



MATEMATICKO-FYZIKÁLNÍ FAKULTA Univerzita Karlova

DIPLOMOVÁ PRÁCE

Bc. David Weber

Fraktální geometrie pro (zdatné) amatéry

Katedra didaktiky matematiky

Vedoucí diplomové práce: RNDr. Martin Rmoutil, Ph.D.

Studijní program: Učitelství matematiky pro střední
školy

Studijní obor: Učitelství matematiky pro střední
školy se sdruženým studiem
Učitelství informatiky pro střední
školy

Praha 2025

Prohlašuji, že jsem tuto diplomovou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne
Podpis autora

Rád bych na tomto místě vyjádřil své upřímné poděkování RNDr. Martinu Rmoučkovi, Ph.D., vedoucímu mé diplomové práce, za jeho odborné vedení, vstřícný a trpělivý přístup a opravdu nemalé množství času, které mi během celé doby zpracovávání této práce věnoval. Jeho podnětné připomínky, věcná zpětná vazba a ochota vést dialog nad složitějšími částmi textu pro mne byly neocenitelné. Velmi si vážím jeho mimořádné trpělivosti a otevřenosti vůči práci, která svou povahou i rozsahem kladla zvýšené nároky na její vedení.

Velké díky patří také mé kolegyni a kamarádce MUDr. Kristině Studénkové, nejen pro její lidskou podporu, ale i za skvělou pomoc s jazykovou a gramatickou korekturou navzdory jejím ostatním náročným povinnostem.

Zvláštní poděkování si zaslouží má rodina, přítelkyně a blízcí přátelé, jejichž trvalá podpora, pochopení a trpělivost mi byly oporou v obdobích intenzivního soustředění i mnohých pochybností.

Název práce: Fraktální geometrie pro (zdatné) amatéry

Autor: Bc. David Weber

Katedra: Katedra didaktiky matematiky

Vedoucí diplomové práce: RNDr. Martin Rmoutil, Ph.D., Katedra didaktiky matematiky

Abstrakt: Tato diplomová práce se zabývá fraktální geometrií s cílem přehledně představit hlavní koncepty a principy této oblasti matematiky. Hlavními tématy jsou soběpodobnost, fraktální dimenze (zejména box-counting a Hausdorffova dimenze), teorie míry (s důrazem na Hausdorffovu a Lebesgueovu míru), L-systémy, systémy iterovaných funkcí (IFS) a Time Escape algoritmy, s nimiž souvisí především Juliovu a Fatouovy množiny.

Práce je koncipována s ohledem na různou úroveň matematické zdatnosti čtenářů. Základní principy jsou vysvětleny v kapitolách 1, 4 a 5, zatímco kapitola 2 je věnována detailnímu matematickému rozboru teorie míry a dimenzí. Praktické aspekty a algoritmy používané pro generování fraktálů jsou popsány v kapitole 5. Důraz je kladen především na obecnou srozumitelnost, která umožňuje pochopit a aplikovat dané postupy bez nutnosti znalosti konkrétního programovacího jazyka.

Důležitou součástí práce jsou ilustrace vytvořené převážně v programu Ipe a pomocí softwaru pro generování fraktálů vytvořeného k této práci, který je dostupný na platformě GitHub.

Klíčová slova: fraktální množina, Hausdorffova dimenze, Hausdorffova míra, Teorie míry, L-systém, Python, formální gramatika, algoritmus

Title: Fractal geometry for (experienced) amateurs

Author: Bc. David Weber

Department: Department of Mathematics Education

Supervisor: RNDr. Martin Rmoutil, Ph.D., Department of Mathematics Education

Abstract: This thesis focuses on fractal geometry with the aim of clearly presenting the fundamental concepts and principles of this branch of mathematics. The main topics include self-similarity, fractal dimensions (particularly box-counting and Hausdorff dimensions), measure theory (especially with emphasis on Hausdorff and Lebesgue measures), L-systems, iterated function systems (IFS), and Time Escape algorithms, particularly those related to Julia and Fatou sets.

The thesis is designed to accommodate readers of varying mathematical proficiency. Basic principles are explained in Chapters 1, 4, and 5, while Chapter 2 provides a detailed mathematical analysis of measure theory and dimensions. Practical aspects and algorithms used for fractal generation are described in Chapter 5, emphasizing general comprehensibility, allowing the reader to understand and apply the discussed methods without prior knowledge of a specific programming language.

An important part of this thesis are the illustrations created mainly in Ipe and using the accompanying fractal generation software developed for this thesis, which is available on GitHub.

Keywords: fractal set, Hausdorff dimension, Hausdorff measure, measure theory, L-system, Python, formal grammar, algorithm

Obsah

Předmluva	3
1 Úvod do fraktálů	5
1.1 Jak dlouhé je pobřeží Velké Británie?	5
1.2 Soběpodobnost	8
1.2.1 Kochova křivka	9
1.2.2 Sierpiňského trojúhelník	10
1.2.3 Kochova vločka	12
1.2.4 Cantorovo diskontinuum	14
1.3 Fraktální dimenze	16
1.3.1 Chápání konceptu dimenze	16
1.3.2 Dimenze fraktálů	19
1.3.3 Topologická dimenze	20
1.4 Co je to fraktál?	21
2 Teorie míry a dimenze	25
2.1 Základní pojmy a značení	26
2.1.1 Metrické pojmy	26
2.1.2 Limity posloupností	28
2.1.3 Limity funkcí	30
2.1.4 Bodová a stejnomořná konvergence	30
2.1.5 Topologické pojmy	31
2.1.6 Lipschitzovská zobrazení	32
2.2 Prostory s mírou	33
2.2.1 Měřitelné prostory	33
2.2.2 Míra	35
2.3 Lebesgueova míra	39
2.4 Box-counting dimenze	43
2.4.1 Definice a výpočet	43
2.4.2 Vlastnosti	51
2.5 Hausdorffova míra a Hausdorffova dimenze	55
2.5.1 Definice Hausdorffovy míry	55
2.5.2 Stručně k vlastnostem Hausdorffovy míry	60
2.5.3 Hausdorffova dimenze	61
3 Hausdorffův metrický prostor	65

3.1	Hausdorffova metrika	65
3.2	Kompaktní množiny a konvergence	67
4	Klasifikace fraktálů	71
4.1	L-systémy	71
4.1.1	Stručně k formálním jazykům a gramatikám	74
4.1.2	Definice L-systému	78
4.1.3	Želví grafika	80
4.1.4	Ukázky fraktálů a jejich L-systémů	84
4.2	Systém iterovaných funkcí	90
4.2.1	Kontrakce na Hausdorffově metrickém prostoru	91
4.2.2	Fraktály generované pomocí IFS	95
4.2.3	Další fraktály a jejich IFS	101
4.2.4	IFS a výpočet dimenze	103
4.3	Time Escape algoritmy	107
4.3.1	Juliovy a Fatouovy množiny	108
4.3.2	Mandelbrotova množina	114
5	Generování fraktálů	117
5.1	Stručně k zápisu programů	117
5.2	Implementace L-systémů a želví grafiky	118
5.2.1	L-systémy	119
5.2.2	Želví grafika	120
5.2.3	Zásobník	123
5.2.4	Vykreslení obrazce pomocí knihovny <code>tkinter</code>	123
5.3	Implementace IFS	125
5.4	Implementace Time Escape algoritmů	128
5.4.1	Aproximace vyplněné Juliovy množiny	129
5.4.2	Aproximace Juliovy množiny	131
5.4.3	Přiřazování barev	133
Závěrem		141
Seznam použité literatury		143
Seznam obrázků		145
Seznam tabulek		146
Rejstřík		147

Předmluva

Co?

Text poskytuje přehled základních aspektů fraktální geometrie, která v současné době tvoří nejen rozsáhlou, ale též poměrně oblíbenou disciplínu, a to jak v řadách matematiky znalejších jedinců (především studentů matematiky a technických oborů), tak i v rámci širší matematické veřejnosti. Lze tak nalézt velké množství literatury, která se jí zabývá z různých úhlů pohledu. Některé publikace cílí více na matematickou podstatu, jiné zase na praktické aspekty. V rámci tohoto textu se pokusíme pojmot nejen povšechnou podstatu fraktální geometrie, ale i její formální stránku.

Kdo?

Relevance jednotlivých kapitol a sekcí zavisí na konkrétním očekávání a zájmu čtenáře. Kapitola 1, věnovaná úvodnímu povídání o fraktální geometrii a výkladu základních konceptů, nepředpokládá téměř žádné předešlé znalosti problematiky. Pro již lehce znalejšího čtenáře stojí za zmínku kapitola 4 věnovaná klasifikaci základních typů fraktálů, kterou si lze také přečíst bez hlubší znalosti související teorie (ač se na ni budeme na vhodných místech odkazovat). Výjimku tvoří pouze sekce 4.1.1 a 4.2.1. Dále pak pro programátory je vhodná kapitola 5 věnovaná rozboru a implementaci algoritmů pro generování fraktálů. Z didaktických důvodů jsme zvolili jazyk Python, který i přes lekteré nedostatky představuje vhodný nástroj pro názornost implementace (především díky jeho jednoduché syntaxi). Avšak i znalci jiných jazyků si mohou přijít na své, neboť každý algoritmus uvádíme i s jeho pseudokódem. Ten sám o sobě představuje pouze abstraktní popis algoritmu, avšak bez potřeby většího úsilí jej lze převést do libovolného programovacího jazyka.

Nakonec pro náročnějšího (a matematicky zdatnějšího) čtenáře jsou připraveny kapitoly 2 a 3 zabývající se teorií míry, konceptem dimenze a Hausdorffovým metrickým prostorem, na nichž je následně vystavěna další teorie.

Znalosti?

Ač jsou kapitoly 1, 4 a 5 určeny pro méně náročné, přesto je nejspíše jasné, že některé znalosti budou přece jen třeba. Celkově se od čtenáře očekává osvojení středoškolské matematiky (především funkce, posloupnosti a základy teorie množin). Naopak kapitoly 2 a 3 předpokládají základní znalosti z oblasti matema-

tické analýzy, především metrických prostorů (shrnutí potřebné teorie lze nalézt v sekci 2.1).

Kam dál?

Jak jsme již na začátku zmínili, knih o fraktální geometrii existuje opravdu mnoho. Na poli české literatury je jednou z nejznámějších (a též kanonických) publikací kniha [22], která se zabývá především praktickými aplikacemi. Též stojí za zmínění i kniha [24] obsahující kapitolu věnovanou zmíněné problematice.

Z cizojazyčných zdrojů doporučujeme pro zdatnější čtenáře především knihu [4], kde lze nalézt mnohé informace, často nad rámec tohoto textu. Též se lze dále podívat do knih [19], [2] a [9].

Kromě základního obsahu a literatury je k textu ještě přiložen program pro generování fraktálů, v němž si čtenář může nejen prohlédnout celkovou implementaci algoritmů zmíněných v kapitole 5, ale také vyzkoušet vygenerovat i další obrazce dle své libovůle.

Odkaz: <https://github.com/D4vE0FF/Py-Fractal-Generator>

Pro více informací o programu si přečtěte soubor README.md.

(Inspirováno knihou [6].)

Kapitola 1

Úvod do fraktálů

Pod pojmem „geometrie“ si čtenář pravděpodobně vybaví rovinnou či prostorovou geometrii pracující s jednoduchými útvary jako trojúhelník, obdélník, kruh, kvádr, jehlan apod. a s útvary z nich složenými. V reálném světě tak lze nalézt mnoho uplatnění této „standardní“ geometrie, kupříkladu ve strojírenství, stavebnictví, i jinde. Často tak máme o světě představu právě ve smyslu eukleidovské geometrie. Lze však nalézt řadu objektů, pro jejichž popis jsou tyto představy limitující. Například v přírodě mrak nelze popsat jako kouli, horu nelze popsat jako jehlan a ani pobřeží nelze určitě popsat jako kružnici.

Mnohé přírodní obrazce již nelze jednoduše modelovat pomocí nástrojů „standardní“ eukleidovské geometrie, s níž jsme seznámeni již od základní školy a která byla po mnoho století základním nástrojem pro popis a porozumění matematickému prostoru. Často hraje roli i jistá nahodilost projevující se v charakteru takových obrazců. *Fraktální geometrie* se zabývá nepravidelnými a často se opakujícími vzory, které se vyskytují v přírodě i umění. Tyto vzory jsou často složité a zdánlivě chaotické, ale fraktální geometrie nám umožňuje je analyzovat a pochopit.

Vznik fraktální geometrie se datuje od roku 1975 a za jejího zakladatele je považován francouzsko-americký matematik BENOÎT MANDELBROT (1924–2010). Historicky za jejím vznikem stály objevy matematických struktur, které nezypadaly do „představ“ v rámci tehdy známé eukleidovské geometrie. Byly často považovány za „patologické“, nicméně matematici, kteří je vytvořili, je považovali za důležité kvůli ukázce bohatých možností, jež nabízí svět matematiky překračující možnosti jednoduchých struktur, s nimiž se setkávali. [8, str. 33]

1.1 Jak dlouhé je pobřeží Velké Británie?

Položme si na začátek otázku, která byla i na počátku zkoumání Benoîta Mandelbreta: *Jaká je podstata tvaru pobřeží?* Ta se stala podstatnou v jeho práci „*How Long Is the Coast of Britain?*“. Uvažujme část pobřeží s počátečním a koncovým bodem (viz obrázek 1.1). Jeho délka je zjevně zdola omezena délkou spojnice koncových bodů A a B , nicméně typické pobřeží je velmi nepravidelné a klikaté. Existují různé metody, které nám umožňují určit jeho délku přesněji. Několik



Obrázek 1.1: Příklad mapy pobřeží se spojnicí bodů A a B

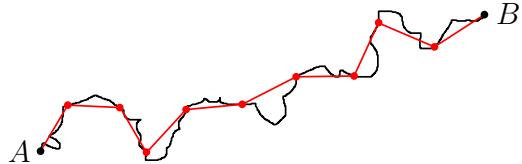
z nich je popsáno v knize [8, str. 79], pro uvedení do problematiky si zde však vystačíme s tou nejjednodušší.

Předpokládejme, že pobřeží, které zkoumáme, má pevné hranice (tj. zanedbáváme např. přílivy a odlivy nastávající během dne), a dále že jsme schopni rozlišovat libovolně krátké vzdálenosti.

Mějme zadané libovolné $\varepsilon > 0$. Podél pobřeží začneme umisťovat tyče tak, že po každém umístění provedeme na mapě krok délky **nejvýše** ε , přičemž začínáme v bodě A a postupujeme až k bodu B (popř. pokud měříme délku pobřeží ostrova, pokračujeme dokud se nevrátíme tam, kde jsme začali). Předpokládejme, že jsme provedli celkově $n(\varepsilon)$ kroků (jejich počet je závislý na zvolené délce kroku). *Přibližnou délku pobřeží* $\ell(\varepsilon)$ pak stanovíme jako

$$\ell(\varepsilon) = n(\varepsilon) \cdot \varepsilon.$$

Nyní by nás mohlo napadnout, že pro zmenšující se ε , tj. $\varepsilon \rightarrow 0_+$, bude hodnota



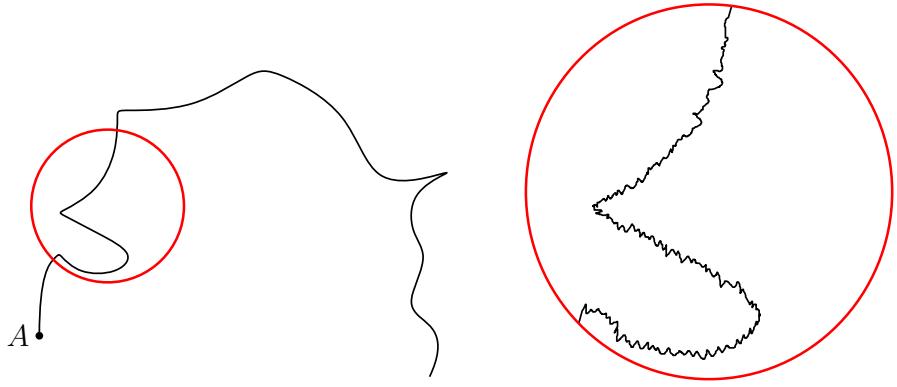
Obrázek 1.2: Odhad délky pobřeží, kde $n = 10$ při zvoleném ε

$\ell(\varepsilon)$ konvergovat ke skutečné délce pobřeží. Tzn. označíme-li skutečnou délku pobřeží L , pak bychom mohli očekávat, že platí

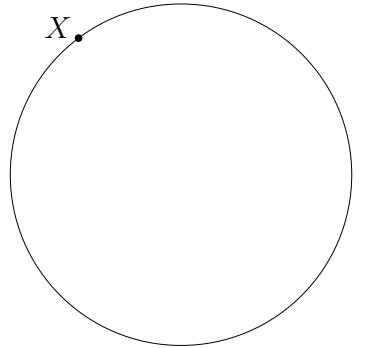
$$L = \lim_{\varepsilon \rightarrow 0_+} \ell(\varepsilon). \quad (1.1)$$

Jenže v reálné situaci bude limita (1.1) rovna ∞ . Proč? Je třeba si uvědomit, že zde pracujeme s mapou pobřeží, která má určité měřítka. Pokud bychom měli pobřeží na mapě s měřítkem 1 : 100 000, uvidíme méně detailů, než kdybychom jej zkoumali na mapě s měřítkem 1 : 1 000 (viz obrázek 1.3). Nově odhalené detaily (menší poloostrovky apod.) zde přispívají k celkové délce pobřeží $\ell(\varepsilon)$. Postupným zvětšováním měřítka mapy bychom tak odhalili další detaily. Naše původní idea tak selhává, neboť (v „klasickém“ pojetí délky) pro $\varepsilon \rightarrow 0_+$ hodnota $\ell(\varepsilon)$ patrně poroste nad všechny meze, tj. $\lim_{\varepsilon \rightarrow 0_+} \ell(\varepsilon) = \infty$.

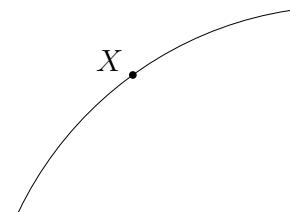
Nabízí se otázka: Proč se toto děje? Pokud se ohlédneme zpět za eukleidovskou geometrií, tento problém v ní nenastává. Např. u kružnice v eukleidovské rovině \mathbb{E}_2 změnou měřítka žádné další detaily křivky neobjevíme (podobně u jiných geometrických útvarů, viz obrázky 1.4a a 1.4b). Díky tomu můžeme v případě počítání obvodu kružnice použít následující postup.



Obrázek 1.3: Část pobřeží od bodu A v menším měřítku



(a) Kružnice v menším měřítku



(b) Část kružnice ve větším měřítku

Máme-li kružnici o poloměru $r > 0$, pak jí můžeme vepsat libovolný pravidelný n -úhelník (viz obrázek 1.5a). Obvod pravidelného n -úhelníku si označíme O_n a délku jeho strany x (viz obrázek 1.5b). Tu jsme schopni stanovit užitím elementární goniometrie, tj.

$$x = 2r \cdot \sin \frac{\pi}{n},$$

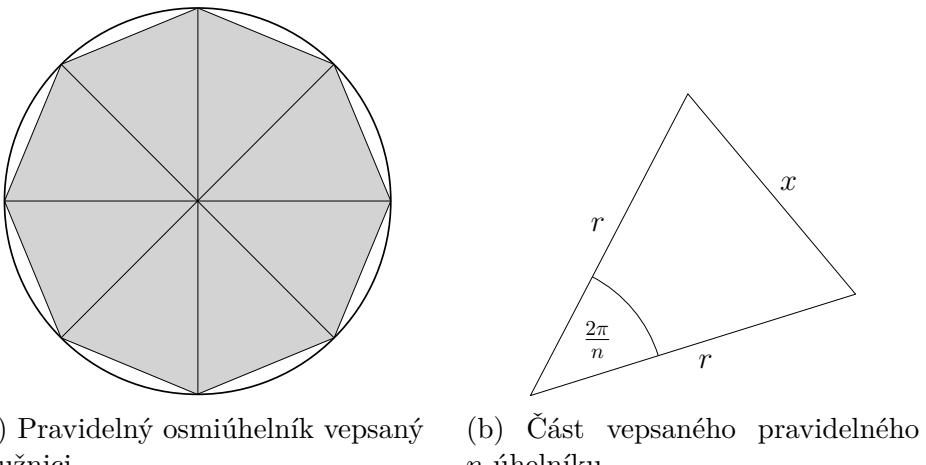
a tedy obvod

$$O_n = 2rn \cdot \sin \frac{\pi}{n}.$$

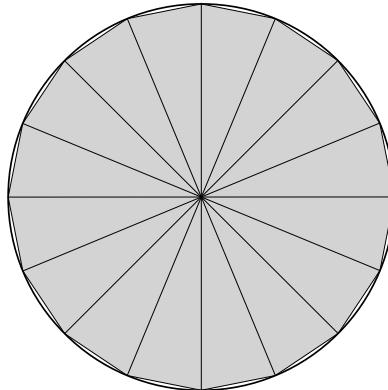
Pro rostoucí n bude obvod pravidelného n -úhelníku stále lépe approximovat obvod původní kružnice (viz obrázek 1.6). Limitním přechodem (tj. pro $n \rightarrow \infty$) tak můžeme odvodit vzorec pro obvod kružnice:

$$O = \lim_{n \rightarrow \infty} 2rn \cdot \sin \frac{\pi}{n} = 2r \cdot \lim_{n \rightarrow \infty} n \cdot \sin \frac{\pi}{n} = 2\pi r \cdot \lim_{n \rightarrow \infty} \frac{\sin \frac{\pi}{n}}{\frac{\pi}{n}} = 2\pi r.$$

Idea approximace pomocí „zjemňování“ zde skutečně funguje, a délka ve standardním pojetí tak dává smysl, jak bychom mohli očekávat. Křivka, kterou tvoří pobřeží, má však oproti kružnici jiný geometrický charakter. Délka pobřeží ∞ , k níž Mandelbrot došel, tak dává smysl z *geometrického pohledu*, avšak výsledek to není moc užitečný.



