

**Zwischen Ordnung und Chaos:  
Das komplexe dynamische Verhalten nichtlinearer Gleichungen**

Seminararbeit von Jakob Gollreiter

Inhaltsverzeichnis

[1. Einleitung 3](#_Toc497341496)

[2. Die Dynamik von Populationen 4](#_Toc497341497)

[2.1 Iterationsverfahren 4](#_Toc497341498)

[2.2 Die uneingeschränkte Populationsdynamik 5](#_Toc497341499)

[2.3 Fixpunkte einer Funktion 7](#_Toc497341500)

[2.4 Die eingeschränkte Populationsdynamik 8](#_Toc497341501)

[2.5 Das Feigenbaum-Diagramm 12](#_Toc497341502)

[3. Die Mandelbrotmenge 16](#_Toc497341503)

[3.1 Die komplexen Zahlen 16](#_Toc497341504)

[3.2 Iterationsverfahren zur Mandelbrotmenge 18](#_Toc497341505)

[3.3 Definition der Mandelbrotmenge 20](#_Toc497341506)

[3.4 Betrachtungen der Mandelbrotmenge 22](#_Toc497341507)

[4. Zusammenfassung 24](#_Toc497341508)

[5. Literaturverzeichnis 25](#_Toc497341509)

[5.1 Buchquellen 25](#_Toc497341510)

[5.2 Internetquellen 25](#_Toc497341511)

[5.3 Bildquellen 26](#_Toc497341512)

[6. Abbildungsverzeichnis 27](#_Toc497341513)

## 1. Einleitung

*„Nichts kann existieren ohne Ordnung. Nichts kann entstehen ohne Chaos.“*

*Albert Einstein [[1]](#footnote-1)*

Was bedeutet Chaos überhaupt? Das Wort leitet sich aus dem altgriechischen Wort χάος her. Es bedeutet so viel wie leerer Raum oder die gestaltlose Urmasse. Mit Chaos kann ein nicht eindeutig vorhersagbares Verhalten naturwissenschaftlicher, wirtschaftlicher und gesellschaftspolitischer Vorgänge bezeichnet werden.[[2]](#footnote-2) In der Physik versteht man chaotische Vorgänge als zeitlich und räumlich nicht vorhersagbare Bewegungen in einem abstrakten Raum. Ordnung heißt κόσμος und beschreibt den Zustand, in dem alles nach einem System sortiert oder angeordnet ist.[[3]](#footnote-3)

Unsere Welt besteht aus einer Vielzahl von dynamischen Prozessen, welche nicht-deterministische Eigenschaften zu besitzen scheinen. Beispiele sind die Wetterentwicklung, Aktienkurse, manche Vorgänge in der Thermodynamik, das Doppelpendel oder gewisse Formen von Herzrhythmusstörungen.

In dieser Seminararbeit werden mathematische Modelle vorgestellt, anhand derer sich das Konvergenzverhalten von dynamischen Systemen, wie z.B. die zeitliche Entwicklung von Populationen bei eingeschränktem Lebensraum oder begrenzter Nahrung untersuchen lässt. Die logistische Gleichung nach P.H. Verhulst und dessen Analyse durch Mitchel Feigenbaum zeigt sehr anschaulich, wie nahe vorhersagbares und nicht vorhersagbares Verhalten beieinander liegen und wie dieses bereits durch winzige Veränderungen in der Steuerung solcher Systeme beeinflusst werden kann. An der „Grenzlinie“ zwischen Ordnung und Chaos können selbstähnliche Strukturen oder Fraktale entstehen. Diese besitzen häufig ästhetische Formen, zum Beispiel das von Benoit Mandelbrot entdeckte sogenannte *Apfelmännchen*.

Alle Funktionsgraphen und Mengendarstellungen in dieser Arbeit wurden durch Softwarecode in der Programmiersprache Python von mir selbst erstellt und liegen in einem gesonderten Dokument bei.

## 2. Die Dynamik von Populationen

Die Dynamik von Populationen kann anhand mathematischer Iterationsverfahren simulieren werden. In der Praxis ist eine Simulation oft das einzige Werkzeug, um ein gewisses Verständnis für sehr komplizierte und analytisch nicht zu formulierende Zusammenhänge zu entwickeln.

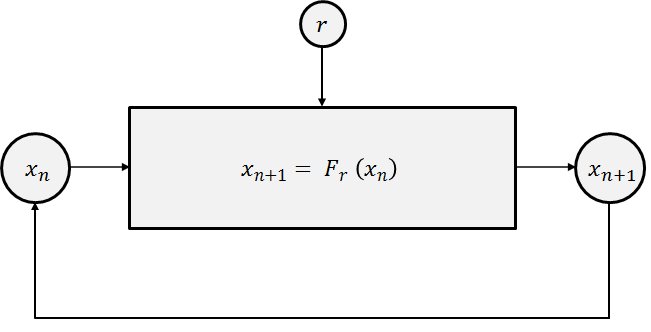
### 2.1 Iterationsverfahren

Viele Prozesse und Vorgehen in der Natur gehen nach dem Prinzip der Iteration vor. Um in etwa eine Prognose für eine Population einer bestimmten Tierart im nächsten Jahr zu geben, braucht man die Anzahl der Tiere in diesem Jahr und ihren Wachstumsfaktor.

In der Mathematik, insbesondere in der Theorie der dynamischen Systeme, bezeichnet man als *Iteration* [[4]](#footnote-4) die wiederholte Anwendung derselben Funktion , also die Bildung von

in einem Zahlenraum , zum Beispiel der reellen Zahlen .

Ein Iterationsverfahren kann auch als ein *Rückkopplungsprozess* betrachtet werden. Das Ergebnis eines Iterationsschrittes wird als neue Eingabe im Folgeschritt verwendet. (engl. *feedback)*.[[5]](#footnote-5) [[6]](#footnote-6) Dies zeigt Abbildung 1.



*Abb.1: Iterationsverfahren als Rückkopplung mit Parameter*

Der Iterationsprozess wird mit einem beliebig gewählten gestartet und hieraus eine Zahlenfolge, , … erzeugt. Diese Zahlenfolge wird über möglichst viele Iterationsschritte betrachtet. Dadurch lässt sich die Entwicklung über lange Zeitperioden genauer simulieren.[[7]](#footnote-7)

Abbildung 1 zeigt eine weitere Eingabe, den *Wachstumsfaktor* Dieser Parameter wird einmalig zu Beginn der Iteration konstant gewählt, bleibt also über den Iterationsverlauf unverändert. In der praktischen Anwendung der Iteration, z.B. bei der Untersuchung der Dynamik von Populationen, stellt der Wachstumsfaktor einen Einflussparameter dar, welcher die Entwicklung der Population, beispielweise bei der Einschränkung von Lebensraum oder der Nahrungsmittelzufuhr, modellieren soll. Dieser Modellierungsparameter wird nun eine zentrale Rolle spielen.

### 2.2 Die uneingeschränkte Populationsdynamik

Im Folgenden werden Grundgrößen zur mathematischen Formulierung einer Populationsdynamik eingeführt:

Anzahl von Tieren

Maximale Anzahl von Tieren

Population

Population im Jahr

Die sogenannte *uneingeschränkte Populationsdynamik* wird durch eine Funktion F(x) und einen Wachstumsfaktor r definiert. Anhand dieser Funktion können Wachstum oder Vermehrung dargestellt werden:

, (negative Populationen sind sinnlos).[[8]](#footnote-8)

Ist die Startpopulation im 0-ten Jahr bekannt, so kann man mit der uneingeschränkten Populationsdynamik die Population im 1-ten und 2-ten, bis zum n-ten Jahr iterativ berechnen:

**…** .

Die folgenden Beispiele sollen zeigen, dass sich eine Population, deren Wachstum sich nach diesem Modell beschreiben lässt, in Abhängigkeit vom Wachstumsfaktor r wie folgt entwickeln:

: die Anfangspopulation bleibt unverändert.

: die Population stirbt aus

: die Population wächst grenzenlos an. [[9]](#footnote-9)

Beispiel 1:  
Mit Wachstumsfaktor und einer beliebigen Startpopulation  bleibt die Population unverändert =

Beispiel 2:  
Mit konstantem Wachstumsfaktor und Startpopulation ergibt sich die monoton fallende, gegen Null konvergierende Iterationsfolge:

… Grenzwert:

Wenn sich eine Iteration immer näher an die Null annähert, diese jedoch nie erreicht, bezeichnet man das als konvergent gegen Null.[[10]](#footnote-10) In unserem Anwendungsfall bedeutet das, die Population stirbt aus. Die Ursache hierfür könnte sein, dass sich der Lebensraum einer bestimmten Spezies, aufgrund der Vermehrung von natürlichen Fressfeinden verringert hat, Nahrungsmittel nicht ausreichend vorhanden waren oder sich Krankheiten ungebremst ausbreiteten konnten.

Beispiel 3:   
Im Gegensatz dazu ergibt sich mit dem Wachstumsfaktor und dem Startwert die streng monoton wachsende, gegen keinen Grenzwert konvergierende Folge:

… Grenzwert: .

