

Heinrich-Böll-Gymnasium Troisdorf

Schuljahr 2022/2023

Die Mandelbrot-Menge

Mathematische Grundlagen und die visuelle Darstellung

verfasst von

Christoph Derszteler

Leistungskurs Mathematik

Betreuerin: Frau Dammers

Abgabetermin: 23.02.2023 12:00 Uhr CET

Inhaltsverzeichnis

1	Einleitung	3
2	Benötigtes Vorwissen	4
2.1	Komplexe Zahlen	4
2.1.1	Multiplikation & Addition von komplexen Zahlen	4
2.1.2	Graphische Darstellung komplexer Zahlen	5
2.2	Iterationen	6
3	Theoretische Grundlage	8
3.1	Mathematische Definition	8
3.2	Grafische Analyse	9
3.2.1	Farbbedeutung	10
3.2.2	Exemplarische Kartografierung	11
4	Praktische Betrachtung	13
4.1	Digitale Bildgenerierung	13
4.1.1	Korrelation zwischen Informationstechnik und Mathematik	13
4.1.2	Farbkodierung	14
4.2	Anderweitige Zusammenhänge	15
5	Fazit	16
6	Literatur und Quellen	17
7	Anhang	21

1 Einleitung

Die Mandelbrot-Menge ist durch ihre hübschen, ansehnlichen Darstellungen verglichen mit anderen mathematischen Phänomenen recht bekannt. Dies liegt jedoch nicht nur an ihrer rein visuellen Attraktivität, sondern vielmehr auch an Benoît Mandelbrot [A.1], dem Entdecker dieser Menge. Dieser sorgte mit seinen vielen Vorträgen und Büchern dafür, dass sich Fraktale, also selbstähnliche¹, geometrische Figuren mit gebrochener Dimension², vornehmlich die Mandelbrot-Menge, in der Bevölkerung weit verbreiteten³.

Obwohl die Natur mit ihren fraktal-ähnlichen Formationen wie dem Aufbau einer Schneeflocke, dem Verlauf eines Flusses oder die Verteilung von Baumästen [nna] die Inspiration für Mandelbrot war [ZK14], so liegt der Ursprung dieser Arbeit in den für manchen simpler erscheinenden, viel moderneren aber dennoch genauso spannenden, computer-generierten Videos⁴, die man im Internet finden kann. Mit unter anderem der Frage, wie diese Videos in Ansätzen generiert werden können und vielem weiteren beschäftigt sich diese Arbeit.

Dafür jedoch und zum vollen Verständnis der Mandelbrot-Menge ist Grundlagenwissen gewisser Themengebiete erforderlich, das in Kapitel 2 näher erörtert wird. Kapitel 3 beschäftigt sich daraufhin mit der rein mathematischen Betrachtung der Mandelbrot-Menge selbst und insbesondere mit der Analyse visueller Darstellungen dieser. Abschließend befasst sich diese Arbeit in Kapitel 4 mit der praktischen Anwendung der Mandelbrot-Menge in Form von Bildgenerierungen mithilfe von Computern als auch anderweitigen Zusammenhänge zwischen dem theoretischen Konzept der Mandelbrot-Menge und der realen Welt.

¹Das heißt, sich selbst wiederholend oder in ähnlicher Form erneut aufkommend.

²Im Vergleich zu zum Beispiel einem zwei-dimensionalen Viereck.

³[IBM11], Vgl. letzten Absatz.

⁴Vgl. bspw. [Tow17].

2 Benötigtes Vorwissen

Dieses Kapitel befasst sich mit den benötigten mathematischen Grundlagen, um der restlichen Arbeit folgen zu können. Dafür wird zunächst das Konzept der komplexen Zahlen als auch der für den weiteren Verlauf benötigter Umgang mit diesen erörtert. Darauffolgend wird das Prinzip und die Eigenschaften von Iterationen grob anhand eines Beispiels skizziert.

2.1 Komplexe Zahlen

Unter den komplexen Zahlen \mathbb{C} versteht man die nächst größere Zahlenmenge nach den reellen Zahlen \mathbb{R} , die zusätzlich zu einem Realteil auch einen sogenannten Imaginärteil besitzen. Sie werden im weiteren Verlauf in der kartesischen Form $z = a + bi$ dargestellt, wobei a der Realteil und bi der Imaginärteil ist. Der Buchstabe i steht hierbei für die imaginäre Einheit und ist definiert durch die Gleichung $i^2 = -1$.

2.1.1 Multiplikation & Addition von komplexen Zahlen

Viele Rechenoperationen mit komplexen Zahlen funktionieren anders, als man sie von den reellen oder natürlichen Zahlen gewohnt ist. Im Folgenden werden zwei dieser unterschiedlich funktionierenden Operationen vorgestellt:

Zur Addition zwei komplexer Zahlen addiert man den Realteil und den Imaginärteil getrennt voneinander und fügt diesen danach wieder zusammen [Lic02, S. 2]: $(a_1 + b_1i) + (a_2 + b_2i) = a_1 + a_2 + (b_1 + b_2)i$.

Um komplexe Zahlen zu multiplizieren, wendet man das Distributivgesetz an, indem man den zweiten Faktor ebenfalls in seinen Realteil und seinen Imaginärteil trennt und diese jeweils einzeln mit dem ersten Faktor multipliziert [Lic02, S. 2f.]. Die zwei entstehenden Produkte lassen sich dann wie oben beschrieben addieren. Bei der Multiplikation mit dem Imaginärteil multipliziert man unter anderem zwei imaginäre Elemente miteinander. Da $i^2 = -1$ gilt, entsteht durch

diese Multiplikation ein negatives, aber reales Produkt. Wie in A.2 gezeigt, gilt somit: $(a + bi)(c + di) = ac - bd + (bc + ad)i$

Das in der Mandelbrot-Menge häufig angewandte Quadrieren von komplexen Zahlen, lässt sich mit der kartesischen Form ebenfalls herleiten [A.3]. Für eine gegebene, zu quadrierende, komplexe Zahl $a + bi$ gilt somit: $(a + bi)^2 = a^2 - b^2 + 2abi$

Ein illustriertes Beispiel soll beide Rechenoperationen veranschaulichen:

$$\begin{aligned}
 & (-3 + 6i)^2 + (7 + (-4i)) \\
 = & ((-3 \cdot (-3)) - (6 \cdot 6) + ((6 \cdot (-3)) + (-3 \cdot 6))i) + (7 + (-4i)) \\
 = & (-27 + (-36i)) + (7 + (-4i)) \\
 = & -20 + (-40i)
 \end{aligned} \tag{2.1}$$

2.1.2 Graphische Darstellung komplexer Zahlen

Komplexe Zahlen können wie auch Zahlen anderer Zahlenmengen grafisch dargestellt werden. Da komplexe Zahlen jedoch sowohl aus einem Realteil und einem Imaginärteil bestehen, reicht eine Achse nicht aus, um diese darzustellen; stattdessen braucht man eine **Ebene**⁵. Diese komplexe Zahlenebene teilt den Realteil auf die waagerechte Achse und den Imaginärteil auf die vertikale Achse auf. Eine komplexe Zahl $z = a + bi$ besitzt somit die Koordinaten $P(a|b)$.