Obrázek 1.5: Princip Archimédovy metody



Obrázek 1.6: Aproximace obvodu kružnice pomocí pravidelného šestnáctiúhelníku

1.2 Soběpodobnost

Mandelbrot si uvědomil, že struktura pobřeží se charakterově vymyká útvarům tehdy známým eukleidovské geometrii, neboť mapy s různými měřítky poskytovaly různou úroveň detailů, které hrály netriviální roli v jeho celkové délce. Učinil však jiné zásadní pozorování, a to sice, že mnoho detailů má společné rysy, které se opakují. Mnohé z nich se tvarem shodovaly s výjimkou jejich měřítka. [8, str. 96]

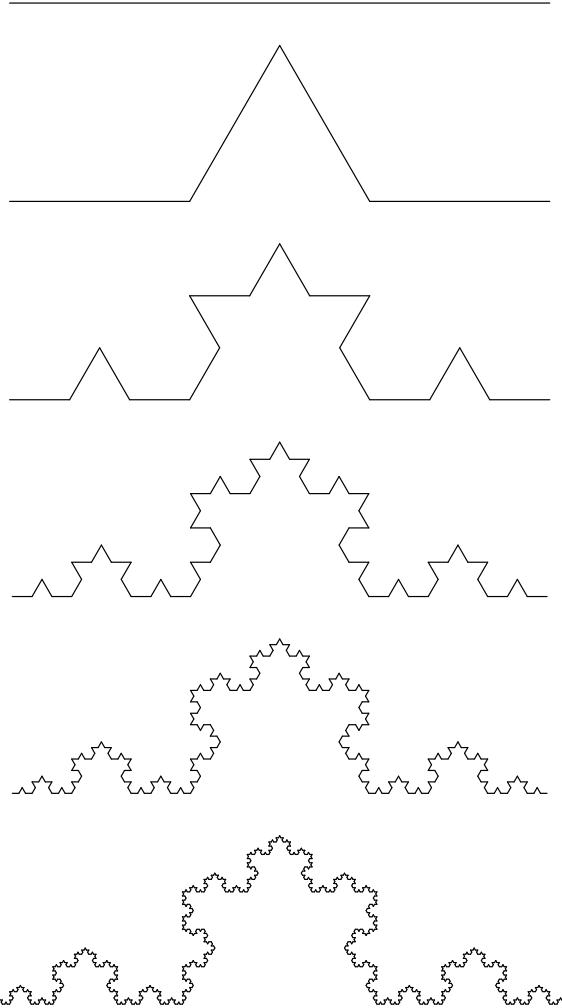
Ve fraktální geometrii se pro tento úkaz uchytil termín *soběpodobnost* (angl. *self-similarity*). Jednoduše řečeno útvar nazýváme *soběpodobným*, pokud se skládá z konečně mnoha částí podobných celku. Každá z těchto částí se tedy dá opět rozdělit na menší části, které se opět podobají celému útvaru, a takto je možno pokračovat libovolně dlouho. Zmíněná podoba může být myšlena přibližně (např. v případě pobřeží je nejspíše jasné, že žádné z jeho detailů nesdílejí společné rysy přesně), ale v dalších částech si předvedeme soběpodobnost *přímou*.

1.2.1 Kochova křivka

Jak jsme si již uvedli, za otce fraktální geometrie je považován Mandelbrot, avšak mnoho fraktálních křivek bylo známo již dříve.

Odpověď na otázku „*Co je to fraktál?*“, popř. „*Co je to fraktální charakter útvaru?*“, je netriviální úkol (jak později uvidíme) a rovnou si zde uvedlme, že jeho přesný význam se nám zde ani nepodaří přesně vymezit. K této záležitosti se ještě budeme opakovaně vracet.

Nejdříve se podíváme na jednu z nejznámějších křivek, kterou objevil roku 1904 švédský matematik HELGE VON KOCH¹, dnes známou jako *Kochova křivka*. [18, str. 61] Na začátku vezmeme úsečku délky 1. Vyjmeme prostřední (tj. druhou) třetinu a nahradíme ji dvěma úsečkami délky $1/3$, tak, aby na sebe navazovaly v krajních bodech. Tento proces následně opakujeme pro nově vzniklé úsečky. Obecně u úsečky délky l nahradíme její prostřední třetinu dvojicí úseček délky $l/3$ (viz obrázek 1.7). V první řadě si můžeme všimnout, že v každé další iteraci jsou

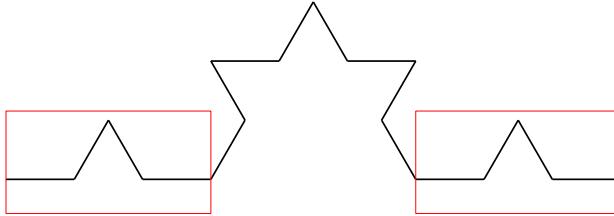


Obrázek 1.7: Prvních pět iterací Kochovy křivky

nově vzniklé úsečky podobné předešlé iteraci i původnímu celku, tedy v předešlé iteraci (viz obrázek 1.8). Pokud by tento proces pokračoval do nekonečna, pak by

¹1870–1924

každá ze čtyř částí křivky představovala **celý původní obrazec** ve zmenšeném měřítku (byla by tedy soběpodobná). Zkusme se nyní podívat na délku křivky.



Obrázek 1.8: První iterace Kochovy křivky „uvnitř“ druhé v menším měřítku

V první iteraci začínáme s úsečkou délky² 1, která se v druhé iteraci změní na křivku délky $4/3$. Není těžké si rozmyslet, že obecně v n -té iteraci bude délka křivky ℓ_n rovna

$$\left(\frac{4}{3}\right)^n.$$

Posloupnost $\{\ell_n\}_{n=1}^{\infty}$ je geometrická s kvocientem $q = 4/3 > 1$, a tedy její limita je ∞ . Kochova křivka má tak *nekonečnou délku*.

1.2.2 Sierpińského trojúhelník

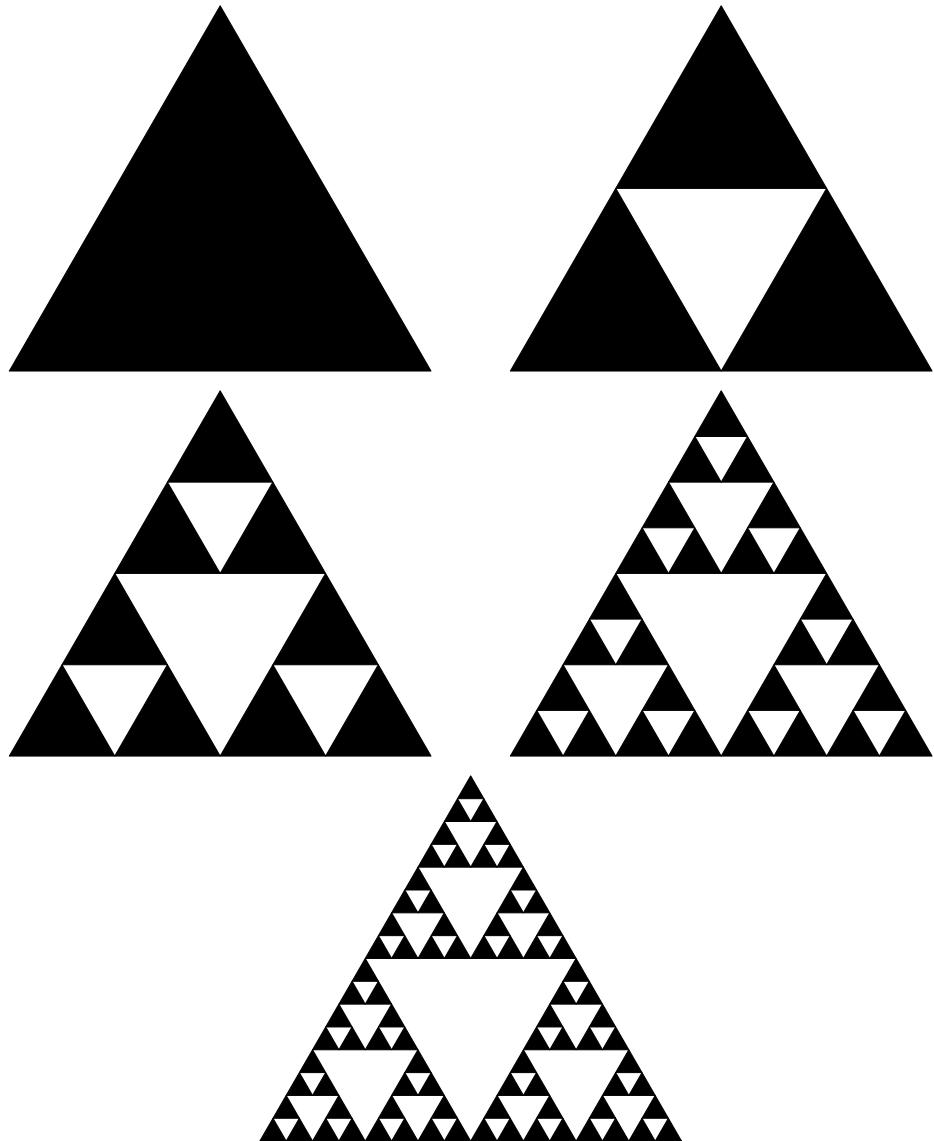
Přesuneme nyní k plošným útvaram, neboť i zde lze sledovat některé zajímavé vlastnosti. Jedním z příkladů je tzv. *Sierpińského trojúhelník*, který publikoval roku 1916 polský matematik WACŁAW SIERPIŃSKI³. [18, str. 61] Popišme si nejdříve jeho konstrukci. Na začátku (v nulté iteraci) začínáme s rovnostranným trojúhelníkem se stranou délky 1 (též lze začít s obecnou délkou ℓ_0). V něm sestojíme střední příčky (tj. spojnice středů stran trojúhelníka), které společně utvoří strany rovnostranného trojúhelníka čtvrtinového obsahu původního trojúhelníka (to vychází z faktu, že střední příčka v libovolném trojúhelníku má délku rovnou polovině délky strany, s níž je rovnoběžná). Vnitřek takto vzniklého trojúhelníka odebereme a postup opakujeme pro zbývající trojici trojúhelníků v původním obrazci (viz obrázek 1.9).

I zde si lze všimnout, že po nekonečně mnoha iteracích budou menší trojúhelníky přesnými kopími původního obrazce. Zkusme se opět podívat, jak je to s obvodem obrazce (nebo přesněji řečeno s celkovou délkou jeho hranice). Každá ze středních příček, které vzniknou v další iteraci, má poloviční délku vůči délce strany l původního trojúhelníka. Obvod obrazce se tak zvětší o $3l/2$. Počet trojúhelníků poroste exponenciálně, neboť v každé iteraci odstraněním jednoho trojúhelníku vzniknou tři nové, tj. obvod se po k -té iteraci zvětší o $3^k \cdot (1/2)^k = (3/2)^k$ (na začátku pro $k = 0$ je obvod 3). Obvod obrazce O_n po n iteracích bude roven součtu přírůstků přes všechny iterace, tj.

$$O_n = 3 + \sum_{k=1}^n \left(\frac{3}{2}\right)^k. \quad (1.2)$$

²Mohli bychom také začít s obecnou délkou ℓ_0 , ale ta by se však při výpočtu projevila pouze jako konstantní násobek.

³1882–1969



Obrázek 1.9: První čtyři iterace Sierpiňského trojúhelníka

Řada $\sum_{k=1}^n (3/2)^k$ je geometrická s kvocientem $3/2$. Zde si vzpomeňme na vzorec pro její součet.

Věta 1.2.1 (Součet geometrické řady). *Nechť je dána geometrická posloupnost $\{q^k\}_{k=1}^\infty$, kde $q \neq 1$. Pak pro součet prvních n členů platí*

$$\sum_{k=1}^n q^k = \frac{1 - q^n}{1 - q}.$$

Důkaz. Důkaz vzorce zde vynecháme, nicméně čtenář si jej může snadno ověřit např. indukcí podle n . \square

Celkově tak po aplikaci vzorce z 1.2.1 v rovnosti (1.2) dostaneme po jednoduché

úpravě

$$O_n = 3 + \sum_{k=1}^n \left(\frac{3}{2}\right)^k = 3 + \frac{3}{2} \cdot \frac{\left(\frac{3}{2}\right)^n - 1}{\frac{3}{2} - 1} = 3 + 3 \left(\left(\frac{3}{2}\right)^n - 1\right) = 3 \left(\frac{3}{2}\right)^n.$$

Posloupnost $\{O_n\}_{n=0}^\infty$ je opět geometrická s kvocientem $q = 3/2 > 1$, tzn. její limita je opět ∞ . Obvod vzniklého obrazce k k -té iteraci pro $k \rightarrow \infty$ tedy roste nade všechny meze. (Výpočet jsme si mohli též zjednodušit pozorováním, že obvod obrazce roste s faktorem $3/2$ a vzorec pro O_n jsme tak mohli určit ihned.)

Podívejme se nyní na obsah útvaru. Zde si již výpočet trochu usnadníme. Po každé iteraci se jeho obsah zmenší na $3/4$ obsahu původního obrazce. Lze snadno odvodit, že obsah rovnostranného trojúhelníka o straně délky a je

$$\frac{\sqrt{3}}{4}a^2.$$

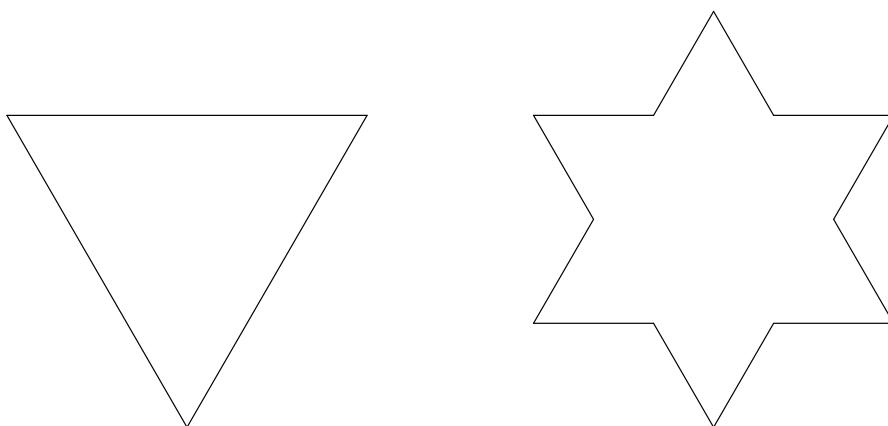
Na začátku je obsah útvaru $S_0 = \sqrt{3}/4$. Tj. celkově po n -iteracích bude obsah S_n roven

$$\frac{\sqrt{3}}{4} \left(\frac{3}{4}\right)^n. \quad (1.3)$$

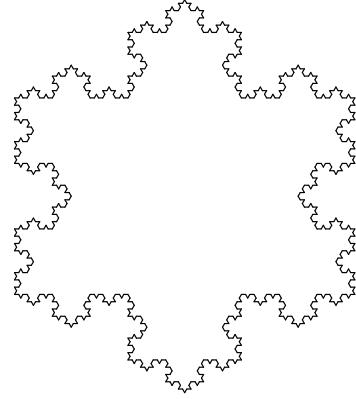
Výraz (1.3) má pro $n \rightarrow \infty$ limitu nulovou, tedy zatímco Sierpiňského trojúhelník má *nekonečný obvod*, jeho obsah je naopak *nulový*.

1.2.3 Kochova vločka

Rozšiřující variantou Kochovy křivky 1.2.1 je tzv. *Kochova vločka*, která se skládá ze tří Kochových křivek. Rozdíl je zde v tom, že začínáme s rovnostranným trojúhelníkem o straně délky 1. Na každou ze stran aplikujeme stejný proces jako předtím, tj. odebereme prostřední třetinu, nahradíme ji dvěma na sebe navazujícími úsečkami délky $1/3$ a opakujeme pro každou nově vzniklou úsečku (viz obrázky 1.10 a 1.11). Podíváme-li se na obvod, nejspíše nás nepřekvapí, že ten

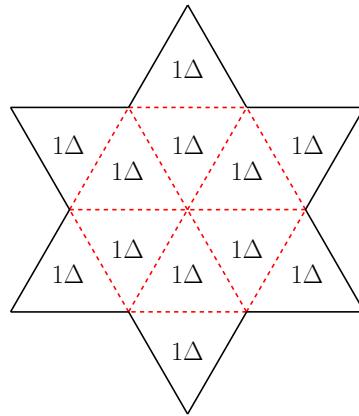


Obrázek 1.10: Nultá a první iterace Kochovy vločky



Obrázek 1.11: Čtvrtá iterace Kochovy vločky

je i zde nekonečný⁴ (už z principu, že každá strana původního rovnostranného trojúhelníka představuje samostatnou Kochovu křivku). Obsah vzniklého útvaru je však již zajímavější. Pro zjednodušení výpočtu si rozdělme útvar podle obrázku 1.12 na *stejné rovnostranné trojúhelníky*, jejichž obsah je roven $1/9$ obsahu obrazce v nulté iteraci. Výsledný obsah se tedy pokusíme vyjádřit relativně vůči



Obrázek 1.12: Rozdělení první iterace Kochovy vločky

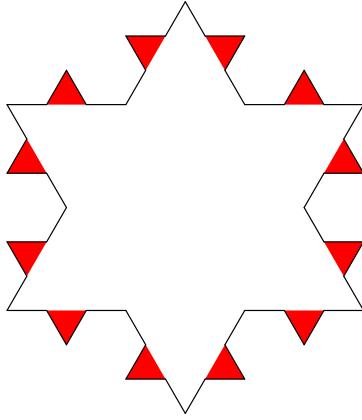
obsahům daných trojúhelníků, na něž jsme obrazec rozdělili, a pak jej pouze přepočítáme. Jejich obsah si označme Δ . Tedy obsah Sierpiňského trojúhelníka v nulté iteraci lze vyjádřit jako $S_0 = 9\Delta$. Po první iteraci vzniknou na každé ze strany 3 nové trojúhelníky o obsahu $1/9 \cdot \Delta$. V každé další iteraci vzniknou z jedné úsečky 4 nové. Obecně po n iteracích jich tedy bude $3 \cdot 4^n$. (Při výpočtu musíme počítat s o jedna nižší mocninou, neboť trojúhelníky vznikají „na úsečkách“ z předešlé iterace, nikoliv té aktuální.)

Zaměřme se nyní pouze trojúhelníky, které vznikly v aktuální iteraci (viz obrázek 1.13). Součet jejich obsahů nám dává *přírůstek obsahu* v obecné n -té iteraci. Označíme-li tento přírůstek B_n , pak platí

$$B_n = \underbrace{3 \cdot 4^{n-1}}_{\text{počet úseček}} \cdot \overbrace{\left(\frac{1}{9}\right)^{n-1} \Delta}^{\text{obsah nových troj.}} = 3 \cdot \left(\frac{4}{9}\right)^{n-1} \Delta,$$

⁴Obvod Kochovy křivky po n -té iteraci je $o_n = 3 \cdot (4/3)^n$.

kde $n \geq 1$. Nyní můžeme již vypočítat obsah v n -té iteraci S_n a posléze i celkový



Obrázek 1.13: Nově vzniklé trojúhelníky v druhé iteraci

obsah výsledného obrazce $S = \lim_{n \rightarrow \infty} S_n$. Pro $n \geq 1$ platí

$$\begin{aligned} S_n &= S_0 + \sum_{k=1}^n B_k = 9\Delta + 3\Delta \cdot \sum_{k=1}^n \left(\frac{4}{9}\right)^{k-1} = 9\Delta + 3\Delta \cdot \frac{1 - \left(\frac{4}{9}\right)^n}{1 - \frac{4}{9}} \\ &= 9\Delta + \frac{27}{5}\Delta \cdot \left(1 - \left(\frac{4}{9}\right)^n\right). \end{aligned}$$

Tedy

$$\begin{aligned} S &= \lim_{n \rightarrow \infty} S_n = 9\Delta + \frac{27}{5}\Delta \cdot \left(1 - \left(\frac{4}{9}\right)^n\right) = 9\Delta + \frac{27}{5}\Delta \cdot \left(1 - \lim_{n \rightarrow \infty} \left(\frac{4}{9}\right)^n\right) \\ &= 9\Delta + \frac{27}{5}\Delta = \frac{72}{5}\Delta. \end{aligned}$$

Obsah trojúhelníků, na než jsme rozdělili první iteraci Kochovy vločky, je

$$\Delta = \frac{\sqrt{3}}{4} \cdot \frac{1}{9},$$

a tedy celkově

$$S = \frac{72}{5}\Delta = \frac{72}{5} \cdot \frac{\sqrt{3}}{4} \cdot \frac{1}{9} = \frac{2\sqrt{3}}{5}.$$

Byl to trochu delší výpočet, nicméně jsme zjistili, že Kochova vločka má (stejně jako Kochova křivka) *nekonečnou délku (obvod)*, ale obsah má *konečný*. To bylo ostatně zjevné již od začátku, neboť celý obrazec lze obsáhnout např. v kruhu s konečným poloměrem.