Dies entspricht unbeschränktem Wachstum und wird *Divergenz* genannt.[[11]](#footnote-11) Das gilt für eine Spezies, die beispielsweise keine natürlichen Feinde und unbegrenzten Lebensraum hat.

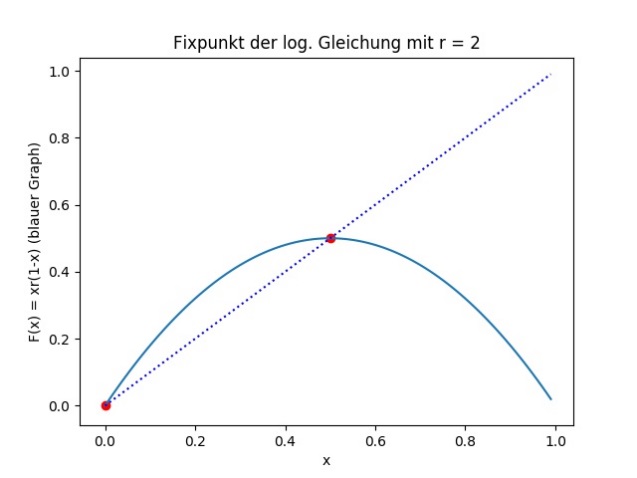
### 2.3 Fixpunkte einer Funktion

Bei der Analyse von Populationsdynamiken spielt der Begriff des *Fixpunktes*[[12]](#footnote-12) eine wichtige Rolle. Ein Fixpunkt der Funktion ist definiert als

.

An einem Fixpunkt hat also die Funktion als Ergebniswert genau den Eingabewert .

Um den Fixpunkt einer Funktion beispielweise graphisch zu bestimmen, zeichnet man in ein Koordinatensystem die Funktion und die Winkelhalbierende des Koordinatensystems ein. Fixpunkte liegen an den Schnittpunkten der beiden Graphen. Abbildung 2 zeigt den Graphen der Funktion im Bereich [0;1] und die Winkelhalbierende. An den Schnittpunkten der beiden Graphen liegen die Fixpunkte und .

**

*Abb.2: Beispiel für ein grafisches Verfahren zur Bestimmung von Fixpunkten*

Betrachtet man an der Stelle ()) die Steigung der Funktion, d.h. die Ableitung ) oder die Steigung *s* einer Tangente an den Graphen der Funktion, so lassen sich Eigenschaften von Fixpunkten definieren:

Fixpunkt heißt *stabil* oder *anziehend*:

Fixpunkt heißt *superstabil*:

Fixpunkt heißt *instabil* oder *abstoßend*: [[13]](#footnote-13)

Ein Wert ) = 1 entspricht praktisch einer Steigung von 45 Grad. Im Beispiel in Abbildung 2 hätte die Tangente an den Graphen an der Stelle ) die Steigung s = 0, d.h. der Fixpunkt 0.5 ist superstabil.

### 2.4 Die eingeschränkte Populationsdynamik

Im Gegensatz zur uneingeschränkten Populationsdynamik ermöglicht die e*ingeschränkte* Varianteeine wesentlich realistischere Modellierung der zeitlichen Entwicklung von Populationen durch Berücksichtigung des Einflusses von Einschränkungen von Nahrung und Lebensraum:

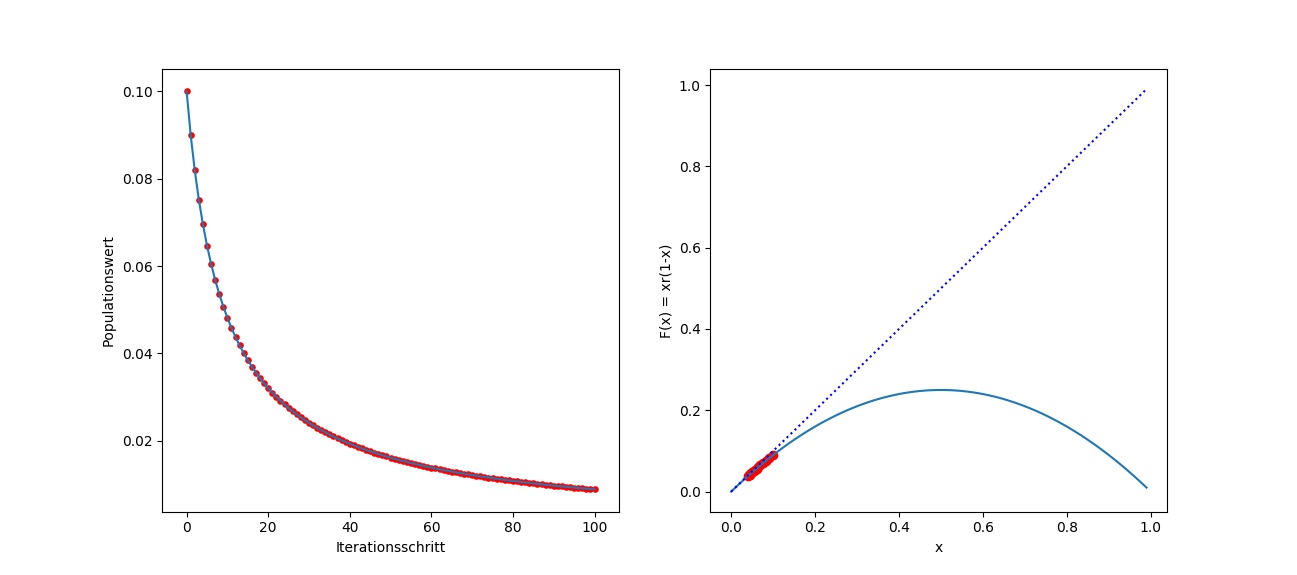
,

Der Term beschreibt diese Einschränkungen. Sobald die Population in die Nähe ihres Maximums wächst, stellt der Term eine Gegenregulierung dar, welche das Wachstum in der nachfolgenden Generation wieder einschränkt. Das heißt, wenn die Population nahezu das Maximum von 1 erreicht hat, entsteht ein Mangel an Nahrung und Lebensraum. Infolge dessen verhungern einige Tiere und die Population sinkt im nächsten Jahr wieder. Die Population reguliert sich also selbst. Dieses Iterationsverfahren ist der Spezialfall einer von P.F. Verhulst im Jahr 1845 formulierten, sogenannten *logistischen Gleichung*.[[14]](#footnote-15)

Die Fixpunkte der logistischen Gleichung sind und mit .[[15]](#footnote-16)

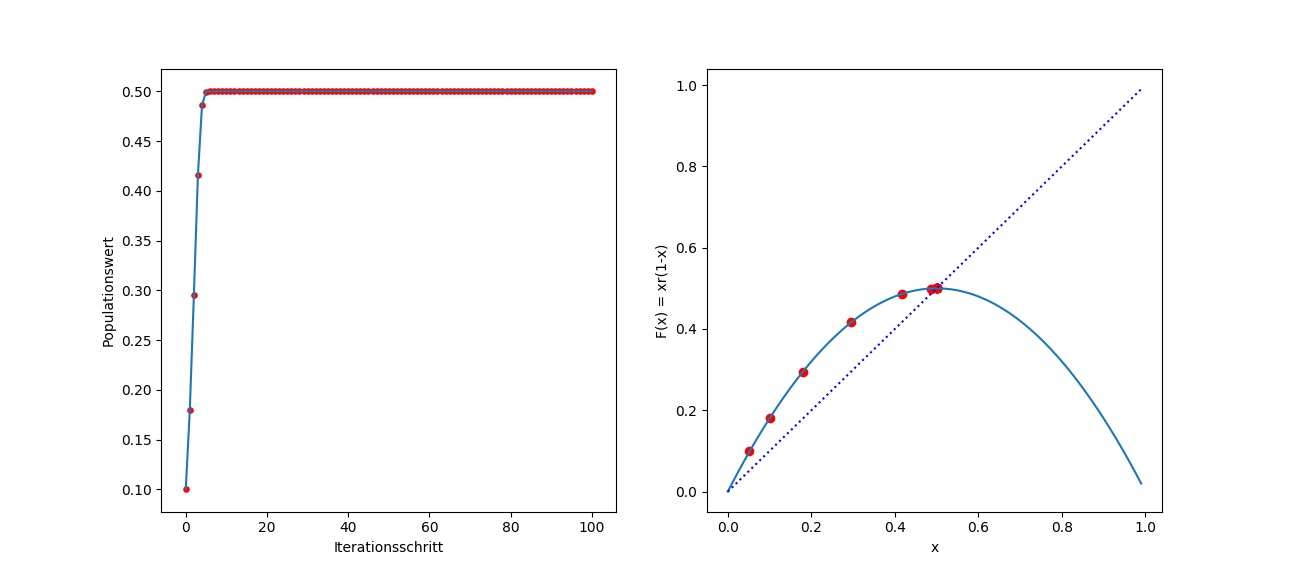
Im Folgenden wird die Dynamik der logistischen Gleichung beispielhaft anhand fünf unterschiedlicher Wachstumsfaktoren untersucht.[[16]](#footnote-17)

Beispiel 1:  
In Abbildung 3 auf der linken Seite sieht man die Iterationsfolge der logistischen Gleichung mit dem Startwert *= 0.1* und dem konstanten Wachstumsfaktor . Man erkennt, dass die Population monoton fällt, gegen Null konvergiert und somit ausstirbt. Auf der rechten Seite ist der Graph der Funktion zur logistischen Gleichung im Intervall dargestellt. Der Schnittpunkt mit der Winkelhalbierenden ist beim Fixpunkt = 0. Die Werte der Iteration (in rot markiert) konvergieren dorthin, was sich im Plot bereits andeutet. Der Fixpunkt ist anziehend.