Eine komplexe Zahl lässt sich wie auch eine reelle Zahl absolut betrachten, wobei dieser absolute Wert ebenfalls als einen Abstand zum Ursprung zu betrachten ist [Lic02, S. 3]. Aufgrund dessen, dass eine komplexe Zahl aus zwei Komponenten besteht, lässt sich der Abstand über den Satz des Pythagoras berechnen:

$$|z|^2 = a^2 + b^2 \quad \text{beziehungsweise} \quad |z| = \sqrt{a^2 + b^2} \tag{2.2}$$

⁵Ebenfalls unter komplexer Zahlenebene und gaußsche Zahlenebene zu finden.

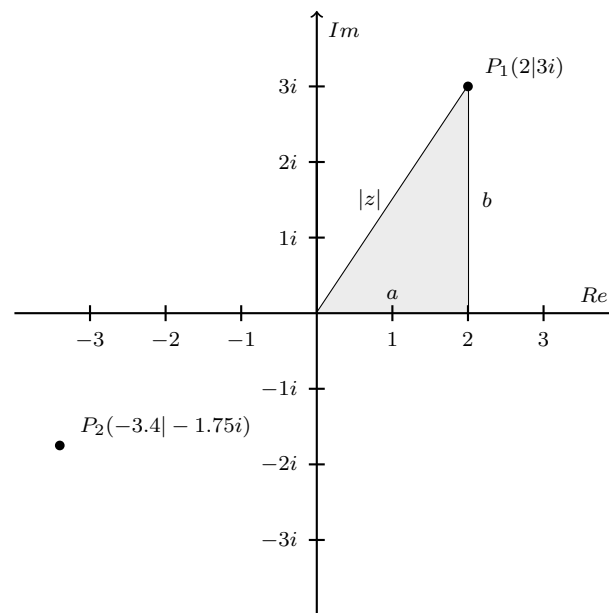


Abbildung 2.1: Komplexe Ebene mit den Punkten P_1 , und P_2 und dem absoluten Wert $|z|$ vom Punkt P_1 .

2.2 Iterationen

Iterationen beziehen sich in der Mathematik auf das Wiederholen einer bestimmten Prozedur beziehungsweise in diesem Fall einer Berechnung. Bei Funktionsiterationen iteriert (also wiederholt) man die Berechnung eines Funktionswerts mit dem Funktionsargument des vorherigen Funktionswerts: $z_1 = f(z_0)$, $z_2 = f(z_1)$, $z_3 = f(z_2)$, \dots , $z_n = f(z_{n-1})$.

Eine wichtige Eigenschaft von Iterationen ist die Entwicklung von z für $z \rightarrow \infty$. Dabei wird unterschieden, ob die Iteration divergent ist, das heißt gegen Unendlich verläuft („explodiert“), oder sich einem bestimmten Punkt, annähert. Letzteres bezeichnet man als einen beschränkten Verlauf.

Dieser Verlauf ist bei Iterationen, die ihre Ausgangswerte als neue Eingangswerte benutzen, schwer vorauszusagen. Dabei können ähnlich Funktionen bereits sehr unterschiedliche Entwicklungen aufweisen. Die Funktionen $f(z) = z^2 + c$ mit $z_0 = 0$ stellt beispielhaft die unterschiedlichen Verlaufsformen für $c_1 =$

1, $c_2 = -1$ und $c_3 = 0.5$ dar:

Iteration	Parameter für $f(z)$		
	$c_1 = 1$	$c_2 = -1$	$c_3 = 0.5$
1.	1	-1	0.5
2.	2	0	0.75
3.	5	-1	1.0625
4.	26	0	≈ 1.6289
5.	667	-1	\approx 3.1533
6.	$\approx 1.9 \times 10^{11}$	0	≈ 10.4433
7.	$\approx 3.9 \times 10^{22}$	-1	≈ 109.5625

Tabelle 2.1: $f(z)$ verläuft mit c_1 und c_3 divergent, hingegen ist der Verlauf für $f(z)$ mit c_2 beschränkt. Die dick markierten Zahlen sind für eine spätere Erwähnung dieser Tabelle relevant.

3 Theoretische Grundlage

Nachdem im vorherigen Kapitel die Grundlagen für die Mandelbrot-Menge erklärt wurden, befasst sich dieses Kapitel mit der rein mathematischen Betrachtung dieser Menge, indem diese zunächst fachlich korrekt definiert und im Anschluss grafisch analysiert wird.

3.1 Mathematische Definition

Die Mandelbrot-Menge wird mit der bereits im vorherigen Kapitel vorgestellten, komplexen Iteration $z_{n+1} = z_n^2 + c$ mit $z_0 = 0$ und einem variablen Wert für c [Wei18, S.477ff.] definiert. Dabei enthält die Menge alle komplexen Werte für c , mit denen die oben angegebene Iteration beschränkt ist. Mathematisch ist die Menge iterativ wie folgt definiert:

$$\mathbb{M} = \{c \in \mathbb{C} \mid \forall n \in \mathbb{N} : |f_c^n(z)| \leq 2\} \quad \text{mit} \quad f_c(z) = z^2 + c; \quad z, c \in \mathbb{C} \quad (3.1)$$

Wie in der Definition zu sehen, wird der Funktionswert der gegen Unendlich strebenden n -ten Iteration, ausgedrückt durch $f^n(z)$, absolut betrachtet, was bedeutet, dass die Funktion symmetrisch zur reellen Achse ist.