1.2.4 Cantorovo diskontinuum

Podívejme se ještě na jeden typ fraktálního objektu, kterým je tzv. *Cantorovo diskontinuum*⁵ Myšlenka konstrukce je zde velmi jednoduchá: začínáme s úsečkou

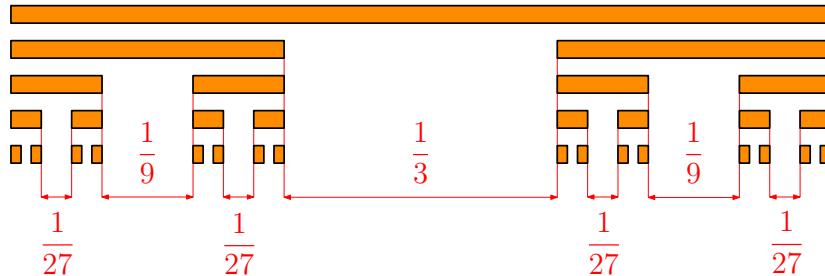
⁵Též se mu říká *Cantorova množina* (angl. „Cantor set“). Dvouozměrnou variantou je pak tzv. *Cantorův prach* (angl. „Cantor dust“). Obširněji, pokud je C Cantorovo diskontinuum, pak množina $C \times C$ je Cantorův prach. Slovo *prach* je zde miněno v přeneseném významu, neboť (podobně jako v této podsekci) lze ukázat, že vzniklé útvary mají nulový obsah.

délky 1 (nultá iterace), kterou následně rozdělíme na trojici shodných úseček a odebereme prostřední z nich, čímž vznikne první iterace. V dalších iteracích postupujeme analogicky pro zbylé úsečky (viz obrázek 1.14). Tímto nekonečným procesem vznikne množina známá jako Cantorovo diskontinuum.



Obrázek 1.14: Nultá až čtvrtá iterace Cantorova diskontinua

Nově vzniklé třetinové délky jsou kopíemi původní úsečky. Stejně jako u předešlých fraktálů nás i zde bude zajímat limitní chování tohoto procesu. Lze očekávat, že postupným odebíráním zbudou úsečky nulové délky. O tom se lze přesvědčit např. tak, že spočítáme limitu celkových délek všech odebraných úseků. V první iteraci odebereme úsečku délky $1/3$, v druhé iteraci odebereme dvě celkové délky $2 \cdot (1/3)^2 = 2/9$, ve třetí vyjmeme čtyři celkové délky $4 \cdot (1/3)^3 = 4/27$ atd. (viz obrázek 1.15). Obecně počet odebraných úseků roste s mocninou dvojkou,



Obrázek 1.15: Znázornění délky vyjmutých úseků

přičemž jejich délky klesají s mocninou trojky, tedy v n -té iteraci odebereme úsek délky $2^n(1/3)^{n+1}$. Označíme-li tedy délku zbylých úseků ℓ_n po n iteracích a $\bar{\ell}_n$ délku odebraných úseků, pak určitě platí $\ell_n = 1 - \bar{\ell}_n$. Protože nás zajímá délka všech odebraných úseků, pak po n iteracích bude celková délka rovna

$$\bar{\ell}_n = \sum_{k=0}^n 2^k \left(\frac{1}{3}\right)^{k+1} = \sum_{k=0}^n \frac{2^k}{3^{k+1}} = \frac{1}{3} \cdot \sum_{k=0}^n \left(\frac{2}{3}\right)^k = \frac{1}{3} \cdot \frac{1 - \left(\frac{2}{3}\right)^{n+1}}{1 - \frac{2}{3}} = 1 - \left(\frac{2}{3}\right)^{n+1}. \quad (1.4)$$

Označíme-li si limitu výrazů posloupnosti ℓ_n , resp. $\bar{\ell}_n$ jako ℓ , resp. $\bar{\ell}$, pak dostaneme

$$\bar{\ell} = \lim_{n \rightarrow \infty} \bar{\ell}_n = \lim_{n \rightarrow \infty} \left(1 - \left(\frac{2}{3}\right)^{n+1}\right) = 1 - \lim_{n \rightarrow \infty} \left(\frac{2}{3}\right)^{n+1} = 1 - 0 = 1.$$

Z toho již triviálně plyne, že

$$\ell = \lim_{n \rightarrow \infty} \ell_n = \lim_{n \rightarrow \infty} (1 - \bar{\ell}_n) = 1 - \lim_{n \rightarrow \infty} \bar{\ell}_n = 1 - 1 = 0, \quad (1.5)$$

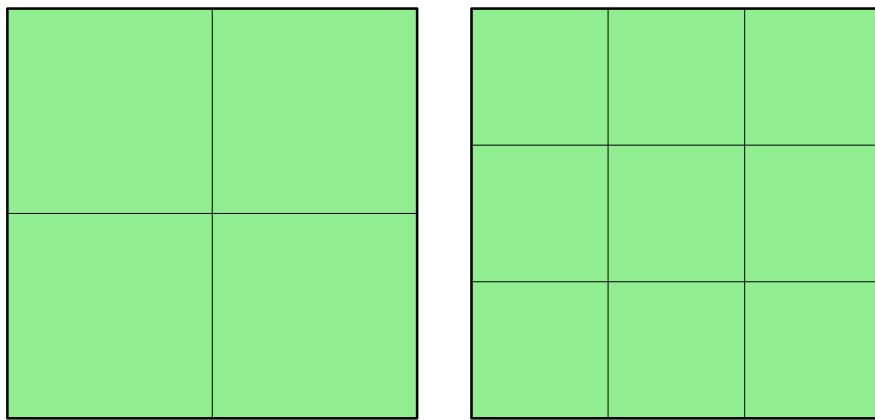
tzn. na konci procesu zbude Cantorovo diskontinuum „délky“ nula.

Všimněme si zde, že i přes to, že $\ell = 0$, je výsledná množina neprázdná. Např. krajní body původní úsečky jsou součástí výsledné množiny. Dokonce není ani příliš pracné ukázat, že Cantorovo diskontinuum je nespočetná množina, což plyne z faktu, že jeho dimenze je (jak dále uvidíme v sekci 1.3) nenulová⁶.

1.3 Fraktální dimenze

1.3.1 Chápání konceptu dimenze

Útvary uvedené v sekci 1.2 ilustrují vlastnost soběpodobnosti, kterou jsme si (zatím neformálně) popsali. V eukleidovské geometrii lze však u mnohých základních objektů pozorovat stejnou vlastnost. Např. čtverec lze určitě prohlásit v jistém smyslu za soběpodobný, neboť jej lze rozdělit na podobné útvary (viz obrázek 1.16). Podobně např. i obyčejná úsečka je taktéž soběpodobná, protože



(a) Rozdělení na čtyři menší čtverce (b) Jiná možnost rozdělení čtverce

Obrázek 1.16: Soběpodobnost čtverce

ji můžeme rozdělit na k stejných částí (viz obrázek 1.17).



Obrázek 1.17: Úsečka rozdělená na šest stejných částí

K čemu nám uvědomění takové skutečnosti vlastně je? Zmenšíme-li úsečku k -krát, pak budeme potřebovat přesně k těchto částí, abychom dostali úsečku původní délky. U čtverce (nebo obdélníku obecně) při změnění délky strany k -krát budeme potřebovat k^2 daných útváří pro obdržení čtverce s původním obsahem.⁷ Pro krychli bude situace podobná, k -krát zmenšená kopie bude potřeba k^3 -krát, abychom dostali krychli o původním objemu (viz obrázek 1.18). Lze si všimnout,

⁶Z dalšího výkladu v kapitole 2 si čtenář bude moci rozmyslet, že Haudorffova dimenze (viz sekce 2.5) spočetných množin je vždy nulová.

⁷Obdélník zmenšený k -krát bude mít strany délek $a/k, b/k$, tedy jeho obsah bude

$$\frac{ab}{k^2} = \frac{S}{k^2},$$

kde S je obsah původního obdélníka.



Obrázek 1.18: Krychle rozdělená na 27 stejných částí

že v závislosti na *dimenzi* objektu se mění daný exponent. Vztah lze tak zobecnit na

$$N(k) = k^d, \quad (1.6)$$

kde $N(k)$ je počet nových útvarů v závislosti na faktoru k a číslu d . Intuitivně je nejspíše jasné, že dimenze útvaru je takové číslo d , pro které platí rovnost (1.6).

Toto je jeden z možných způsobů, jak lze chápout koncept dimenze. Jednoduchou úpravou rovnosti (1.6) dostaneme

$$d = \log_k N(k) = \frac{\ln N(k)}{\ln k}.$$

(Obecně lze volit jakýkoliv přípustný základ logaritmů, tj $d = \log_b N(k)/\log_b k$ pro $b \in \mathbb{R}_+ \setminus \{1\}$.)

Dimenze v tomto pojetí skutečně dává dobrý smysl. Pro „klasické“ geometrické objekty vychází dimenze vždy celočíselně (viz tabulka 1.1).

Útvar	$N(k)$	$d = \ln N(k)/\ln k$
Úsečka	3	1
Čtverec	9	2
Krychle	27	3
Teserakt	81	4

Tabulka 1.1: Hodnoty dimenze d pro různé útvary

Na této myšlence je založen pojem tzv. *fraktální dimenze*. Existuje více neekvivalentních způsobů její definice. Jedna z nich, které se dálé nyní v této sekci budeme držet, se v anglicky psané literatuře nazývá „*box-counting dimension*“⁸, odkud plyne i značení \dim_B . Pro útvar F (formálně vzato množinu bodů) definujeme

$$\dim_B F = \lim_{\varepsilon \rightarrow 0_+} \frac{\ln N_\varepsilon(F)}{\ln \left(\frac{1}{\varepsilon}\right)}. \quad (1.7)$$

Výraz $1/\varepsilon$ zde představuje faktor podobnosti jako původní k (samotné ε tak hráje roli měřítka). Číslo $N_\varepsilon(F)$ je počet soběpodobných útvarů, na které jsme rozdělili

⁸Převzato z [22, str. 93] a [5, str. 28].

původní útvar F v měřítku ε . Největší rozdíl zde však představuje zkoumání „limitního chování“ daného výrazu.

Nejdříve poznamenejme, že pro „klasické“ geometrické útvary hodnota $\dim_B F$ vychází skutečně tak, jak jsme zvyklí. Toto si ilustrujeme na příkladech 1.3.1, 1.3.2 a 1.3.3.

Příklad 1.3.1 (Fraktální dimenze úsečky). Začněme asi nejednodušším příkladem výpočtu fraktální dimenze, a to u úsečky (označme ℓ). Představme si, že úsečku *jednotkové délky* rozdělíme na $N_\varepsilon(\ell) = n$ shodných dílů. Pak měřítko libovolného dílu je

$$\varepsilon = \frac{1}{n} = n^{-1}.$$

(Zde je dobré si uvědomit, že pro $n \rightarrow \infty$, tedy zjemňování dělení úsečky, platí, že $\varepsilon \rightarrow 0_+$.) Fraktální dimenzi úsečky vypočteme z definice jako

$$\dim_B \ell = \lim_{\varepsilon \rightarrow 0_+} \frac{\ln N_\varepsilon(\ell)}{\ln \left(\frac{1}{\varepsilon} \right)} = \lim_{n \rightarrow \infty} \frac{\ln n}{\ln n} = 1.$$

Příklad 1.3.2 (Fraktální dimenze čtverce). Podobně jako v příkladu 1.3.1 výše můžeme stanovit i fraktální dimenzi čtverce (označme S). Uvažujme tedy čtverec o jednotkovém obsahu, který rozdělíme $N_\varepsilon(S) = n$ shodných útvarů. Přitom víme, že obsah mění kvadraticky vůči délce strany. Měřítko nového čtverce tak bude

$$\varepsilon = \sqrt{\frac{1}{n}} = n^{-1/2}$$

a fraktální dimenze vychází

$$\dim_B S = \lim_{n \rightarrow \infty} \frac{\ln n}{\ln n^{1/2}} = \lim_{n \rightarrow \infty} \frac{\ln n}{\frac{1}{2} \ln n} = 2.$$

Pro krychli bude výpočet naprosto analogický (viz příklad 1.3.2). Obecně pro d -rozměrnou krychli bude její fraktální dimenze⁹ rovna d .

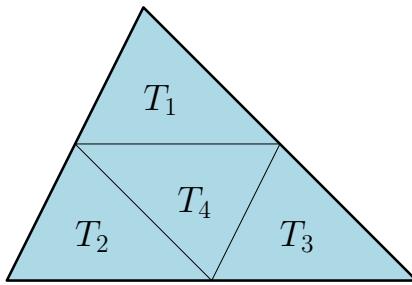
Zkusme se nyní oprostit od krychle k dalšímu útvaru.

Příklad 1.3.3 (Fraktální dimenze trojúhelníka). Podívejme se, jak to dopadne s fraktální dimenzí *obecného trojúhelníku*. Každý trojúhelník T lze rozdělit na čtvereči *vzájemně shodných trojúhelníků* T_1, \dots, T_4 , které vzniknou sestrojením středních příček v původním trojúhelníku (viz obrázek 1.19). Délka každé střední příčky odpovídá polovině délky strany, s níž je rovnoběžná, tedy obsah každého z menších trojúhelníků je *čtvrtina obsahu původního* trojúhelníka T . Tento postup můžeme opakovat pro každý z menších trojúhelníků, čímž dostaneme 4^2 trojúhelníků. Takto můžeme postupovat libovolně dlouho, přičemž po n krocích bude počet¹⁰ vzniklých trojúhelníků $N_\varepsilon(T) = 4^n$ a měřítko každého z nich bude $\varepsilon = (1/2)^n$. Fraktální dimenze tak vychází:

$$\dim_B T = \lim_{n \rightarrow \infty} \frac{\ln 4^n}{\ln 2^n} = \lim_{n \rightarrow \infty} \frac{2n \ln 2}{n \ln 2} = 2.$$

⁹Obdobnou úvahou dojdeme k měřítku $\varepsilon = n^{-1/d}$.

¹⁰Výpočet bychom mohli i zde provést ve stejném duchu jako u úsečky, čtverce nebo krychle. Počet částí, na něž rozdělíme trojúhelník, označíme $N_\varepsilon(T) = n$, přičemž měřítko pak bude $\varepsilon = n^{-1/2}$.



Obrázek 1.19: Trojúhelník T rozdělený na trojúhelníky T_1, \dots, T_4

Pro jednoduché útvary vychází dimenze tak, jak bychom mohli očekávat. K zajímavějším výsledkům však dospějeme u fraktálů, na něž se blíže podíváme v následující podsekci 1.3.2.

1.3.2 Dimenze fraktálů

Co kdybychom však zkusili podobnou myšlenku aplikovat i na *fraktální objekty* jako např. Kochovu křivku, Kochovu vločku, Sierpiňského trojúhelníku nebo Cantorovo diskontinuum? Zkusme to. Pro připomenutí jednotlivých křivek a výsledků k nim si dovolujeme čtenáře opětovně odkázat na sekci 1.2, kde jsou podrobněji rozebrány.

V tomto případě uvidíme, že dochází na první pohled k docela zvláštnímu jevu. Fraktální dimenze již totiž nemusí vycházet celočíselně, jak jsme zvyklí.

- **Kochova křivka F_{KC} .** V každé iteraci nahrazujeme každou úsečku čtyřmi novými. Kompletní Kochova křivka tak obsahuje právě čtyři kopie sebe sama zmenšené na třetinu, tj. v n -té iteraci je $N_\varepsilon(F_{KC}) = 4^n$, jak jsme již odvodili (viz podsekce 1.2.1).¹¹ Měřítko nově vzniklých částí je tak $\varepsilon = (1/3)^n$.

$$\dim_B F_{KC} = \lim_{\varepsilon \rightarrow 0+} \frac{\ln N_\varepsilon(F_{KC})}{\ln(\frac{1}{\varepsilon})} = \lim_{n \rightarrow \infty} \frac{\ln 4^n}{\ln 3^n} = \frac{\ln 4}{\ln 3} \approx 1,2618595\dots \quad (1.8)$$

- **Kochova vločka F_{KS} .** Začínáme s rovnostranným trojúhelníkem o straně délky 1, na jehož stranách postupně vznikne Kochova křivka. V n -té iteraci je obvod Kochovy vločky o_n roven $3 \cdot 4^n$, tj. i $N_\varepsilon(F_{KS}) = 3 \cdot 4^n$, kde měřítko¹² nově vzniklých úseček je $\varepsilon = 1/3 \cdot (1/3)^n = (1/3)^{n+1}$. Není těžké se

¹¹Lze však zvolit i jiné dělení. Např. lze na Kochovu křivku nahlížet tak, že obsahuje 16 kopií sebe sama zmenšených na devítinu.

¹²Měřítko se ve srovnání s Kochovou křivkou liší v mocnině, neboť délku nových úseků porovnáváme s obvodem celého trojúhelníka, nikoliv pouze délkom jedné jeho strany. Nicméně ve výpočtu bychom se mohli omezit i jen na jednu ze stran, výpočet by byl tak zcela identický jako u Kochovy křivky.

přesvědčit, že fraktální dimenze vychází stejně jako u Kochovy křivky:

$$\dim_B F_{KS} = \lim_{n \rightarrow \infty} \frac{\ln 3 \cdot 4^n}{\ln 3^{n+1}} = \lim_{n \rightarrow \infty} \frac{\ln 4^n \overbrace{\left(1 + \frac{\ln 3}{\ln 4^n}\right)}^{\rightarrow 1}}{\ln 3^n \overbrace{\left(1 + \frac{\ln 3}{\ln 3^n}\right)}^{\rightarrow 1}} = \frac{\ln 4}{\ln 3}. \quad (1.9)$$

- **Sierpińského trojúhelník F_{ST} .** V každé iteraci vynecháme prostřední trojúhelník, čímž vznikne *trojice* nových trojúhelníků s *polovičním* měřítkem. Tzn. $N_\varepsilon(F_{ST}) = 3^n$ pro $\varepsilon = (1/2)^n$, a tedy

$$\dim_B F_{ST} = \lim_{n \rightarrow \infty} \frac{\ln 3^n}{\ln 2^n} = \frac{\ln 3}{\ln 2} \approx 1,5849625\dots \quad (1.10)$$

- **Cantorovo diskontinuum F_{CD} .** Vždy vyjmeme prostřední třetinu úsečky, čímž obdržíme *dvojici* úseček *třetinové* délky, tj. $N_\varepsilon(F_{CD}) = 2^n$ pro $\varepsilon = (1/3)^n$. Fraktální dimenze tak vychází

$$\dim_B F_{CD} = \lim_{n \rightarrow \infty} \frac{\ln 2^n}{\ln 3^n} = \frac{\ln 2}{\ln 3} \approx 0,6309297\dots \quad (1.11)$$

Udělejme si nyní menší souhrn a porovnání dosud získaných výsledků (viz tabulka 1.2). Můžeme si všimnout, že zatímco u „klasických“ objektů vychází frak-

Útvar F	ε	$N_\varepsilon(F)$	$\dim_B F$
Úsečka	n^{-1}	n	1
Čtverec	$n^{-1/2}$	n	2
Krychle	$n^{-1/3}$	n	3
Teserakt	$n^{-1/4}$	n	4
d -rozměrná krychle	$n^{-1/d}$	n	d
Obecný trojúhelník	$(1/2)^n$	4^n	2
Kochova křivka	$(1/3)^n$	4^n	$1,2618595\dots$
Kochova vločka	$(1/3)^{n+1}$	$3 \cdot 4^n$	$1,2618595\dots$
Sierpińského trojúhelník	$(1/2)^n$	3^n	$1,5849625\dots$
Cantorovo diskontinuum	$(1/3)^n$	2^n	$0,6309297\dots$

Tabulka 1.2: Porovnání fraktálních dimenzí d_k různých objektů

tální dimenze *celočíselná*, u (zmíněných) fraktálů vychází *neceločíselně*, ba dokonce i iracionálně.

