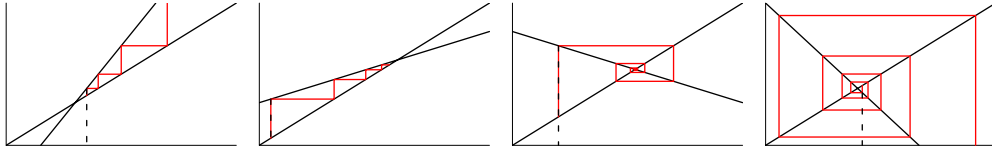


Abbildung 2.1: Einfache Spinnwebdiagramme.

Die entsprechenden Diagramme sind in Abbildung 2.1 dargestellt.

Wie man sieht, hat jede dieser Funktionen einen Fixpunkt.

1. $a > 1$: Der Orbit verläuft in dem Bereich zwischen Funktion und Winkelhalbierenden, und zwar auswärts. Den Fixpunkt nennt man *abstoßend* oder *instabil*; er heißt *Repeller*.
2. $0 < a < 1$: Der Orbit verläuft in dem Bereich zwischen Funktion und Winkelhalbierenden, und zwar einwärts. Den Fixpunkt nennt man *anziehend* oder *stabil*; er heißt *Attraktor*.
3. $-1 < a < 0$: Der Orbit verläuft spiralförmig um den Fixpunkt, und zwar einwärts. Den Fixpunkt nennt man *anziehend* oder *stabil*; er heißt *Attraktor*.
4. $a < -1$: Der Orbit verläuft spiralförmig um den Fixpunkt, und zwar auswärts. Den Fixpunkt nennt man *abstoßend* oder *instabil*; er heißt *Repeller*.

Im Falle einer nicht-linearen Funktion tritt $f'(z)$ (z sei ein Fixpunkt von f) an die Stelle von a . Dann gibt es noch zwei Sonderfälle, die bisher keinen Sinn machten:

- Ist $|f'(z)| = 1$, so heißt z *indifferent*.
- Ist $|f'(z)| = 0$, so heißt z *superstabil*.

Der Fixpunkt der oben betrachteten Fälle heißt zusammengefasst *hyperbolisch*. Es gilt also:

- $|f'(z)| > 1 \iff z$ ist Repeller.
- $|f'(z)| < 1 \iff z$ ist Attraktor.

Bei einer nicht-monotonen Funktion mit mehreren Fixpunkten ist zu beachten, dass die Aussagen über das Verhalten des Orbits jeweils nur für ein bestimmtes *Intervall* gelten.

3 Wachstumsprozesse

3.1 Exponentielles Wachstum

Das einfachste Modell des *Wachstumsverhaltens* einer Population besteht wohl darin, anzunehmen, dass die Zunahme der *Populationsgröße* in einem bestimmten Zeitintervall direkt proportional zur Populationsgröße im vorhergehenden Zeitintervall ist. Ist x_n die Population in einem solchen Zeitintervall, dann erhält man aus dieser Überlegung die *Differenzgleichung*

$$x_{n+1} - x_n = r \cdot x_n$$

mit dem *Proportionalitätsfaktor* r . Ist $r > 0$, so heißt r *Wachstumskonstante*; für $r < 0$ heißt r *Zerfallskonstante*. Diese Gleichung lässt sich umformen zu

$$x_{n+1} = (1 + r) \cdot x_n.$$

Dies ist die Gleichung des *exponentiellen Wachstums*. Man erhält nämlich leicht als *explizite Formel* für das n -te Zeitintervall

$$x_n = (1 + r)^n \cdot x_0 = x_0 \cdot e^{n \ln(1+r)},$$

wobei x_0 die Anfangspopulation ist. Dies ist jedoch ein sehr primitives Modell, da in Wirklichkeit aufgrund der Ressourcenbegrenztheit unbeschränktes Wachstum nicht möglich ist.

3.2 Logistisches Wachstum

Realistischer wird es schon, wenn wir auch eine *Todesrate* s berücksichtigen. Man könnte sie auch proportional zur bestehenden Populationsgröße ansetzen. Unter bestimmten Umständen ist aber die Todesrate sogar proportional zum Quadrat der Populationsgröße. So erhält man die Gleichung

$$x_{n+1} = x_n + r \cdot x_n - s \cdot x_n^2.$$

Dieses Wachstum heißt *Verhulst-Wachstum* oder *logistisches Wachstum*. Die explizite Form dieses Wachstums zu ermitteln ist schwieriger. Sie gilt auch nur für infinitesimal kleine Zeitintervalle und lautet

$$x_n = \frac{rx_0}{sx_0 + (r - sx_0)e^{-rn}}.$$

Wie man sieht, strebt dieses Wachstum einem Grenzfalle entgegen, und zwar ist die Schlusspopulation gegeben durch r/s . Das bedeutet also: Die Schlusspopulation ist unabhängig von der Anfangspopulation!

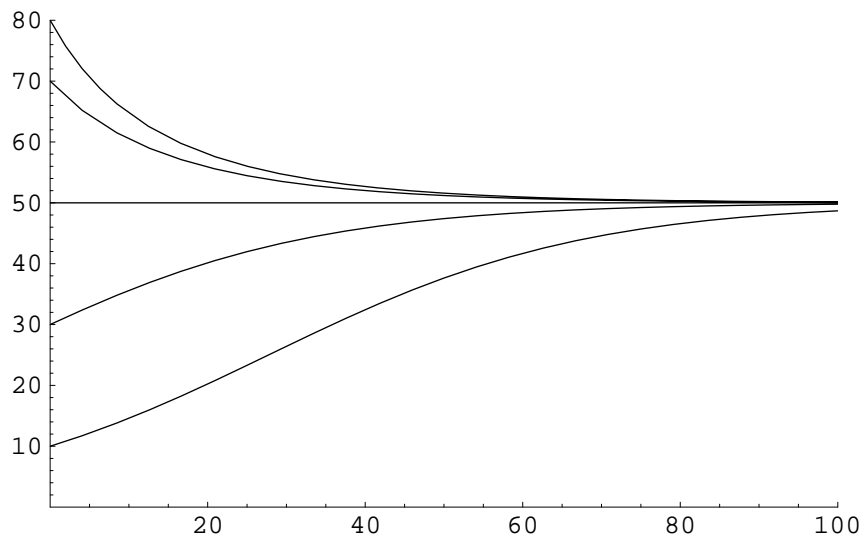


Abbildung 3.1: Das logistische Wachstum für infinitesimale Zeitintervalle.

Abbildung 3.1 zeigt das logistische Wachstum mit den Parametern $r = 0,05$ und $s = 0,001$ für die Startwerte 10, 30, 50, 70 und 80. 50 ist sogleich die Schlusspopulation.

Das Erstaunliche ist nun, dass sich die Differenzengleichung ganz anders verhält, als die explizite Formel vermuten lässt. Während letztere für infinitesimal kleine Zeitintervalle stetiges Wachstum (oder stetigen Zerfall) voraussagt, lässt die Differenzengleichung mehr Verhaltensvariationen zu. Das soll im folgenden Kapitel näher untersucht werden.

4 Die logistische Gleichung

Die logistische Gleichung ist wohl eine der meist untersuchten Gleichungen der gesamten Mathematik. Sie kann als Spezialfall des logistischen Wachstums angesehen werden, und zwar für den Fall $r = s$ und lautet

$$x_{n+1} = r \cdot x_n \cdot (1 - x_n).$$

Anschaulich sieht man an dieser Gleichung, dass die Population sowohl proportional zur gegenwärtigen Population als auch zum verbleibenden Lebensraum ist.

Betrachtet man den rechten Teil der Gleichung als Funktion von x_n , so hat diese offensichtlich Nullstellen bei $x_n = 0$ und $x_n = 1$ sowie einen Extrempunkt bei $(0,5 \mid 0,25 r)$.

Wir werden diese Funktion benutzen, um das *Einheitsintervall* $[0; 1]$ auf sich selbst abzubilden. Aus diesem Grund muss $0 \leq r \leq 4$ gelten.

Nun sollen die Fixpunkte der Funktion $f(x) = r \cdot x \cdot (1 - x)$ bestimmt werden.

$$\begin{aligned} & x = r \cdot x \cdot (1 - x) \\ \Leftrightarrow & x \cdot (r \cdot (1 - x) - 1) = 0 \\ \Leftrightarrow & x \cdot (r - 1 - r \cdot x) = 0 \\ \Leftrightarrow & x = 0 \quad \vee \quad x = 1 - \frac{1}{r} \end{aligned}$$

Für $r \neq 0$ erhalten wir also zwei Fixpunkte. Um herauszufinden, welcher Art diese Fixpunkte sind, bilden wir die Ableitung der Funktion:

$$\begin{aligned} f'(x) &= r - 2 \cdot r \cdot x \\ f'(0) &= r \\ f'\left(1 - \frac{1}{r}\right) &= 2 - r \end{aligned}$$

Daraus erhält man, dass $x = 0$ für $r < 1$ anziehender Fixpunkt ist, für $r > 1$ ist er Repeller. Dagegen ist $x = 1 - 1/r$ für $1 < r < 3$ Attraktor und für $r < 1$ und $r > 3$ instabil.

Für diese Fälle zeigt Abbildung 4.1 zur Verdeutlichung die entsprechenden Spinnwebdiagramme.

Wie erwartet konvergiert der erste Orbit gegen 0, wogegen der zweite gegen 0,5 konvergiert. Der dritte Orbit ist interessant: Beide Fixpunkte sind Repeller, aber der Orbit divergiert nicht gegen unendlich. Vielmehr hat sich ein 2-er Zyklus gebildet. Die Entstehung solcher Zyklen wird uns im Weiteren näher beschäftigen.

Wichtig ist, dass das Verhalten des Orbits unabhängig vom Startwert, sondern nur abhängig von r ist!

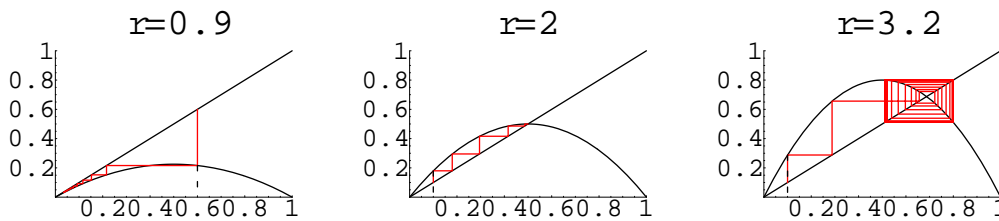


Abbildung 4.1: Spinnwebdiagramme für verschiedene r .

4.1 Periodenverdopplung

Wir haben gesehen, dass sich für bestimmte Parameterwerte 2-er Zyklen bilden, d. h. der Orbit konvergiert abwechselnd gegen zwei verschiedene Werte. Sind x_1 und x_2 zwei solche Zahlen, so muss $f(x_1) = x_2$ und $f(x_2) = x_1$ gelten. Oder anders formuliert: $f^2(x_1) = x_1$ und $f^2(x_2) = x_2$. Die Zahlen mit Periode 2 sind also die Fixpunkte der zweiten Iterierten! Oder allgemein:

$$\hat{x} \text{ Orbit der Periode } n \Rightarrow f^n(x) = x.$$

Zunächst werden wir die Fixpunkte der zweiten Iterierten bestimmen. Dazu sei gesagt, dass natürlich auch die Fixpunkte der Funktion Fixpunkte der zweiten Iterierten sind. Denn ist $f^n(x) = x$, so ist auch $f^{nm}(x) = x$. Man kann die Fixpunkte der Funktion auch als Punkte der Periode eins ansehen. Uns sollen aber nur die Punkte der Periode zwei interessieren.

$$f(f(x)) = x \iff f(f(x)) - x = 0$$

Diese Gleichung ist 4. Grades, hat also vier Lösungen, die nur umständlich zu bestimmen sind. Dividiert man aber das Polynom $f(f(x)) - x$ durch x und $(x - 1 + 1/r)$, was den beiden Fixpunkten von f entspricht, so erhält man eine quadratische Gleichung mit den Lösungen

$$x_{1,2} = \frac{1}{2} + \frac{1}{2r} \pm \frac{1}{2} \sqrt{\left(1 + \frac{1}{r}\right) \left(1 - \frac{3}{r}\right)}.$$

Diese Gleichung liefert für unser Beispiel die Werte $x_1 = 0,513045$ und $x_2 = 0,799455$. Zeichnet man in das Spinnwebdiagramm auch noch die 2. Iterierte, so erhält man Abbildung 4.2.

Wie erwartet schneidet der Graph von $f(f(x))$ die Winkelhalbierende an vier Punkten: Den Fixpunkten von f und den Punkten der Periode 2.

Wir haben also eine notwendige Bedingung für periodische Orbits gefunden. Ob der Zyklus stabil ist, ob die Fixpunkte also Attraktoren sind, ist eine andere Frage. Dazu muss $|(f^2)'(x)| < 1$ sein.

$$f(f(x))' = f'(x) \cdot f'(f(x)) = f'(x)^2 = 4 + 2r - r^2$$

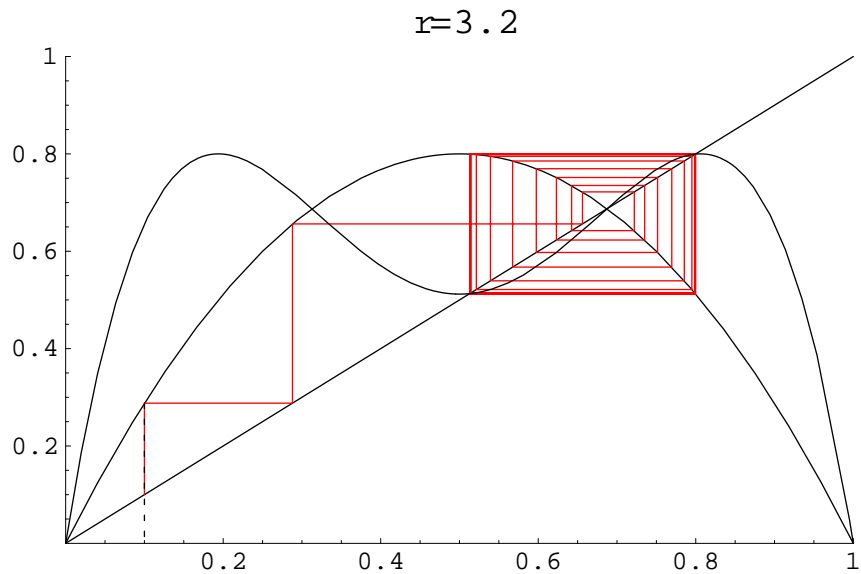


Abbildung 4.2: Spinnwebdiagramm und 2. Iterierte.

Die Ungleichung $|4 + 2r - r^2| < 1$ ist wahr für $3 < r \leq 1 + \sqrt{6}$. Ist r in diesem Intervall, so führt die Iteration zu einem 2-er Zyklus.

Im Vergleich zum 1-er Zyklus hat sich ab $r = 3$ die Periode verdoppelt. Man spricht daher von *Periodenverdopplung*. Die nächste Periodenverdopplung findet bei $1 + \sqrt{6}$ statt. Die Punkte, an denen eine Periodenverdopplung eintritt, nennt man *Bifurkationspunkte*. Weitere Bifurkationspunkte sind

3,544090
 3,564407
 3,568759
 3,569692
 3,569891
 3,569934

Diese Folge nähert sich einem Grenzwert, und zwar 3,569946. Ab dort wird (bis auf noch zu betrachtende Ausnahmen) das Verhalten des Orbits *chaotisch*. Bei $r = 4$ werden sogar alle reellen Zahlen im Intervall $[0; 1]$ von einem Orbit besucht. Die oben aufgelisteten Bifurkationspunkte lassen sich nur noch numerisch bestimmen, da der Grad der entstehenden Gleichungen zu hoch für eine analytische Behandlung ist.

Abbildung 4.3 und 4.4 zeigt das Spinnwebdiagramm für einen 8-er Zyklus und schließlich für den Chaosfall.

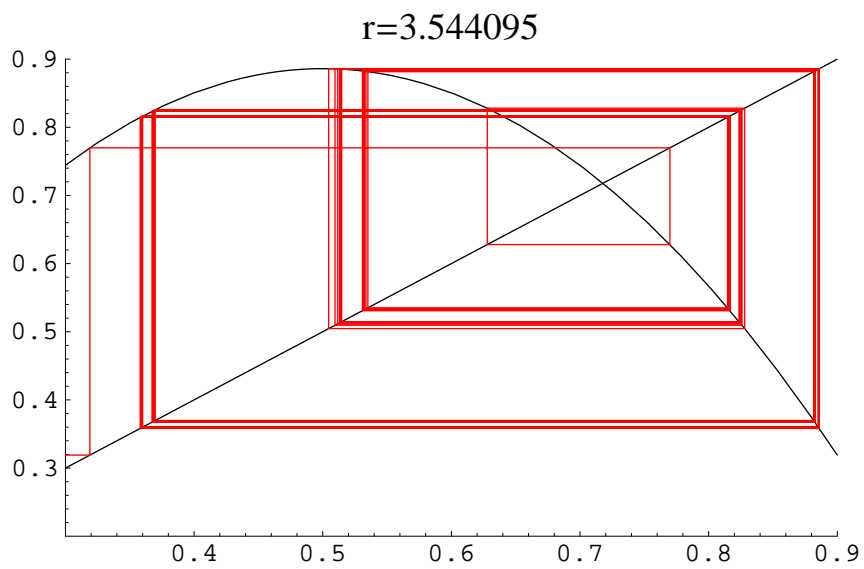


Abbildung 4.3: Periode 8.

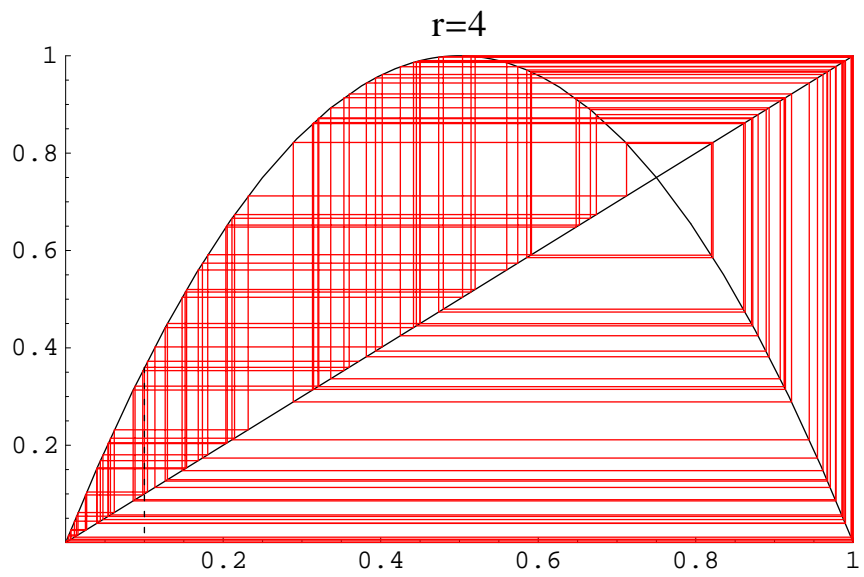


Abbildung 4.4: Chaos.

4.2 Bifurkations-Diagramme

Das alles ist noch wenig spektakulär. Betrachtet man jedoch den Quotienten der Abstände sukzessiver Periodenverdopplungen, so erhält man einen Grenzwert: 4,669202. Diese Zahl ist nach ihrem Entdecker Mitchell Feigenbaum *Feigenbaum-Konstante* getauft worden. Sie besitzt enorme mathematische Bedeutung, denn anders als der oben erwähnte Grenzwert der Bifurkationspunkte ist diese Konstante *universell*. Das bedeutet, dass es eine ganze Klasse von Funktionen gibt, deren Abstände sukzessiver Periodenverdopplungen gegen diesen Grenzwert konvergiert!

Einen guten Überblick über die Entwicklung des Chaos aus den Bifurkationen erhält man mit Hilfe der *Bifurkations-* oder *Feigenbaum-Diagramme*. Dabei plottet man die x -Werte des Orbits (ausgehend von einem beliebigen Startwert) über r . Um ein „einpendeln“ des Orbits zu ermöglichen, wählt man eine gewisse *Iterationszahl*, ab der die Werte erst ausgegeben werden. Das Bifurkationsdiagramm für die logistische Gleichung ist zwischen 0 und 4 in [Abbildung 4.5](#) dargestellt.

In diesem Diagramm finden wir unsere oben algebraisch hergeleiteten Ergebnisse wieder. Bis $r = 1$ konvergiert der Orbit gegen 0. Von 1 bis 3 ist $1 - 1/r$ Attraktor. Dann findet die erste Periodenverdopplung statt. Die zweite und dritte Periodenverdopplung ist noch zu erkennen, dann folgt Chaos. [Abbildung 4.6](#) zeigt die Bifurkationen vergrößert.

In diesem Diagramm wurde ein anderes *Färbe-Verfahren* angewendet, um die Strukturen im chaotischen Bereich sichtbar zu machen. Es wurde nicht, wie in obigem Diagramm, jeder Pixel gleich gefärbt, sondern die Pixelfarbe wurde bei jedem Treffer inkrementiert. Man erkennt deutlich *periodische Fenster*, in denen plötzlich wieder periodische Zyklen auftreten. Ihre Periode beträgt jedoch 3, 5, 6 usw. und zugehörige Bifurkationen. Nach dem wichtigen *Sarkowskii-Theorem* zieht ein Zyklus der Periode 3 alle weiteren Perioden und damit Chaos nach sich. [Abbildung 4.7](#) eines dieser periodischen Fenster.

4.3 Ausblicke

Die obigen Diagramme wurden alle nach dem gleichen Verfahren erstellt. Man nimmt für jedes abzubildende r einen Startwert (Zufallszahl aus $]0; 1[$) und iteriert sie z. B. 1000 mal. Ab der 500sten Iteration wird der Orbit ausgegeben.

Interessante Diagramme erhält man, wenn man die *Voriteration* weglässt, und jeden Punkt des Orbits plottet. Hier zeigen sich deutliche Unterschiede: Wird als Startwert keine Zufallszahl genommen, sondern für jedes r die gleiche Zahl, so zeigen sich deutliche Muster im Bifurkationsdiagramm.

So viel zur logistischen Gleichung. Wie gesagt ist die Feigenbaum-Konstante universell. [Abbildung 4.10](#) zeigt noch ein Bifurkationsdiagramm einer Funktion, bei der die Feigenbaum-Konstante auftritt. Natürlich kann man von beliebigen Funktionen Bifurkationsdiagramme erstellen. Unter Umständen tritt dann ebenfalls eine Konstante auf, die die Bedeutung der Feigenbaum-Konstanten einnimmt. Den Diagrammen ist das nicht anzusehen! Hätte man damals mit diesen Grafiken statt mit riesigen Zahlentabellen die logistische Gleichung untersucht, so wäre die Konstante wohl nicht entdeckt worden!

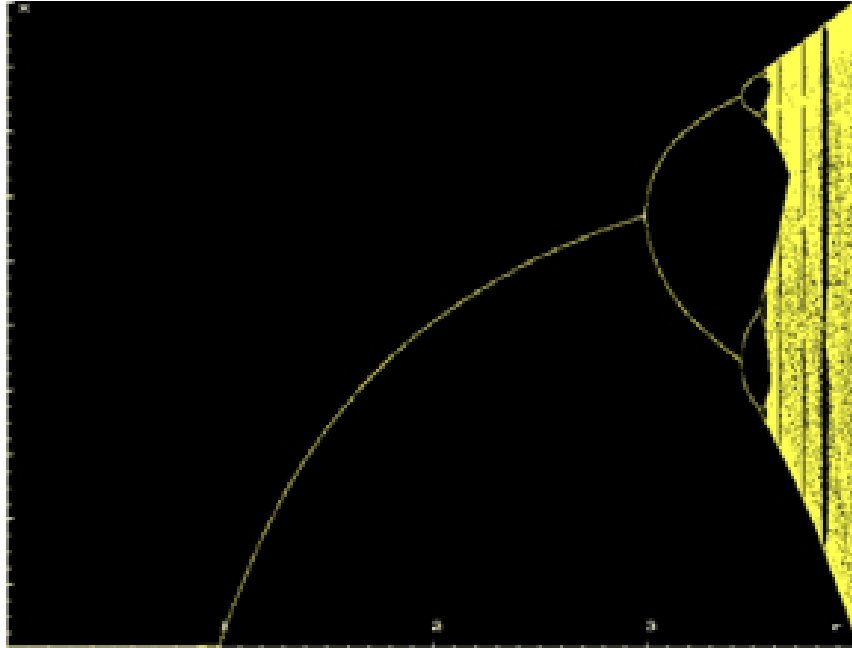


Abbildung 4.5: Bifurkationsdiagramm von $r = 0$ bis $r = 4$.