Ebenfalls zu betrachten ist die Einschränkung auf Funktionswerte ≤ 2 , denn für alle Funktionswerte, die sich in der oben genannten Iteration ergeben und > 2 sind, lässt sich das jeweilige c aus der Mandelbrot-Menge ausschließen. Obwohl der gesamte Beweis dessen über den Rahmen dieser Arbeit hinausginge, so soll dennoch angemerkt werden, dass mithilfe der Dreiecksungleichung und vollständiger Induktion unter der Voraussetzung von $|z_n| > 2$ und $|z_n| > |c|$ folgende Ungleichung, die eine divergente Entwicklung repräsentiert, $\frac{|z_{n+1}|}{|z_n|} > 1$ bewiesen werden kann [MH97], wobei sich zusätzlich zeigen lässt, dass für alle Werte von $|c| > 2$ nach spätestens 2 Iterationen gilt: $z_2 = |c^2 + c| \geq |c|^2 - |c| > 2$.

Es befinden sich deshalb alle Werte für c als auch somit die grafische Darstellung der Mandelbrot-Menge in einem Einheitskreis mit dem Radius 2 [Vgl. A.4].

3.2 Grafische Analyse

Im Folgenden soll der grundlegende Aufbau der in einem kartesischen Diagramm entstehenden Formation der Mandelbrot-Menge erörtert als auch eine Erklärung zur Farbbedeutung gegeben werden. Zusätzlich zeigt dieses Unterkapitel mit einer exemplarischen Kartografierung verschiedene, sich wiederholende Bereiche der visuellen Darstellung auf.

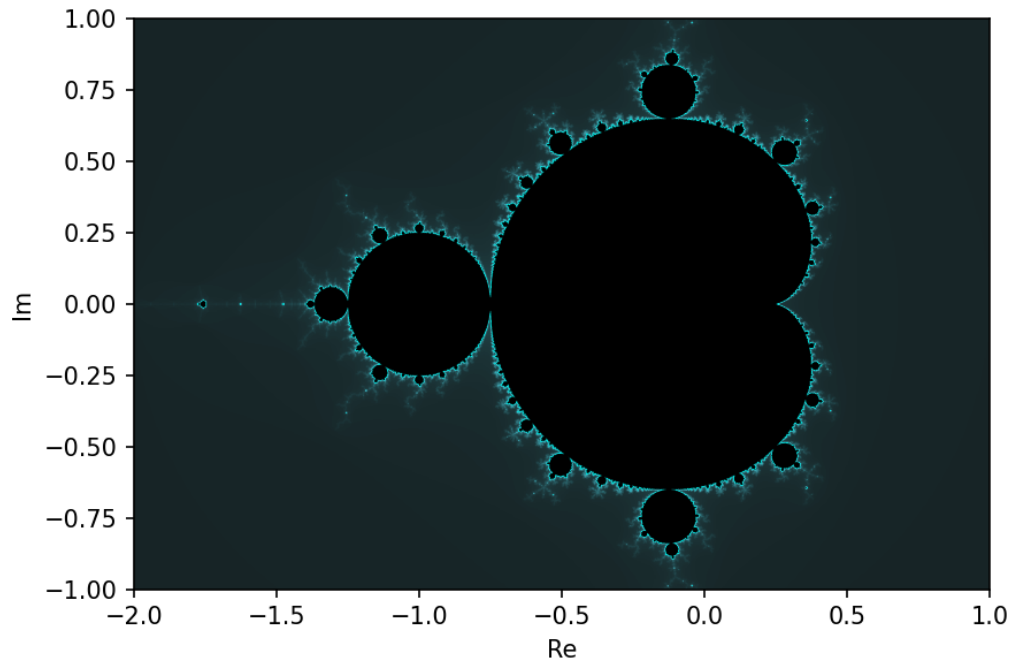


Abbildung 3.1: Exemplarische Darstellung der Mandelbrot-Menge⁶

⁶Generiert durch den Code A.Last mit einer „RESOLUTION“ von 2000 und einer „MAX_ITERATIONS“ von 500

Die hier zu sehende Grafik entspricht der Darstellung der Mandelbrot-Menge in einer komplexen Zahlenebene und wird aufgrund seiner Form „Apfelmännchen“ genannt. Die zu sehenden, schwarz gefärbten Pixel repräsentieren einen jeweiligen Wert für c , der sich in der Mandelbrot-Menge befindet. Auffällig ist bei erster Betrachtung, dass sich neben der großen, einheitlichen Struktur in der Mitte, deutlich kleinere, ähnlich aussende Formationen um den eigentlichen Hauptkörper dem Apfelmännchen, zum Beispiel im negativen Teil der reellen Achse, erkennen lassen. Diese werden **Satelliten** genannt und existieren aufgrund der Selbstähnlichkeit der Mandelbrot-Menge in unendlicher Stückzahl - und zwar nicht nur für den Hauptkörper, sondern auch für jeden Satelliten selbst [LP18].

3.2.1 Farbbedeutung

Im Gegensatz zu den ersten grafischen Darstellungen der Mandelbrot-Menge, auf denen, aufgrund ihrer deutlich geringeren Auflösung, kleine Satelliten als Druckfehler gewertet wurden⁷ und die mit ihrem schwarz-weißen Druck nur zwischen Werte für c **in** und **außerhalb** der Mandelbrot-Menge unterschieden, besitzt die oben dargestellte Figur (3.1) einen Farbverlauf. Dieser, in diesem Fall türkisblaue, Farbgradient gibt an, wie viele Iterationen es benötigte, um festzustellen, ob das jeweilige c außerhalb der Mandelbrot-Menge liegt. Dabei gilt, dass je heller der Pixel beziehungsweise c ist, desto mehr Iterationen hat es benötigt, um c aus \mathbb{M}

⁷Dies ist eine recht amüsante Anekdote: Während den frühesten Forschungen, die Benoît Mandelbrot in den 1970er bei IBM anstellte, war das Drucken deutlich mühseliger und aufwendiger, als es heutzutage ist. Deshalb existierte ein ganzes Abteil nur für die Herstellung und Bearbeitung von Drucks, die - da es damals häufig vorkam - kleine Satelliten am Rande der ersten Darstellungen [A.5] gutgemeint wegretuschierten. Die ersten Bilder, die Herr Mandelbrot also erhielt, verwunderten ihn sehr und er war äußerst aufgebracht, als er von der tolpatschigen Wahrheit erfuhr [Num19].

auszuschließen⁸. Deshalb existiert ein hell erscheinender Rand um die schwarzen Formationen, da es für diese Werte außerhalb von \mathbb{M} , die sehr nah an tatsächlichen Werten für \mathbb{M} sind, es viele Iterationen benötigt, um diese auszuschließen.