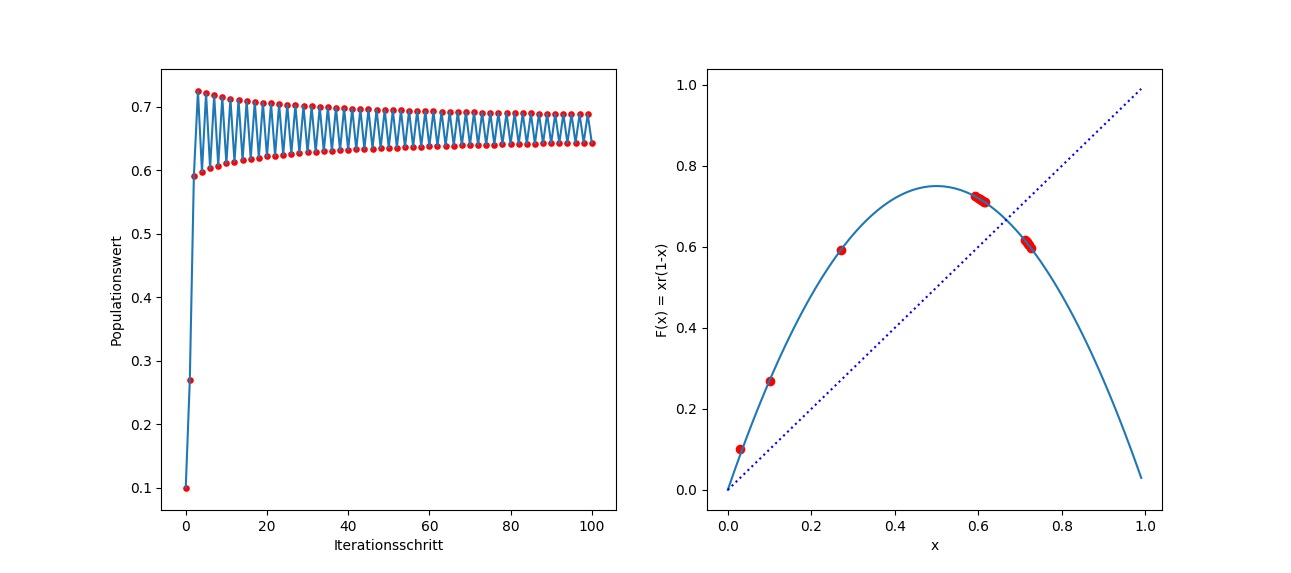
*Abb.3: Iterationsfolge der logistischen Gleichung (Wachstumsfaktor r = 1, = 0.1)*

Beispiel 2:   
Die Abbildung 4 zeigt auf der linken Seite die ersten 100 Iterationsschritte der logistischen Gleichung mit dem Wachstumsfaktor . Schon nach geringer Zeit stabilisiert sich die Population auf einen konstanten Wert von und ist damit in einem Gleichgewichtszustand. Dies ist in der Natur möglicher Weise der Idealzustand, da die Population immer gleich groß bleibt, jedoch ist das nur in einem theoretischen Modell exakt möglich und in der Natur, wegen der Vielfalt an Einflussfaktoren, wohl äußerst unrealistisch. Auf der rechten Seite der Abbildung 4 zeigt sich ein Fixpunkt graphisch als Schnittpunkt zwischen Funktionengraph und Winkelhalbierender bei . Die Iterationsfolge wächst ausgehend vom Startwert streng monoton und konvergiert zu diesem Fixpunkt. Der Fixpunkt ist superstabil.



*Abb.4: Iterationsfolge mit konstantem Verhalten (Wachstumsfaktor r = 2, = 0.1)*

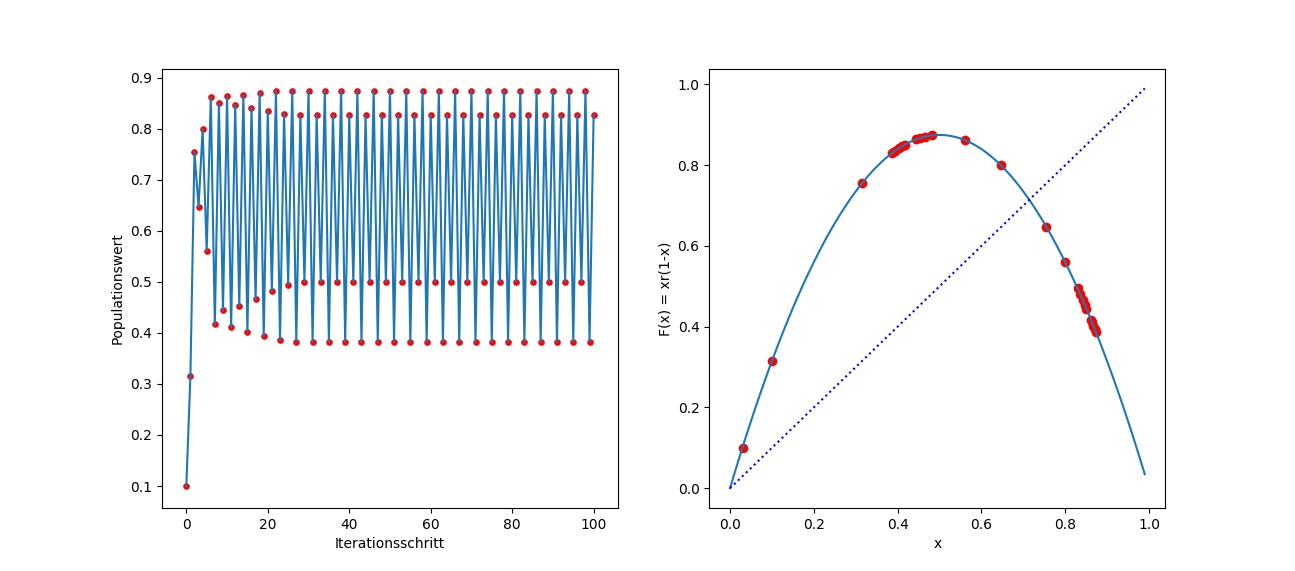
Beispiel 3:



*Abb.5: Iterationsfolge mit periodischem Verhalten (Wachstumsfaktor r = 3.01, x0 =0.1)*

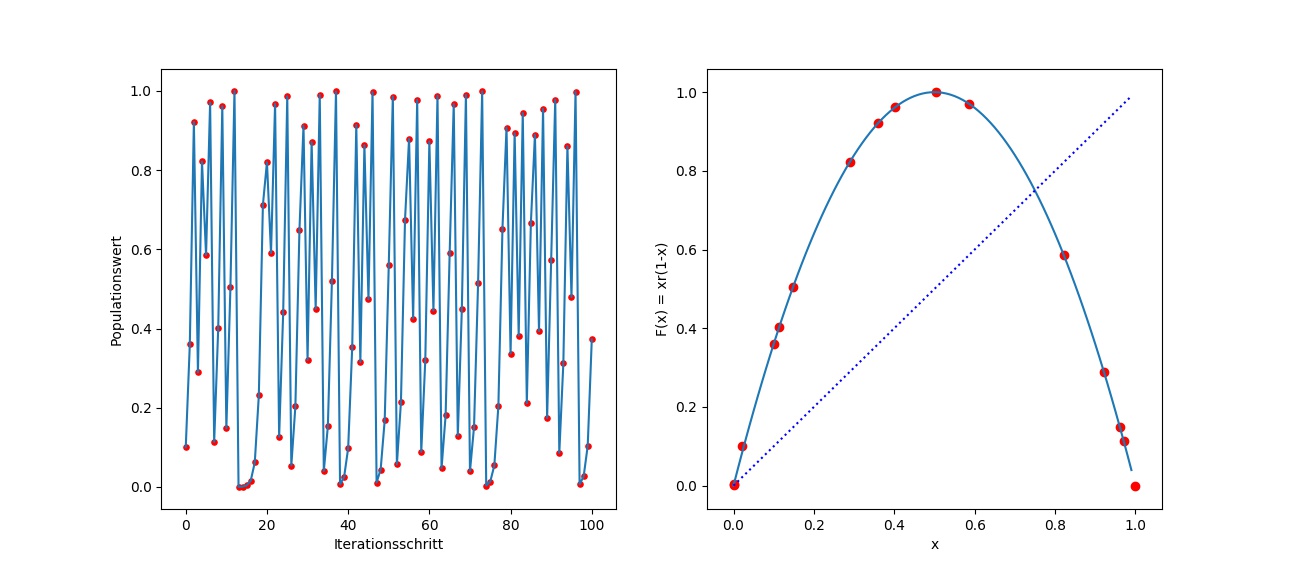
Die Abbildung 5 zeigt die Iterationsfolge zu der logistischen Gleichung mit dem Wachstumsfaktor . Die Zahlenfolge der Iterationswerte verläuft periodisch. Nach der Einschwingphase, in diesem Fall nach ungefähr 60 Iterationen, bleibt der maximale und minimale Wert unverändert und springt zwischen diesen beiden Werten hin und her.[[17]](#footnote-18) Der Graph auf der linken Seite der Abbildung hat aufgrund der zwei abwechselnden Werte keinen genau festgelegten Grenzwert, sondern einen sogenannten *Grenzzyklus.* Bemerkenswert ist die Betrachtung zum Verhalten am Fixpunkt auf der rechten Seite der Abbildung. Obwohl offensichtlich ein Fixpunkt existiert, erreicht die Zahlenfolge der Iterationen mit dem Startwert  *= 0.1* diesen Fixpunkt nicht. Stattdessen oszilliert die Zahlenfolge um diesen herum. Der Fixpunkt ist abstoßend.