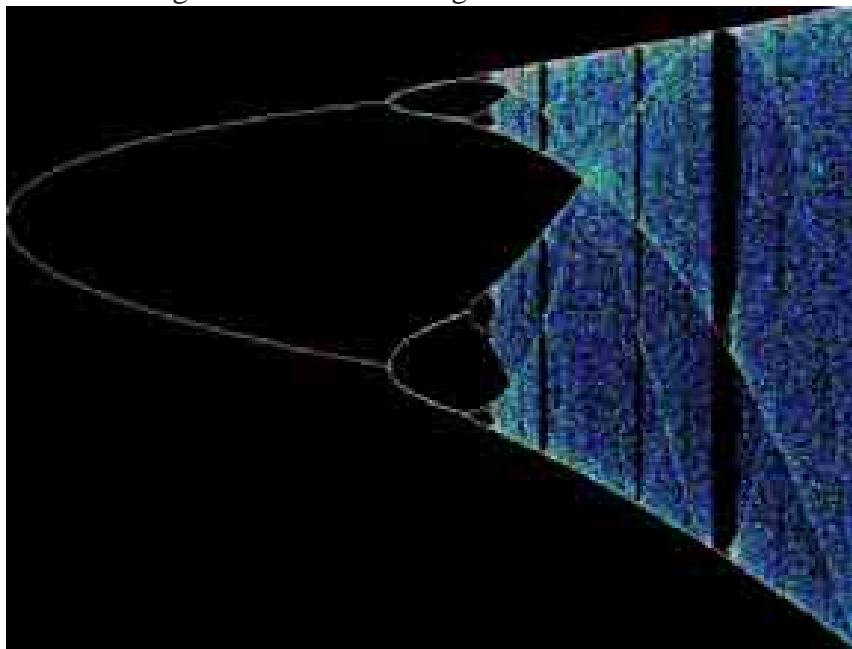


Abbildung 4.6: Bifurkationsdiagramm von $r = 3$ bis $r = 4$.

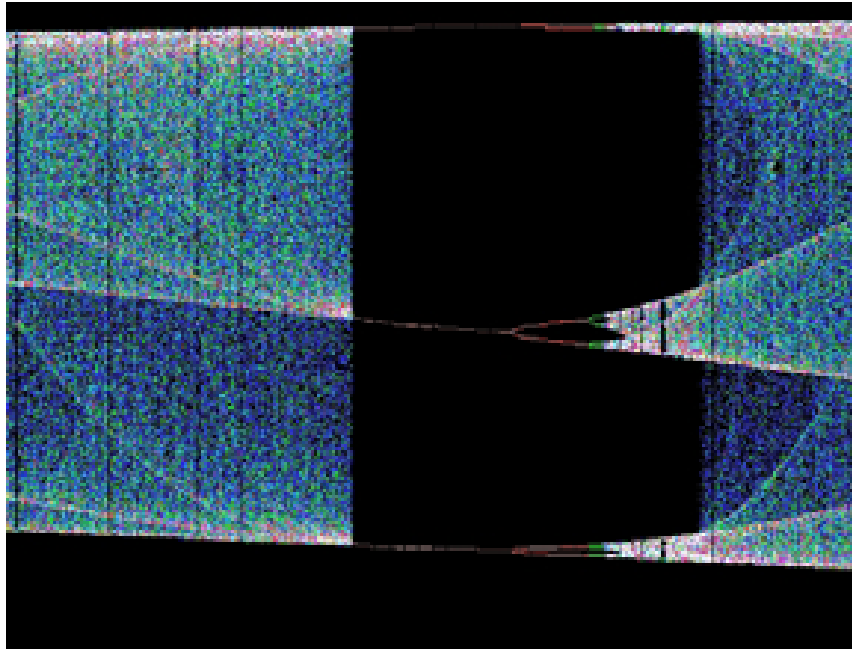


Abbildung 4.7: Periodisches Fenster.

4.4 Aufgaben

1. Erstelle Spinnwebdiagramme für Zyklen mit der Periode 2, 4 und 8 mit je verschiedenen Startwerten!
2. Suche einen Zyklus der Periode 3 und erstelle ein Spinnwebdiagramm!
3. Ermittle die Gleichungen der ersten drei Iterierten und stelle sie gemeinsam mit der Winkelhalbierenden und der Funktion dar. Was fällt auf?
4. Was geschieht mit dem Orbit von x für $r > 4$?
5. Finde weitere periodische Fenster und erstelle Bifurkationsdiagramme von ihnen!
6. Wie sind die Kurven im Bifurkationsdiagramm zu erklären, die entstehen, wenn man die Voriteration weglässt und einen konstanten Startwert wählt? Ermittle explizite Funktionsgleichungen für die ersten 5 Kurven!
7. Wieso muss der Startwert aus $]0; 1[$ statt $[0; 1]$ sein?

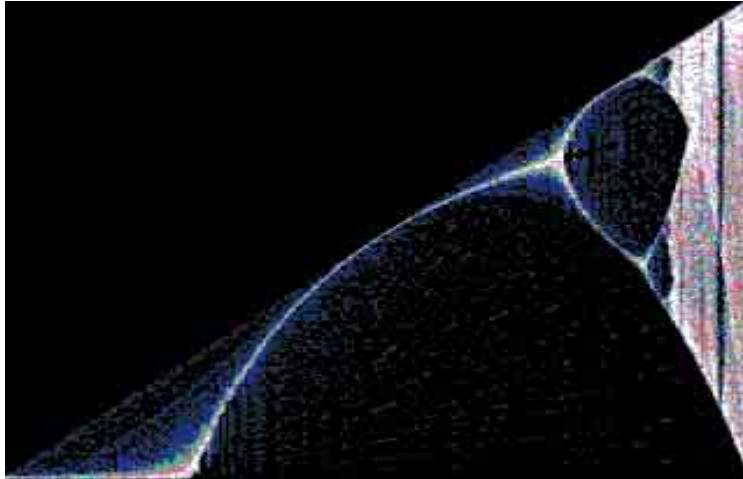


Abbildung 4.8: Voriterationen bei Zufallszahl.

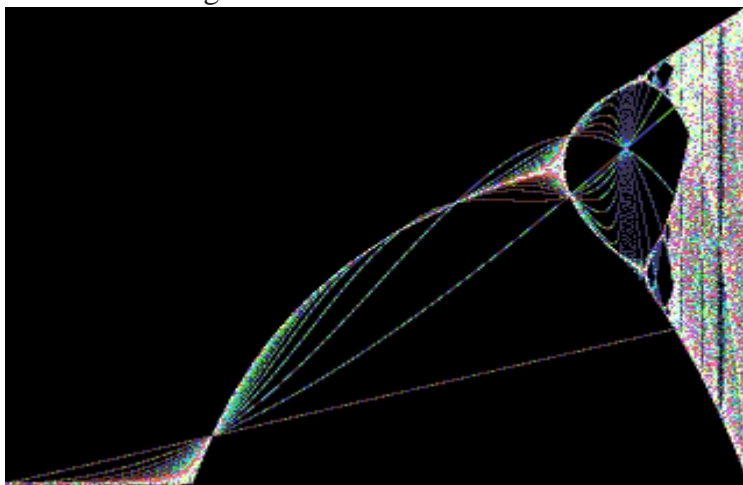


Abbildung 4.9: Voriterationen bei festgehaltenem Startwert.

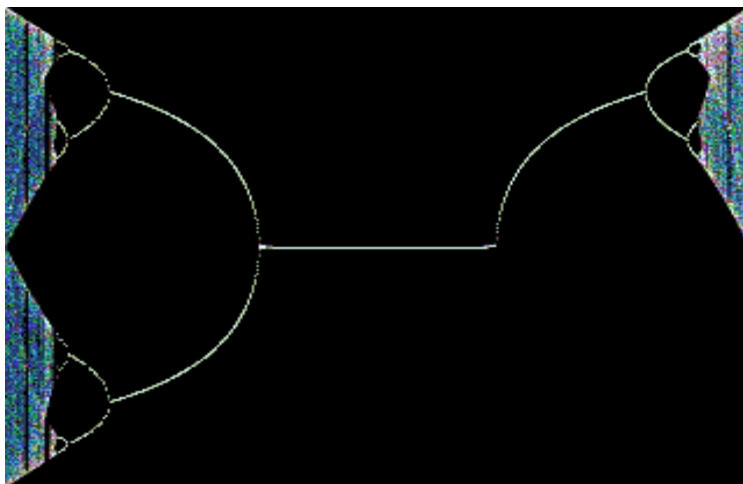


Abbildung 4.10: Bifurkationsdiagramm der Gleichung $x_{n+1} = r \sin(\pi x_n)$.

5 Julia-Mengen

Man kann das Iterieren von Funktionen auch für komplexe Zahlen betreiben. Betrachtet wird die Funktion

$$f_c(z) = z^2 + c.$$

Das ist so zu verstehen, dass ausgehend vom Startwert z der Orbit mit den Punkten $z^2 + c$, $(z^2 + c)^2 + c$ usw. gebildet wird. Die Iteration erfolgt also in der z -Ebene; c ist ein *Parameter* dieser Funktion. Dies ist ein wichtiger Unterschied zur Mandelbrot-Menge.

Es ist nun interessant zu fragen, für welche komplexe Zahl z der Orbit mit dem Parameter c beschränkt ist, d. h. nicht gegen unendlich divergiert. Dieses Problem hat Gaston Julia untersucht; die Mengen dieser z heißen *Julia-Mengen*. Mathematisch formuliert ist

$$K_c = \left\{ z \in \mathbb{C} \mid \lim_{n \rightarrow \infty} |f_c^n(z)| < \infty \right\}.$$

Manchmal bezeichnet der Begriff Julia-Menge auch nur den Rand dieser Menge. Sie wird dann mit J_c statt K_c bezeichnet.

Trivial ist der Fall $c = 0$. Hier bildet der Einheitskreis die Menge J_0 ; die Punkte seines Innern sind K_0 .

Beweis:

$$z = r \cdot e^{i\varphi} \Rightarrow f^n(z) = r^{2^n} \cdot e^{i2^n\varphi}$$

Daraus sieht man, dass $f^n(z)$ für $r > 1$ gegen unendlich divergiert, wogegen der Attraktor für $r < 1$ der Ursprung ist. Bei $r = 1$ wandert der Orbit auf dem Einheitskreis umher. \square

Der einzige weitere Fall, für den die Julia-Menge kein Fraktal ist, ist der Fall $c = -2$. Hier ist $K_{-2} = J_{-2} = [-2; 2]$.

Man kann Julia-Mengen darstellen, indem man für jedes z der komplexen Ebene testet, ob der Orbit beschränkt bleibt. Ist das der Fall, so wird der Punkt schwarz eingefärbt, ansonsten bleibt der Punkt weiß.

Doch hier ergibt sich ein Problem: Wie soll man wissen, ob der Orbit beschränkt bleibt? Man kann ihn schließlich nur für eine endliche Anzahl von Iterationen untersuchen. Es ist durchaus möglich, dass der Orbit 1000 Iterationen lang sehr nah am Ursprung verläuft, um dann plötzlich gegen unendlich zu verschwinden, das kann man nicht ausschließen. Aber das Gegenteil ist möglich: Es gibt eine Bedingung, die festlegt, wann ein Orbit auf jeden Fall gegen unendlich divergiert.

Satz: Ist $|z_n| > 2$ und $|z_n| > |c|$, so divergiert der Orbit von z gegen unendlich.

Beweis: $|z_{n+1}| > |z_n|^2 - |c| > |z_n|^2 - |z_n| = |z_n| \cdot (|z_n| - 1) > |z_n|$. \square

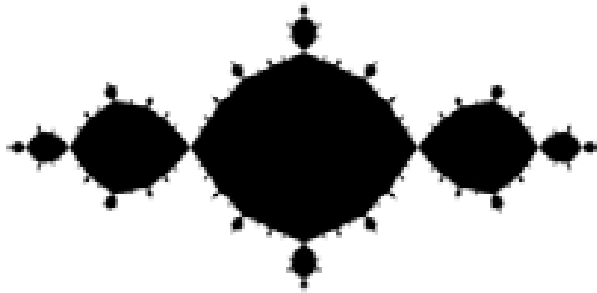


Abbildung 5.1: Julia-Menge für $c = -1$.

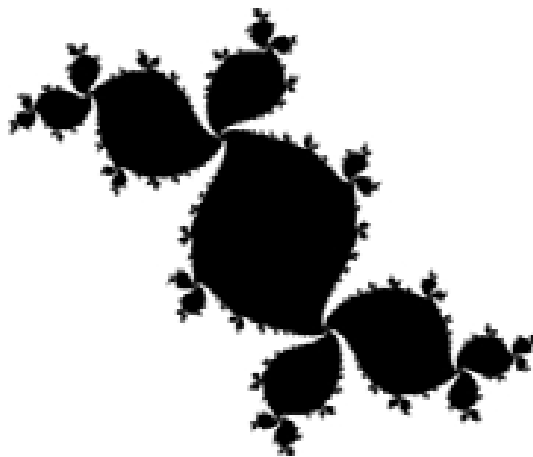


Abbildung 5.2: Julia-Menge für $c = -0,1 + 0,7i$.

Aus Gründen, auf die wir noch zu sprechen kommen, ist $|z_n| > |c|$ fast automatisch erfüllt. Wichtig ist also nur $|z_n| > 2$.

Jetzt können wir unser erstes Bild generieren. Dabei nehmen wir an, dass ein Punkt, dessen Orbit nach 100 Iterationen den Kreis mit Radius 2 noch nicht verlassen hat, zur Julia-Menge gehört. Näheres zu diesem Verfahren in Kapitel 8. Wir wählen $c = -1$ und erhalten Abbildung 5.1.

Die Julia-Menge ist offenbar *zusammenhängend*. Außerdem ist sie stark *selbstähnlich*. Diese Selbstähnlichkeit ist zwar nicht exakt, doch sie setzt sich bis in beliebig kleine Skalen fort, und die charakteristische Form der Ränder bleibt erhalten.

Mit $c = -0,1 + 0,7i$ bekommen wir Abbildung 5.2. Man erkennt die *Punktsymmetrie* der Julia-Menge zum Ursprung. Wählen wir stattdessen das hierzu konjugiert komplexe c , so erhalten wir das gleiche Bild, das nur an der reellen Achse gespiegelt ist. Der Grund hierfür wird weiter unten erläutert.

Um uns ein Bild von der möglichen Formenvielfalt der Julia-Mengen zu machen, probieren

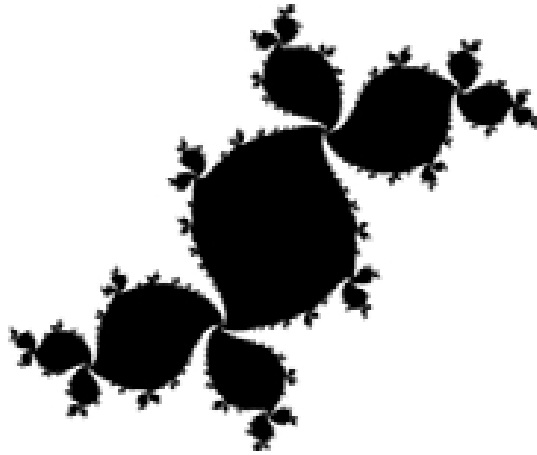


Abbildung 5.3: Julia-Menge für $c = -0,1 - 0,7i$.

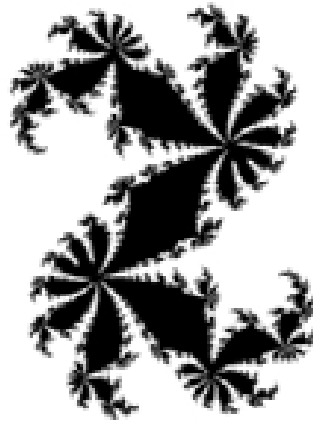


Abbildung 5.4: Julia-Menge für $c = 0,360284 + 0,1i$.

wir einfach weitere Parameterwerte aus. Wir beschränken uns dabei aus den oben gefundenen Symmetrieprinzipien auf ein c mit positivem Imaginärteil.

Um die Julia-Mengen aus Abbildung 5.5-5.7 sichtbar zu machen, wurde ein anderes Färbe-Verfahren verwendet (siehe Kapitel 8).

Diese beiden letzten Julia-Mengen sind nicht mehr zusammenhängend, was zumindest in Abbildung 5.6 deutlich zu erkennen ist. Genauer gesagt sind diese Mengen *Cantor-Mengen*, bestehen also aus unendlich vielen, separaten Punkten, die absolut unzusammenhängend sind. Das heißt, kein Punkt dieser Menge kann mit einem anderen Punkt der Menge durch eine Kurve verbunden werden, deren Punkte sämtlich Teil der Julia-Menge wären. In den oberen Fällen ist das relativ problemlos möglich.

Wir haben eine fundamentale Eigenschaft der Julia-Mengen gefunden.

Satz: Eine Julia-Menge ist entweder zusammenhängend, oder sie ist eine Cantor-Menge. Eine Julia-Menge ist genau dann zusammenhängend, wenn c in der Mandelbrot-Menge liegt. Dieser Zusammenhang heißt *Fundamentale Dichotomie*.



Abbildung 5.5: Julia-Menge für $c = i$.

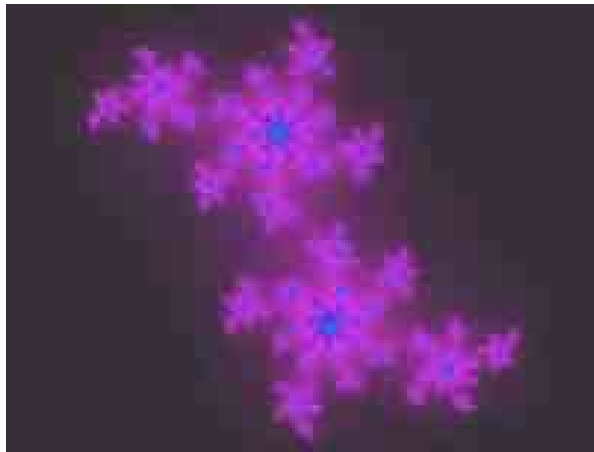


Abbildung 5.6: Julia-Menge für $c = 0,11301 + 0,67037i$.

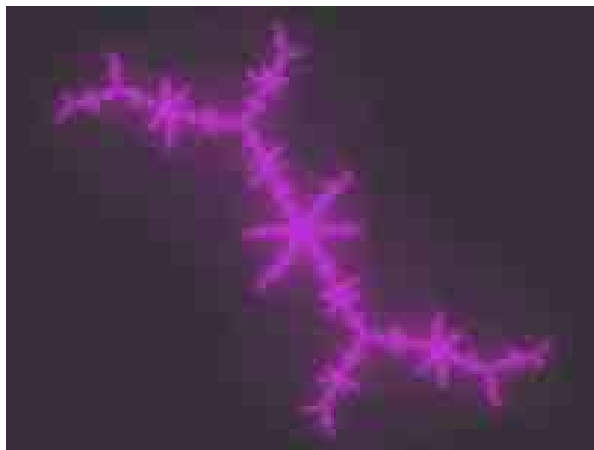


Abbildung 5.7: Julia-Menge für $c = -0,1011 + 0,9563i$.

Das bedeutet anschaulich: Eine Julia-Menge besteht entweder aus einem oder aus unendlich vielen Teilen (und zwar dann aus separaten Punkten). Eine Julia-Menge, die aus zwei, drei oder vier Teilen besteht, wird man nicht finden!

Hiermit lässt sich nun die oben gefundene Symmetrie zwischen der Julia-Menge von c und der des dazu konjugiert komplexen \bar{c} erklären. Denn die Mandelbrot-Menge ist symmetrisch zur reellen Achse. Ist also c Element der Mandelbrot-Menge, so ist auch das konjugiert komplexe \bar{c} in ihr.

Besonders interessant sind die Julia-Mengen der c , die am Rand der Mandelbrot-Menge liegen. Mehr dazu im folgenden Kapitel.

Auch klar wird nun, wieso $|z_n| > 2$ wichtiger ist als $|z_n| > |c|$. Denn für die interessantesten Fälle, d. h. die zusammenhängenden (oder gerade nicht mehr zusammenhängenden) Julia-Mengen, liegt c in der Mandelbrot-Menge (oder dicht daneben), welche im Kreis um den Ursprung mit dem Radius 2 eingeschlossen ist. Die zweite Bedingung ist also automatisch erfüllt, wenn es die erste ist.

6 Die Mandelbrot-Menge

Die *Mandelbrot-Menge* (nach Benoit Mandelbrot) ist die Menge aller c , für die die Julia-Mengen zusammenhängend sind. Ihre formale Definition ist

$$M = \left\{ c \in \mathbb{C} \mid \lim_{n \rightarrow \infty} |f_c^n(z)| < \infty ; z_0 = 0 \right\}.$$

Sie ist also die Menge der c , für die die Iteration $f_c(z)$ mit dem Startwert $z_0 = 0$ beschränkt bleibt. Warum, wird weiter unten erläutert. Wichtig ist, dass Darstellungen der Mandelbrot-Menge Bilder in der *Parameter-Ebene* (c -Ebene) sind, wogegen Julia-Mengen in der *dynamischen Ebene* (z -Ebene) dargestellt werden, d. h. in der Ebene, in der die Iteration erfolgt. Abbildung 6.1 zeigt ein Bild der Mandelbrot-Menge.

Wie schon erwähnt, ist sie symmetrisch zur reellen Achse. Außerdem ist sie selbstähnlich, wie wir an einem Beispiel noch sehen werden. Allerdings ist sie nicht exakt selbstähnlich, die kleinen Kopien des Ganzen sind etwas verzerrt. Die Mandelbrot-Menge selbst ist wieder zusammenhängend. Die Hausdorff-Dimension des Randes der Mandelbrot-Menge sowie die des Randes fast aller¹ zusammenhängender Julia-Mengen ist 2!

Abbildung 6.2 zeigt eine Gegenüberstellung der Mandelbrot-Menge mit dem Bifurkationsdiagramm von $x_{n+1} = x_n^2 + c$. Man sieht, wie die Mandelbrot-Menge an den Bifurkationspunkten die Struktur verändert.

6.1 Die Struktur der Mandelbrot-Menge

Wir sind bis jetzt noch nicht auf die Definitionsgleichung der Mandelbrot-Menge eingegangen. Kritisch ist besonders die Wahl der Startwertes z_0 (siehe Kapitel 8). Warum wurde $z_0 = 0$ gewählt? Nun, $z_0 = 0$ ist superstabiler Fixpunkt der Iterationsgleichung $f_c(z)$. Nach einem wichtigen Theorem wird, falls es einen stabilen periodischen Zyklus gibt, der Orbit, der von $z_0 = 0$ ausgeht, von ihm angezogen (man denke an die Spinnwebdiagramme!). Daraus folgt, dass es für jedes c höchstens einen stabilen periodischen Zyklus geben kann.

Zur Verdeutlichung: Wenn eine andere Zahl als Startwert gewählt wird, so kann ein c , das zu einem stabilen Zyklus fähig wäre, gegen unendlich divergieren und so fälschlicherweise aus der Mandelbrot-Menge ausgeschlossen werden. Daher wählt man $z_0 = 0$, um ein Maximum an zugehörigen c zu erhalten.

Ein Verfahren, die Mandelbrot-Menge zu berechnen, besteht darin, die Iteration durchzuführen, bis eine bestimmte Iterationszahl erreicht wird oder der Betrag von z einen gewissen Wert r erreicht (siehe Kapitel 8). Üblicherweise ist $r = 2$, wie bei den Julia-Mengen ausgeführt wird.

¹Eine offensichtliche Ausnahme ist der Fall $c = 0$ mit der Hausdorff-Dimension 1.