1.3.3 Topologická dimenze

Výsledky z předešlé části 1.3.2 se můžou zdát poněkud překvapující. Jak je vůbec možné, že dimenze nemusí vycházet nutně celočíselná? Ač se to možná zdá jako nesmyslný výsledek, je třeba si uvědomit, jak vlastně koncept dimenze chápeme. Na jednu stranu na ni lze nahlížet jako na mocninu „s níž se zvyšuje“ obsah/objem

tělesa, ménime-li měřítko. Naopak čtenář znalý lineární algebry si možná vzpomene, že v této matematické disciplíně se na dimenzi nahlíží jako na *mohutnost libovolné báze daného vektorového prostoru*. Ta naopak vychází vždy pouze celočíselně, případně nekonečná, avšak nelze s ní dobře zachytit hlubší detail geometrie u objektů, jako jsou právě fraktály.

Další možné pojetí pojmu dimenze nám dává tzv. *topologická dimenze*. Ta totiž daleko více odpovídají našemu intuitivnímu chápání tohoto pojmu, neboť se vždy jedná o celé číslo, jak ho známe ze školní geometrie. Existuje více topologických dimenzí¹³, které co do definice nejsou ekvivalentní, ačkoliv ve většině standardních případů splývají. My se zde pro ilustraci podíváme na tzv. *Lebesgueovu pokrývací dimenzi* (dále jen již „topologickou dimenzi“) pojmenovanou po francouzském matematikovi HENRI LEBESGUEOVĚ¹⁴. Myšlenka definice je založena na pokrývání objektu (formálně vzato *množiny bodů*) tzv. *otevřenými množinami*.¹⁵ Formální definici si zde v rámci zachování jednoduchosti odustíme, avšak pro hlubší matematický základ si dovolíme čtenáře odkázat např. na knihu [3]. Avšak pokusíme se ji alespoň nastínit.

Množina X má topologickou dimenzi $\dim_L X = n$, pokud n je nejmenší číslo takové, že pro každé pokrytí otevřenými množinami¹⁶ \mathcal{U} existuje zjemnění¹⁷ \mathcal{U}' takové, že každý bod $x \in X$ naleží nejvýše $n + 1$ množin pokrytí \mathcal{U}' .

Tuto ideu si zkusíme přiblížit na příkladu topologické dimenze úsečky (viz obrázek 1.20). Pro libovolné pokrytí lze ukázat, že každý bod je obsažen maximálně ve dvou množinách vhodně zvoleného zjemnění, tedy topologická dimenze úsečky je 1. Podobně např. pro čtverec lze dojít k závěru, že pro každé pokrytí existuje zjemnění takové, že každý bod je obsažen maximálně ve třech množinách, tedy jeho topologická dimenze je 2, jak bychom očekávali (viz obrázek 1.21). Porovnejme nyní topologickou dimenzi vůči dimenzi fraktální. Jak jsme se již přesvědčili v příkladech 1.3.1, 1.3.2 a 1.3.3, pro „standardní“ útvary je fraktální dimenze celočíselná (ač jsou i další, které jsme neuvedli), zatímco v podsekci 1.3.2 jsme zjistili, že u fraktální dimenze fraktálů tomu tak být nemusí. Přitom však topologická dimenze fraktálních útvarů je (a dokonce musí být) celočíselná (viz tabulka 1.3). Fraktální dimenze tak oproti té topologické daleko lépe zachycuje informaci o detailní geometrii daných objektů.

1.4 Co je to fraktál?

Tak co je to tedy ten „fraktál“? Odpovědi na tuto otázku jsme se poměrně dlouhou dobu vyhýbali a onen termín, popř. jeho přídavnou variantu „fraktální“, jsme používali čistě na intuitivní úrovni. Ač jsme se zatím obešli bez jeho formálnějšího

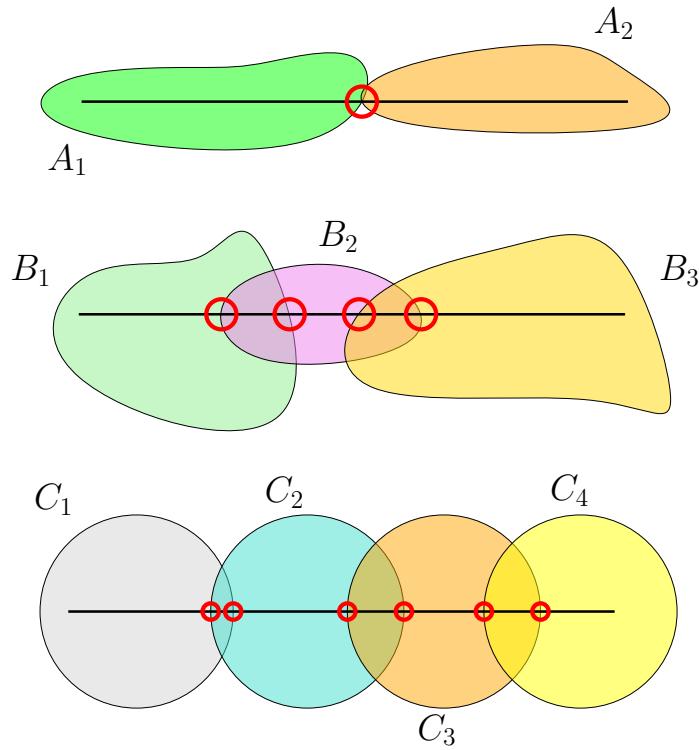
¹³Jiným příkladem takové dimenze je *induktivní dimenze*.

¹⁴1875–1941

¹⁵Otevřená množina je zobecnění pojmu otevřeného intervalu reálných čísel. Neformálně řečeno je to taková množina X , kde pro každý její bod $x \in X$ patří do této množiny i nějaké ε -okolí tohoto bodu (patří do ní i body, které jsou „dostatečně blízko“).

¹⁶Formálněji to znamená, že A_1, \dots, A_n jsou otevřené množiny takové, že platí $X \subseteq \bigcup_{i=1}^n A_i$.

¹⁷Zjemněním pokrytí \mathcal{U} nazýváme takové pokrytí \mathcal{U}' množiny X , kde každá množina $A'_i \in \mathcal{U}'$ je podmnožinou nějaké množiny A_j původního pokrytí \mathcal{U} .

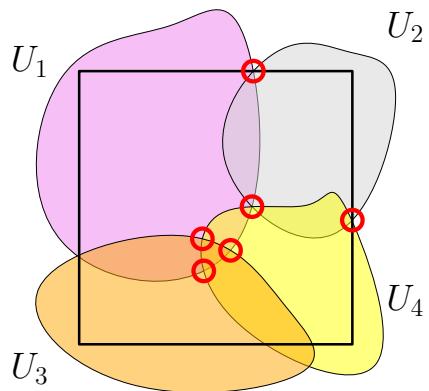


Obrázek 1.20: Různé možnosti (pod)pokrytí úsečky

upřesnění, bylo by možná přinejmenším slušné se o to alespoň pokusit. V předešlé sekci 1.3 jsme si (alespoň částečně) ujasnili *fraktální* a *topologickou dimenzi*, které jsme následně použili na příkladech konkrétních útvarů (konkrétně viz tabulky 1.2 a 1.3).

Již jsme si všimli, že u fraktálních útvarů vychází fraktální dimenze \dim_H neceločíselně, zatímco jejich topologická dimenze \dim_L je vždy celočíselná. To by se mohlo zdát jako dobrá charakteristika fraktálů. Avšak existují útvary, jejichž fraktální a topologická dimenze se shodují, přestože také mají „fraktální charakter“. Pro příklad nemusíme chodit nikterak daleko, pravděpodobně nejznámějším útvarem je v tomto ohledu *Mandelbrotova množina*, jejíž fraktální i topologická dimenze je rovna 2 (blíže se na ni podíváme v podsekci 4.3.2). Jiná definice zase naopak popisuje fraktál jako útvar, jehož Hausdorffova dimenze (na tu se blíže podíváme v sekci 2.5) je ostře větší než dimenze topologická. To bychom však ale nemohli považovat za fraktál např. již zmíněný Sierpiňského trojúhelník. Problém (a také důvod, proč jsme se definici toho pojmu vyhýbali) je však zkrátka ten, že dodnes **není známá** žádná univerzální definice fraktálu. [24, str. 226]

Je to možná trochu zklamání, nicméně dobrou zprávou je, že ani pro další výklad ona absence formální definice fraktálu nebude překážkou. V dalším textu se zaměříme (mj.) především na klasifikaci fraktálů (viz kapitola 4) a další jejich vlastnosti.



Obrázek 1.21: Možné (pod)pokrytí čtverce

Útvar F	$\dim_B F$	$\dim_L F$
Úsečka	1	1
Čtverec	2	2
Krychle	3	3
Teserakt	4	4
d -rozměrná krychle	d	d
Kochova křivka	1,2618595...	1
Kochova vločka	1,2618595...	1
Sierpiňského trojúhelník	1,5849625...	2
Cantorovo diskontinuum	0,6309297...	0

Tabulka 1.3: Porovnání fraktální a topologické dimenze útvarů

Kapitola 2

Teorie míry a dimenze

V této kapitole se budeme nyní věnovat fraktálům a jim příbuzným záležitostem trochu formálněji. Do této chvíle jsme si již představili některé základní fraktály, jako je např. *Sierpiňského trojúhelník*, *Kochova vločka* nebo *Cantorovo diskontinuum*, na nichž jsme si ilustrovali především myšlenku soběpodobnosti a na to navazující pojetí dimenze (viz kapitola 1, sekce 1.2 a 1.3).

Ačkoliv leckterý čtenář by se s poskytnutým vysvětlením asi spokojil, jiný by mohl namítat, že jsme řadu věcí vynechali. A měl by jistě pravdu. Proto se v této kapitole budeme věnovat některým záležitostem z tzv. *teorie míry*, která je v tomto ohledu klíčová a poskytne nám nástroje pro měření fraktálních útvarů, jejichž geometrie často přesahuje možnosti klasické „eukleidovské analýzy“. *Míra* pro nás představuje zobecnění pojmu jako je *délka*, *obsah* a *objem*, které známe ze školní výuky geometrie. Na jejím základě pak budeme schopni detailněji prozkoumat fraktální dimenzi, kterou jsme již v základu předestřeli v předešlé kapitole. Jmenovitě se budeme zabývat

- *měřitelnými prostory* a *prostory s mírou* obecně,
- *Lebesgueovou mírou*,
- *box-counting dimenzi*¹
- *Hausdorffovou mírou* a z ní vycházející *Hausdorffovou dimenzi*.

Ačkoliv je toto téma jinak velice obsáhlé a jsou mu věnované samostatné texty i knihy, spokojíme se pouze s naprostým základem. Pro podrobnější výklad si dovolíme čtenáře odkázat na knihy [5], [7], [12] a [2].

Než se však pustíme do problematiky samotné, je důležité zmínit, že pro rigorózní budování příslušné teorie budeme potřebovat některé základní znalosti. V dalším textu předpokladáme, že je s nimi čtenář obeznámen. Na začátek si zde pro úplnost dovolíme začít soupisem pojmu a značení, které budeme dále potřebovat. Některé získané znalosti využijeme ještě v kapitole 3 o Hausdorffově metrickém prostoru.

¹Též ji lze nalézt pod názvem *Minkowského dimenze* nebo *Minkowského-Bouligandova dimenze*. Je pojmenována po polském matematikovi HERMANNOVÍ MINKOWSKÉM (1864–1909) a francouzském matematikovi GEORGESOVÍ BOULIGANDOVÍ (1889–1979).

2.1 Základní pojmy a značení

V tomto oddílu se v krátkosti zaměříme na připomenutí některých pojmu a značení, které budeme dále využívat. Související teorii v tomto případě vynecháme s předpokladem, že ji čtenář již zná. Pokud tomu však v některých případech takto nebude, lze tuto část textu považovat za výčet konceptů, které pro zvládnutí nadcházející teorie budeme potřebovat. V tomto ohledu lze např. nahlédnout do knihy [11].

Tuto část tedy vnímaje spíše jako referenční, než rigorózní výklad.

Výklad v této kapitole a dále v kapitole 3 se bude týkat především tzv. *metrických prostorů* (viz definice 2.1.1).

Definice 2.1.1 (Metrický prostor). *Metrickým prostorem* nazýváme uspořádanou dvojici (X, ϱ) , kde $X \neq \emptyset$ a $\varrho : X \times X \rightarrow \langle 0, \infty \rangle$ je zobrazení splňující:

- (a) $\forall x, y \in X : \varrho(x, y) = 0 \iff x = y$,
- (b) $\forall x, y \in X : \varrho(x, y) = \varrho(y, x)$, \triangleleft symetrie
- (c) $\forall x, y, z \in X : \varrho(x, z) \leq \varrho(x, y) + \varrho(y, z)$. \triangleleft trojúhelníková nerovnost

Zobrazení ϱ nazýváme *metrika*.

Zobrazení ϱ v definici 2.1.1 pro nás představuje způsob měření vzdálenosti mezi prvky množiny X .

2.1.1 Metrické pojmy

Začneme některými základními pojmy sousvisejícími s metrickými prostory.

Definice 2.1.2 (Otevřená/uzavřená koule). Nechť (X, ϱ) je metrický prostor. Pak definujeme

- otevřenou koulí se středem v bodě $x \in X$ o poloměru $r > 0$

$$B_r(x) = \{y \in X \mid \varrho(y, x) < r\},$$

- resp. uzavřenou koulí se středem v bodě $x \in X$ o poloměru $r > 0$

$$K_r(x) = \{y \in X \mid \varrho(y, x) \leq r\}.$$

Např. v \mathbb{R} představuje otevřená koule $B_r(x)$ otevřený interval $(x - r, x + r)$ a $K_r(x) = \langle x - r, x + r \rangle$. V \mathbb{R}^2 s eukleidovskou metrikou se jedná o standardní kruh. Ještě jiným příkladem může být prostor všech spojitých funkcí na zadaném intervalu $\langle a, b \rangle$, značíme $X = \mathcal{C}(\langle a, b \rangle)$, kde metriku definujeme vzorcem

$$\varrho(f, g) = \sup_{x \in \langle a, b \rangle} |f(x) - g(x)|.$$

Pak pro libovolnou funkci $f \in X$ představuje $B_r(f)$ a $K_r(f)$ pás šířky $2r$ sestřelený okolo grafu funkce f .

S otevřenými, resp. uzavřenými koulemi se pojí další terminologie.

Definice 2.1.3 (Otevřená, uzavřená a omezená množina). Množina M v metrickém prostoru (X, ϱ) se nazývá

- *otevřená*, pokud pro každé $x \in M$ existuje $r > 0$ takové, že $B_r(x) \subseteq M$.
- *uzavřená*, pokud její doplněk $X \setminus M$ je otevřená množina.
- *omezená*, pokud existuje $x \in X$ a $r > 0$ takové, že $M \subseteq B_r(x)$.

Uzavřenosť a otevřenosť množiny závisí na volbě konkrétního metrického prostoru. Např. interval (a, b) je v $X = \mathbb{R}$ otevřená množina, avšak v $X = (a, b)$ je to otevřená i uzavřená množina, neboť její doplněk \emptyset je otevřený.

Definice 2.1.4 (Průměr množiny). Nechť (X, ϱ) je metrický prostor. *Průměr množiny* $M \subseteq X$, $M \neq \emptyset$ definujeme jako

$$\text{diam } M = \sup \{\varrho(x, y) \mid x, y \in M\}.$$

Není těžké dokázat, že v \mathbb{R}^n je

$$\text{diam } B_r(x) = \text{diam } K_r(x) = 2r.$$

Obecně to však neplatí. Uvažme metrický prostor (X, ϱ) , kde je metrika ϱ

$$\varrho(x, y) = \begin{cases} 0 & x = y, \\ 1 & x \neq y. \end{cases}$$

Jedná se o takzvaný *diskrétní metrický prostor* (o tom, že ϱ je metrika, se lze snadno přesvědčit). Pro $x \in X$ a $r > 0$ platí, že

$$\text{diam } B_r(x) = \begin{cases} 0 & r \leq 1, \\ 1 & r > 1. \end{cases}$$

Koule $B_r(x)$ v diskrétním metrickém prostoru totiž představuje buď singleton $\{x\}$ pro $r \leq 1$, nebo celý prostor X . Tedy poloměr koule $B_r(x)$ může být větší než její průměr.

Definice 2.1.5 (Vzdálenost bodu od množiny, vzdálenost množin). Nechť (X, ϱ) je metrický prostor.

- *Vzdáleností bodu* $x \in X$ od množiny $M \subseteq X$ rozumíme

$$\varrho(x, M) = \inf \{\varrho(x, y) \mid y \in M\}.$$

- *Vzdáleností množin* $M, N \subseteq X$ rozumíme

$$\varrho(M, N) = \inf \{\varrho(x, y) \mid x \in M, y \in N\}.$$

Definice 2.1.6 (Vnitřek, hranice a uzávěr množiny). Nechť (X, ϱ) je metrický prostor a $M \subseteq X$.

- *Vnitřek množiny* M je množina

$$M^\circ = \{x \in X \mid \exists r > 0 : B_r(x) \subseteq M\}.$$

- *Hranice množiny* M je množina

$$\partial M = \{x \in X \mid \forall r > 0 \exists y, z \in B_r(x) : y \in X \wedge z \in X \setminus M\}.$$

- *Uzávěr množiny* M je množina

$$\overline{M} = M \cup \partial M.$$

V textu též budeme hodně pracovat s pojmem δ -okolí.

Definice 2.1.7 (δ -okolí). Nechť (X, ϱ) je metrický prostor a $M \subseteq X$. Pak δ -okolím množiny M rozumíme množinu

$$(M)_\delta = \{x \in X \mid \varrho(x, M) < \delta\}.$$

Zde je zvlášť vhodné zmínit, že $(M)_\delta$ je v prostoru \mathbb{R}^n , s nímž budeme často pracovat, otevřená množina, je-li M neprázdná.

Jako poslední zde uvedeme termíny, které využijeme zejména v sekci věnující se Lebesgueově míře a posléze box-counting dimenzi (viz sekce 2.3 a 2.4).

Definice 2.1.8 (Kvádr). Kvádrem v \mathbb{R}^n nazveme množinu

$$I = \prod_{i=1}^n \langle a_i, b_i \rangle,$$

kde $a_i, b_i \in \mathbb{R}$ a $a_i \leq b_i$ pro každé $i \in \mathbb{N}$. Objem kvádru I definujeme jako

$$\text{vol}_n(I) = \prod_{i=1}^n (b_i - a_i).$$

Definice 2.1.9 (δ -mříž). δ -mříž v \mathbb{R}^n názýváme množinu

$$\mathcal{Q}_\delta = \left\{ \prod_{i=1}^n \langle m_j \delta, (m_j + 1) \delta \rangle \mid m_1, \dots, m_n \in \mathbb{Z} \right\}.$$

Množinu \mathcal{Q}_δ si můžeme představit jako rozdělení \mathbb{R}^n na krychle o straně délky δ dotýkajících se pouze na hranici.

2.1.2 Limity posloupnosti

S limitou posloupnosti a funkce jedné proměnné je čtenář nejspíše dobře seznámen. V kontextu metrických prostorů definujeme pojem limity posloupnosti následovně.

Definice 2.1.10 (Limita posloupnosti). Mějme metrický prostor (X, ϱ) a posloupnost $\{x_n\}_{n=1}^{\infty}$, kde $x_i \in X$ pro každé $i \in \mathbb{N}$. Pak posloupnost $\{x_n\}_{n=1}^{\infty}$ má limitu $x \in X$, píšeme

$$\lim_{n \rightarrow \infty} x_n = x,$$

nebo též $x_n \rightarrow x$, pokud

$$\forall \varepsilon > 0 \exists n_0 \in \mathbb{N} \forall n \geq n_0 : \varrho(x_n, x) < \varepsilon, \quad \text{tj.} \quad \lim_{n \rightarrow \infty} \varrho(x_n, x) = 0.$$

Limita posloupnosti je vždy určena jednoznačně (pokud existuje). Ve spojitosti s limitami pro nás bude dále relevantní i tzv. *limes superior* a *limes inferior*.