Beispiel 4:  
Die Abbildung 6 zeigt die Iterationswerte der logistischen Gleichung mit dem Wachstumsfaktor . Die Zahlenfolge der Iterationswerte besitzt nun einen Grenzzyklus mit Periode 4. Nach der Einschwingphase, in diesem Fall nach ungefähr 40 Iterationen, spaltet sich der Graph in vier Äste auf und besitzt somit ebenfalls einen Grenzzyklus und keinen Grenzwert.[[18]](#footnote-19) Auf der rechten Seite der Abbildung ist ersichtlich, dass wieder ein Fixpunkt existiert, dieser aber von der Iterationsfolge nicht erreicht wird. Der Fixpunkt ist abstoßend und instabil.



*Abb.6: Iterationsfolge mit periodischem Verhalten (Wachstumsfaktor r = 3.5, = 0.1)*

Beispiel 5:   
In Abbildung 7 verhält sich der Populationsprozess nun weder konvergent noch periodisch. Vielmehr springt die Population zwischen einer beliebigen Vielzahl von Werten hin und her. Der Fixpunkt ist abstoßend. Anders, als in den zuvor betrachteten Iterationsfolgen, kann hier zu einem bestimmten, festgelegten Zeitpunkt aus der Folge der zurückliegenden Populationswerte keine Prognose für den Verlauf der zukünftigen Werte getroffen werden. Dieser Populationsprozess wird deshalb als *chaotisch* bezeichnet, obwohl ihm eine *deterministische* Berechnungsvorschrift zu Grunde liegt.[[19]](#footnote-20)



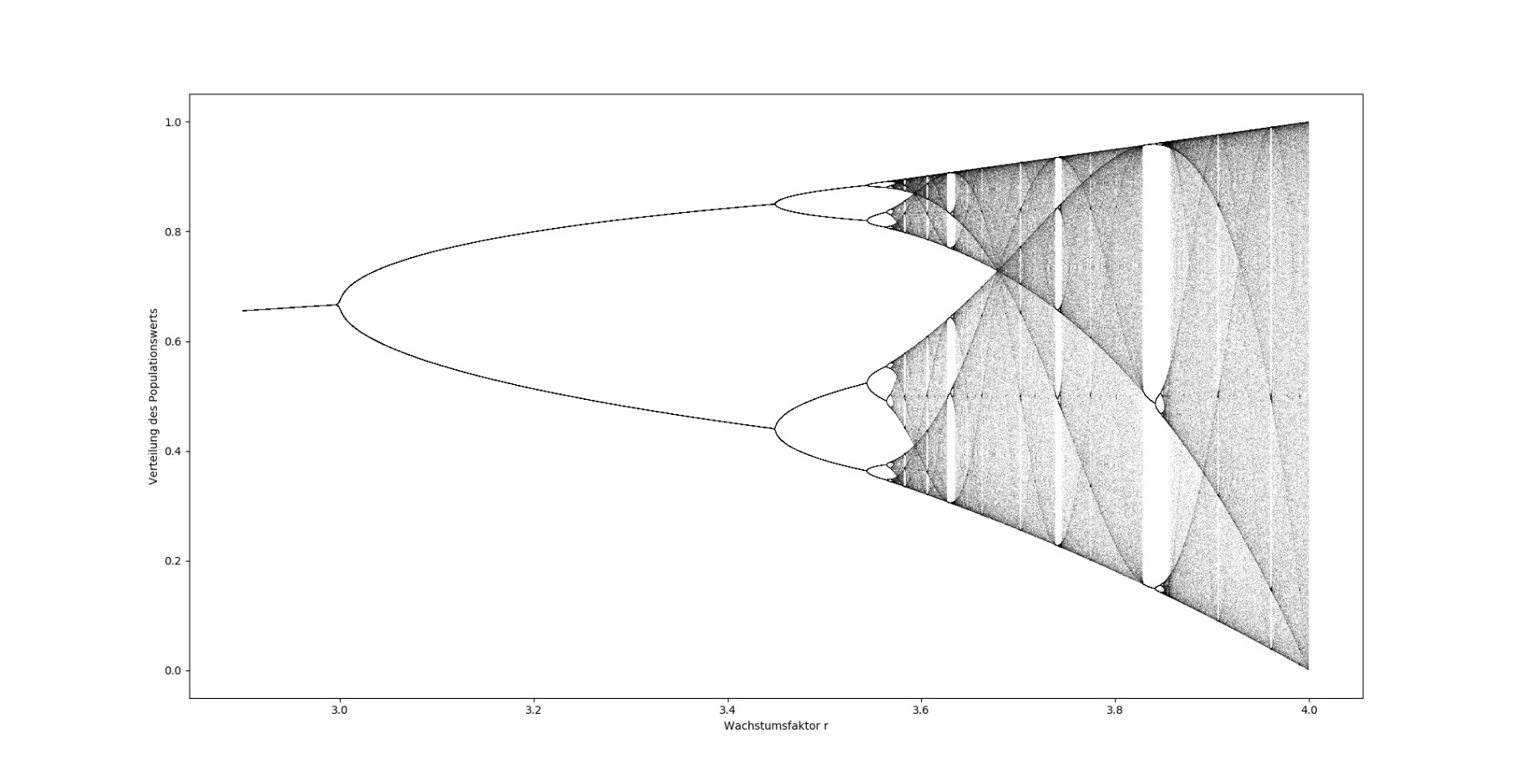
*Abb.7: Iterationsfolge mit chaotischem Verhalten (Wachstumsfaktor r = 4, = 0.1)*

Die genauere Betrachtung der Entwicklung dieser fünf unterschiedlichen Populationsprozesse folgt im nächsten Kapitel.

### 2.5 Das Feigenbaum-Diagramm

Um das Verhalten von der Ordnung zum deterministischen Chaos eindrucksvoll zu veranschaulichen, betrachten wir das sogenannte *Feigenbaum-Diagramm*, welches der amerikanische Mathematiker M.J. Feigenbaum entdeckt hat. Das Feigebaum-Diagramm veranschaulicht die dynamische Entwicklung einer Population gemäß der logistischen Gleichung. Dabei werden die Wachstumsfaktoren r auf der x-Achse und die Iterations-werte der logistischen Gleichung auf der y-Achse dargestellt. Der Fokus liegt bei diesem Diagramm nicht auf dem Einschwingvorgang, sondern auf den Grenzwerten, Grenzzyklen oder dem chaotischen Verhalten der Iterationen der logistischen Gleichung für die Wachstumsfaktoren .