Neben dieser einen, verhältnismäßig simplen und dementsprechend auf den ersten Blick aussagekräftigeren Farbkodierung, existiert eine Vielzahl teils deutlich umfangreicheren Algorithmen, mit denen sich jedoch eine auf das menschliche Auge ansprechendere Farbgestaltung erzielen lässt. Diese werden in der Sektion 4.1.2 genauer beschrieben. Eine Reihe an solchen komplexere Farbverläufe benutzenden Beispielen, auf die im nächsten Unterkapitel Bezug genommen wird, lässt sich in A.6 betrachten.

3.2.2 Exemplarische Kartografierung

Die hingegen von der Farbe unabhängigen, entstehenden Formationen der Mandelbrot-Menge, die teilweise erst bei sehr kleinen Ausschnitten erkennbar sind, sind kartografiert und teilweise, wegen einer gewissen Ähnlichkeit, nach Objekten aus der realen Welt benannt. So bezeichnet man die größte kreisförmige Kardioide oder auch „Knospe“ als „Körper“ (wobei dieser genauer unterteilt werden kann) und die daran angrenzende Kardioide als „Kopf“⁹.

Obwohl man jeden Punkt beziehungsweise jeden Ausschnitt einzeln beliebig detailliert analysieren kann, werden aufgrund der Selbstähnlichkeit Elemente mit ähnlichem oder gleichen Aufbau erneut aufkehren und dementsprechend gleich benannt. Im Folgenden soll beispielhaft ein Ausschnitt des in diesem Video [Bey17] gezeigten „Tal der Seepferdchen“ analysiert werden:

Die Spalte zwischen Kopf und Körper wird „Tal der Seepferdchen“ genannt¹⁰ [Rob10], da bei Vergrößerung dieses Ausschnitts sich unter anschaulicher Farb-

⁸Vgl. z.B. dick markierten Werte für $c_1 = 1$ und $c_3 = 0.5$ in Tabelle 2.1. c_1 ließ sich nach der 3. Iteration aus der Mandelbrot-Menge ausschließen, hingegen war dies bei c_3 erst nach der 5. Iteration der Fall.

⁹Vgl. A.7

¹⁰Vgl. A.6.1

kodierung auf der rechten Seite Seepferdchen-ähnliche Formationen erkennen lassen¹¹. Vergrößert man die Sicht auf das Seepferdchen-Tal stark, so lassen sich, neben weiteren (teils deformierten) Satelliten¹², bei genauerer Betrachtung des „Seepferdchenschwanzes“ ein Misiurewicz-Punkt erkennen¹³. Dieser Misiurewicz-Punkt zeigt ebenfalls die Selbstähnlichkeit der Mandelbrot-Menge auf, da dieser Punkt sich neben einer Drehung kaum von der eigentlichen Mandelbrot-Menge unterscheidet [Lei89]. Vergrößert man diesen Punkt weiter, so findet man erneut einen im Vergleich zum Apfelmännchen sehr ähnlichen aussehenden Satelliten¹⁴.

¹¹Vgl. A.6.2

¹²Vgl. A.6.3

¹³Vgl. A.6.4

¹⁴Vgl. A.6.5

4 Praktische Betrachtung

Im Folgenden werden die im letzten Kapitel erklärten mathematischen Grundlagen unter Realbedingungen betrachtet, indem die dadurch entstehenden Grenzen spezifiziert als auch die konkreten Umsetzungen exemplarisch dargestellt werden. Dafür beschäftigt sich diese Arbeit zunächst mit der informationstechnischen Herangehensweise zur Generierung bereits dargestellter Bilder. Daraufgehend sollen zusätzliche Informationen zu anderweitigen mit der Mandelbrot-Menge in Verbindung stehenden Phänomen gegeben werden.

4.1 Digitale Bildgenerierung

Im Folgenden werden informationstechnische Aspekte hinsichtlich der Bildgenerierung der Mandelbrot-Menge untersucht. Dafür wird zunächst die Funktionsweise als auch die im Vergleich zur Mathematik entstehenden Probleme erläutert. Im Anschluss werden verschiedene Herangehensweisen und Algorithmen zur Farbkodierung grob skizziert.

4.1.1 Korrelation zwischen Informationstechnik und Mathematik

Die im letzten Kapitel vorgestellte und analysierte Abbildung 3.1 wurde, wie es in der Beschriftung steht, mit dem Code aus [A.Last] generiert. Im Vergleich zu den theoretischen Überlegungen, ist die informationstechnische Herangehensweise dabei sehr ähnlich:

Grundlegend wird jeder Pixel mit seiner x - und y -Koordinate des zu generierenden Ausschnitts einer komplexen Zahl zugeordnet. Dabei entspricht (in der regulären Darstellung) die x -Koordinate dem Realteil a und die y -Koordinate dem Imaginärteil b , sodass eine komplexe Zahl $c = a + bi$ in einem Pixel P durch $x = a$ und $y = b$ ausgedrückt werden kann.

Das naheliegendste und offensichtlichste Problem, neben vielen rein informationstechnischen Optimierungsaufgaben, ergibt sich bei der Berechnung, ob es sich

beim jeweiligen Pixel beziehungsweise bei der jeweiligen komplexen Zahl, um ein Element aus \mathbb{M} handelt. Denn im Gegensatz zu der mathematischen Betrachtung, die die Iterationsanzahl n bei $f^n(z)$ gegen unendlich laufen lässt, ist dies in der konkreten Umsetzung nicht möglich. Deshalb setzt man einen Grenzwert an Iterationsdurchläufen, ab dem, sofern der zu überprüfende Wert für c noch nicht aus der Mandelbrot-Menge ausgeschlossen wurde, dieser als Element von \mathbb{M} angenommen wird.

Folglich bestimmt dieser Wert indirekt die Auflösung beziehungsweise Genauigkeit des zu generierenden Bilds und muss deshalb bei kleineren Ausschnitten besonders hoch sein, da dabei ein Unterschied zwischen komplexen Zahlen ausgemacht werden muss, dessen Werte sich lediglich um geringe Nachkommastellen unterscheiden. Die Auswirkungen dieser Iterationsgrenze sind durch die Bildergalerie A.8, die generierte Bilder der Mandelbrot-Menge mit unterschiedlich (niedrigen) Grenzen darstellt, anschaulich visualisiert.

