

**MASARYKOVA UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA
ÚSTAV MATEMATIKY A STATISTIKY**

Bakalářská práce

BRNO 2022

JAN MACHARÁČEK

Juliovy množiny

Bakalářská práce

Jan Macharáček

Bibliografický záznam

Autor:	Jan Macharáček Přírodovědecká fakulta, Masarykova univerzita Ústav matematiky a statistiky
Název práce:	Juliovy množiny
Studijní program:	Matematika
Studijní obor:	Finanční a pojistná matematika
Vedoucí práce:	RNDr. Veronika Eclerová, Ph.D.
Akademický rok:	2021/2022
Počet stran:	x + 82
Klíčová slova:	Fraktál; Juliovy množiny; Mandelbrotova množina; Komplexní dynamika; Nelineární dynamický systém; Iterace; Algoritmus únikového času; Metoda odhadu hranice; Metoda inverzní iterace; Webová aplikace; JavaScript

Bibliographic Entry

Author:	Jan Macharáček Faculty of Science, Masaryk University Department of Mathematics and Statistics
Title of Thesis:	Julia sets
Degree Programme:	Mathematics
Field of Study:	Financial and acturial mathematics
Supervisor:	RNDr. Veronika Eclerová, Ph.D.
Academic Year:	2021/2022
Number of Pages:	x + 82
Keywords:	Fractal; Julia sets; Mandelbrot set; Complex dynamics; Non-linear dynamical system; Iteration; Escape time algorithm; Distance estimator method; Inverse iteration method; Web application; JavaScript

Abstrakt

V této bakalářské práci se věnujeme Juliovým množinám. V teoretické části jsou shrnutý základní pojmy z fraktální geometrie a komplexní analýzy. Dále jsou zde popsány důležité vlastnosti Juliových množin, které jsou využity v praktické části této práce. Následně se práce zabývá souvislostmi mezi Juliovými množinami a Mandelbrotou množinou. Praktická část spočívá v naprogramování webové aplikace s využitím jazyka JavaScript ve spojení s HTML a CSS pro vykreslení Juliových množin pomocí různých vizualizačních metod, která je navíc doplněna o výtahy z textu teoretické části, a tak je možné tuto aplikaci využít i jako doplňující výukovou pomůcku pro názorné demonstrování diskutovaných vlastností.

Abstract

In this thesis, we focus on Julia sets. The theoretical part summarizes the basic concepts of fractal geometry and the complex analysis. Next, important properties of the Julia sets, which are used in the practical part of this thesis are described. Then, we deal with the connections between the Julia sets and the Mandelbrot set. The main aim of the practical part is to develop a web application using JavaScript with HTML and CSS for visualization of the Julia sets by various methods, supplemented by key sections from the text of the theoretical part therefore it is possible to use this application as an additional teaching tool for illustrative demonstration of the discussed properties.

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Akademický rok: 2021/2022

Ústav:	Ústav matematiky a statistiky
Student:	Jan Macharáček
Program:	Matematika
Specializace:	Finanční a pojistná matematika

Ředitel ústavu PřF MU Vám ve smyslu Studijního a zkušebního řádu MU určuje bakalářskou práci s názvem:

Název práce:	Juliovy množiny
Název práce anglicky:	Julia sets
Jazyk závěrečné práce:	čeština

Oficiální zadání:

Cílem práce je definování a popis základních vlastností Juliových množin. Student má dále za úkol uvést souvislost Juliových množin a Mandelbrotovy množiny. Praktická část v práce bude spočívat v navržení a naprogramovaní algoritmů pro vizualizaci Juliových množin.

Literatura: Douady, A. In The Beauty of Fractals: Images of Complex Dynamical Systems. Berlin: Springer-Verlag, 1986.

MANDELBROT, Benoit B. Fractals and chaos: the Mandelbrot set and beyond: volume C. New York: Springer, 2004. ISBN 978-0-387-20158-0.

Peitgen, Heinz-Otto, Hartmut Jürgens, and Dietmar Saupe. Chaos and fractals: new frontiers of science. Springer Science & Business Media, 2006. ISBN 978-0-387-20229-7.

Vedoucí práce:	RNDr. Veronika Eclerová, Ph.D.
Datum zadání práce:	13. 5. 2021
V Brně dne:	27. 4. 2022

Zadání bylo schváleno prostřednictvím IS MU.

Jan Macharáček, 18. 10. 2021

RNDr. Veronika Eclerová, Ph.D., 4. 11. 2021

RNDr. Jan Vondra, Ph.D., 4. 11. 2021

Poděkování

Na tomto místě bych chtěl poděkovat RNDr. Veronice Eclerové, Ph.D. za odborné vedení, cenné rady a připomínky při konzultacích a ochotu kdykoliv pomoci během zpracovávání bakalářské práce. Dále bych také rád poděkoval mé rodině a blízkým, za motivaci a podporu během celého studia.

Prohlášení

Prohlašuji, že jsem svoji bakalářskou práci vypracoval samostatně pod vedením vedoucího práce s využitím informačních zdrojů, které jsou v práci citovány.

Brno 10. května 2022

.....

Jan Macharáček

Obsah

Přehled použitého značení	ix
Úvod	1
Kapitola 1. Potřebné základní vědomosti	2
1.1 Metrické prostory	2
1.2 Soběpodobnost	3
1.3 Fraktální dimenze	4
Kapitola 2. Komplexní analýza	8
2.1 Zavedení komplexních čísel	8
2.1.1 Goniometrický tvar	9
2.1.2 Operace s komplexními čísly	10
2.2 Iterace	17
2.2.1 Lineární iterace	18
2.2.2 Nelineární iterace	22
Kapitola 3. Juliový množiny	23
3.1 Historie	23
3.2 Dualita	24
3.2.1 Orbita	25
3.2.2 Pevné body	27
3.2.3 Oblast přitažení	28
3.3 Limitní poloměr divergence	29
3.3.1 Aproximace prisoner set	31
3.4 Fatouova množina	33
3.5 Juliový množina vs. Filled Juliový množina	33
3.5.1 Nejznámější filled Juliový množiny	34
3.6 Vlastnosti Juliových množin	37
Kapitola 4. Souvislosti mezi Mandelbrotovou a Juliovou množinou	43
4.1 Historie	43
4.2 Mandelbrotova množina	44
4.2.1 Aproximace Mandelbrotovy množiny	45
4.2.2 Symetrie	47

4.2.3 Soběpodobnost	48
4.3 Mapa Juliových množin	48
4.3.1 Geometrie Mandelbrotovy množiny	49
4.3.2 Rotační čísla	53
4.4 Asymptotická soběpodobnost M a J_c okolo Misiurewiczova bodu	57
Kapitola 5. Webová aplikace	64
5.1 HTML, CSS, JavaScript	64
5.1.1 Použitá rozšíření	65
5.2 Popis aplikace	65
5.3 Vizualizace	66
5.3.1 Metody	69
Závěr	78
Příloha	79
Seznam použité literatury	80

Přehled použitého značení

Pro snazší orientaci v textu zde čtenáři předkládáme přehled základního značení, které se v celé práci vyskytuje.

\mathbb{C}	množina všech komplexních čísel
\mathbb{R}	množina všech reálných čísel
\mathbb{Z}	množina všech celých čísel
\mathbb{E}^n	n -dimenzionální Euklidovský prostor
$X = [x_1, x_2, \dots, x_n]$	uspořádaná n -tice souřadnic bodu X
$d(x, y)$	vzdálenost bodu x a y
$O_\varepsilon(x)$	okolí bodu x s poloměrem ε
$D_r(x)$	kruh se středem v x a poloměrem r
$H^s(G)$	s -dimenzionální Huasdorffova míra objektu G
$\dim_T(G), \dim_H(G), \dim_C(G)$	topologická, Hausdorffova, kapacitní dimenze objektu G
$a \approx b$	aproximace hodnoty a jako b (a je přibližně b)
\bar{z}	komplexně sdružené číslo k z
$\arg z$	orientovaný úhel (argument) z
$z \mapsto z^2 + c, f(z) = z^2 + c$	zobrazení (funkce)
$z_{n+1} = z_n^2 + c$	nelineární iterační předpis
$f^k(z_0)$	k -tá iterace funkce f pro počáteční hodnotu z_0
$f \propto g$	f je úměrné g
f^{-1}	inverzní funkce k funkci f
$f'(x)$	derivace funkce f v bodě x
P_c	(prisoner set) oblast počátečních hodnot c nedivergujících orbit
E_c	(escape set) oblast počátečních hodnot c divergujících orbit
J_c	Juliova množina
F_c	Fatouova množina
K_c	filled Juliova množina
M	Mandelbrotova množina
$r(c)$	limitní poloměr divergence $\max(c , 2)$
A_k	k -tá approximace výsledného objektu

\bar{A}

uzávěr množiny A

L_A

limitní objekt počátečního objektu A

Úvod

Fraktální geometrie je poměrně mladá vědní disciplína. K jejímu velkému rozmachu došlo teprve nedávno kolem 70. let minulého století, za což se velkou měrou zasloužil Benoit B. Mandelbrot, který je považován za jejího zakladatele a který jako první definoval pojem *fraktál*. Důvod, proč o tuto disciplínu nebyl dříve takový zájem z velké části souvisí s rozvojem počítačové grafiky. Ještě před dostatečně výkonými počítači, schopnými vykreslit fraktální obrazec v celé své komplexnosti, bylo velmi obtížné k fraktální geometrii přitáhnout jakékoli větší spektrum matematického pléna.

Hlavním cílem této práce je definování a popis základních vlastností Juliových množin a uvedení souvislostí s Mandelbrotovou množinou. Praktická část spočívá v navržení a naprogramování algoritmů pro vizualizaci Juliových množin, na jejichž základně vznikne webová aplikace.

Text je členěn do pěti kapitol. První kapitola představuje souhrn základních vět, definic a pojmu, často obohacených o názorné příklady či poznámky, z teorie metrických prostorů a fraktální geometrie.

Druhá kapitola shrnuje základy komplexní analýzy, využívané nejen při výpočtech v celé práci. Je zadefinována imaginární jednotka a následně jsou rozebrány elementární operace na komplexních číslech. Na závěr zde vystupuje pojem iterace, který je základním stavebním kamenem celé práce.

Třetí kapitola je stejně, jelikož se zabývá přímo hlavním tématem práce. Dochází zde k formulování významných pojmu pro definici Juliovu množiny a následně její různé definice. V druhé části kapitoly se poté nachází sekce zobrazující nejznámější zástupce Juliových množin navazující sekcí s jejími základními vlastnostmi, využitými ve velkém míře v praktické části práce.

Čtvrtá kapitola pojednává o souvislostech Juliových množin s Mandelbrotovou množinou. Nejprve je zadefinována Mandelbrotova množina a její důležité vlastnosti. Následně je popsáno několik souvislostí, které jsou doplněny o obrázky napomáhající lepšímu chápání problematiky.

Poslední kapitola se zabývá již webovou aplikací. Rozebírá jednotlivé vizualizační metody: Escape time algorithm, Distance estimator method a Inverse iteration method. U každé metody jsou uvedeny úryvky ze zdrojového kódu a obrázky vygenerované pomocí něj.

Celá práce je vysázena v systému L^AT_EX. Webová aplikace je naprogramována jazykem JavaScript ve spojení s HTML a CSS ve vývojářském prostředí Microsoft Visual Studio. Pomocí stejného jazyka byla vygenerována většina obrázků, zbytek potom pomocí matematické aplikace GeoGebra.

Kapitola 1

Potřebné základní vědomosti

Cílem této kapitoly je poskytnout čtenáři vědomostní základ, pro jednodušší chápání rozebrané problematiky. Úvodní kapitola byla sepsána převážně na základě informací z [1] a [2]. Dále také čerpá z [3] a [4]. Pro úplnost byly využity i další zdroje, které jsou navíc i s výše zmíněnými, poznačeny vždy přímo u konkrétní definice, či věty.

1.1 Metrické prostory

Definice 1.1. [1] *Metrický prostor* je definován jako uspořádaná dvojice (M, d) , kde M je libovolná neprázdná množina a d je zobrazení $d : M \times M \rightarrow [0, \infty)$. Zobrazení d nazveme *metrikou*¹, splňuje-li pro každé $x, y, z \in M$ následující vlastnosti:

- $d(x, y) \geq 0$ a $d(x, x) = 0$,
- $d(x, y) = 0 \Leftrightarrow x = y$,
- $d(x, y) = d(y, x)$,
- $d(x, z) \leq d(x, y) + d(y, z)$.

Číslo $d(x, y)$ nazýváme vzdálenost bodů x, y v prostoru (M, d) .

Příklad 1. Uveďme dva metrické prostory a jejich metriky, které se nám budou později hodit.

Euklidovský prostor \mathbb{E}^n . Nechť $X = [x_1, x_2, \dots, x_n]$ a $Y = [y_1, y_2, \dots, y_n]$ jsou dva body se souřadnicemi jako uspořádanými n -ticemi v n -dimenzionálním Euklidovském prostoru, což odpovídá \mathbb{R}^n . Vzdálenost $d(X, Y)$ je zde myšlena délka úsečky s krajními body X a Y , která je určena vztahem

$$d(X, Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}.$$

Vidíme, že takto definovaná metrika, často označována jako ρ_2 , splňuje všechny náležitosti z definice 1.1 a můžeme tak Euklidovský prostor prohlásit za metrický prostor (\mathbb{R}^n, ρ_2) .

¹viz [1]

Komplexní čísla \mathbb{C} . Jelikož \mathbb{C} jsou totožná s uspořádanými dvojicemi \mathbb{R}^2 , potom pro vzdálenost komplexních čísel $z = a + bi$ a $w = c + di$ definujme metriku

$$d(z, w) = \sqrt{(a - c)^2 + (b - d)^2}.$$

Opět $d(x, y)$ splňuje vše potřebné, aby byla komplexní čísla metrický prostor (\mathbb{R}^2, d) .

Definice 1.2. [1] Metrický prostor (M, d) se nazývá *souvislý*, jestliže M nelze vyjádřit jako sjednocení dvou neprázdných disjunktních podmnožin M , které jsou uzavřené v M . Množina $X \subseteq M$ se nazývá *souvislá*, je-li metrický prostor (M, d) souvislý.

Definice 1.3. [1] Nechť (M, d) je metrický prostor. Pak množina $X \subseteq M$ je *souvislá*, pokud ji nelze vyjádřit jako sjednocení dvou disjunktních *uzavřených* množin v X .

Definice 1.4. [1] Nechť (M, d) je metrický prostor. Množina $X \subseteq M$ se nazývá *hustá* v M , jestliže $A \bar{A} = M$, kde \bar{A} značí uzávěr množiny² A .

Věta 1.1.1. *Množina $A \subseteq M$ je hustá v metrickém prostoru (M, d) právě, když ke každému $\epsilon > 0$ a každému $x \in A$ existuje okolí $\mathcal{O}_\epsilon(x)$ pro které platí $\mathcal{O}_\epsilon(x) \cap P \neq \emptyset$.*

Důkaz. Viz [1, strana 22]. \square

Poznámka. Ekvivalentně potom platí i vlastnost býti hustou, pro podmnožinu v množině.

1.2 Soběpodobnost

Definice 1.5. [2] Nechť (M, d) je metrický prostor a $k \in [0, 1]$. Zobrazení $f : M \rightarrow M$ takové, že

$$d(f(x), f(y)) = kd(x, y) \quad \forall x, y \in M$$

se nazývá *podobnost* a k , pro které platí rovnost, *faktor podobnosti*.

Definice 1.6. [2] Nechť (M, d) je metrický prostor a $k \in [0, 1]$. Zobrazení $f : M \rightarrow M$ takové, že

$$d(f(x), f(y)) \leq kd(x, y) \quad \forall x, y \in M$$

se nazývá *kontrakce* a nejmenší k , pro které platí nerovnost, *faktor kontrakce*.

Definice 1.7. [5] Striktně soběpodobná množina G n -dimenzionálního Euklidovského prostoru E_n je taková množina, pro níž existuje konečně mnoho kontraktí f_1, \dots, f_n takových, že

$$G = \bigcup_{i=1}^n f_i(G).$$

Poznámka. Jinými slovy G má konečně mnoho částí, které jsou jí podobné a sjednocením těchto částí dostaneme celkový obraz množiny G . Tomuto popisu odpovídá například Sierpińského trojúhelník na obrázku 1.1 níže.

²Definice uzávěru množiny zde 3.13.

Pro další definici si pod pojmem affinní transformace v jednoduchosti představujme posunutí, škálování³, rotaci, souměrnost a jejich kombinace. Více informací a exaktní definici lze najít v [4, strana 139].

Připustíme-li, že zobrazení f_1, \dots, f_n nebudou pouze kontrakce ale i jiné affinní transformace a jejich kombinace, můžeme definovat *statistickou soběpodobnost*, kterou lze v přírodě na rozdíl od striktní soběpodobnosti velmi často najít.

Definice 1.8. [6] Statisticky soběpodobná množina G n -dimenzionálního Euklidovského prostoru E_n je taková množina, pro níž existuje konečně mnoho affinních transformací f_1, \dots, f_n takových, že platí

$$G = \bigcup_{i=1}^n f_i(G).$$

Poznámka. Obecně u soběpodobnosti platí, že při izotropickém⁴ přibližování části soběpodobného objektu bude přiblížená část a celý fraktál vypadat stejně. Existují však fraktální útvary, které tuto vlastnost nesplňují a říkáme jim proto *soběpříbuzné* útvary. Mezi tento typ patří i Juliovy množiny a Mandelbrotova množina. K soběpříbuznosti se dostaneme pouze pomocí anizotropického⁵ zvětšování vyjmuté části. Hlavní rozdíl je tedy značná deformace, která se u soběpodobných útvarů nevyskytovala.

Definice 1.9. [7] Nechť $G \subset \mathbb{C}$ je *soběpodobná* množina v bodě x , jestliže platí

$$\rho(G - x) \cap D_r(0) = (G - x) \cap D_r(0),$$

kde $D_r(0)$ je kruh s malým kladným poloměrem r se středem v 0 a $G - x = \{g - x \mid g \in G\}$. Komplexní číslo ρ představuje faktor zvětšení a splňuje $|\rho| > 1$.

1.3 Fraktální dimenze

Definice 1.10. [3] *Topologická dimenze* objektu G je označována jako $\dim_T(G)$ a určuje počet parametrů potřebných k určení polohy libovolného bodu na G .

Poznámka. Polohu bodu na lineárním objektu G (úsečka, přímka, křivka...) určíme třeba jako vzdálenost od počátku, tedy $\dim_T(G) = 1$. Určení polohy bodu například u Platonovských těles dosáhneme nalezením souřadnic x, y, z v trojrozměrném souřadnicovém systému, tedy $\dim_T(G) = 3$. Obecně tento způsob funguje pro n -rozměrný objekt.

Definice 1.11. [2] Nechť $\{U_i\}$ je spočetná nebo konečná třída množin pokrývající množinu G , tedy

$$G \subseteq \bigcup_{i=1}^{\infty} U_i$$

a $\forall i$ platí $0 \leq |U_i| \leq \varepsilon$, kde $|U_i| = \sup\{d(x, y) \mid x, y \in U_i\}$ a ε značí průměr množin. Pak $\{U_i\}$ je ε -pokrytí G .

³Speciálním typem škálování je kontrakce, kterou lze chápout jako zmenšení.

⁴Změna velikosti objektu ve všech směrech stejně.

⁵Změna velikosti objektu v různých směrech odlišně.

Definice 1.12. [2] Nechť $G \subset \mathbb{C}$. Pro každé $\varepsilon > 0$ platí

$$H_\varepsilon^s(G) = \inf \left\{ \sum_{i=1}^{\infty} |U_i|^s \mid G \subseteq \bigcup_{i=1}^{\infty} U_i \right\},$$

kde $s \geq 0$ a platí, že $|U_i| < \varepsilon$. $H_\varepsilon^s(G)$ je potom infimum všech spočetných nebo konečných pokrytí množiny G .

Definice 1.13. [2] $H^s(G)$ se nazývá *Hausdorffova míra* s dimenzí s , pro kterou platí

$$H^s(G) = \lim_{\varepsilon \rightarrow 0^+} H_\varepsilon^s(G)^6.$$

Definice 1.14. [2] Nechť $G \subset \mathbb{C}$. Potom *Hausdorffova dimenze* $\dim_H(G)$ množiny G je definována jako

$$\dim_H(G) = \inf\{s \geq 0 \mid H^s(G) = 0\},$$

ekvivalentně platí

$$\dim_H(G) = \sup\{s \geq 0 \mid H^s(G) = \infty\},$$

vezmeme-li $\sup \emptyset = 0$ a $\inf \emptyset = \infty$.

Definice 1.15. [8] Nechť $G \subset \mathbb{C}$ je ohraničená množina a nechť $N(\varepsilon)$ označuje nejmenší možný počet čtverců o straně ε potřebných k pokrytí⁷ G . Pokud se pro malé ε zvětšuje $N(\varepsilon)$ podle vztahu

$$N(\varepsilon) \propto \varepsilon^{-\dim_C(G)},$$

pak $\dim_C(G)$ nazýváme *kapacitní dimenze* (box-counting), pro kterou platí

$$\dim_C(G) = \lim_{\varepsilon \rightarrow 0^+} \frac{\log N(\varepsilon)}{\log \frac{1}{\varepsilon}}.$$

Poznámka. Kapacitní dimenze je často označována také jako *Hausdorffova dimenze*. Existují mezi nimi menší technické odlišnosti, které nám však pro náš účel definice fraktálního objektu 1.16 nebudou vadit (viz [9, strana 41]).

Věta 1.3.1. *Hausdorffova dimenze hranice Mandelbrotovy množiny je 2.*

Důkaz. Viz [10, strana 17]. \square

Definice 1.16. [3] Množina $G \subset \mathbb{C}$ se nazývá *nekonečně členitý útvar*, nebo také *fraktál*, jestliže

$$\dim_T(G) < \dim_H(G).$$

⁶Budeme-li zmenšovat ε , pak se třída všech možných pokrytí bude také zmenšovat.

⁷Označme čtverec jako množinu S_i . Pak pokrytí G můžeme definovat jako $G \subset \bigcup_{i=1}^N S_i$.

Příklad 2. Nyní uveďme příklad výpočtu \dim_T a \dim_H ⁸ geometricky hladkého útvaru (čtverce), pro jednoduchost označme jako G_1 a fraktálního útvaru (Sierpińského trojúhelníka) ve výpočtu značeném G_2 . Sierpińskiego trojúhelník vzniká opakováním dělením počátečního trojúhelníka na čtyři menší shodné kopie s poloviční délkou strany. Tento proces opakujeme pro všechny nově vzniklé trojúhelníky vyjma prostředního z nich, pro názornost viz obrázek 1.1.

Čtverec:

Topologická dimenze již byla zmíněna v poznámce pod definicí 1.10, tedy $\dim_T(G_1) = 2$. Pro výpočet \dim_H postačí definice 1.15 a fakt, že $N(\varepsilon)$ a ε jsou ve vztahu $N(\varepsilon) \cdot \varepsilon^2 = 1$ ⁹, z čehož získáváme

$$\dim_H(G_1) = \lim_{\varepsilon \rightarrow 0^+} \frac{\log N(\varepsilon)}{\log \frac{1}{\varepsilon}} = \lim_{\varepsilon \rightarrow 0^+} \frac{\log N(\varepsilon)}{\log N(\varepsilon)^{\frac{1}{2}}} = \lim_{\varepsilon \rightarrow 0^+} \frac{\log N(\varepsilon)}{\frac{1}{2} \cdot \log N(\varepsilon)} = \lim_{\varepsilon \rightarrow 0^+} \frac{1}{\frac{1}{2}} = 2.$$

Vidíme tedy, že jistě $\dim_H(G_1)$ není ostře větší jako $\dim_T(G_1)$ a můžeme tak potvrdit, že čtverec není fraktální útvar, nýbrž opravdu geometricky hladký útvar.

Sierpińskiego trojúhelník:

Tento fraktální útvar je složen pouze z úseček různých délek, bez jakékoliv souvislé plochy, tedy $\dim_T(G_2) = 1$. K nalezení \dim_H stačí zjistit změnu $N(\varepsilon)$ a ε po každé provedené iteraci.

1. iterace: $N = 3$, $\varepsilon = \frac{1}{2}$

2. iterace: $N = 9 = 3^2$, $\varepsilon = \frac{1}{4} = \left(\frac{1}{2}\right)^2$

⋮

n . iterace: $N = 3^n$, $\varepsilon = \left(\frac{1}{2}\right)^n$

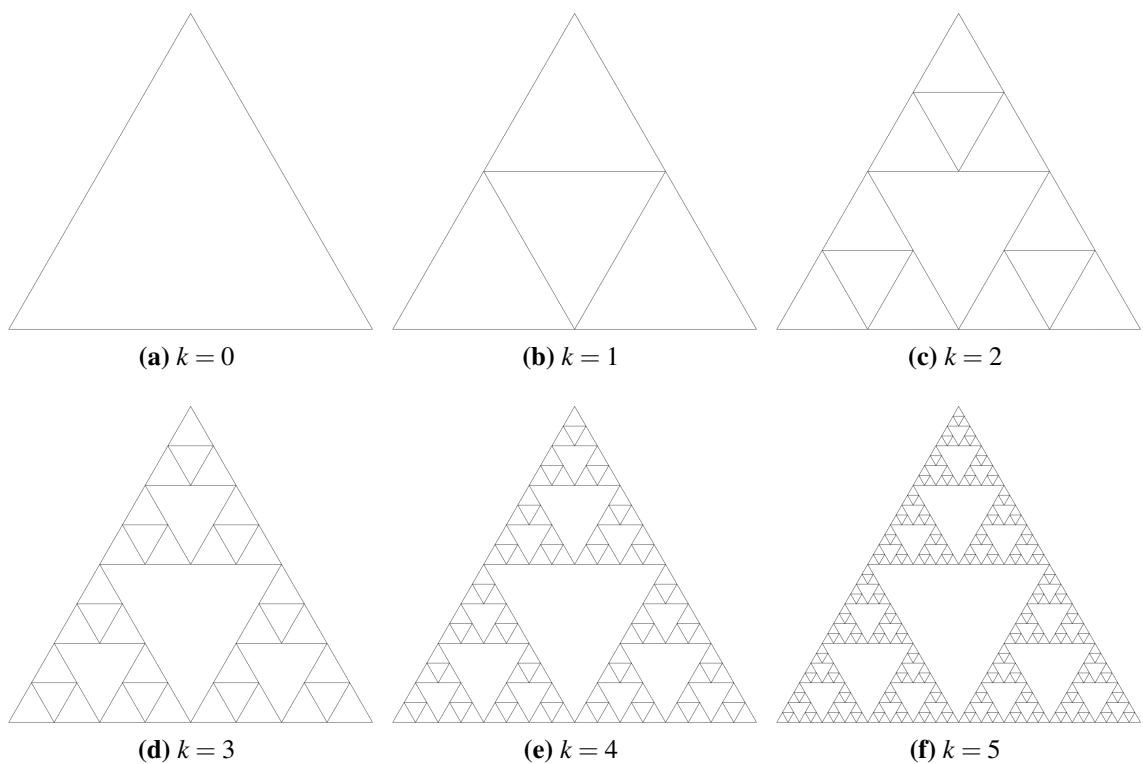
Nyní již pouhým dosazením do vztahu v definici 1.15 máme výsledek

$$\dim_H(G_2) = \lim_{\varepsilon \rightarrow 0^+} \frac{\log N(\varepsilon)}{\log \frac{1}{\varepsilon}} = \lim_{\varepsilon \rightarrow 0^+} \frac{\log 3^n}{\log 2^n} = \lim_{\varepsilon \rightarrow 0^+} \frac{n \cdot \log 3}{n \cdot \log 2} = \lim_{\varepsilon \rightarrow 0^+} \frac{\log 3}{\log 2} \approx 1.58496.$$

Neustálým iterováním vyplňujeme plochu úsečkami tak hustě, že se dimenze blíží plošnému útvaru. Podle definice 1.16 splňuje Sierpińskiego trojúhelník nerovnost $1 < 1.58496$ a můžeme ho tak nazvat fraktálem.

⁸Pro striktně soběpodobný útvar G platí $\dim_H(G) = \dim_C(G)$ [4]. K výpočtu proto využijeme vztah pro $\dim_C(G)$.

⁹ N menších čtverců o ploše ε^2 pokrývá počáteční čtverec o ploše 1.



Obrázek 1.1: Prvních pět iterací Sierpiňského trojúhelníka.

Kapitola 2

Komplexní analýza

Protože se budeme po většinu času pohybovat v oboru komplexních čísel, je vhodné znát základy komplexní analýzy, k čemuž dopomáhá právě tato kapitola. Většina obsahu první části této kapitoly pochází z [11] a [12]. Dalším doplňujícím zdrojem byla publikace [7]. Druhá část pojednávající o iteracích pak vychází hlavně z [13].

2.1 Zavedení komplexních čísel

Historická motivace k zavedení komplexních čísel byla z velké části založena na hledání kořenů rovnic, které nemají v oboru \mathbb{R} řešení. Zkusme například najít kořeny polynomu druhého stupně

$$x^2 + 1 = 0. \quad (2.1)$$

Při hledání kořenů pomocí diskriminantu přijdeme na problém

$$D = 0 - 4 = -4.$$

Uvědomíme-li si, že pak

$$x_{1,2} = \frac{\pm\sqrt{-4}}{2}.$$

Což v oboru reálných čísel nemá řešení, neboť neexistuje žádné reálné číslo, jehož druhá mocnina je -4 . Kvůli takovým případům vznikl prvek nazývající se *imaginární jednotka* i , pro kterou platí

$$i^2 = -1. \quad (2.2)$$

Vrátíme-li se k (2.1), pak můžeme psát

$$x_{1,2} = \frac{0 \pm \sqrt{-4}}{2} = \frac{0 \pm \sqrt{(-1)4}}{2} = \frac{0 \pm 2i}{2} = 0 \pm i.$$

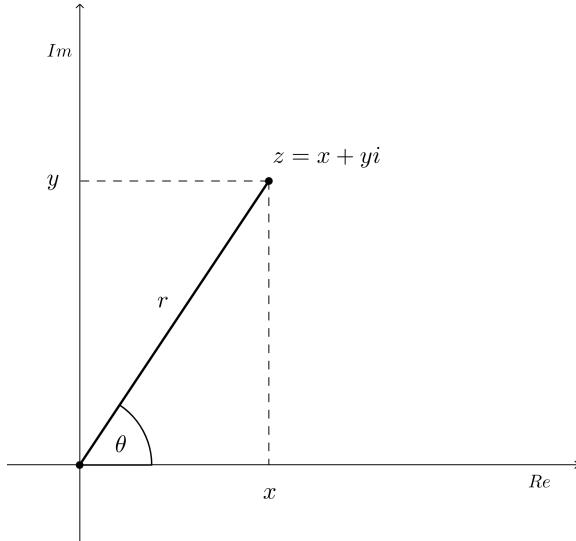
Obecně tedy můžeme definovat komplexní číslo z jako

$$z = x + yi \quad x, y \in \mathbb{R}. \quad (2.3)$$

Kde x nazveme *reálná část* a y *imaginární část* komplexního čísla. Rovina se souřadnicovým systémem, na které jsme definovali komplexní čísla se nazývá *Gaussova rovina*, kde osu x nazýváme *reálná* a představuje reálnou část komplexního čísla. Osa y se nazývá *imaginární* a představuje imaginární část komplexního čísla. Vzdálenost z od počátku souřadnicového systému označme r , pro které platí

$$r = |z| = \sqrt{x^2 + y^2}. \quad (2.4)$$

Grafické znázornění komplexního čísla z můžeme vidět na obrázku 2.1.