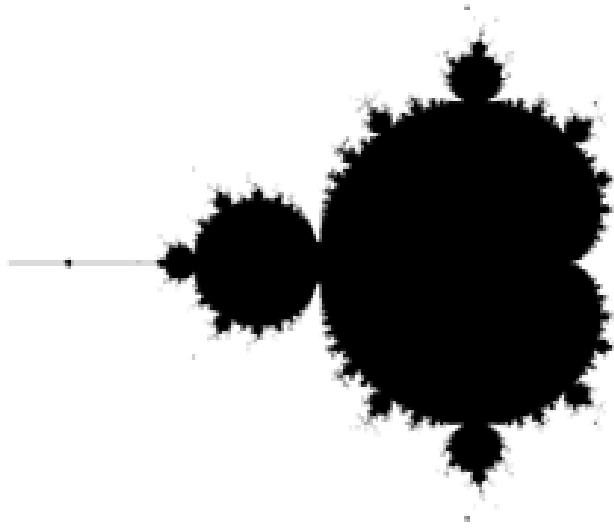


Abbildung 6.1: Die Mandelbrot-Menge.

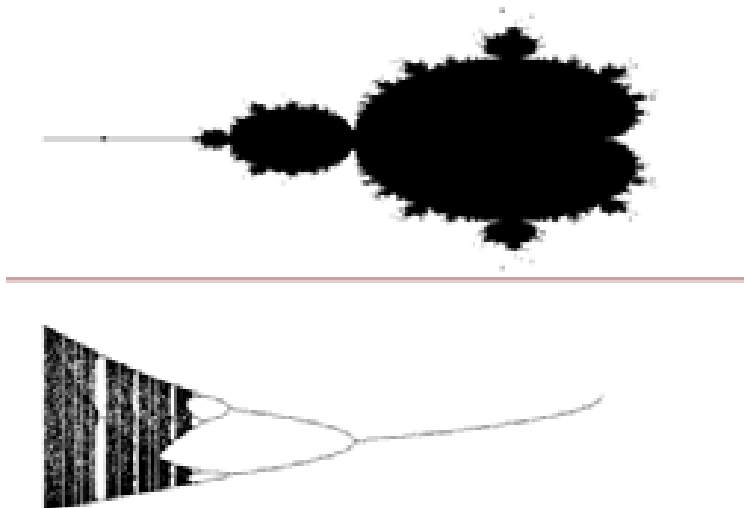


Abbildung 6.2: Mandelbrotmenge und Bifurkationsdiagramm.

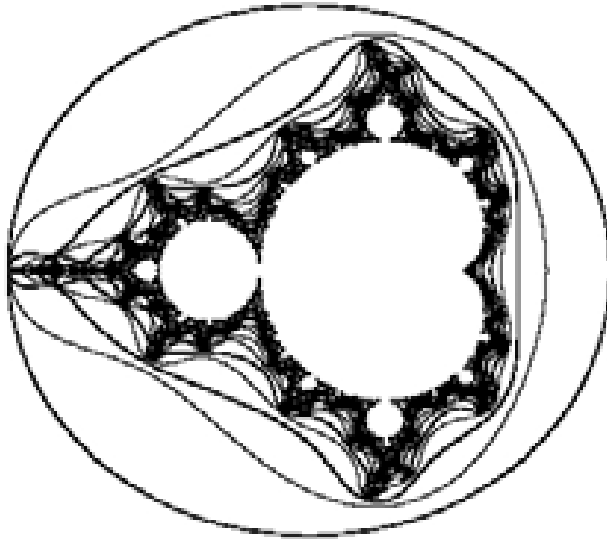


Abbildung 6.3: Äquipotentialkurven der Mandelbrotmenge.

Die c -Werte, die alle nach der gleichen Anzahl von Schritten ausscheiden, bilden eine zusammenhängende *Äquipotentialfläche*. Die Kurven, die diese Flächen trennen, heißen *Äquipotentialkurven* L_n . Sie kann man folgendermaßen berechnen:

Für alle c , deren Orbits nach einer Iteration den Abstand r vom Ursprung haben, gilt:

$$L_1 : |c| = r.$$

Die erste Äquipotentialkurve ist also ein Kreis um den Ursprung vom Radius r (wie man an der Koordinatendarstellung $x^2 + y^2 = r^2$ abliest). Die nächste Kurve ist die Menge aller c , deren Orbit nach zwei Iterationen den Abstand r erreicht:

$$L_2 : |c^2 + c| = r.$$

Hierbei handelt es sich um eine Ellipse. Die nächsten Kurven sind:

$$L_3 : |(c^2 + c)^2 + c| = r$$

$$L_4 : |((c^2 + c)^2 + c)^2 + c| = r$$

$$L_5 : |(((c^2 + c)^2 + c)^2 + c)^2 + c| = r$$

All dies sind implizit gegebene Kurven, für die die Bestimmung der Koordinatenform jedoch zunehmend kompliziert und umfangreich wird. Das ist auch verständlich, wenn man die vielen „Zacken“ der Kurven betrachtet, die mehr und mehr ausgeprägt werden. [Abbildung 6.3](#) zeigt ein Bild der Äquipotentialkurven.

Im Fall n gegen unendlich ist L_n der Rand der Mandelbrot-Menge.

Bis jetzt haben wir das Äußere der Mandelbrot-Menge betrachtet, nun werden wir in ihr Inneres sehen. Dazu suchen wir die Menge aller c , welche bei der Iteration gegen einen Fixpunkt konvergieren. Dann muss gelten:

$$z^2 + c = z \text{ und } |2z| < 1$$

bzw. für die Grenzkurve $|2z| = 1$

$$\Rightarrow 2z = e^{it} \Rightarrow z = 0,5e^{it}$$

Eingesetzt in die Fixpunktgleichung ergibt sich

$$c = 0,5e^{it} - 0,25e^{2it},$$

wobei der Parameter t von 0 bis 2π läuft. Dies ist die Gleichung der *Herzkurve*, die den „Rumpf“ der Mandelbrot-Menge bildet. Sie schneidet die reelle Achse an den Punkten $c = -0,75$ und $c = 0,25$. Die gesamte Mandelbrot-Menge reicht übrigens von $c = -2$ bis $c = 0,25$, d. h. wir haben ihren rechten Rand schon gefunden.

Als nächstes bestimmen wir die c , die einen 2er-Zyklus bilden. Die Fixpunktgleichung ist

$$(z^2 + c)^2 + c = z \Rightarrow z^4 + 2z^2c - z + c^2 + c = 0.$$

Natürlich kommen auch hier wieder die Fixpunkte als Lösungen vor; nach Division durch $z^2 - z + c$ ergibt sich

$$z^2 + z + c + 1 = 0 \Rightarrow z = \frac{-1 \pm \sqrt{-3 - 4c}}{2}.$$

Setzt man in der Ableitung $|4z^3 + 4zc| = 1$ die Fixpunkte ein, so erhält man

$$|4(1 + c)| = 1 \Rightarrow |1 + c| = 0,25 \Rightarrow c = 0,25e^{it} - 1.$$

Dies ist ein Kreis um $c = -1$ mit dem Radius 0,25. Abbildung 6.4 zeigt die bisher erhaltenen Ergebnisse.

Jede weitere natürliche Zahl kommt ebenfalls als Periode vor, und zwar in der jeweils nächstgrößeren „Beule“ der Mandelbrot-Menge. Am Rand der Mandelbrot-Menge schließlich sind die Orbits chaotisch.

Daraus erhalten wir ein wichtiges Ergebnis: Die Punkte einer „Beule“ haben alle dieselbe Periode. Aber nicht nur das, man kann die Periode der „Beule“ geradezu ansehen. Sie ist nämlich gleich der „Strahlen“, die von ihr ausgehen, und zwar den „Hauptstrahl“ mitgezählt. Abbildung 6.5 zeigt eine „Beule“ der Periode 7.

Außerdem erhält man die Periode einer „Beule“, indem man die Julia-Menge eines Punkts im Innern der „Beule“ berechnet und die Anzahl der „Arme“ zählt. Abbildung 6.6 stellt die zur obigen „Beule“ zugehörige Julia-Menge dar.

6.2 Eine fraktale Reise

Wie schon erwähnt ist die Mandelbrot-Menge selbstähnlich. Das soll an einer beispielhaften Reise verdeutlicht werden. Zum besseren Nachvollziehen ist auch jeweils die Koordinate der oberen linken und rechten unteren Bildschirmecke gegeben.

Wir haben wirklich eine Mini-Mandelbrot-Menge gefunden! Tatsächlich ist der Rand der Mandelbrot-Menge voll davon.

Am Ende befinden sich noch ein paar weitere Ausschnitte aus der Mandelbrot-Menge, die den angedeuteten Formenreichtum verdeutlichen sollen. Bei der Erstellung der Bilder wurde auf Effekte, wie sie in Kapitel 8 erklärt werden, verzichtet.

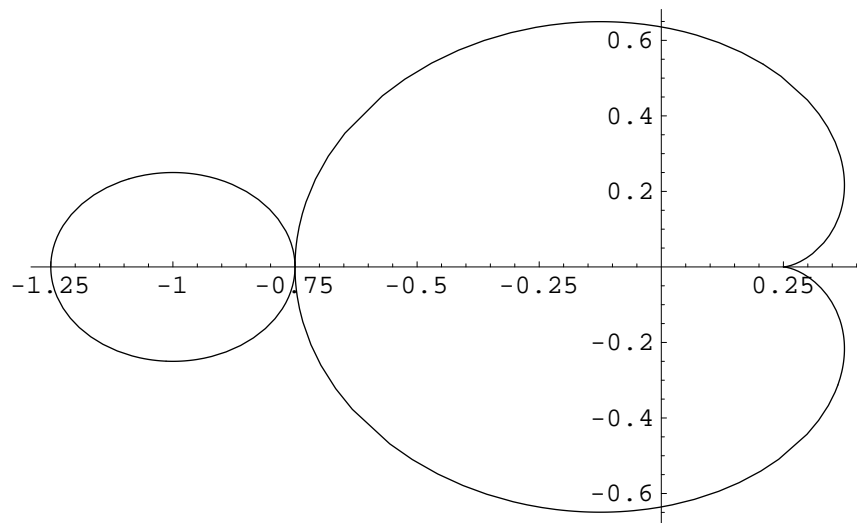


Abbildung 6.4: Rumpf und Kopf der Mandelbrot-Menge.

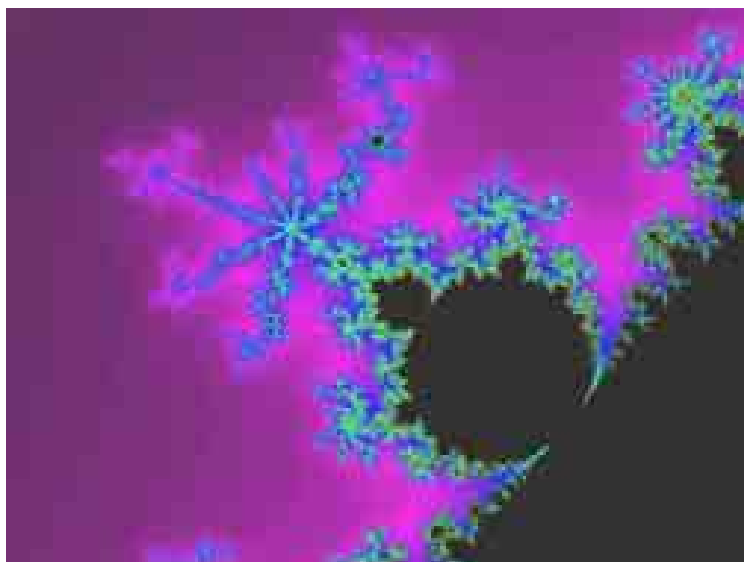


Abbildung 6.5: Ausschnitt aus der Mandelbrot-Menge.

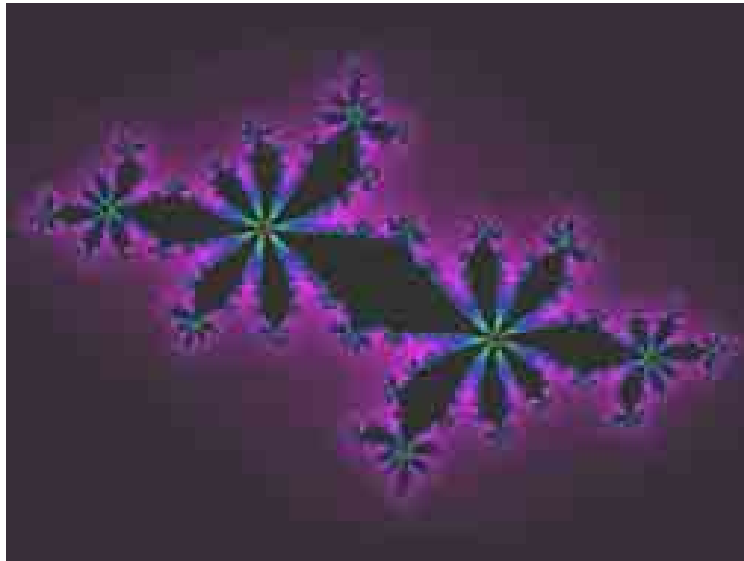


Abbildung 6.6: Entsprechende Julia-Menge.

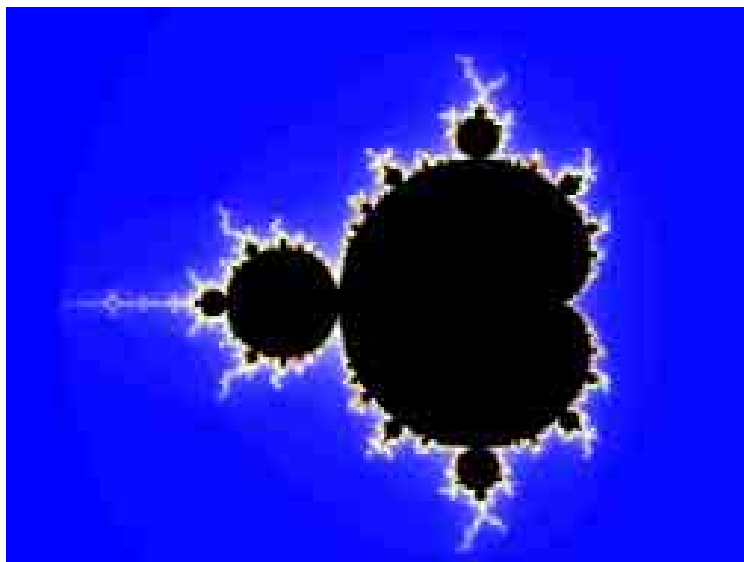


Abbildung 6.7: $(-2,237251;1,324524)$ $(1,110561;-1,186334)$.

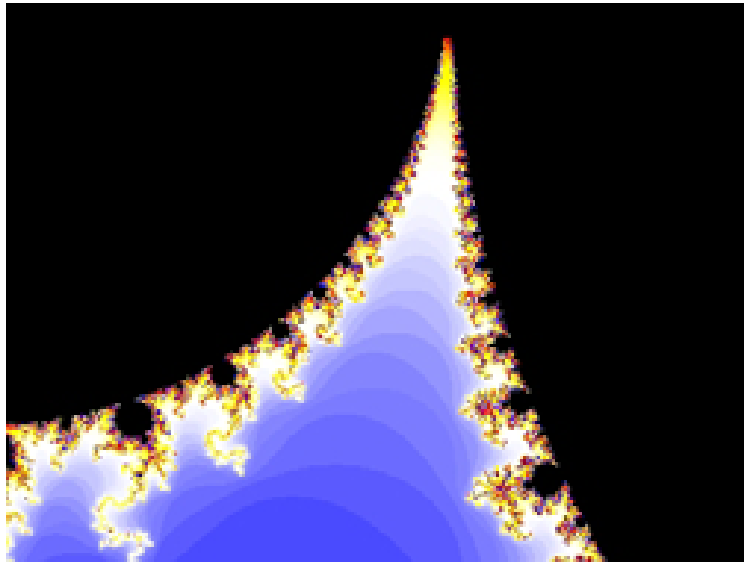


Abbildung 6.8: $(-0,9875;-0,025)$ $(-0,5875;-0,325)$.

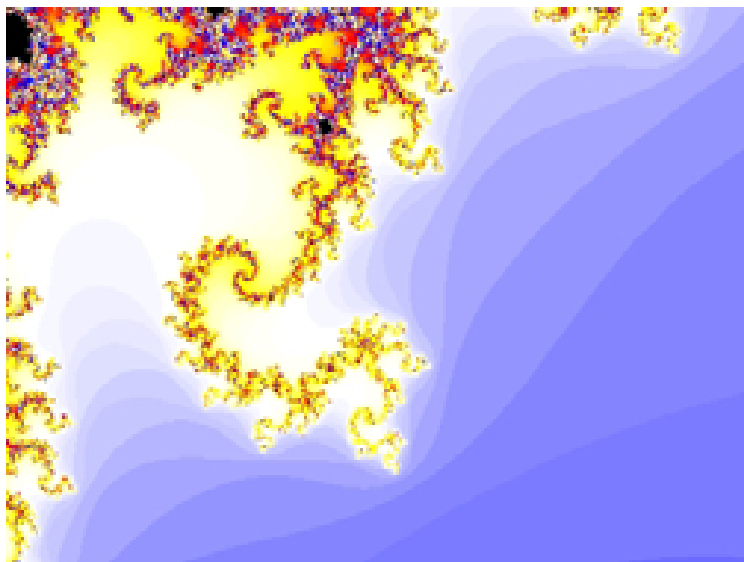


Abbildung 6.9: $(-0,9;-0,11625)$ $(-0,67;-0,28875)$.

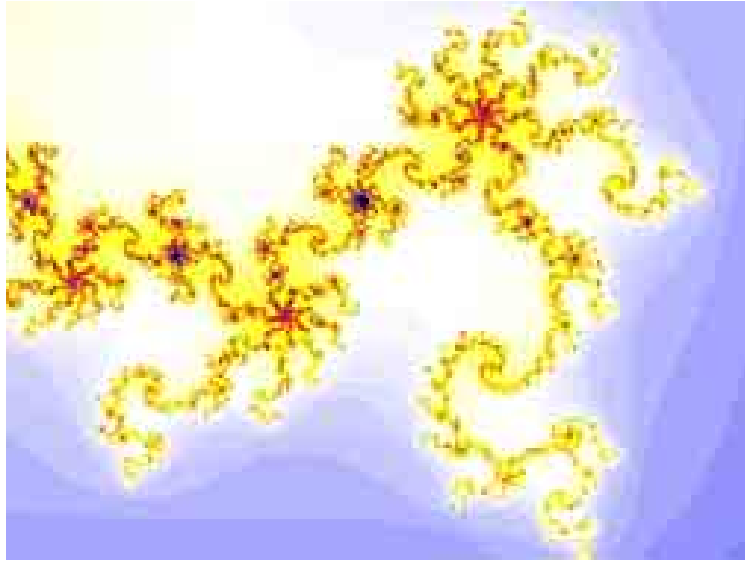


Abbildung 6.10: $(-0,841374;-0,226834)$ $(-0,825733;-0,238565)$.

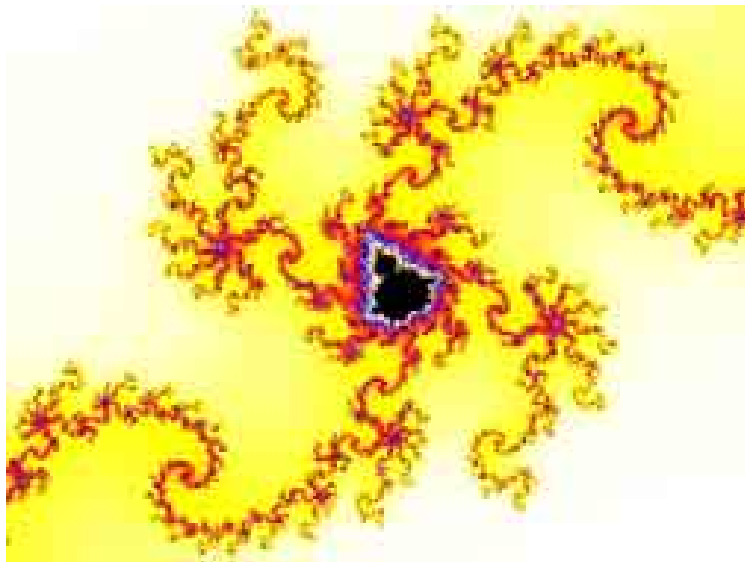


Abbildung 6.11: $(-0,835655;-0,229743)$ $(-0,832331;-0,232236)$.

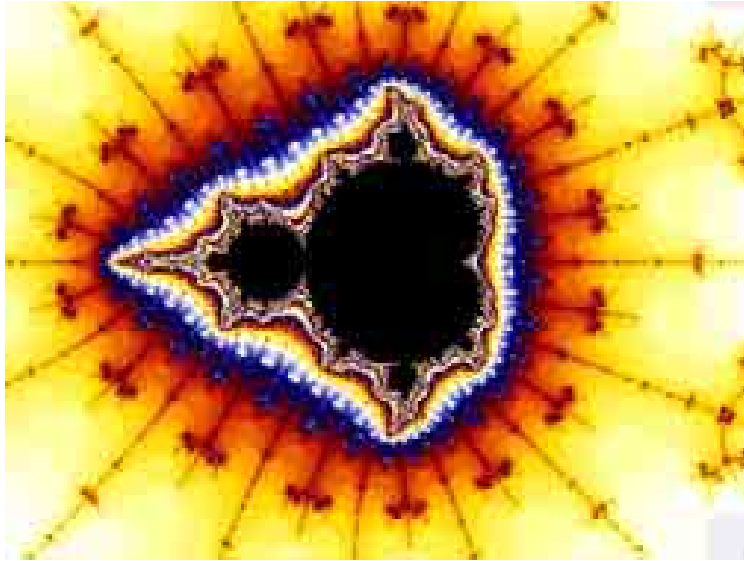


Abbildung 6.12: Mini-Mandelbrot-Menge.

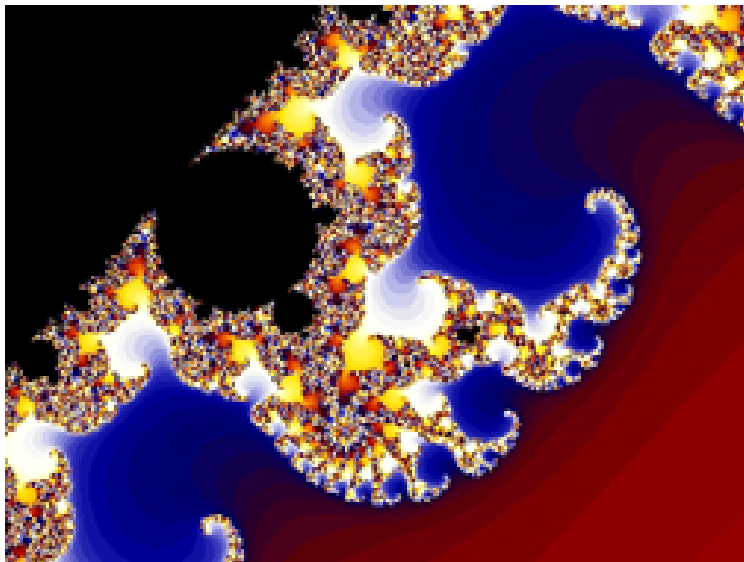


Abbildung 6.13: „Beule“ am Rand der Mandelbrot-Menge.

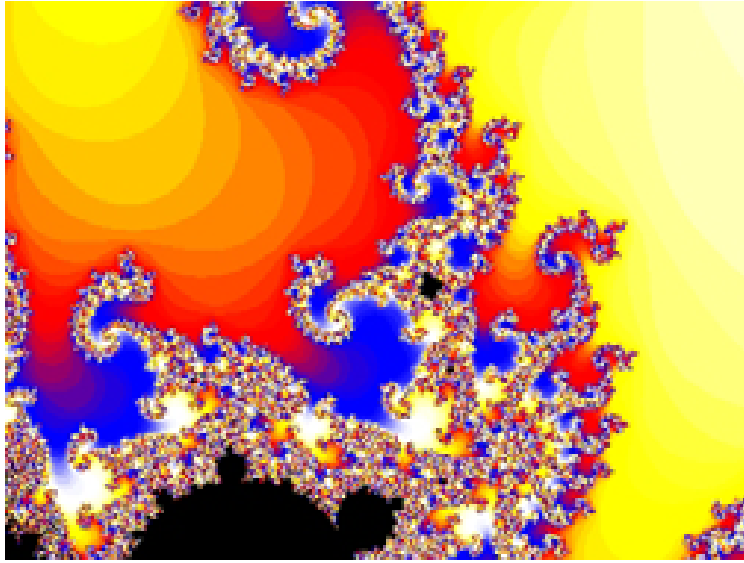


Abbildung 6.14: Mini-Mandelbrot-Menge in der Nähe einer „Beule“.