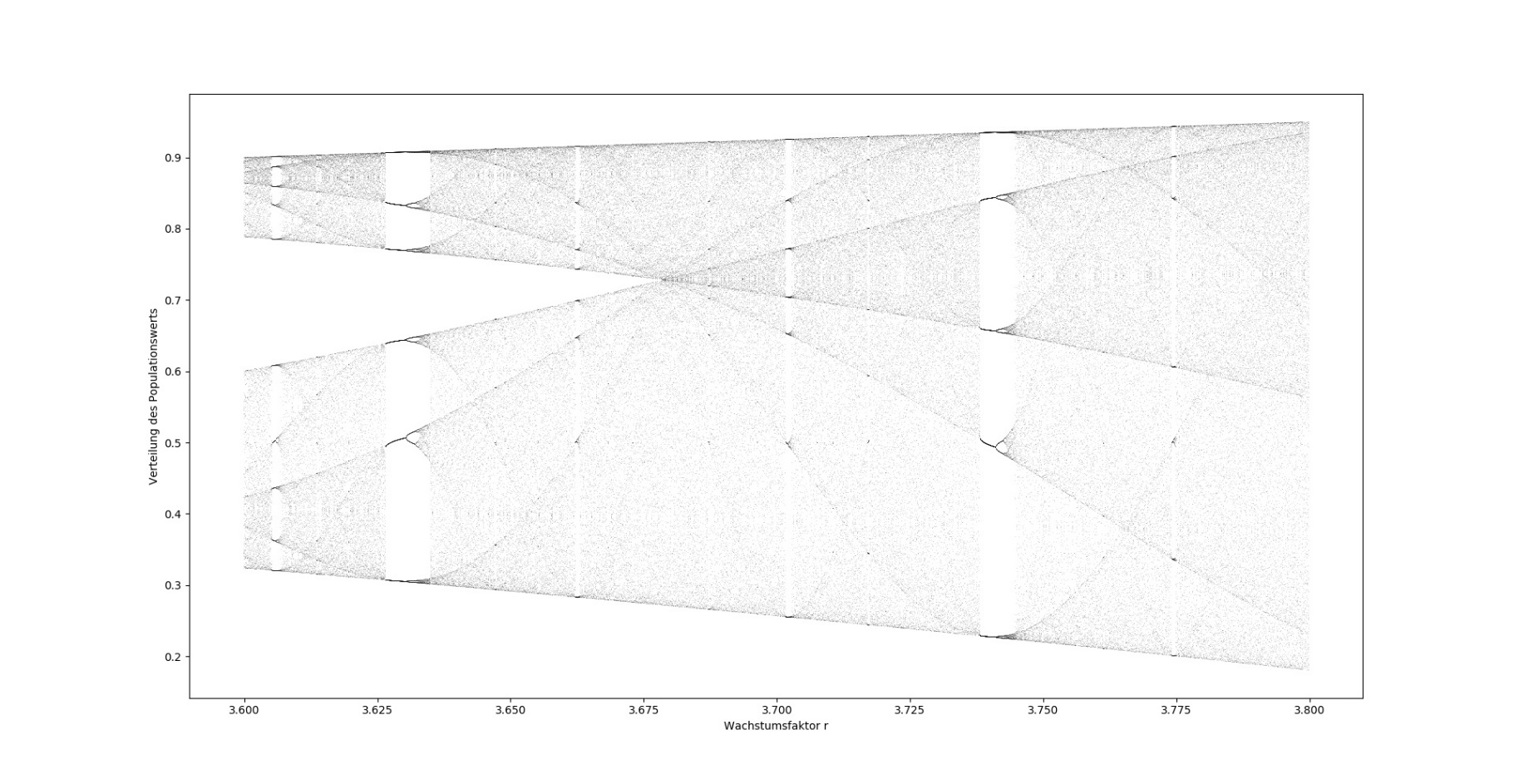
Im vorhergehenden Kapitel wurde die logistische Gleichung mit genau fünf unterschiedlichen Wachstumsfaktoren betrachtet. Nun wird aber das Verhalten der Iterationsfolgen der logistischen Gleichung in einem ganzen *Intervallbereich* des Wachstumsfaktors r untereinander verglichen und graphisch dargestellt. Zu jedem betrachteten Wachstumsfaktor werden zunächst 1000 Iterationen der logistischen Gleichung durchgeführt, bevor anschließend 100 Iterationswerte geplottet werden.



*Abb.8: Das Feigenbaum-Diagramm im Intervall für r = [2.9; 4]*

Der Graph der Iterationsergebnisse weist einige sehr interessante Stellen auf. Die Feigenbaum-Diagramme in den Abbildungen 8 und 9 zeigen die Grenzwerte oder Grenzzyklen einer Population in Abhängigkeit von ihrem Wachstumsfaktor. Um möglichst exakte Daten zu erhalten, erfasst man die Daten der ersten Iterationen nicht. Die Ergebnisse werden erst nach der 1000. Iteration geplottet um im Falle eines Grenzyklus diesen bestmöglich abzubilden.[[20]](#footnote-21)

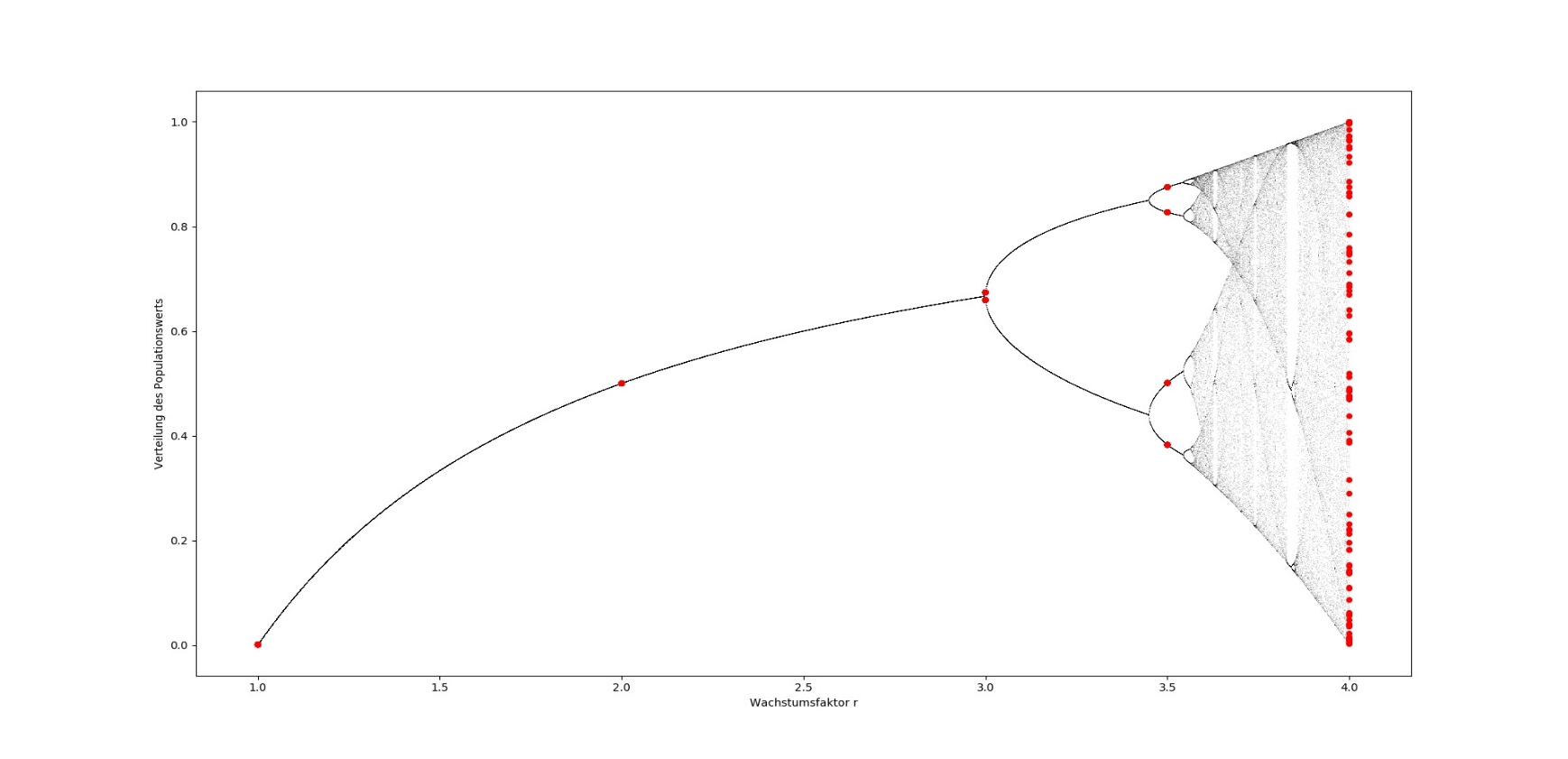
Betrachten wir nun die Abbildungen 8 und 9 etwas genauer: Je größer der Wachstumsfaktor r wird, desto chaotischer und unübersichtlicher wird das System[[21]](#footnote-22). Jenseits des Wertes fängt das System an, sich chaotisch zu verhalten. Jedoch finden sich immer wieder in unregelmäßigen Abständen Lücken, die auf ein geordnetes Verhalten schließen lassen.[[22]](#footnote-23)

**

*Abb.9: Das Feigenbaum-Diagramm im Intervall für r = [3.6; 3.8]*

In diesem Feigenbaum-Diagramm lassen sich Punkte erkennen, an welchen eine Periodenverdopplung stattfindet. Diese werden auch als *Bifurkationen* bezeichnet. Die Folge der Wachstumsfaktoren r, an welchen eine Bifurkation stattfindet, wird über eine Beziehung mit der sogenannten *Feigenbaumkonstante* ausgedrückt.[[23]](#footnote-24)

In Abbildung 10 werden die Grenzwerte, Grenzzyklen bzw. das chaotisches Verhalten der fünf verschiedenen Iterationsfolgen, welche in Kapitel 2.4 vorgestellt wurden, im Gesamtzusammenhang mit der Feigenbaummenge eingetragen: Für r = 1 konvergiert die Iterationsfolge gegen Null, d.h. Population stirbt aus. Für r = 2 schwingt sich die Iterationsfolge auf den stabilen Grenzwert 0.5 ein. Für r = 3.01 und r = 3.5. schwingt die Iterationsfolge periodisch zwischen zwei bzw. vier Werten hin und her. Und bei r = 4.0 verhält sich die Iterationsfolge chaotisch.