4.1.2 Farbkodierung

Um die in der Sektion 3.2.1 erklärten Farben zu kodieren, eignet sich zunächst das HSV (Hue, Saturation, Value) Farbmodel, womit sich mithilfe von prozentualen Werten eine Farbveränderung hinsichtlich der Helligkeit als auch der Farbsättigung erzielen lässt. Dies wird anhand der benötigten Iterationen n und der Iterationsgrenze m für jeden komplexen Wert c aus \mathbb{M} berechnet [Rob22]. Als Beispiel sei Hue mit 186° und Vue mit 100% gegeben, woraus sich für jedes c folgendes HSV ergibt: $HSV(186^\circ, \frac{n}{m} \cdot 100\%, 100\%)$.

Diese Farbkodierung ist aufgrund ihrer Simplizität in ihrer beschriebenen Form direkt informationstechnisch umsetzbar, jedoch eignet sich solch eine Herangehensweise aus Performance- beziehungsweise Optimierungsgründen nicht für jede berechnete komplexe Zahl c , wenn es sich bei der Farbzusammenstellung um einen komplexeren Zusammenhang wie in den Elementen der Bildergalerie A.6 handelt. Um dieses Problem zu lösen, erstellt man sogenannte Colormaps (engl.: Farb-

paletten), wie zum Beispiel A.9.1 aus den Werten des oberen Beispiels. Für ein jeweiliges c wird mithilfe solcher Farbpaletten eine Farbe erneut anhand dessen Verhältnis von benötigten Iterationen n zu der Iterationsgrenze m ermittelt, dabei existieren jedoch bereits alle vorkommenden Farben, was somit die Berechnung des aufwendigeren Farbgenerierungsprozesses¹⁵ von dem Berechnungsprozess der eigentlichen Mandelbrot-Menge entkoppelt. In A.9.2 ist die für alle für die Arbeit generierten Abbildungen benutzte Colormap zu sehen, die im Vergleich zur vorherigen dargestellten Palette mehr als zwei (insgesamt 4) Ankerpunkte¹⁶ besitzt, wobei diese, im Gegensatz zu einer komplexen¹⁷ Colormap [z.B. A.9.3], mit der man visuell anschaulichere Bilder generieren kann, weiterhin recht minimal ist.

4.2 Anderweitige Zusammenhänge

Neben den bisher bereits mehrfach angesprochenen „Mandelbrot Videos“¹⁸, die ebenfalls einen digitalen Zusammenhang zu der Mandelbrot-Menge besitzen und die trotz ihres Primärziels, das normalerweise die Unterhaltung ist, eine ausgesprochene Komplexität besonders hinsichtlich ihrer Optimierungsmöglichkeiten, aufweisen, der man problemlos eine eigene Arbeit dedizieren könnte, gibt es, wie in der Einleitung erwähnt, viele biologische Zusammenhänge zwischen der Natur und Fraktalen: Obwohl es über den Umfang dieser Arbeit hinausgehen würde, auf diesen Bereich der Biologie genauer einzugehen, so soll dennoch angemerkt werden, dass Fraktale wie die Mandelbrot-Menge der Natur ermöglichen, komplexe Anordnungen und Konstruktionen möglichst ressourcen- und platzschonend darzustellen, da Informationen sowohl für beispielsweise den Baumstamm als auch einen kleinen Ast aufgrund der, wie bei Satelliten der Mandelbrot-Menge auftretenden, Selbstähnlichkeit von Fraktalen nur einmal gespeichert werden müssen.

¹⁵Also die Auswahl und korrekte Zusammenstellung der Farben einer Farbpalette

¹⁶Hier: Eindeutige HSV-Farben, die in gewissen Abständen auf einer Palette platziert werden können und zwischen denen ein Gradient entsteht.

¹⁷Eine Farbpalette mit vielen Ankerpunkten

¹⁸Auch unter „Zoomvideos“ oder „Bilderfahrten“ zu finden

5 Fazit

6 Literatur und Quellen

- [Bey05a] Wolfgang Beyer. *Partial view of the Mandelbrot set. Step 1 of a zoom sequence: Gap between the "head" and the "body" also called the Seahorse valley*". 12. Sep. 2005. URL: https://de.wikipedia.org/wiki/Datei:Mandel_zoom_01_head_and_should.jpg (besucht am 30.01.2023).
- [Bey05b] Wolfgang Beyer. *Partial view of the Mandelbrot set. Step 2 of a zoom sequence: On the left double-spirals, on the right Seahorses*". 12. Sep. 2005. URL: https://de.wikipedia.org/wiki/Datei:Mandel_zoom_02_seehorse_valley.jpg (besucht am 30.01.2023).
- [Bey05c] Wolfgang Beyer. *Partial view of the Mandelbrot set. Step 3 of a zoom sequence: Seahorse upside down. Its "body" is composed by 25 spokes consisting of 2 groups of 12 spokes each and one spoke connecting to the main cardioid. These 2 groups can be attributed by some kind of metamorphosis to the 2 fingers of the upper hand of the Mandelbrot set. Therefore the number of spokes increases from one Seahorse to the next by 2. The "hub" is a so called Misiurewicz point. Between the upper part of the body and the tail a distorted satellite can be recognized*. 12. Sep. 2005. URL: https://de.wikipedia.org/wiki/Datei:Mandel_zoom_03_seehorse.jpg (besucht am 30.01.2023).
- [Bey05d] Wolfgang Beyer. *Partial view of the Mandelbrot set. Step 4 of a zoom sequence: The central endpoint of the Seahorse tail is also a Misiurewicz point*. 12. Sep. 2005. URL: https://de.wikipedia.org/wiki/Datei:Mandel_zoom_04_seehorse_tail.jpg (besucht am 30.01.2023).
- [Bey05e] Wolfgang Beyer. *Partial view of the Mandelbrot set. Step 8 of a zoom sequence: Antenna of the satellite. Several satellites of second order can be recognized*. 12. Sep. 2005. URL: <https://de.wikipedia.org/>