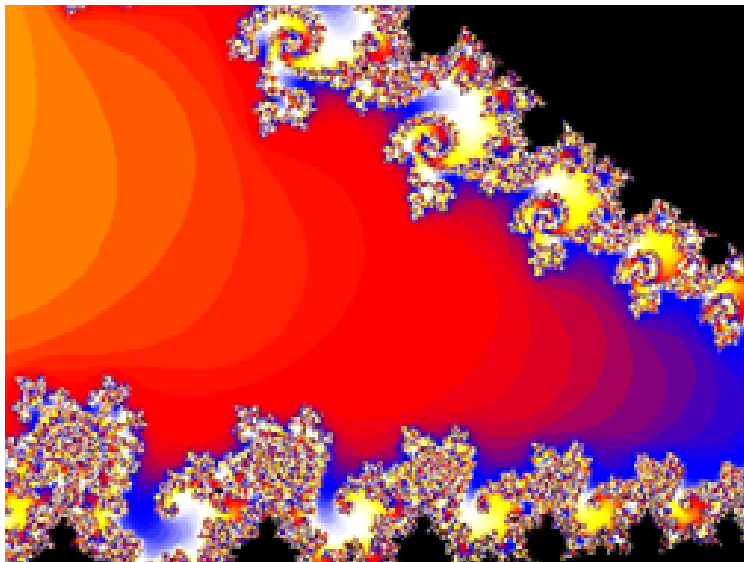


Abbildung 6.15: Der obere Rand des „Rumpfes“.

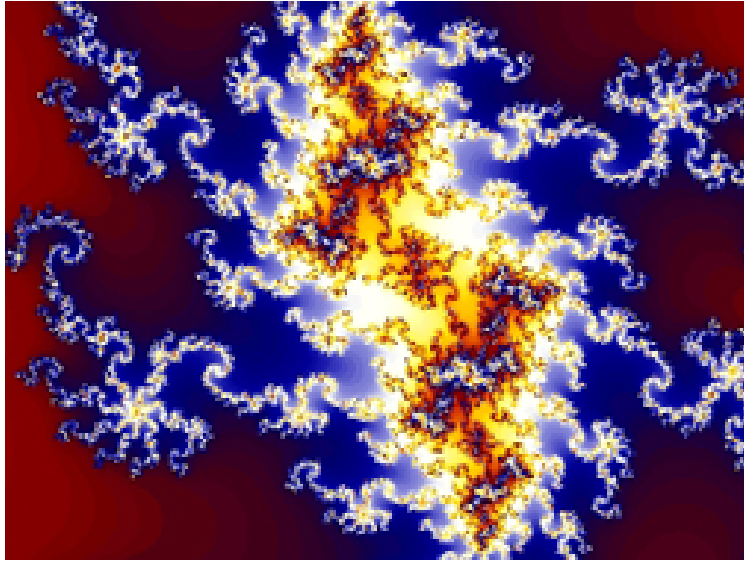


Abbildung 6.16: Filamente.

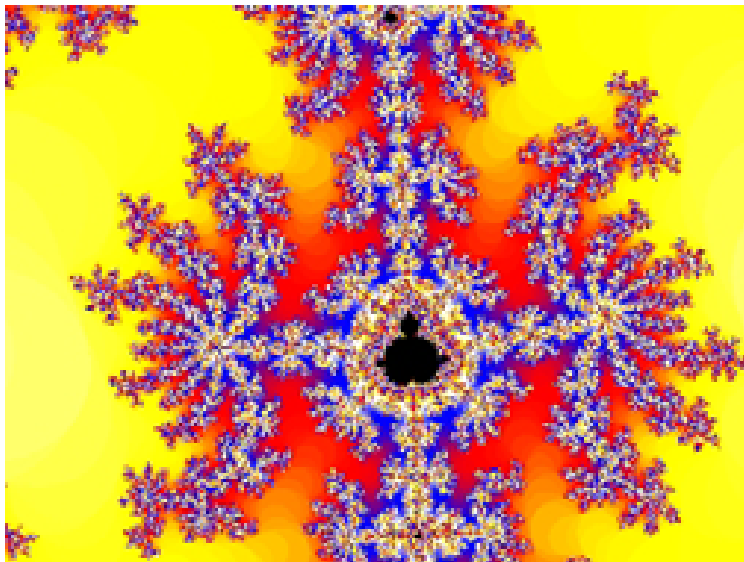


Abbildung 6.17: Mini-Mandelbrot-Menge.

6.3 π und die Mandelbrot-Menge

Eines der faszinierendsten Dinge an der Mandelbrot-Menge ist, dass man alles in ihr finden kann, wenn man nur lange genug sucht. So ist es höchst erstaunlich, in welcher Art die Kreiszahl π sich in ihr wiederfindet.

Der oben berechnete Berührungspunkt zwischen „Kopf“ und „Rumpf“ der Mandelbrot-Menge liegt bei $c = -0,75$. An dieser Stelle ist die Mandelbrot-Menge unendlich dünn, nur der Punkt $c = -0,75$ gehört dazu, alle anderen Punkte der Form $c = -0,75 + i10^{-k}$ divergieren gegen unendlich. Man kann nun bestimmen, nach wie vielen Iterationen n das der Fall ist. Man erhält folgende Tabelle:

k	n
0	3
1	33
2	315
3	3143
4	31417
5	314160
6	3141593
7	31415928

Was sofort auffällt: Für genügend großes k ist $10^{-k} \cdot n = \pi$!

Eine weitere interessante Stelle ist der rechte Rand der Mandelbrot-Menge bei $c = 0,25$. Bestimmt man wieder eine Tabelle für $c = 0,25 + 10^{-k}$, so erhält man:

k	n
0	2
1	8
2	30
3	97
4	312
5	991
6	3140
7	9933
8	31414
9	99344
10	314157

Diesmal ist $\sqrt{10^{-k}} \cdot n = \pi$!

Plausibilitätsbetrachtung:

Bei der Iteration von $z_{n+1} = z_n^2 + 0,25 + \varepsilon$ mit $z_0 = 0$ erhöht sich der Wert von z langsam bis $0,5$ und verschwindet dann schnell nach unendlich. Kritisch ist also der Fall $z = 0,5$. Setzt man $z = t + 0,5$, so erhält man $t_{n+1} = t_n^2 + t_n + \varepsilon$. Zumindest bei $t = 0$ erhöhen sich die Werte nur langsam, daher kann t näherungsweise als eine stetige Funktion von n angesehen werden. Die Differenz $t_{n+1} - t_n$ entspricht dann der ersten Ableitung $t'(n)$. Man erhält die

Differentialgleichung $t'(n) = t(n)^2 + \varepsilon$ mit der Lösung $t = \sqrt{\varepsilon} \cdot \tan(\sqrt{\varepsilon} \cdot n + C)$, wobei C die Integrationskonstante ist. Die Anfangs- und Endpunkte der Iteration entsprechen zwei sukzessiven Polen der Tangensfunktion. Daher gilt $\sqrt{\varepsilon} \cdot n = \pi$, was unserer Beobachtung entspricht.

6.4 Verallgemeinerungen

Die Mandelbrot-Menge kann sehr leicht verallgemeinert werden, indem man die Funktionsgleichung $f_c(z) = z^n + c$ betrachtet. Dann fällt auf, dass sich die Anzahl der „Bäuche“ der Mandelbrot-Menge erhöht, und zwar ist sie gleich $n - 1$. Die Abbildungen 6.18 und 6.19 zeigen die Mandelbrot-Mengen für $n = 3$ und $n = 4$.

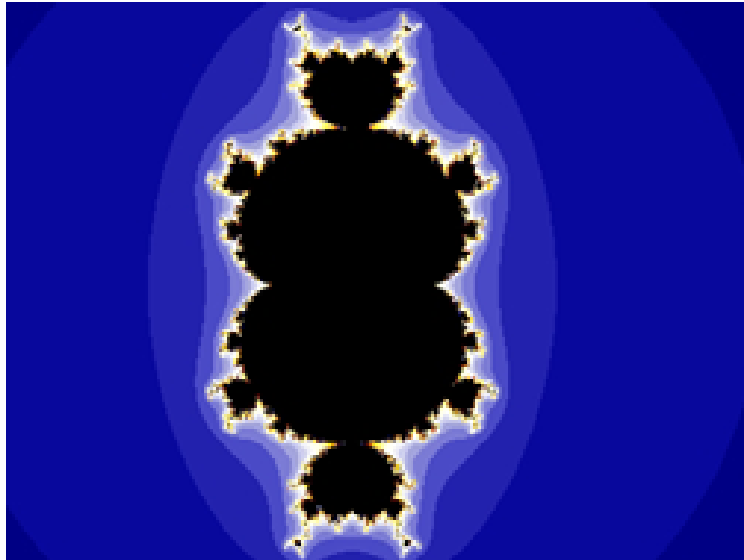


Abbildung 6.18: Mandelbrot-Menge zu $z_{n+1} = z_n^3 + c$.

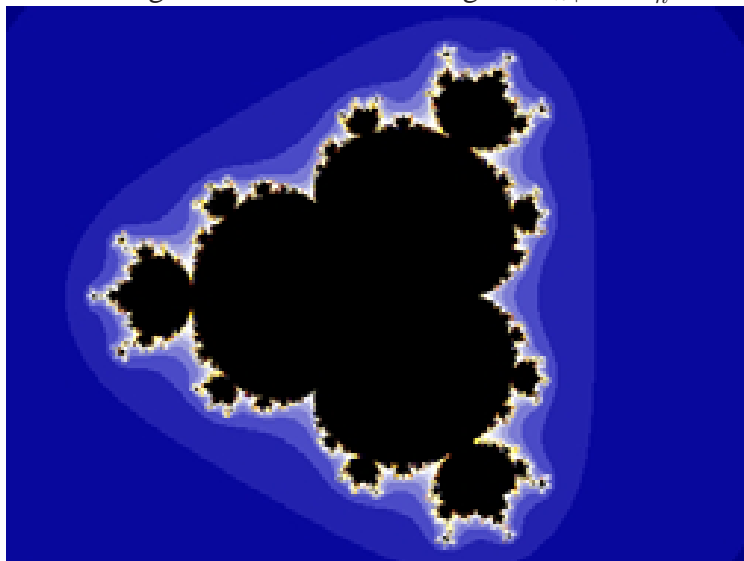


Abbildung 6.19: Mandelbrot-Menge zu $z_{n+1} = z_n^4 + c$.

7 Quaternionen

Die Notwendigkeit zur Erweiterung der Zahlenbereiche ist mit der Einführung komplexer Zahlen abgeschlossen, denn jede algebraische Gleichung mit komplexen Koeffizienten ist in ihnen lösbar. Dennoch existieren weitere Zahlenbereichserweiterungen, wie hyperkomplexe Zahlen, Oktaven und *Quaternionen*. Letztere werden wir uns hier genauer ansehen und daraus eine Verallgemeinerung der Mandelbrot-Menge entwickeln.

So wie komplexe Zahlen aus Real- und Imaginärteil bestehen, bestehen auch Quaternionen aus mehreren Teilen. Allerdings sind es hier drei (verschiedene) Imaginärteile. Die entsprechenden imaginären Einheiten heißen i (bekannt von den komplexen Zahlen), j und k . Im Einzelnen sind ihre Beziehungen untereinander wie folgt (siehe den Artikel [Zahlentheorie](#)):

$$ij = k, ik = -j$$

$$ji = -k, jk = i$$

$$ki = j, kj = -i$$

$$ii = jj = kk = -1$$

$$ijk = -1$$

Wir sehen: Sowohl i als auch j und k sind Lösungen der Gleichung $q^2 = -1$. Allerdings ist in Quaternionen die Multiplikation nicht mehr kommutativ, denn $ij = k$, wogegen $ji = -k$!

Ein Quaternion ist also eine Linearkombination der Einheitsvektoren, als Formel

$$q = x + yi + zj + wk.$$

Damit folgt für die Multiplikation zweier Quaternionen

$$\begin{aligned}
q_1 \cdot q_2 &= (x_1 + y_1i + z_1j + w_1k) \cdot (x_2 + y_2i + z_2j + w_2k) \\
&= x_1x_2 + x_1y_2i + x_1z_2j + x_1w_2k \\
&\quad + y_1ix_2 + y_1iy_2i + y_1iz_2j + y_1iw_2k \\
&\quad + z_1jx_2 + z_1jy_2i + z_1jz_2j + z_1jw_2k \\
&\quad + w_1kx_2 + w_1ky_2i + w_1kz_2j + w_1kw_2k \\
&= x_1x_2 + x_1y_2i + x_1z_2j + x_1w_2k \\
&\quad + y_1x_2i - y_1y_2 - y_1z_2k - y_1w_2j \\
&\quad + z_1x_2j - z_1y_2k - z_1z_2 + z_1w_2i \\
&\quad + w_1x_2k + w_1y_2j - w_1z_2i - w_1w_2 \\
&= (x_1x_2 - y_1y_2 - z_1z_2 - w_1w_2) \\
&\quad + i(x_1y_2 + y_1x_2 + z_1w_2 - w_1z_2) \\
&\quad + j(x_1z_2 - y_1w_2 + z_1x_2 + w_1y_2) \\
&\quad + k(x_1w_2 + y_1z_2 - z_1y_2 + w_1x_2).
\end{aligned}$$

Daraus folgt sofort

$$q^2 = x^2 - y^2 - z^2 - w^2 + i(2xy) + j(2xz) + k(2xw).$$

Nun wissen wir genug, um eine Mandelbrot- oder Julia-Menge in Quaternionen berechnen zu können. Doch es ergibt sich ein Problem: Die resultierende Menge ist vierdimensional, wie soll man sie darstellen? Dazu gibt es prinzipiell zwei Möglichkeiten: Man kann zwei Parameter festhalten und die restlichen zwei Komponenten des Quaternions ähnlich der komplexen Zahlenebene zweidimensional darstellen. Tatsächlich erhält man dann für festes $z = w = 0$ die uns bekannte Mandelbrot-Menge.

Ganz andere Bilder kann man erzeugen, wenn man nur einen Parameter festhält und einen dreidimensionalen Schnitt durch die Menge macht. Dann erhält man z. B. Abbildung [7.1](#) oder [7.2](#) ($w = 0$).

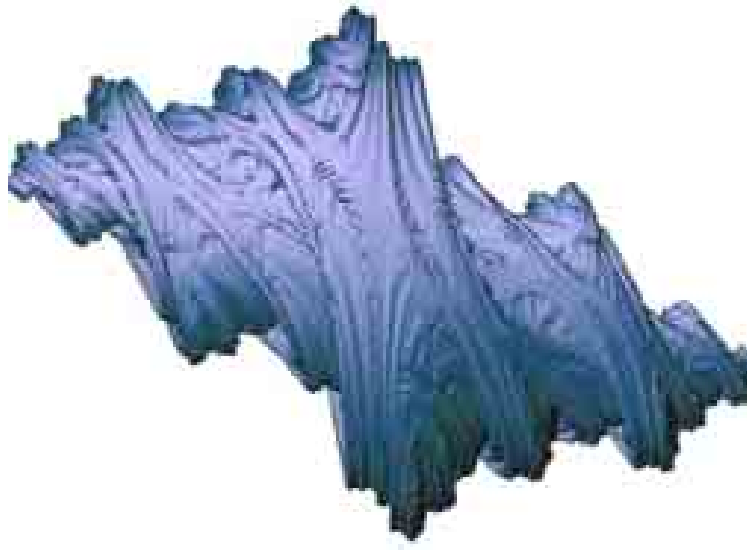


Abbildung 7.1: Quaternionen-Julia-Menge von $(-0,456426,0,510553,0,1224,-0,0924)$.

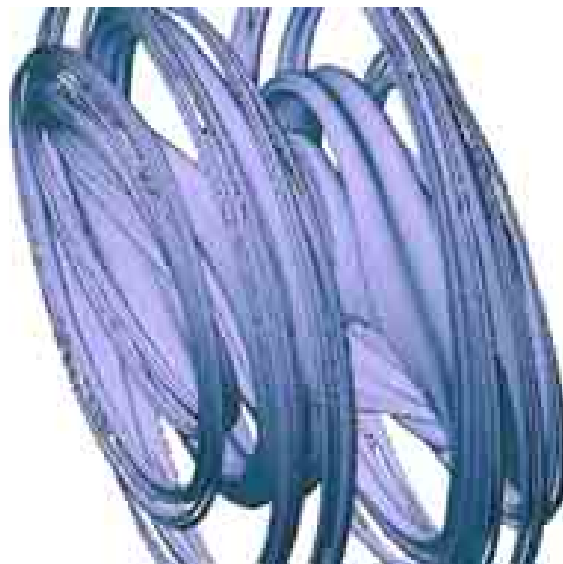


Abbildung 7.2: Quaternionen-Julia-Menge von $(0,5857,0,4385,0,0)$.

8 Eine kurze Geschichte der Farbe I

Die verschiedenen Methoden zur Farbgebung von Fraktalen zu behandeln ist eine sehr umfangreiche Aufgabe. Daher wurde sie in zwei Hälften geteilt: Diese beschäftigt sich ausschließlich mit den Mandelbrot- und Julia-Mengen, die anderen Fraktale werden in Kapitel 15 besprochen.

Dieses Kapitel führt nicht nur die Standard-Methoden vor, sondern auch weitergehende Effekte und Besonderheiten des Escape-Time-Algorithmus:

1. Initialisiere z . (Je nachdem, ob es sich um eine Mandelbrot- oder Julia-Menge handelt, verschieden: Für die Farbgebung spielt das keine Rolle — hier müssen die Fraktaltypen nicht unterschieden werden!)
2. Iteriere $f(z)$, bis ein bestimmter Betrag überschritten ist (Fall a), oder eine vorgegebene Iterationszahl erreicht wird (Fall b).
3. Je nachdem, ob der Punkt zur Menge gehört (Fall b) oder nicht (Fall a), wird der Pixel entsprechend gefärbt.
4. Verfahre so mit allen zu berechnenden Punkten.

Der Escape-Time-Algorithmus ist nicht die einzige Möglichkeit, Mandelbrot- und Julia-Mengen zu berechnen, wohl aber die intuitivste. Eine weitere Möglichkeit wird in Kapitel 15 vorgeführt. Dieses Kapitel handelt einzig und allein vom Escape-Time-Algorithmus.

Um die Verfahren besser vergleichen zu können, wird ausschließlich die Mandelbrot-Menge betrachtet. Die wahre Kunst der fraktalen Bilder besteht jedoch darin, Funktionsgleichung, Ausschnitt, Färbeverfahren und die Farben selbst optimal zu kombinieren.

Bevor wir intensiv auf Schritt 3 des Algorithmus eingehen, machen wir eine Vorbetrachtung von Schritt 2. Wie der Abbruchwert zu wählen ist, wird im Kapitel über Julia-Mengen behandelt. Wie oft man die Gleichung iterieren muss, hängt aber ganz entscheidend von der Skala ab, auf der man sich befindet: Je höher die Vergrößerung, desto öfter muss man die Gleichung iterieren, um das „wahre Bild“ zu erhalten. Abbildung 8.1 zeigt ein konkretes Beispiel mit der Mandelbrot-Menge (Zentrum des Bildes: $-0,746137 + 0,092681i$, Vergrößerung etwa 537).

Man sieht: Je größer die Iterationszahl, desto mehr Details werden sichtbar. Ganze Bereiche, die scheinbar zur Mandelbrot-Menge gehört haben, liegen nun „weit“ im äußeren Gebiet! Das sind genau die Punkte, die in der Nähe des Randes der Menge liegen, wo die Iteration chaotisch verläuft.

Wir gehen nun davon aus, dass die Menge selbst richtig berechnet wurde. Doch wie soll ich den Pixel färben? Allein mit der Information, ob der Punkt dazugehört oder nicht, kann man nicht viel anfangen. Färbt man den zugehörigen Pixel schwarz, den nicht zugehörigen weiß, ergibt sich das bekannte Bild (Abbildung 6.1).

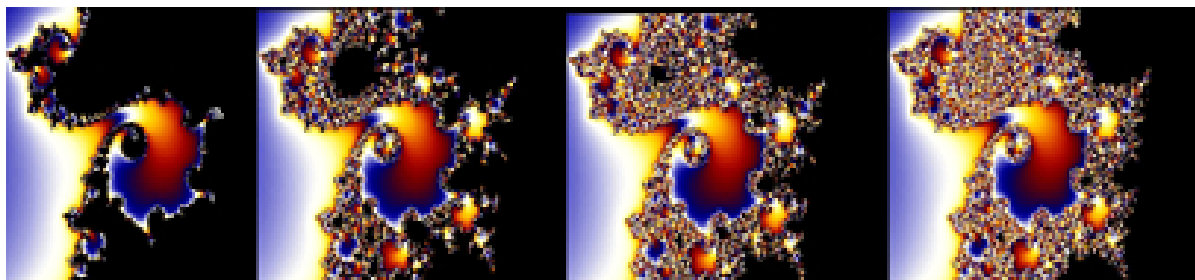


Abbildung 8.1: Iterationen: 100, 250, 500, 1000.

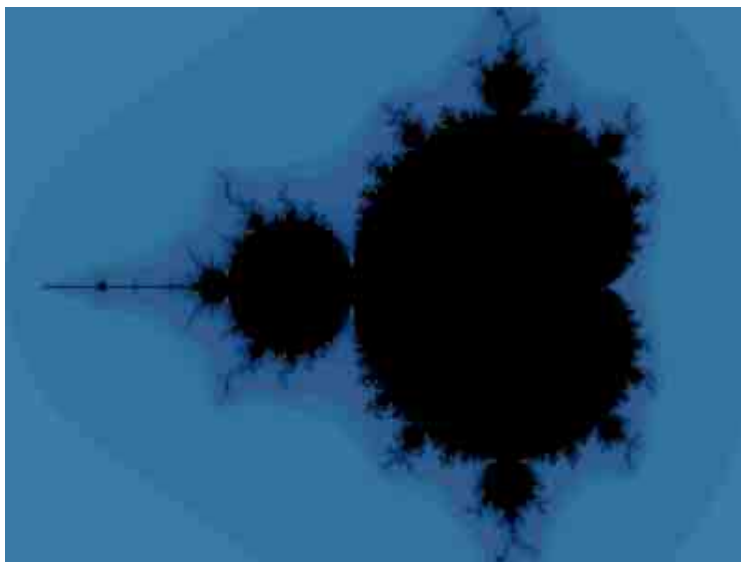


Abbildung 8.2: Färben nach der Iterationszahl.

Wie langweilig! Will man dem Ganzen etwas Farbe verleihen, muss man mehr Informationen über den Punkt einholen.

8.1 Die Iterationszahl

Das Standardverfahren besteht darin, einen Pixel, der nicht zur Menge gehört, danach zu färben, wie viele Iterationen er in der Iterationsschleife verbracht hat. Dazu bietet sich eine *Palette* an¹. Es ist klar: Je höher die Iterationszahl, desto mehr Farben kann die Palette enthalten.

8.2 Dekomposition

Bei der Dekomposition wird die komplexe Zahlenebene in gleichgroße Sektoren unterteilt. Je nachdem, in welchem Sektor z nach der Iteration liegt, wird der Punkt entsprechend gefärbt.

¹Im Übrigen verwenden alle hier erwähnten Verfahren eine Farbpalette.

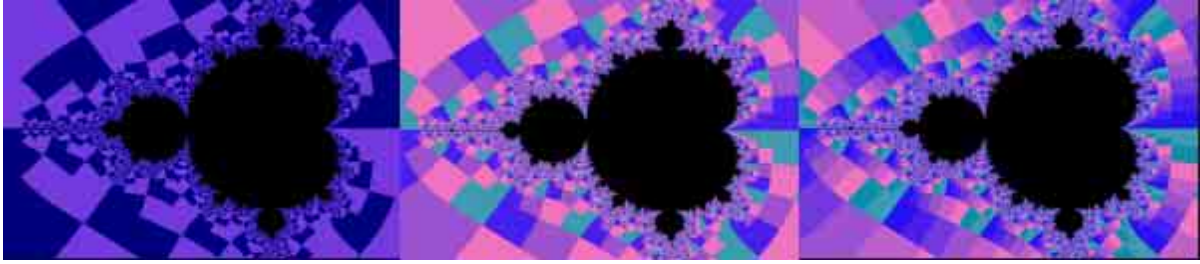


Abbildung 8.3: Dekomposition.