*Abb.10: Das Feigenbaum-Diagramm mit ausgewählten Wachstumsfaktoren r*

## 3. Die Mandelbrotmenge

### 3.1 Die komplexen Zahlen

Die komplexen Zahlen sind notwendig, um die Mandelbrotmenge zu verstehen. Eine komplexe Zahl besteht aus zwei Teilen. Der erste ist der „reale“ und der zweite der „imaginäre“ Teil. Man kann so eine Zahl wie ein Objekt mit zwei unterschiedlichen Attributen betrachten.

Damit man die beiden Zahlen voneinander unterscheiden kann, erhält der imaginäre Teil der Zahl ein i. Theoretisch könnte man so eine Zahl auch als zweidimensional bezeichnen. Man kann komplexe Zahlen wie gewohnt addieren, subtrahieren, multiplizieren bzw. dividieren.

In der üblichen Notation wird eine *komplexe Zahl* durch die Einführung der sogenannten *imaginären Zahl* ***i*** mit der Eigenschaft definiert. Die Schreibweise lautet:

Die Zahl wird als *Realteil* und die Zahl y als *Imaginärteil* von z bezeichnet. Abgekürzt lautet dies x = Re(z) und y = Im(z). Mit wird der Raum der komplexen Zahlen bezeichnet. Dieser ist eine Erweiterung des Raums der reellen Zahlen .

Re(z)

Im(z)

1

2

3

-1

*1*

*2*

*3*

*-1*

*Abb.11: Darstellung einer komplexen Zahl z in Vektorform* [[24]](#footnote-25)

Bevor man versucht Funktionen mit komplexen Zahlen zu verstehen, sollte man zuerst nachvollziehen können, wie man solche Zahlen visualisiert. Zum einfachen, ersten Verständnis kann eine komplexe Zahl auch als zweidimensionaler Vektor aus zwei reellen Zahlen betrachtet werden. Man zeichnet ein Koordinatensystem und trägt auf der x-Achse den Realteil und auf der y-Achse den Imaginärteil der komplexen Zahl ein.[[25]](#footnote-26) Siehe Abb. 11.

Im Folgenden werden die Rechenregeln der Addition, Subtraktion sowie der Multiplikation erläutert. Die Division ist nicht notwendig, um die Mandelbrotmenge zu verstehen.[[26]](#footnote-27)

|  |  |
| --- | --- |
| Addition: |  |
| Subtraktion: |  |
| Multiplikation: | = |

Wie man anhand der oberen Beispiele sieht, sind die Rechenregeln für die komplexen Zahlen die gleichen wie mit allen anderen Zahlen. Jedoch gibt es eine einzige Ausnahme! Es gilt:

Beispiel 1:   
Das Rechnen mit komplexen Zahlen ist vergleichbar mit dem Rechnen zweidimensionaler Vektoren. Hier wird als Beispiel die Addition zweier komplexer Zahlen in Vektorschreibweise vorgestellt:

Nun folgt die übliche Notation:

Beispiel 2:   
Hier ist ein Rechenbeispiel für das Multiplizieren zweier komplexer Zahlen.

[[27]](#footnote-28)

Da gilt, ist .

Beispiel 3: Der *Betrag* einer komplexen Zahl wird als definiert. Dies entspricht der Distanz vom Ursprung des Koordinatensystems bis zum Punkt und kann unter Zuhilfenahme des [Satz des Pythagoras](https://de.wikipedia.org/wiki/Satz_des_Pythagoras) berechnet werden. Für die Abbildung 11 gilt dann:

**=**

### 3.2 Iterationsverfahren zur Mandelbrotmenge

Die Mandelbrotmenge wurde von dem französischen Mathematiker Benoit Mandelbrot entdeckt und wird in der Literatur, aufgrund ihrer Gestalt, oft auch als *Apfelmännchen* bezeichnet.

Für die Definition der Mandelbrotmenge wird - wie bei der die logistische Gleichung, siehe Kapitel 2.1 - zunächst ein Iterationsverfahren benötigt.

Erzeugt wird mit einem Startwert und einem konstanten Parameter eine Iterationsfolge mit komplexen Zahlen.[[28]](#footnote-29)

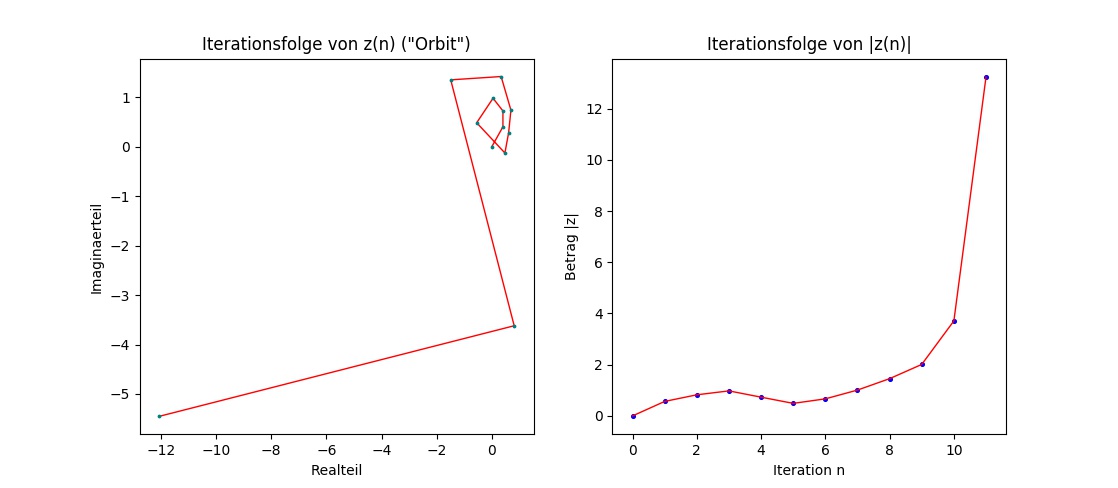
Die komplexe Form der Iterationsfolge kann auch in Form von zweidimensionalen Vektoren mit den reellen Zahlen x, y, a, b geschrieben werden. Dies erleichtert das Verständnis und wird für die praktische Implementierung in einem Algorithmus benötigt. Mit und ergibt sich dann eine Iterationsfolgeim zweidimensionalen Raum:

Im Anschluss wird analysiert, wie sich die Iteration bei immer gleichem Startwert

, aber mit unterschiedlichen Parametern , entwickelt.

Beispiel 1:

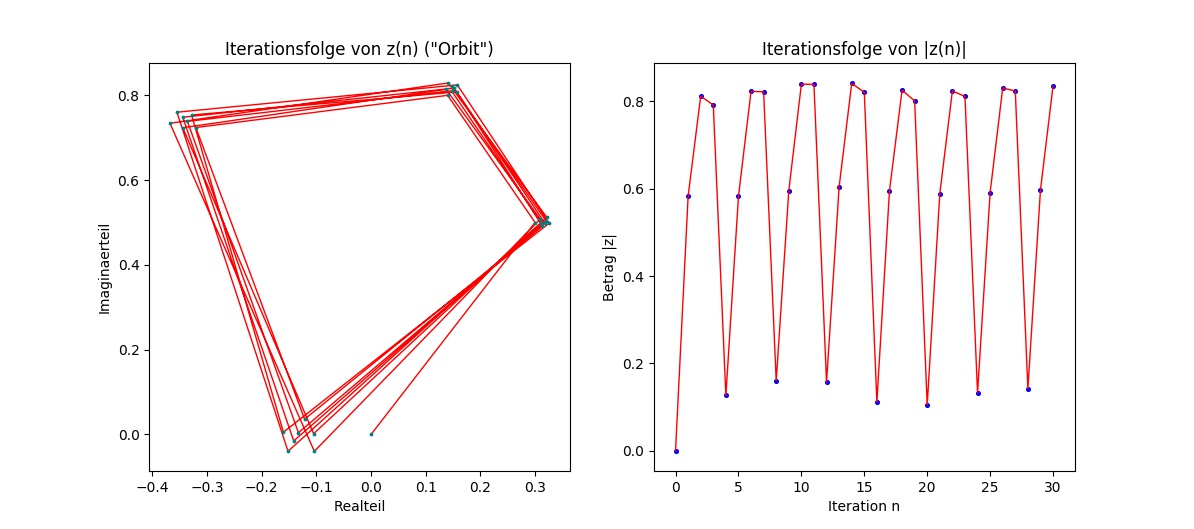
Das Diagramm in Abbildung 12 wird *Orbit* oder *Orbit-Plot* genannt. In diesem Falle ist der Parameter. Auf der linken Seite des Abbildes erkennt man, dass die Werte initial in der Nähe des Ursprungs umherkreisen und dann plötzlich ins Unendliche divergieren*.* Auf der rechten Seite werden die Beträge der Iterationsfolge für die ersten Iterationen geplottet. Hieraus ist das divergente Verhalten gut zu erkennen.[[29]](#footnote-30)



*Abb.12: Divergente Iterationsfolge, außerhalb der Mandelbrotmenge; , c=(0.4+i 0.4)*

Beispiel 2:

Im Gegensatz dazu gibt es auch Parameter c, mit welchen die Iteration anfängt zu oszillieren, wie zum Beispiel mit dem Parameter , in Abbildung 13. Schon nach kurzer Zeit nähern sich die Iterationswerte einer Gruppe von vier verschiedenen Wertebereichen an und oszillieren dann auch in diesen Bereichen.[[30]](#footnote-31) Dies zeigt auch der Plot auf der rechten Seite.