- wiki/Datei:Mandel_zoom_08_satellite_antenna.jpg (besucht am 30.01.2023).
- [Bey17] Wolfgang Beyer. *Zoomfahrt in die Mandelbrotmenge*. 16. Juni 2017. URL: <https://www.wolfgangbeyer.de/chaos/mandelzoom.htm> (besucht am 29.01.2023).
- [Elp07] Elphaba. *Command-line depiction of the Mandelbrot set, just like the picture Brooks and Matelski included in their article of 1978 on Kleinian groups*. 2. Feb. 2007. URL: <https://en.wikipedia.org/wiki/File:Mandel.png> (besucht am 30.01.2023).
- [Gai97] Raphael Gaillarde. *Benoit Mandelbrot, mathematician, inventor of fractals*. 9. Feb. 1997. URL: <https://www.gettyimages.co.uk/detail/news-photo/benoit-mandelbrot-mathematician-inventor-of-fractals-in-news-photo/110137025> (besucht am 22.01.2023).
- [IBM11] IBM. *Fractal Geometry*. 21. Mai 2011. URL: <https://www.ibm.com/ibm/history/ibm100/us/en/icons/fractal/> (besucht am 16.01.2023).
- [Lei89] TAN Lei. “Similarity Between the Mandelbrot Set and Julia Sets”. In: *Communications in Mathematical Physics*. 10. Juli 1989, S. 587–617. URL: <https://arxiv.org/pdf/1410.6729v2.pdf> (besucht am 29.01.2023).
- [Lic02] Klaus Lichtenegger. *Komplexe Analysis*. Mai 2002. URL: <https://www.math.tugraz.at/~lichtenegger/kompan.pdf> (besucht am 16.01.2023).
- [LP18] Luna Lomonaco und Carsten Lunde Petersen. “On quasi-conformal (in-) compatibility of satellite copies of the Mandelbrot set”. In: (10. Okt. 2018). URL: <https://arxiv.org/pdf/1505.05422.pdf> (besucht am 29.01.2023).

- [MH97] Robert P. Munafo und Mike Hurley. “Escape Radius, Mu-Ency at MROB”. In: (19. Sep. 1997). URL: <http://mrob.com/pub/muency/escaperadius.html> (besucht am 24.01.2023).
- [MSC16] Arun Mahanta, Hemanta Sarmah und Gautam Choudhury. “MANDELBROT SET, THE MESMERIZING FRACTAL WITH INTEGER DIMENSION”. In: *International Journal of Applied Mathematics and Statistical Sciences* 6 (Dez. 2016), S. 1–18. URL: https://www.researchgate.net/figure/The-Body-B-and-Head-H-of-the-Mandelbrot-Set-The-surface-of-these-two-parts-are-covered_fig6_310798765 (besucht am 29.01.2023).
- [nna] Mike (nnart). *Fractals in Nature*. How Do Fractals Appear in Nature? 10 Outstanding Examples. URL: <https://nnart.org/fractals-in-nature/> (besucht am 16.01.2023).
- [Num19] Numberphile. *What’s so special about the Mandelbrot Set? - Numberphile*. 18. Apr. 2019. URL: <https://youtu.be/FFftmWSzgmK?t=602> (besucht am 30.01.2023).
- [Rob10] Munafo Robert P. “Seahorse Valley, Mu-Ency at MROB”. In: (7. Sep. 2010). URL: <http://www.mrob.com/pub/muency/seahorsevalley.html> (besucht am 30.01.2023).
- [Rob22] Munafo Robert P. “Color, Mu-Ency at MROB”. In: (12. Dez. 2022). URL: <http://www.mrob.com/pub/muency/color.html>.
- [Tow17] Maths Town. *Eye of the Universe*. Eye of the Universe - Mandelbrot Fractal Zoom (e1091) (4k 60fps). 28. Aug. 2017. URL: <https://www.youtube.com/watch?v=pCpLWbHVNhk> (besucht am 16.01.2023).
- [Tro] Heinrich Böll Gymnasium Troisdorf. *HBG Logo*. In der Titelseite zu finden. URL: https://www.hbg-troisdorf.de/images/hbg_logo_web.png (besucht am 18.12.2022).

-
- [Wei18] Edmund Weitz. *Konkrete Mathematik (nicht nur) für Informatiker: Mit vielen Grafiken und Algorithmen in Python*. 1. Aufl. Springer Spektrum, 8. Aug. 2018. 942 S. ISBN: 978-3-658-21565-1.
- [ZK14] Iris Zink und Hanna Kotarba. *Der kosmische Code*. Der kosmische Code. 28. Sep. 2014. URL: <https://www.zdf.de/dokumentation/terra-x/faszination-universum-der-kosmische-code-mit-harald-lesch-100.html> (besucht am 16.01.2023).

7 Anhang

A.1:



Abbildung A.1: Benoît Mandelbrot 1997 in Frankreich [Gai97].

A.2:

$$\begin{aligned}
 & z_1 \cdot z_2 \\
 & (a + bi) \cdot (c + di) \\
 & = c(a + bi) + di(a + bi) \\
 & = ac + bci + adi + bdi^2 \\
 & = ac + bci + adi - bd \\
 & = ac - bd + (bc + ad)i
 \end{aligned} \tag{A.2}$$

A.3:

$$\begin{aligned}
 & z_1^2 = z_1 \cdot z_1 \\
 & = (a + bi) \cdot (a + bi) \\
 & = a \cdot (a + bi) + bi \cdot (a + bi) \\
 & = a^2 + abi + abi - b^2 \\
 & = a^2 - b^2 + 2abi
 \end{aligned} \tag{A.3}$$

A.4:

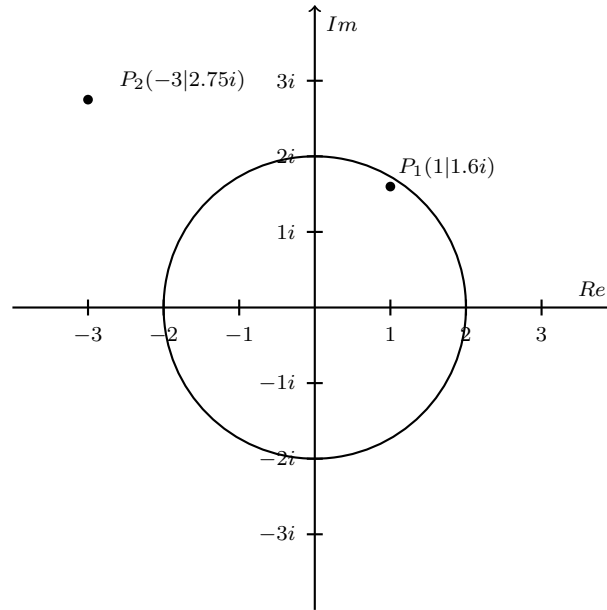


Abbildung A.4: Einheitskreis mit dem Radius 2. Zu sehen ist der Punkt P_1 , der im Einheitskreis liegt und Punkt P_2 , der außerhalb des Einheitskreises liegt.