Obrázek 2.1: Grafické znázornění komplexního čísla z .

Pomocí osové souměrnosti podle reálné osy získáme *komplexně sdružené číslo* \bar{z} , viz 2.2, pro které platí

$$\bar{z} = x - yi \quad x, y \in \mathbb{R}.$$

2.1.1 Goniometrický tvar

Úhel θ s rameny r a pozitivní částí reálné osy na obrázku 2.1 označuje orientovaný úhel proti chodu hodinových ručiček a nazývá se argument

$$\theta = \arg z. \quad (2.5)$$

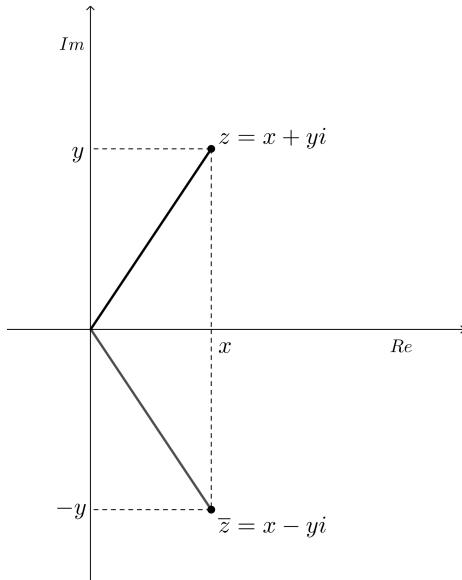
Pro orientovaný úhel platí, že pokud komplexní číslo $z \neq 0$ má argument θ , pak má také argument $\theta + k \cdot 2\pi$ pro libovolné $k \in \mathbb{Z}$.

Doposud jsme se bavili o komplexním čísle pouze v algebraickém tvaru, tedy $z = x + yi$. Komplexní číslo však lze určit i pomocí jeho vzdálenosti r od počátku a pomocí orientovaného úhlu θ . Tyto pojmy jsou již definovány výše v (2.4) a (2.5). Jak je z obrázku 2.1 zřejmé: $\cos \theta = \frac{x}{r}$, $\sin \theta = \frac{y}{r}$ a potom tedy

$$\begin{aligned} x &= r \cos \theta, \\ y &= r \sin \theta. \end{aligned}$$

Dosazením do algebraického tvaru komplexního čísla pak dostáváme *goniometrický tvar* komplexního čísla

$$z = x + yi = r \cos \theta + r \sin \theta i = r(\cos \theta + i \sin \theta).$$



Obrázek 2.2: Grafické znázornění komplexně sdruženého čísla \bar{z} k číslu z .

2.1.2 Operace s komplexními čísly

Aritmetické vlastnosti komplexních čísel jsou velmi podobné vlastnostem čísel reálných. Uvědomme si, že komplexní číslo je dvojčlen, tedy na něj při operacích budeme tímto způsobem nahlížet. V této podkapitole budou popsány základní operace s komplexními čísly a jejich náležitosti.

Nejdříve si však definujme základní vlastnost dvou komplexních čísel z a w . Řekneme, že $z = x + yi$ a $w = u + vi$ jsou si rovny právě tehdy, když se rovnají jejich reálné a imaginární části. Tedy

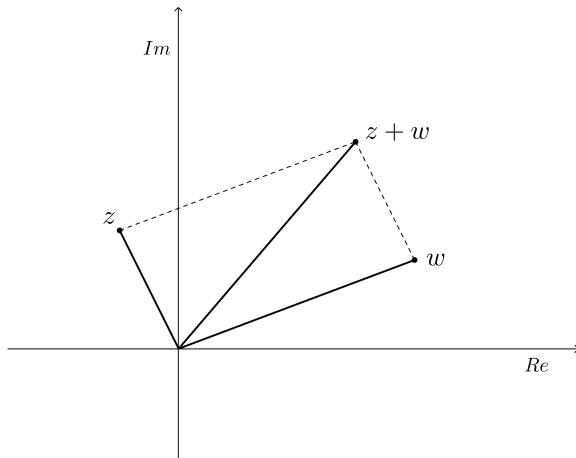
$$x + yi = u + vi \iff (x = u \wedge y = v). \quad (2.6)$$

Sčítání

Součet dvou komplexních čísel je opět komplexní číslo. Tento fakt plyne přímo z definice operace sčítání:

$$z + w = (x + yi) + (u + vi) = (x + u) + (y + v)i.$$

Je zřejmé, že součet funguje identicky, jako součet dvojčlenu, tedy zvlášť *Re* části dvojčlenu a pak *Im* části. Grafické znázornění komplexního čísla je stejné jako vyjádření vektoru, tedy součet komplexních čísel z, w v Gaussově rovině vytvoříme jako součet vektorů, jak vidíme na obrázku 2.3.



Obrázek 2.3: Součet z a w metodou doplnění na rovnoběžník.

Násobení

Při násobení postupujeme stejně jako při násobení dvojčlenem, tedy

$$z \cdot w = (x + yi)(u + vi) = xu + xvi + uyi - yv = (xu - yv) + (xv + uy)i. \quad (2.7)$$

Obecně můžeme násobení komplexního čísla rozdělit do dvou případů: součin komplexního čísla a reálné konstanty a součin dvou komplexních čísel. Pro první případ mějme reálnou konstantu k . Pak součin probíhá stejně jako roznásobování dvojčlenu jednočlenem, tedy pro komplexní číslo z platí

$$kz = k(x + yi) = kx + kyi.$$

V grafickém vyjádření tedy pouze zvětšíme x a y k -krát. Tento výsledek demonstруje obrázek 2.4.

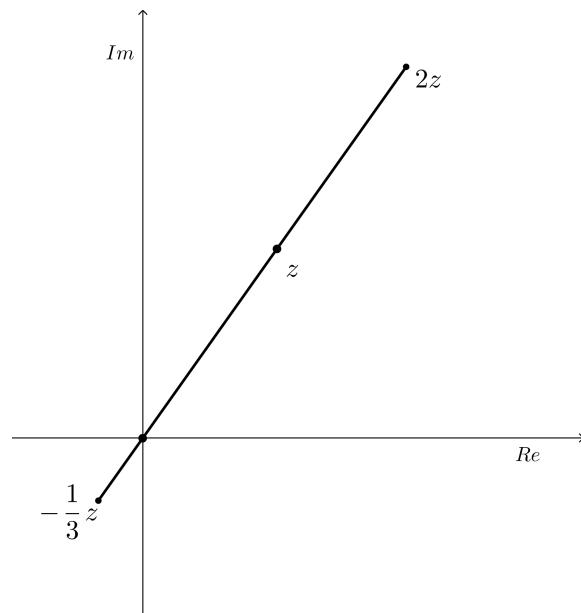
U druhého případu pro aritmetický tvar máme identickou rovnost jako v (2.7). Grafické znázornění pomocí definice čísla v aritmetickém tvaru je složité, ukážeme si však elegantní řešení pomocí goniometrický tvaru. Odvoďme si nejdříve součin komplexních čísel $z = r(\cos \theta + i \sin \theta)$ a $w = s(\cos \psi + i \sin \psi)$ v goniometrickém tvaru, tedy

$$\begin{aligned} z \cdot w &= r(\cos \theta + i \sin \theta)s(\cos \psi + i \sin \psi) \\ &= rs(\cos \theta \cos \psi + i \cos \theta \sin \psi + i \sin \theta \cos \psi - \sin \theta \sin \psi). \end{aligned}$$

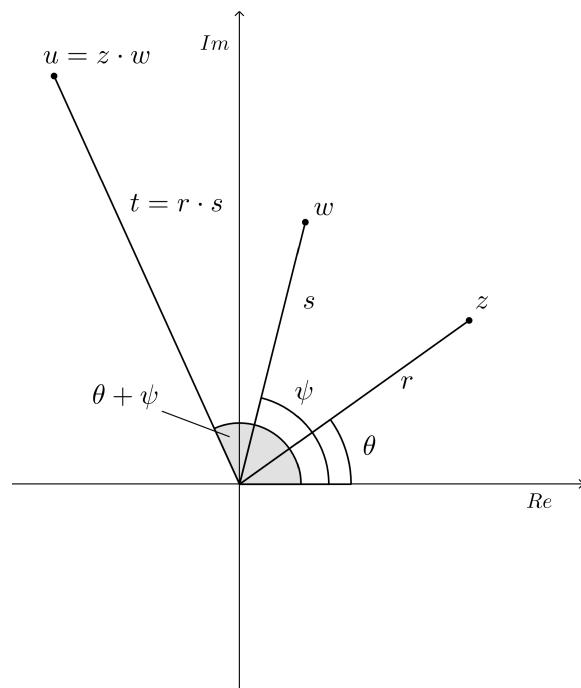
Pomocí součtových vzorců pro *sinus* a *cosinus* dostáváme

$$z \cdot w = rs(\cos(\theta + \psi) + i \sin(\theta + \psi)). \quad (2.8)$$

Vidíme tedy, že graficky dostaneme výslednou vzdálenost jako součin vzdáleností obou komplexních čísel v součinu a úhel jako součet obou úhlů, viz obrázek 2.5.



Obrázek 2.4: Součin komplexního čísla a reálné konstanty, pro $k = \{-\frac{1}{3}, 1, 2\}$.



Obrázek 2.5: Nechť u je součin komplexních čísel a t je výsledná velikost. Úhel součinu je znázorněn šedým odstínem.

Odčítání

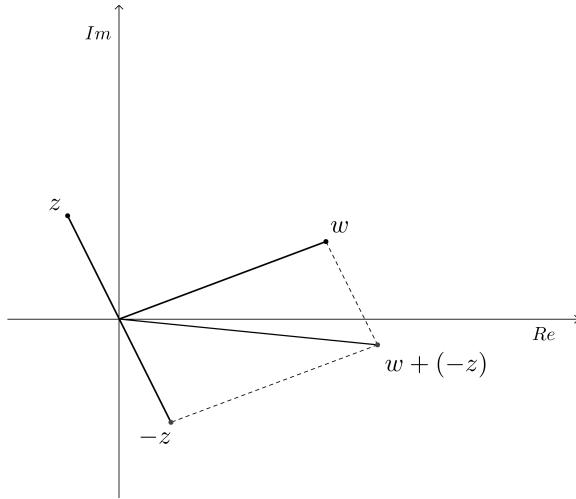
Odčítání komplexních čísel funguje identicky jako sčítání, uvědomíme-li si, že

$$w - z \iff w + (-z).$$

kde $-z$ nazveme číslo *opačné*, k číslu $z = x + yi$, pro které platí $-z = -x - yi$. Rozdíl pak definujeme jako součet w a čísla opačného k číslu z . Tedy

$$w + (-z) = u + vi + (-x - yi) = (u - x) + (v - y)i.$$

Grafické znázornění vychází přímo z definice, což lze vidět na obrázku 2.6.



Obrázek 2.6: Součet w a čísla opačného k z .

Dělení

Podíl komplexních čísel $z = x + yi$ a $w = u + vi \neq 0$ je opět komplexní číslo, tedy se snažíme vyjádřit $\frac{x+yi}{u+vi}$ jak $a + bi$. K tomu nám pomůže následující vlastnost součinu komplexního čísla z a jemu čísla sdruženého, tedy \bar{z} . Platí totiž

$$z\bar{z} = (x + yi)(x - yi) = x^2 - xyi + xyi + y^2 = x^2 + y^2. \quad (2.9)$$

Při hledání podílu z a w postupujeme tak, že se rozšířením zlomku, číslem komplexně sdruženým k w , zbavíme komplexního čísla ve jmenovateli, pomocí (2.9) a tím získáme výsledný tvar

$$\frac{z}{w} = \frac{x + yi}{u + vi} = \frac{x + yi}{u + vi} \cdot \frac{u - vi}{u - vi} = \frac{(x + yi)(u - vi)}{u^2 + v^2} = \frac{xu - xvi + yvi + yv}{u^2 + v^2}.$$

Po rozseparování na reálnou a komplexní část a vytknutí dostáváme kýženou formu podílu dvou komplexních čísel

$$\frac{z}{w} = \frac{xu + yv}{u^2 + v^2} + \frac{(uy - xv)i}{u^2 + v^2}.$$

Stejně jako u násobení můžeme rozlišit dva možné případy: dělení reálnou konstantou k a dělení komplexním číslem. Prvně zmiňovaný je identický k násobení reálnou konstantou, všimněme si, že

$$\frac{z}{k} = \frac{x+yi}{k} = \frac{1}{k}(x+yi).$$

Tedy podíl z a k , je vlastně součin z a převrácené hodnoty k . Z logiky věci můžeme při grafickém znázornění očekávat stejný výsledek jako na obrázku 2.4 s tím rozdílem, že pro větší k se velikost z bude zmenšovat.

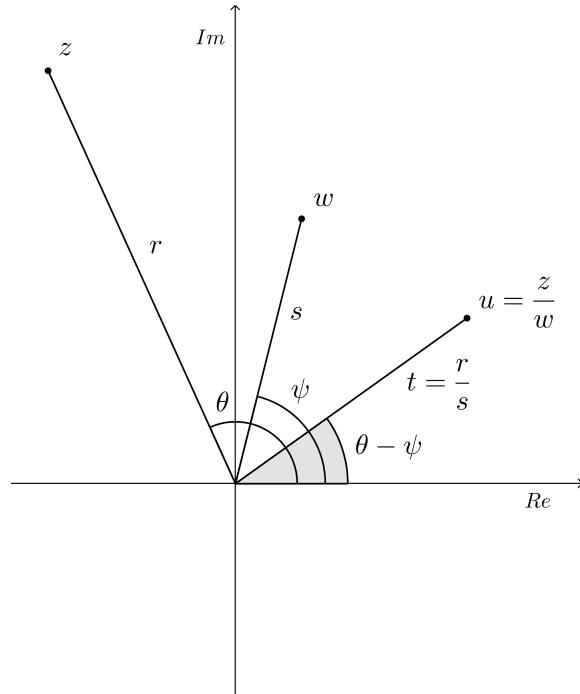
Podíl dvou komplexních čísel v aritmetickém tvaru je opět na grafické vyjádření velmi složitý, takže si opět pomůžeme goniometrickým tvarem. Nechť $z = r(\cos \theta + i \sin \theta)$ a $w = s(\cos \psi + i \sin \psi)$ jsou komplexní čísla v goniometrickém tvaru. Pak podíl těchto čísel najdeme stejně jako pro aritmetický tvar, tedy

$$\begin{aligned} \frac{z}{w} &= \frac{r(\cos \theta + i \sin \theta)}{s(\cos \psi + i \sin \psi)} \cdot \frac{(\cos \psi - i \sin \psi)}{(\cos \psi - i \sin \psi)} \\ &= \frac{r(\cos \theta \cos \psi - i \sin \psi \cos \theta + i \sin \theta \cos \psi + \sin \psi \sin \theta)}{s(\cos^2 \psi - i^2 \sin^2 \psi)}. \end{aligned}$$

Součtové vzorce ve spojení s goniometrickou jedničkou pak dávají výsledek

$$\frac{z}{w} = \frac{r}{s}(\cos(\theta - \psi) + i \sin(\theta - \psi)).$$

Vidíme, že nebude problém graficky vyjádřit podíl komplexních čísel v goniometrickém tvaru. Výsledná velikost je podílem velikostí délence a dělitele a výsledný orientovaný úhel je rozdíl počátečních úhlů.



Obrázek 2.7: Nechť u je podíl z a w a t je výsledná velikost. Úhel podílu je znázorněn šedým odstínem.

Umocňování

Druhá mocnina je zřejmá, uvědomíme-li si, že $z^2 = z \cdot z$, tedy součin dvou stejných komplexních čísel. Pro goniometrický tvar, druhá mocnina komplexního čísla podle (2.8) vypadá následovně

$$z \cdot z = rr(\cos(\theta + \theta) + i \sin(\theta + \theta)) = r^2(\cos 2\theta + i \sin 2\theta).$$

Věta 2.1.1 (Moivreova). *Pro každé přirozené číslo n a libovolné reálné číslo θ platí*

$$(\cos \theta + i \sin \theta)^n = \cos n\theta + i \sin n\theta.$$

Důkaz. Viz [14]. □

Zobecněním předchozího vztahu a pomocí Moivreovy věty, můžeme definovat n -tu mocninu komplexního čísla z jako

$$z^n = r^n(\cos n\theta + i \sin n\theta).$$

Stále přirovnáváme komplexní číslo ke dvojčlenu. Umocňování komplexního čísla je však o poznání jednodušší. Díky vlastnosti (2.2) je totiž výsledek vždy opět dvojčlen, Konkrétně totiž první člen je součet reálných částí součinů a druhý člen součet imaginárních částí, tedy

$$z^2 = (x + yi)^2 = x^2 + 2xyi - y^2 = (x^2 - y^2) + 2xyi.$$

Tohoto vztahu budeme dále hojně využívat v kapitole 5 u vizualizace zobrazovacích metod.

Operace s mocninami v \mathbb{C} jsou definovány identicky jako pro reálná čísla. Tedy pro libovolná komplexní čísla z a w platí

$$z^m \cdot z^n = z^{m+n}, \quad (z \cdot w)^n = z^n \cdot w^n, \quad (z^m)^n = z^{mn}, \quad \frac{z^m}{z^n} = z^{m-n}.$$

Odmocňování

Definovat odmocňování můžeme podobně jako pro reálná čísla. Za n -tou odmocninu z komplexního čísla z označme každé komplexní číslo w pro které platí

$$z = w^n. \quad (2.10)$$

Zkusme tedy obecně hledat druhou odmocninu z čísla $z = x + yi$. Podle definice tedy platí

$$z = w^2 = (u + yi)^2 = u^2 + 2uvi - v^2.$$

Tedy

$$x + yi = u^2 + 2uvi - v^2.$$

Podle rovnosti komplexních čísel (2.6) pak platí

$$\begin{aligned} x &= u^2 - v^2 \\ y &= 2uv. \end{aligned} \quad (2.11)$$

Vyjádříme v a dosadíme do (2.11)

$$v = \frac{y}{2u}.$$

Pak

$$x = u^2 - \frac{y^2}{4u^2}.$$

Celý výraz roznásobíme u^2

$$\begin{aligned} xu^2 &= u^4 - y^2 \\ 0 &= u^4 - xu^2 - y^2. \end{aligned}$$

Substitucí $t = u^2$ ještě lze upravit na

$$0 = t^2 - xt - y^2. \quad (2.12)$$

Dále už bychom jen spočítali kořeny polynomu druhého stupně a dosadili zpět do odpovídajícího výrazu a tím získali odmocninu z komplexního čísla. Poslední úkony už nejsou potřeba dopočítávat, protože důležitý fakt, co jsme tímto výpočtem chtěli ukázat je, že pro

druhou odmocninu získáme v rovnosti (2.12) polynom druhého stupně. Pro odmocniny vyšších řádů pak budeme ekvivalentně dostávat polynom vyššího řádu. Tím se řešení n -té odmocniny z čísla z v algebraickém tvaru stává skoro nemožným.

Efektnější způsob je počítat odmocninu pomocí goniometrického tvaru komplexního čísla. Opět využijeme vztahu (2.10), jen tentokrát máme z i w v goniometrickém tvaru. Tedy platí

$$r(\cos \theta + i \sin \theta) = s^n (\cos n\psi + i \sin n\psi).$$

Tedy pak využitím faktu o rovnosti komplexních čísel společně s vlastností argumentu z 2.1.1 platí

$$r = w^n \quad \wedge \quad n\psi = \theta + k \cdot 2\pi.$$

Z čehož dostáváme

$$w = \sqrt[n]{r} \quad \wedge \quad \psi = \frac{\theta + k \cdot 2\pi}{n}.$$

A konečně tedy komplexní číslo w , které je n -tou odmocninou z lze vyjádřit jako

$$w = \sqrt[n]{r} \left(\cos \frac{\theta + k \cdot 2\pi}{n} + i \sin \frac{\theta + k \cdot 2\pi}{n} \right), \quad k \in \mathbb{Z}.$$

2.2 Iterace

V této podkapitole vysvětlíme základní náležitosti týkající se iteračního procesu na množině komplexních čísel, který je základním stavebním kamenem Juliových množin. Tyto základní znalosti rozšíříme v následujících pasážích textu. V rychlosti připomeňme pojednání *iterace*. Zavedeme si transformační operátor \mapsto a na příkladu si ukažeme, jak funguje. Mějme

$$z \mapsto z + 2$$

To značí, že na vstupu máme libovolnou hodnotu z a transformací získáváme z zvětšené o dva. Nechť počáteční $z = 0$. Potom pokud tento proces budeme opakovat, tedy, že výslednou hodnotu po transformaci vezmeme a opět vložíme jako vstupní hodnotu pro následnou transformaci

$$z \mapsto z + 2 \mapsto (z + 2) + 2 \mapsto ((z + 2) + 2) + 2 \mapsto \dots$$

dostáváme stále se opakující smyčku, generující následující hodnoty

$$0 \mapsto 2 \mapsto 4 \mapsto 6 \mapsto 8 \dots$$

Takové smyčce, kterou opakujeme stále dokola říkáme iterační proces, nebo také iterační cyklus a jednotlivé transformace nazýváme iterace. Tedy zopakujeme-li transformaci n -krát, dostáváme iterační cyklus délky n skládající se z n iterací. Identicky můžeme náš příklad přepsat do tvaru

$$z_{n+1} = z_n + 2.$$

Tím říkáme, že následující hodnotu transformace vypočítáme pomocí předchozí, což po zamýšlení říká naprostě to samé jako zápis pomocí \mapsto .

2.2.1 Lineární iterace

Lineární iterační proces definujme takto

$$z \mapsto kz + c.$$

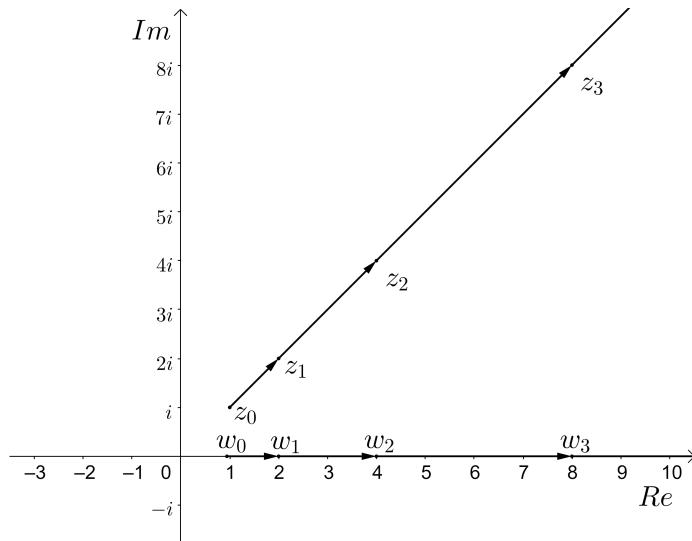
Nebo též

$$z_{n+1} = kz_n + c.$$

V celé podkapitole budeme uvažovat pouze iterační proces s $c = 0$ pro jednodušší porozumění. Nyní si rovnou uveďme příklad. Nechť tedy $c = 0$ a $k, z \in \mathbb{R}$. Mějme $k = 2$ a $z = 1$. Zvolenou hodnotu z nazýváme *počáteční hodnota* iterace a označujeme ji z_0 . Jednotlivé výsledky z iteračních kroků tvoří posloupnost hodnot $z_0, z_1, z_2, \dots, z_n$. Potom

$$\begin{aligned} z_0 &= 1 \\ z_1 &= 2 \cdot z_0 = 2 \cdot 1 = 2 \\ z_2 &= 2 \cdot z_1 = 2 \cdot 2 = 4 \\ z_3 &= 8 \\ &\vdots \end{aligned}$$

Vidíme, že pro takto zvolené hodnoty jde posloupnost do nekonečna. Například ale pro $z = 0$, bude posloupnost obsahovat pouze nuly. Pokud zvolíme k , nebo z záporné, posloupnost půjde do míinus nekonečna. Podobně i pokud z je z oboru komplexních čísel. Jak můžeme vidět z obrázku 2.8, body z posloupnosti vždy kopírují přímku.



Obrázek 2.8: Vidíme dvě posloupnosti, kde $z \in \mathbb{C}$ a $w \in \mathbb{R}$. Konkrétně $z = 1 + i$ a $w = 1$.

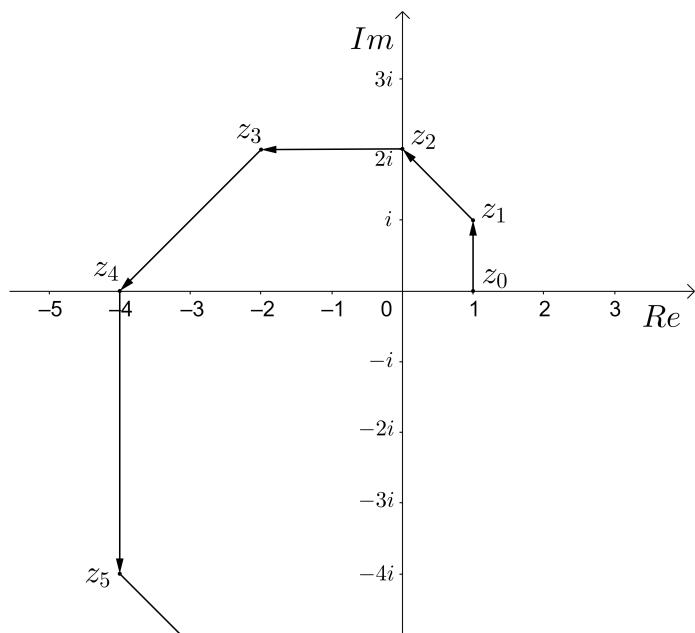
Pojďme se však podívat na případ, pro $k \in \mathbb{C}$, kde už bude situace zajímavější. Nastávají tři různé situace, které interpretuje jeden konkrétní příklad. Pro jednoduchost bude ve všech případech $z = 1$.

Pro první případ zvolme libovolně k , pro které platí $|k| > 1$.

$k = 1 + i$:

$$\begin{aligned} z_0 &= 1 \\ z_1 &= (1+i) \cdot z_0 = (1+i) \cdot 1 = 1+i \\ z_2 &= (1+i) \cdot z_1 = (1+i) \cdot (1+i) = 2i \\ z_3 &= (1+i) \cdot 2i = -2+2i \\ z_4 &= (1+i) \cdot (-2+2i) = -4 \\ &\vdots \end{aligned}$$

Jak můžeme vidět i z grafické interpretace 2.9, posloupnost se každou iterací vzdaluje od počátku a míří do nekonečna.



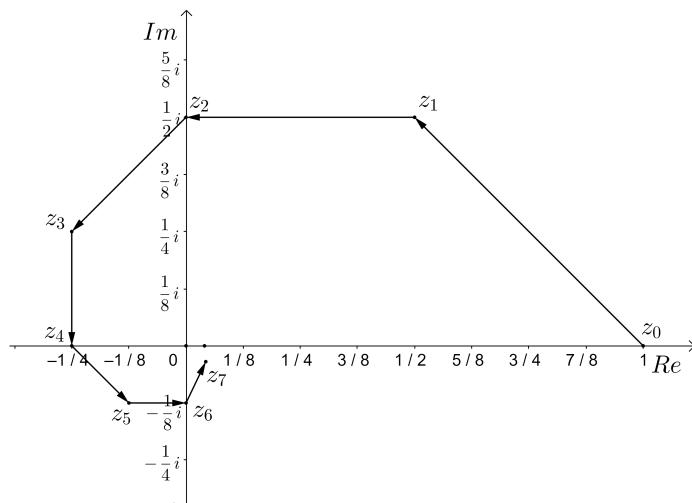
Obrázek 2.9: Znázorněno prvních pět členů posloupnosti.

Pro druhý případ zvolme naopak $|k| < 1$.

$$k = \frac{1}{2} + \frac{1}{2}i:$$

$$\begin{aligned}z_0 &= 1 \\z_1 &= \left(\frac{1}{2} + \frac{1}{2}i\right) \cdot 1 = \frac{1}{2} + \frac{1}{2}i \\z_2 &= \left(\frac{1}{2} + \frac{1}{2}i\right) \cdot \left(\frac{1}{2} + \frac{1}{2}i\right) = \frac{i}{2} \\z_3 &= \left(\frac{1}{2} + \frac{1}{2}i\right) \cdot \frac{i}{2} = -\frac{1}{4} + \frac{1}{4}i \\z_4 &= \left(\frac{1}{2} + \frac{1}{2}i\right) \cdot \left(-\frac{1}{4} + \frac{1}{4}i\right) = -\frac{1}{4} \\&\vdots\end{aligned}$$

Z obrázku 2.10 je zřejmé, že posloupnost se stále více přibližuje počátku.



Obrázek 2.10: Znázorněno prvních sedm členů posloupnosti.

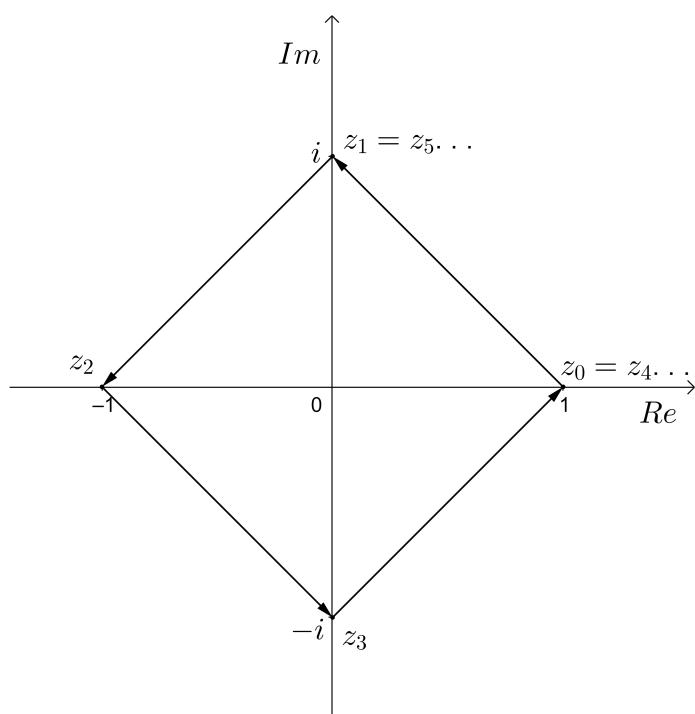
Můžeme také říci, že posloupnost konverguje k *přitahujícímu pevnému bodu*, o tomto pojmu však více až v sekci 3.2.2.