**

*Abb.13: Zyklische Iterationsfolge, innerhalb der Mandelbrotmenge;*

*, c=(0.3+i 0.5)*

### 3.3 Definition der Mandelbrotmenge

Die Mandelbrotmenge ist definiert als die komplexen Zahlen , für welche die Iterationsfolge nicht divergiert. Der Startwert liegt hierbei stets im Koordinatenursprung.[[31]](#footnote-32)

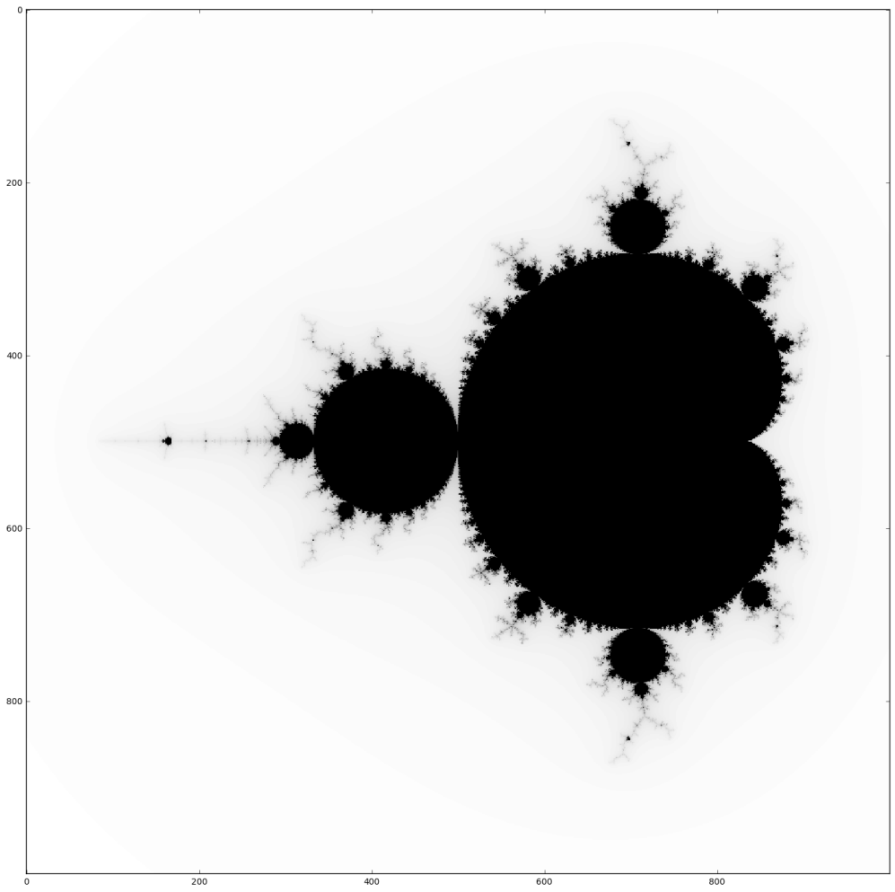
Die Werte der Iterationsfolgen können gegen einen bestimmten Grenzwert konvergieren, ins Unendliche divergieren oder in einem Grenzzyklus zwischen verschiedenen Werten oszillieren.

Je nachdem, welcher Parameterwert c gewählt wurde, gibt es vier unterschiedliche Möglichkeiten, wie sich die Werte der Iterationsfolge verhalten:

1. Die Werte konvergieren gegen einen Fixpunkt.
2. Die Werte besitzen einen periodischen Grenzzyklus, d.h. die Werte springen zwischen zwei oder mehreren Werten hin und her
3. Die Werte bleiben beschränkt, verhalten sich aber chaotisch mit Wechsel zwischen fast periodischen Grenzzyklen und scheinbar zufälligem Verhalten.
4. Die Werte divergieren ins Unendliche. [[32]](#footnote-33)

Die Möglichkeiten 1, 2 und 3 bilden die Mandelbrotmenge.

In Abbildung 14 ist ein bestimmter Punkt c im Diagramm schwarz eingefärbt, wenn die Iterationsfolge für diesen Parameterwert nicht divergiert. [[33]](#footnote-34) Außerhalb der Mandelbrotmenge divergieren die Iterationen. Zur besseren Veranschaulichung, wie stark die Funktion divergiert, kann man das Diagramm auch noch mit Graustufen oder Farben verfeinern.[[34]](#footnote-35)



Im(c)

Re(c)

1

1

-1

-1

-2

*Abb.14: Die Mandelbrotmenge in der komplexen Zahlenebene*

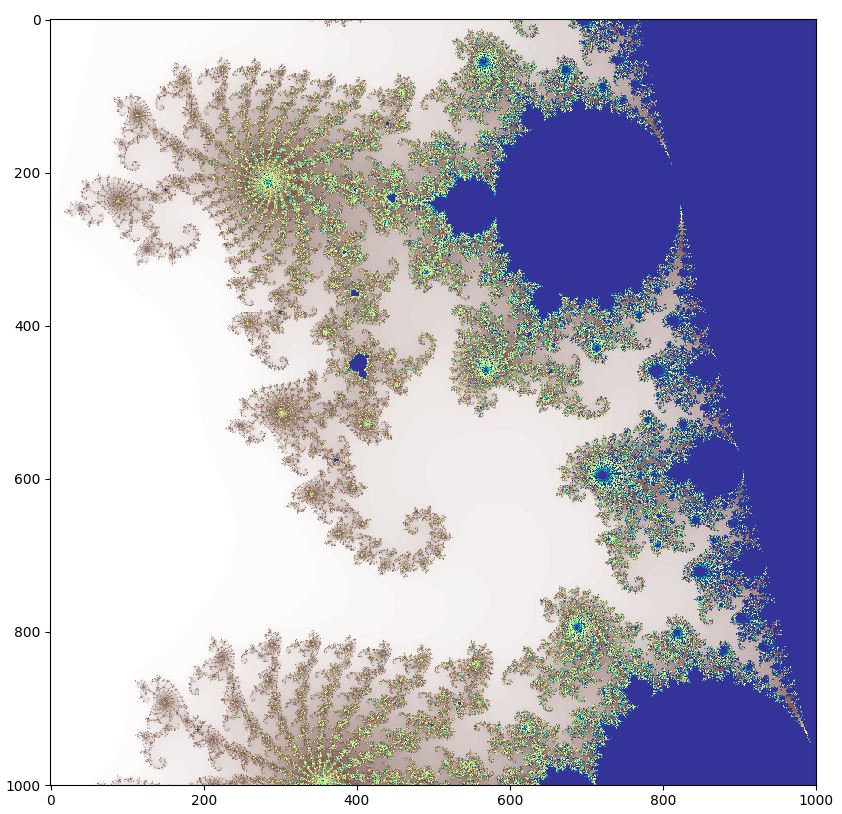
### 3.4 Betrachtungen der Mandelbrotmenge

Die Mandelbrotmenge ist selbstähnlich, jedoch ist diese Ähnlichkeit nicht immer ganz exakt.

Die Mandelbrotmenge ist symmetrisch zur x-Achse (Realteil) und hat eine Ausdehnung auf der x-Achse von [-2; 0,25]. Die Menge ist beschränkt auf einen Kreis um den Ursprung mit Radius *2, d.h.* .

Entgegen der intuitiven Annahme ist dieses sogenannte *Apfelmännchen* eine zusammenhängende Menge, was erst 1982 von dem Mathematiker Douady Hubbard bewiesen werden konnte. An der Grenze zwischen Konvergenz und Divergenz ist die Mandelbrotmenge fraktal. In der Anwendung heißt das, je näher man an das Objekt heranzoomt, desto öfters kann man weitere Apfelmännchen erkennen, die abgesehen von ihrer Größe, fast exakt gleich aussehen. Das heißt, dass man weitere Apfelmännchen erkennen kann, je näher man an das Objekt heran zoomt.[[35]](#footnote-36)

Die Abbildung 16 zeigt einen Teilbereich aus der Mandelbrotmenge. Die linke obere Ecke hat den Parameterwert c = (-0.7496875 + i 0.1439375) und die rechte untere Ecke den Wert c = (0.1439375 + i 0.1639375). Insgesamt hat das Bild eine Pixelauflösung von 1000 mal 1000, und für jedes Pixel wurde die Iteration genau 700 mal berechnet, um den Einschwingvorgang zu berücksichtigen. Für die insgesamt 700.000.000 Berechnungen hat mein Intel i5 Notebook ungefähr vier Minuten gerechnet. Die Farbgebung wurde automatisch durch die Farbskala „terrain“ der Python-Softwarebibliothek *matplotlib* festgelegt*.*

****

*Abb.15: Ausschnitt der Mandelbrotmenge* [[36]](#footnote-37)

4. Zusammenfassung   
Diese Seminararbeit handelt von einfachen mathematischen Modellen der Chaostheorie. Anfangs werden verschiedene Iterationsverläufe zur Bestimmung der dynamischen Entwicklung von Populationen anhand einer linearen und einer quadratischen Gleichung vorgestellt. Im weiteren Verlauf ist dann, unter Zuhilfenahme des Feigenbaum-Diagramms, die Entwicklung der einzelnen Populationsdynamiken mit ihren unterschiedlichen Wachstumsfaktoren zu beobachten. Hierbei zeigt sich der Übergang zwischen Ordnung und Chaos sehr deutlich. Zum Verständnis der Mandelbrotmenge werden zunächst die komplexen Zahlen und einige Rechenregeln eingeführt. Abschließend wird dann das Iterationsverfahren zur Berechnung der Mandelbrotmenge erläutert und analysiert. Die graphischen Ergebnisse dieser Computersimulationen zeigen faszinierende und ästhetische Formstrukturen. Die verblüffenden Grenzverläufe zwischen Ordnung und Chaos beeindrucken nicht nur Mathematiker.