Definice 2.1.11 (Limes superior, limes inferior). Nechť je dána posloupnost $\{x_n\}_{n=1}^{\infty}$, kde $x_n \in \mathbb{R}$ pro každé $n \in \mathbb{N}$. Pak definujeme:

- limes superior posloupnosti $\{x_n\}_{n=1}^{\infty}$ jako

$$\limsup_{n \rightarrow \infty} x_n = \lim_{n \rightarrow \infty} (\sup \{x_k \mid k \geq n\}),$$

- limes inferior posloupnosti $\{x_n\}_{n=1}^{\infty}$ jako

$$\liminf_{n \rightarrow \infty} x_n = \lim_{n \rightarrow \infty} (\inf \{x_k \mid k \geq n\}).$$

Jinak lze definovat limes superior, resp. limes inferior jako supremum, resp. infimum hromadných bodů posloupnosti. Speciálně platí, že posloupnost $\{x_n\}_{n=1}^{\infty}$ má limitu L , právě tehdy, když

$$\limsup_{n \rightarrow \infty} x_n = \liminf_{n \rightarrow \infty} x_n = L.$$

Čtenář již nejspíše slyšel i o tzv. *Bolzanově-Cauchyově podmínce*. Posloupnosti ji splňující nazýváme *cauchyovské*.

Definice 2.1.12 (Cauchyovská posloupnost). Nechť (X, ϱ) je metrický prostor. Posloupnost $\{x_n\}_{n=1}^{\infty}$, kde $x_i \in X$ pro každé $i \in N$, nazveme *cauchyovskou*, pokud splňuje:

$$\forall \varepsilon > 0 \exists n_0 \in \mathbb{N} \forall n, m \geq n_0 : \varrho(x_n, x_m) < \varepsilon.$$

Připomeňme rovněž, že Bolzanova-Cauchyova podmínka je nutná, nikoliv však postačující pro konvergenci posloupnosti v X . Uvažme např. prostor $(\mathbb{Q}, |\cdot|)$ se standardní eukleidovskou metrikou. Posloupnost $\{x_n\}_{n=1}^{\infty}$ definujeme následovně:

$$x_0 > 0 \quad \text{a} \quad x_n = \frac{1}{2} \left(x_{n-1} + \frac{2}{x_{n-1}} \right), \quad n \in \mathbb{N}.$$

Všechny její členy jsou v \mathbb{Q} a lze též ověřit, že je cauchyovská (např. proto, že má limitu v \mathbb{R}). Nicméně pro její limitu platí

$$\lim_{n \rightarrow \infty} x_n = \sqrt{2} \notin \mathbb{Q}.$$

V rámci metrických prostorů tedy mnohdy dává smysl omezovat se pouze na takové prostory, kde je Bolzanova-Cauchyova podmínka nutná i postačující pro konvergenci. Též jim říkáme úplné.

Definice 2.1.13 (Úplný metrický prostor). Metrický prostor (X, ϱ) se nazývá *úplný*, pokud každá cauchyovská posloupnost v X má limitu v X .

Příklady úplných metrických prostorů jsou např. uzavřený interval $\langle 0, 1 \rangle$ se standardní eukleidovskou metrikou, nebo prostor $\mathcal{C}(\langle 0, 1 \rangle)$ s již zmíněnou supremovou metrikou.

2.1.3 Limity funkcí

Od posloupností se přesuneme k limitám funkcí.

Definice 2.1.14 (Limita funkce/zobrazení v bodě). Nechť $(X, \varrho_1), (Y, \varrho_2)$ jsou metrické prostory a $f : X \rightarrow Y$ je zobrazení. Řekneme, že f má limitu $y \in Y$ v bodě $x_0 \in X$, píšeme

$$\lim_{x \rightarrow x_0} f(x) = y,$$

pokud

$$\forall \varepsilon > 0 \exists \delta > 0 \forall x \in X : x \in B_\delta(x_0) \setminus \{x_0\} \implies f(x) \in B_\varepsilon(y).$$

Navíc pokud platí

$$\lim_{x \rightarrow x_0} f(x) = f(x_0),$$

říkáme, že funkce f je spojitá v bodě x_0 . Je-li funkce f spojitá v každém bodě množiny $M \subseteq X$, pak říkáme, že f je spojitá na množině M .

V případě že $Y = \mathbb{R}$, lze definovat limes superior a limes inferior podobně jako pro posloupnosti výše. Omezíme se opět jen na funkce $f : M \rightarrow \mathbb{R}$.

Definice 2.1.15 (Limes superior a limes inferior pro funkce). Nechť (X, ϱ) je metrický prostor a funkce $f : M \rightarrow \mathbb{R}$, kde $M \subseteq X$. Pak definujeme:

- limes superior funkce f v bodě x_0 :

$$\limsup_{x \rightarrow x_0} f(x) = \lim_{\varepsilon \rightarrow 0} (\sup \{f(x) : x \in M \cap B_\varepsilon(x_0) \setminus \{x_0\}\}),$$

- limes inferior funkce f v bodě x_0 :

$$\liminf_{x \rightarrow x_0} f(x) = \lim_{\varepsilon \rightarrow 0} (\inf \{f(x) : x \in M \cap B_\varepsilon(x_0) \setminus \{x_0\}\}).$$

2.1.4 Bodová a stejnoměrná konvergence

V matematické analýze často pracuje s posloupnostmi funkcí. K jejich limitám se váže dvojice důležitých termínů: *bodová* a *stejnoměrná konvergence*.

Definice 2.1.16 (Bodová a stejnoměrná konvergence). Nechť $\{f_n\}_{n=1}^\infty$ je posloupnost funkcí z metrického prostoru $(\mathcal{C}(\langle a, b \rangle), \varrho)$, kde ϱ je suprémová metrika, tj. pro každou $f, g \in \mathcal{C}(\langle a, b \rangle)$

$$\varrho(f, g) = \sup_{x \in \langle a, b \rangle} |f_n(x) - f(x)|,$$

přičemž $\{f_n\}_{n=1}^\infty$ konverguje k funkci $f \in \mathcal{C}(\langle a, b \rangle)$. Pak říkáme, že posloupnost $\{f_n\}_{n=1}^\infty$ konverguje k f

- *bodově*, pokud pro každé x platí $\lim_{n \rightarrow \infty} f_n(x) = f(x)$,
- *stejnoměrně*, pokud

$$\forall \varepsilon > 0 \exists n_0 \in \mathbb{N} \forall n \geq n_0 \forall x \in \langle a, b \rangle : \varrho(f_n, f) < \varepsilon.$$

Píšeme $f_n \rightrightarrows f$.

Stejnoměrná konvergence implikuje konvergenci bodovou, opačné tvrzení však neplatí. Např. posloupnost funkcí $f_n(x) = x^n$, kde $x \in \langle 0, 1 \rangle$, konverguje k funkci

$$f(x) = \begin{cases} 0 & x < 1, \\ 1 & x = 1, \end{cases}$$

pouze bodově, nikoliv stejnoměrně, protože

$$\sup_{x \in \langle 0, 1 \rangle} |f_n(x) - f(x)| = \sup_{x \in \langle 0, 1 \rangle} |x^n - 0| = 1.$$

2.1.5 Topologické pojmy

Důležitým pojmem v teorii metrických prostorů jsou tzv. kompaktní množiny. S tím se pojí následující termíny.

Definice 2.1.17 (Pokrytí, δ -pokrytí a zjemnění). Nechť je dán metrický prostor (X, ϱ) , $M \subseteq X$ a systém množin $\mathcal{U} \subseteq \mathcal{P}(X)$.

- \mathcal{U} tvoří tzv. *pokrytí* množiny M , pokud

$$M \subseteq \bigcup_{U \in \mathcal{U}} U.$$

Navíc o pokrytí \mathcal{U} říkáme, že je *otevřené*, pokud každá množina $U \in \mathcal{U}$ je otevřená.

- Pokud existuje $\delta > 0$ takové, že pro každé $i \in \mathbb{N}$ platí $\text{diam } U_i < \delta$, pak \mathcal{U} nazýváme δ -*pokrytím*.
- Platí-li pro systém $\mathcal{G} \subseteq \mathcal{P}(X)$, že

$$M \subseteq \bigcup_{G \in \mathcal{G}} G$$

a zároveň $\mathcal{G} \subseteq \mathcal{U}$, pak \mathcal{G} je tzv. *podpokrytí* pokrytí \mathcal{U} .

- Je-li \mathcal{G} pokrytí M a pro každou $G \in \mathcal{G}$ existuje množina $U \in \mathcal{U}$ taková, že $G \subseteq U$, pak \mathcal{G} nazýváme *zjemněním* pokrytí \mathcal{U} .

Definice 2.1.18 (Kompaktní množina). Nechť (X, ϱ) je metrický prostor. Množinu $M \subseteq X$ nazveme *kompaktní*, jestliže pro každé otevřené pokrytí \mathcal{U} existují indexy $n_1, \dots, n_k \in I$, kde I je indexová množina, přičemž

$$M \subseteq \bigcup_{i=1}^k U_{n_i}.$$

Volněji řečeno, pro kompaktní množinu lze z jejího libovolného otevřeného pokrytí vybrat konečné podpokrytí. Dokazovat kompaktnost množiny z definice může být mnohdy nepraktické, nicméně vzhledem k tomu, že budeme často pracovat s prostorem \mathbb{R}^n , záležitost se nám v tomto ohledu značně zjednoduší.

Věta 2.1.19 (Heineho-Borelova). *Nechť ϱ_e je obvyklá eukleidovská metrika na \mathbb{R}^n a $M \subseteq \mathbb{R}^n$. Následující výroky jsou ekvivalentní:*

- (i) *Množina M je kompaktní.*
- (ii) *Množina M je uzavřená a omezená.*

Důkaz věty lze nalézt např. v [11, str. 166]. Pro obecný metrický prostor toto tvrzení již neplatí. Např. v diskrétním metrickém prostoru je každá množina uzavřená i omezená (stačí zvolit $r > 1$, tzn. $B_r(x) = X$). Nicméně uvážíme-li prostor $X = (0, 1)$ s dírkovitou metrikou ϱ , pak z otevřeného pokrytí

$$(0, 1) \subseteq \bigcup_{n=1}^{\infty} \left(\frac{1}{n+1}, \frac{1}{n} \right)$$

nelze vybrat žádné konečné navzdory faktu, že interval $(0, 1)$ je v X uzavřený a omezený.

Věta 2.1.20. *Nechť M je kompaktní množina v metrickém prostoru (X, ϱ) . Pak M je omezená.*

K důkazu věty 2.1.20 lze využít faktu, že každá spojitá funkce definovaná na kompaktní množině nabývá svého maxima.

Věta 2.1.21. *Nechť M je kompaktní množina v \mathbb{R}^n se standardní eukleidovskou metrikou. Pak ∂M je neprázdná množina.*

Věta 2.1.22. *Nechť (X, ϱ) je metrický prostor a $M \subseteq X$ je uzavřená množina. Pak $(\partial M)^\circ = \emptyset$.*

2.1.6 Lipschitzovská zobrazení

Definice 2.1.23 (Lipschitzovské zobrazení). Nechť $(X, \varrho_1), (Y, \varrho_2)$ jsou metrické prostory a $f : X \rightarrow Y$ je zobrazení.

- Zobrazení f nazveme *lipschitzovské*, pokud existuje konstanta $K > 0$ taková, že pro každé $x, y \in X$ platí

$$\varrho_2(f(x), f(y)) \leq K \varrho_1(x, y).$$

Navíc pokud lze volit $K < 1$, pak f nazýváme *kontrakcí*.

- Zobrazení f nazveme *bilipschitzovské*, pokud existují konstanty $K_1, K_2 > 0$ takové, že pro každé $x, y \in X$ platí

$$K_1 \varrho_1(x, y) \leq \varrho_2(f(x), f(y)) \leq K_2 \varrho_1(x, y).$$

Je celkem zjevné, že lipschitzovská zobrazení jsou vždy spojité. Pro každé $\varepsilon > 0$ stačí zvolit kouli o poloměru ε/K , tedy

$$\varrho_2(f(x_0), f(x)) \leq K\varrho_1(x_0, x) \leq K \cdot \frac{\varepsilon}{K} = \varepsilon,$$

což implikuje spojitost f v bodě x_0 .

2.2 Prostory s mírou

Jak již bylo zmíněno v úvodu, klíčovým pojmem v této kapitole (a pro studium fraktálů obecně) je tzv. *míra*. Ta pro nás představuje obecný způsob, jak můžeme množinám přiřadit v jistém smyslu „velikost“. Například Lebesgueova míra v \mathbb{R}^2 je zobecněním pojmu obsahu obdélníku (popř. jiných běžných geometrických útvarů) na obecnější množiny. Konkrétněji, byť vágně lze říci, že sestává-li množina z konečného nebo spočetného množství „rozumných“ částí, pak součet velikostí všech těchto dílčích množin je roven velikosti celé množiny, kterou nazveme její *mírou*. Pro začátek celkem jednoduchá myšlenka.

Pro formální zavedení tohoto pojmu však budeme muset nejprve definovat ještě jiný pojem, a to σ -algebru.

2.2.1 Měřitelné prostory

Definice 2.2.1 (σ -algebra). Nechť je dána libovolná množina X a systém podmnožin $\mathcal{A} \subseteq \mathcal{P}(X)$. Pak \mathcal{A} je σ -algebra na množině X , pokud:

- (a) $X \in \mathcal{A}$.
- (b) Pro každou množinu $A \in \mathcal{A}$ platí $X \setminus A \in \mathcal{A}$.
- (c) Pro libovolné množiny $A_1, A_2, \dots \in \mathcal{A}$ platí $\bigcup_{i=1}^{\infty} A_i \in \mathcal{A}$.

Dvojice (X, \mathcal{A}) se nazývá měřitelný prostor.

Příklad 2.2.2. Jednoduché příklady σ -algeber:

- Triviálními příklady σ -algeber jsou množiny $\mathcal{P}(X)$ a $\{\emptyset, X\}$ pro libovolnou množinu X .
- Pro konečnou množinu $X = \{a, b, c, d\}$ je jednou možnou σ -algebrou systém množin

$$\Sigma = \{\emptyset, \{a, b\}, \{c, d\}, \{a, b, c, d\}\}.$$

Sami se zkuste přesvědčit, že všechny zmíněné příklady jsou σ -algebrou ve smyslu definice 2.2.1.

Než vyslovíme něco dalšího o σ -algebrách a jejich významu, podíváme se na seznam některých vesměs jednoduchých pozorování zformulovaných v tvrzení níže.

Věta 2.2.3 (Vlastnosti σ -algebry). *Nechť (X, \mathcal{A}) je meřitelný prostor. Pak platí:*

(i) $\emptyset \in \mathcal{A}$.

(ii) Pro libovolné množiny $A_1, A_2, \dots \in \mathcal{A}$ platí $\bigcap_{i=1}^{\infty} A_i \in \mathcal{A}$.

(iii) Pro všechny množiny $A_1, A_2, \dots, A_n \in \mathcal{A}$ platí

$$\bigcup_{i=1}^n A_i \in \mathcal{A} \quad a \quad \bigcap_{i=1}^n A_i \in \mathcal{A}.$$

(iv) Jsou-li $A, B \in \mathcal{A}$, pak $A \setminus B \in \mathcal{A}$.

Z tohoto tvrzení je již lépe vidět, proč jsou pro nás σ -algebry tak příjemným objektem. Jsou totiž *uzavřené* na všechny základní množinové operace. To se nám bude později hodit při zavedení míry, ke kterému směřujeme. Důkaz těchto dílčích tvrzení přitom není nikterak složitý.

Důkaz. Mějme σ -algebru \mathcal{A} na množině X .

(i) Z podmínky (a) definice 2.2.1 víme, že $X \in \mathcal{A}$, a z podmínky (b) tedy plyne $\emptyset = X \setminus X \in \mathcal{A}$.

(ii) Mějme množiny $A_1, A_2, \dots \in \mathcal{A}$. Aplikací De Morganových zákonů plyne následující:

$$\bigcap_{i=1}^{\infty} A_i = X \setminus \underbrace{\bigcup_{i=1}^{\infty} \overbrace{(X \setminus A_i)}^{\in \mathcal{A} \text{ podle (b)}}}_{\in \mathcal{A} \text{ podle (c)}} \in \mathcal{A}.$$

(iii) Nechť jsou dány množiny $A_1, A_2, \dots, A_n \in \mathcal{A}$. Když pro každé $j > n$ položíme $A_j = \emptyset \in \mathcal{A}$, pak platí

$$\bigcup_{i=1}^n A_i = \bigcup_{i=1}^{\infty} A_i \in \mathcal{A}$$

a podobně pro $\bigcap_{i=1}^n A_i \in \mathcal{A}$ podle předešlého bodu.

(iv) Pro libovolné množiny $A, B \in \mathcal{A}$ platí

$$A \setminus B = \overbrace{A \cap \overbrace{(X \setminus B)}^{\in \mathcal{A} \text{ podle (b)}}}^{\in \mathcal{A} \text{ podle (c)}} \in \mathcal{A}.$$

□

2.2.2 Míra

V tuto chvíli máme již vše potřebné k zavedení pojmu míry, resp. prostoru s mírou.

Definice 2.2.4 (Prostor s mírou). Nechť (X, \mathcal{A}) je měřitelný prostor. Zobrazení $\mu : \mathcal{A} \rightarrow \langle 0, \infty \rangle$ se nazývá *míra* na X , pokud platí:

$$(a) \quad \mu(\emptyset) = 0,$$

(b) pro množiny $A_1, A_2, \dots \in \mathcal{A}$ po dvou disjunktní je

$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu(A_i). \quad \triangleleft \sigma\text{-aditivita}$$

Uspořádanou trojici (X, \mathcal{A}, μ) nazýváme *prostor s mírou*.

Vzhledem k tomu, co míra reprezentuje (tj. zobecnění délky, obsahu, objemu), jsou tyto požadavky intuitivně dosti smysluplné.

Příklad 2.2.5. Příklady prostorů s mírou:

- Asi pro nás nejtypičtější způsob, jak měřit „velikost“ množiny, je podle *počtu prvků*. Pro libovolnou množinu X a potenční množinu $\mathcal{P}(X)$ lze definovat prostor s mírou $(X, \mathcal{P}(X), \mu)$, kde pro libovolnou konečnou množinu $A \in \mathcal{P}(X)$ položíme $\mu(A) = |A|$ a pro nekonečnou množinu $B \in \mathcal{P}(X)$ položíme $\mu(B) = \infty$. Takto definované míře μ říkáme *aritmetická míra*.
- Máme libovolnou množinu X a σ -algebru \mathcal{A} na X . Zvolme si pevně $x \in X$. Míru libovolné množiny $A \in \mathcal{A}$ lze definovat jako $\delta_x(A) = \chi_A(x)$, kde χ_A je charakteristická funkce množiny A . Podrobněji

$$\delta_x(A) = \chi_A(x) = \begin{cases} 1 & x \in A, \\ 0 & x \notin A. \end{cases}$$

Zobrazení δ_x je tzv. *Diracova míra* příslušná v bodě x .

- V moderní teorii pravděpodobnosti je obvyklé zobrazení přiřazující náhodnému jevu pravděpodobnost definovat jako míru. Označíme-li si $\Omega = \{\omega_1, \omega_2, \dots, \omega_n\}$ množinu všech elementárních jevů a σ -algebru $\mathcal{F} \subseteq \mathcal{P}(\Omega)$, pak $\mathsf{P} : \mathcal{F} \rightarrow \langle 0, 1 \rangle$ definovaná pro $A \in \mathcal{F}$ jako

$$\mathsf{P}(A) = \frac{|A|}{|\Omega|}$$

je mírou na \mathcal{F} . Speciálně $\mathsf{P}(\Omega) = 1$.

Ve všech případech zobrazení μ v příkladu 2.2.5 se lze snadno přesvědčit, že se jedná o míru, tedy že je definováno na příslušné σ -algebře, je nezáporné a splňuje podmínky (a) a (b) uvedené v definici 2.2.4 výše.

Pojďme nyní prozkoumat vlastnosti míry trochu hlouběji.

Věta 2.2.6 (Vlastnosti míry). *Nechť μ je míra na σ -algebře \mathcal{A} . Pak platí následující:*

(i) *Jsou-li množiny $A_1, A_2, \dots, A_n \in \mathcal{A}$ po dvou disjunktní, pak*

$$\mu\left(\bigcup_{i=1}^n A_i\right) = \sum_{i=1}^n \mu(A_i). \quad \triangleleft \text{aditivita}$$

(ii) *Pokud $A, B \in \mathcal{A}$ a $A \subseteq B$, pak*

$$\mu(A) \leq \mu(B). \quad \triangleleft \text{monotonie míry}$$

Navíc pokud $\mu(A) < \infty$, pak $\mu(B \setminus A) = \mu(B) - \mu(A)$.