Jako poslední volba je $|k| = 1$.

$k = i$:

$$\begin{aligned} z_0 &= 1 \\ z_1 &= i \cdot 1 = i \\ z_2 &= i \cdot i = -1 \\ z_3 &= i \cdot (-1) = -i \\ z_4 &= i \cdot (-i) = 1 \\ z_5 &= i \cdot 1 = i \\ z_6 &= i \cdot i = -1 \\ &\vdots \end{aligned}$$

Můžeme si všimnout, že se posloupnost dostává po čtyřech krocích do cyklu délky 4, který se takto opakuje až do nekonečna, což nám potvrzuje i obrázek 2.11.



Obrázek 2.11: Cyklus délky čtyři pro posloupnost z .

2.2.2 Nelineární iterace

Nyní si už zadefinujme nelineární iterační proces tvaru, jakým budeme generovat Mandelbrotovu množinu a Juliovu množinu.

$$z \mapsto z^2 + c. \quad (2.13)$$

Ekvivalentní zápis, lépe vysvětlující fungování při programování iteračních procesů definujeme jako

$$z_{n+1} = z_n^2 + c.$$

Vzhledem k tomu, že už víme, jak funguje iterace v lineárním tvaru, nebudeme se již kvadratickou iterací příliš dlouho zabývat, protože všechny náležitosti jsou zde velmi podobné, ne-li stejné. Jako u lineární iterace předpokládejme, že $c = 0$. Je důležité si uvědomit, že pro čísla z pro které platí $|z_0| < 1$ posloupnost konverguje k nule. Volme například $z_0 = \frac{1}{2}$:

$$\begin{aligned} z_0 &= \frac{1}{2} \\ z_1 &= z_0^2 = \left(\frac{1}{2}\right)^2 = \frac{1}{4} \\ z_2 &= \frac{1}{16} \\ &\vdots \end{aligned}$$

Pro $|z_0| > 1$ posloupnost diverguje k nekonečnu. Demonstrujme pro $z_0 = 2$:

$$\begin{aligned} z_0 &= 2 \\ z_1 &= z_0^2 = 4 \\ z_2 &= 16 \\ &\vdots \end{aligned}$$

Nakonec pro $|z_0| = 1$, resp. $|z_0| = 0$ se posloupnost skládá pouze z hodnot 1, resp. 0. Tedy nechť $z_0 = 1$:

$$\begin{aligned} z_0 &= 1 \\ z_1 &= 1^2 = 1 \\ z_2 &= 1 \\ &\vdots \end{aligned}$$

Pro komplexní čísla je mnoho dalších možností, jak posloupnost může vypadat. O tom je však pojednáváno v sekci [3.2.1](#).

Kapitola 3

Juliovy množiny

Následující kapitola rozebírá hlavní téma celé práce a to právě Juliovy množiny. Nejprve seznámí čtenáře se základními pojmy a náležitostmi, aby byl následně bez problému schopen porozumět vlastnostem v druhé části kapitoly. Pravdivost informací je ve velké míře založena na zdrojích [1], [4], [7] a také [15], [16] a [24]. Další zdroje využity v menší míře jsou vždy uvedeny přímo u daného faktu.

3.1 Historie

Objevitelem Juliových množin je francouzský matematik Gaston Maurice Julia. Vyrůstal v západním Alžírsku, kam se jeho rodina přestěhovala ze Španělska. Po vystudování katelické školy v Oranu dostal stipendium na střední školu v Paříži. Ve svých osmnácti letech byl první v přijímacím řízení na školy École polytechnique a École normale supérieure, které mají podobnou prestiž, jako například americký MIT. Nastoupil na École normale supérieure, avšak první světová válka jeho studium předčasně ukončila.

V roce 1915 byl poslán do zákopů jako poručík pěchoty, bohužel při jednom z útoků Němců dostal zásah kulkou do hlavy a přišel tak o nos a skoro oslepl. Z tohoto důvodu je na fotografiích vyobrazen s černou páskou přes nos.

Během pobytu v nemocnici získal doktorát z přírodních věd a o dva roky později v roce 1918 vydal svou nejznámější studii o iteraci racionálních funkcí *Mémoire sur l'itération des fonctions rationnelles*, kde mimo jiné představil právě Juliovu množinu. V této době nezávisle na Juliově prováděl podobou studii i francouzský astronom a matematik Pierre Fatou, po kterém je také pojmenován jeden typ Juliových množin. Bohužel Julia a Fatou předběhli svou dobu. Bez počítačů a výkonných zařízení nebylo možné zcela naplnit potenciál Juliových množin, a tak se tyto studie nepotkaly s oblastí zájmu čtenářů. Až v 80. letech se opět dostaly na světlo světa díky Benoît Mandelbrotovi, který Juliovy množiny jako první vykreslil pomocí počítače ve vývojovém středisku IBM a ukázal tak pravou krásu těchto fraktálních obrazců.

3.2 Dualita

Definice 3.1. [17] Označme \mathbb{X} jako metrický prostor (X, d) . (Diskrétní) dynamický systém je potom definován jako uspořádaná dvojice (\mathbb{X}, f) , kde \mathbb{X} je metrický prostor, nazývaný jako fázový (stavový) prostor a $f : \mathbb{X} \rightarrow \mathbb{X}$ označuje zobrazení z počátečního stavu $x_0 \in \mathbb{X}$ na nějaký stav $f^k(x_0)$ ¹ $\in \mathbb{X}$ pro libovolné $k = 1, 2, \dots, n$.

Juliova množina je nelineární dynamický systém definovaný na \mathbb{C} , řídící se nekonstantním zobrazením $f : \mathbb{C} \rightarrow \mathbb{C}$, kde f je racionální funkcí komplexní paraboly

$$f(z) = z^2 + c \quad (3.1)$$

tvořena podílem dvou komplexních polynomů.

Juliovu množinu pak vytváříme iterováním f , tedy pro fixní c vybereme z_0 , provedeme iteraci a výsledek iterace dosadíme za z_1 a pro stejně fixní c operaci provádíme stále dokola čímž získáváme iterační proces

$$z \mapsto z^2 + c \mapsto (z^2 + c)^2 + c \dots$$

Stejný postup provádíme pro všechna $z_0 \in \mathbb{C}$, ale stále pro jedno konkrétní zafixované c . Ve výše zmíněném iteračním procesu tvoří jednotlivé iterační kroky posloupnost hodnot z_0, z_1, \dots, z_n , viz sekce 2.2. Pro takovou posloupnost platí jedna ze dvou následujících variant, čemuž se říká *dualita*, běžně používaný je také anglický ekvivalent *dichotomy*:

- **posloupnost je ohraničená** - existuje kruh², který obsahuje všechny hodnoty posloupnosti,
- **posloupnost nelze ohraničit** - pro libovolně velký kruh vždy existují hodnoty posloupnosti mimo kruh.

Komplexní rovina se touto vlastností rozdělí na dvě podmnožiny. Podmnožinu, která obsahuje všechny počáteční hodnoty ohraničitelných posloupností - *Prisoner set*, značíme ji P_c pro daný parametr c a podmnožinu všech počátečních hodnot pro které nelze posloupnost ohraničit - *Escape set*, značíme ji jako E_c taktéž pro daný parametr c .

Platí, že obě tyto podmnožiny jsou neprázdné. Jednoduše se to dá odůvodnit tak, že pro dané c , pokud vybereme z komplexní roviny z takové, pro které platí, že $z = z^2 + c$, pak posloupnost bude ohraničitelná, protože se bude skládat pouze ze samých z a tedy toto z bude náležet prisoner set. Naopak zvolíme-li dostatečně velké z , například libovolné $|z| > 2$, pak $z^2 + c$ bude ještě větší, tedy se budou hodnoty posloupnosti neustále zvětšovat a z toho plyne, že nejsme schopni tuto posloupnost ohraničit, takže i escape set nebude prázdná množina. Můžeme také říct, že P_c je doplněk množiny E_c do množiny \mathbb{C} a samozřejmě i naopak.

Definovaní P_c a E_c bylo velmi důležité, protože pomocí nich můžeme definovat Juliovu množinu, kterou budeme značit J_c . Uvědomme si totiž, že P_c a E_c musí mít stejnou hranici a touto společnou hranicí dvou podmnožin z komplexní roviny je právě Juliova množina.

¹Jedná se složené zobrazení. Tedy platí $f^k(x_0) = \underbrace{f \circ f \circ \dots \circ f}_{k \text{ krát}}(x_0)$, viz [4, strana 215].

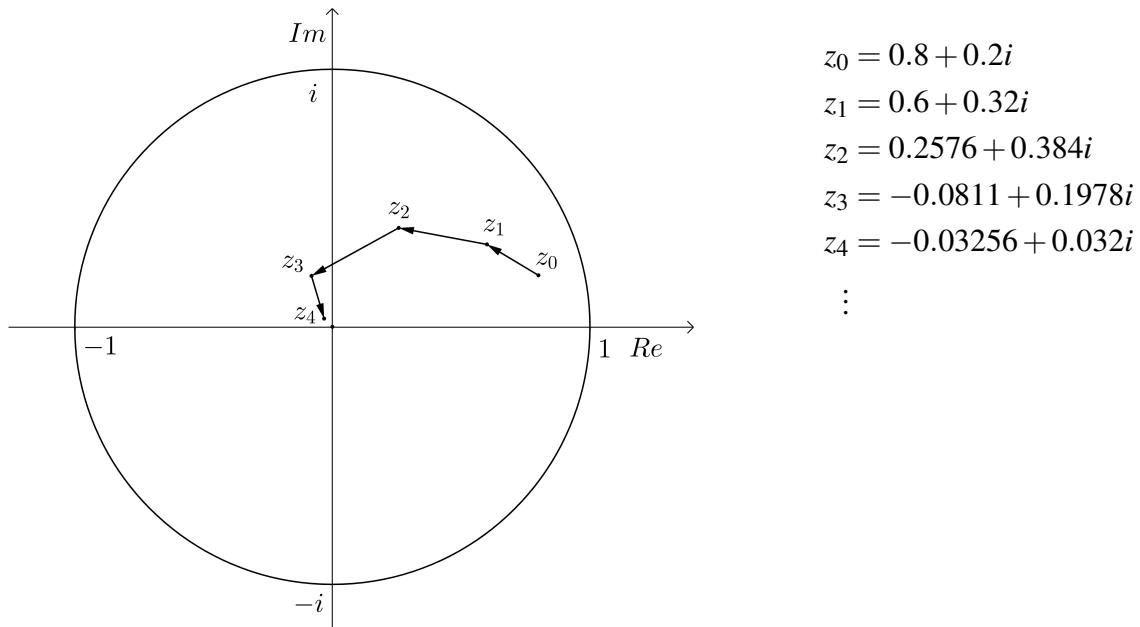
²Kruhem v komplexním oboru je myšlena množina $D \subset \mathbb{C}$ všech bodů x , splňující $D_r(s) = \{x \in \mathbb{C} \mid |x-s| \leq r\}$, kde $r > 0$ je poloměr kruhu se středem s .

3.2.1 Orbita

Definice 3.2. [2] *Orbita* bodu $z_0 \in \mathbb{C}$ vzhledem k zobrazení f je posloupnost hodnot

$$\{f^n(z_0)\}_{n=0}^{\infty}.$$

Velmi často se pro jednodušší zkoumání orbit využívá grafického znázornění. Tedy, že hodnoty posloupnosti vyneseme do grafu komplexní roviny. Jak graficky znázornit orbitu rozebírá sekce 2.2, zde si jen na obrázcích 3.1, 3.2 a 3.3 ukážeme, jak ovlivňuje orbitu pozice počáteční hodnoty v závislosti na výše definované dualitě. Pro jednoduchost je volba $c = 0$. Počáteční hodnota z_0 a prvních pár iterací jsou znázorněny vždy vedle obrázku. V popisku obrázku pak vidíme obecně chování z_0 ve třech možných variantách pozic na komplexní rovině (viz [4, strana 216]).

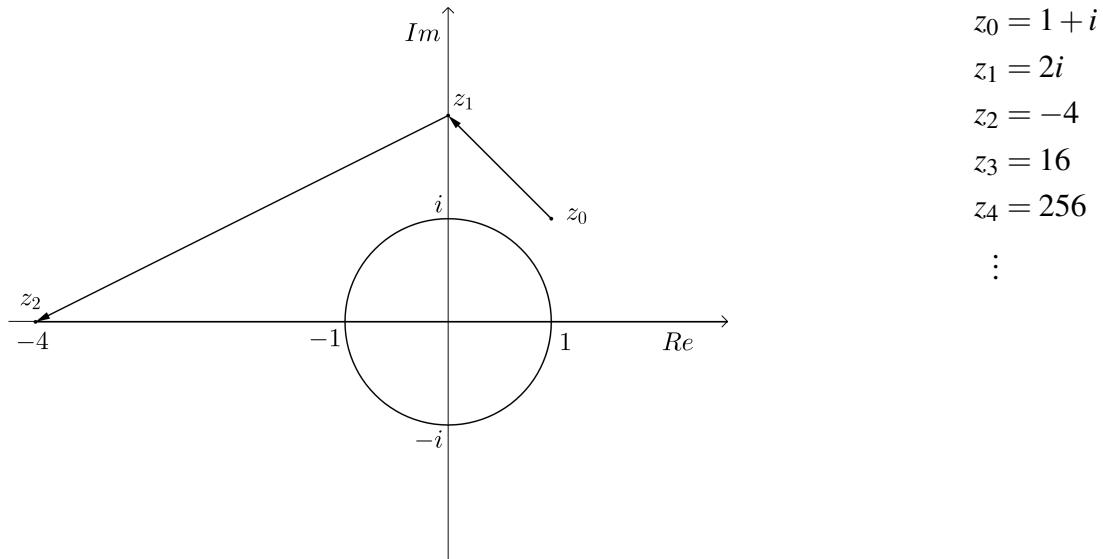


Obrázek 3.1: $z_0 \in P_c$. Orbita konverguje. Konkrétně pro $c = 0$ k bodu nula.

Může také nastat jeden speciální případ a to ve chvíli, kdy $z_0 \in J_c$. Jak je vidno, pro všechny takové body platí, že jejich orbita je uvězněna v Juliově množině. Pro některé počáteční hodnoty pak platí, že jejich orbita je tzv. *periodická*³, tedy, že se opakuje určitý počet výsledných hodnot po iteraci stále dokola, v přesně daném pořadí. Například pro obrázek 2.11

$$1 \mapsto i \mapsto -1 \mapsto -i \mapsto 1 \dots$$

³Formálněji v definici 3.2.



Obrázek 3.2: $z_0 \in E_c$. Orbita diverguje k nekonečnu.

Tedy vidíme, že se pro cyklus délky čtyři v posloupnosti opakují pouze čtyři hodnoty.

$$\begin{aligned}f(z_0) &= z_1 \\f(z_1) &= z_2 \\f(z_2) &= z_3 \\f(z_3) &= z_4 \\f(z_4) &= z_1 \\&\vdots\end{aligned}$$

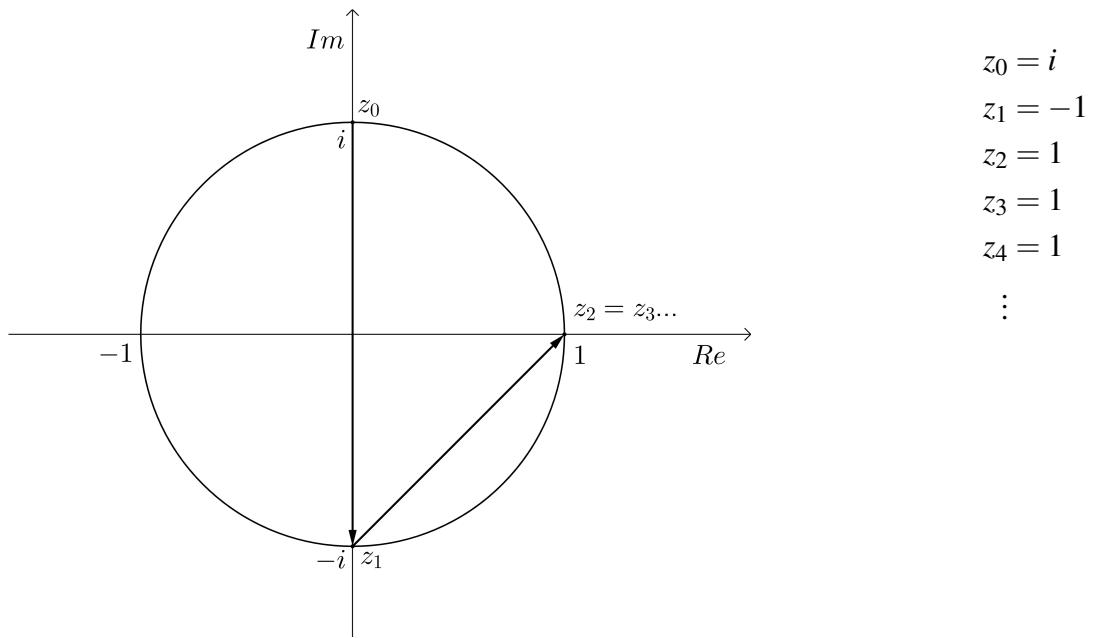
Zadefinujme ještě pojem kritický bod a s ním spojenou kritickou orbitu, které budou hrát významnější roli až v následující kapitole, kde pomocí nich můžeme definovat Mandelbrotovu množinu. Pro zformulování vět o vlastnostech Juliových množin však je definice nutná již nyní.

Definice 3.3. [4] Bod $z \in D(f)$ ⁴ nazveme *kritický bod* funkce f , jestliže v tomto bodě neexistuje derivace ⁵ f , nebo $f'(z) = 0$. Orbita z se pak nazývá *kritická orbita*.

Konkrétně pro (3.1) je první derivace $f'(z) = 2z$ a potom jediné řešení $2z = 0$ je $z = 0$, což je tedy kritický bod pro funkci iteračního předpisu Mandelbrotovy a Juliový množiny.

⁴Definiční obor funkce f .

⁵Derivace funkce komplexní proměnné viz [11, kapitola 2].



Obrázek 3.3: $z_0 \in J_c$. Orbita zůstává stále v Juliově množině.

3.2.2 Pevné body

Pro porozumění některým z následujících vlastností Juliových množin si zadefinujme co to je pevný bod a jaké jsou jeho základní typy.

Definice 3.4. [3] Bod $z \in \mathbb{C}$ se nazývá *pevný bod* zobrazení f , pokud

$$f(z) = z.$$

Jinými slovy, pokud je vstupní hodnota pevným bodem, pak se po transformaci jeho hodnota nezmění. Z definice 3.4 plyne, že hledáme takový bod, který splňuje

$$z = z^2 + c.$$

Což se dá zapsat také jako

$$0 = z^2 - z + c.$$

Pro nalezení pevného bodu tedy musíme najít kořeny polynomu druhého stupně.

Zadefinujme si nyní pojem *periodický bod*, který je spojen s periodickou orbitou zmíněnou výše.

Definice 3.5. [3] Bod $z \in \mathbb{C}$ se nazývá *periodický bod* s periodou n zobrazení f , pokud

$$f^n(z) = z \wedge f^m(z) \neq z \quad \forall m, n \quad 1 \leq m < n. \quad (3.2)$$

Orbita periodického bodu zobrazení f se nazývá *cyklus*. Nejmenší n splňující (3.2) určuje počet různých hodnot v cyklu.

Je vhodné připomenout, že f^n neznačí n -tou mocninu funkce f , ale složené zobrazení f . Např. $f(f(f(z))) = f^3(z)$, tedy pak obecně pro n kroků iteračního procesu skládáme n zobrazení, a proto f^n . Potom zcela intuitivně podle definice 3.4 můžeme říci, že pevný bod, je také periodický bod s periodou jedna.

Definice 3.6. [18] Pevný bod nazveme *přitahující*, jestliže existuje okolí⁶ pevného bodu, pro které platí, že vybereme-li libovolný bod z tohoto okolí, pak jeho orbita konverguje k tomuto pevnému bodu.

Jinými slovy, pokud víme, že například x je pevný bod, pak pro malé ϵ platí, že zvolíme-li počáteční hodnotu

$$z_0 = x + \epsilon \quad x, \epsilon \in \mathbb{C},$$

kde $x + \epsilon$ náleží okolí přitahujícího pevného bodu, pak orbita z_0 konverguje k x .

Definice 3.7. [19] Pevný bod nazveme *odpuzující*, jestliže existuje okolí pevného bodu, pro které platí, že vybereme-li libovolný bod z tohoto okolí, vyjma daného pevného bodu, pak jeho orbita po k iteracích opustí toto okolí.

Následující věta, která se bude hodit v praktické části práce, rozebírá stabilitu pevného bodu pomocí derivace funkce v bodě.

Věta 3.2.1. [15] Je-li x pevný bod funkce f diferencovatelné v bodě x , potom platí:

- Jestliže $|f'(x)| < 1$, pak je x přitahující pevný bod.
- Jestliže $|f'(x)| > 1$, pak je x odpuzující pevný bod.
- Jestliže $|f'(x)| = 1$, pak je x neutrální pevný bod.

Doplňme ještě definici stability periodického bodu.

Definice 3.8. [15] Periodický⁷ bod x zobrazení f se nazývá přitahující (resp. odpuzující, nebo neutrální), pokud je x přitahující (resp. odpuzující, nebo neutrální) pevný bod pro f^n .

3.2.3 Oblast přitažení

Definice 3.9. [18] Je-li x přitahující pevný bod funkce f , potom se největší možné okolí $\mathcal{O}(x)$ bodu x , pro které platí

$$f^n(z) \rightarrow x,$$

kde $z \in \mathcal{O}(x)$ pro $n \rightarrow \infty$. nazývá *oblast přitažení* bodu x .

⁶Exaktní definici okolí bodu najeznete v [1, strana 23].

⁷O stabilitě neutrálního pevného nebo periodického bodu je třeba rozhodovat pomocí druhé, někdy až třetí derivace. Podrobnosti najeznete v [20, strana 95].

Pojem oblast přitažení je velmi často používán v anglickém ekvivalentu *Basin of attraction*. Basin je v překladu povodí. Tento velmi výstižný překlad může pomoci k pochopení definovaného pojmu. Povodí je totiž oblast, z které se voda slévá do jedné konkrétní řeky, či jezera. Tedy oblast přitažení, je oblast, z které se všechny orbity počátečních hodnot této oblasti „slévají“ do jednoho přitahujícího pevného bodu.

Z výše zmíněných faktů můžeme interpretovat escape set a prisoner set také jako oblasti přitažení. Protože pro escape set platí, že orbita všech počátečních hodnot diverguje do nekonečna, tedy vlastně escape set je oblast přitažení pro bod ∞ . Naopak prisoner set je oblast přitažení určitého přitahujícího pevného bodu uvnitř prisoner set.

Uveďme si nyní jednoduchý příklad, kterým se všechny zadefinované pojmy spojí dohromady: Zvolme $c = 0$ a hledejme pevné body. Tedy řešíme rovnici

$$\begin{aligned} z &= z^2 + 0 \\ 0 &= z^2 - z \\ 0 &= z(z - 1). \end{aligned}$$

Našli jsme dva kořeny z_1 a z_2 , což odpovídá dvěma pevným bodům $z_1 = 0$ a $z_2 = 1$. Nyní chceme vyšetřit, který pevný bod je přitahující a který odpuzující. Sledujeme tedy hodnotu derivace funkce f v pevném bodě. Tedy

$$\begin{aligned} f(z) &= z^2 + c \\ f'(z) &= 2z. \end{aligned}$$

Pro jednotlivé pevné body pak

$$\begin{aligned} |f'(z_1)| &= |2z_1| = |2 \cdot 0| = 0, \\ |f'(z_2)| &= |2z_2| = |2 \cdot 1| = 2. \end{aligned}$$

Vidíme tedy, že $|f'(z_1)| < 1$, tedy 0 je přitahující pevný bod, naopak $|f'(z_1)| > 1$, tedy 1 je odpuzující pevný bod.

Můžeme tedy říct, že pro tento konkrétní případ je escape set oblast přitažení pro bod ∞ a prisoner set je oblast přitažení pro bod 0. Příklad je graficky znázorněn na obrázcích 3.1 a 3.3. Všimneme si, že prisoner set je vnitřek kruhu o poloměru jedna a escape set, je vše ostatní, kromě hranice kruhu. Pro $c = 0$ je hranice kruhu Juliovu množinu. Je to jediná varianta, kde J_c nemá fraktální tvar, jak si ukážeme dále. Z obrázků je také zřejmé, že orbita pro počáteční hodnotu uvnitř prisoner set konverguje k jejímu přitahujícímu pevnému bodu, naopak volba počáteční hodnoty z escape set zapříčiní divergenci k nekonečnu.

3.3 Limitní poloměr divergence

V předchozích částech jsme se zabývali Juliovou množinou pro konstantu $c = 0$ a řekli jsme, že Juliovu množinu je hranicí P_c a E_c . Zjistit, jak vypadá escape set pro tento konkrétní případ bylo triviální, protože si stačilo uvědomit, že všechny hodnoty větší než jedna po opakování umocnění budou růst do nekonečna. Tedy stačilo najít takovou hodnotu orbity,

která bude mít velikost větší jak jedna.

Vlastnost, že J_c je hranicí P_c a E_c platí i pro libovolné c . Jak už bylo poznamenáno výše, pro $c \neq 0$ již Juliova množina má fraktální tvar, tedy nalezení E_c bude značně obtížnější a pro grafické zobrazení bude potřeba určitého zařízení, které je přesnější a rychlejší, než ruka s tužkou a papírem. Je logické, že pro dostatečně velká z umíme i s libovolným c říct, že orbita diverguje, protože význam konstanty c pro velká z po umocnění přestává být důležitý. Naskytá se ale otázka, jak najít hranici, kde budeme pro každé z schopni spolehlivě říct, že pokud hodnota orbity přesáhne tuto hranici, pak orbita diverguje k nekonečnu. K zodpovězení této otázky vznikl pojem *Limitní poloměr divergence*.

Tvrzení 3.3.1. *Pro kvadratické zobrazení $f(z) = z^2 + c$ platí, jestliže*

$$|z| > \max(|c|, 2),$$

pak orbita z diverguje k nekonečnu. Hodnota $r(c) = \max(|c|, 2)$ se nazývá limitní poloměr divergence.

Důkaz. Využijeme zápisu iteračního předpisu ve tvaru $z_{n+1} = z_n^2 + c$. Ukažme, že pro libovolné $|z_n| > r(c)$ orbita diverguje k nekonečnu. Předpokládejme tedy, že $|z_n| > |c|$ a zároveň $|z_n| > 2$. Pak jistě existuje malé kladné číslo $\varepsilon > 0$ pro které platí

$$|z_n| = 2 + \varepsilon. \quad (3.3)$$

Nyní pomocí inverzní trojúhelníkové nerovnosti $|A + B| \geq ||A| - |B||$ máme

$$|z_{n+1}| = |z_n^2 + c| \geq \left| |z_n^2| - |c| \right|^{\textcolor{red}{8}} = |z_n^2| - |c|.$$

Protože $z_n^2 \geq 0$, pak určitě platí $|z_n^2| = |z_n|^2$ a navíc z předpokladu $|z_n| > |c|$ plyne

$$|z_n|^2 - |c| \geq |z_n|^2 - |z| = |z_n|(|z_n| - 1).$$

Konečně podle (3.3) můžeme psát

$$|z_n^2 + c| \geq |z_n|(|z_n| - 1) = |z_n|(1 + \varepsilon).$$

Vidíme tedy, že $|z_{n+1}|$ bude $(1 + \varepsilon)$ -krát větší než $|z_n|$. To stejně zřejmě platí i pro $|z_{n+2}|$, kde dostáváme

$$|z_{n+2}| = |z_{n+1}|(1 + \varepsilon).$$

Z toho plyne, že potom

$$|z_{n+2}| = |z_n|(1 + \varepsilon)^2.$$

Pomocí matematické indukce pak dostáváme vztah pro libovolné $k = 0, 1, 2, \dots$

$$|z_{n+k}| = |z_n|(1 + \varepsilon)^k.$$

A protože víme, že $(1 + \varepsilon) > 1$, potom se hodnota $(1 + \varepsilon)^k$ bude zvětšovat při každé iteraci, a tedy pro $k \rightarrow \infty$ bude orbita z_n divergovat k nekonečnu. \square

Tímto tvrzením se mnohonásobně zrychlí celý proces zjišťování, zda-li bod náleží E_c , či nikoliv. Nemusíme totiž počítat vždy všech n iterací pro jednotlivou počáteční hodnotu, ale pokud již například z_2 překročí $r(c)$, řekneme, že tato počáteční hodnota náleží E_c a můžeme testovat další hodnotu.

⁸Protože je $|z| > |c|$, pak i $|z^2| > |c|$, což znamená, že výsledek bude vždy kladný, nezávisle na vnější absolutní hodnotě.

3.3.1 Aproximace prisoner set

Pojem limitního poloměru divergence máme zadefinovaný, a tak nám nebrání nic v představení algoritmu odhadujícího výsledný tvar E_c respektive P_c právě pomocí $r(c)$.

Pro zvolené c nejprve spočítáme $r(c)$. Pro počáteční hodnotu je tedy limitní poloměr divergence kruh o poloměru $r(c)$. Tímto jsme získali první approximaci výsledného prisoner set. Proces opakujeme, čili tentokrát testujeme, které počáteční hodnoty mají velikost po první iteraci větší než $r(c)$ a ty z výsledného tvaru vyloučíme. Je potřeba dbát na to, že vylučujeme počáteční hodnoty z_0 , ne hodnoty po k -té iteraci!