## 5. Literaturverzeichnis

### 5.1 Buchquellen

[1] Bräuer, Kurt, Chaos, Attraktoren und Fraktale, Logos-Verlag, Berlin, 2002,   
1. Auflage

[2] Herrmann, Dietmar, Algorithmen für Chaos und Fraktale, Addison-Wesley, Bonn, 1994, 1. Auflage

[3] Louis, Dirk, Java Eine Einführung in die Programmierung, Carl Hanser-Verlag, München, 2014, 1. Auflage

[4] Mandelbrot, Benoît B., Fraktale Geometrie der Natur, Birkhäuser Verlag, 1987,   
1. Auflage

[5] Peitgen, Heinz-Otto, The Beauty of Fractals, Springer-Verlag, Berlin, 1986,   
1. Auflage

[6] Rashid, Tariq, Make your own Mandelbrot, CreateSpace Independent Publishing Platform, 2014, 1. Auflage

[7] Voß, Herbert, Chaos und Fraktale selbst programmieren, Franzis, Berlin, 1994,   
1. Auflage

### 5.2 Internetquellen

1. <https://de.glosbe.com/de/grc/Ordnung>  
 (Letztes Zugriffsdatum 05.11.2017)

2. <http://www.sasserlone.de/tag/244/chaos/>  
 (Letztes Zugriffsdatum 05.11.2017)

3. <https://de.wikipedia.org/wiki/Mandelbrot-Menge#Galerie_der_Iteration>  
 (Letztes Zugriffsdatum 05.11.2017)

### 5.3 Bildquellen

*Titelbild*: Erstellt auf Basis des Python-Softwarecodes aus dem Buch: Rashid, Tariq, Make your own Mandelbrot, CreateSpace Independent Publishing Platform, 2014, 1. Auflage, Kapitel Fractals

*Abb.1:* Peitgen, Heinz-Otto (1986): The Beauty of Fractals, Springer-Verlag, Berlin, 1. Auflage, S. 5

*Abb.2:* Abbildung selbst erstellt (Sourcecode beigelegt)

*Abb.3:* Abbildung selbst erstellt (Sourcecode beigelegt)

*Abb.4:* Abbildung selbst erstellt (Sourcecode beigelegt)

*Abb.5:* Abbildung selbst erstellt (Sourcecode beigelegt)

*Abb.6:* Abbildung selbst erstellt (Sourcecode beigelegt)

*Abb.7:* Abbildung selbst erstellt (Sourcecode beigelegt)

*Abb.8:* Abbildung selbst erstellt (Sourcecode beigelegt)

*Abb.9:* Abbildung selbst erstellt (Sourcecode beigelegt)

*Abb.10:* Abbildung selbst erstellt (Sourcecode beigelegt)

*Abb.11:* Abbildung selbst erstellt

*Abb.12:* Abbildung selbst erstellt (Sourcecode beigelegt)

*Abb.13:* Abbildung selbst erstellt (Sourcecode beigelegt)

*Abb.14:* Erstellt auf Basis des Python-Softwarecodes aus dem Buch: Rashid, Tariq, Make your own Mandelbrot, CreateSpace Independent Publishing Platform, 2014, 1. Auflage, Kapitel Fractals

*Abb.15:* Erstellt auf Basis des Python-Softwarecodes aus dem Buch: Rashid, Tariq, Make your own Mandelbrot, CreateSpace Independent Publishing Platform, 2014, 1. Auflage, Kapitel Fractals

## 6. Abbildungsverzeichnis

Seite

*Titelbild: Ausschnitt aus der Mandelbrotmenge*

*Abb.1: Iterationsverfahren als Rückkopplung mit Parameter 4*

*Abb.2: Beispiel für ein grafisches Verfahren zur Bestimmung von Fixpunkten 7*

*Abb.3: Iterationsfolge der logistischen Gleichung 9*

*Abb.4: Iterationsfolge mit konstantem Verhalten 10*

*Abb.5: Iterationsfolge mit periodischem Verhalten 10*

*Abb.6: Iterationsfolge mit periodischem Verhalten 11*

*Abb.7: Iterationsfolge mit chaotischem Verhalten 12*

*Abb.8: Das Feigenbaum-Diagramm im Intervall für r = [2.9; 4] 13*

*Abb.9: Das Feigenbaum-Diagramm im Intervall für r = [3.6; 3.8] 14*

*Abb.10: Das Feigenbaum-Diagramm mit ausgewählten Wachstumsfaktoren r 15*

*Abb.11: Darstellung einer komplexen Zahl z in Vektorform 16*

*Abb.12: Divergente Iterationsfolge, außerhalb der Mandelbrotmenge 19*

*Abb.13: Zyklische Iterationsfolge, innerhalb der Mandelbrotmenge 20*

*Abb.14: Die Mandelbrotmenge in der komplexen Zahlenebene 21*

*Abb.15:Ausschnitt der Mandelbrotmenge 23*

Ich habe diese Seminararbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benützt.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ort, Datum

1. Sasserlone.de [↑](#footnote-ref-1)
2. Voß, Herbert (1994), S. 15 [↑](#footnote-ref-2)
3. Glosbe.com [↑](#footnote-ref-3)
4. Bräuer, Kurt (2002), S. 29ff [↑](#footnote-ref-4)
5. Peitgen, Heinz-Otto / Richter, Peter H. (1986), S. 5 [↑](#footnote-ref-5)
6. Rashid, Tariq, (2014), Iteration [↑](#footnote-ref-6)
7. Peitgen, Heinz-Otto / Richter, Peter H. (1986), S. 6 [↑](#footnote-ref-7)
8. Bräuer, Kurt (2002), S. 29f [↑](#footnote-ref-8)
9. Voß, Herbert (1994), S. 117 [↑](#footnote-ref-9)
10. Rashid, Tariq, (2014), Divergence, Convergence [↑](#footnote-ref-10)
11. Rashid, Tariq, (2014), Iteration [↑](#footnote-ref-11)
12. Voß, Herbert (1994), S. 118 [↑](#footnote-ref-12)
13. Herrmann, Dietmar (1994), S. 31f [↑](#footnote-ref-13)
14. Peitgen, Heinz-Otto / Richter, Peter H. (1986), S. 6 [↑](#footnote-ref-15)
15. Herrmann, Dietmar (1994), S. 32 [↑](#footnote-ref-16)
16. Peitgen, Heinz-Otto / Richter, Peter H. (1986), S. 23ff [↑](#footnote-ref-17)
17. Rashid, Tariq, (2014), Periodic Cycles ‘Flip Flopping‘ [↑](#footnote-ref-18)
18. Voß, Herbert (1994), S. 127 [↑](#footnote-ref-19)
19. Peitgen, Heinz-Otto / Richter, Peter H. (1986), S. 24 [↑](#footnote-ref-20)
20. Bräuer, Kurt (2002), S. 38 [↑](#footnote-ref-21)
21. Bräuer, Kurt (2002), S. 39 [↑](#footnote-ref-22)
22. Voß, Herbert (1994), S. 127 [↑](#footnote-ref-23)
23. Herrmann, Dietmar (1994), S. 33 [↑](#footnote-ref-24)
24. Rashid, Tariq, (2014), Visualising Complex Numbers [↑](#footnote-ref-25)
25. Rashid, Tariq, (2014), Visualising Complex Numbers [↑](#footnote-ref-26)
26. Rashid, Tariq, (2014), Complex Numbers [↑](#footnote-ref-27)
27. Rashid, Tariq, (2014), Complex Numbers [↑](#footnote-ref-28)
28. Herrmann, Dietmar (1994), S. 226ff [↑](#footnote-ref-29)
29. Rashid, Tariq, (2014), Functions [↑](#footnote-ref-30)
30. Rashid, Tariq, (2014), Functions [↑](#footnote-ref-31)
31. Voß, Herbert (1994), S. 173 [↑](#footnote-ref-32)
32. https://de.wikipedia.org/wiki/Mandelbrot-Menge#Galerie\_der\_Iteration [↑](#footnote-ref-33)
33. Voß, Herbert (1994), S. 175 [↑](#footnote-ref-34)
34. Rashid, Tariq, (2014), Completing the atlas [↑](#footnote-ref-35)
35. Herrmann, Dietmar (1994), S. 229 [↑](#footnote-ref-36)
36. Rashid, Tariq, (2014), Fractals [↑](#footnote-ref-37)