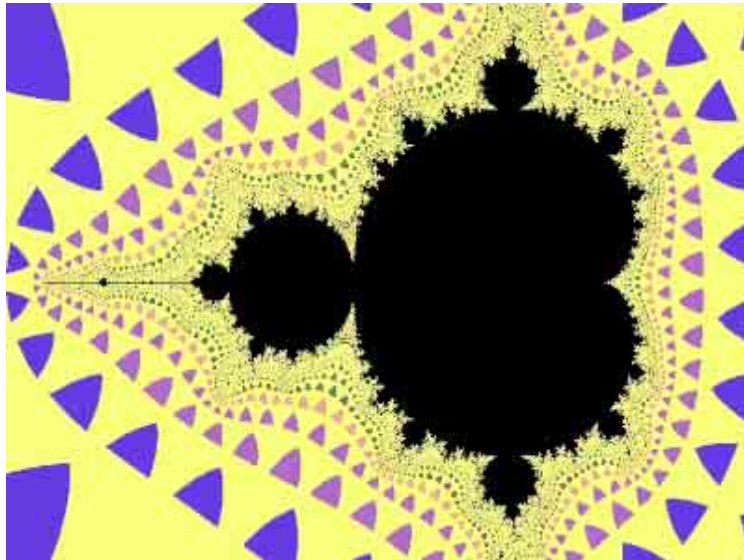


Abbildung 8.4: Biomorph.

Die Abbildung zeigt binäre Dekomposition, Dekomposition mit 4 Quadranten und 8 Sektoren.

8.3 Biomorph

Hier wird die Iteration abgebrochen, sobald der Betrag der komplexen Zahl größer ist als der Abbruchswert. Ist jedoch trotzdem der Betrag des Real- oder Imaginärteils der Zahl kleiner als der Abbruchswert, so wird nicht anhand der Iterationen gefärbt, sondern der Pixel bekommt die Biomorph-Farbe zugewiesen. Dieses Verfahren stammt von Pickover und erhielt diesen Namen, da viele Fraktale so gefärbt aussehen wie einzellige Lebewesen. Im Beispiel ist die Biomorph-Farbe gelb.

8.4 Diverse Verfahren (äußere Punkte)

Bei der Entwicklung von Färbeverfahren sind der Phantasie praktisch keine Grenzen gesetzt. Hier ist eine kleine Zusammenstellung sehr einfacher Verfahren abgebildet.

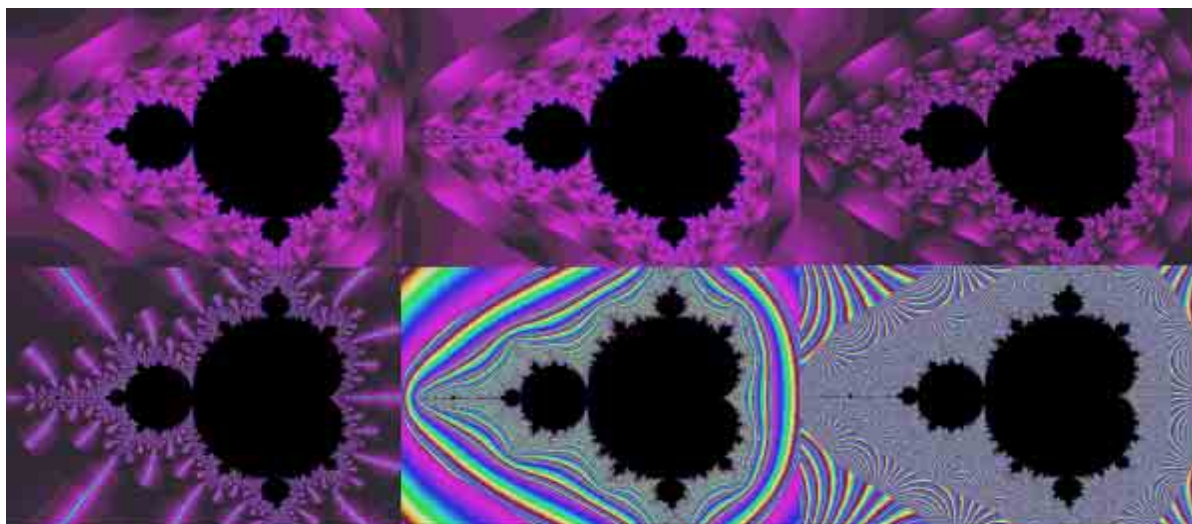


Abbildung 8.5: Diverse Verfahren.

Es ist von links nach rechts und von oben nach unten: Iterationen + $\text{Re}(z)$, Iterationen + $\text{Im}(z)$, Iterationen + $\text{Re}(z) + \text{Im}(z)$, Iterationen $\cdot \text{Re}(z)/\text{Im}(z)$, $\text{Re}(z)^2 + \text{Im}(z)^2$, $|z|$. Die Werte wurden entsprechend skaliert, um die Palette möglichst optimal abzudecken. Eine weitere Möglichkeit ist etwa $\arg(z)$.

8.5 Diverse Verfahren (innere Punkte)

Prinzipiell kann jedes der oben erwähnten Verfahren auch auf das Innere der Mandelbrot-Menge angewendet werden. Dabei ist jedoch zu beachten, dass nun die Iterationen für alle z gleich sind. Diese Formeln machen also keinen Sinn mehr. Dafür kommen neue Möglichkeiten hinzu: Beispielsweise kann man den Punkt danach färben, welche Periode sein Orbit hat. Der Phantasie sind auch hier keine Grenzen gesetzt.

8.6 3D-Darstellung

Man kann über den Escape-Time-Algorithmus auf einfache Weise aus den ebenen Bildern der Mandelbrot- und Julia-Mengen dreidimensionale Gebilde werden lassen, indem man die Anzahl der durchgeführten Iterationen als Höhe interpretiert. Das Ergebnis ist ein Gebirge, dessen oberes Plateau von der jeweiligen Menge gebildet wird.

Dieses Verfahren ist nicht zu vergleichen mit den 3D-Bildern, die man mit Hilfe von Quaternionen erzeugt: Dabei handelt es sich um 3-dimensionale Schnitte durch den 4-dimensionalen Raum, wobei hier die 2-dimensionale Ebene in den 3-dimensionalen Raum „projiziert“ wird.

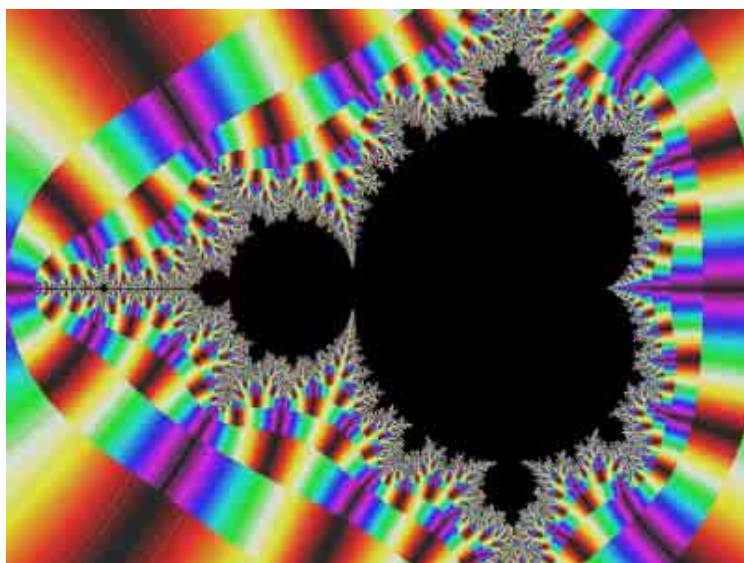


Abbildung 8.6: Gefärbt nach $\arg(z)$.



Abbildung 8.7: 3D-Darstellung der Mandelbrot-Menge.

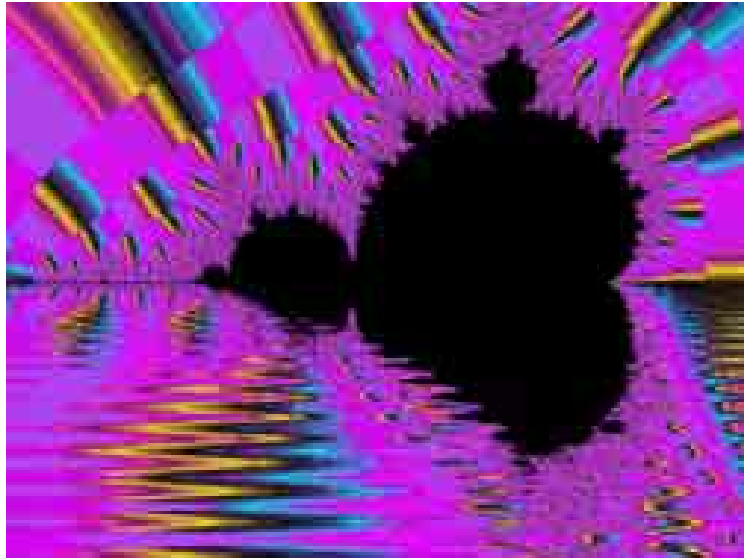


Abbildung 8.8: Dekomposition mit Lake-Transformation.

8.7 Die Lake-Transformation

Die Lake-Transformation ist eine Verzerrung des eigentlichen fraktalen Bildes, die schon während der Berechnung des Bildes vonstatten geht. Bis zu einer gewissen Höhe (der „Wassertiefe“) wird jeder Punkt vor der Iteration sinusförmig transformiert, so dass der Eindruck entsteht, dass das Bild aus einem See herausragt. In der Abbildung ist eine eindrucksvolle Kombination von Dekomposition und Lake-Transformation zu sehen.

8.8 Filter

Auch nach vollendeter Bildberechnung lassen sich noch einige interessante Effekte anbringen. Zum Sichtbarmachen der Äquipotentiallinien muss man z. B. nichts weiter tun, als das Äußere der Mandelbrot-Menge nach den Iterationen zu färben und dann einen Edge-Detection-Filter wie

$$\begin{array}{ccc} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{array}$$

auf das Bild anzuwenden.

Außerdem sollte man die Bilder immer größer generieren, als man sie später verwenden will, um durch späteres Resampling Antialiasing-Effekte zu erzielen.

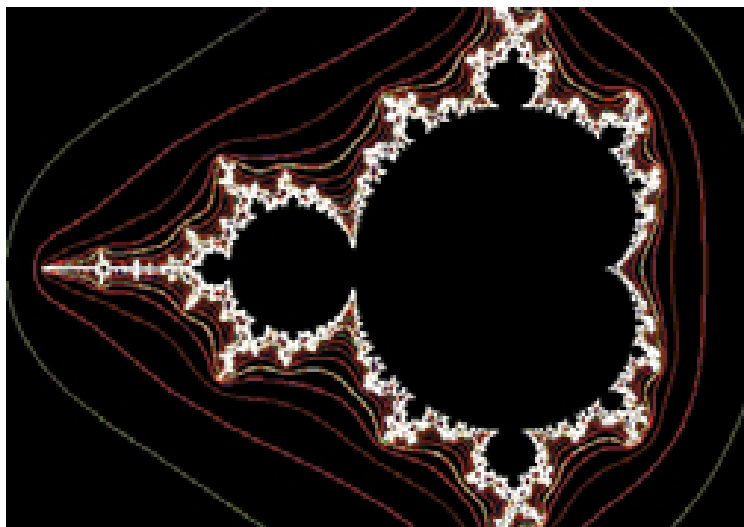


Abbildung 8.9: Äquipotentiallinien der Mandelbrot-Menge.

9 Das Curlicue-Fraktal

Sei s eine irrationale Zahl und $\theta_0 = \varphi_0 = 0$. Zu Beginn der Iteration zeichnet man eine Strecke der Länge 1 mit Winkel φ_0 zur horizontalen Achse. Dann startet die Iteration mit den Gleichungen

$$\begin{aligned}\theta_{n+1} &= \theta_n + 2\pi s \pmod{2\pi} \\ \varphi_{n+1} &= \varphi_n + \theta_n \pmod{2\pi},\end{aligned}$$

wobei nach jedem Schritt wieder eine Strecke der Länge 1 und einem Winkel φ_n zur Horizontalen gezeichnet wird.

Es zeigt sich, dass durch diese einfachen Gleichungen vielfältige Formen entstehen, die natürlich von der gewählten Zahl s abhängen. Die Abbildungen 9.1 bis 9.3 zeigen die Fraktale für den Goldenen Schnitt, die Euler-Mascheroni-Konstante und π nach den ersten 10000 Iterationen.

Elegant lässt sich diese Grafik als Punktfolge in der komplexen Zahlenebene erzeugen. Dann lautet die Iterationsgleichung für den nächsten Punkt nämlich einfach

$$z_{n+1} = z_n + e^{i\varphi_n} \quad \text{mit } z_0 = 0.$$

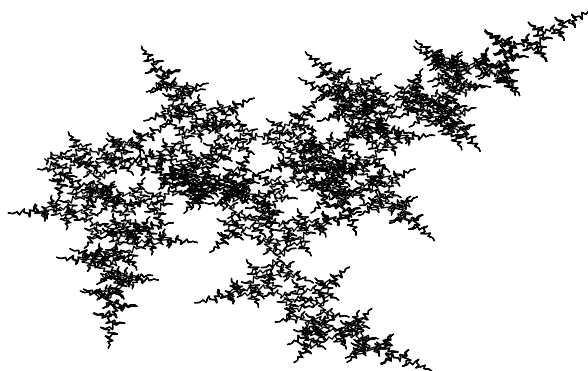


Abbildung 9.1: Der Goldene Schnitt.

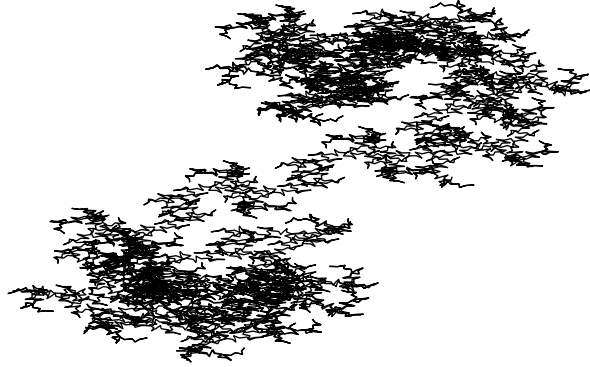


Abbildung 9.2: Die Euler-Mascheroni-Konstante.

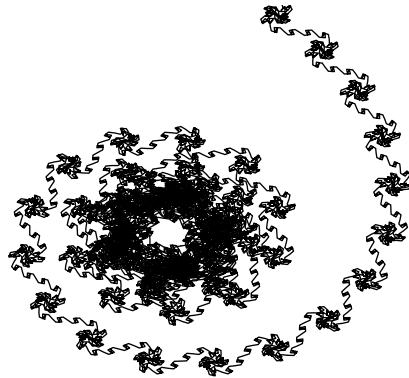


Abbildung 9.3: π .

10 Das Newton-Fraktal

10.1 Reelle Funktionen

Das Newton-Verfahren gehört wohl zu den Standard-Algorithmen zur *Nullstellenbestimmung* von Funktionen. Mit ihm können über die Iterationsgleichung

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

die Nullstellen von Funktionen, die sich analytisch nicht bestimmen lassen, angenähert werden. Zum Beispiel findet man für die Funktion

$$f(x) = x^4 - x^2 + 0,1$$

die Nullstellen

−0,941965

−0,335711

0,335711

0,941965

So weit, so gut. Doch wo kommt das Chaos ins Spiel?

Interessant wird die Sache, wenn man betrachtet, zu welcher Nullstelle der Startwert x_0 der Iteration führt. Diese kann nämlich, betrachtet man die x_n als Orbit, als *Attraktor* angesehen werden. Die Frage ist also, gegen welchen Attraktor konvergiert der Orbit von x_0 ?

In Abbildung 10.1 ist die Funktion eingezeichnet. Außerdem sind aus dem Intervall $[-1; 1]$ alle Punkte daraufhin untersucht worden, welche Nullstelle ihr Attraktor ist. Konvergiert der Orbit gegen die erste Nullstelle, so ist der Punkt durch eine rote Linie markiert. Der zweiten Nullstelle entspricht eine grüne Linie, der dritten eine blaue und der vierten eine gelbe.

Das Ergebnis ist verblüffend: Es gibt nämlich Bereiche (in der Nähe der Extremstellen), in denen offenbar alle vier Nullstellen als Attraktoren vorkommen! Ändert man x_0 nur um einen winzig kleinen Betrag, so konvergiert der Orbit gegen eine andere Nullstelle. Wir haben eine wichtige Eigenschaft dynamischer Systeme entdeckt: *Sensitive Abhängigkeit von den Anfangsbedingungen*.

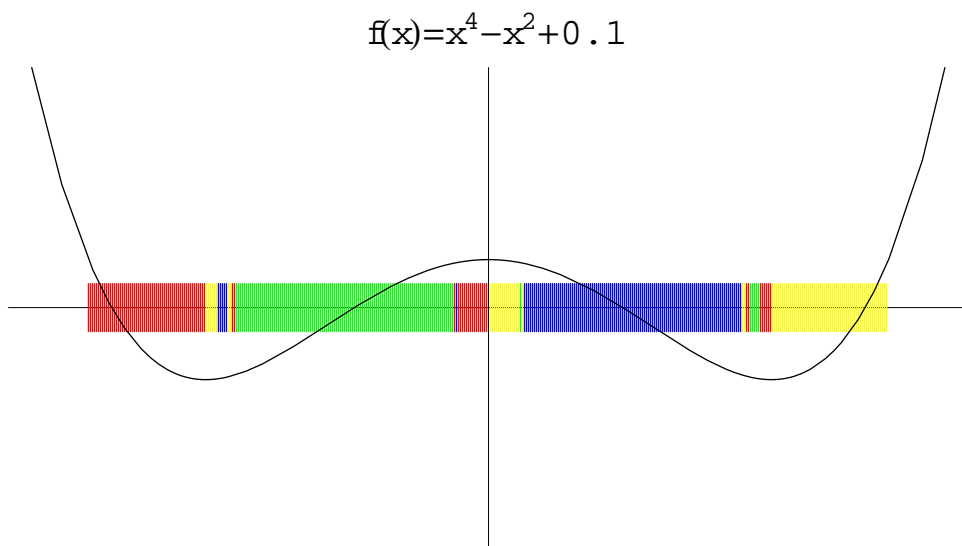


Abbildung 10.1: Beispiel zum reellen Newton-Verfahren.

10.2 Komplexe Funktionen

Das Newton-Verfahren lässt sich auf komplexe Funktionen übertragen. Wie untersuchen die *Kreisteilungsgleichung*

$$z^n = 1.$$

Nach dem *Satz von Moivre* hat diese Gleichung n Lösungen, nämlich die *n -ten Einheitswurzeln*

$$\cos\left(2\pi\frac{k}{n}\right) + i \sin\left(2\pi\frac{k}{n}\right)$$

mit k von 0 bis $n - 1$. Diese Einheitswurzeln sind sogleich Nullstellen der Funktion $f(z) = z^n - 1$.

Man kann nun, ausgehend von einer beliebigen komplexen Zahl, das Newton-Verfahren starten und sehen, welche der Einheitswurzeln Attraktor ist. Die Fälle $n = 0, 1, 2$ sind uninteressant. Aber ab $n = 3$ setzt sich unser oben betrachtetes Verhalten ins Komplexe fort. Färbt man jeden Punkt der komplexen Zahlenebene gemäß seines Attraktors, so ergibt sich [Abbildung 10.2](#).

Die Ränder der Attraktorbereiche sind besonders auffällig. Dort ist jede Nullstelle als Attraktor vertreten, und zwar unendlich tief verästelt! [Abbildung 10.3](#) zeigt einen Ausschnitt.

Diese Fraktale werden als *Newton-Fraktale* bezeichnet. Der Fall $n = 7$ ist in [Abbildung 10.4](#) zu sehen.

Ebenso, wie die Einheitswurzeln auf dem Einheitskreis dichter liegen, wird auch das Randverhalten immer komplizierter. Doch immer sind alle Nullstellen als Attraktor gegenwärtig.

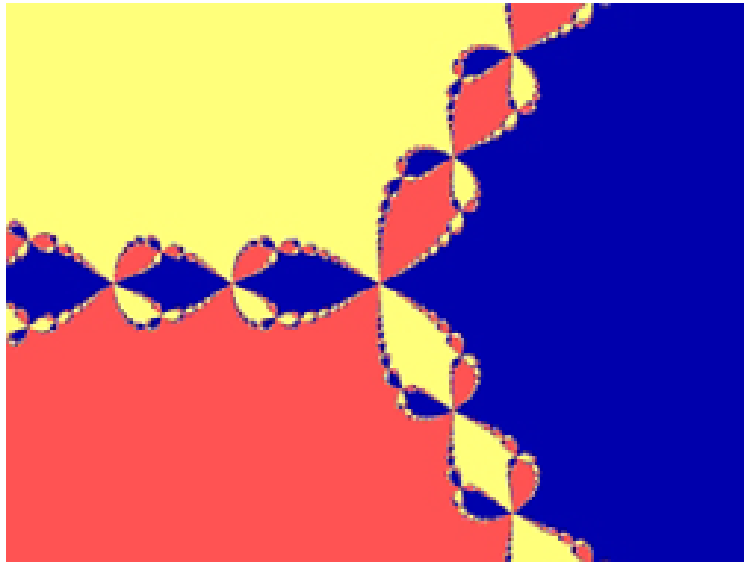


Abbildung 10.2: Newton-Fraktal für $n = 3$.

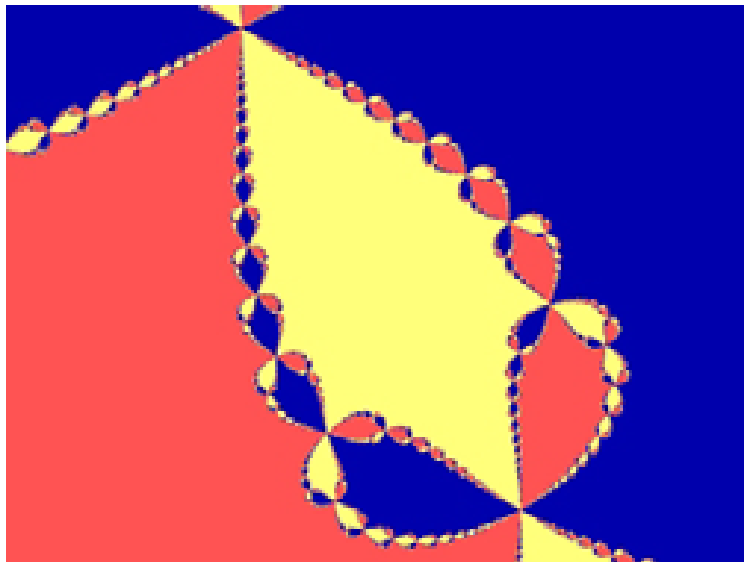


Abbildung 10.3: Newton-Fraktal für $n = 3$ (Ausschnitt).



Abbildung 10.4: Newton-Fraktal für $n = 7$.



Abbildung 10.5: Newton-Fraktal für $n = 7$ (Ausschnitt).

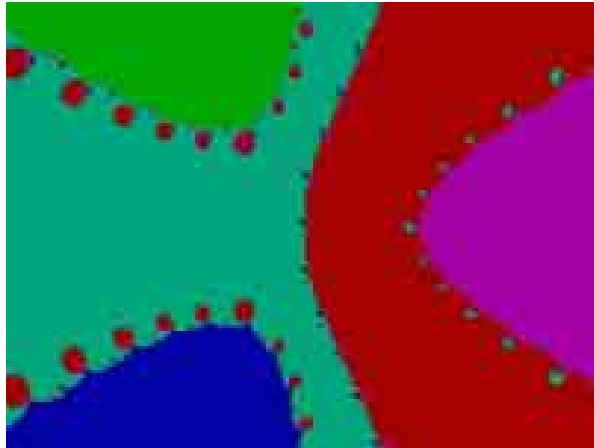


Abbildung 10.6: Newton-Fraktal für $f(z) = z^5 - z^2 + 0,1$.