Opětovným testováním tak dostaváme stále lepší approximaci výsledného prisoner set. Tuto approximaci závislou na konstantě c definujme jako

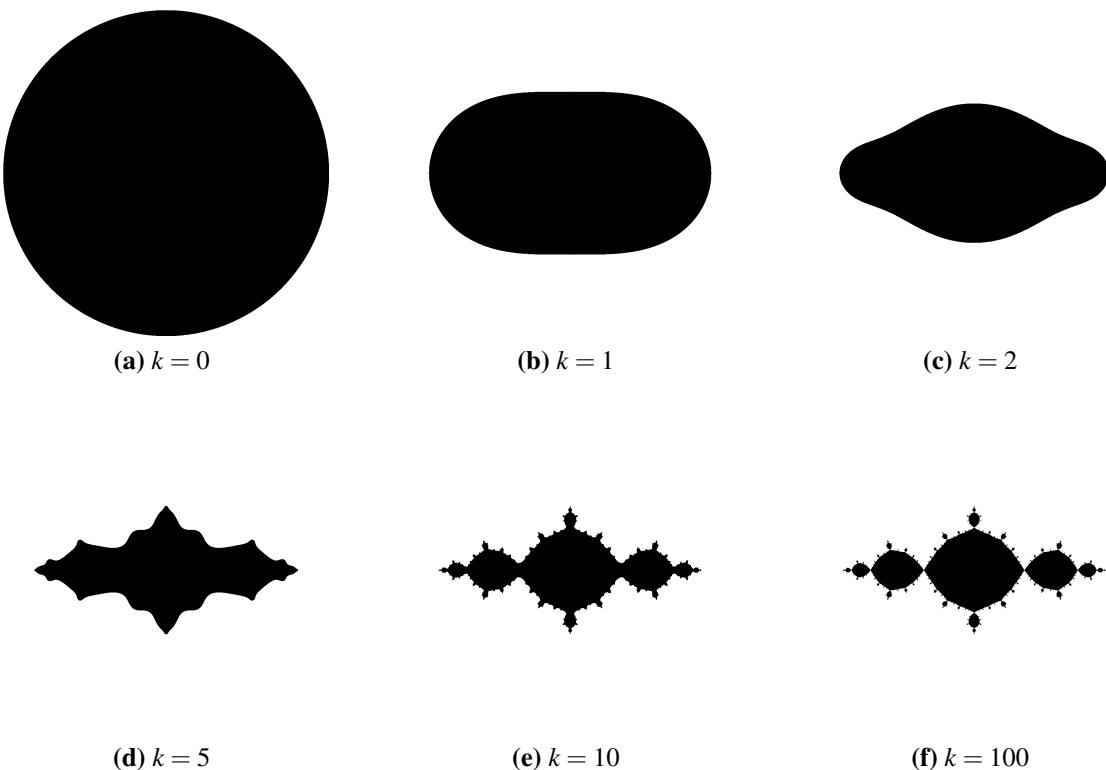
$$A_k(c) = \{z_0 \mid |z_k| \leq r(c)\},$$

kde k je počet proběhlých iteračních kroků, tedy k říká kolikátá je to approximace prisoner set. Logicky z toho plyne, že čím vyšší k , tím přesnější approximace.

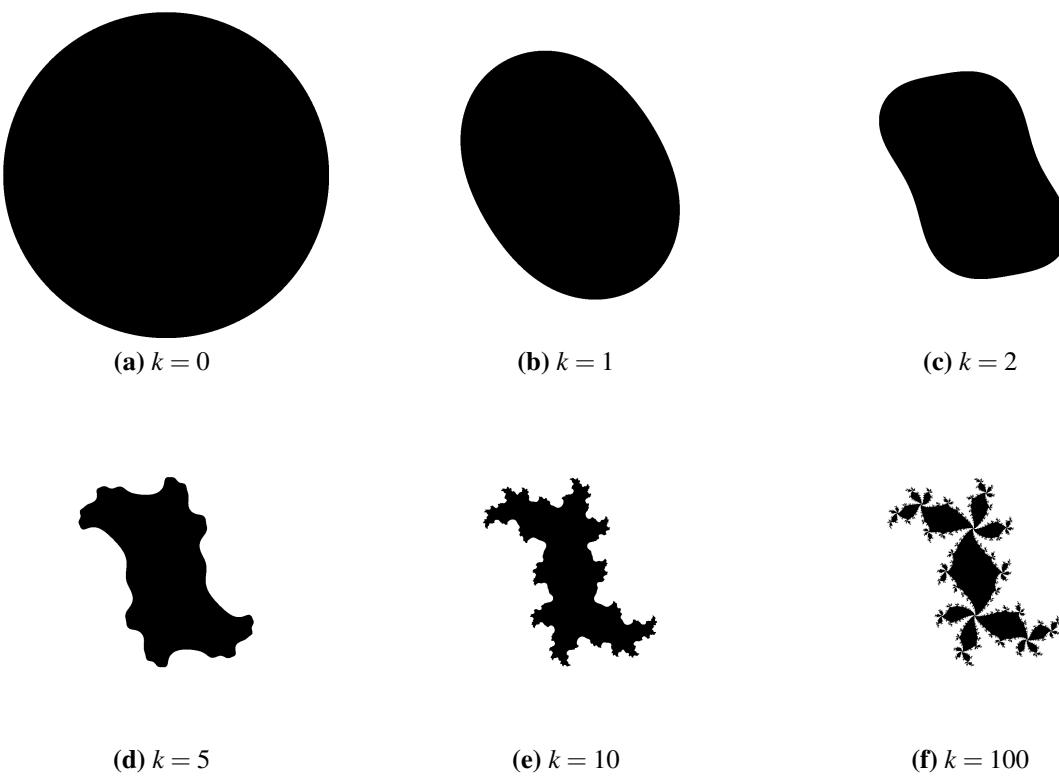
Navíc platí

$$\lim_{k \rightarrow \infty} A_k(c) = P_c.$$

My se samozřejmě v počítačové grafice musíme smířit s konečným počtem iterací (approximací). V praxi se tak volí nějaké dostatečně velké k , aby approximace byla co nejpřesnější, ale také proveditelná v rozumném časovém horizontu. Na obrázcích 3.4 a 3.5 lze vidět celý výše popsaný proces pro dvě různá c . Jak si lze všimnout, $A_0(c)$ je vždy kruh o poloměru $r(c)$ a následně se začíná tvarovat výsledný vzhled P_c . Otestoval jsem, že pro $k > 200$ již rozdíl ve výsledném tvaru není lidským okem patrný, naopak během prvních approximací vnímáme výrazné změny, které se postupně stabilizují.



Obrázek 3.4: Aproximace prisoner set pro $c = -1$.



Obrázek 3.5: Aproximace prisoner set pro $c = 0.2776 + 0.5445i$.

3.4 Fatouova množina

Nyní umíme najít tvar prisoner set, tedy z toho plyne, že víme také, jak vypadá escape set a pomocí těchto dvou množin dokážeme najít Juliovu množinu pro libovolné c . Pro Juliovu množinu obecně platí, že jakákoli nepatrná změna počáteční hodnoty z Juliovu množiny dokáže výrazně změnit chování orbity této hodnoty [15, strana 18], což vyplývá z informací v sekci 3.2.1, kde říkáme, že orbita jde k nekonečnu, pokud je $z_0 \in E_c$, konverguje k přitahujícímu pevnému bodu P_c , pokud $z_0 \in P_c$ a zůstává uvězněna v J_c , pokud $z_0 \in J_c$. A protože J_c je hranicí⁹, kterou lze definovat tak, že v libovolně malém okolí bodu z této hranice musí ležet jak bod z P_c , tak i bod z E_c , stačí nepatrná změna z_0 a orbita okamžitě změní své chování.

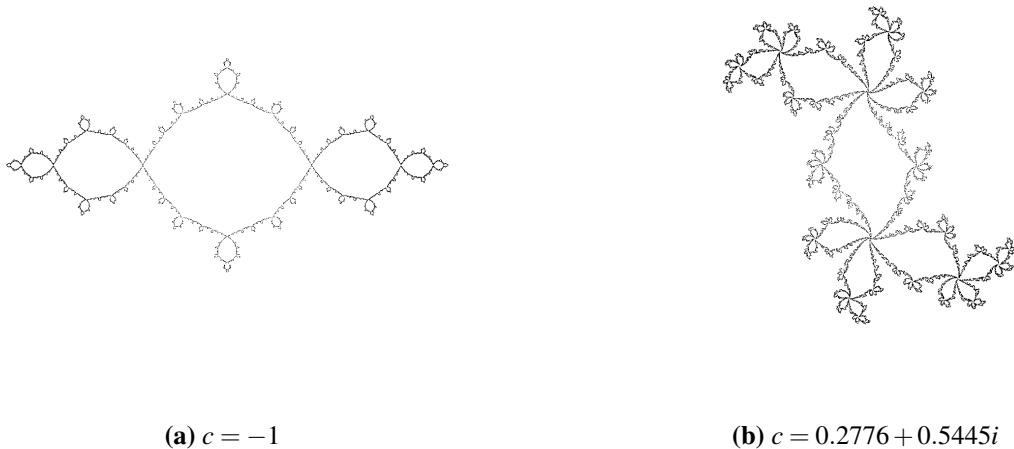
Naopak pro escape set a prisoner set tato vlastnost neplatí. Zde spíše naopak můžeme tvrdit, že změna v počáteční hodnotě nijak výrazně průběh orbity nezmění. V tomto se chovají P_c a E_c stejně, můžeme je tedy sjednotit a vzniká *Fatouova množina*, kterou značíme F_c a platí, že

$$F_c = \mathbb{C} \setminus J_c. \quad (3.4)$$

Pokud bychom chtěli z obrázku 3.6 určit Juliovu množinu a Fatouovu množinu, pak vždy hodnoty obarvené černou barvou tvoří Juliovu množinu a vše ostatní zvýrazněné bíle, Fatouovu množinu.

3.5 Juliovu množinu vs. Filled Juliovu množinu

Jak už bylo řečeno několikrát výše, Juliovu množinu je hranicí mezi P_c a E_c . Tedy konkrétně pro approximace zobrazené na 3.4 a 3.5 je J_c uvedena na obrázku 3.6.



Obrázek 3.6: J_c jako hranice K_c .

⁹Exaktní definici hranice množiny najdete v [1, strana 24].

Otázkou je, zda-li existuje název i pro Juliovu množinu s „vnitřkem“. Stejně jako spousta článků jsme i my definici Juliovy množiny jako hranice přehlíželi a nerozlišovali s Juliovou množinou i s oblastí přitažení uvnitř.

Korektní název pro sjednocení prisoner set a Juliovy množiny je *filled* Juliová množina a značíme ji K_c . Můžeme tedy také říct, že filled Juliová množina je doplněk oblasti přitažení nekonečna do komplexní roviny. Nyní konečně uvedeme formální definici K_c a J_c .

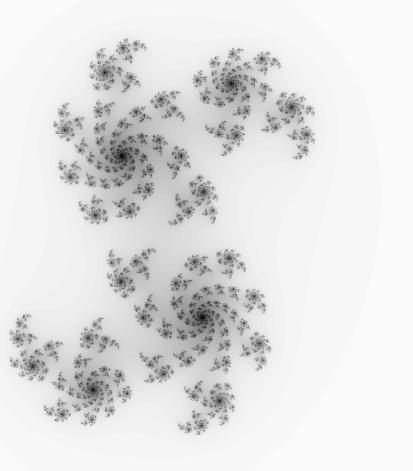
Definice 3.10. [4] Nechť $f : \mathbb{C} \rightarrow \mathbb{C}$ je polynomická funkce, pak *filled Juliová množina* K_c funkce f je množina počátečních hodnot, jejichž orbita nediverguje. Juliová množina J_c funkce f je potom hranicí K_c .

3.5.1 Nejznámější filled Juliovy množiny

Vzhledem k tomu, že iterační proces pro každé $c \in \mathbb{C}$ generuje odlišnou (filled) Juliovu množinu, i když pro opravdu velká c již není vidět rozdíl, formálně existuje nekonečně mnoho (filled) Juliových množin. Bylo by tedy nemožné zde ukázat všechny z nich. Uvedeme si proto jen ty nejzajímavější a nejznámější typy filled Juliových množin. Spousta z nich má i své jméno, které většinou vystihuje jejich tvar. Můžeme se tak setkat s názvy jako *rabbit* (králík), či *dragon* (drak).

Cantor set

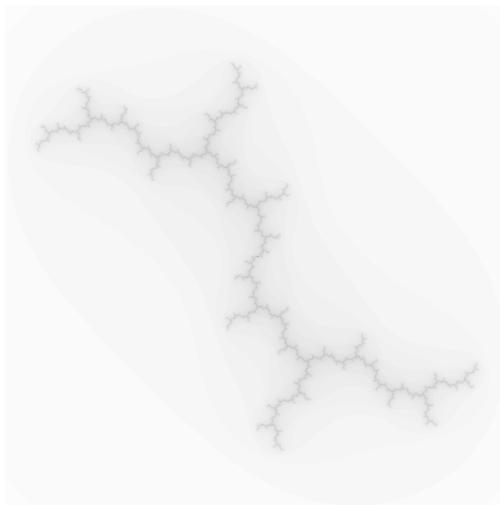
Filled Juliová množina nesoucí název Cantor set, nebo chcete-li česky Cantorova množina je nesouvislá množina bodů skládající s nekonečně mnoha částí, které mají nulovou plochu. Na první pohled to nemusí být úplně patrné, ale kdybychom tuto množinu přiblížovali, stále bychom nacházeli jen body, bez žádných souvislých oblastí. Můžeme se také setkat s názvem Fatou dust (prach), což velmi výstižně popisuje její tvar, viz obrázek 3.7.



Obrázek 3.7: Cantor set pro $c = 0.4 - 0.25i$.

Dendrite

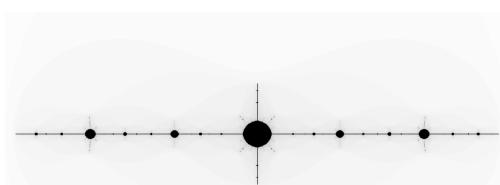
Pro K_c tohoto typu je stejně jako pro Cantor set typická nulová plocha. Na rozdíl od ní však je již souvislá, to tedy znamená, že K_c je jeden celek. Jinými slovy lze také souvislost definovat tak, že pokud vezmeme libovolné dva body z množiny, jsme schopni je spojit křivkou bez toho aniž bychom opustili množinu. Na první pohled se nulovost plochy může zdát podivná. Zamysleme se však nad tím, že podobnou vlastnost má z geometricky hladkých útvarů přímka, která je také souvislá, ale její plocha je stejně tak nulová. Dendrite vidíme na obrázku 3.8.



Obrázek 3.8: Dendrite pro $c = i$.

Airplane

Posuňme se zase o jeden příklad dál a ukažme si, jaké vlastnosti má K_c pojmenované jako letadlo. Stejně jako u Dendrite platí vlastnost souvislosti, navíc však už nemá nulovou plochu. Samotný tvar nás asi moc vzhledem k názvu nepřekvapí. Opět graficky znázorněno na 3.9.



Obrázek 3.9: Airplane pro $c = -1.756$.

Dragon

Stejně jako u předchozích dvou je i Dragon souvislý. Tento fakt už je mnohem více zřejmý, než u Dendrite a Airplane. I tak se stále jedná o velice členitý útvar s výraznou fraktální

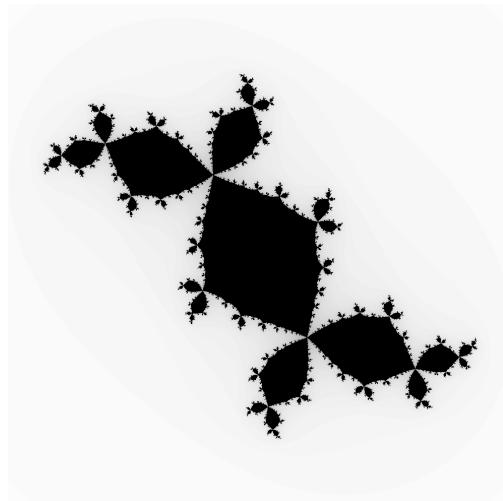
strukturou. Sám název opět nese informaci, co daný tvar připomíná, což s trochou fantazie můžeme vidět na obrázku 3.10.



Obrázek 3.10: Dragon pro $c = 0.36 + 0.1i$.

Rabbit

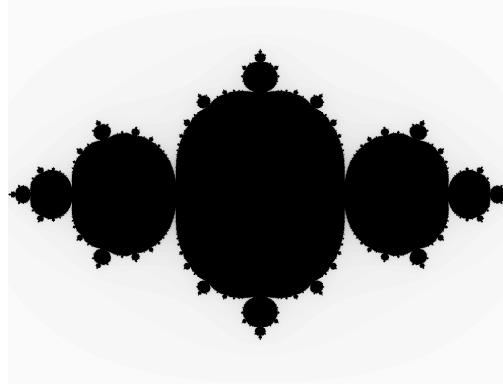
K_c pojmenovanou v překladu jako králík, můžeme často znát také pod názvem Douady's rabbit po známém francouzském matematikovi Adrienu Douadym, který se zabýval studiem komplexní dynamiky v letech 1935 – 2006. Vlastnosti už zůstávají identické jako u předchozího příkladu tedy souvislá množina s nenulovou plochou a znova velice členitým fraktálním tvarem připomínající králíka, viz 3.11.



Obrázek 3.11: Rabbit pro $c = -0.12 + 0.75i$.

San Marco

Posledním zajímavým případem je K_c nazývaná San Marco, nebo také Basilica pojmenovaná po Benoit B. Mandelbrotovi. O tomto významném matematikovi však až později. Podívejme se opět na geometrické znázornění na obrázku 3.12.



Obrázek 3.12: San Marco pro $c = -0.75$.

3.6 Vlastnosti Juliovu množiny

Tato sekce představí základní vlastnosti, nezmíněné výše, které budou převážně geometrického charakteru a jsou implementovány ve vizualizačních metodách rozebraných v sekci 5.3.1. Následující vlastnosti jsou ekvivalentní i pro filled Juliovu množinu, abychom však stále nemuseli opakovat jednu definici dvakrát, omezíme se pouze na Juliovu množinu a budeme předpokládat, že to stejně platí i pro filled Juliovu množinu.

Kompaktnost

Věta 3.6.1. *Množina $J \subseteq \mathbb{C}$ se nazývá kompaktní, právě tehdy, když je uzavřená a omezená.*

Důkaz. Viz [1]. □

Pokud bychom si představili, jak vypadá uzavřenosť na \mathbb{R} , pak se bavíme o uzavřeném intervalu. Uzavřenosť množiny na oboru komplexních číslech znamená uzavřenosť roviny.

Definice 3.11. [1] Množina $F \subseteq \mathbb{C}$ je *otevřená*, pokud pro každý bod $z \in F$ existuje okolí $\mathcal{O}(z)$ bodu z , pro které platí $\mathcal{O}(z) \subseteq F$. Řekneme, že množina $J \subseteq \mathbb{C}$ je *uzavřená*, pokud $\mathbb{C} \setminus J$ je *otevřená*.

Ekvivalentně lze definovat uzavřenou množinu jako množinu obsahující i svou hranici, a tedy již z definice J_c , kde říkáme, že se jedná o hranici P_c a E_c plyne, že Juliova množina je uzavřená. Naopak potom Fatouova množina¹⁰ je otevřená.

Definice 3.12. [1] Množina $J \subseteq M$ je *omezená* (ohraničená), existuje-li kruh K_r s poloměrem r a středem v počátku, pro který platí, že $J \subset K_r$.

¹⁰Za předpokladu platnosti (3.4).

Ohraničenost, nebo také omezenost chápejme jednoduše tak, že neexistuje bod množiny jehož velikost je větší jak zvolený poloměr kruhu.

Věta 3.6.2. *Juliovu množinu J_c funkce $f(z) = z^2 + c$ je kompaktní pro všechna $c \in \mathbb{C}$.*

Důkaz. Viz [21, strana 3]. □

Kompaktnost je velice důležitá vlastnost pro testování, jestli bod náleží Juliově množině, či nikoliv. Konkrétně využíváme hlavně omezenosti Juliových množin. Bez tohoto předpokladu by nebylo možné Juliovu množinu sestrojit.

Symetrie

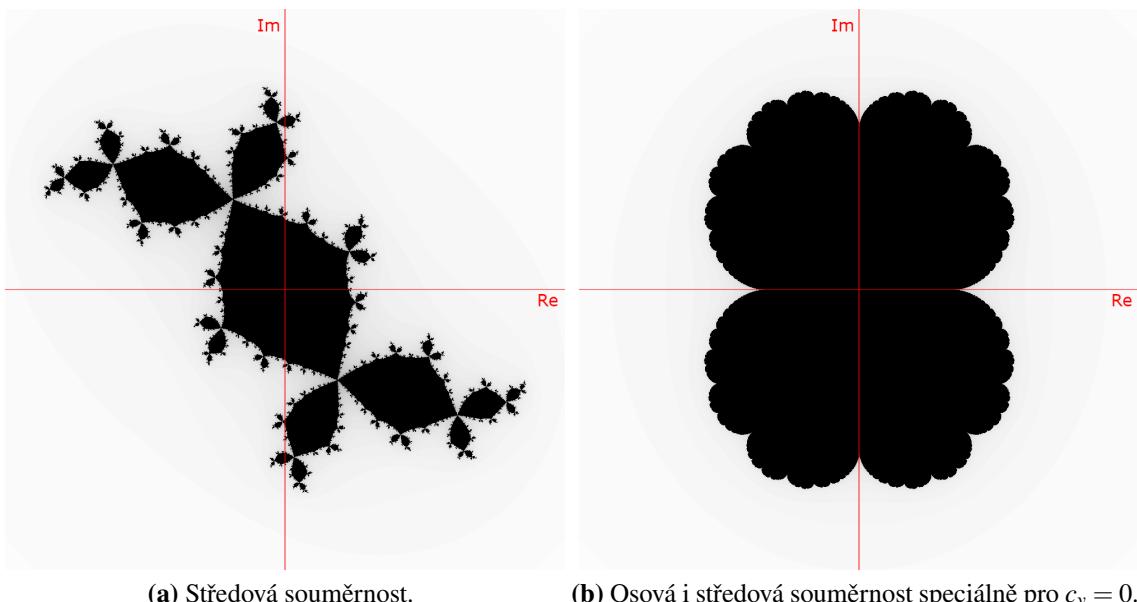
Věta 3.6.3. *Juliovu množinu je symetrická podle počátku, neboli*

$$z \in J_c \iff -z \in J_c.$$

Důkaz. Viz [21, strana 4]. □

Tato vlastnost je do velké míry zřejmá. Stačí si totiž uvědomit, jak vypadá iterační předpis Juliovu množiny $f(z) = z^2 + c$. Tedy vyberu-li libovolný z , potom symetrický bod vzhledem k počátku je $-z$. Vidíme, že po prvním iteračním kroku $f(-z) = f(z)$, protože umocněním opačného čísla k číslu z dostáváme stejný výsledek (viz pasáž 2.1.2), tedy jejich obory jsou od prvního iteračního kroku identické. Z toho plyne, že buď oba dva body náleží J_c , nebo oba nenáleží J_c , což je ekvivalentní se symetrií podle počátku jak můžeme vidět na obrázku 3.13a.

Navíc, je-li imaginární část c nulová, potom J_c splňuje i osovou souměrnost jak podle reálné, tak také podle imaginární osy 3.13b.



Obrázek 3.13: Symetrie K_c .

Základní dualita

Tento pojem vymezuje Juliovy množiny na dva základní typy. Prvním typem je zcela nesouvislá množina skládající se z nekonečně mnoha jednotlivých částí. Druhým typem je Juliova množina, která je jako jeden souvislý kus. Nejjazjímavější na tom však je, že neexistuje nic mezi tím. Tedy, že máme buď nekonečně mnoho částí Juliovy množiny, neboli Cantor set, či Fatou dust, nebo naopak pouze jeden kus, ale nemůže se stát, že by se Juliova množina skládala s konečného počtu kusů. Jestli se jedná o první či druhý typ, rozhodujeme na základě kritické orbity.

Věta 3.6.4 (Základní dualita). *Pro kvadratické zobrazení $f(z) = z^2 + c$ platí jedna z následujících dvou variant:*

- *Kritická orbita diverguje k nekonečnu a v tomto případě je Juliova množina kompletne nesouvislá, skládající se z nekonečně mnoha disjunktních částí, např. obrázek 3.7.*
- *Kritická orbita je ohraničená a v tomto případě je Juliova množina jednoduše souvislá¹¹, např. obrázek 3.11.*

Důkaz. Viz [15, strana 21]. □

Tato vlastnost má úzkou spojitost s Mandelbrotovou množinou o které bude řeč v kapitole 4.

Soběpodobnost

Jestliže fraktální útvar, nebo jeho část vypadá shodně, resp. podobně, nezávisle na jeho zvětšení, pak se bavíme o striktní, resp. statistické soběpodobnosti, často využívané je ekvivalentní slovní spojení *invariance vůči změně měřítka*. Do prvního typu patří například Sierpieńského trojúhelník¹². U Juliových množin se kvůli nelinearitě transformace nedá mluvit ani o jednom z výše zmíněných typů soběpodobnosti 1.7. Podobnost Juliových množin se dá lépe nazvat jako quasi-soběpodobnost, nebo také soběpříbuznost¹³. Menší kopie Juliovy množiny, nebo její části totiž nejsou přesnými, či podobnými kopiemi, ale jsou výrazně deformovány. Soběpodobnost J_c splňují pouze v určitých částech, konkrétně například v okolí Misiurewitzova bodu, o tom však později.

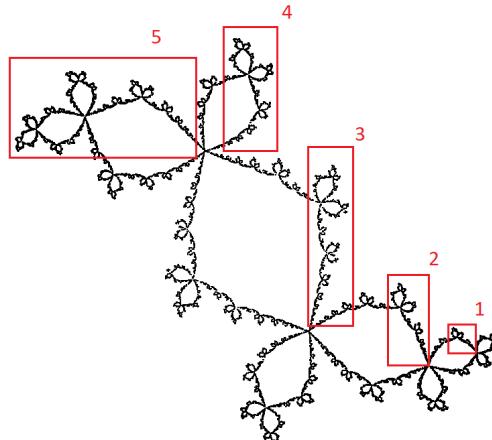
Pro Juliovy množiny platí jedna velice zajímavá skutečnost. Vybereme-li jakkoliv malou část Juliovy množiny například pomocí malého kruhu a na všechny body J_c obsažené v tomto výběru aplikujeme iterační proces, dostaneme obvykle větší podmnožinu dané Juliovy množiny. Opakujeme-li tento proces, dostaneme po konečném počtu takto provedených operací celou Juliovu množinu. To jinými slovy také říká, že již v libovolné malé části je obsažena globální struktura Juliovy množiny. Vidíme, že definice soběpodobnosti jde s menšími odchylkami aplikovat i na soběpříbuznost. Na obrázku 3.14 jsou znázorněny pomocí červených obdélníků soběpříbuzné podmnožiny. Navíc je zde také demonstrován

¹¹ Každou uzavřenou křivku uvnitř množiny jde stáhnout do bodu, který se také nachází v množině. Neformálně řekneme, že jednoduše souvislá je množina, která je tvořena jedním kusem bez „dér“ uvnitř množiny.

¹² Striktně soběpodobný fraktální útvar, který lze vidět na obrázku 1.1.

¹³ Více v sekci 1.2.

výše zmíněný fakt, že pokud na všechny body podmnožiny v obdélníku s číslem 1 aplikujeme iterační proces, dostaneme podmnožinu v obdélníku číslo 2. To samé platí pro všechny ostatní obdélníky.



Obrázek 3.14: Prvních pár transformací procesu získávání globální struktury J_c pomocí jejich podmnožin.

Pro následující vlastnosti již dále neplatí ekvivalence mezi J_c a K_c . Juliova množina bude v této sekci také někdy označována $J_c(f)$, čímž bude kladen důraz na funkci f podle které je J_c generována.

Invariance vůči transformaci

Věta 3.6.5 (Fatou-Julia). *Pro libovolný polynom f stupně ≥ 2 platí, že odpuzující periodické body f jsou husté¹⁴ v $J_c(f)$.*

Důkaz. Viz [15, strana 16]. □

Důsledek 3.6.1. *Je-li f polynom, pak Juliova množina je kompletně invariantní vůči f , neboli*

$$f(J_c(f)) = J_c(f),$$

a také

$$f^{-1}(J_c(f)) = J_c(f).$$

Důkaz. Viz [15, strana 16]. □

Vlastnost, že Juliova množina je invariantní vůči tzv. *forward iteration* a také vůči tzv. *backward iteration* je velice důležitá, protože díky tomuto faktu můžeme Juliovu množinu generovat pomocí Inverse iteration method, o kterém bude řeč dále.

¹⁴viz věta 1.1.1

Aplikujeme-li jednu z inverzních transformací $z \mapsto \pm\sqrt{z-c}$ na libovolný bod z z Juliovu množiny, tak opět získáme bod z Juliovu množiny. Totiž pokud obecně platí, že $z_n \in J_c$, pak také $z_{n-1} \in J_c$. Protože není možné, aby se hodnota orbity dostala mimo Juliovu množinu a pak se tam vrátila. Tedy řekneme, že Juliovu množinu je invariantní vůči zpětné iteraci.

Pro forward iteration uvažujeme tak, že pokud vybereme bod z Juliovu množiny a aplikujeme na něj transformaci $z \mapsto z^2 + c$, dostáváme opět bod z Juliovu množiny, jak už bylo rozebráno výše v sekci o orbitě. Bod po iteraci nemůže být ani z E_c , protože by divergoval do nekonečna a zároveň nemůže být ani z P_c , protože by konvergoval k přitahujícímu pevnému bodu a netvořil tak tvar J_c .

Vše dohromady tedy dává, že Juliovu množinu není invariantní pouze vůči inverzní transformaci $z \mapsto \pm\sqrt{z-c}$, ale také vůči $z \mapsto z^2 + c$. Invarianci J_c tak můžeme říkat *kompletní invariance*.

Juliovu množinu jako repelor

V této pasáži si uvedeme důležitá tvrzení, z kterých budeme následně vycházet při generování Juliovu množiny. Pro jednodušší pochopení následující věty však nejprve zadefinujme pojmem uzávěr množiny.

Definice 3.13. [1] Nechť $G \subseteq \mathbb{C}$. Množina $\bar{G} = \{x \in \mathbb{C} \mid d(x, G) = 0\}$ se nazývá *uzávěr množiny* G . Množina G se nazývá *uzavřená*, pokud $G = \bar{G}$.

Uzávěrem množiny je tedy méně formálně myšleno sjednocení hranice množiny a všech bodů množiny. Uzávěr lze také definovat jako nejmenší možnou uzavřenou množinu, která danou množinu obsahuje.