(iii) *Pokud A_1, A_2, \dots , kde $A_i \in \mathcal{A}$ pro každé $i \in \mathbb{N}$, je neklesající posloupnost množin², pak*

$$\lim_{j \rightarrow \infty} \mu(A_j) = \mu\left(\bigcup_{j=1}^{\infty} A_j\right).$$

(iv) *Pokud A_1, A_2, \dots , kde $A_i \in \mathcal{A}$ pro každé $i \in \mathbb{N}$, je nerostoucí posloupnost množin³ a navíc $\mu(A_1) < \infty$, pak*

$$\lim_{j \rightarrow \infty} \mu(A_j) = \mu\left(\bigcap_{j=1}^{\infty} A_j\right).$$

(v) *Pokud A_1, A_2, \dots , kde $A_i \in \mathcal{A}$ pro každé $i \in \mathbb{N}$, pak*

$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) \leq \sum_{i=1}^{\infty} \mu(A_i). \quad \triangleleft \sigma\text{-subaditivita}$$

Poslední vlastnost (v) je tzv. σ -subaditivita⁴. Od σ -aditivity se liší tím, že u množin A_1, A_2, \dots se nepožaduje, aby byly po dvou disjunktní, tzn. mohou se „překrývat“. Je však intuitivně nejspíše jasné, že součtem měr všech těchto množin určitě nemůžeme získat míru nižší než je míra jejich sjednocení (dané „překryvy“ započítáváme v sumě vícekrát). Podobně i monotonie dává intuitivně smysl, neboť část větší množiny jistě nemůže mít větší míru než celek. Na formální stránku věci se podíváme nyní.

Důkaz. V důkazu využijeme některé vlastnosti σ -algebry z věty 2.2.3, zejména, že všechny množiny níže jsou opět prvky σ -algebry \mathcal{A} .

(i) Pokud pro každé $j > n$ položíme $A_j = \emptyset$, pak z definice míry plyne

$$\mu\left(\bigcup_{i=1}^n A_i\right) = \mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu(A_i) = \sum_{i=1}^n \mu(A_i).$$

²Posloupnost množin, kde $A_i \subseteq A_{i+1}$ pro každé $i \in \mathbb{N}$.

³ $A_i \supseteq A_{i+1}$ pro každé $i \in \mathbb{N}$.

⁴V matematické terminologii se předpona „ σ “ běžně týká spočetných sjednocení. [7, str. 2]

- (ii) Nechť jsou dány $A, B \in \mathcal{A}$ takové, že $A \subseteq B$. Pak $B = A \cup (B \setminus A)$, přičemž A a $B \setminus A$ jsou disjunktní. Tedy podle bodu (b) lze psát $\mu(B) = \mu(A) + \mu(B \setminus A) \geq \mu(A)$, protože $\mu(B \setminus A) \geq 0$.
- (iii) Mějme neklesající posloupnost množin A_1, A_2, \dots , kde $A_i \in \mathcal{A}$ pro každé $i \in \mathbb{N}$ (viz obrázek 2.1a). Definujeme posloupnost⁵ množin B_1, B_2, \dots následovně:

$$A_0 = \emptyset, \quad B_1 = A_1, \quad B_2 = A_2 \setminus A_1, \quad B_i = A_i \setminus A_{i-1}, \quad i \geq 2.$$

Množiny B_1, B_2, \dots jsou po dvou disjunktní a zároveň $\bigcup_{i=1}^{\infty} B_i = \bigcup_{i=1}^{\infty} A_i$. Libovolnou množinu A_n lze totiž zapsat jako

$$A_n = \bigcup_{i=1}^n (A_i \setminus A_{i-1}) = \bigcup_{i=1}^n B_i.$$

Podle již dokázaného bodu (i) (aditivita míry) tedy pro každé n platí $\mu(A_n) = \sum_{i=1}^n \mu(B_i)$. Celkově

$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \mu\left(\bigcup_{i=1}^{\infty} B_i\right) = \sum_{i=1}^{\infty} \mu(B_i) = \lim_{n \rightarrow \infty} \sum_{i=1}^n \mu(B_i) = \lim_{n \rightarrow \infty} \mu(A_n).$$

- (iv) Nechť je dána nerostoucí posloupnost množin A_1, A_2, \dots , kde $A_i \in \mathcal{A}$ pro každé $i \in \mathbb{N}$ (viz obrázek 2.1b). Podobně jako v předešlém bodě, i zde definujeme novou posloupnost množin B_1, B_2, \dots takto:

$$B_i = A_1 \setminus A_i, \quad i \in \mathbb{N}.$$

Zde si můžeme všimnout, že pro každé i platí $B_i \subseteq B_{i+1}$, a splňuje tak předpoklad předešlého bodu (iii). Dle De Morganových zákonů můžeme psát

$$\bigcup_{i=1}^{\infty} B_i = \bigcup_{i=1}^{\infty} (A_1 \setminus A_i) = A_1 \setminus \bigcap_{i=1}^{\infty} A_i.$$

Výraz $\lim_{j \rightarrow \infty} \mu(B_j)$ lze rozepsat dvěma způsoby:

$$\begin{aligned} \lim_{j \rightarrow \infty} \mu(B_j) &\stackrel{(iii)}{=} \mu\left(\bigcup_{i=1}^{\infty} B_i\right) = \mu\left(A_1 \setminus \bigcap_{i=1}^{\infty} A_i\right) \stackrel{(ii)}{=} \mu(A_1) - \mu\left(\bigcap_{i=1}^{\infty} A_i\right), \\ \lim_{j \rightarrow \infty} \mu(B_j) &= \lim_{j \rightarrow \infty} (A_1 \setminus A_j) \stackrel{(ii)}{=} \lim_{j \rightarrow \infty} (\mu(A_1) - \mu(A_j)) = \mu(A_1) - \lim_{j \rightarrow \infty} \mu(A_j). \end{aligned}$$

Porovnáním obou rovností lze vidět, že

$$\lim_{j \rightarrow \infty} \mu(A_j) = \mu\left(\bigcap_{i=1}^{\infty} A_i\right).$$

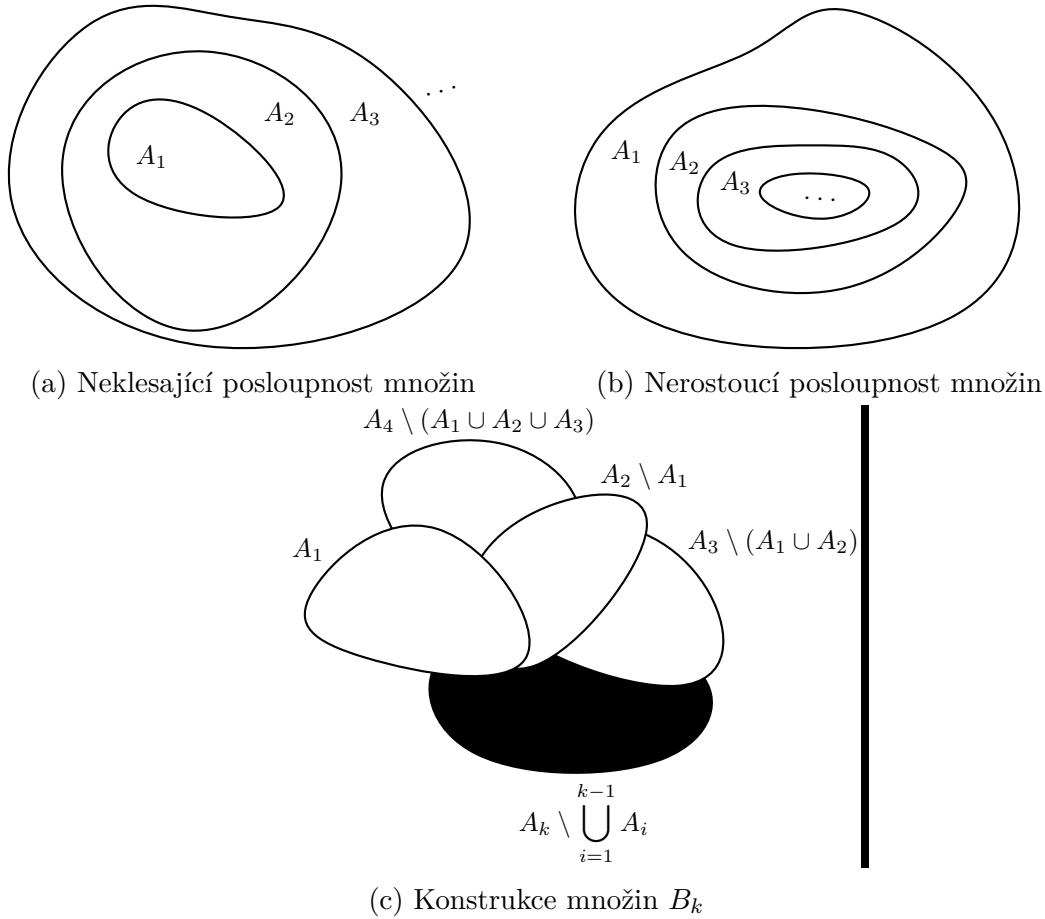
⁵V podstatě konstruujeme množiny A_1, A_2, \dots tak, aby v následující množině A_i nebyl obsažen prvek, který se nachází již v některé z množin A_1, A_2, \dots, A_{i-1} . Podobná myšlenka je využita i při důkazu budú (iv) a (v).

(v) Nechť jsou dány množiny A_1, A_2, \dots , kde $A_i \in \mathcal{A}$ pro každé $i \in \mathbb{N}$. Definujeme posloupnost množin B_1, B_2, \dots takto:

$$B_1 = A_1, \quad B_k = A_k \setminus \bigcup_{i=1}^{k-1} A_i, \quad k \leq 2.$$

Pro představu viz obrázek 2.1c. Lze si rozmyslet, že množiny B_1, B_2, \dots jsou po dvou disjunktní. Zároveň platí $\bigcup_{i=1}^{\infty} B_i = \bigcup_{i=1}^{\infty} A_i$ a $B_j \subseteq A_j$ pro každé $j \in \mathbb{N}$. Tím je dokázáno, že

$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \mu\left(\bigcup_{i=1}^{\infty} B_i\right) = \sum_{i=1}^{\infty} \mu(B_i) \stackrel{(ii)}{\leq} \sum_{i=1}^{\infty} \mu(A_i).$$



Obrázek 2.1: Ilustrace k důkazu věty 2.2.6

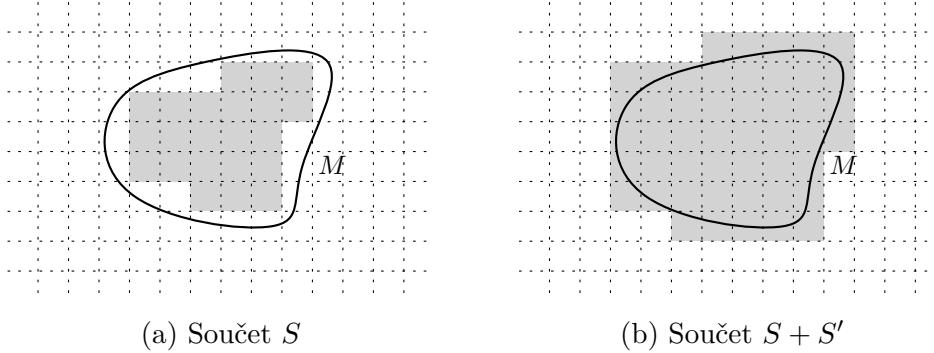
(Převzato z [12, str. 19])

□

Poznámka 2.2.7. Předpoklad $\mu(A_1) < \infty$ ve větě 2.2.6 v bodě (iv) nelze vynechat. Jednoduchý protipříklad si uvedeme v sekci 2.3.

2.3 Lebesgueova míra

Jedněmi z nejznámějších průkopníků v oblasti teorie míry byli francouzský matematik CAMILLE JORDAN⁶ a italský matematik GIUSEPPE PEANO⁷, kteří ve svých publikacích popsali způsob měření „velikosti“ množin dnes známý jako *Jordanova-Peanova míra*. Rovinu \mathbb{R}^2 si rozdělíme na čtvercovou síť, přičemž S bude představovat součet obsahů všech čtverců síťě, jež jsou obsaženy ve vnitřku množiny M a S' je součet obsahů všech čtverců síťě mající společný alespoň jeden bod s hranicí množiny M . Součet $S + S'$ pak představuje obsah všech čtverců síťě, které obsahují body uzávěru množiny M (viz obrázek 2.2). Při zjemňování



Obrázek 2.2: Vnitřní a vnější Jordanova-Peanova míra množiny M

čtvercové síťě konvergují součty S a $S + S'$ k limitám, které po řadě nazýváme *vnitřní* a *vnější Jordanova-Peanova míra*. Pokud se obě tyto limity shodují, mluvíme zkrátka *Jordanově-Peanově míře* množiny M . Tato úvaha položila základy pro vznik teorie míry. [23]

V předešlé sekci 2.2 jsme si přiblížili pojmem „míra“ obecně a podívali jsme se na několik příkladů. Obecnou ideu měření „velikosti“ lze založit např. na approximaci obecné množiny pomocí *spočetných sjednocení útvarů*, jejichž „velikost“ umíme jednoduše určit. V dalším textu se omezíme pouze na množinu \mathbb{R}^n .

Na zmíněné myšlence je postavena definice *n-rozměrné Lebesgueovy míry*, kdy obecnou množinu budeme pokrývat pomocí *kvádrů*. Připomeňme, že obecně *n*-rozměrným kvádrem I rozumíme kartézský součin *intervalů*

$$\langle a_1, b_1 \rangle, \dots, \langle a_n, b_n \rangle \subseteq \mathbb{R},$$

tj.

$$I = \prod_{i=1}^n \langle a_i, b_i \rangle = \langle a_1, b_1 \rangle \times \langle a_2, b_2 \rangle \times \cdots \times \langle a_n, b_n \rangle,$$

a jeho objem definujeme jako

$$\text{vol}_n(I) = \prod_{i=1}^n (b_i - a_i).$$

Lze nejspíše ihned vidět, že objem $\text{vol}_n(I)$ je *aditivní* i *subaditivní*.

Nyní si definujeme tzv. *vnější Lebesgueovu míru*.

⁶1832–1922

⁷1958–1932

Definice 2.3.1 (Vnější Lebesgueova míra). Nechť $A \subseteq \mathbb{R}^n$. Pak vnější n -rozměrnou Lebesgueovou mírou A je

$$\lambda_n^*(A) = \inf \left\{ \sum_{j=1}^{\infty} \text{vol}_n(I_j) \mid I_j \text{ je kvádr pro každé } j \in \mathbb{N}, A \subseteq \bigcup_{j=1}^{\infty} I_j \right\}.$$

Vnější Lebesgueova míra množiny intuitivně zachycuje informaci o „velikosti“ dané množiny. Zjevně platí pro libovolnou množinu $A \subseteq \mathbb{R}^n$, že $\lambda_n^*(A) \in \mathbb{R}_0^+$, protože $\text{vol}_n(I_j) \geq 0$ pro každé $j \in \mathbb{N}$.

Příklad 2.3.2. Ukažme si některé triviální příklady výpočtů vnější Lebesgueovy míry z definice (viz 2.3.1). Budeme tedy hledat příslušné pokrytí dané množiny.

- Pro prázdnou množinu \emptyset je $\lambda_n^*(\emptyset) = 0$, neboť $\emptyset \subseteq \prod_{i=1}^n \langle 0, 0 \rangle$ (prázdnou množinu lze pokrýt jakýmkoliv kvádrem) a

$$\text{vol}_n \left(\prod_{i=1}^n \langle 0, 0 \rangle \right) = 0.$$

- Mějme libovolnou konečnou množinu $A = \{x_1, x_2, \dots, x_n\} \subseteq \mathbb{R}^n$. Pro každé x_j stačí položit $I_j = \{x_j\}$ pro každé $1 \leq j \leq n$, což je degenerovaný interval splňující $\text{vol}_n(I_j) = 0$. Poznamenejme, že i singleton představuje kvádr.
- Pro libovolnou spočetnou množinu $A = \{x_i \mid i \in \mathbb{N}\} \subseteq \mathbb{R}^n$ je $\lambda_n^*(A) = 0$. Pokrytí volíme stejně jako v předešlém bodě. Tedy např. pro \mathbb{Q} je $\lambda_1^*(\mathbb{Q}) = 0$, neboť \mathbb{Q} je spočetná.
- Pro množinu \mathbb{R} je $\lambda_1^*(\mathbb{R}) = \infty$, avšak pro

$$A = \{(x, 0) \mid x \in \mathbb{R}\} \subset \mathbb{R}^2$$

(osa x v \mathbb{R}^2) je $\lambda_2^*(A) = 0$. Stačí definovat kvádr

$$I_j = \langle j, j+1 \rangle \times \langle 0, 0 \rangle$$

pro každé $j \in \mathbb{Z}$.

Jako poslední si ukážeme, že vnější Lebesgueova míra n -rozměrného kvádru je rovna jeho objemu.

Tvrzení 2.3.3. Je-li $I \subset \mathbb{R}^n$ kvádr, pak $\lambda_n^*(I) = \text{vol}_n(I)$.

Důkaz. Ukážeme zvlášť, že $\lambda_n^*(I) \leq \text{vol}_n(I)$ a $\lambda_n^*(I) \geq \text{vol}_n(I)$.

- Zvolme pokrytí $\mathcal{I} = \{I_1, I_2, \dots\}$ kvádru I , tzn. $I \subseteq \bigcup_{i=1}^{\infty} I_i$, tak, aby platilo

$$\sum_{i=1}^{\infty} \text{vol}_n(I_i) \leq (1 + \varepsilon) \lambda_n^*(I)$$

pro nějaké $\varepsilon > 0$. Nyní si zvolíme nové kvádry⁸ $\mathcal{J} = \{J_1, J_2, \dots\}$ tak, aby pro každé $i \in \mathbb{N}$ bylo

$$I \subset J^\circ \wedge \text{vol}_n(J_i) \leq (1 + \varepsilon) \text{vol}_n(I_i).$$

To není nikterak složité, stačí např. pro každé $i \in \mathbb{N}$ položit

$$J_i = \prod_{j=1}^n \left\langle x_j - r_j \sqrt[n]{1+\varepsilon}, x_j + r_j \sqrt[n]{1+\varepsilon} \right\rangle,$$

kde

$$I_i = \langle x_1 - r_1, x_1 + r_1 \rangle \times \dots \times \langle x_n - r_n, x_n + r_n \rangle.$$

Protože však I je uzavřená a omezená množina, je podle věty 2.1.19 kompaktní, tedy z otevřeného pokrytí $J_1^\circ, J_2^\circ, \dots$ lze vybrat konečné podpokrytí. Existuje tedy $m \in \mathbb{N}$ takové, že

$$I \subseteq \bigcup_{i=1}^m J_i.$$

Celkově tedy dostáváme

$$\begin{aligned} \text{vol}_n(I) &\leq \sum_{i=1}^m \text{vol}_n(J_i) \leq (1 + \varepsilon) \sum_{i=1}^m \text{vol}_n(I_i) \leq (1 + \varepsilon) \sum_{i=1}^{\infty} \text{vol}_n(I_i) \\ &\leq (1 + \varepsilon)^2 \lambda_n^*(I) \end{aligned}$$

pro každé $\varepsilon > 0$, což dokazuje požadovanou nerovnost.

- Důkaz opačné nerovnosti je velmi jednoduchý. Kvádr I totiž reprezentuje pokrytí sebe samotného, tzn. lze volit $I_1 = I$ a zbylé kvádry I_j , kde $j \geq 2$, mohou být libovolné s nulovým objemem.

□

Poznámka 2.3.4. Vraťme se ještě k větě 2.2.6 o vlastnostech míry, konkrétně bod (iv). Předpoklad $\mu(A_1) < \infty$ zde vynechat nelze. Snadno si rozmyslíme, že pokud uvážíme množiny $A_j = \langle j, \infty \rangle$, pak $\lambda_n^*(A_j) = \lambda_n^*(\langle j, \infty \rangle) = \infty$ pro každé $j \in \mathbb{N}$ a tedy

$$\bigcap_{i=1}^{\infty} A_i = \emptyset.$$

Lze vidět, že zatímco $\lim_{j \rightarrow \infty} \mu(A_j) = \infty$, tak $\mu(\bigcap_{i=1}^{\infty} A_i) = 0$.

Z příkladů 2.3.2 a 2.3.3 můžeme tušit, že pro rozumně zvolené množiny zachycuje vnější Lebesgueova míra jejich intuitivní „velikost“. V případě intervalu odpovídá jeho délce, v případě diskrétní množiny je nulová a podobně např. pro obdélník odpovídá jeho obsahu, pro kvádr jeho objemu atd.

Nyní se však nabízí jedna otázka. Čtenář by mohl již od chvíle, kdy jsme zavedli pojem vnější Lebesgueovy míry (opět viz definice 2.3.1), namítat, co nás

⁸Formálně \mathcal{I} tvoří zjemnění pokrytí \mathcal{J} .

opravňuje nazývat zobrazení λ_n^* mírou ve smyslu definice 2.2.4. Jak víme, že splňuje podmínu σ -aditivity? Odpověď na tuto otázku není zcela přímočará ani jednoduchá.

Bohužel v případě vnější Lebesgueovy míry pro obecnou množinu neplatí vlastnost aditivity, tedy existují množiny $A, B \in \mathcal{A}$ takové, že

$$\lambda_n^*(A \cup B) \neq \lambda_n^*(A) + \lambda_n^*(B).$$

Příklad takové množiny poskytuje např. takzvaná *Vitaliho konstrukce*, se kterou přišel italský matematik GIUSEPPE VITALI⁹ roku 1905, využívající invariance vnější Lebesgueovy míry vůči posunutí, tzn. $\lambda_n^*(x + A) = \lambda_n^*(A)$. [17] V rámci tohoto textu se jí zde zabývat nebudeme, avšak pro zájemce doporučujeme zdroje [7, str. 3] a [21], kde je Vitaliho množina podrobněji rozepsána.