Natürlich können beliebige Funktionen untersucht werden. Das Newton-Fraktal für die Funktion $f(z) = z^5 - z^2 + 0,1$ zeigt [Abbildung 10.6](#).

Man sieht, dass die Regelmäßigkeit, die durch das Auftreten der Einheitswurzeln als Nullstellen hervorgerufen wurde, verloren geht. Die grundsätzliche Eigenschaft der Randstellen, sämtliche Attraktoren unendlich verschachtelt zu beinhalten, bleibt jedoch erhalten.

11 Iterierte Funktionssysteme

Als Michael Barnsley seine *Iterierten Funktionssysteme* oder kurz *IFS* vorstellte, sorgte er weltweit für Aufsehen. Niemals zuvor konnten komplexe Strukturen in so wenig Daten kodiert werden. In diesem Kapitel werden zweidimensionale IFS betrachtet, die nur aus linearen affinen Abbildungen bestehen. Es wird sich zeigen, dass damit eine Vielzahl der uns bekannten fraktalen Kurven und vieles mehr dargestellt werden kann. Als ersten Zugang betrachten wir als einfaches Beispiel das *Sierpinski-Dreieck* (Abbildung 11.1).

In ihm sind drei verkleinerte Kopien seiner selbst enthalten. Man kann das große Sierpinski-Dreieck auf die kleineren Kopien mittels dreier *affiner Abbildungen* abbilden, und zwar (die Ecken seien $(0|0)$, $(1|0)$, $(0|1)$):

1. Das Dreieck unten links entsteht durch zentrische Streckung mit dem Streckfaktor 0,5:

$$w_1(x, y) = (0,5x, 0,5y)^T$$

2. Das Dreieck unten rechts entsteht durch dieselbe Streckung mit anschließender Parallelverschiebung in Richtung der x -Achse um 0.5:

$$w_2(x, y) = (0,5x + 0,5, 0,5y)^T$$

3. Das obere Dreieck entsteht ebenfalls aus dieser Streckung mit Parallelverschiebung nach oben:

$$w_3(x, y) = (0,5x, 0,5y + 0,5)^T$$

Das Sierpinski-Dreieck S ist somit

$$W(S) = w_1(S) \cup w_2(S) \cup w_3(S).$$

W wird *Hutchinson-Operator* genannt. In den Kopien sind wiederum Kopien des Sierpinski-Dreiecks enthalten. Diese erhält man durch iterative Anwendung der affinen Abbildungen, also $w_{a_1}(w_{a_2}(w_{a_3}(\dots(S)\dots)))$, mit $a_i \in \{1, 2, 3\}$.

Mittels des Hutchinson-Operators lässt sich das Sierpinski-Dreieck folgendermaßen definieren:

- S bleibt bei Anwendung des Hutchinson-Operators invariant, d. h. $W(S) = S$.
- S kann als Lösung der Gleichung $W(X) = X$ interpretiert werden.

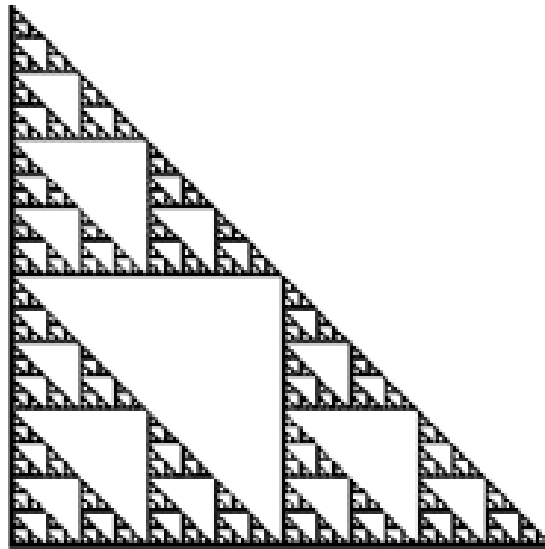


Abbildung 11.1: Das Sierpinski-Dreieck.

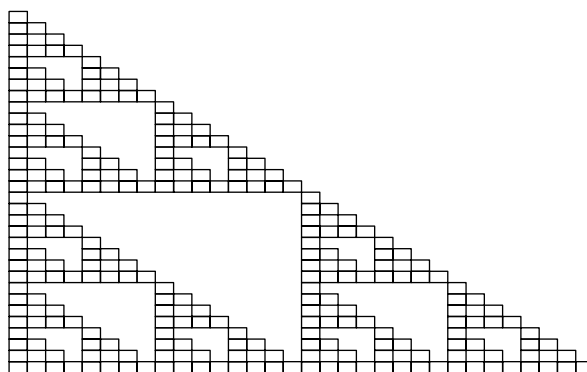


Abbildung 11.2: Die ersten vier Iterationen des Sierpinski-IFS.

Mit welchem Bild die Iteration beginnt ist dabei egal. Die Iteration von W liefert unabhängig vom Startbild S . Allgemein hat jedes IFS einen Attraktor (vorausgesetzt alle affinen Abbildungen sind Kontraktionen, s.u.), und dieser ist unabhängig vom Ausgangspunkt der Iteration.

Abbildung 11.2 zeigt die ersten vier Iterationen des Sierpinski-IFS, die auf das Quadrat $((0, 0), (1, 0), (1, 1), (0, 1))$ angewendet wurden.

Das Sierpinski-Dreieck ist also der *Attraktor* unter der Iteration dieser Abbildungen. Doch wieso konvergiert jedes beliebige Ausgangsbild gegen diesen Attraktor? Um das zu verstehen definieren wir eine *Distanzfunktion*:

$$P_1 = (x_1|y_1), P_2 = (x_2|y_2)$$

$$d(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

was nichts anderes ist als der *Satz des Pythagoras*. Weiter definieren wir:

Eine affine Abbildung heißt *Kontraktion*, wenn es eine Zahl s mit $0 \leq s < 1$ gibt, so dass für zwei beliebige Punkte P_1, P_2 die Distanz von

$$d(w(P_1), w(P_2)) \leq sd(P_1, P_2)$$

ist. s heißt Kontraktionsfaktor.

Unsere Sierpinski-Abbildungen sind klarerweise Kontraktionen, schließlich wurden sie so konstruiert. Kontraktionen haben eine extrem wichtige Eigenschaft unter Iteration, die man wie folgt einsieht:

$$d(w^2(P_1), w^2(P_2)) \leq sd(w(P_1), w(P_2)) \leq s^2d(P_1, P_2)$$

$$d(w^3(P_1), w^3(P_2)) \leq sd(w^2(P_1), w^2(P_2)) \leq s^3d(P_1, P_2)$$

...

$$d(w^n(P_1), w^n(P_2)) \leq s^nd(P_1, P_2)$$

Wenn nun $n \rightarrow \infty$ geht, so geht $s^n \rightarrow 0$, also ist

$$P_w = \lim_{n \rightarrow \infty} w^n(P)$$

unabhängig von der Wahl von P ! Darüberhinaus ist P_w der einzige Fixpunkt von w . Ein IFS, dessen affine Abbildungen sämtlich Kontraktionen sind, heißt *hyperbolisch*.

Fassen wir zusammen: Durch kombinierte Iteration der affinen Abbildungen des Sierpinski-IFS kann jede beliebige Kopie des Sierpinski-Dreiecks erzeugt werden. Die Vereinigung all dieser Kopien ist wieder das Sierpinski-Dreieck. Der Grenzpunkt bei unendlicher Iteration einer dieser Abbildungen ist vom Startwert unabhängig.

Natürlich kann man $w(S)$ praktisch nicht berechnen, schließlich ist S eine Menge aus unendlich vielen Punkten. $w(P)$ dagegen lässt sich sehr leicht berechnen. Die Frage ist nun, wie kann man das Sierpinski-Dreieck erzeugen, indem man die affinen Abbildungen nur auf Punkte anwendet?

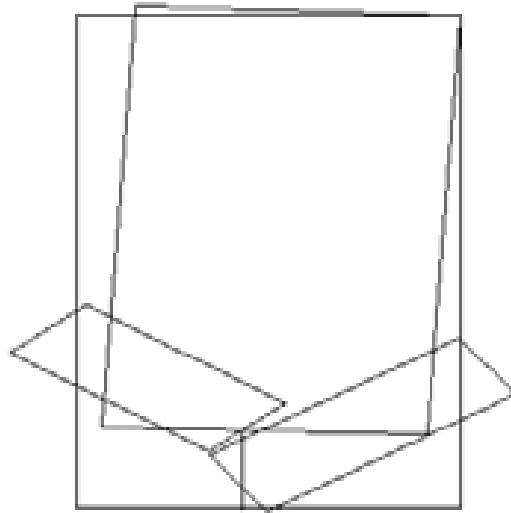


Abbildung 11.3: Die affinen Abbildungen des IFS-Farns.

Die Antwort liefert das *Chaos-Spiel*:

Mit Hilfe eines Zufallszahlengenerators wählt man einen Punkt P . Ferner wird aus den drei Abbildungen eine Abbildung ausgewählt (im Fall des Sierpinski-Dreiecks sind alle Abbildungen gleich wahrscheinlich). Diese wird auf P angewendet. Dann wird wieder eine Abbildung ausgewählt, und wieder auf $w_a(P)$ angewendet. Führt man dieses unendlich fort, so entsteht das Sierpinski-Dreieck.

Der Gedanke dabei ist folgender: Aufgrund der Sensitivität dieser Abbildungen „vergisst“ die Iteration mit der Zeit die vorangegangenen ausgeführten Iterationen. Wenn nun die die Ergebnisse des Zufallszahlengenerators unabhängig voneinander sind (das ist eine sehr wichtige Eigenschaft), wird jede mögliche Reihenfolge von Iterationen im Laufe der Zeit ausgeführt. Das Chaos-Spiel deckt also alle Abbildungen des iterierten Hutchinson-Operators ab.

Was die IFS berühmt gemacht hat, war nicht das Sierpinski-Dreieck, sondern ein Farn, der aus folgenden affinen Abbildungen besteht:

Der Stengel entsteht durch die Abbildung

$$w_1(x, y) = (0, 0, 16y)^T \quad (p_1 = 0,03)$$

Linkes Blatt:

$$w_2(x, y) = (0, 197x - 0, 226y, 0, 226x + 0, 197y + 1, 6)^T \quad (p_2 = 0,13)$$

Rechtes Blatt:

$$w_3(x, y) = (-0, 15x + 0, 283y, 0, 26x + 0, 237y + 0, 44)^T \quad (p_3 = 0,11)$$

Oberes Blatt:

$$w_4(x, y) = (0, 849x + 0, 037y, -0, 037x + 0, 849y + 1, 6)^T \quad (p_4 = 0,73)$$

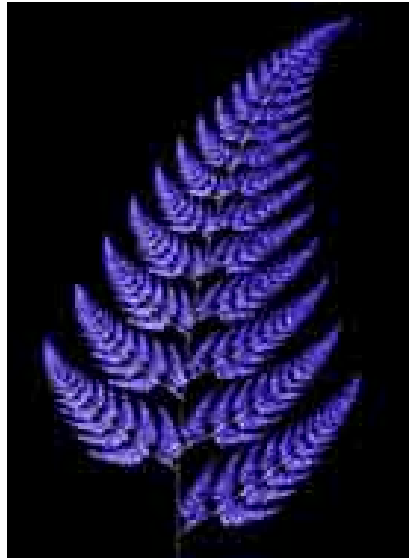


Abbildung 11.4: Der IFS-Farn.

Verwendet man die Matrizen-Darstellung der Abbildungen in der Form

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix},$$

so lassen sich sämtliche Daten zur Erzeugung des Farns einschließlich der Wahrscheinlichkeiten in einer Matrix der Form

a	b	c	d	e	f	p
0	0	0	0,16	0	0	0,03
0,197	-0,226	0,226	0,197	0	1,6	0,13
-0,15	0,283	0,26	0,237	0	0,44	0,11
0,849	0,037	-0,037	0,849	0	1,6	0,73

darstellen. Das resultierende Fraktal ist in [Abbildung 11.4](#) dargestellt.

Natürlich können nur selbstähnliche Gebilde mit Hilfe von IFS erzeugt werden. Die [Abbildungen 11.5 bis 11.7](#) zeigen einige weitere bemerkenswerte IFS-Fraktale.

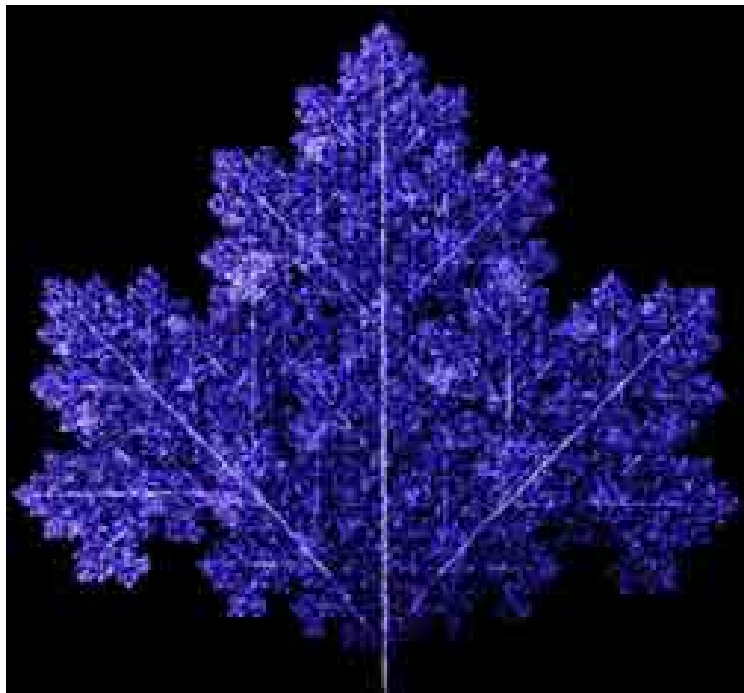


Abbildung 11.5: Ein IFS-Blatt.



Abbildung 11.6: Ein IFS-Baum.

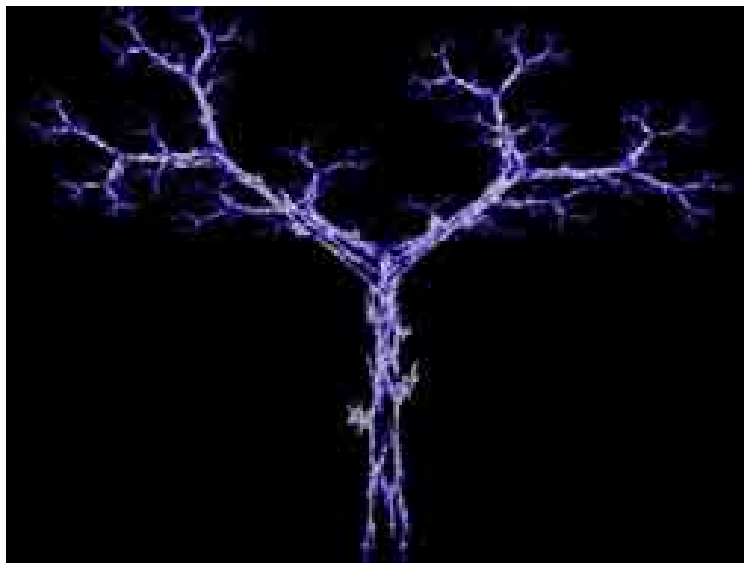


Abbildung 11.7: Noch ein IFS-Baum.

12 Fraktale Kurven

Wie lang ist die Küste Großbritanniens? Bevor Benoit Mandelbrot sich diese Frage stellte, schien die Antwort klar. Das heißt, so klar nun auch wieder nicht, schließlich findet man in jedem Nachschlagewerk einen (nicht nur geringfügig) anderen Wert. Der Grund dafür ist einfach: Beim Abstechen der Küste mit dem Zirkel wurden Karten mit verschiedenem Maßstab verwendet. Je mehr Details eine Karte zeigt, desto länger wird die Küste. Wählt man den Maßstab immer kleiner, so ist sie im Grenzfall auf Molekül- und Atomebene unendlich lang! Es ist ein Kuriosum, dass die Grenze zwischen zwei Ländern von diesen als verschieden lang angegeben wird.

Wie kann man sich helfen? Was ist „länger“, die Küste Großbritanniens oder die deutsche Nordseeküste? Eine Möglichkeit, die Längen dieser Kurven zu vergleichen, liefert die fraktale Dimension. Denn je größer sie ist, desto *flächenfüllender* ist die Kurve. Als Beispiele betrachten wir die bekanntesten fraktalen Kurven in der Reihenfolge aufsteigender Hausdorff-Dimension. Weitere fraktale Kurven findet man im Kapitel über Lindenmayer-Systeme.

Die fraktalen Kurven dieses Kapitels werden folgendermaßen konstruiert: Man geht von einem *Initiator*, für gewöhnlich der Einheitsstrecke, aus. Dann ersetzt man diese durch den *Generator*, und zwar so, dass Anfang und Ende der Generatorkurve an Anfang und Ende des Initiators eingepasst werden. Für gewöhnlich enthält der Generator kleinere Kopien des Initiators, welche in der nächsten Iteration durch den Generator ersetzt werden usw. Die so erzeugte Kurve zeichnet sich durch *Selbstähnlichkeit* und *Skaleninvarianz* aus. Das bedeutet, dass man, wenn man einen beliebigen Ausschnitt der Kurve betrachtet, nicht sagen kann, in welcher Größenordnung man sich befindet, da sich die Details bis ins unendlich kleine fortsetzen (denn genau so wurde die Kurve ja konstruiert).

Hinweis: Damit die Kurve wirklich skaleninvariant ist, muss sie nicht nur im Kleinen, sondern auch im Großen fortgesetzt werden. Täte sie das nicht, so würde beispielsweise eine Kurve, deren Initiator $[0; 1]$ ist, in verkleinerter Form nicht mehr wie sie selbst aussehen, da plötzlich Anfang und Ende erkennbar sind. Daher muss die Kurve bei der Konstruktion auch über das Initiatorintervall $[0; 1]$ hinaus extrapoliert werden.

Die Berechnung der *Hausdorff-Dimension* dieser Kurven gestaltet sich (sofern es keine Selbstschnitte gibt) besonders einfach, da sie sich auf die Berechnung der *Ähnlichkeitsdimension* reduziert. Wird beispielsweise der Initiator durch n Kopien seiner selbst von $1/r$ seiner Länge ersetzt, so beträgt

$$D_H = \log_r n.$$

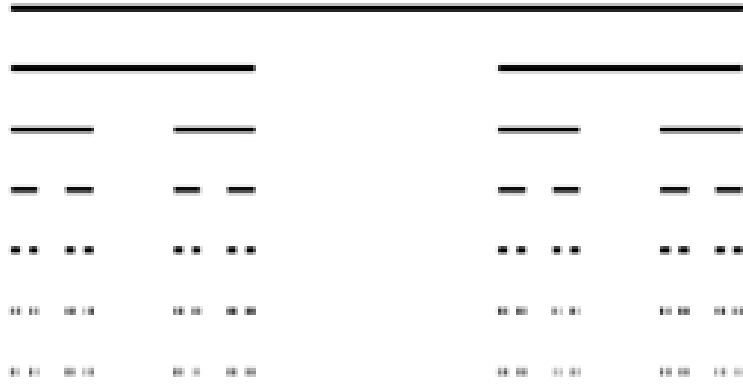


Abbildung 12.1: Der Cantor-Staub.

12.1 Der Cantor-Staub

Der *Cantor-Staub* entsteht folgendermaßen: Aus dem Einheitsintervall $[0; 1]$ wird das mittlere Drittel $]1/3; 2/3[$ entfernt. Im nächsten Schritt wird aus den beiden verbliebenen Strecken das mittlere Drittel entfernt usw. Die Grenzfigur hat schließlich keine Länge mehr (oder technisch: Lebesgue-Maß 0), da die Gesamtlänge der übrig bleibenden Intervalle nach dem n -ten Schritt $(2/3)^n$ beträgt, was mit wachsendem n verschwindet. Der Cantor-Staub ist also, wie der Name vermuten lässt, eine *Cantor-Menge*.

Dennoch enthält er überabzählbar viele Punkte. Betrachten wir nämlich in [Abbildung 12.1](#) die zur Entstehung des Cantor-Staubes führenden Schritte, so können wir für jeden Punkt des Cantor-Staubes in jedem Schritt fragen, ob sich der Punkt des Cantor-Staubes links oder rechts vom jeweils ihm an nächsten gelegenen entfernten Intervall befindet. Schreiben wir für „links“ eine 0 und für „rechts“ eine 1, so erhalten wir eine unendliche Folge dieser Ziffern, und diese legt den Punkt im Cantor-Staub fest. Umgekehrt wird durch jede solche Ziffernfolge ein Punkt im Cantorstaub angegeben. Es wird aber auch jede reelle Zahl im Intervall $[0; 1]$ durch eine solche Folge dargestellt, wenn man das Dualsystem verwendet. Da es überabzählbar viele reelle Zahlen im Intervall $[0; 1]$ gibt, muss es auch überabzählbar viele Punkte im Cantor-Staub geben. Man kann sich leicht überlegen, dass der Cantor-Staub die reellen Zahlen enthält, die in ternärer Darstellung keine 1 enthalten.

Seine fraktale Dimension liegt bei $D_H = \log_3 2 \approx 0,63$. Wegen $D_T = 0$ ist $D_H > D_T$ und somit der Cantor-Staub tatsächlich ein Fraktal. Seine Dimension liegt zwischen der eines Punktes und der einer Geraden.

Die Entstehung des Cantor-Staubes kann man sich auch anders entstanden denken. Man nimmt an, eine Masse vom Betrag 1 ist gleichmäßig auf das Einheitsintervall verteilt. Dann fließt diese Masse im ersten Schritt aus dem mittleren Drittel in die beiden anderen Drittel und vergrößert dort die Dichte (sie „*gerinnt*“); in der Mitte bleibt eine Lücke oder *Trema* zurück. Dann zerteilt sich die Masse der beiden Drittel weiter auf. Im Grenzfall ist die Masse auf einzelne Punkte mit unendlicher Dichte verteilt. Trägt man die Masse zwischen 0 und x über x auf, so erhält man die *Teufelstreppe*.

Diese Kurve hat einige interessante Eigenschaften: Sie ist stetig aber nicht differenzierbar!

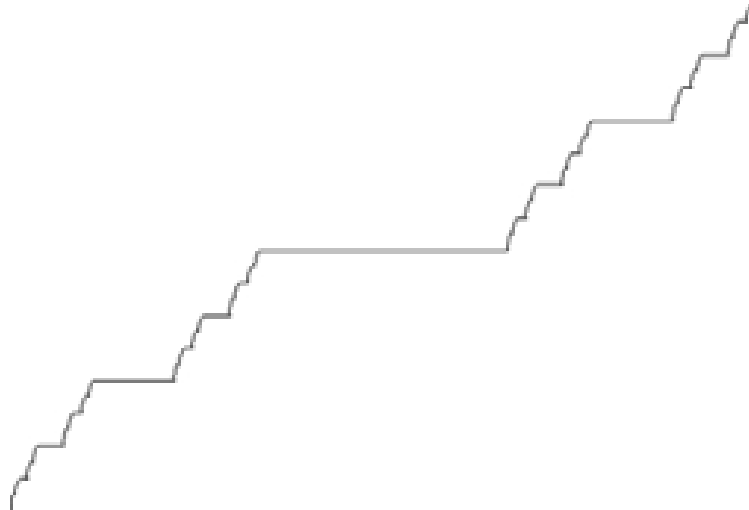


Abbildung 12.2: Die Teufelstreppe.

Ihre Höhe wächst über unendlich viele, unendlich kleine Sprungstellen von 0 bis 1 an. Ihr Wert ist jedoch fast überall konstant (nämlich bei den Tremata).

Die Teufelstreppe ist nicht skaleninvariant, denn z. B. für x aus $]1/3; 2/3[$ ist sie einfach eine Strecke. Darüber hinaus ist sie laut unserer Definition kein Fraktal. Denn man kann sie durch 3 Kreise vom Durchmesser $1/3$, 9 Kreise vom Durchmesser $1/9$ usw. abdecken, man erhält also $D_H = 1$. Wegen $D_H = D_T$ ist die Teufelstreppe keine fraktale Kurve!