Věta 3.6.6. *Juliovu množinu je uzávěrem množiny všech periodických bodů $f(z) = z^2 + c$.*

Důkaz. Viz [22, strana 17]. □

Pokud tuto větu spojíme s faktem, že odpuzujících periodických orbit je konečný počet, což vychází ze známé Fatou-Shishikura nerovnosti [22], můžeme pak uvést další možnou definici Juliovu množiny.

Věta 3.6.7. *Juliovu množinu je uzávěrem množiny všech odpuzujících periodických bodů f .*

Důkaz. Viz [4, strana 221]. □

S pomocí faktu, že J_c je invariantní vůči iteraci, můžeme ukázat, že J_c je repelor pro body mimo Juliovu množinu. Totiž víme, že orbita s počáteční hodnotou v Juliově množině zůstává v množině uvězněna. Naopak počáteční hodnoty mimo J_c pro které platí $z_0 \in E_c$ divergují k nekonečnu a pro $z_0 \in P_c$ konvergují k přitahujícímu pevnému bodu. V každém případě se však vždy oddalují od J_c . Tedy Juliovu množinu funkce f působí odpudivě na orbity s počáteční hodnotou $z_0 \notin J_c$.

Významná vlastnost pro generování Juliových množin v počítačové grafice je, že jestliže J_c kde $z_0 \notin J_c$ je pro f repelor, pak pro inverzní iterační proces f^{-1} bude J_c atraktor [7, strana 764], tedy, že orbity okolních počátečních hodnot budou přitahovány k J_c .

Věta 3.6.8. Pokud je z libovolný bod z Juliový množiny funkce f , potom vzory z jsou husté v J_c .

Důkaz. Viz [23, strana 13]. □

Toto tvrzení je jedním ze základních stavebních kamenů pro funkčnost Inverse iteration method.

Kapitola 4

Souvislosti mezi Mandelbrotovou a Juliovou množinou

Jak již název napovídá, záměrem kapitoly je uvést souvislosti mezi Mandelbrotovou a Juliovou množinou. Na začátek je uvedeno pár základních a důležitých vlastností Mandelbrotovy množiny a po zbytek kapitoly je obsah zaměřen již výhradně na souvislosti a vztahy mezi množinami. Čerpáno je převážně z [7], [15] a [24]. Stejně jako u všech předcházejících kapitol jsou i všechny ostatní zdroje uvedeny přímo u daného faktu.

4.1 Historie

Benoit B. Mandelbrot byl francouzsko-americký matematik a ekonom s polskými kořeny. Narodil se ve Varšavě roku 1924, avšak již jako dítě se s rodinou přestěhoval do Francie. Po druhé světové válce využil svých vynikajících matematických znalostí a po absolvování přijímacích zkoušek s nejlepšími výsledky nastoupil na univerzitu École polytechnique¹. Po studiích v roce 1950 dostal nabídku pracovat ve Philips Electronics, kde působil ve vědeckém týmu zabývajícím se barevnými televizory.

Nejúspěšnější léta přišla s rokem 1958, kdy se Mandelbrot s rodinou přestěhoval do Ameriky, odkud přišla nabídka práce z IBM, kde nakonec zůstal až do roku 1993. Během tohoto působení přispěl svými vědomostmi k rozvoji mnohých vědních disciplín jako jsou ekonomie, lingvistika, či kosmologie.

Převážně je však spojován s pojmem fraktální geometrie. Ve své knize *Les Objets Fractals: Forme, Hasard et Dimension* v roce 1975 popsal a jako první definoval fraktální objekty a zasloužil se tak o velký rozmach v tomto velmi mladém odvětví matematiky. Následně se začal zabývat studií Pierre Fatoua a Gaston Julii a v roce 1979 díky přístupu k počítačům v IBM vykreslil Juliovu množinu a Mandelbrotovu množinu, která po něm nese jméno za velké zásluhy v rozvoji fraktální geometrie, i když byla objevena právě francouzským matematikem Pierre Fatouem, již dříve. Líbivou vizualizací fraktálních obrazců dokázal přitáhnou velkou pozornost nejen matematické komunity, což Fatou a Julia bez počítačové grafiky, více jak sedesát let dříve, nedokázali [25].

¹Zmiňována také v sekci 3.1.

4.2 Mandelbrotova množina

V následující pasáži nebudeme odlišovat J_c a K_c , a tak můžeme předpokládat, že zmíněná fakta jsou pro J_c i K_c ekvivalentní.

Již jako první souvislost můžeme uvést, že Mandelbrotova množina, značíme ji jako M , je generována pomocí stejného iteračního procesu jako Juliová množina, tedy

$$z \mapsto z^2 + c \quad z, c \in \mathbb{C}.$$

Zde však nastává rozdíl. Jak již víme, pro Juliovu množinu fixujeme konstantu c a postupně testujeme všechny počáteční hodnoty, zda-li jejich orbita přesáhne limitní poloměr divergence. U Mandelbrotovy množiny nás však bude zajímat pouze jedno z a tím je $z = 0$, což odpovídá kritickému bodu $z^2 + c$. Naopak tedy bude zafixované $z_0 = 0$ a iterační proces se bude provádět postupně pro všechna c . Pokud je Juliová množina určitá část komplexní roviny počátečních hodnot z_0 , pak o Mandelbrotově množině můžeme tvrdit, že je to určitá část komplexní roviny konstant c . Prvních pár iterací M vypadá následovně:

$$0 \mapsto c \mapsto c^2 + c \mapsto (c^2 + c)^2 + c \dots$$

Můžeme si všimnout, že předpis tvoří kritickou orbitu pro různá c , protože pro libovolné c je pokaždé $z_0 = 0$. Uveďme si nyní první možnou formální definici Mandelbrotovy množiny.

Definice 4.1. [7] Mandelbrotova množina M je množina všech parametrů c , pro které platí

$$M = \left\{ c \in \mathbb{C} \mid \{f_c^n(0)\}_{n=0}^{\infty} \not\rightarrow \infty \right\}.$$

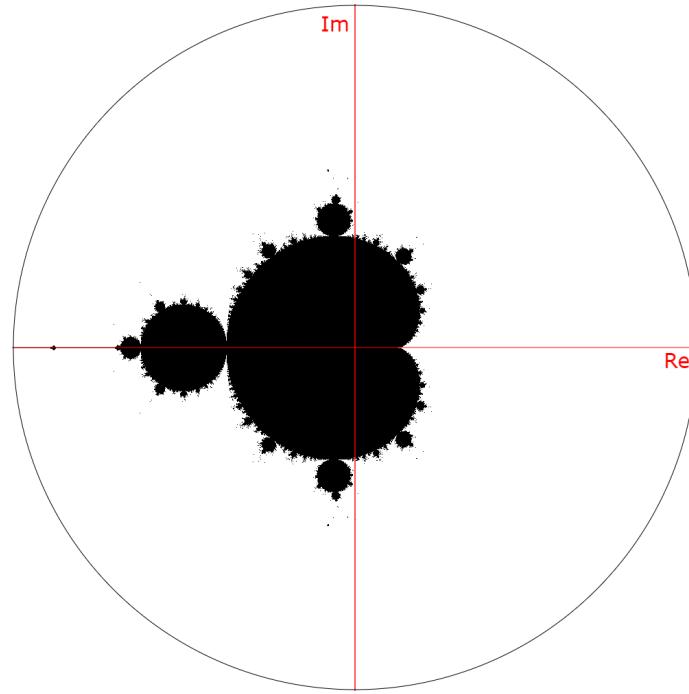
S mírnou modifikací tvrzení 3.3.1, kde pro M stačí uvažovat pouze $r(c) = 2$ platí, že opustí-li kritická orbita c kruh o poloměru 2 se středem v počátku, pak si můžeme být jisti, že diverguje, z čehož plyne, že neexistuje c mimo tento kruh, které by náleželo M [15], což umocňuje i věta 4.2.1 o souvislosti M .

Věta 4.2.1 (Douady-Hubbard). *Mandelbrotova množina je souvislá.*²

Důkaz. Viz [9, strana 75]. □

Navíc, jak demonstruje obrázek 4.1, lze vidět, že jediné c , kde $|c| = 2$ je $c = -2$.

²Předpokládá se navíc i jednoduchá souvislost, to však ještě není dokázáno.



Obrázek 4.1: Pro $c \in M$ jsou hodnoty obarveny černě, pro $c \notin M$ bíle. Celé M je potom ohrazeno kružnicí se středem v počátku o poloměru $r(c) = 2$.

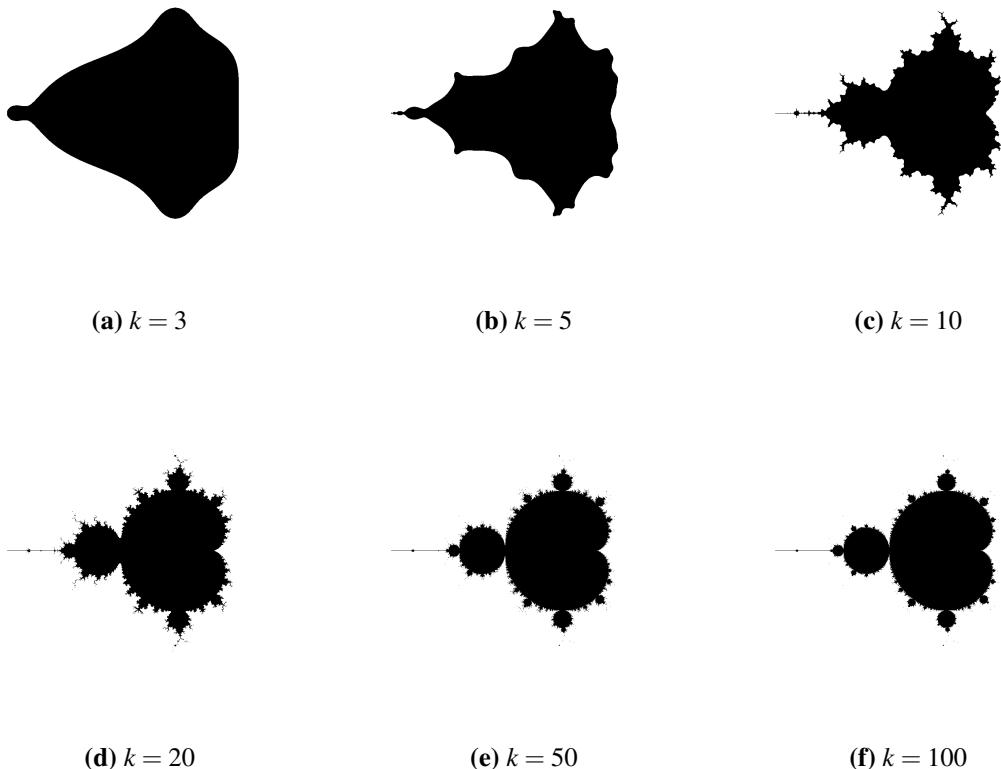
4.2.1 Aproximace Mandelbrotovy množiny

Všimněme si, že definice M je velice podobná definici P_c . Obě množiny obsahují body, jejichž orbita nediverguje. Tedy zřejmě, budeme-li chtít approximovat tvar Mandelbrotovy množiny, využijeme stejného postupu jako u approximace P_c , díky němuž budeme schopni vykreslit vícebarevný a propracovanější obrazec, než ten na obrázku 4.1.

Nechť tedy opět approximace má stejný tvar jako v předešlé kapitole, jen s tím rozdílem, pro co approximaci počítáme

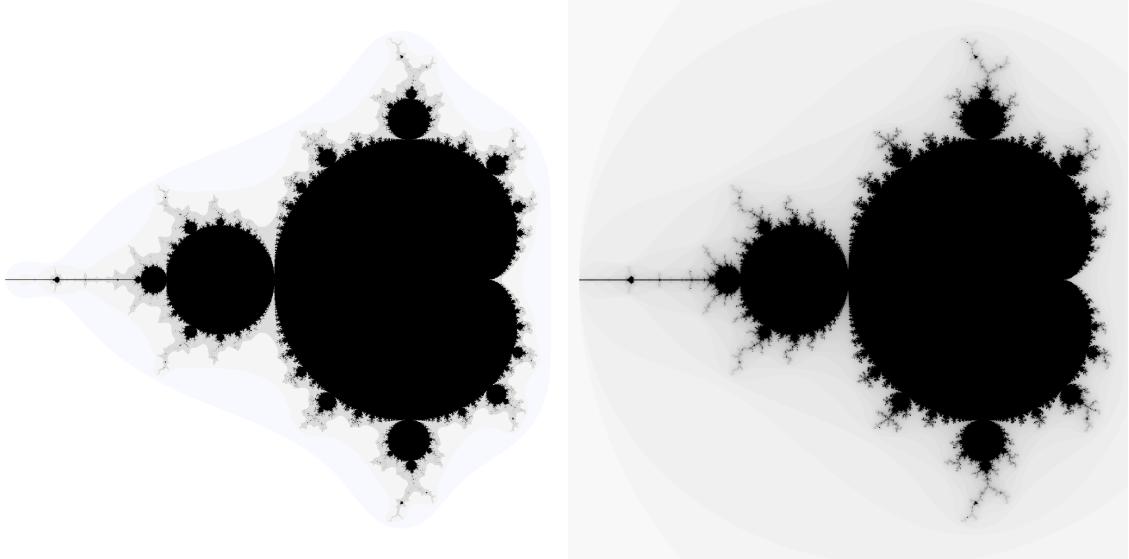
$$A_k(c) = \{c \mid |z_k| \leq 2\}.$$

Zřejmě $A_0(c)$ bude tedy kruh o poloměru 2. Na obrázku 4.2 vidíme pár dalších vybraných approximací pro odhad M .



Obrázek 4.2: Aproximace Mandelbrotovy množiny.

I zde, stejně jako u odhadu prisoner set, si lze všimnou, že čím více iterací provedeme, tím menší změny ve výsledném tvaru dostaváme. Nyní pokud každé vykreslené approximaci přiřadíme určitý odstín šedi a překryjeme je přes sebe shora sestupně, podle počtu provedených approximací, získáme již vizuálně zajímavější obrazec, viz 4.3a. Přechody mezi vybranými approximacemi jsou ale při tak malém počtu velmi daleko od sebe a nejsou příliš plynulé. Využitím některé z vizualizačních metod ze sekce 5.3.1 však přiřadíme každé approximaci pro $k = 0, 1, 2 \dots$ určitý odstín a zároveň dosáhneme mnohem plynulejšího přechodu mezi approximacemi, jak můžeme vidět na obrázku 4.3b.



(a) Překrytí vybraných aproximací $A_3(c)$, $A_5(c)$, $A_{10}(c)$, $A_{20}(c)$, $A_{50}(c)$, $A_{100}(c)$. (b) Vizualizační metoda Escape Time Algorithm.

Obrázek 4.3: Rozdíl v plynulosti přechodů barev.

4.2.2 Symetrie

Věta 4.2.2. *Mandelbrotova množina je symetrická podle reálné osy, tedy*

$$c \in M \iff \bar{c} \in M.$$

Důkaz. Viz [26]. □

Komplexní číslo c má svou reálnou část, označme ji c_x a imaginární část c_y a tedy lze rozepsat jako $c = c_x + c_y i$. Podobně pak pro komplexně sdružené číslo \bar{c} k c jako $\bar{c} = c_x - c_y i$, viz sekce 2.1. Navíc z obrázku 2.2 ve stejné sekci je zřejmé, že c a \bar{c} jsou symetrické podle reálné osy. Abychom ukázali, že M je osově souměrná, potřebujeme zajistit, aby všechny hodnoty v orbitě pro libovolné c byly postupně komplexně sdružené s hodnotami orbity pro \bar{c} . Mějme tedy $c = x + yi$ a $\bar{c} = x - yi$. Proveďme první dvě iterace pomocí $z_{n+1} = z_n^2 + c$, kde $z_0 = 0$, jak plyne z definice M .

Pro c :

$$\begin{aligned} z_0 &= 0 \\ z_1 &= x + yi \\ z_2 &= (x + yi)^2 + (x + yi) = x^2 + 2xyi - y^2 + x + yi = \underbrace{x^2 + x - y^2}_{c_x} + \underbrace{2xyi + yi}_{c_y}. \end{aligned}$$

Pro \bar{c} :

$$z_0 = 0$$

$$z_1 = x - yi$$

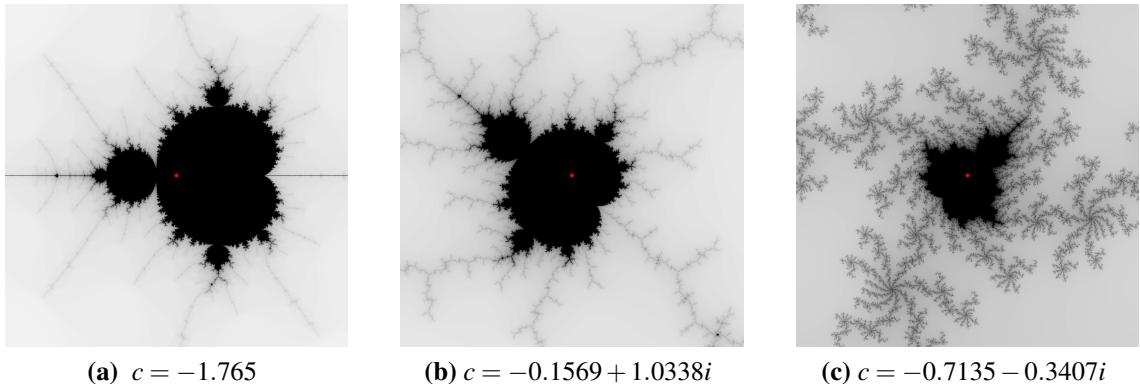
$$z_2 = (x - yi)^2 + (x - yi) = x^2 - 2xyi - y^2 + x - yi = \underbrace{x^2 + x - y^2}_{c_x} - \underbrace{(2xyi + yi)}_{c_y}.$$

Vidíme, že i po dvou iteracích jsou hodnoty orbit pro libovolné c a k němu \bar{c} k sobě komplexně sdružené. Z toho tedy plyne, že podmínce o divergenci budou splňovat vždy c i \bar{c} stejně, viz (2.4) a to znamená, že Mandelbrotova množina je opravdu symetrická podle Re osy jak ilustruje obrázek 4.1.

4.2.3 Soběpodobnost

Pro úplnost ještě doplňme vlastnost soběpodobnosti, která stejně jako u Juliových množin je tzv. quazi-soběpodobnost (soběpríbuznost). Navíc z [7, strana 733] víme, že Mandelbrotova množina obsahuje nekonečně mnoho menších kopií sebe sama. Pár příkladů ukazuje obrázek 4.4.

U Mandelbrotovy množiny se však častěji potkáváme s asymptotickou soběpodobností o které bude řeč v sekci 4.4. Má totiž úzkou spojitost s Juliovými množinami.



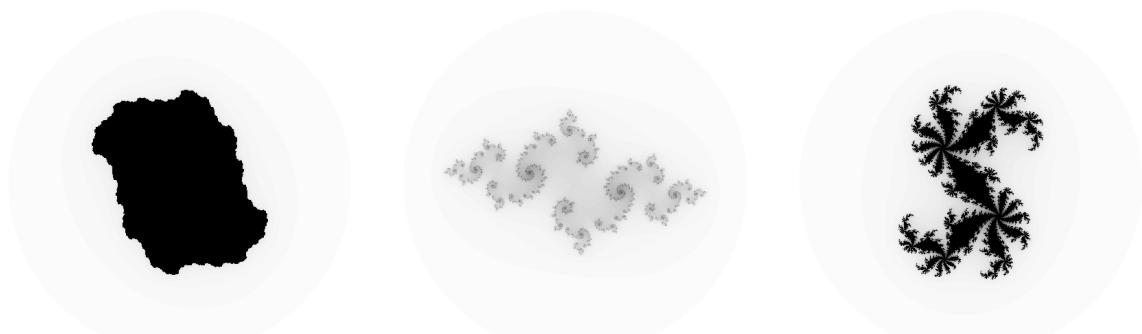
Obrázek 4.4: Kopie M , pro různá přiblížení, schované ve fraktální struktuře originální M . Přesná pozice c je znázorněna červeným bodem.

4.3 Mapa Juliových množin

V návaznosti na základní dualitu Juliových množin můžeme považovat Mandelbrotovu množinu za oblast, která určuje souvislost (filled) Juliovu množiny. Nastávají dvě varianty:

- $c \in M$, pak je J jednoduše souvislá,
- $c \notin M$, pak je J nesouvislá.

Tento fakt plyne z vlastnosti kritické orbity, pomocí které jsme definovali Mandelbrotovu množinu. Pár příkladů demonstруje obrázek 4.5.



(a) Uvnitř pro $c = 0.13 + 0.3i$. (b) Vně pro $c = -0.8 + 0.23i$. (c) Blízko hranice pro $c = 0.38 - 0.14i$.

Obrázek 4.5: Vizualizace K_c pro různé pozice c v M .

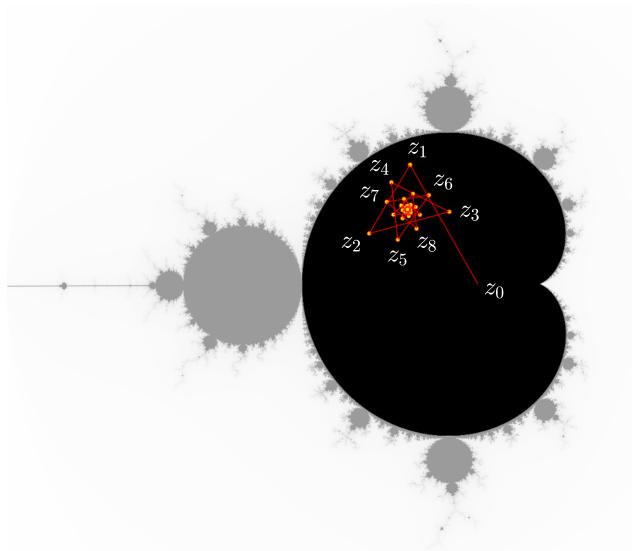
4.3.1 Geometrie Mandelbrotovy množiny

Mandelbrotova množina neurčuje pouze souvislost (filled) Juliovu množinu, ale také další zajímavou vlastnost a to konkrétně periodu její orbity. Pro jednodušší pochopení jen zdůrazněme, že kritická orbita pro iterační předpis M i J_c má vždy počáteční hodnotu $z_0 = 0$ a tedy potom $z_1 = 0^2 + c = c$.

Kardioida

Při pohledu na Mandelbrotovu množinu si jako první všimneme největší části, která svým tvarem připomíná srdce a podle něj také nese název *Kardioida*. Pro zajímavost pojďme ještě nazvat, že tento tvar lze sestrojit pomocí stopy pevně zvoleného bodu na kružnici s daným poloměrem, která se valí bez prokluzu okolo druhé kružnice o stejném poloměru. Tedy konkrétně u M je kardioida tvořena pomocí pevné kružnice o poloměru 0.25 se středem v počátku a rotující kružnice o stejném poloměru. Valení začíná v bodě $0.25 + 0i$, kdy nastává jediná situace, kde fixně zvolený bod je společným bodem obou kružnic, viz [27].

Kardioida splňuje jeden zásadní fakt. Vybereme-li jakékoli c z této oblasti, pak jeden z pevných bodů c je přitahující pevný bod. To tedy znamená, že kritická orbita je ohrazená a pro libovolné c z kardiody konverguje přímo k přitahujícímu pevnému bodu, jak můžeme vidět na obrázku 4.6. Navíc, každá orbita pro libovolné $z_0 \in P_c$ a pro odpovídající c z kardiody konverguje k přitahujícímu pevnému bodu [7].



Obrázek 4.6: Prvních osm hodnot kritické orbity konvergující k pevnému bodu.

Kruhy

Ke kardioide je po obvodu připojeno nekonečně mnoho kruhů, v anglické literatuře často známé pod názvem *bulbs*, na kterých jsou opět další kruhy, což takto funguje až do nekonečna díky fraktálním vlastnostem Mandelbrotovy množiny.

Protože c vybrané z libovolného kruhu nenáleží kardioide, pak ani jeden pevný bod pro c není přitahující. Kritická orbita pak pro jeden pevný bod diverguje k nekonečnu, ale pro druhý pevný bod je cyklická s periodou, která se liší v závislosti na tom, v jakém kruhu se nachází konstanta c .

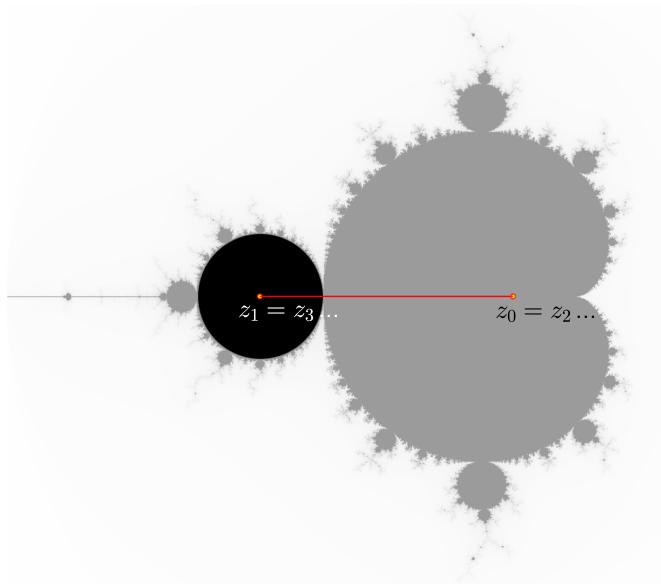
Kruh s periodou dva

Vlevo od kardioidy se nachází největší ze všech kruhů. Tento kruh má poloměr 0.25 a jeho střed se nachází v $-1 + 0i$. S kardioidou je spojen bodem v $-0.75 + 0i$. Poloměr tohoto kruhu je identický s poloměrem kružnic pomocí kterých jsme vytvořili kardioidu.

Důležitou vlastností je, že zvolíme-li c libovolně z tohoto kruhu, pak kritická orbita zůstává uvězněna v cyklu délky s periodou dva. Pravdivost tvrzení pro $c = -1$ dokazuje obrázek 4.7. Z tohoto důvodu pak pro rozlišení označujeme kruh jako kruh s periodou dva.

Kruh s periodou tři

Obdobně se bude chovat kritická orbita pro c zvolené z kruhu nacházejícího se na vrcholu kardioidy. Pro tento kruh je typické, že perioda kritické orbity je tři. Opět si ukažme jeden ilustrační příklad na obrázku 4.8 pro $c = -0.15 + 0.75i$.



Obrázek 4.7: Kritická orbita s periodou dva pro c z kruhu s periodou dva.

Podobně zjistíme periodu pro kritickou orbitu u všech ostatních kruhů.

Věta 4.3.1. [15] Pro libovolné c uvnitř kruhu s periodou k platí, že jeho kritická orbita je přitahující s periodou k .

Jelikož jsou M i K_c generovány pomocí (3.1), pak je zřejmé, že kritická orbita pro c je u M a K_c totožná, protože je vždy generována pro kritický bod $z_0 = 0$. Navíc víme, že P_c je oblastí přitažení pevného bodu pro určité c . Pak všechny orbity s počáteční hodnotou $z_0 \in P_c$ konvergují k přitažlivé orbitě s určitou periodou. Pro $c = -1$ na obrázku 4.7 vidíme, že orbita je právě výše zmíněná přitažlivá orbita s periodou dva. Zvolíme-li za $z_0 = -0.8$, tedy jiný bod z kruhu s periodou dva, orbita se pro takto zvolené z_0 po pár iteracích opět ustálí v cyklu mezi hodnotami -1 a 0 . Stejně tak na příkladu pro kruh s periodou tři na obrázku 4.8 vidíme, že orbita se ustálí na hodnotách odpovídající přitažující orbitě s periodou tří.

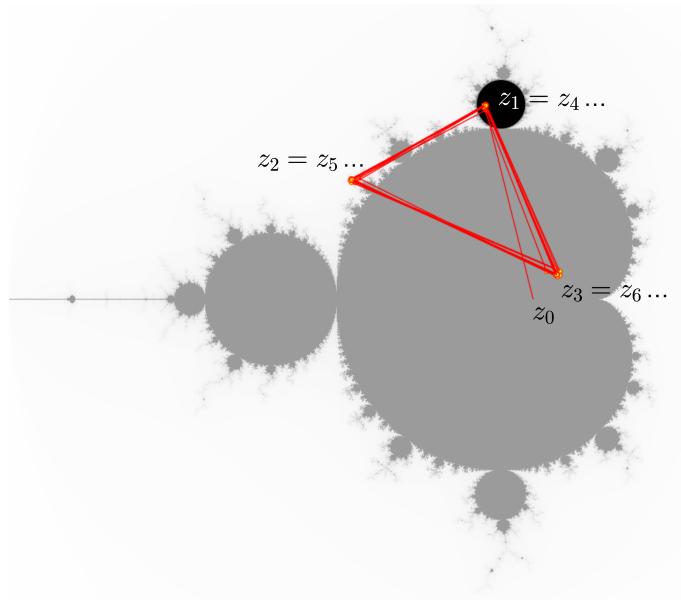
Získáváme velmi významnou spojitost mezi Mandelbrotovou množinou a libovolnou (filled) Juliovou množinou, kde obecně výběrem konstanty c z kruhu s periodou n z M zaručíme, že pro všechna $z_0 \in P_c$ bude orbita uvězněna v K_c s periodou n . Tato část vyšvětuje také větu 3.6.7 z předchozí kapitoly.

Jak je vidět, zjišťovat periodu každého kruhu tak, že zvolíme náhodné c , provedeme iterační proces a podíváme se, po kolika hodnotách se výsledky začnou opakovat je velmi časově náročné. Existuje však vlastnost, která celý proces hledání periody mnohonásobně urychlí.

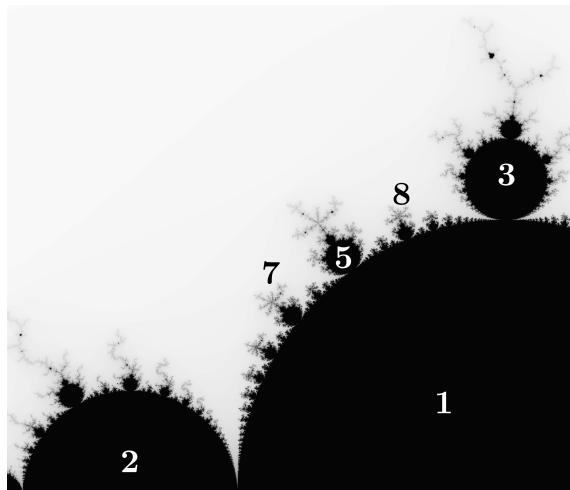
Věta 4.3.2. ³ Jestliže jsou a a b po sobě jdoucí periody kruhů na kardiodě Mandelbrotovy množiny, pak největší kruh nacházející se mezi nimi má periodu $a + b$.