A.5:

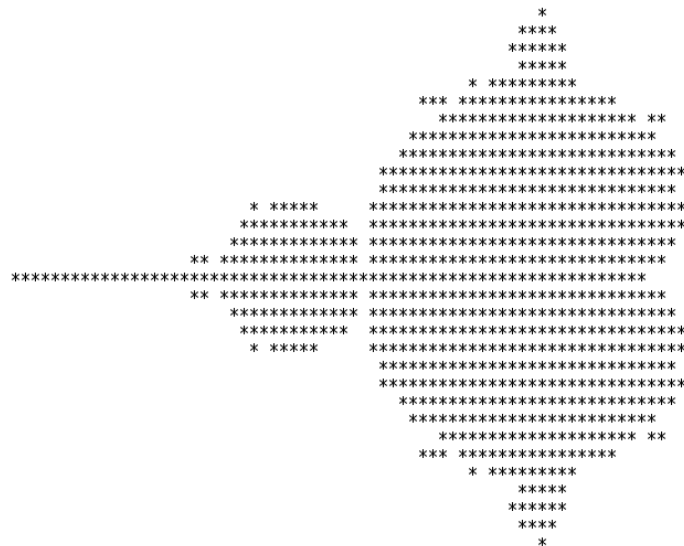


Abbildung A.5: Darstellung der ersten Drucks der Mandelbrot-Menge [Elp07].

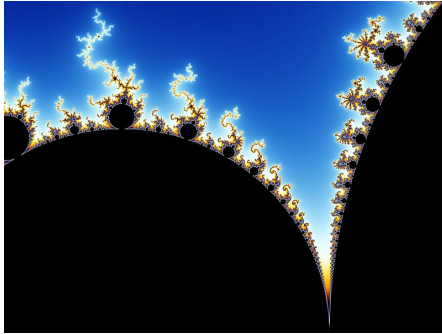
A.6:

Abbildung A.6.1: Spalte zwischen Kopf und Körper [Bey05a]

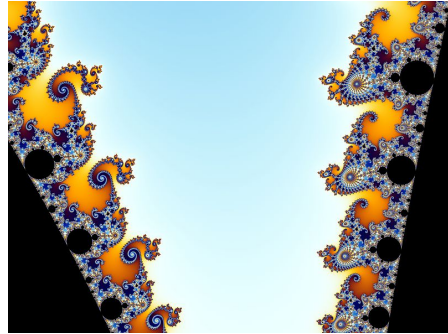


Abbildung A.6.2: "Tal der Seepferdchen", [Bey05b]

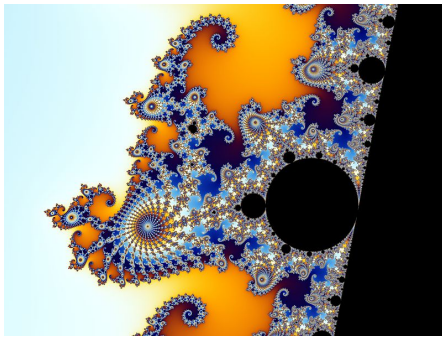


Abbildung A.6.3: Rechts ein deformierter Satellit und links Misiurewicz-Punkt [Bey05c]

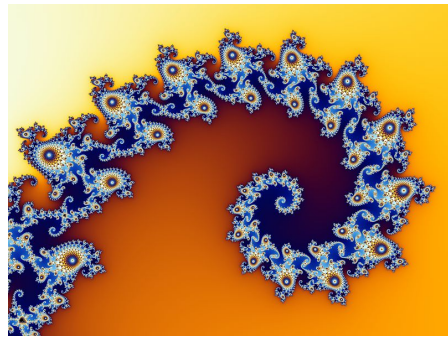


Abbildung A.6.4: Misiurewicz-Punkt [Bey05e]

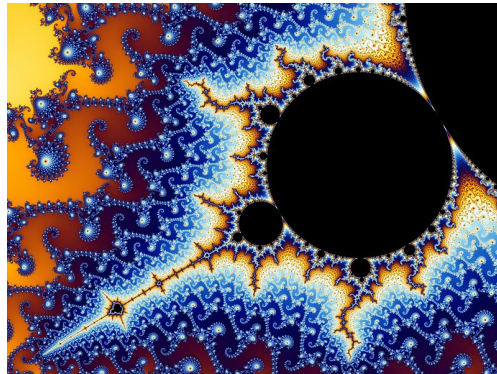


Abbildung A.6.5: Satellit mit ähnlicher Struktur wie das Apfelmännchen [Bey05d]

A.7:

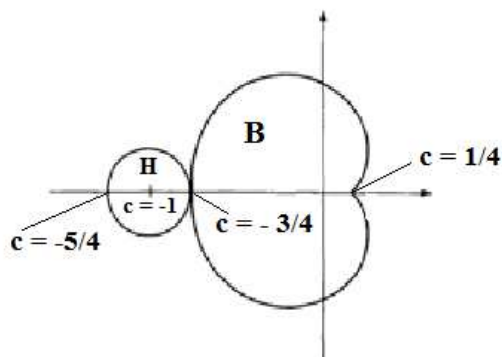


Abbildung A.7: Körper (B) und Kopf (H) der Mandelbrot-Menge [MSC16].

A.8:

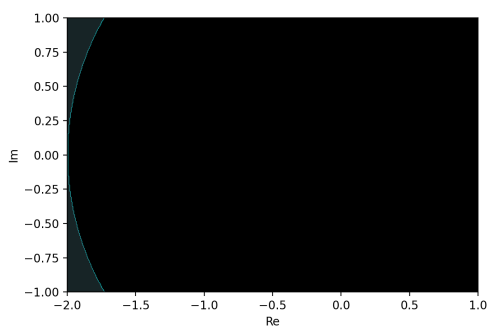


Abbildung A.8.1: Generiertes Bild mit einer Iterationsgrenze von 1

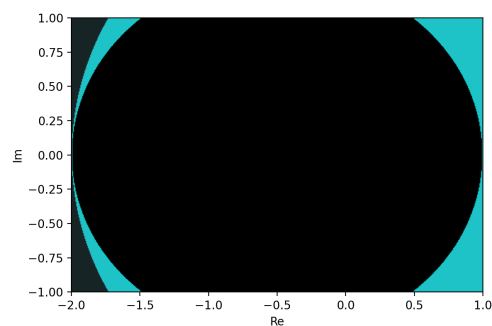


Abbildung A.8.2: Generiertes Bild mit einer Iterationsgrenze von 2

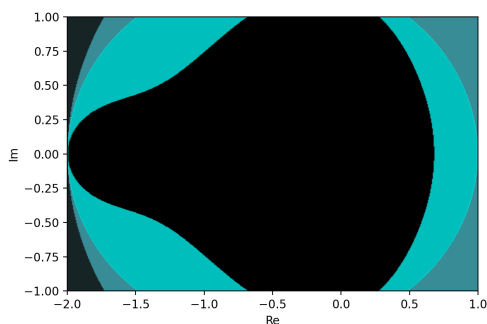


Abbildung A.8.3: Generiertes Bild mit einer Iterationsgrenze von 3

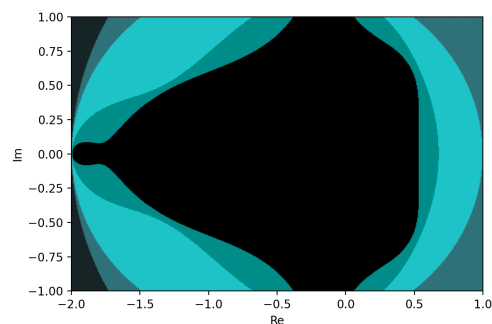


Abbildung A.8.4: Generiertes Bild mit einer Iterationsgrenze von 4

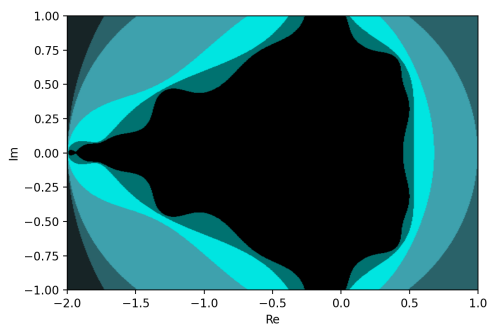


Abbildung A.8.5: Generiertes Bild mit einer Iterationsgrenze von 5

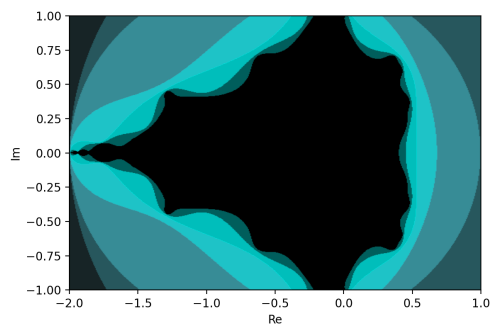


Abbildung A.8.6: Generiertes Bild mit einer Iterationsgrenze von 6

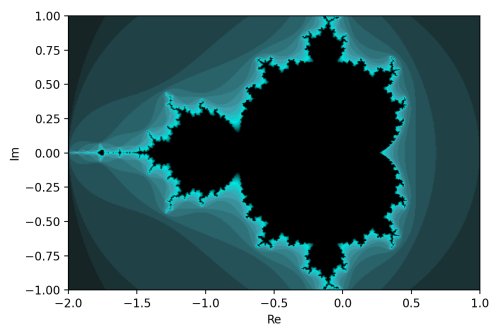


Abbildung A.8.7: Generiertes Bild mit einer Iterationsgrenze von 20

A.9:



Abbildung A.9.1: Simple HSV-Colormap mit einem linearen Farbverlauf



Abbildung A.9.2: Komplexe Colormap mit vier verschiedenen Ankerpunkten



Abbildung A.9.3: Komplexe Colormap mit vielen unterschiedlichen Ankerpunkten

A.Last:

```

import matplotlib.pyplot as plot
import numpy

def divergent_iterations(real, imaginary, max_iterations):
    c = complex(real, imaginary)
    z = 0.0j

    for iteration in range(max_iterations):
        z = z ** 2 + c
        if abs(z) > 2:
            return iteration

    return max_iterations

def generate_pixel_iterations_map(resolution, max_iterations):
    xs = resolution
    ys = resolution

    pixels = numpy.zeros([xs, ys])

    for x, real in enumerate(numpy.linspace(SCOPE[0], SCOPE[1], num=
        xs)):
        for y, imaginary in enumerate(numpy.linspace(SCOPE[2], SCOPE
            [3], num=ys)):
            pixels[x, y] = divergent_iterations(real, imaginary,
                max_iterations)
    return pixels

def draw_plot(pixels, color_map, size):
    plot.figure(dpi=size)
    plot.imshow(pixels.T, cmap=color_map, interpolation="bilinear",
        extent=SCOPE)
    plot.xlabel("Re")
    plot.ylabel("Im")

# This value defines the quality/resolution of the image. For quick
# generations,
# 500 is recommended. For higher quality images, use a value between
# 2000–5000
RESOLUTION = 500
# This value defines the image size. It should be updated accordingly
# with the
# resolution. An image size above 300 should often not be necessary.
IMAGE_SIZE = min(RESOLUTION * 1.25, 300)
# This value defines the scope/detail of the image. Default: [-2, 1,
# -1, 1]
SCOPE = [-2, 1, -1, 1]
# This value should be changed if the scope is not the default.
# Default: 100
MAX_ITERATIONS = 100
# This value changes the color mapping of the generated image. For a
# small
# scope, "twilight" might be interesting. The default value uses
# a custom generated color map
COLOR_MAP = {
    'red': (

```

```

        (0.0, 0.09019607843137255, 0.09019607843137255),
        (0.4, 0.24313725490196078, 0.24313725490196078),
        (0.6, 0.0, 0.0),
        (1.0, 0.0, 0.0)
    ),
    'green': (
        (0.0, 0.1411764705882353, 0.1411764705882353),
        (0.4, 0.6313725490196078, 0.6313725490196078),
        (0.6, 0.8980392156862745, 0.8980392156862745),
        (1.0, 0.0, 0.0)
    ),
    'blue': (
        (0.0, 0.14901960784313725, 0.14901960784313725),
        (0.4, 0.6784313725490196, 0.6784313725490196),
        (0.6, 0.8823529411764706, 0.8823529411764706),
        (1.0, 0.0, 0.0)
    )
}

pixels = generate_pixel_iterations_map(RESOLUTION, MAX_ITERATIONS)
draw_plot(pixels, COLOR_MAP, IMAGE_SIZE)
plot.show()

```

Code 1: Python-Code zur Generierung der Abbildung 3.1