Durch den oben beschriebenen Gerinnungsprozess lässt sich Cantor-Staub beliebiger Dimension zwischen 0 und 1 kreieren. Dazu muss lediglich die Anzahl der Teilstücke oder die Länge der Tremata variiert werden. Denn die fraktale Dimension ergibt sich ja aus $D_H = \log_r n$. Der *Cantor-Vorhang* zeigt auf der vertikalen Achse eine Variation von D_H von 0 bis 1 bei festem $n = 2$.

Leider ist aufgrund der Zerstreutheit der Cantor-Menge und der begrenzten Punktzahl des Bildes die Cantor-Menge für Dimensionen nahe 0 nur schlecht zu erkennen.

12.2 Die Koch-Kurve

Die Konstruktion der *Koch-Kurve* baut auf der Konstruktion der Cantor-Menge auf. Nur errichtet man noch zusätzlich auf dem herausgenommenen Geradenstück ein gleichseitiges Dreieck. In Abbildung 12.4 sind der Generator und die Grenzkurve dargestellt.

Diese Kurve ist schon recht gut mit einer Küstenlinie zu vergleichen. Natürlich ist sie viel zu regelmäßig, um realistisch zu sein. Ihre fraktale Dimension liegt jedoch mit $D_H = \log_3 4 \approx 1,26$ im Bereich realer Küstenlinien. Wegen $D_T = 1$ ist auch sie ein Fraktal (man braucht nur einen Punkt, um sie in zwei Hälften zu teilen).

Die Länge der Koch-Kurve ist tatsächlich unendlich. Denn die Folge der Länge der Iterationen ist $1, 4 \cdot 1/3, 16 \cdot 1/9, \dots$ oder allgemein $(4/3)^n$, was mit n gegen unendlich ebenfalls gegen unendlich geht.

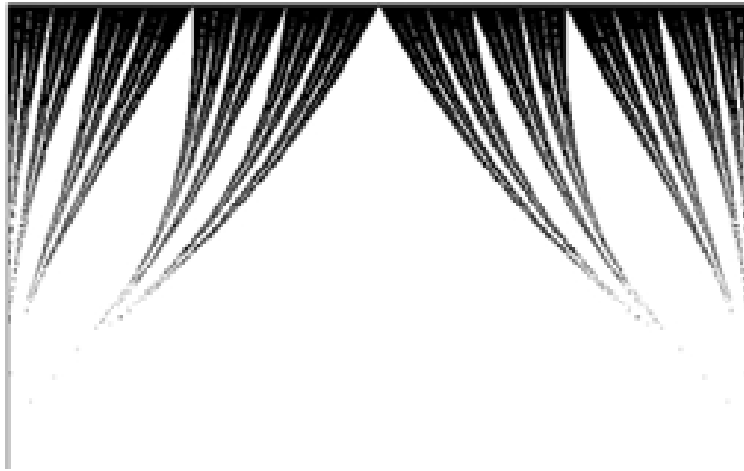


Abbildung 12.3: Der Cantor-Vorhang.

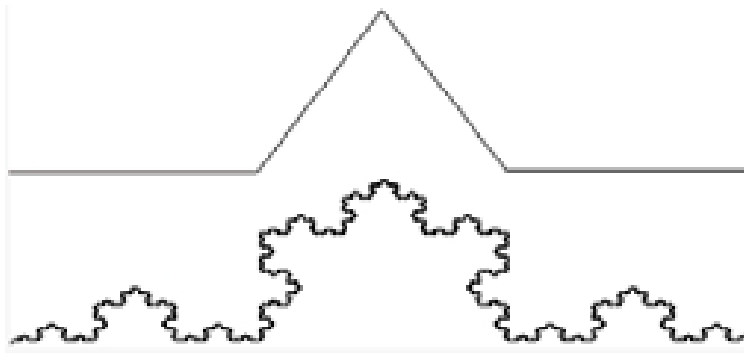


Abbildung 12.4: Die Koch-Kurve.

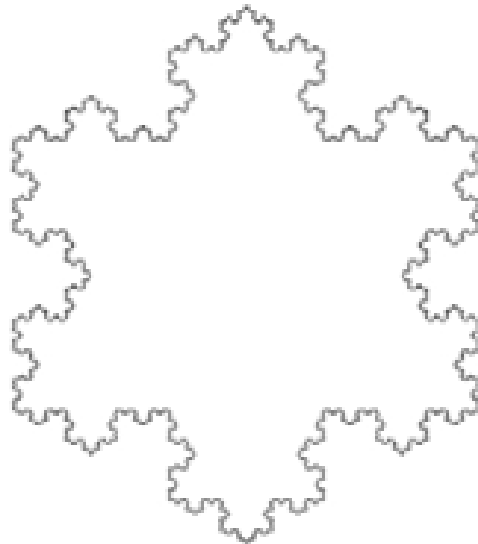


Abbildung 12.5: Die Schneeflockenkurve.

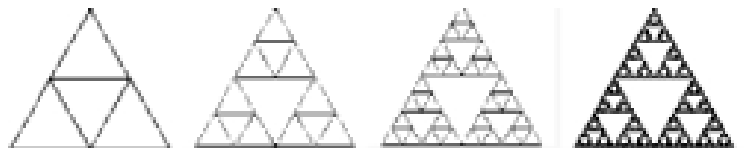


Abbildung 12.6: Die Entstehung des Sierpinski-Dreiecks.

Die Fläche, die die Koch-Kurve mit der Einheitsstrecke einschließt, ist hingegen endlich. Sie gehorcht der Folge $1/(3^2 \cdot 4) \cdot \sqrt{3}$, $1/(3^2 \cdot 4) \cdot \sqrt{3} + 4 \cdot 1/(9^2 \cdot 4) \cdot \sqrt{3}$, ... mit dem Grenzwert $1/20 \cdot \sqrt{3}$.

Die Koch-Kurve ist stetig, doch nirgends differenzierbar. Sie ist eine jener *Monsterkurven*, die die Mathematiker um die letzte Jahrhundertwende noch als nichtsnutziges Spielzeug abgetan hatten. Die Vorstellung, praktisch alle stetigen Kurven seien auch differenzierbar, wurde durch sie zerstört. Um so erstaunlicher, dass gerade diese Monsterkurven jetzt so erfolgreich zur Modellierung der Natur eingesetzt werden!

Benutzt man als Initiator nicht die Einheitsstrecke, sondern ein gleichseitiges Dreieck, so ergibt sich die *Koch'sche Schneeflockenkurve*.

12.3 Das Sierpinski-Dreieck

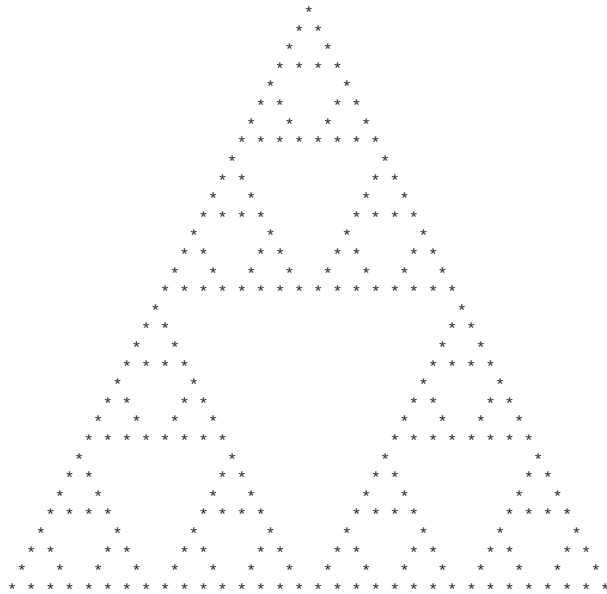
Man nehme ein gleichseitiges Dreieck und entferne das die Seitenmittelpunkte verbindende Dreieck. Mit den übriggebliebenen drei Dreiecken verfähre man genauso. Führt man diesen Prozess unendlich oft fort, so entsteht das *Sierpinski-Dreieck*.

Die Anzahl der übriggebliebenen Dreiecke nach der n -ten Iteration ist $N_n = 3^n$, ihre Fläche $A_n = (3/4)^n$. Die Länge der Dreiecksseiten berechnet sich nach $L_n = 2^{-n}$. Die Grenzfigur hat also verschwindende Fläche und dennoch die topologische Dimension 2. Die fraktale Di-

mension berechnet sich nach

$$D_H = -\lim \frac{\ln N_n}{\ln A_n} = -\lim \frac{\ln 3^n}{\ln 2^{-n}} = \log_2 3 \approx 1,58.$$

Das Sierpinski-Dreieck erhält man auch aus dem Boole'schen Zellularautomaten mit Regel 90 und einer einzigen Zelle im Zustand 1. Diese Regel setzt die Zelle $(x; y)$ auf 0, wenn $(x - 1; y - 1) = (x + 1; y - 1)$ ist. Ansonsten ist die Zelle 1.



Auch das Pascal'sche Dreieck enthält das Sierpinski-Dreieck, und zwar wenn man es mod 2 darstellt, d. h. für alle geraden Zahlen eine 0 schreibt und für alle ungeraden Zahlen eine 1.

1	1
1 1	1 1
1 2 1	1 0 1
1 3 3 1	1 1 1 1
1 4 6 4 1	1 0 0 0 1
1 5 10 10 5 1	1 1 0 0 1 1
1 6 15 20 15 6 1	1 0 1 0 1 0 1
1 7 21 35 35 21 7 1	1 1 1 1 1 1 1 1

Als letzte Erscheinungsform sei das Sierpinski-Dreieck als Julia-Menge erwähnt, mit den Formeln $(x = \text{Re}(z); y = \text{Im}(z))$

$$z = \begin{cases} (2 \cdot x, 2 \cdot y - 1) & \text{für } y > 0,5 \\ (2 \cdot x - 1, 2 \cdot y) & \text{für } y = 0,5 \text{ und } x > 0,5 \\ (2 \cdot x, 2 \cdot y) & \text{für } y = 0,5 \text{ und } x = 0,5 \end{cases}$$

Eine weitere Möglichkeit der Generierung des Sierpinski-Dreiecks ist uns im Kapitel über IFS begegnet.

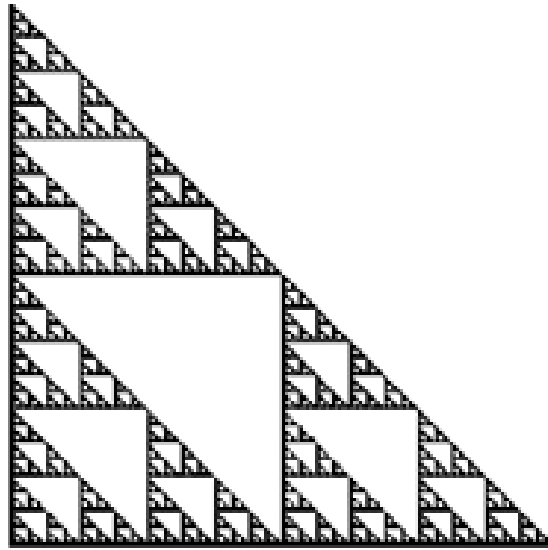


Abbildung 12.7: Das Sierpinski-Dreieck als Julia-Menge.

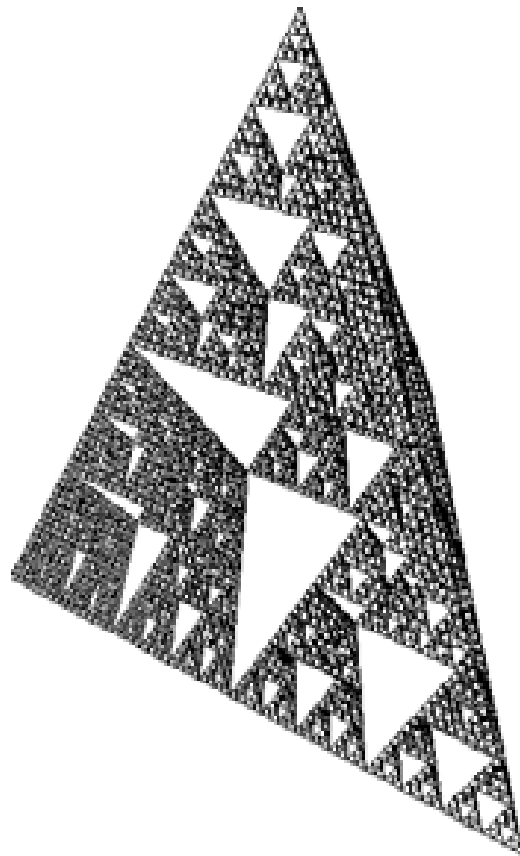


Abbildung 12.8: Die Sierpinski-Pyramide.

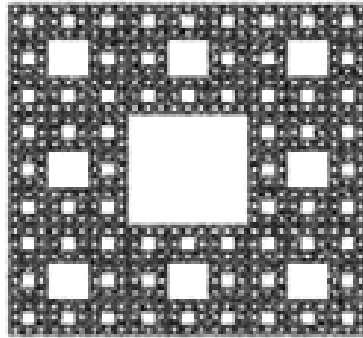


Abbildung 12.9: Der Menger-Teppich.

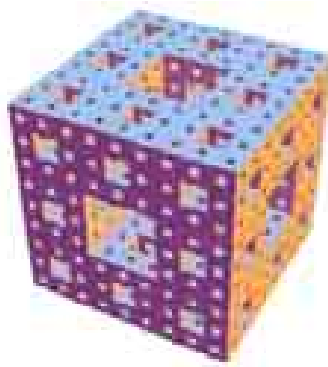


Abbildung 12.10: Der Menger-Schwamm.

Die Idee des Sierpinski-Dreiecks lässt sich auf drei Dimensionen erweitern. Entfernt man aus einer gleichseitigen Pyramide den Körper, der durch Verbindung der Kantenmittelpunkte entsteht, und führt dieses in gleicher Weise fort, so entsteht [Abbildung 12.8](#).

Eine weitere Abwandlung besteht darin, nicht aus einem Dreieck Dreiecke zu entfernen, sondern aus einem Quadrat Quadrate (aus der Mitte mit einem Drittel der Seitenlänge). Das entsprechende Fraktal heißt *Menger-Teppich* ($D_H = \log_3 8 \approx 1,89$), sein 3D-Analogon Menger-Würfel oder *Menger-Schwamm* ($D_H = \log_3 20 \approx 2,73$).

12.4 Die Drachenkurve

Der Initiator der *Drachenkurve* ([Abbildung 12.11](#)) ist das Intervall $[0; 1]$. Der Generator jedoch (1.Bild) wird abwechselnd rechts und links angelegt. Die Grafik zeigt noch Iterationsstufe 2 und 3.

Die Drachenkurve ist selbstverständlich selbstähnlich. Darüberhinaus besteht sie aus zwei Segmenten, die untereinander, aber nicht zum Drachen selbst, ähnlich sind.

Mit zunehmender Iterationsstufe wird die Kurve selbst nur noch schwer zu erkennen, vielmehr nimmt man nur noch ihre Umrisse wahr. Die Kurve wird im Grenzfall *flächenfüllend*. Das bestätigt auch die fraktale Dimension. Man erhält nämlich $D_H = \log_{\sqrt{2}} 2 = 2$.

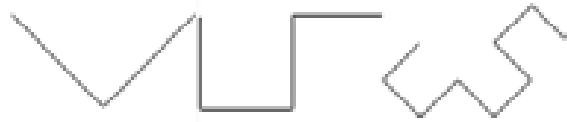


Abbildung 12.11: Die Entwicklung der Drachenkurve.

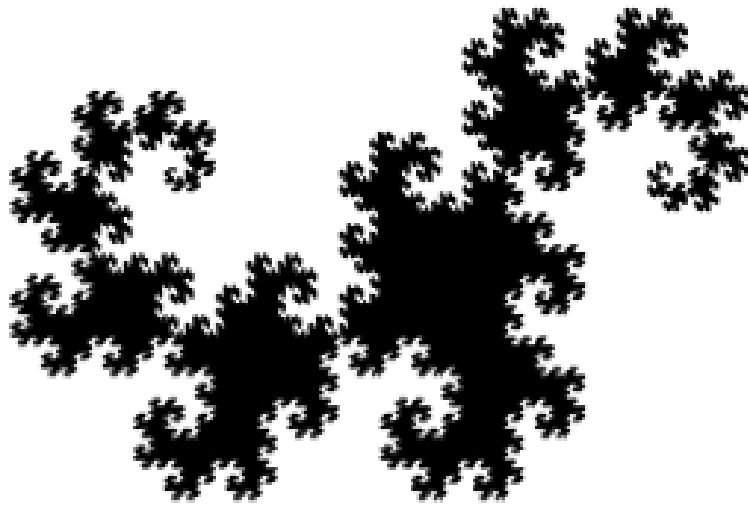


Abbildung 12.12: Die Drachenkurve.

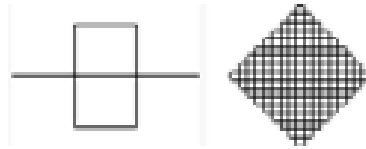


Abbildung 12.13: Die Peano-Kurve.

12.5 Die Peano-Kurve

Als letztes Beispiel soll die *Peano-Kurve* (Abbildung 12.13) behandelt werden. Sie ist vom ästhetischen Standpunkt her zwar eher langweilig, dafür aber von enormer mathematischer Bedeutung (siehe Kapitel 1).

Der Initiator ist das Einheitsintervall, der Generator zwei auf dem mittleren Drittel errichtete Quadrate (Bild 1). Die Grafik zeigt die 3. Iteration.

Die fraktale Dimension ist $D_H = \log_3 9 = 2$. Die Kurve ist also tatsächlich flächenfüllend.

13 Lindenmayer-Systeme

Lindenmayer- oder kurz *L-Systeme* wurden von Aristid Lindenmayer zur Modellierung biologischen Wachstums eingeführt. Sie bestehen aus

- einem *Alphabet* V , einer endlichen Menge aus formalen Symbolen,
- einem *Axiom* (Initiator) w , einer Zeichenkette (einem *Wort*) von Symbolen aus V (V^* ist die Menge aller Wörter über V),
- einer *Produktionsregel* (Ersetzungsregel) $P \subset V \times V^*$, einer Abbildung eines Symbols $a \in V$ in ein Wort $w \in V^*$.

Die Länge einer Wortes $|w|$ ist die Anzahl seiner Symbole.

Das Interessante an L-Systemen ist die *Evolution*: die Folge $\{g_n\}$, $n = 0, 1, 2, 3, \dots$, wobei g_n ein Wort aus V^* ist, das aus g_{n-1} durch Anwendung der Produktionsregeln entsteht. Die erste Generation g_0 ist das Axiom w .

Natürlich müssen so viele Produktionsregeln existieren, wie Symbole in V sind. Zur Vereinfachung wird per Konvention ein Symbol, für das keine explizite Regel angegeben wird, auf sich selbst abgebildet. Ein solches System heißt *deterministisch* und wird mit DL bezeichnet. Existiert für ein Symbol mehr als eine Produktionsregel, und verwendet man für die Anwendung einen Zufallsgenerator, so heißt das System *stochastisch*. Entscheidet über die Anwendung der Ersetzungsregel nicht nur das Symbol selbst, sondern auch seine Nachbarn, so heißt das System *kontext-sensitiv* (IL), andernfalls *kontext-frei* (OL). In diesem Kapitel beschäftigen wir uns ausschließlich mit DOL-Systemen.

13.1 Das Fibonacci-L-System

Zum Einstieg werden wir die bekannte Zeichenkette zur Bestimmung der Kaninchen-Konstante mit Hilfe eines L-Systems formalisieren:

$$\begin{aligned}V &= \{a, b\} \\w &= a \\P &= \{a \mapsto b, b \mapsto ba\}\end{aligned}$$

Die ersten Generationen sind:

a
 b
 ba

bab
 $babba$
 $babbabab$
 $babbababbabba$
 $babbababbabbababbabab$

Wir entdecken das bekannte Muster $g_n = g_{n-1}g_{n-2}$, das nun per vollständiger Induktion bewiesen werden soll.

Induktionsanfang: $g_0 = a, g_1 = b, g_2 = ba = g_1g_0$

Induktionsschritt: g_{n+1} entsteht aus g_n durch Anwendung der Produktionsregeln. g_n ist aber laut Induktionsvoraussetzung $g_{n-1}g_{n-2}$. Wendet man hierrauf die Regeln an, erhält man $g_n g_{n-1}$. \square

13.2 Das Thue-Morse-L-System

Das Thue-Morse-L-System ist durch

$$\begin{aligned}
 V &= \{a, b\} \\
 w &= a \\
 P &= \{a \mapsto ab, b \mapsto ba\}
 \end{aligned}$$

definiert. Die ersten Generationen lauten

a
 ab
 $abba$
 $abbabaab$
 $abbabaabbaababba$

Wie man leicht sieht, ist $|g_n| = 2^n$. Außerdem ist

$$g_{n+1} = g_n S(g_n),$$

wobei S die Spiegelfunktion ist, d. h. jedes a auf ein b und jedes b auf ein a abbildet.

Beweis:

Induktionsanfang: $g_0 = a, g_1 = ab = g_0 S(g_0)$

Induktionsschritt: g_{n+1} erhält man aus g_n durch Anwendung der Produktionsregeln. Laut Induktionsvoraussetzung ist $g_n = g_{n-1} S(g_{n-1})$. Wendet man hierrauf die Regeln an, erhält man $g_n S(g_n)$. \square

Die Grenzfolge g_∞ heißt *Thue-Morse-Folge*. Sie hat einige bemerkenswerte Eigenschaften, von denen hier einige vorgestellt werden sollen. Codiert man die Folge so, dass man statt eines a eine 1 schreibt, und anstatt eines b eine -1 , so kann die Spiegelfunktion als Multiplikation mit -1 aufgefasst werden. Dann ist

$$g_\infty = c_0 c_1 c_2 c_3 \dots \quad \text{mit } c_n = \pm 1.$$

Ferner erhält man

$$g_n = c_0 c_1 c_2 \dots c_{2^n - 1}$$

und

$$\begin{aligned} g_{n+1} &= c_0 c_1 c_2 \dots c_{2^n - 1} (-c_0) (-c_1) (-c_2) \dots (-c_{2^n - 1}) \\ &\Rightarrow c_{2^n + i} = -c_i \quad \text{mit } 0 \leq i \leq 2^n - 1 \end{aligned}$$

Man kann jedes beliebige Folgenglied berechnen, indem man den Index m als Summe von 2er-Potenzen schreibt:

$$c_m, m = 2^{k_1} + 2^{k_2} + \dots + 2^{k_i}, k_1 > k_2 > \dots > k_i$$

Beispiel: $37 = 2^5 + 2^2 + 2^0 \Rightarrow 37 = 100101_2$

$c_{37} = (-1)c_5 = (-1)^2 c_1 = (-1)^3 c_0 = -1$, da $c_0 = 1$

Oder allgemein: $c_m = (-1)^j$, wobei j die Anzahl der Einsen in der Binärdarstellung von m ist.

Darüberhinaus gibt es noch einen bemerkenswerten Satz über die Struktur von g_∞ , der lautet:

Es gibt kein Wort der Form www in der Thue-Morse-Folge, wobei w eine beliebige endliche Kombination von a s und b s ist.

Das heißt: Weder die Wörter aaa , noch $aabaabaab$ oder $baabbaabbaab$ wird man in g_∞ finden!

Die Wörter, die in g_∞ vorkommen hingegen, kommen mehrfach vor. Das besagt der Satz: n sei eine natürliche Zahl. Dann gibt es eine andere natürliche Zahl N so, dass wenn ein beliebiges Wort w der Länge n in der Thue-Morse-Folge gegeben ist, jedes beliebige Wort W der Länge N eine Kopie von w enthält.

Für die Länge des Wortes W gilt folgende Abschätzung: Ist k die kleinste Zahl, so dass $2^k \geq n$, dann enthält ein Wort der Länge 2^{k+4} auf jeden Fall das Wort w der Länge n .

Das Wort aab (und alle anderen Wörter der Länge 3!) ist also in jedem Wort der Länge 64 aus der Thue-Morse-Folge enthalten.

13.3 Graphische Interpretation

Das bloße Hinschreiben der Generationen ist natürlich bei komplizierteren Systemen ermüdend und wenig hilfreich. Außerdem verfolgen wir ja das Ziel, biologisches Wachstum zu modellieren. Dazu definieren wir ein paar Symbole, denen eine festgelegte graphische Bedeutung zukommen soll:

- F : Vorwärtsschritt (zeichnen)



Abbildung 13.1: Die Evolution einer botanischen Form.