Tento fakt je naprostě dostačující. Známe totiž periody prvních dvou největších kruhů, které si jen těžko splátejme. Potom již není problém zbylé periody dopočítat. Názorná ukázka zjištění periody lze vidět na obrázku 4.9.

³Vychází přímo z věty 4.3.3 níže.



Obrázek 4.8: Při volbě c z kruhu s periodou tří se kritická orbita po páru hodnotách ustálí v cyklu s periodou tří.

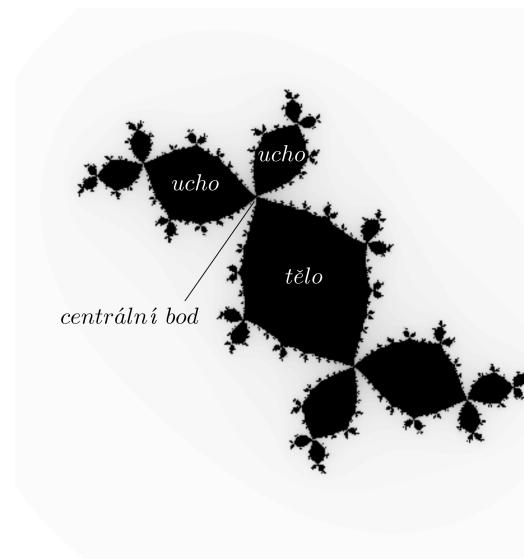


Obrázek 4.9: Princip nalezení periody pro páry kruhů na kardioidě.

4.3.2 Rotační čísla

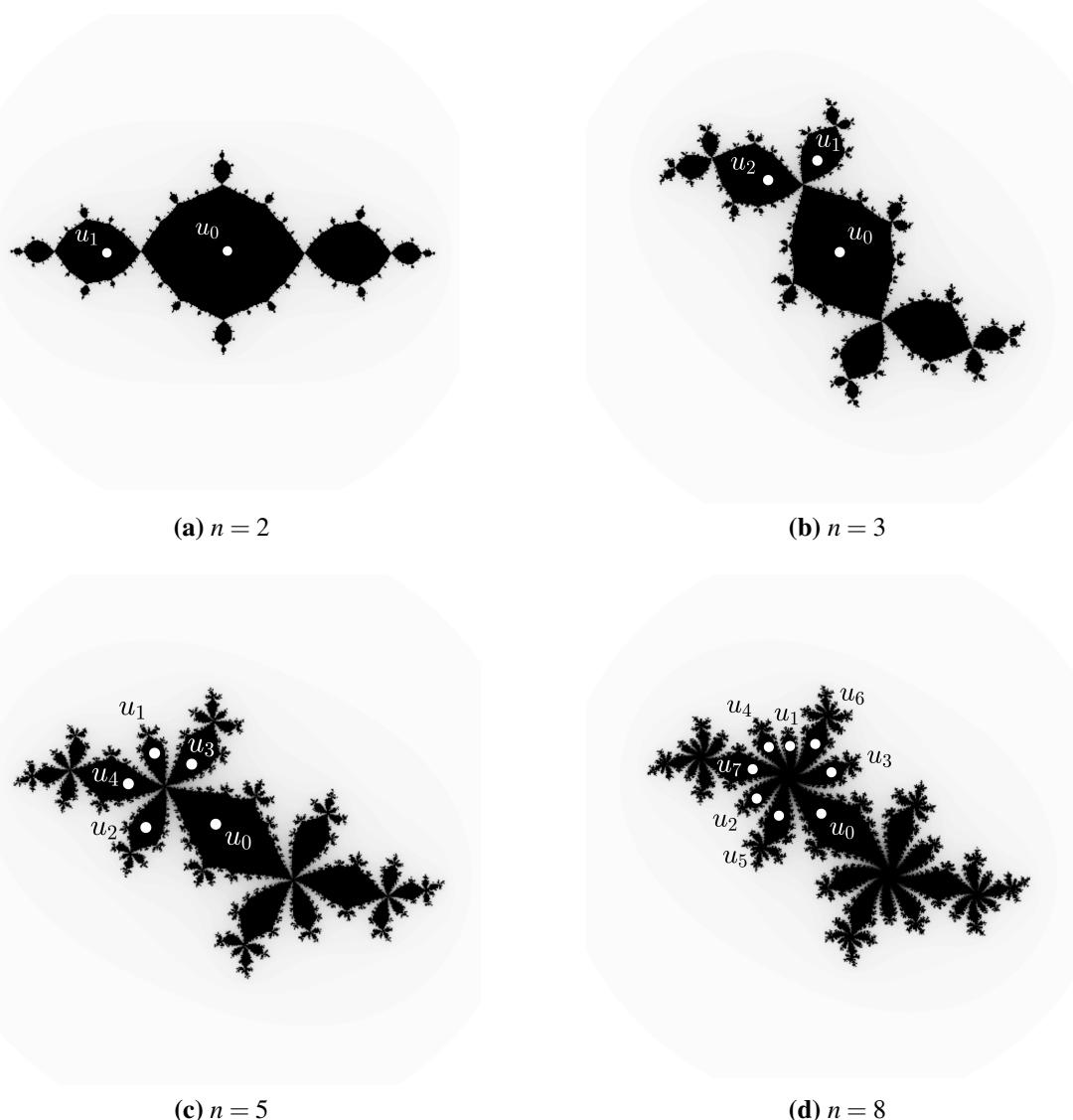
Rozeberme podrobněji tvar K_c pro c nacházející se v kruzích s určitou periodou, viz 4.11. Na první pohled je patrná určitá podobnost K_c pro různé periody n , hlavně pak pro periody $n > 2$. Vidíme, že každá K_c má ve středu největší část, nazveme ji *tělo*, z kterého pak vyrůstají menší napodobeniny na každé straně těla přesně v počtu $n - 1$ v závislosti na periodě [15]. Nejčastěji tyto části nazýváme *uši*. Na nich opět můžeme spatřit menší podobné útvary, které nás již však nebudou příliš zajímat.

Tělo a uši jsou spojeny společným bodem zvaném *centrální bod*. Popis jednotlivých částí demonstruje obrázek 4.10.



Obrázek 4.10: Popis jednotlivých částí K_c s periodou tří.

Samotná orbita se pak po ustálení točí kolem centrálního bodu a postupně navštěvuje tělo a všechny uši a následně tento proces opakuje v přesně daném pořadí. Pro zjištění chování orbity existuje několik metod, avšak na konci produkuje všechny stejnou výstupní hodnotu a tou je *rotační číslo*. Zapisujeme jej jako zlomek $\frac{k}{n}$, kde n značí periodu orbity a k v jednoduchosti říká, které ucho orbita navštíví jako první ve směru proti hodinovým ručičkám. Druhá možná interpretace k je, že udává velikost kroku, podle kterého orbita navštěvuje jednotlivé uši. Například rotační číslo, pro orbitu z obrázku 4.11b, je rovno $\frac{1}{3}$. Orbita jako první navštíví první ucho z prava a následně krokem o velikosti 1 navštíví i zbylé uši. Dodejme, že kardioida má rotační číslo $\frac{0}{1}$, jelikož příslušná K_c nemá žádné ucho a perioda přitahujícího pevného bodu je jedna.



Obrázek 4.11: K_c , kde c je voleno z kruhů s periodou n .

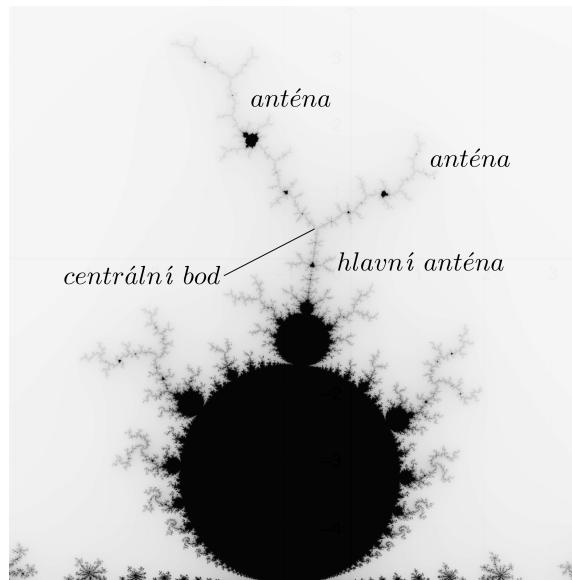
U první metody, jak určit rotační číslo a chování orbit, stačí najít nejmenší ucho a zjistit jeho pořadí proti směru hodinových ručiček.

Vezměme jako příklad K_c z obrázku 4.11c. Nejmenší je druhé ucho zprava. Tedy $k = 2$, což nám říká, že orbita z těla navštíví nejprve druhé ucho, dále čtvrté, pak krok vychází na šesté ucho, což odpovídá prvnímu uchu, dále ucho číslo tří a jako poslední číslo pět, což odpovídá tělu množiny a proces se opakuje. Označíme-li tělo u_0 a jednotlivé uši jako u_1, u_2, \dots , pak

$$u_0 \mapsto u_2 \mapsto u_4 \mapsto u_1 \mapsto u_3 \mapsto u_5 \mapsto u_7 \dots$$

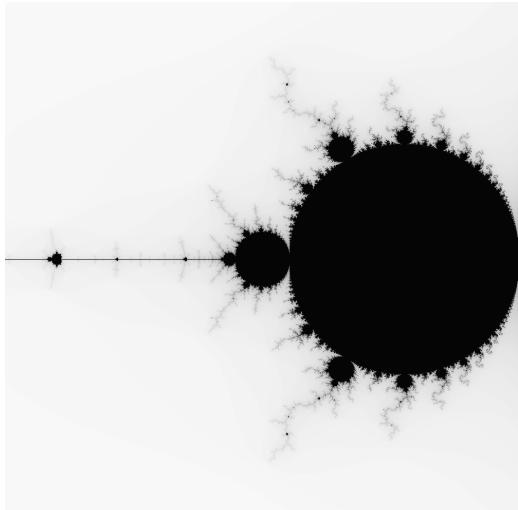
Tedy rotační číslo konkrétně pro tuto filled Juliovu množinu je $\frac{2}{5}$. Problém však nastává pro množiny s větší periodou. Jak se můžeme přesvědčit, že pro $n = 8$ je velmi těžko odhadnutelné, které ucho je nejmenší.

Představme si proto druhou metodu, která alespoň částečně tento problém řeší. Stejně jako pro K_c s různými periodami, platí podobnost i pro jednotlivé kruhy, jak se lze přesvědčit z obrázku 4.13. Namísto těla a uší však kruhy mají *antény*. Anténě, která je připojená přímo na kruh se říká *hlavní anténa*, nebo také vedoucí anténa a na ni jsou pak připojeny všechny ostatní antény, opět pomocí centrálního bodu, viz obrázek 4.12.

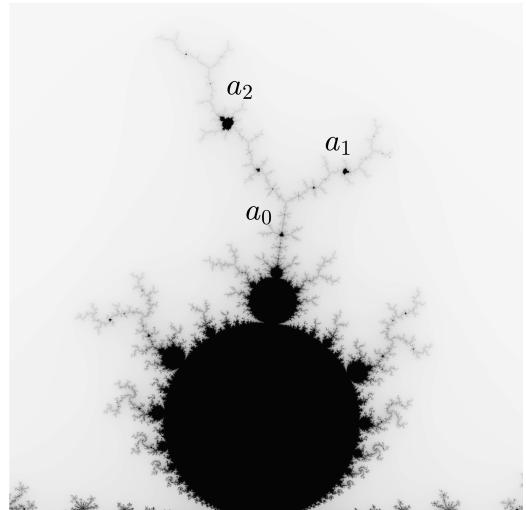


Obrázek 4.12: Popis jednotlivých částí pro kruh s periodou tří.

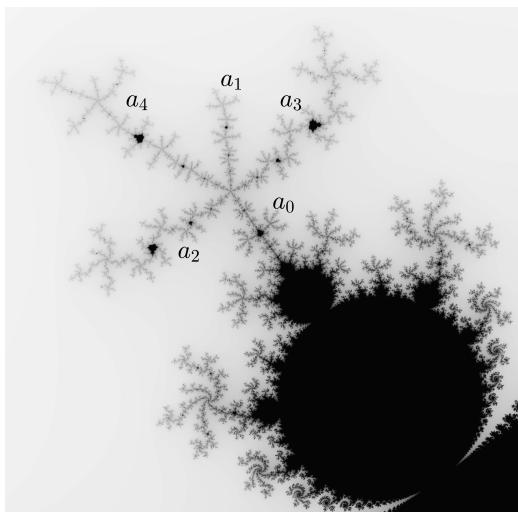
Chování orbity pak určíme zcela stejně jako u předešlé metody. Najdeme nejmenší anténu a pomocí ní určíme rotační číslo. Jediný problém nastává u kruhu s periodou dva, jelikož zde anténa a hlavní anténa splývá v jednu, viz obrázek 4.13a. Bohužel i zde narazíme na periody, kde již nejmenší anténa bude lidským okem nerozeznatelná od ostatních.



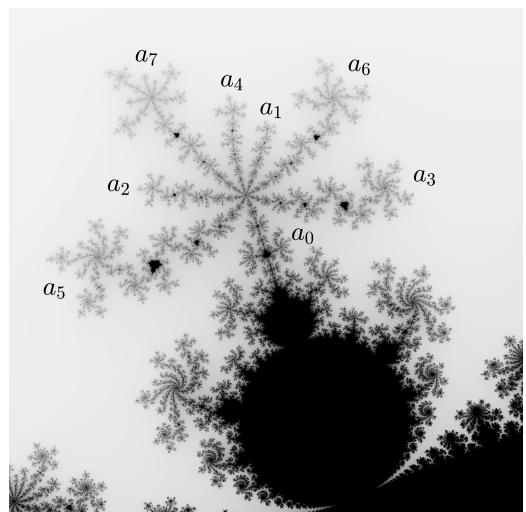
(a) $n = 2$



(b) $n = 3$



(c) $n = 5$



(d) $n = 8$

Obrázek 4.13: Kruhy s periodou n .

Fareyho sčítání

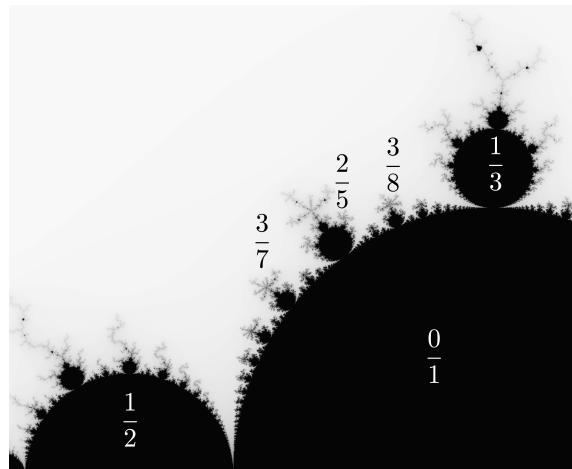
Poslední metoda využívá tzv. Fareyho sčítání. Zapomeneme-li na chvíli, jak se sčítají zlomky, Fareyho sčítání definuje součet dvou zlomků jako součet jmenovatele a čitatele

$$\frac{p}{q} + \frac{m}{n} = \frac{p+m}{q+n}.$$

Toto zvláštní sčítání určuje rotační číslo pro libovolný kruh Mandelbrotovy množiny a tedy i rotační číslo pro libovolnou (filled) Juliovu množinu.

Věta 4.3.3. [15] *Jestliže jsou $\frac{a}{b}$ a $\frac{c}{d}$ po sobě jdoucí rotační čísla kruhů na kardioide Mandelbrotovy množiny, pak největší kruh nacházející se mezi nimi má rotační číslo $\frac{a+c}{b+d}$.*

Metoda funguje stejně jako hledání periody, což demonstруje obrázek 4.14. Jsme tak schopni dopočítat libovolně velké rotační číslo, bez vizuálního hledání nejmenšího ucha, či antény.



Obrázek 4.14: Princip nalezení rotačního čísla pro pár kruhů na kardioidě.

4.4 Asymptotická soběpodobnost M a J_c okolo Misiurewiczova bodu

Pro jednodušší pochopení Misiurewiczova bodu si nejdříve zadefinujme *pre-periodický bod*.

Definice 4.2. [7] Bod $z_n \in \mathbb{C}$ se nazývá *pre-periodický* bod, jestliže

$$z_m = z_{m+p} \quad m > n, \quad p \geq 1.$$

To jsou všechna z_n , které nejsou součástí periodické části jejich orbit. Například

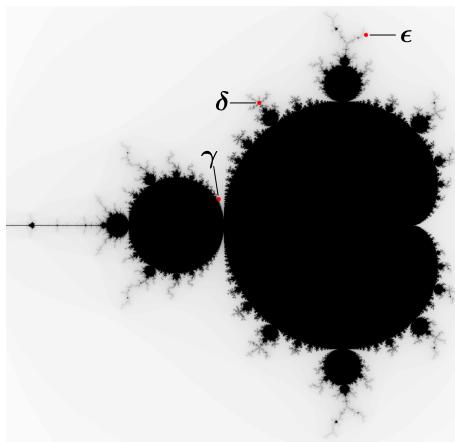
$$z_0 \mapsto z_1 \mapsto z_2 \mapsto z_3 \mapsto z_4 \mapsto z_5 \mapsto z_3 \mapsto z_4 \dots$$

vidíme, že od z_3 je orbita periodická, tedy z_0, z_1 a z_2 jsou pre-periodické body s periodou tři, protože platí $z_3 = z_{3+3}$. Pokud bychom připustili $n \leq m$, pak by se pro $n = m$ jednalo o periodickou orbitu již od počáteční hodnoty, tedy by i z_n byly periodické, ne však pre-periodické.

Nyní již nebrání nic v definici Misiurewiczova bodu.

Definice 4.3. [7] Parametr $c \in \mathbb{C}$ se nazývá *Misiurewiczův bod*, jestliže kritický bod $z_0 = 0$ funkce f je pre-periodický.

Na následujícím obrázku 4.15 jsou příklady Misiurewiczových bodů společně s pozicí na Mandelbrotově množině.



Obrázek 4.15: Příklady tří Misiurewitzových bodů: $\gamma = -0.77568377 + 0.13646737i$, $\delta = -0.562203 + 0.642818i$ a $\epsilon = i$.

Ověřme, že γ , δ i ϵ jsou Misiurewiczovými body. Tedy zkoumáme, zda-li je kritický bod z_0 pre-periodický postupně pro c rovno γ , δ a ϵ . Jako první vezměme γ .

$c = \gamma$:

$$\begin{aligned} z_0 &= 0 \\ z_1 &= -0.77568377 + 0.13646737i \\ z_2 &= -0.19262180203330403 - 0.07524367808716981i \\ z_3 &= -0.7442422224735282 + 0.16545451572952896i \\ &\vdots \\ z_{22} &= 1.139587642988388 - 0.08937093578476413i \\ z_{23} &= 0.5149890618887852 - 0.06722465812525186i \\ z_{24} &= -0.514989\dots + 0.06722\dots i \\ z_{25} &= -0.514989\dots + 0.06722\dots i \\ &\vdots \end{aligned}$$

Po 23. iteraci jsme se dostali do cyklu s periodou jedna. Dále ověříme δ .

$c = \delta$:

$$\begin{aligned} z_0 &= 0 \\ z_1 &= -0.562203 + 0.642818i \\ z_2 &= -0.659345767915 - 0.079970416108i \\ z_3 &= -0.13386142578506566 + 0.7482743108384228i \\ z_4 &= -1.104198562947506 + 0.4424878677456626i \\ z_5 &= 0.4612559533132343 - 0.3343709353729335i \\ z_6 &= -0.46124\cdots + 0.33437\cdots i \\ z_7 &= -0.46124\cdots + 0.33437\cdots i \\ &\vdots \end{aligned}$$

Pro $c = \delta$ se kritická orbita ustálila již po 5. iteraci opět v cyklu s periodou jedna. Jako poslední ověříme ε stejným způsobem, kde k ustálení dojde již po první iteraci do cyklu tentokrát s periodou dva.

$c = \varepsilon$:

$$\begin{aligned} z_0 &= 0 \\ z_1 &= i \\ z_2 &= -1 + i \\ z_3 &= -i \\ z_4 &= -1 + i \\ z_5 &= -i \\ &\vdots \end{aligned}$$

Tedy můžeme vzhledem k definici říci, že se jedná o Misiurewiczovy body. Protože z_0 pro všechny ověřované body je pre-periodický bod.

Z definice plyne, že kritická orbita je pro každý Misiurewiczův bod ohraničená a tedy, že všechny Misiurewiczovy body náleží Mandelbrotově množině. Další užitečná vlastnost je, že pokud c je Misiurewiczův bod, pak $J_c = P_c = K_c$, čili jak víme z pasáže 3.5.1, jedná se o K_c typu Dendrite.

Uvědomíme-li si, že pro všechna c uvnitř M platí $J_c \neq K_c$, tedy, že K_c má „vnitřek“ a na druhou stranu také víme, že všechny Misiurewiczovy body jsou v M , pak se mohou nacházet pouze na hranici M . Navíc platí, že Misiurewiczovy body jsou husté na hranici M , tedy, že pro každý bod c na hranici Mandelbrotovy množiny, platí, že kruh s jakkoliv malým kladným poloměrem se středem v c obsahuje nekonečně mnoho Misiurewiczových bodů [7]. Předešlý fakt posiluje věta 1.3.1, která říká, že dimenze hranice M je dva⁴. Důležité je podotknout, že topologická dimenze je jedna, protože hranice je pouze jednorozměrná křivka. To značí, že hranice M pokrývá plochu tak důkladně a hustě, že se přibližuje plošnému útvaru.

⁴Pro více informací doporučují [16, kapitola C8].

Asymptotická soběpodobnost

Při definování asymptotické soběpodobnosti v bodě vyjdeme z definice 1.9 o soběpodobnosti množiny v bodě.

Definice 4.4. [7] Nechť $G \subset \mathbb{C}$ je asymptoticky soběpodobná množina v bodě z_0 , jestliže platí

$$\lim_{n \rightarrow \infty} \rho^n (G - z_0) \cap D_r(0) = L \cap D_r(0),$$

kde $D_r(0)$ je kruh s malým kladným poloměrem r se středem v 0 a $G - z_0 = \{g - z_0 \mid g \in G\}$. Komplexní číslo ρ představuje faktor zvětšení a splňuje $|\rho| > 1$. L je limitní objekt⁵ G , který je soběpodobný v počátku.

Tedy jednoduše chceme-li zvětšit, či zmenšit původní objekt ρ -krát, pak platí $\rho \cdot G$. Operací $G - z_0$ dosáhneme posunutí G do počátku.

Čínská matematická Tan Lei se zabývala podobnostmi mezi Mandelbrotovou množinou a (filled) Juliovými množinami a dokázala velmi zajímavé vlastnosti.

Zvolíme-li konstantu c tak, že je Misiurewiczovým bodem, pak Juliová množina a Mandelbrotova množina budou asymptoticky soběpodobné v bodě $z = c$ za použití stejného ρ . Tedy pro Juliovu množinu:

$$\lim_{n \rightarrow \infty} \rho^n (J_c - c) \cap D_r = L_J \cap D_r.$$

Podobně pro Mandelbrotovu množinu, kde $0 < r' < r$:

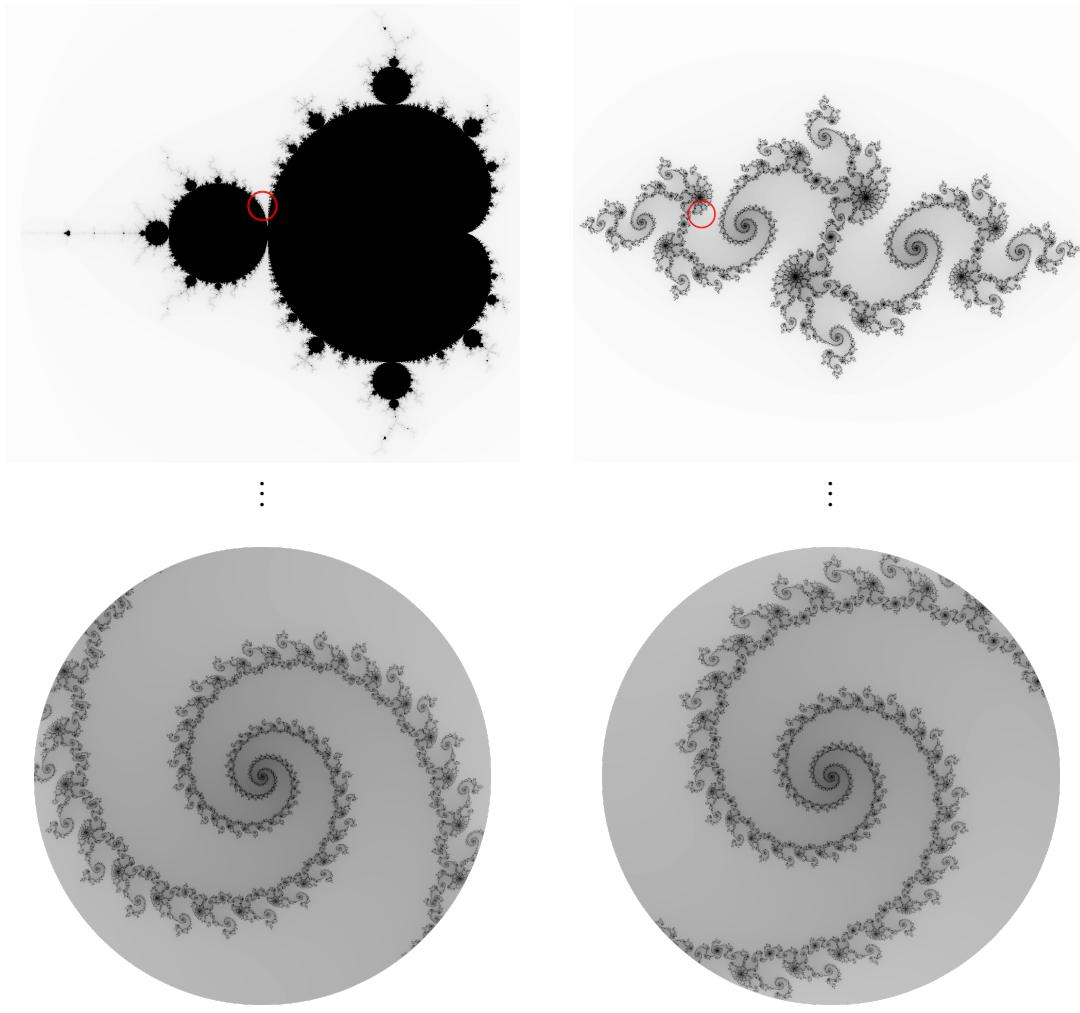
$$\lim_{n \rightarrow \infty} \rho^n (M - c) \cap D_{r'} = L_M \cap D_{r'},$$

kde L_J a L_M jsou limitní objekty pro Juliovu a Mandelbrotovu množinu korespondující s c jako Misiurewiczovým bodem. Navíc platí, že L_J a L_M jsou v podstatě stejné, pouhý rozdíl je v přiblížení a rotaci, tedy $L_M = aL_J$, kde a je vhodné komplexní číslo. Tuto podobnost demonstriuje vizualizace 4.16.

Jak již bylo zmíněno, Misiurewiczových bodů je na hranici M nekonečně mnoho, tedy existuje nekonečně mnoho L_J a L_M , které si jsou podobné a tato podobnost zesiluje při větším ρ .

Naskytá se však rozporuplná otázka. Abychom splnili definici soběpodobnosti rozebranou v části 4.2.3, pak L_M musí obsahovat menší kopie Mandelbrotovy množiny. Jenže L_J nic takového nedokáže, což jde proti vlastnosti asymptotické soběpodobnosti.

⁵ L_G je dostatečně přiblížená část G , aby splňovala soběpodobnost v počátku.



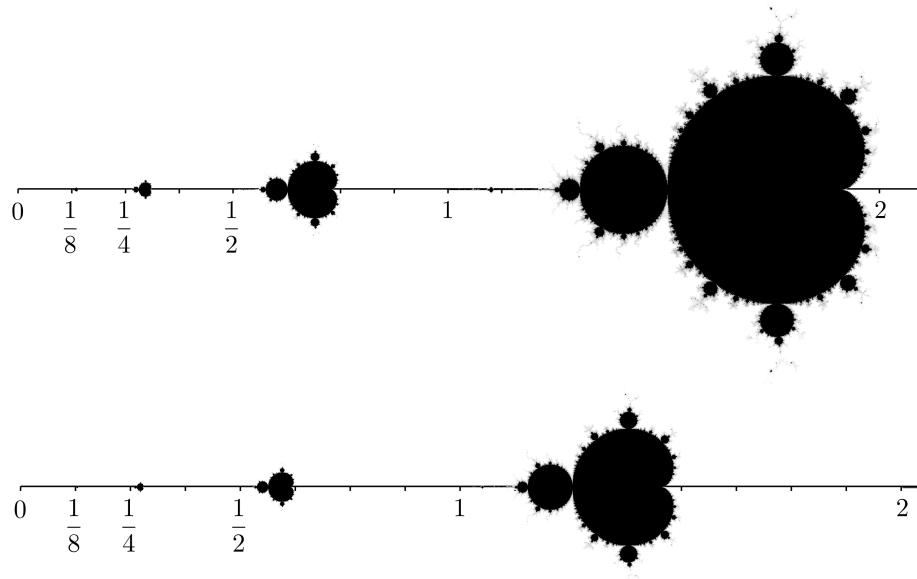
Obrázek 4.16: Mnohonásobné přiblížení okolí Misiurewitzova bodu $-0.77568377 + 0.13646737i$ vyznačeného červenou kružnicí. Ve spodní části tak vzniknou L_M s poloměrem okolí 0.000192 a L_J s poloměrem okolí 0.00094 a mírnou rotací.

Existuje však vlastnost, která tento problém řeší a my si ji vysvětlíme na jednoduchém příkladu, který však koresponduje přesně s tím, co se děje u kopií Mandelbrotových množin.

Představme si úsečku která začíná v počátku a končí v hodnotě jedna. Na její pravý konec připojíme menší kopii Mandelbrotovy množiny její nejvzdálenější hodnotou od počátku, negativním směrem reálné osy, která by u originální Mandelbrotovy množiny odpovídala $-2 + 0i$. Nalezneme polovinu úsečky a zde umístíme další menší Mandelbrotovu množinu a tento proces opakuje tak, že každá Mandelbrotova množina má čtyřikrát menší velikost, než ta předešlá.