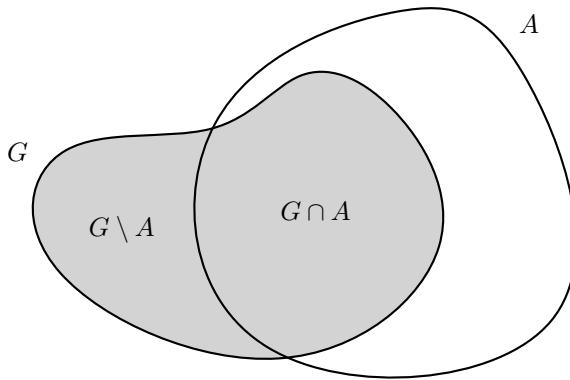
Je tedy potřeba se omezit na takové množiny, kde je λ_n^* aditivní. Existuje více způsobů jejich charakterizace, avšak my si zde uvedeme ten, se kterým přišel řecký matematik CONSTANTIN CARATHÉODORY¹⁰.

Definice 2.3.5 (Lebesgueovská měřitelnost). Množinu $A \subseteq \mathbb{R}^n$ nazveme (lebesgueovsky) měřitelnou, pokud pro každou množinu G platí

$$\lambda_n^*(G) = \lambda_n^*(G \cap A) + \lambda_n^*(G \setminus A).$$

Systém všech měřitelných množin v \mathbb{R}^n značíme $\mathcal{L}(\mathbb{R}^n)$. Pokud $A \in \mathcal{L}(\mathbb{R}^n)$, pak číslo $\lambda_n(A) = \lambda_n^*(A)$ nazýváme n -rozměrnou Lebesgueovou mírou množiny A .

Podmínka v definici 2.3.5 se někdy nazývá *Carathéodoryho kritérium*. Zjednodušeně říká, že množina A je lebesgueovsky měřitelná, když při „rozdělení“ *libovolně* zvolené množiny G na dvě části pomocí A lze míru G stanovit součtem měr daných částí (viz obrázek 2.3). Zároveň je dobré (a snadné) si rozmyslet, že kvádry, které figurují v definici vnější Lebesgueovy míry, jsou měřitelné. O systému $\mathcal{L}(\mathbb{R}^n)$



Obrázek 2.3: Ilustrace měřitelnosti množiny A

a Lebesgueově míře λ_n lze dokázat následující tvrzení.

Věta 2.3.6. Platí:

⁹1875–1932

¹⁰1873–1950

- (i) $(\mathbb{R}^n, \mathcal{L}(\mathbb{R}^n))$ je měřitelný prostor.
- (ii) $(\mathbb{R}^n, \mathcal{L}(\mathbb{R}^n), \lambda_n)$ je prostor s mírou.

Čtenář snad promine, že formální důkaz v případě tohoto tvrzení v zájmu zachování stručnosti textu zcela vynecháme, nicméně zvídavý jedinec jej může nalézt např. v knize [20, str. 347], kde jsou příslušné záležitosti rozepsány.

2.4 Box-counting dimenze

Tomuto typu dimenze jsme se již v základu věnovali v sekci 1.3, kde jsme rozebrali způsob jejího výpočtu a ukázali jsme si jej na několika příkladech. V této části si blíže rozebereme některé další vlastnosti týkající se právě *box-counting dimenze*¹¹ a pokusíme se ji lépe zasadit do kontextu teorie míry, které jsme se samostatně věnovali.

2.4.1 Definice a výpočet

Jako první si přiblížíme myšlenku box-counting a malíčko si ji zobecníme. Původně jsme nahlíželi na dimenzi jako na exponent, s nímž roste „velikost“ zkoumaného útvaru. Tato myšlenka se ukázala jako rozumná, neboť pro „klasické“ geometrické útvary vycházela tato dimenze vždy celočíselně, nicméně už tomu tak nebylo v případě fraktálních útvarů. Podstata byla taková, že jsme útvar rozdělili na určitý počet stejně „velkých částí“. Mějme tedy útvar F a rozdělme jej na části F_1, F_2, \dots, F_m v nějakém měřítku $\varepsilon > 0$. Zkusme nyní požadavek na striktně stejnou velikost (formálně vzato míru) trochu rozvolnit. Bude nám stačit, když pro každé i je

$$\text{diam } F_i \leq \delta, \text{ kde } \delta > 0.$$

Zároveň nebudeme požadovat, aby množiny F_1, F_2, \dots, F_m byly všechny po dvou téměř disjunktními¹² podmnožinami F , ale stačí, když budou tvořit pokrytí F .

Mějme tedy nějakou neprázdnou omezenou množinu $F \subset \mathbb{R}^n$, kde pro každé $\delta > 0$ budeme hledat *nejmenší počet* množin pokrývajících F . Toto číslo si označíme $N_\delta(F)$. Dimenze množiny F by tedy měla odrážet „rychlosť“ růstu $N_\delta(F)$ pro $\delta \rightarrow 0_+$. Je-li splněna approximace

$$N_\delta(F) \approx c\delta^{-s} \tag{2.1}$$

pro $c > 0$, pak řekneme, že množina F má box-counting dimenzi s . (Převzato z [5, str. 27].)

Poznámka 2.4.1. V dalším textu budeme místo $\delta \rightarrow 0_+$ psát pro jednoduchost pouze $\delta \rightarrow 0$, byť by se slušelo používat první variantu. Čtenáři je však nejspíše jasné, že uvažovat záporný průměr množiny nemá smysl.

¹¹V kapitole 1 jsme pro jednoduchost používali obecnější termín *fraktální dimenze*. Ten však zahrnuje daleko širší škálu možných definic než jen tu, kterou jsme si představili. Proto dále v tomto textu budeme používat výhradně její skutečný název, tj. *box-counting dimenze*.

¹²Množiny M, N jsou *téměř disjunktní*, pokud $M^\circ \cap N^\circ = \emptyset$, tedy může nastat, že se na hranici mohou „dotýkat“, tzn. $\partial M \cap \partial N \neq \emptyset$.

Logaritmováním a úpravou výrazu (2.1) dostaneme:

$$\ln N_\delta(F) \approx \ln c + \ln \delta^{-s} \quad (2.2)$$

$$\ln N_\delta(F) \approx \ln c - s \ln \delta \quad (2.3)$$

$$s \approx \frac{\ln N_\delta(F)}{-\ln \delta} + \frac{\ln c}{\ln \delta}. \quad (2.4)$$

Když porovnáme výsledek v (2.2) s rovností (1.7) z minulé kapitoly, všimneme si, že zde navíc figuruje člen $\ln c / \ln \delta$. Když však uvážíme limitu daného výrazu pro $\delta \rightarrow 0$, dostaneme původní vzorec, který jsme již viděli, tj.

$$\lim_{\delta \rightarrow 0} \left(\frac{\ln N_\delta(F)}{-\ln \delta} + \frac{\ln c}{\ln \delta} \right) = \lim_{\delta \rightarrow 0} \frac{\ln N_\delta(F)}{-\ln \delta} + \lim_{\delta \rightarrow 0} \frac{\ln c}{\ln \delta} = \lim_{\delta \rightarrow 0} \frac{\ln N_\delta(F)}{-\ln \delta}.$$

Předchozí úvahu můžeme shrnout do následující definice.

Definice 2.4.2 (Box-counting dimenze). Nechť $F \subset \mathbb{R}^n$ je neprázdná omezená množina. Pak definujeme následující:

(a) *Nejmenší počet množin v δ -pokrytí množiny F značíme $N_\delta(F)$, tj.*

$$N_\delta(F) = \min \left\{ m \in \mathbb{N} \mid F \subseteq \bigcup_{i=1}^m F_i, \text{ diam } F_j \leq \delta \text{ pro } 1 \leq j \leq m \right\}.$$

(b) *Horní box-counting dimenze množiny F je*

$$\overline{\dim}_B F = \limsup_{\delta \rightarrow 0} \frac{\ln N_\delta(F)}{-\ln \delta}.$$

(c) *Dolní box-counting dimenze množiny F je*

$$\underline{\dim}_B F = \liminf_{\delta \rightarrow 0} \frac{\ln N_\delta(F)}{-\ln \delta}.$$

V případě, že $\underline{\dim}_B F = \overline{\dim}_B F$, pak společnou hodnotu nazýváme *box-counting dimenzí* množiny F , značíme $\dim_B F$, přičemž platí

$$\dim_B F = \lim_{\delta \rightarrow 0} \frac{\ln N_\delta(F)}{-\ln \delta}.$$

Poznámka 2.4.3. Zde je důležité zmínit, že v dalším textu budeme uvažovat δ dostatečně malé, konkrétně $0 < \delta < 1$, tzn. hodnota $-\ln \delta$ je vždy kladná. Dále budeme pracovat (podle definice 2.4.2) pouze s neprázdnými omezenými množinami, abychom se vyhnuli problémům s případy, kdy „ $N_\delta(F) = \infty$ “ nebo $N_\delta(F) = 0$. Každá omezená množina v \mathbb{R}^n je totiž obsažena v nějaké kompaktní množině, tzn. vždy nám postačí konečně mnoho množin o průměru nejvýše δ k jejímu pokrytí.

pro úplnost ještě zmíníme, že box-counting dimenzi lze taktéž definovat více způsoby. V tuto uvažujeme obecně δ -pokrytí dané množiny F , tj. pokrytí *obecnými* množinami o průměru maximálně $\delta > 0$. Lze se však zaměřit i na konkrétní útvary, jak ukazuje následující věta.

Věta 2.4.4 (Ekvivalentní definice box-counting dimenze). *Nechť $F \subset \mathbb{R}^n$ je neprázdná omezená množina. Pak*

$$\begin{aligned}\underline{\dim}_B F &= \liminf_{\delta \rightarrow 0} \frac{\ln M_\delta(F)}{-\ln \delta}, \\ \overline{\dim}_B F &= \limsup_{\delta \rightarrow 0} \frac{\ln M_\delta(F)}{-\ln \delta}, \\ \dim_B F &= \lim_{\delta \rightarrow 0} \frac{\ln M_\delta(F)}{-\ln \delta},\end{aligned}$$

kde $M_\delta(F)$ je definováno jedním z následujících vzorců:

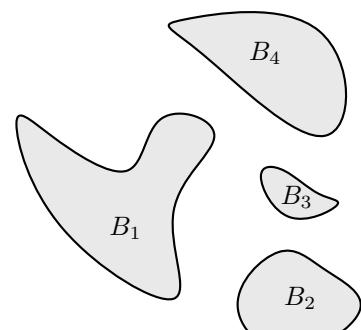
- (i) $M_\delta(F) = \inf \left\{ m \mid F \subseteq \bigcup_{i=1}^m K_\delta(x_i), x_j \in \mathbb{R}^n \text{ pro } 1 \leq j \leq m \right\};$
- (ii) $M_\delta(F) = \inf \left\{ m \mid F \subseteq \bigcup_{i=1}^m I_i, I_j = \prod_{k=1}^n \langle a_k, a_k + \delta \rangle \text{ pro } 1 \leq j \leq m \right\};$
- (iii) $M_\delta(F) = |\{I \mid I \cap F \neq \emptyset, I \in \mathcal{Q}_\delta\}|,$ kde $\delta > 0;$
- (iv) $M_\delta(F) = \sup \{m \mid B_\delta(x_i) \cap B_\delta(x_j) = \emptyset; x_i, x_j \in \mathbb{R}^n \text{ pro } 1 \leq i, j \leq m\}.$

Pojďme si větu 2.4.4 trochu rozebrat.

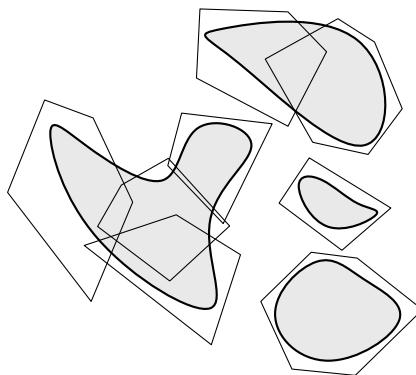
- Body (i) a (iv) říkají, že $M_\delta(F)$ je rovno nejmenšímu počtu uzavřených koulí o poloměru δ , které pokrývají F , resp. nejvyšší počet disjunktních otevřených koulí o poloměru δ , které mají střed v F .
- Podobně body (ii) a (iii) tvrdí, že $M_\delta(F)$ lze definovat jako pokrytí kvádry o stranách délky δ , resp. jako počet všech kvádrů z δ -mříže, které mají s F neprázdný průnik.
- Výše zmíněné definice $M_\delta(F)$ si obecně nejsou ekvivalentní a pevně zvolenou množinu F a δ mohou dávat různé hodnoty. Hodnoty dimenzí však vydou (nezávisle na definici $M_\delta(F)$) vždy stejně.

Pro představu viz obrázek 2.4. Důkaz věty je delší a opět jej vynecháme, nicméně lze jej nalézt v knize [5, str. 30].

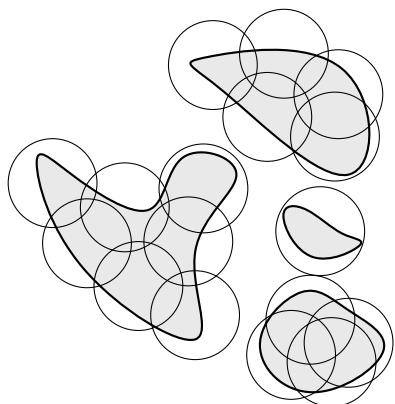
Zároveň body (ii) a (iii) nám dívají dobré opodstatnění názvu tohoto typu dimenze, neboť v podstatě zkoumáme pokryvání daného obrazce „kostkami“. Při approximacích box-counting dimenze obrazce $F \subset \mathbb{R}^2$ tak lze pracovat s mřížkou čtverců o libovolné straně $\delta > 0$, kdy $M_\delta(F)$ stanovíme jako počet čtverců, které se překrývají se zkoumaným obrazcem F . Když se tedy zpět vrátíme k otázce rozebírané v úvodu tohoto textu týkající se délky pobřeží (viz kapitola 1), lze jeho „fráktálnost“ do jisté míry vyjádřit právě popsaným způsobem (viz obrázek 2.5). Nyní se opět vrátíme k fraktálům a výpočtům jejich dimenze, čemuž jsme se věnovali již v podsekci 1.3.2. Tentokrát však budeme postupovat přímo podle definice box-counting dimenze 2.4.2, tedy budeme zvlášt zkoumat horní a dolní box-counting dimenzi.



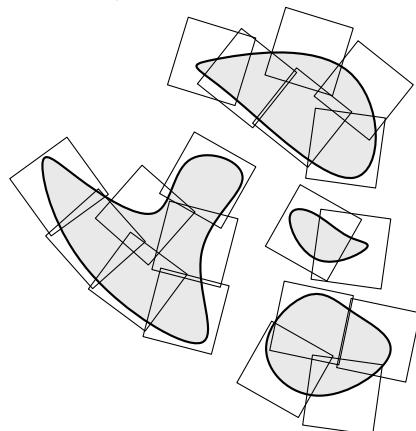
(a) Množina $B = \bigcup_{i=1}^4 B_i$



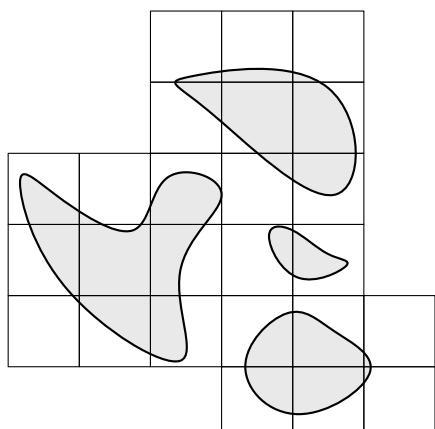
(b) δ -pokrytí množiny B (viz definice 2.4.2)



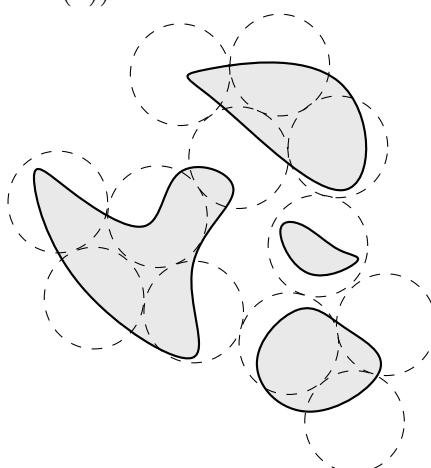
(c) Pokrytí uzavřenými koulemi (viz bod (i))



(d) Pokrytí pomocí kvádrů (viz bod (ii))

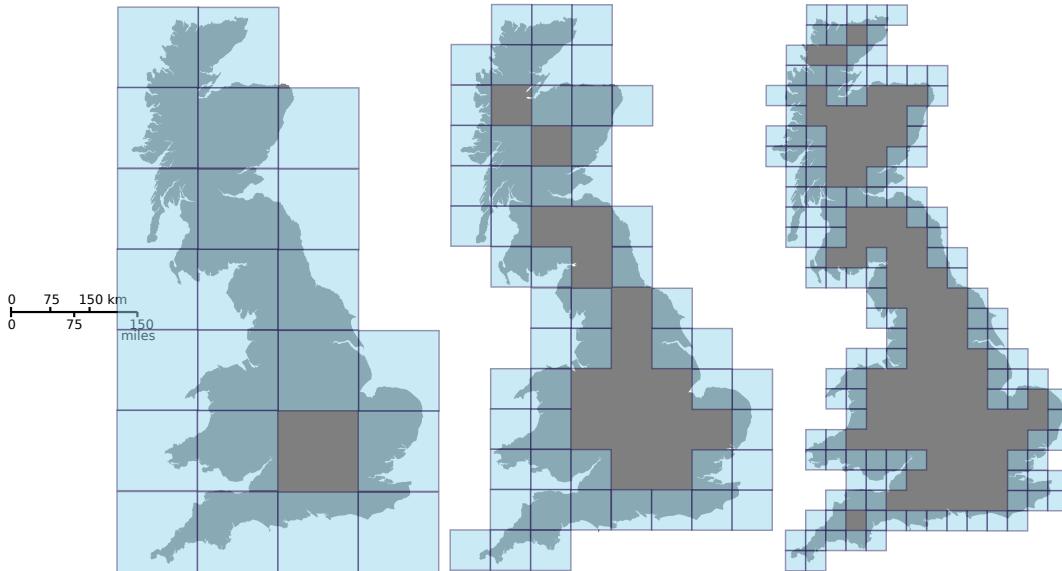


(e) δ -mříž (viz bod (iii))



(f) Pokrytí otevřenými po dvou disjunktními koulemi (viz bod (iv))

Obrázek 2.4: Ilustrace věty 2.4.4 (Inspirováno [5, str. 29])



Obrázek 2.5: Aproximace box-counting dimenze pobřeží Velké Británie (Převzato z Wikipedia Commons, viz [odkaz](#))

Příklad 2.4.5 (Cantorovo diskontinuum). Formálně můžeme popsat Cantorovo diskontinuum C jako průnik množin C_k pro $k = 0, 1, 2, \dots$, přičemž

$$C_k = \bigcup_{j=0}^{3^{k-1}-1} \left(\left\langle \frac{3j+0}{3^k}, \frac{3j+1}{3^k} \right\rangle \cup \left\langle \frac{3j+2}{3^k}, \frac{3j+3}{3^k} \right\rangle \right).$$

Platí, že $C = \bigcap_{k=0}^{\infty} C_k$, přičemž k -tou iterací budeme rozumět množinu $C_k = \bigcap_{i=0}^k C_i$. Již jsme měli možnost se přesvědčit, že tento fraktál má box-counting dimenzi $\ln 2 / \ln 3$. Zkusme nyní výpočet zopakovat, avšak zvlášť vypočítáme $\underline{\dim}_B C$ a $\overline{\dim}_B C$. V závěru se podíváme, zda se shodují.