- G : Vorwärtsschritt (nicht zeichnen)
- $+$: Drehung nach links
- $-$: Drehung nach rechts
- $[$: aktuelle Position speichern
- $]$: zu zuletzt gespeicherter Position gehen

Viele der uns bekannten fraktalen Kurven kann man so schön mit Hilfe von L-Systemen ausdrücken. Der *Cantor-Staub* ist zum Beispiel durch das System

$$\begin{aligned} V &= \{F, G\} \\ w &= F \\ P &= \{F \mapsto FGF, G \mapsto GGG\} \end{aligned}$$

gegeben. Der *Koch-Kurve* entspricht das System

$$\begin{aligned} V &= \{F, +, -\} \\ w &= F \\ P &= \{F \mapsto F + F - -F + F\} \end{aligned}$$

wobei die Drehung um 60° erfolgt.

Ein schönes Beispiel für eine botanische Form ist das System

$$\begin{aligned} V &= \{F, +, -, [,]\} \\ w &= F \\ P &= \{F \mapsto FF - [-F + F + F] + [+F - F - F]\} \end{aligned}$$

mit einem Rotationswinkel von $180/7^\circ$. Abbildung 13.1 zeigt die Generationen 1 bis 3.

In Abbildung 13.2 ist g_6 mit einer anderen Darstellung (Diskussion in Kapitel 15) abgebildet.

Die Ähnlichkeit mit einem realen Zweig ist verblüffend! Weitere Beispiele sind die Systeme mit den Produktionsregeln

$$P = \{F \mapsto FF - [-F + F + F] + [+F - F - F]\} \quad (\text{Winkel } 20^\circ)$$

und

$$P = \{F \mapsto F[+FF][-FF]F[+FF][-FF]F\} \quad (\text{Winkel } 36^\circ).$$

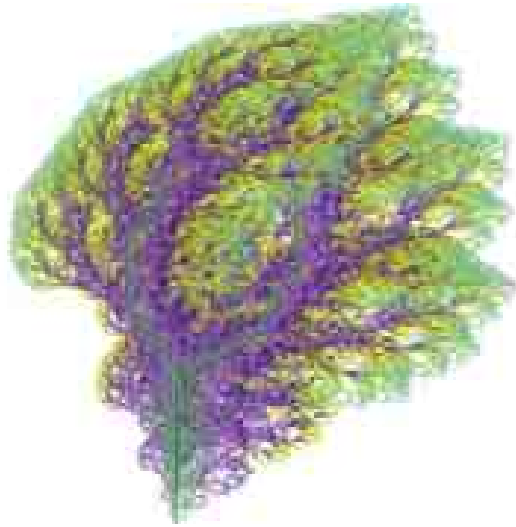


Abbildung 13.2: Das fertige Bild.



Abbildung 13.3: $F \mapsto FF - [-F + F + F] + [+F - F - F]$.

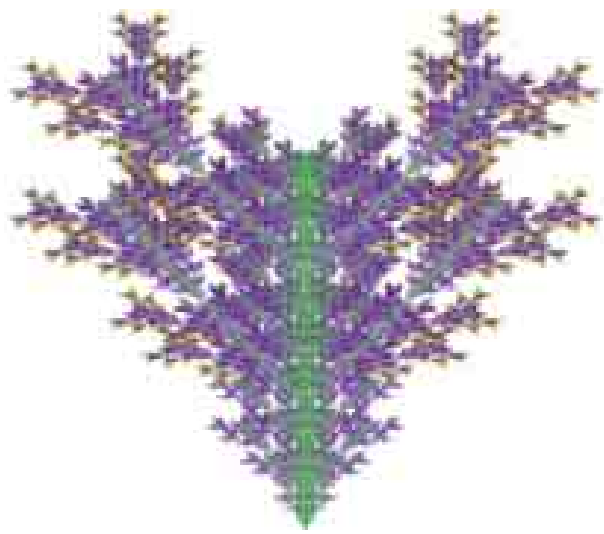


Abbildung 13.4: $F \mapsto F[+FF][-FF]F[+FF][-FF]F$.

14 Zellularautomaten

Ein *Zellularautomat* besteht aus einer Ansammlung von *Zellen*, die einer festgelegten *Regel* folgend ihren Zustand ändern können. Ein *Boole'scher* oder *Binärer Zellularautomat* ist ein Zellularautomat, dessen Zellen genau zwei verschiedene Zustände haben können, bezeichnet mit „an“ und „aus“ bzw. 1 und 0.

14.1 Binäre eindimensionale Automaten

Wir betrachten zunächst eindimensionale binäre Zellularautomaten. Dabei soll der Zustand der Zelle von ihrem eigenen und dem ihrer beiden Nachbarn abhängen. Es gibt also $2^3 = 8$ Möglichkeiten, nämlich 111, 110, 101, 100, 011, 010, 001 und 000. Die Ersetzungsregel muss also aus 8 Teilen bestehen.

Ordnet man die verschiedenen Stadien des Zellularautomaten untereinander an, so kann man seine Entwicklung verfolgen. Die vertikale Achse ist hierbei also praktisch die Zeitachse.

Als Beispiel nehmen wir den Zellularautomat, der durch die Regel

$$\begin{aligned} 111 \mapsto 0, 110 \mapsto 1, 101 \mapsto 0, 100 \mapsto 1, \\ 011 \mapsto 1, 010 \mapsto 0, 001 \mapsto 1, 000 \mapsto 0 \end{aligned}$$

bestimmt wird. Nimmt man als Anfangszustand eine einzige angeschaltete Zelle, so ergibt sich das *Sierpinski-Dreieck*. Im Kapitel über *fraktale Kurven* ist eine ausführliche Abbildung der ersten Stufe, wobei der * für eine 1 steht.

Um die Klasse der binären Zellularautomaten mit zwei Nachbarn systematisch zu ordnen, schreibt man die Ersetzungsregel in der Form 01011010, was in binärer Form der dezimalen 90 entspricht. Daher bezeichnet man die oben angegebene Ersetzungsregel kurz als Regel 90.

Als nächstes betrachten wir binäre Automaten mit 5 Nachbarn, d. h. in die Regel wird nicht nur der Zustand der nächsten Nachbarn, sondern auch noch deren Nachbarn einbezogen. Es gibt $2^5 = 32$ Möglichkeiten. Da eine 32-stellige Regel zu unhandlich ist, unterscheiden wir nicht mehr zwischen den einzelnen Zuständen der Zellen, sondern nur zwischen der Gesamtsumme der Zustände, welche zwischen 0 und 5 liegen muss. Damit reduziert sich unsere Regel auf 5 Stellen, welche man wie oben codieren kann.

Die Abbildungen 14.1 und 14.2 zeigen zwei Bilder, die ausgehend von einer aus zufällig gewählten Zuständen bestehenden Strecke, von solchen Automaten erzeugt wurden. Die Darstellung wurde abgebrochen, sobald sich die neue Zeile zum ersten Mal nicht von der Ausgangszeile unterschieden hat



Abbildung 14.1: Binärer Zellularautomat.



Abbildung 14.2: Noch ein binärer Zellularautomat.

14.2 Binäre zweidimensionale Automaten

Der wohl bekannteste binäre zweidimensionale Zellularautomat oder sogar der bekannteste Zellularautomat überhaupt ist durch John Conways *Game of Life* gegeben.

Im Game of Life sind für den Zustand einer Zelle ihre 8 Nachbarn verantwortlich, und zwar in folgender Weise:

Eine Zelle bleibt im Zustand 1, wenn sie mindestens 2 und höchstens 3 Nachbarzellen im Zustand 1 hat, ansonsten wird sie in den Zustand 0 versetzt.

Eine Zelle wird von 0 auf 1 gesetzt, wenn sie genau 3 Nachbarzellen im Zustand 1 hat.

Als Analogie bezeichnet man Zellen im Zustand 1 als „lebend“, Zellen im Zustand 0 als „tot“.

Damit ergibt sich folgende Zusammenfassung:

Leben: Minimal: 2, Maximal: 3

Geburt: Minimal: 3, Maximal: 3

Daher bezeichnet man diese Version des Game of Life als Life-2333. Es stellt sich heraus, dass man zwar als Minimal- und Maximalwerte beliebige Zahlen von 0 bis 8 wählen kann, dieses aber die einzige sinnvolle Kombination ist.

Interessant ist nun, die Anordnung der Zellen mit der Zeit zu verfolgen. Es zeigt sich, dass sich ganze Zellgruppen bilden, die sich über die Ebene bewegen, sich irgendwo festsetzen oder sich auflösen bzw. neue Zellgruppen schaffen.

Wie man sich das vorzustellen hat, kann hier nur an einem Beispiel gezeigt werden, einer Zellgruppe, die sich, einer Lebensform gleich, quer über die Ebene bewegt. Diese Zellgruppe hat eine Zykluslänge von 4, in welcher sie sich um eine Einheit nach links und oben bewegt.

14.3 Die Ameisensimulation

Die Ameisensimulation ist ein Automat von anderem Charakter als die bisher beschriebenen. Turk, Bunivomitch und Troubetzkoy entwickelten diese Simulation, die mit einfachsten Mit-

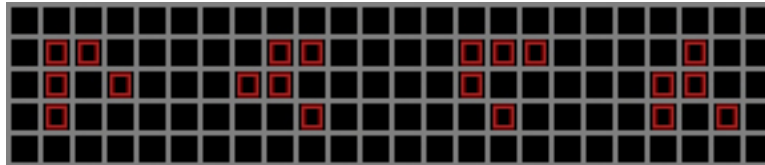


Abbildung 14.3: Läufer im Game of Life.

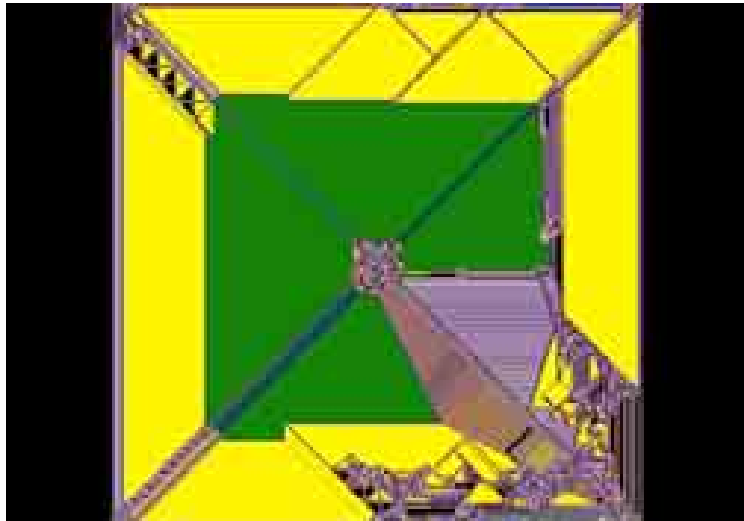


Abbildung 14.4: 2 Ameisen, Regel 10010011.

tehn erstaunliches Verhalten produziert:

Zu Beginn der Simulation befindet sich eine gewisse Anzahl von „Ameisen“ in der Mitte der Ebene. Diese Ameisen bewegen sich nun folgendermaßen:

Die Zelle, auf der sich die Ameise befindet, sei im Zustand i . Wenn die i -te Ziffer der Regel eine 1 ist, dreht sich die Ameise um 90° nach rechts und geht einen Schritt vor. Ist sie eine 0, dreht sie sich nach links und geht einen Schritt vor. Beim Verlassen wird der Zustand der aktuellen Zelle um 1 erhöht.

Man kann die Ameisensimulation schön visualisieren, indem man die Zellen gemäß ihres Zustands einfärbt. Hier sind 4 Bilder von Ameisensimulationen. Das durchweg chaotische Verhalten der Ameisen beim Hinterlassen ihrer Spur kann leider nicht abgebildet werden.

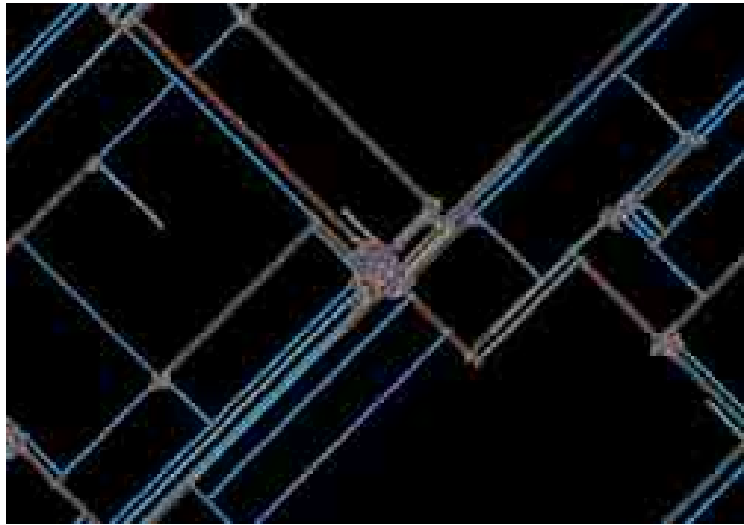


Abbildung 14.5: 10 Ameisen, Regel 111000.

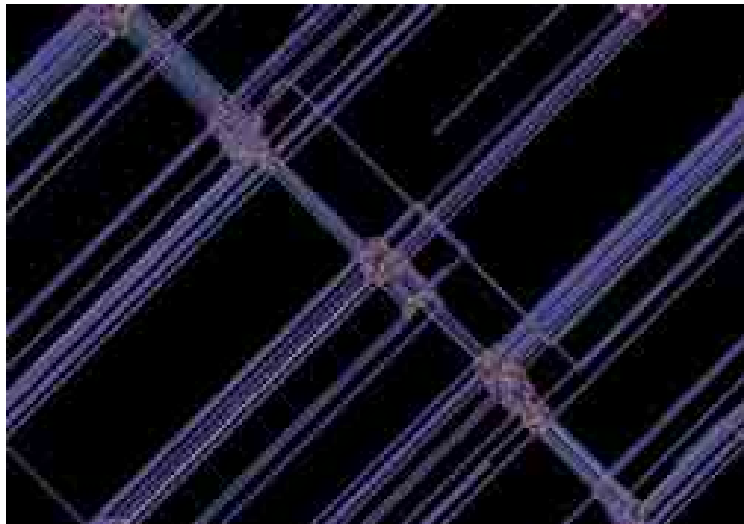


Abbildung 14.6: 10 Ameisen, Regel 110.

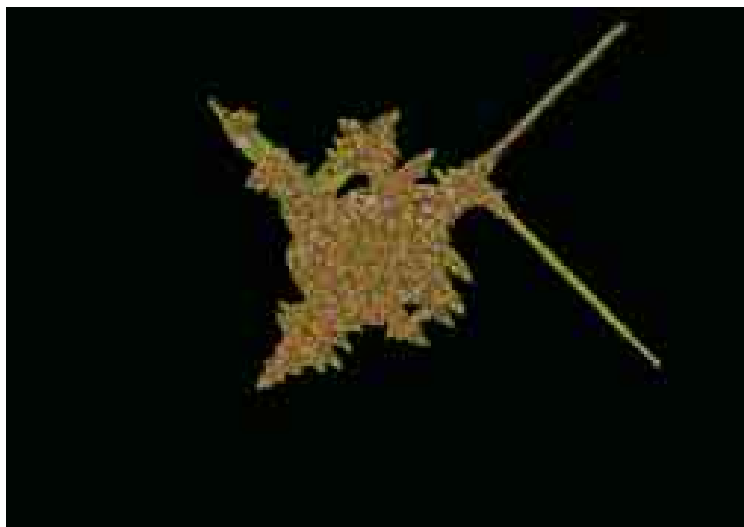


Abbildung 14.7: 10 Ameisen, Regel 111100000.

15 Eine kurze Geschichte der Farbe II

15.1 Mandelcloud

In Kapitel 8 wurde der *Escape-Time-Algorithmus* vorgestellt, mit dem Mandelbrot- und Julia-Mengen dargestellt werden können. *Mandelcloud* von Noel Giffin liefert ein weiteres Verfahren zur Darstellung der Mandelbrot-Menge. Hierbei wird die Mandelbrot-Menge jedoch nicht in der *Parameter-Ebene*, sondern in der *dynamischen Ebene* dargestellt, und zwar indem der Orbit der zu untersuchenden Punkte geplottet wird. Da die Mandelbrot-Menge der Attraktor dieser Iteration ist, entsteht ihr „Abdruck“ aus einer „Wolke“ von Orbits.

Abbildung 15.1 zeigt ein Bild dieser „Wolke“, bei dem die Farbe des Pixels jedesmal inkrementiert wurde, wenn er erneut in einem Orbit auftauchte. Man sieht deutlich, dass bestimmte Bereiche der Mandelbrot-Menge häufiger von Orbits besucht werden als andere.

Bei Abbildung 15.2 wurden weniger Punkte untersucht, um die einzelnen Orbits besser erkennen zu können. Die Färbung hing diesmal vom Orbit ab.

15.2 Das Newton-Fraktal

Die offensichtlichste Möglichkeit, Newton-Fraktale zu färben, besteht natürlich darin, jeder Nullstelle eine Farbe zuzuordnen und die Punkte danach zu färben, gegen welche Nullstelle ihr Orbit konvergiert. Deutlich interessanter wird das Bild nun, wenn man auch noch die Iterationszahl mit einbezieht in der Art, dass man die Pixel nach den Iterationen mod 2 färbt. Damit ergeben sich für jede Nullstelle zwei Farben, eine bei geradzahligem Iteration und eine bei ungeradzahligem.

Auch interessant ist die Möglichkeit, die Punkte ganz nach den Iterationen zu färben. Man erhält zwar keine Informationen mehr über den Attraktor, aber dafür erhält man mehr Einblicke in die Dynamik des Fraktals.

Das Newton-Fraktal lässt sich analog zu Mandelbrot- und Julia-Mengen auch dreidimensional darstellen, wenn man die Iterationszahl als Höhe interpretiert. Auf die Färbung des Pixels hat das keinen Einfluss; alle Möglichkeiten funktionieren auch in 3D.

15.3 Iterierte Funktionssysteme

Die Bilder im Kapitel über IFS wurden so erzeugt: Jedesmal, wenn ein Pixel von der Iteration getroffen wurde, wurde seine Farbe inkrementiert. Eine andere Möglichkeit entsteht durch die Natur der IFS: Da bei jedem Iterationsschritt ein Zufallsgenerator auslöst, welche affine

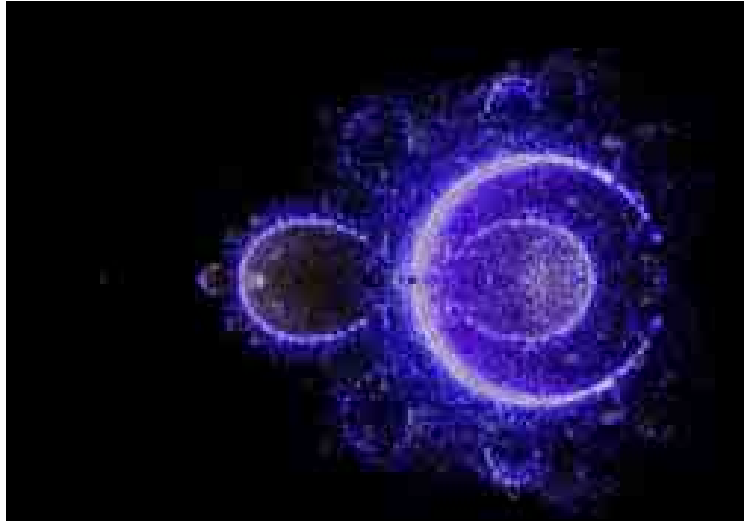


Abbildung 15.1: Die Mandelbrot-Menge als Mandelcloud.



Abbildung 15.2: Einzelne Orbits durch Mandelcloud.

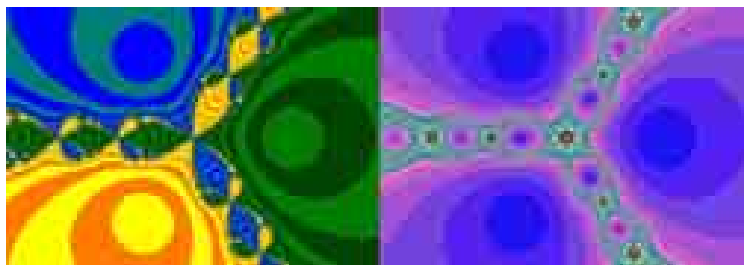


Abbildung 15.3: Attraktor & Iterationen mod 2 und Iterationen.



Abbildung 15.4: Das Newton-Fraktal in 3D.



Abbildung 15.5: Die affinen Abbildungen des Farns.

Abbildung gewählt werden soll, kann man den Pixel auch nach der affinen Abbildung färben. Dann sieht der Farn beispielsweise so aus wie in [Abbildung 15.5](#).

Man erkennt klar die vier affinen Abbildungen: Stengel, linkes Blatt, rechtes Blatt, oberes Blatt.

15.4 Lindenmayer-Systeme

Bei der gewöhnlichen Darstellung von Lindenmayer-Systemen bleibt nicht viel Spielraum für eine Färbung, ein Strich wird einfach schwarz gezeichnet. Anders sieht es aus bei Systemen, die einen *Stack* benutzen, d. h. Systeme, die Regeln haben, die die Zeichen „[“ und „]“ benutzen. In diesem Fall kann man die Färbung nämlich von der aktuellen Stackhöhe abhängig machen. Damit wurden die Bilder der Pflanzen generiert wie [Abbildung 13.2](#). Es wird deutlich: Je näher der Strich am Hauptast des Zweigs ist, desto geringer ist die Stackhöhe.

Index

- Äquipotentialkurve, 30, 50
- affine Abbildung, 59
- Ameisensimulation, 82
- Attraktor, 12, 54, 60
- Bifurkationspunkt, 17
- Cantor-Menge, 25
- Cantor-Staub, 66, 78
- Cantor-Vorhang, 66
- Chaos, 11, 17
 - deterministisches, 11
- Chaos-Spiel, 61
- Curlicue-Fraktal, 52
- Dekomposition, 47
- Dimension
 - Ähnlichkeitsdimension, 10, 65
 - Box-Counting-Dimension, 9
 - euklidische, 8
 - fraktale, 9
 - globale, 9
 - Hausdorff-Dimension, 9, 28, 65
 - Kapazitätsdimension, 9
 - lokale, 9
 - topologische, 9
- Distanzfunktion, 60
- Einheitswurzel, 55
- Escape-Time-Algorithmus, 45
- Feigenbaum-Diagramm, 19
- Feigenbaum-Konstante, 19
- Fibonacci-L-System, 75
- Fixpunkt, 11, 15
- Fundamentale Dichotomie, 25
- Game of Life, 82
- Generator, 65
- Herzkurve, 31
- Hutchinson-Operator, 59
- IFS, 59
- Initiator, 65
- Iteration, 11, 54, 60
 - graphische, 11
- Iterierte Funktionssysteme, 59, 86
- Julia-Menge, 23
 - zusammenhängende, 24
- Kontraktion, 61
- Kreisteilungsgleichung, 55
- Kreiszahl, 39, 52
- Kurve
 - Drachenkurve, 71
 - fraktale, 10, 65
 - Koch-Kurve, 68, 78
 - Peano-Kurve, 9, 10, 71
- Lindenmayer-System, 75, 88
- logistische Gleichung, 15
- Mandelbrot-Menge, 28
 - für Quaternionen, 42
- Menger-Teppich, 71
- Newton-Fraktal, 54, 86
- Orbit, 11, 23, 54

Pascal'sches Dreieck, 70
Periode, 11
Periodenverdopplung, 16
periodisches Fenster, 19
Populationsdynamik, 13

Quaternionen, 42

Repeller, 12

Sarkowskii-Theorem, 19
Satz von Moivre, 55
Selbstähnlichkeit, 10, 24, 28, 65
Sierpinski-Dreieck, 59, 68, 81
Skaleninvarianz, 65
Spinnwebdiagramme, 11, 15

Teufelstreppe, 66
Thue-Morse-L-System, 76

Vorfixpunkt, 11

Wachstum
 exponentielles, 13
 logistisches, 13
 Verhulst-Wachstum, 13
Wachstumskonstante, 13

Zellularautomat, 70, 81
Zyklus, 11