Nyní podle definice asymptotické soběpodobnosti budeme přibližovat tento vzniklý objekt pomocí ρ vzhledem k bodu 0, protože víme, že limitní objekt tohoto útvaru je soběpodobný právě v počátku. Totiž Mandelbrotova množina nikdy nebude umístěna přímo v bodě 0, protože pokaždé bude existovat, ač velmi malá polovina, zbývající úsečky. Můžeme si však

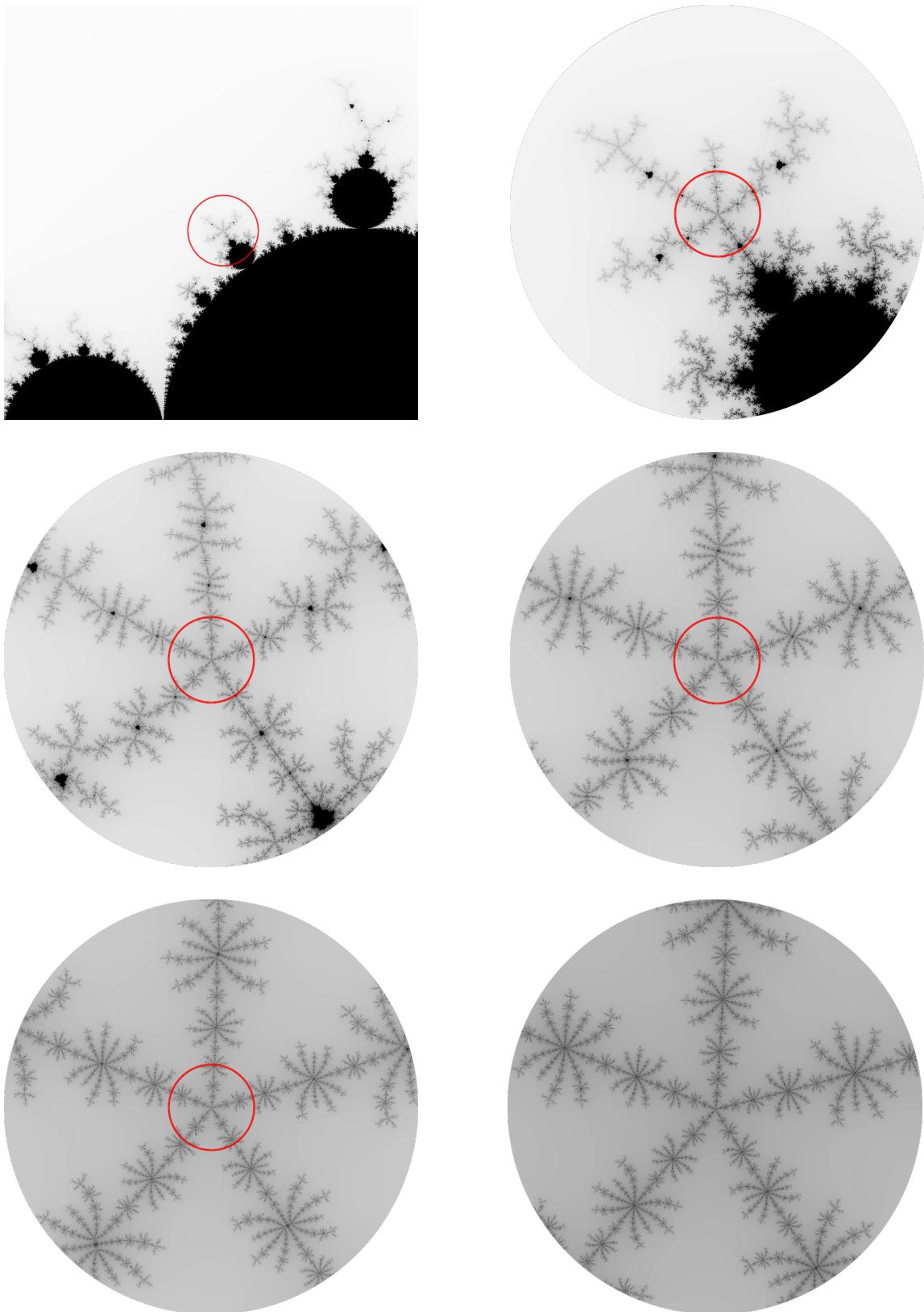
všimnout, že velikost kopií Mandelbrotových množin je zvolena tak sofistikovaně, že pokud námi vytvořený objekt budeme postupně přibližovat k bodu 0, pak Mandelbrotovy množiny zcela zmizí, protože je již nebude možné pomocí počítačové grafiky vykreslit. Velikost množin umístěných blíže k počátku se totiž snižuje rychleji, než jsme schopni postupným přibližováním vykompenzovat. Pro lepší porozumění se podívejme na obrázek 4.17, který ilustruje postupné přiblížení $\rho = 2$.



Obrázek 4.17: V horní části se nachází asymptoticky soběpodobný objekt v 0. Pod ním pak ten samý objekt, jen přiblížený ρ -krát pro $\rho = 2$. Mandelbrotovy množiny se zmenšují dvakrát rychleji, než je postupné přiblížování k nule. Po dostatečném počtu přiblížení tak všechny M zmizí.

Vidíme, že například Mandelbrotova množina umístěna v $\frac{1}{2}$ má velikost $\frac{1}{4}$. Přiblížíme-li se ale k počátku dvojnásobně, pak se vzdálenost i velikost relativně zvětší dvakrát, tedy ta samá M bude ve vzdálenosti 1 od počátku, avšak její velikost bude pouze $\frac{1}{2}$, což jak vidíme, je méně, než pro M , která měla před zvětšením ve vzdálenosti 1 od počátku velikost 1. Tedy z definice 4.4 plyne, že po n přiblížení pro $n \rightarrow \infty$ bude v libovolné relativní vzdálenosti od počátku Mandelbrotova množina s velikostí limitně blížící se 0.

Vraťme se zpět k námi sledované problematice. Přiblížujeme-li postupně libovolným ρ oblast Misiurewitzova bodu, pak velikost Mandelbrotových množin klesá přesně tak, že jsou na hranici neviditelnosti, jak se můžeme přesvědčit na obrázku 4.18. S tímto faktem opět přišla Tan Lei, která se mimo jiné také zabývala odhady velikosti těchto kopií okolo Misiurewitzova bodu.



Obrázek 4.18: Asymptotická soběpodobnost v okolí Misiurewitzova bodu $-0.562203 + 0.642818i$ vyznačeným červenou kružnicí. Postupným přibližováním pro $\rho = 5$ mizí kopie M . Pořadí jako při čtení textu.

Kapitola 5

Webová aplikace

Závěrečná kapitola nejprve popíše webovou aplikaci, naprogramovanou zejména za účelem vizualizace Juliových množin pomocí různých metod. Následně podrobněji rozebere jednotlivé vizualizační metody a popíše jejich silné stránky a nedostatky. V úvodní části opět nebude potřeba rozlišovat mezi J_c a K_c . To se změní až u popisu metody IIM v části 5.3.1, kde již bude nezbytné rozdíl dodržovat. Pro inspiraci u programování jednotlivých metod byly využity zdroje [7], [24], [28], [29] nebo [30]. Velkou roli zde také hrají jednotlivé vlastnosti rozebírané v předešlých kapitolách, na které bude často odkazováno.

5.1 HTML, CSS, JavaScript

HTML neboli *HyperText Markup Language* je označení pro značkovací jazyk, který tvoří pomocí značek (tagů) a jejich vlastností (atributů), základní strukturu webu.

CSS se stará o esteticky hezčí vzhled výsledné webové stránku. Jak už samotný název napovídá *Cascading Style Sheets*, pomocí kaskádovitého stylu zápisu kódu, formátuje rozvržení jednotlivých prvků webu. Konkrétně například font textu, velikost a umístění obrázků, nebo také možnost zobrazení stránky v mobilním zařízení.

Javascript je objektově orientovaný skriptovací jazyk, syntaxí velmi podobný C/C++¹, dodávající webu potřebnou dynamiku. Můžeme také říct, že JavaScript vdechne statickým prvkům stránky život. V případě zájmu doporučuji publikaci [32].

Ukažme si propojení této trojice na názorném příkladu. Mějme jednoduché tlačítko „OK“. Pozici na stránce a text tlačítka obstará HTML, tím ale jeho práce končí a nastupuje CSS, které dodá tlačítka barvu, naformátuje text, nebo například zaoblí rohy. Jako poslední přichází na řadu JavaScript, který definuje funkci tlačítka, například až se na něj klikne myší.

Výběr správného nástroje

Před programováním webové aplikace je potřeba dobře zvolit skriptovací jazyk, protože jeho produkt je pro aplikace zásadní. Mezi nejpoužívanější a vývojáři nejoblíbenější patří Python, JavaScript, PHP nebo také C++. V mém případě jsem zvolil JavaScript. Jeho syntaxe je uživatelsky velice příjemná, a i pro začátečníka jednoduchá na osvojení základních náležitostí. Velkou výhodou je jednoduché propojení s HTML a CSS, které postrádá

¹Více informací doporučuji [31].

například Python, kde je proces složitější. V průběhu skriptování můžeme sledovat výsledek přímo v okně webu, což je pohodlnější, než sledovat nejprve konzoli v editoru a po následném připojení skriptu k HTML vidět výsledek na webové stránce.

Jako vývojářské prostředí jsem využil volně dostupný editor Visual Studio Code, který poskytuje programátorovi nástroje pro zpřehlednění kódu a zjednodušuje s ním manipulaci, což se projeví hlavně u delších kódů.

5.1.1 Použitá rozšíření

MathJax

Jedná se o volně dostupnou javascriptovou knihovnu pro korektní sázení matematického textu za pomoci L^AT_EXu a MathML.

Math.js

Stejně jako u MathJax, tak i Math.js je volně dostupná javascriptová knihovna zaměřená zejména na usnadnění práce s matematikou. Konkrétně v této webové aplikaci jsou použity funkce:

math.complex(x, y) - zápis komplexního čísla $x + yi$,
math.sqrt(x, y) - odmocnina z komplexního čísla $x + yi$,
math.multiply(x, y) - roznásobení dvou hodnot x a y ,
math.add(x, y) - součet hodnot x a y .

5.2 Popis aplikace

Webová aplikace je zaměřena na generování Juliových množin pomocí různých zobrazovacích metod. Dále jsou zde graficky znázorněny některé její základní vlastnosti a také souvislosti z Mandelbrotovou množinou. Vše je rozděleno na čtyři části, které na sebe logicky navazují:

- Úvod - potřebná teorie pro pochopení problematiky,
- Orbita - vykreslení orbity pro Mandelbrotovu a Juliovu množinu,
- M vs. J - vykreslení Juliovu množiny v závislosti na Mandelbrotově množině,
- J vs. J - porovnání vizualizačních metod.

U programování celé aplikace jsem se zaměřil na jednoduchost při používání, bezproblémovou orientaci ve struktuře webu a také intuitivnost ovládání interaktivních pasáží, aby každý uživatel věděl, co má dělat a také co od webu očekávat za odpověď. V každé části je vždy pár odstavců postihujících vše důležité pro danou část a následně již samotná interaktivní aplikace. Teoretická pasáž popisovaná v textech je velmi důležitá pro pochopení celé problematiky, které se tato webová aplikace věnuje. Je tedy potřeba dbát na porozumění, čehož jsem se snažil dosáhnout ne zcela exaktními definicemi všech potřebných pojmu, ale spíše srozumitelněji podaným souvislejším textem popisujícím vše potřebné. Aplikace je tak mnohem přívětivější i k uživatelům, kteří o fraktální geometrii ještě neslyšeli.

Úvod

V této části uživateli popisuji nejprve co vše je v aplikaci možné zkoumat a následně již základní náležitosti Juliových množin a Mandelbrotovy množiny sloužící jako potřebný vědomostní základ k pochopení dané problematiky.

Orbita

Uživatel se nejprve seznámí s pojmem orbita a jakou hraje roli u obou fraktálních množin. Interaktivní část se skládá ze dvou pláten pod sebou, kde horní plátno je pro Mandelbrotovu množinu a spodní pro Juliovu množinu. Pohybem kurzoru myši po horním resp. dolním plátně se uživateli neustále ukazují v levém dolním resp. horním rohu plátna souřadnice polohy myši v komplexní rovině. Dále poté v dolním panelu uprostřed obou pláten aplikace popisuje chování orbity s počáteční hodnotou zvolenou opět kurzorem myši. Po kliknutí kdekoli na plátno určené Mandelbrotově množině se vykreslí kritická orbita pro c zvolené při kliknutí myší. Stejně tak se děje u druhého plátna, jen s rozdílem, že kliknutím se zvolí z_0 . Následně se všechny předešlé vykreslené orbity smažou po stisknutí „Reset orbit“ tlačítka v pravém dolním rohu obou pláten.

Zvolené c z horního plátna se navíc automaticky uloží do souřadnic konstanty c pro vygenerování Juliovu množiny pomocí tlačítka „Generate“ na spodním plátně. Díky tomu lze porovnat chování orbity u obou množin pro stejně c . V případě, že by uživatel chtěl využít své vlastní souřadnice, jednoduše stačí do reálné a imaginární části c zadat hodnoty manuálně.

M vs. J

Předposlední sekce aplikace popisuje vztah mezi Mandelbrotovou a Juliovou množinou, kde Mandelbrotovu množinu můžeme nazvat jako mapu pro Juliovu množiny. Tento fakt je demostrován v programu, kde uživatel opět zvolí pomocí kurzoru myši souřadnice c a v závislosti na zvolené hodnotě se vygeneruje příslušná Juliova množina. Opět se aktuální poloha kurzoru zobrazuje v levém horním rohu v podobě souřadnic v komplexní rovině. Máme také možnost vybrat ze tří různých zobrazovacích metod: Escape time algorithm, Distance estimator method a Inverse iteration method. Program je vizuálně řešen dvěma plátny vedle sebe, tedy opět velmi intuitivní na ovládání a jednoduché na pochopení. Po každém kliknutí navíc kurzor zanechá na Mandelbrotově množině červený bod v místě volby, tedy po změně metody můžeme jednoduše porovnat dvě metody opětovným kliknutím na to stejné místo.

J vs. J

V posledním bloku aplikace je kladen důraz na porovnání zobrazovacích metod, zmíněných v předešlé sekci, z různých úhlů pohledu. Porovnává se rychlosť vykreslení jednotlivých metod, která se zobrazuje v milisekundách v dolním panelu zvlášť pro obě metody, vizuální vzhled, či přesnost. Pro vygenerování množin stačí zvolit komplexní konstantu c , pro kterou budeme množiny generovat. Následně zvolíme pro každé plátno zvlášť metodu vizualizace. Po volbě všech potřebných možností stačí opět pouze stisknout tlačítko „Generate“.

5.3 Vizualizace

Ukažme si nyní, jak vygenerovat množiny pomocí JavaScriptu. Už víme, že oba typy množin jsou generovány pomocí iteračního předpisu (3.1), produkujícího pro jednotlivá

z_0 posloupnost hodnot z_0, z_1, \dots, z_n . Pomocí těchto hodnot zjišťujeme, zda-li posloupnost po n iteracích diverguje, či nikoliv. V počítačové grafice však nemůžeme opakovat iterační proces nekonečně krát. Musíme se tedy omezit na určitý počet iterací, pro který budeme schopni bezpečně rozhodnout, jestli posloupnost pro počáteční hodnotu z_0 zůstane ohraničená, nebo diverguje k nekonečnu.

Mandelbrotova množina

V praxi, vycházející z teorie v části 4.2, celý cyklus probíhá následovně. Zafixujeme $z = 0 + 0i$ a zvolíme maximální počet iterací. Následně bereme postupně každý bod z výrezu komplexní roviny a po každé smyčce cyklu sledujeme jeho hodnotu. Když $|z_k| \leq 2$ pro libovolné k , cyklus pokračuje. Pokud se tak děje až do maximálního počtu iterací, tedy $k = n$, řekneme, že bod leží uvnitř Mandelbrotovy množiny. Zjistíme-li však v některé smyčce, že $|z_k| > 2$, pro $k < n$, cyklus zastavíme a řekneme, že bod neleží v Mandelbrotově množině, viz důkaz 3.3 na straně 30. Pro lepší pochopení tohoto cyklu si uvedeme pseudokód:

```

Nechť iter = 0;
Nechť maxIter = 100;
Nechť z = 0;

Jestliže iter < maxIter a zároveň |z| <= 2 proved:
{
    z = z^2 + c;
    iter = iter + 1;
}
Jestliže iter == maxIter, bod leží v M-setu;
Jinak, bod neleží v M-setu;

```

Samotný pseudokód není nikterak složitý, jenomže v JavaScriptu narazíme na menší problém, protože z i c jsou komplexní čísla, která se neřadí mezi základní datové typy jazyka JavaScript. Musíme se tedy nějak zbavit imaginární jednotky ve výpočtu. Z (2.3) víme, že z a c lze rozepsat jako

$$\begin{aligned} z &= z_x + z_y, \\ c &= c_x + c_y, \end{aligned}$$

kde z_x a c_x jsou reálné části a z_y a c_y imaginární části. Dosazením do (3.1) dostáváme

$$f(z) = (z_x + z_y)^2 + (c_x + c_y) = z_x^2 + 2z_x z_y - z_y^2 + c_x + c_y,$$

když víme, že $z_y = yi$, pak $z_y^2 = -y^2$ a tedy $-z_y^2 = y^2$. Nyní už nebude problém rozepsat reálnou a imaginární část z pro iterační proces a vyhnout se přitom imaginární jednotce i , protože každou část počítáme zvlášť, můžu i vynechat. Máme tedy

$$\begin{aligned} z_x &= z_x^2 - z_y^2 + c_x, \\ z_y &= 2z_x z_y + c_y. \end{aligned}$$

Tím jsme vyřešili problém s komplexními čísly. Je však potřeba ještě zmínit, že výpočet absolutní hodnoty, viz rovnost (2.4), při testování limitního poloměru divergence, je velmi

časově náročný. Je potřeba počítat druhou mocninu a následně ještě výraz odmocnit. Tento problém vyřešíme umocněním a proces tak výrazně urychlíme. V našem případě máme

$$z_x^2 + z_y^2 \leq 4.$$

V tuto chvíli nám již nic nebrání v zapsání iteračního procesu v jazyce JavaScript. Program testuje pro zvolenou hodnotu c , zda leží v Mandelbrotově množině, či nikoliv. Výsledek testu vypisuje do konzole.

```

1 function mandelbrot(cx, cy) // funkce se vstupní hodnotou c
2 {
3     var zx = 0; // fixní z = 0 + 0i
4     var zy = 0; // druhá mocnina z
5     var zx2, zy2; // druhá mocnina z
6
7     var iter = 0; // počítadlo iterací
8     var maxIter = 100; // maximální počet iterací
9
10    while(zx2 + zy2 <= 4 && iter < maxIter) // dokud |z|^2 < 4 a iter < maxIter
11        {
12            zx2 = zx*zx; // provadej smyčku
13            zy2 = zy*zy; // zx^2
14            zy = 2*zx*zy + cy; // zy^2
15            zx = zx2 - zy2 + cx; // komplexní část z
16            zy = zy - 2*zx*zy; // reálná část z
17            iter++ // iter = iter + 1
18        }
19        if(iter == maxIter) // pokud iter rovno maxIter
20        {
21            console.log("lezi"); // bod leží v M-setu
22        } else { // jinak bod neleží v M-setu
23            console.log("nelezi");
24        }
25 }
```

Juliova množina

Jak už víme z úvodu sekce 3.5.1, Juliových množin je nekonečně mnoho a každé takové množině odpovídá jiné c . V praxi tedy pro určitou Juliovu množinu zafixujeme hodnotu c a zvolíme maximální počet iterací, po který se bude provádět proces. Dále už vše probíhá stejně jako u Mandelbrotovy množiny, jen s tím rozdílem, že testujeme jednotlivé průchody vzhledem k $r(c)$, nikoliv pouze k hodnotě dva. Tedy provádíme iterační proces a po jeho ukončení rozhodujeme, jestli počáteční hodnota z_0 leží uvnitř množiny, či nikoliv. Opět pro větší názornost uvedeme pseudokód:

```

Nechť iter = 0;
Nechť maxIter = 100;
Nechť z = hodnota z v komplexní rovině;
Jestliže iter < maxIter a zarovně |z| <= r(c) proved:
{
    z = z^2 + c;
    iter = iter+1;
}
```

```

        }
Jestliže iter == maxIter, bod leží v J-setu;
Jinak, bod neleží v J-setu;
```

Problém s komplexními čísly a absolutní hodnotou v kódu řešíme identicky jako u Mandelbrotovy množiny, a tak si rovnou uvedeme program, který pro fixní c testuje zvolenou hodnotu z_0 z komplexní roviny a výsledek opět vypíše do konzole.

```

1 function julia(cx,cy,zx,zy) // funkce se vstupnimi hodnotami c a z
2 {
3     if(cx*cx+cy*cy > 4)                                //r(c) = max(|c|,2)
4     {
5         r_c = cx*cx+cy*cy;
6     }else{
7         r_c = 4;
8     }
9     var zx2, zy2;                                         // druhá mocnina z
10
11    var iter = 0;                                         // pocitadlo iteraci
12    var maxIter = 100;                                     // maximalni pocet iteraci
13
14    do                                                    //provadej smycku
15    {
16        zx2 = zx*zx;                                     // zx^2
17        zy2 = zy*zy;                                     // zy^2
18        zy = 2*zx*zy + cy;                               // komplexni cast z
19        zx = zx2 - zy2 + cx;                            // realna cast z
20        iter++                                           // iter = iter + 1
21    }
22    while(zx2 + zy2 <= r_c && iter < maxIter);       // dokud obe podminky zaroven plati, opakuj
23    if(iter == maxIter)                                 // pokud iter rovno maxIter
24    {
25        console.log("lezi");                           // bod lezi v M-setu
26    }else{
27        console.log("nelezi");                         // jinak bod nelezi v M-setu
28    }
29}
30 }
```

5.3.1 Metody

Snaha bude pomocí modifikací testovacích algoritmů zmíněných výše, vykreslit fraktály na dvourozměrné plátno² pomocí JavaScriptu. Už umíme otestovat každý bod v komplexní rovině a říci, kdy jeho velikost překročí limitní poloměr divergence. Zbývá nám vyřešit poslední úskalí. V iteračním procesu počítáme s pozicí komplexního čísla v rovině, naopak u přiřazování barvy musíme uvažovat souřadnice odpovídajícího pixelu na plátně. Potřebujeme tedy převodní vztah, který zaručí, že každé komplexní číslo bude mít odpovídající pozici na plátně a vše se vykreslí na správné místo. Mějme tedy opět $c = c_x + c_y$ a označme $cxComplex$ a $cyComplex$ jako reálnou a imaginární část čísla c v komplexní rovině.

²Pod pojmem plátno si lze představit plochu na obrazovce, složenou z pixelů, na kterou budeme vizualizovat fraktální obrazce.

Dále nechť $cxCanvas$ a $cyCanvas$ označují x -ovou a y -ovou souřadnici pixelu na plátně. Dále je třeba znát maximum a minimum výřezu komplexní roviny pro Re osu a Im osu. Protože nebude třeba z velké části aplikaci volit rozdílná maxima a minima pro jednotlivé osy, pak $cMax$ je maximum obou os a $cMin$ je jejich minimum. Jediný posun ve směru Re osy je třeba při generování Mandelbrotovy množiny. tuto skutečnost ošetříme pomocí proměnné $posunX$. Stejně tak se dá vyřešit případný posun po Im ose za pomoci proměnné $posunY$. Navíc, všechna plátna v aplikaci jsou čtvercová, tedy aby zůstal zachován originální tvar množin, $cMax = -cMin$. Konečně šířku plátna označme jako $widthCanvas$ a výšku $heightCanvas$. Při převodu musíme dát pozor na orientaci y -ové osy na plátně. Počátek souřadnicového systému je totiž v levém horním rohu s orientací y -ové osy směrem dolů. Pak tedy pro převodní vztah platí

$$cxComplex = \frac{cMax}{widthCanvas/2} (cxCanvas - widthCanvas/2) + posunX,$$

$$cyComplex = -\frac{cMax}{heightCanvas/2} (cyCanvas - heightCanvas/2) + posunY.$$

Pro hodnoty $cxCanvas$ a $cyCanvas$ pak

$$cxCanvas = \left(\frac{cxComplex - posunX}{cMax} + 1 \right) widthCanvas/2, \quad (5.2a)$$

$$cyCanvas = \left(\frac{cyComplex - posunY}{cMax} + 1 \right) heightCanvas/2. \quad (5.2b)$$

Můžeme si všimnout, že formálně je potřeba u obou směrů převodu vyžadovat nenulovost hodnot ve jmenovateli. Nenulovost plátna je již ze začátku z logických důvodů a protože se v celé aplikaci generují pouze celé Juliovu množiny, bez možnosti přibližování jejich částí, tak i $cMax$ bude vždy nenulové.

Escape time algorithm

Řadí se mezi jedny z nejstarších algoritmů určených pro generování fraktálních obrazců. V základní verzi tohoto algoritmu přiřazujeme pouze černou barvu počátečním hodnotám z_0 , jejichž orbita nediverguje a bílou těm, které divergují. Jedná se tak pouze o volbu *true/false* a výsledný fraktál je tedy dvoubarevný obrazec. V počátcích počítačové grafiky byl i tento způsob brán za obrovský úspěch, jenomže s rapidním rozvojem začala být pouze dvoubarevná vizualizace nedostatečná. Vykreslit Juliovu množinu pomocí této základní metody můžeme díky níže uvedeném kódu:

```

1 var maxIter = 255;
2 var cmax = 1.3;
3 var zx2,zy2;
4
5 function draw_eta(cx,cy)
6 {
7   if(cx*cx+cy*cy > 4)
8   {
9     r_c = cx*cx+cy*cy;
10 } else{

```

```

11     r_c = 4;
12 }
13 for(var x = 0; x < canvas.width; x++)
14 {
15     for(var y = 0; y < canvas.height; y++)
16     {
17         var zxComplex = cmax/(canvas.width/2)*(x - canvas.width/2);
18         var zyComplex = -cmax/(canvas.height/2)*(y - canvas.height/2);
19         var iter = 0;
20         do{
21             zx2 = zxComplex*zxComplex;
22             zy2 = zyComplex*zyComplex;
23             zyComplex = 2*zxComplex*zyComplex + cy;
24             zxComplex = (zx2 - zy2) + cx;
25             iter++;
26         }
27         while(zx2 + zy2 <= r_c && iter < maxIter);
28
29         if(iter == maxIter)
30         {
31             kontext.fillStyle = "black";
32             kontext.fillRect(x,y,1,1);
33         }else{
34             kontext.fillStyle = "white";
35             kontext.fillRect(x,y,1,1);
36         }
37     }
38 }
39 }
```

Zcela logické rozšíření základní verze escape time algorithm (ETA) je, že nesledujeme pouze jestli přesáhne velikost z_k hodnotu $r(c)$, ale také zkoumáme, pro jak velké k se tak stane a podle toho přiřadíme pixelu určitou barvu. Nabízí se tedy zvolit pro zjednodušení číslo 255 jako maximální počet iterací, aby pro každou číselnou hodnotu úniku posloupnosti odpovídal jeden odstín barvy.

Ve webové aplikaci je implementována právě rozšířená varianta ETA. Je využito funkce, která každý pixel v poli rozdělí na čtyři části odpovídající barevnému modelu RGBA vycházejícího z klasického RGB modelu pro aditivního míchání barev (viz [33]). Je však ještě rozšířen o tzv. alfa kanál udávající průhlednost. Tedy každé části pixelu reprezentující červenou, zelenou a modrou složku výsledné barvy, přiřadíme hodnotu 0–255, která udává významnost zastoupení určité složky a poté stejně tak pro alfa-kanál. Například zápis *rgba(255,0,0,255)*, je zápis čistě červené, neprůhledné barvy, protože červená složka má nejvyšší možnou hodnotu a naopak složky modré a zelené barvy jsou na minimu. Použití této funkce v kódu je demonstrováno níže, kde je ukázána pouze část odlišující se od základní ETA.

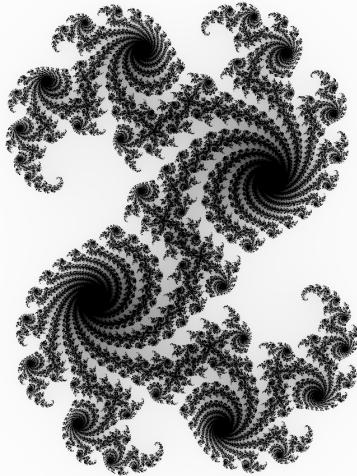
```

1 var input = 255 - 255*iter/(100);
2 var pozice = (x+y*canvas.width) * 4;
3 imgData.data[pozice + 0] = input;
4 imgData.data[pozice + 1] = input;
5 imgData.data[pozice + 2] = input;
6 imgData.data[pozice + 3] = 255;
```

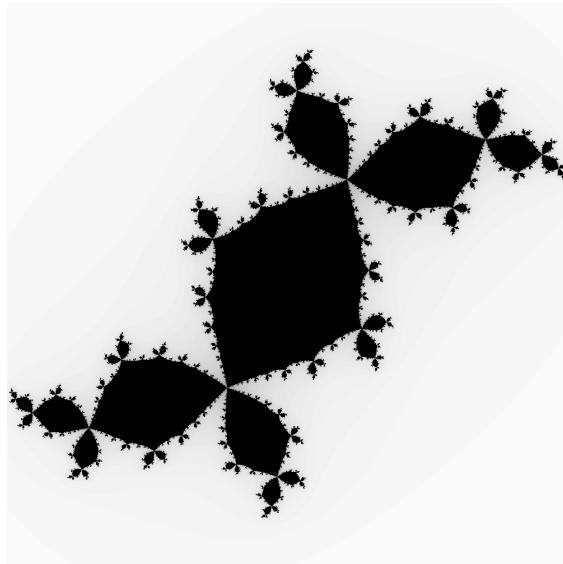
```

7     }
8     }
9 }kontext.putImageData(imgData, 0, 0);
10 }
```

Na obrázcích 5.1a a 5.1b vidíme filled Juliové množiny, vygenerované pomocí rozšířené ETA pro dvě různá c .



(a) $c = 0.32 + 0.044i$



(b) $c = -0.1225 - 0.75i$

Obrázek 5.1: Juliové množiny vykreslené pomocí ETA.

Distance estimator method

Postup při vizualizaci pomoci této metody bude velmi podobný jako u předešlé, ale jak již název napovídá, nebudeme sledovat po kolikáté iteraci posloupnost překročí limitní poloměr divergence, ale budeme sledovat vzdálenost počátečního bodu iterace od hranice množiny. Zcela formálně jsou následující vztahy definovány pro Mandelborotovu množinu a pro souvislé zástupce Juliových množin³. Tato metoda ale i v případě nesouvislosti generuje velmi zdařilé vizuální výsledky, a tak si dovolím, stejně jako autor [29], Distance estimator method zobecnit pro libovolnou Juliovu množinu.

Vztah pro výpočet vzdálenosti vychází z funkce nazývané *Hubbard-Douady potential*, která je definována v bodě z_0 jako

$$G(z_0) = \lim_{k \rightarrow \infty} \frac{1}{2^k} \ln |z_k|. \text{⁴}$$

Za jejího použití pak platí, že vzdálenost bodu od množiny lze odhadnout pomocí vztahu

$$d \leq \frac{2 \sinh G(z_0)}{|G'(z_0)|},$$

³Více podrobností najeznete v [29].

⁴Označení G pro funkci vychází z používaného názvu „Greenova funkce.“

kde $|G'(z_0)|$ vyjadřuje gradient G . Uvědomíme-li si, že pro z_0 blízko Juliově množině je $G(z_0) = 0$ a že $\sinh 0 = 0$, pak můžeme využít praktické approximace $\sinh G(z_0) \approx G(z_0)$, z které dostáváme

$$d \approx \frac{2G(z_0)}{|G'(z_0)|},$$

kde derivace $G(z_0)$ podle z_0 je

$$G'(z_0) = \lim_{k \rightarrow \infty} \frac{1}{2^k} \frac{|z'_k|}{|z_k|}.$$

Konečně potom pro d máme po menších úpravách vztah

$$d \approx \frac{2|z_k| \ln |z_k|}{|z'_k|}.$$

Při počítání derivace si musíme dát pozor podle čeho derivujeme. U Juliově množiny je to derivace podle z_0 , tedy derivací složené funkce dostáváme

$$z'_{n+1} = 2z_n z'_n \quad z'_0 = 1.$$

Pro Mandelbrotovu množinu by byl postup identický, jen místo $G(z_0)$ máme $G(c)$, tedy pro derivaci podle c platí

$$z'_{n+1} = 2z_n z'_n + 1 \quad z'_0 = 0.$$

V programu je potřeba rozdělit derivaci na reálnou a komplexní část, abychom se opět vyhli i .

$$z'_{n+1} = 2z_n z'_n = 2(z_x + z_y)(z'_x + z'_y) = 2(z_x z'_x + z_x z'_y + z_y z'_x - z_y z'_y).$$

Pro reálnou a imaginární složku z'_{n+1} platí

$$\begin{aligned} z'_x &= 2(z_x z'_x - z_y z'_y), \\ z'_y &= 2(z_x z'_y + z_y z'_x). \end{aligned}$$

Jednodušší způsob vizualizace pomocí tohoto algoritmu spočívá pouze ve stanovení limitní meze R pro d . Tedy $d \leq R$ značí, že bod leží v množině a obarvíme jej černou barvou. Body se vzdáleností větší, než tato mez, budou označeny jako body mimo množinu a budou bílé.

Vizuálně mnohem zajímavější modifikace distance estimator method (DEM) je pro každé d volba určitého odstínu barvy. V praxi je však výsledek vzdálenosti bodů od množiny velmi malé číslo, bude tedy potřeba převést d tak, abychom mohli každému pixelu přiřadit hodnotu $index_{color}$ v intervalu 0–255. Využijeme vztah pro převod vzdálenosti na rozumné hodnoty udávající barvu

$$index_{color} = -k \ln(d),$$

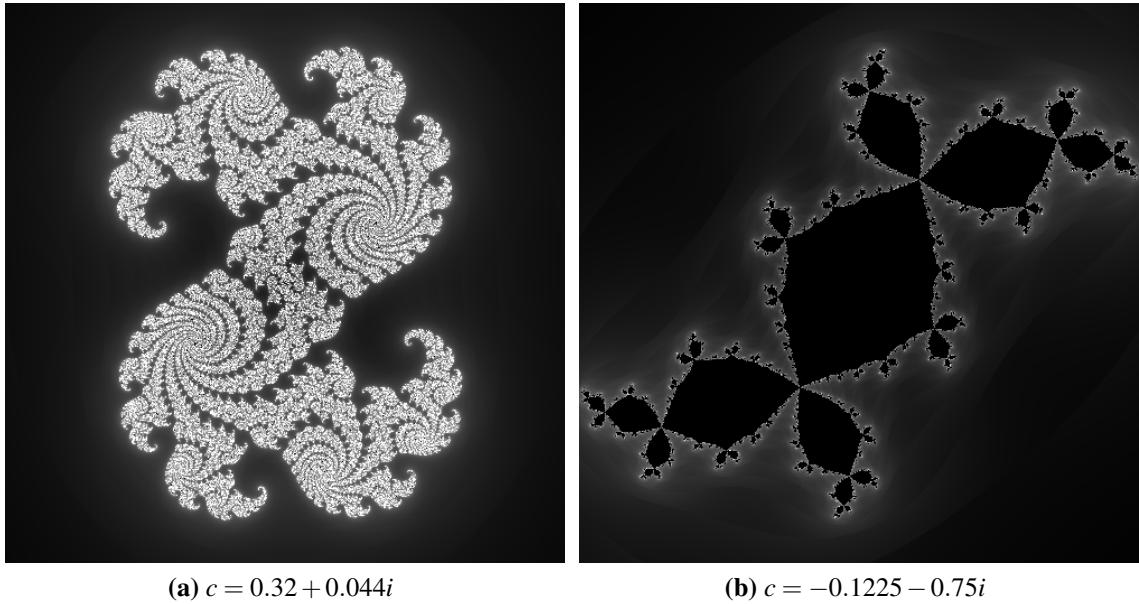
kde k je nezáporná konstanta udávající jas⁵. Úryvek programu, využívající modifikovanou vizualizační metodu odhadu vzdálenosti pro Juliovu množinu se v JavaScriptu dá napsat následovně:

⁵Čím nižší číslo, tím budou hodnoty $index_{color}$ blíž k 255 a tedy celkový jas fraktálu bude vyšší.

```

1 var maxIter = 255;
2 var cmax = 1.3;
3 function draw_dem(cx, cy)
4 {
5     if(cx*cx+cy*cy > 4)
6     {
7         r_c = cx*cx+cy*cy;
8     }else{
9         r_c = 4;
10    }
11 for(var x = 0; x < canvas.width; x++)
12 {
13     for(var y = 0; y < canvas.height; y++)
14     {
15         var zxComplex = cmax/(canvas.width/2)*(x - canvas.width/2);
16         var zyComplex = -cmax/(canvas.height/2)*(y - canvas.height/2);
17
18         var dzx = 2*zxComplex;
19         var dzy = 2*zyComplex;
20         var zx2 = zxComplex*zxComplex;
21         var zy2 = zyComplex*zyComplex;
22         var iter = 0;
23         do{
24             var dzxN = 2*zxComplex*dzx - 2*zyComplex*dzy;
25             var dzyN = 2*zxComplex*dzy + 2*zyComplex*dzx;
26
27             var zxN = (zxComplex*zxComplex - zyComplex*zyComplex) + cx;
28             var zyN = 2*zxComplex*zyComplex + cy;
29
30             zxComplex = zxN;
31             zyComplex = zyN;
32
33             dzx = dzxN;
34             dzy = dzyN;
35
36             zx2 = zxComplex*zxComplex;
37             zy2 = zyComplex*zyComplex;
38
39             iter++;
40         }
41         while(iter < maxIter && zx2 + zy2 <= r_c);
42         var dzx2 = dzx*dzx;
43         var dzy2 = dzy*dzy;
44         var d = 2*(Math.sqrt(zx2+zy2)*Math.log(Math.sqrt(zx2+zy2)))/Math.
45             sqrt(dzx2+dzy2);
46         var input = -17*Math.log(d);
47         var pozice = (x+y*canvas.width) * 4;
48         imgData.data[idx + 0] = input;
49         imgData.data[idx + 1] = input;
50         imgData.data[idx + 2] = input;
51         imgData.data[idx + 3] = 255;
52     }
53 }kontext.putImageData(imgData, 0, 0);
54 }
```

Na obrázcích 5.2a a 5.2b lze vidět výsledek z výše napsaného programu pro dvě různá c .



Obrázek 5.2: Juliovu množinu vykreslené pomocí DEM.

Inverse iteration method

U poslední metody tohoto textu využijeme důsledku 3.6.1 o invarianci Juliovu množiny vůči iteraci a věty 3.6.7 popisující Juliovu množinu jako uzávěr množiny všech odpuzujících periodických bodů. Jen připomeňme, že periodický bod s periodou jedna je pevný bod. Spojením těchto dvou vlastností dostáváme základ inverse iteration method (IIM). Z první vlastnosti vychází, že můžeme využít iteraci inverzní funkce k $f(z) = z^2 + c$, tedy

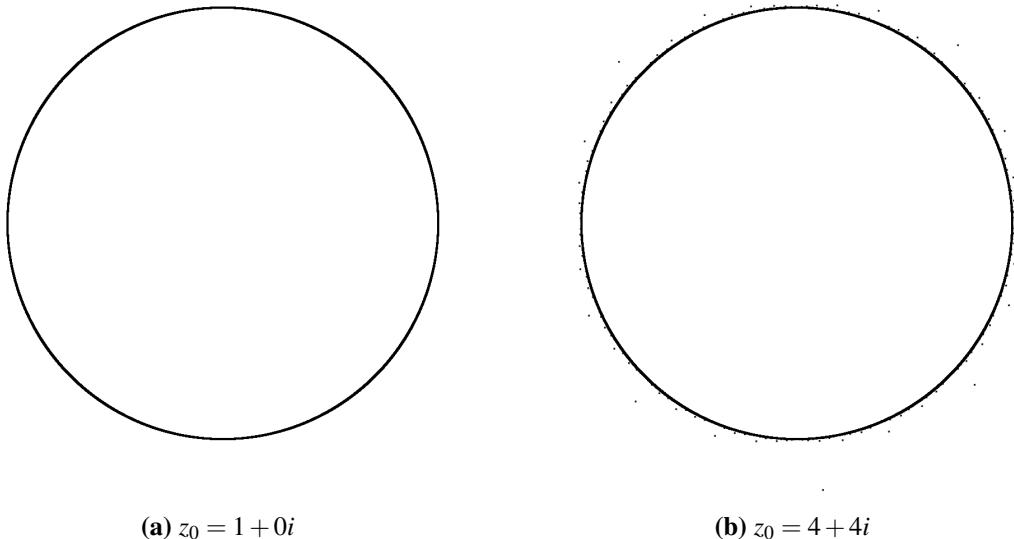
$$f^{-1}(z) = \pm\sqrt{z-c}. \quad (5.3)$$

Dále uvažme, že pokud je Juliovu množinu repelor pro $f(z)$, potom musí být zákonitě atraktor pro $f^{-1}(z)$. Vyberu-li tedy bod z *Escape set* blízko Juliově množině pro $f(z)$, pak jeho orbita v iteračním procesu diverguje do nekonečna, ale naopak pro $f^{-1}(z)$ orbita bodů blízko Juliově množině z *Escape set* k množině atrahuje. Naskytá se otázka, jak vhodně zvolit počáteční hodnotu z_0 , aby algoritmus generoval celou Juliovu množinu. Odpovědí je, že odpuzující periodické body (věta 3.6.5) a vzory bodů v Juliově množině (věta 3.6.8) jsou v ní husté. Tímto máme zajištěno, že pokud najdeme pro počáteční hodnotu z_0 odpuzující pevný bod, budeme mít dostatek různých výsledků pro uspokojivé vykreslení fraktálního obrazce.

Implementováním do praxe tedy pro vygenerování Juliovu množinu potřebujeme nejprve zvolit konstantu c , pro kterou budeme vykreslovat obrazec. Následně hledáme pevný bod což nám v \mathbb{C} dává dva výsledky, z kterých vybereme odpuzující pevný bod pomocí věty 3.2.1. Z (5.3) vidíme, že v každém kroku dostaneme dvě další hodnoty. Znamená to, že počet hodnot pro generování poroste velmi rychle, přesněji v n -tému kroku přibude 2^n

nových hodnot. Tedy obvyklou volbou 255 iterací bychom dostali v poslední smyčce 2^{255} hodnot, což je skoro nemožné na běžném počítači v rozumném časovém horizontu provést, proto volba hodnoty pro maximální počet iterací mezi 15 – 20 je zcela dostačující.

Počáteční test na odpuzující pevný bod je velmi časově náročný. Snaha tedy bude nalézt ekvivalentní řešení počáteční podmínky tak, aby klesla časová náročnost co nejvíce, ale přesnost zůstala stejná. K tomu nám postačí rozebíraná vlastnost o pár odstavců dříve. Zvolíme-li libovolně velké z_0 , které zaručeně nebude v Juliově množině, pak jeho orbita pro $f^{-1}(z)$ atrahuje k Juliově množině. Na obrázku 5.3 můžeme vidět porovnání mezi Juliovou množinou s počáteční hodnotou rovnou odpuzujícímu pevnému bodu $z_0 = 1 + 0i$ v části 5.3a a Juliovou množinou, kde $z_0 = 4 + 4i$ v části 5.3b, pro stejné c . Jediný rozdíl je v pár pixelech⁶ vykreslených mimo množinu, což na plátně 500×500 pixelů nelze skoro zaznamenat.



Obrázek 5.3: Juliova množina pro $c = 0 + 0i$.

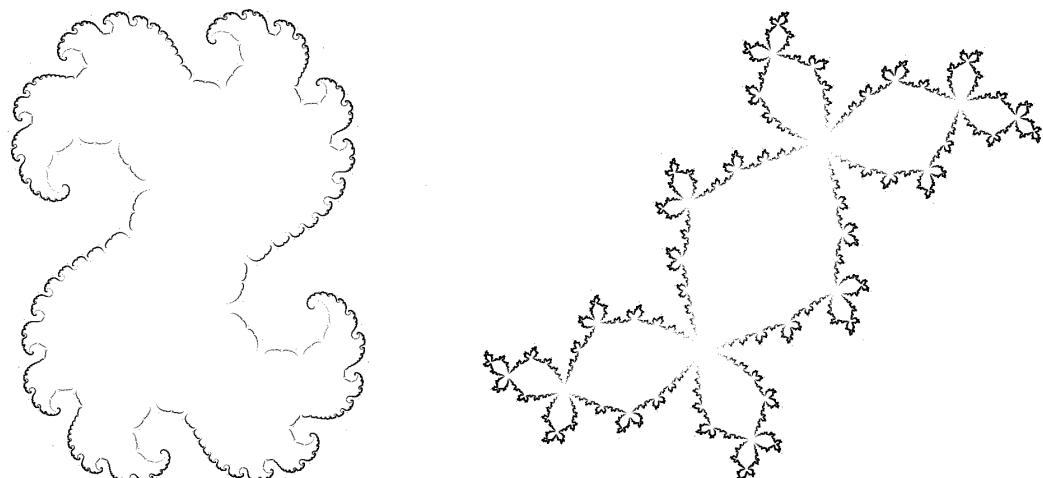
Při implementaci do webové aplikace nejprve zvolíme konstantu c , která nám určí výsledný tvar Juliově množiny, dále z_0 zafixujeme a nastavíme počet iterační kroků. Po převedení komplexního čísla do souřadnic pixelu podle (5.2a) a (5.2b), vykreslíme v každém průchodu výsledky na plátno, čímž se stává program efektivnějším, než kdybychom nejprve hodnoty ukládali a následně je až po skončení iteračního cyklu všechny vykreslovali zároveň. Metoda používá knihovnu Math.js využitou pro odmocninu z komplexního čísla, kterou bychom museli počítat zdlouhavě přes soustavu rovnic (viz [7, strana 731]).

⁶Pro demonstrativní účel a lepší viditelnost byla velikost jednotlivých pixelů mimo množinu zvětšena třikrát.

```

1  function draw_iim(cx ,cy ,zx0 ,zy0){
2    var cmax = 1.3;
3    var zx = [];
4    var zy = [];
5    var z = [];
6    var maxIter = 20;
7    var z0 = math.complex(zx0 ,zy0);
8    var c = math.complex(cx ,cy);
9
10   z[0] = math.sqrt(math.add(z0 ,math.multiply(c , -1)));
11   z[1] = math.multiply(z[0] , -1);
12
13   zx[0] = (canvas.width/2)*((z0.re/cmax)+1);
14   zy[0] = (canvas.height/2)*((z0.im/-cmax)+1);
15
16   var j = 0;
17   for(var i = 0; i < 2*Math.pow(2 ,maxIter); i=i+2)
18   {
19     z[i] = math.sqrt(math.add(z[j] ,math.multiply(c , -1)));
20     z[i+1] = math.multiply(z[i] , -1);
21     zx[j] = (canvas.width/2)*((z[j].re/cmax) + 1);
22     zy[j] = (canvas.height/2)*((z[j].im/-cmax) + 1);
23     kontext.fillRect(zx[j] ,zy[j] ,1,1);
24     j++;
25   }
26 }
```

Jedním z problémů IIM, jak jde vidět na obrázku 5.4 je, že určité body jsou mnohem více přitahující, než ostatní, tedy se často stává, že se nevykreslí všechny body množiny. Tento problém řeší tzv. *Modified Inverse Iteration Method* (MIIM). MIIM je však nad rámec této práce. V případě zájmu doporučuji [24] nebo [29].

(a) $c = 0.32 + 0.044i$ (b) $c = -0.1225 - 0.75i$ **Obrázek 5.4:** Juliové množiny vykreslené pomocí IIM.

Závěr

Hlavním cílem této práce bylo definování a popis základních vlastností Juliových množin a uvedení souvislostí Juliových množin a Mandelbrotovy množiny. Praktická část spočívala v navržení a naprogramování algoritmů pro vizualizaci Juliových množin, na jejichž základně vznikla webová aplikace dostupná pro potřeby obhajoby na <https://www.math.muni.cz/~xmacharacekj/>. Veškeré zdrojové kódy jsou k nalezení v archivu zde <https://is.muni.cz/auth/th/oo0gk/>.

Pomocí prvních dvou kapitol jsme čtenáři poskytli potřebný matematický aparát využívaný po zbytek práce. Ve třetí kapitole jsme dokázali stěžejní pojem limitního poloměru konvergence a následně pomocí něj a approximace prisoner set nastínili první možné vykreslení Juliovu množiny a poté podrobně rozebrali jejich vlastnosti, které jsme využili při definování vizualizačních metod.

Zjistili jsme, že při generování Juliových množin záleží na faktu, zda-li konstanta c leží uvnitř Mandelbrotovy množiny, či nikoliv a že to má velký dopad na výsledný tvar a souvislost množiny. Dále jsme přišli na velmi užitečnou vlastnost, že Mandelbrotova množina určuje periodu orbity Juliovu množiny. Tato zjištění je možno zkoumat i ve webové aplikaci.

V praktické části práce jsme naprogramovali jazykem JavaScript s pomocí HTML a CSS webovou aplikaci představující základní náležitosti Juliových množin, které má uživatel možnost interaktivně prozkoumat. Většina aplikace je však zaměřena na vizualizaci Juliových množin pomocí různých metod a jejich následné porovnávání.

Nejprve jsme odvodili převodní vztah čísla z komplexní roviny do souřadnic plátna a poté si představili tři vizualizační metody, kde každá stojí na jiném principu generovaní. U metody escape time algorithm jsme po modifikaci základního kódu dokázali detektovat rychlosť divergence orbit počátečních hodnot a dosáhli tak plynulého přechodu barev ve výsledném obrazci. Stejně tomu bylo i u distance estimator method, kde jsme po základní teorii a odvození vzorce byli schopni odhadnout vzdálenost počáteční hodnoty a množiny a generovat opět obrazec s plynulými přechody. U poslední metody inverse iteration method jsme byli nuceni, jako u jediné, využít speciální knihovnu Math.js pro práci s komplexními čísly. Na základě vlastností Juliovu množiny z teoretické části jsme využili iteraci inverzní funkce a získali tentokrát pouze Juliovu množinu jako hranici bez vnitřní části. Fakt, že jsme nebyli schopni zobrazit filled Juliovu množinu vychází přímo z teorie metody.

Obzvláště u inverse iteration method jsme pozorovali značné nedostatky ve vizualizaci množiny, kdy se pro komplikovanější útvary, hlavně blízko hranice Mandelbrotovy množiny, nevykreslil celý obrazec ale jen jeho obrys. Tento fakt by bylo určitě vhodné zkoumat podrobněji, například modifikací na modified inverse iteration method, ale to je již nad rámec této bakalářské práce.

Příloha

V archivu této práce je obsažena příloha s následujícími složkami a soubory:

1. Složka `web_js` se skripty JavaScript k interaktivním částem webové aplikace:

- `roll.js` (rolovací okna s odstavcem textu),
- `mandelbrot0.js` (plátno Mandelbrotovy množiny v sekci orbit a vše s ním spojené),
- `julia0.js` (plátno Juliový množiny v sekci orbit a vše s ním spojené),
- `mandelbrotMJ.js` (plátno Mandelbrotovy množiny v sekci M vs. J a vše s ním spojené),
- `juliaMJ.js` (plátno Juliový množiny v sekci M vs. J a vše s ním spojené),
- `juliaVsJuliaL.js` (levé plátno Juliový množiny v sekci J vs. J a vše s ním spojené),
- `juliaVsJuliaR.js` (pravé plátno Juliový množiny v sekci J vs. J a vše s ním spojené).

2. Složka `web_html` se soubory `html` dokumentů jednotlivých částí webové aplikace:

- `index.html` (podstránka s názvem *Introduciton*),
- `orbit.html` (podstránka s názvem *Orbit*),
- `mandelbrotVsJulia.html` (podstránka s názvem M vs. J),
- `juliaVsJulia.html` (podstránka s názvem J vs. J).

3. `style.css` (soubor CSS k naformátování vizuálního vzhledu aplikace).

4. `sierpinkskiTriangle.js` (skript JavaScript k vygenerování Sierpińského trojúhelníka).

5. `obrazky` (barevné obrázky Juliových množin ve vysokém rozlišení).

Příslušný archiv je dostupný zde <https://is.muni.cz/auth/th/oo0gk/>.

Seznam použité literatury

- [1] DOŠLÁ, Zuzana a Ondřej DOŠLÝ. *Metrické prostory: teorie a příklady*. 4. vydání. Brno: Masarykova univerzita v Brně, 2016, viii, 90 s. ISBN 978-80-210-8357-8.
- [2] ANDRES, Jan, Jiří FIŠER a Miroslav RYPKA. *Dynamické systémy 3: úvod do teorie deterministického chaosu a fraktální geometrie*. Olomouc: Univerzita Palackého v Olomouci, 2015, 240 s. ISBN 978-80-244-4646-2.
- [3] LAMPART, Marek, Jiří HORÁK a Igor IVAN. *Úvod do dynamických systémů: teorie a praxe v geoinformatici*. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, 2013, 200 s. ISBN 978-80-248-3185-5.
- [4] FALCONER, Kenneth. *Fractal Geometry: Mathematical Foundations and Application*. 2nd ed. University of St Andrews, UK: Wiley, 2003, xxvii, 337 s. ISBN 0-470-84861-8.
- [5] WEISNER, Robert. *Užití a zneužití fraktálů*. Brno, 2006, 66 s. Diplomová práce. Masarykova univerzita v Brně. Vedoucí práce Zdeněk Pospíšil.
- [6] KURAČ, Zbyněk. *Úvod do fraktální geometrie*. Olomouc, 2015, 93 s. Bakalářská práce. Univerzita Palackého, Přírodovědecká fakulta, Katedra algebry a geometrie. Vedoucí práce Michal Botur.
- [7] PEITGEN, Heinz-Otto, Hartmut JÜRGENS a Dietmar SAUPE. *Chaos and Fractals: New Frontiers of Science*. 2nd ed. New York: Springer-Verlag, 1992, xvi, 984 s. ISBN 0-387-20229-3.
- [8] RASBAND, S. Neil. *Chaotic Dynamics Of Nonlinear Systems*. New York: Wiley, 1990, x, 230 s. ISBN 0-471-63418-2.
- [9] KNILL, Oliver. *Dynamical Systems* [online]. Cambridge (Massachusetts): Harvard University, 2005, 179 s. [cit. 2022-03-30]. Dostupné z: https://people.math.harvard.edu/~knill/teaching/math118/118_dynamicalsystems.pdf
- [10] MASCELLANI, Giovanni. *The Hausdorff dimension of the boundary of the Mandelbrot set*. Pisa, 2011, v, 40 s. Dostupné také z: <https://poisson.phc.dmi.unipi.it/~mascellani/lib/exe/fetch.php?media=university:thesis.pdf>. Bakalářská práce. Università degli Studi di Pisa, Facoltà di Scienze Matematiche, Fisiche e Naturali. Vedoucí práce Marco Abate.

- [11] KALAS, Josef. *Analýza v komplexním oboru*. Brno: Masarykova univerzita v Brně, 2006, iv, 202 s. ISBN 80-210-4045-9.
- [12] CALDA, Emil. *Matematika pro gymnázia: Komplexní čísla*. 3. vydání. Praha: Prometheus, 2000, 136 s. ISBN 80-7196-187-6.
- [13] DEVANEY, Robert L. *The Mandelbrot and Julia Sets: A Tool Kit of Dynamics Activities*. Emeryville, CA: Key Curriculum Press, 2000, xiv, 151 s. ISBN 1-55953-357-9.
- [14] REAKES, Kai. *A Couple of Proofs of De Moivre-s Theorem: My Favourite Piece of Maths* [online]. 2008 [cit. 2022-03-15]. Dostupné z: <http://mathsathawthorn.pbworks.com/f/De+Moivre%27s+Theorem+and+my+favourite+piece+of+maths.pdf>
- [15] DUMMIT, Evan. *Dynamics, Chaos, and Fractals: Introduction to Complex Dynamics* [online]. 2015 [cit. 2022-03-27]. Dostupné z: https://web.northeastern.edu/dummit/docs/dynamics_5_introduction_to_complex_dynamics.pdf
- [16] MANDELBROT, Benoit B. *Fractals and chaos: The Mandelbrot set and beyond*. volume C. New York: Springer-Verlag, 2004, 308 s. ISBN 0-387-20158-0.
- [17] PŘIBYLOVÁ, Lenka. *M9BCF Teorie bifurkací, chaos a fraktály* [online]. Brno, 2020 [cit. 2022-03-13]. Dostupné z: <https://is.muni.cz/auth/el/sci/jaro2020/M6201/um/NelinDyn.pdf>
- [18] PŘIBYLOVÁ, Lenka. *M6201 Nelineární dynamika a její aplikace* [online]. Brno, 2020 [cit. 2022-03-13]. Dostupné z: <https://is.muni.cz/auth/el/sci/jaro2020/M6201/um/NelinDyn.pdf>
- [19] DVOŘÁKOVÁ, Jana, Marek LAMPART, Michaela MLÍCHOVÁ a Lenka OBADA-LOVÁ. *Dynamické systémy I*. Opava: Matematický ústav Slezské univerzity, 2008, 55 s.
- [20] POSPÍŠIL, Zdeněk. *Diskrétní deterministické modely* [online]. 2020, iv, 169 s. [cit. 2022-03-26]. Dostupné z: <https://is.muni.cz/auth/el/sci/podzim2020/M8230/um/DiskrModT.pdf>
- [21] FRASER, Jonathan. *An Introduction to Julia sets* [online]. 16 April 2009 [cit. 2022-03-27]. Dostupné z: http://www.gvp.cz/~vinkle/mafynet/GeoGebra/mathematika/fraktaly..../linearni_system/julia.pdf
- [22] *Fatou and Julia set* [online]. 2013, 13-22 [cit. 2022-03-30]. Dostupné z: https://webspace.maths.qmul.ac.uk/s.r.bullett/LTCCcourse/Holodyn2013notes_week2.pdf
- [23] SUTHERLAND, Scott. An Introduction to Julia and Fatou sets. *Springer Proceedings in Mathematics and Statistics* [online]. 2014, 37-60 [cit. 2022-03-30]. Dostupné z: doi:10.1007/978-3-319-08105-2_3

- [24] PEITGEN, Heinz a H. Peter RICHTER. *Beauty of fractals: images of complex dynamical systems*. Berlin: Springer-Verlag, 1986, 199 s. ISBN 3-540-15851-0.
- [25] Benoit Mandelbrot. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2022-03-15]. Dostupné z: https://en.wikipedia.org/wiki/Benoit_Mandelbrot
- [26] JOYCE, David. Is the Mandelbrot set completely symmetrical about the real axis?. In: *Quora* [příspěvek v diskuzním fóru]. 22 December 2020 [cit. 2022-03-30]. Dostupné z: <https://www.quora.com/Is-the-Mandelbrot-set-completely-symmetrical-about-the-real-axis>
- [27] Kardioida. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2022-03-13]. Dostupné z: <https://en.wikipedia.org/wiki/Cardioid?oldid=520352912>
- [28] PAUŠ, Petr. *Počítacové generování fraktálních množin*. Praha, 2004, 33 s. Dostupné také z: <http://docplayer.cz/27108444-Resersni-prace-pocitacove-generovani-fraktalnich-mnozin-petr-paus-skolitel-zamereni-katedra-akademicky-rok-rok-studia.html>. Bakalářská práce. České vysoké učení technické v Praze. Vedoucí práce Michal Beneš.
- [29] PEITGEN, Heinz-Otto a Dietmar SAUPE. *Science of fractal images*. New York: Springer-Verlag, 1988, xiii, 312 s. ISBN 0-387-96608-0.
- [30] Distance to fractals. *Inigo Quilez* [online]. 2004 [cit. 2022-03-25]. Dostupné z: <http://www.iquilezles.org/www/articles/distancefractals/distancefractals.htm>
- [31] VIRIUS, Miroslav. *Jazyky C a C++: kompletní průvodce*. 2., aktualiz. vyd. Praha: Grada, 2011, 368 s. Knihovna programátora (Grada). ISBN 978-80-247-3917-5.
- [32] FLANAGAN, David. *JavaScript: the definitive guide*. 6st ed. Sebastopol: O'Reilly, 2011. ISBN 978-0-596-80552-4.
- [33] STRATMANN, Lukas. *Conceptualization and Prototypical Implementation in WebGL of an Exercise in Color Theory*. Paderborn, 2017, viii, 66 s. Dostupné také z: <https://color.lukas-stratmann.com/downloads/thesis.pdf>. Bakalářská práce. Paderborn university, Department of Electrical Engineering, Computer Science, and Mathematics. Vedoucí práce Sabrina Heppner.