Jako první provedeme horní odhad. Je potřeba zvolit δ a na jeho základě dopočítat $N_\delta(C)$. V k -té iteraci, kde $k = 0, 1, 2, \dots$, bude obecně 2^k intervalů, každý o délce $(1/3)^k$, tedy pokud zvolíme $3^{-k} < \delta \leq 3^{-k+1}$, pak intervaly o délce nejvýše δ (viz věta 2.4.4, bod (i)) tvoří δ pokrytí, přičemž $N_\delta(C) \leq 2^k$. Tedy celkově pro δ -pokrytí všech intervalů bude potřeba nejvýše $N_\delta(C) \leq 2^k$ intervalů $I_1, I_2, \dots, I_{N_\delta(C)}$ o průměru, tj. délce $3^{-k} < \text{diam } F_i \leq 3^{-k+1}$ pro každé i . Z toho dostáváme

$$\overline{\dim}_B C = \limsup_{\delta \rightarrow 0} \frac{\ln N_\delta(C)}{-\ln \delta} \leq \limsup_{k \rightarrow \infty} \frac{\ln 2^k}{-\ln 3^{-k+1}} = \limsup_{k \rightarrow \infty} \frac{k \ln 2}{(k-1) \ln 3} = \frac{\ln 2}{\ln 3}.$$

Naopak pokud uvážíme intervaly délky $3^{-k-1} \leq \delta < 3^{-k}$, pak každý z nich má neprázdný průnik s maximálně jedním intervalom k -té iterace C . Těch je, jak již víme, 2^k , tedy intervalů $I_1, I_2, \dots, I_{N_\delta(C)}$ bude nejméně 2^k pro pokrytí C , tzn. $N_\delta(C) \geq 2^k$. Tím dostáváme dolní odhad:

$$\underline{\dim}_B C = \liminf_{\delta \rightarrow 0} \frac{\ln N_\delta(C)}{-\ln \delta} \geq \liminf_{\delta \rightarrow 0} \frac{\ln 2^k}{-\ln 3^{-k-1}} = \liminf_{\delta \rightarrow 0} \frac{k \ln 2}{(k+1) \ln 3} = \frac{\ln 2}{\ln 3}.$$

Protože $\underline{\dim}_B C = \overline{\dim}_B C = \ln 2 / \ln 3$, tak box-counting dimenze Cantorova diskontinua je $\dim_B C = \ln 2 / \ln 3$. (Převzato z [5, str. 32])

Podobně bychom postupovali pro rovinné obrazce.

Příklad 2.4.6 (Kochova křivka). Zde se zatím s formální definicí nebudeme zatěžovat. Opět ukážeme horní a dolní odhad zvlášť. Kochovu křivku si označíme K .

Obecně k -tá iterace Kochovy křivky bude obsahovat 4^n úseček, každá o délce $(1/3)^k$. Podobně jako v předchozím příkladu 2.4.5 zvolíme $3^{-k} < \delta \leq 3^{-k+1}$. Pokud si pro pokrytí zvolíme uzavřené koule (v rovině se tedy jedná o kruhy)

$$K_\delta(x_1), K_\delta(x_2), \dots, K_\delta(x_{M_\delta(K)}), \text{ kde } x_1, x_2, \dots, x_{M_\delta(K)} \in \mathbb{R}^2,$$

pak $M_\delta(K) \leq 4^k$. Tedy

$$\overline{\dim}_B K = \limsup_{\delta \rightarrow 0} \frac{\ln M_\delta(K)}{-\ln 2\delta} \leq \limsup_{k \rightarrow \infty} \frac{\ln 4^k}{-\ln 3^{-k+1}} = \limsup_{k \rightarrow \infty} \frac{k \ln 4}{(k-1) \ln 3} = \frac{\ln 4}{\ln 3}.$$

Podobně pro dolní odhad uvážíme $3^{-k-1} \leq \delta < 3^{-k}$. Vezmeme-li uzavřené koule $K_\delta(x_1), K_\delta(x_2), \dots, K_\delta(x_{M_\delta(K)})$, pak žádná nemůže mít neprázdný průnik s více než čtyřmi úsečkami k -té iterace, a lze tedy snadno odvodit, že pro jejich pokrytí je zapotřebí alespoň $M_\delta(K) \geq 4^k/4 = 4^{k-1}$, čímž dostáváme

$$\underline{\dim}_B K = \liminf_{\delta \rightarrow 0} \frac{\ln M_\delta(K)}{-\ln 2\delta} \geq \liminf_{k \rightarrow \infty} \frac{\ln 4^{k-1}}{-\ln 3^{-k-1}} = \liminf_{k \rightarrow \infty} \frac{(k-1) \ln 4}{(k+1) \ln 3} = \frac{\ln 4}{\ln 3}.$$

Tzn. $\dim_B K = \ln 4 / \ln 3$.

Poznámka 2.4.7. Obecně množina F skládající se z m disjunktních kopií sebe samotné, kde každá z nich je r -krát menší, má dimenzi $\dim_B F = \ln m / \ln r$. [4]

Nyní se podíváme ještě na jedno možné pojetí box-counting dimenze. Připomeňme, že δ -okolím množiny F v metrickém prostoru (X, ϱ) rozumíme

$$(F)_\delta = \{x \in X \mid \exists y \in X : \varrho(x, y) < \delta\}.$$

Budeme nyní sledovat, jak „rychle“ se mění objem $(F)_\delta$ pro $\delta \rightarrow 0$. A se zmínkou o objemu nám zde do hry opět vstupuje Lebesgueova míra λ_n , o níž jsme si povídali v sekci 2.3. Podívejme se nejdříve na několik příkladů v \mathbb{R}^3 , kde si probereme situaci po řadě u množin dimenzí 0, 1, 2 a 3.

- Pro n -prvkovou množinu $F = \{x_1, x_2, \dots, x_n\} \subset \mathbb{R}^3$ je

$$\lambda_3((F)_\delta) \leq n \cdot \frac{4}{3}\pi\delta^3.$$

Pro $\delta \leq 1/2 \min \{\varrho(x, y) \mid x, y \in F\}$ nastává rovnost.

- Pro úsečku $U \subset \mathbb{R}^3$ o délce ℓ lze objem jejího δ -okolí stanovit jako

$$\lambda_3((U)_\delta) = \frac{4}{3}\pi\delta^3 + \pi\delta^2\ell.$$

Pokud však uvážíme δ dostatečně malé, lze první člen zanedbat a psát

$$\lambda_3((U)_\delta) \approx \pi\ell\delta^2.$$

- V případě neprázdného obdélníku

$$I = \{(x, y, 0) \mid x \in \langle a_1, b_1 \rangle, y \in \langle a_2, b_2 \rangle\}$$

dostáváme

$$\begin{aligned}\lambda_3(I_\delta) &= 2(b_1 - a_1)(a_2 - b_2)\delta + 2(b_1 - a_1)\pi\delta^2 + 2(b_2 - a_2)\pi\delta^2 + \frac{4}{3}\pi\delta^3 \\ &\approx 2(b_1 - a_1)(a_2 - b_2)\delta\end{aligned}$$

- Pro kouli $B_r(x) \subset \mathbb{R}^3$, kde $x \in \mathbb{R}^3$ a $r > 0$, je objem

$$\lambda_3((B_r(x))_\delta) = \frac{4}{3}\pi(r + \delta)^3 = \frac{4}{3}\pi r^3 + 4\pi r^2\delta + 4\pi r\delta^2 + \frac{4}{3}\pi\delta^3 \approx \frac{4}{3}\pi r^3.$$

Změna objemu při velmi malém δ je v tomto případě vzhledem k původnímu objemu zanedbatelná.

Výsledky si srovnejme v tabulce 2.1. Můžeme si všimnout, že ve všech přípa-

Útvar F	$\lambda_3((F)_\delta)$
Konečná množina $\{x_1, \dots, x_n\}$	$\frac{4n}{3}\pi\delta^3 = c_1\delta^3$
Úsečka U	$\pi\ell\delta^2 = c_2\delta^2$
Kvádr I	$2(b_1 - a_1)(a_2 - b_2)\delta = c_3\delta^1$
Koule $B_r(x)$	$\frac{4}{3}\pi r^3 = c_4\delta^0$

Tabulka 2.1: Odhadování λ_3 pro vybrané útvary

dech odhad objemu vychází $\lambda_3((F)_\delta) \approx c\delta^{3-s}$, kde $c > 0$ je závislé na původní „velikosti“ F a s udává dimenzi F . Obecněji pro množinu $F \subseteq \mathbb{R}^n$ bychom došli k $\lambda_n((F)_\delta) \approx c\delta^{n-s}$. Nyní, podobně jako v úvodu této sekce, zkusme opět vyjádřit s :

$$\begin{aligned}\ln \lambda_n((F)_\delta) &\approx \ln c + (n - s) \ln \delta \\ s \ln \delta &\approx n \ln \delta - \ln \lambda_n((F)_\delta) + \ln c \\ s &\approx n - \frac{\ln \lambda_n((F)_\delta)}{\ln \delta} + \frac{\ln c}{\ln \delta}.\end{aligned}$$

Poslední člen bude v limitě opět nulový.

Lze ukázat, že s není v tomto případě nic jiného, než již námi zkoumaná box-counting dimenze. To si shrneme a dokážeme v následující větě.

Věta 2.4.8. *Nechť $F \subseteq \mathbb{R}^n$. Pak platí:*

$$(i) \underline{\dim}_B F = n - \limsup_{\delta \rightarrow 0} \frac{\ln \lambda_n((F)_\delta)}{\ln \delta},$$

$$(ii) \overline{\dim}_B F = n - \liminf_{\delta \rightarrow 0} \frac{\ln \lambda_n((F)_\delta)}{\ln \delta}.$$

Důkaz. V rámci důkazu využijeme větu 2.4.4.

Mějme $F \subseteq \mathbb{R}^n$. Označme v_n objem jednotkové koule $K_1(x)$ ¹³ v \mathbb{R}^n pro $x \in \mathbb{R}^n$ libovolné. Dále mějme pokrytí

$$\mathcal{K} = \left\{ K_\delta(x_1), K_\delta(x_2), \dots, K_\delta(x_{M_\delta(F)}) \right\}$$

množiny F , kde $0 < \delta < 1$ a $x_j \in \mathbb{R}^n$ pro každé $1 \leq j \leq M_\delta(F)$, přičemž $M_\delta(F)$ je definováno podle bodu (i) věty 2.4.4. Pak lze zvolit pokrytí

$$\mathcal{K}' = \left\{ K_{2\delta}(x_1), K_{2\delta}(x_2), \dots, K_{2\delta}(x_{M_\delta(F)}) \right\},$$

tzn. \mathcal{K} je zjemnění pokrytí \mathcal{K}' . Zároveň však platí, že \mathcal{K}' je i pokrytím $(F)_\delta$. Pro libovolné $x \in (F)_\delta$ existuje totiž $y \in F$ takové, že $\varrho(x, y) < \delta$. Ovšem pro toto y existuje nějaká koule $K_\delta(x_\ell) \in \mathcal{K}$ taková, že $y \in K_\delta(x_\ell)$, což znamená, že

$$\varrho(x_\ell, x) \leq \varrho(x_\ell, y) + \varrho(y, x) \leq \delta + \delta = 2\delta.$$

a tedy $x \in K_{2\delta}(x_\ell)$. Tzn. míru F lze shora odhadnout jako

$$\lambda_n((F)_\delta) \leq M_\delta(F)v_n(2\delta)^n,$$

kde v_n je (pro připomenutí) objem jednotkové koule. Úpravou získáme:

$$\begin{aligned} \ln \lambda_n((F)_\delta) &\leq n \ln \delta + \ln M_\delta(F) + \ln 2^n v_n, \\ \frac{\ln \lambda_n((F)_\delta)}{-\ln \delta} &\leq -n + \frac{\ln M_\delta(F)}{-\ln \delta} + \frac{\ln 2^n v_n}{-\ln \delta}, \end{aligned}$$

tedy v limitě

$$\begin{aligned} \liminf_{\delta \rightarrow 0} \frac{\ln \lambda_n((F)_\delta)}{-\ln \delta} &\leq -n + \underline{\dim}_B F \\ \underline{\dim}_B F &\geq n - \limsup_{\delta \rightarrow 0} \frac{\ln \lambda_n((F)_\delta)}{\ln \delta}. \end{aligned}$$

Nyní uvažujme po dvou disjunktní otevřené koule $B_\delta^j(x_j)$, kde $x_j \in F$ pro $1 \leq j \leq M_\delta(F)$. Pak součtem jejich objemů získáme

$$M_\delta(F)v_n\delta^n \leq \lambda_n((F)_\delta).$$

Obdobnou úpravou této nerovnosti získáme opačnou nerovnost. K odhadu $\overline{\dim}_B F$ lze dospět analogicky. \square

Zkusme si aplikaci věty ilustrovat opět na příkladu fraktálu.

¹³Objem koule v \mathbb{R}^n lze vyjádřit vztahem

$$V_n(r) = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2} + 1)} r^n,$$

kde Γ je tzv. *gamma funkce*. Se vzorcem však dále v textu pracovat nebudeme.

Příklad 2.4.9 (Cantorovo diskontinuum potřetí). Pro Cantorovo diskontinuum v k -té iteraci, označme C_k , lze odhadnout délku $(C_k)_\delta$ pro $3^{-k-2} \leq \delta \leq 3^{-k-1}$ jako

$$\lambda_1((C_k)_\delta) \geq 2^k(3^{-k-2} + 2\delta) \leq 2^k \cdot 3 \cdot 3^{-k-2} = 2^k \cdot 3^{-k-1}.$$

Tedy podle věty 2.4.8

$$\begin{aligned}\overline{\dim}_B C &= n - \liminf_{\delta \rightarrow 0} \frac{\ln \lambda_1((C)_\delta)}{\ln \delta} \leq 1 - \liminf_{k \rightarrow \infty} \frac{\ln (2^k \cdot 3^{-k-1})}{\ln 3^{-k-1}} \\ &= \limsup_{k \rightarrow \infty} \frac{k \ln 2}{(k+1) \ln 3} = \frac{\ln 2}{\ln 3}.\end{aligned}$$

Podobně zvolíme-li $3^{-k-1} \leq \delta \leq 3^{-k}$, pak

$$\lambda_1((C_k)_\delta) \leq 2^k(3^{-k} + 2\delta) \leq 2^k \cdot 3 \cdot 3^{-k} = 2^k \cdot 3^{-k+1},$$

a tedy

$$\begin{aligned}\underline{\dim}_B C &= n - \limsup_{\delta \rightarrow 0} \frac{\ln \lambda_1((C)_\delta)}{\ln \delta} \geq 1 - \limsup_{k \rightarrow \infty} \frac{\ln 2^k \cdot 3^{-k+1}}{\ln 3^{-k+1}} \\ &= \liminf_{k \rightarrow \infty} \frac{k \ln 2}{(k-1) \ln 3} = \frac{\ln 2}{\ln 3}.\end{aligned}$$

2.4.2 Vlastnosti

V minulé podsekci 2.4.1 jsme si povídali o možnostech pojetí box-counting dimenze. S tím pak souvisely zejména věty 2.4.4 a 2.4.8. Nyní trochu ještě blíže prozkoumáme některé její vlastnosti, na něž se podíváme ve větě 2.4.10.

Věta 2.4.10 (Vlastnosti box-counting dimenze). *Nechť jsou dány $F, G \subseteq \mathbb{R}^n$.*

- (i) Pokud $G \subseteq F$, pak $\underline{\dim}_B G \leq \underline{\dim}_B F$ a $\overline{\dim}_B G \leq \overline{\dim}_B F$. \triangleleft monotonie
- (ii) Je-li $F \neq \emptyset$ omezená, pak $0 \leq \underline{\dim}_B F \leq \overline{\dim}_B F \leq n$. \triangleleft rozsah hodnot
- (iii) $\overline{\dim}_B(F \cup G) = \max \{\overline{\dim}_B F, \overline{\dim}_B G\}$. \triangleleft stabilita

Důkaz.

- (i) Plyne triviálně z faktu, že pro libovolné $\delta > 0$ je $N_\delta(G) \leq N_\delta(F)$, neboť každé δ -pokrytí $\mathcal{F} \supset F$ je zároveň δ -pokrytím G .
- (ii) První dvojice nerovností je zjevná z definice (viz 2.4.2). Pro třetí nerovnost zvolme kvádr I takový, že $F \subset I$. Zvolíme-li $\delta > 0$ a δ -mříž \mathcal{Q}_δ , pak

$$\begin{aligned}M_\delta(F) &= |\{J \mid J \cap F \neq \emptyset, J \in \mathcal{Q}_\delta\}| \\ &\leq |\{J \mid J \cap I \neq \emptyset, J \in \mathcal{Q}_\delta\}| \\ &= M_\delta(I) \leq c\delta^{-n},\end{aligned}$$

kde $c > 0$. Poslední nerovnost v odhadu výše. Tedy podle věty 2.4.4 a přechozího bodu (i) máme

$$\overline{\dim}_B F \leq \overline{\dim}_B I = \limsup_{\delta \rightarrow 0} \frac{\ln N_\delta(I)}{-\ln \delta} \leq \limsup_{\delta \rightarrow 0} \frac{\ln c\delta^{-n}}{-\ln \delta} = n.$$

(iii) Pro $\delta > 0$ volme δ -pokrytí $\mathcal{F} \supset F$ a $\mathcal{G} \supset G$. Je celkem zjevné, že $N_\delta(F \cup G) \leq N_\delta(F) + N_\delta(G)$, neboli

$$\begin{aligned}\ln(N_\delta(F) + N_\delta(G)) &\leq \ln(2 \max\{N_\delta(F), N_\delta(G)\}) \\ &= \ln 2 + \ln(\max\{N_\delta(F), N_\delta(G)\}).\end{aligned}$$

Tedy

$$\begin{aligned}\overline{\dim}_B(F \cup G) &\leq \limsup_{\delta \rightarrow 0} \left(\frac{\ln 2}{-\ln \delta} + \frac{\ln(\max\{N_\delta(F), N_\delta(G)\})}{-\ln \delta} \right) \\ &\leq \limsup_{\delta \rightarrow 0} \frac{\ln(\max\{N_\delta(F), N_\delta(G)\})}{-\ln \delta} \\ &= \limsup_{\delta \rightarrow 0} \left(\max \left\{ \frac{\ln N_\delta(F)}{-\ln \delta}, \frac{\ln N_\delta(G)}{-\ln \delta} \right\} \right) \\ &\leq \max \left\{ \limsup_{\delta \rightarrow 0} \frac{\ln N_\delta(F)}{-\ln \delta}, \limsup_{\delta \rightarrow 0} \frac{\ln N_\delta(G)}{-\ln \delta} \right\} \\ &= \max \left\{ \overline{\dim}_B F, \overline{\dim}_B G \right\}.\end{aligned}$$

Opačná nerovnost plyne z faktu, že $F \subset F \cup G$ a $G \subset F \cup G$, tedy

$$\overline{\dim}_B(F \cup G) \geq \overline{\dim}_B F \text{ a } \overline{\dim}_B(F \cup G) \geq \overline{\dim}_B G$$

podle bodu (i), neboli

$$\overline{\dim}_B(F \cup G) = \max \left\{ \overline{\dim}_B F, \overline{\dim}_B G \right\}.$$

□

(Převzato a upraveno z [5, str. 35].)

Poslední bod (iii) tvrzení 2.4.10 lze pochopitelně rozšířit indukcí. Čtenář se sám může přesvědčit, že se jedná o relativně jednoduché cvičení.

Důsledek 2.4.11. Pro $F_1, F_2, \dots, F_m \subseteq \mathbb{R}^n$ platí:

$$\overline{\dim}_B \left(\bigcup_{i=1}^m F_i \right) = \max \left\{ \overline{\dim}_B F_j \mid 1 \leq j \leq m \right\}.$$

Důkaz. Pro $m = 1$ a $m = 2$ víme, že tvrzení platí. Pro $m + 1$ lze psát:

$$\begin{aligned}\overline{\dim}_B \left(\bigcup_{i=1}^{m+1} F_i \right) &= \overline{\dim}_B \left(\left(\bigcup_{i=1}^m F_i \right) \cup F_{m+1} \right) \\ &= \max \left\{ \overline{\dim}_B \left(\bigcup_{i=1}^{m+1} F_i \right), \overline{\dim}_B F_{m+1} \right\} \\ &\stackrel{\text{I.P.}}{=} \max \left\{ \max \left\{ \overline{\dim}_B F_i \mid 1 \leq i \leq m \right\}, \overline{\dim}_B F_{m+1} \right\} \\ &= \max \left\{ \overline{\dim}_B F_j \mid 1 \leq j \leq m + 1 \right\}.\end{aligned}$$

□

Jako poslední se ještě nabízí otázka, jak se bude dimenze \dim_B chovat vůči zobrazením. V tomto kontextu pro nás budou relevantní především *lipschitzovská* a *bilipschitzovská zobrazení*. Připomeňme, že lipschitzovské zobrazení je takové zobrazení $f : X \rightarrow Y$ mezi metrickými prostory (X, ϱ_1) a (Y, ϱ_2) , že existuje konstanta $K > 0$ taková, že pro každé $x, y \in X$ platí

$$\varrho_2(f(x), f(y)) \leq K\varrho_1(x, y).$$

Pokud navíc platí, že existují konstanty $K_1, K_2 > 0$ takové, že platí

$$K_1\varrho_1(x, y) \leq \varrho_2(f(x), f(y)) \leq K_2\varrho_1(x, y),$$

pak f nazýváme bilipschitzovské.

Než se však podíváme na samotný vztah box-counting dimenze a lipschitzovských, resp. bilipschitzovských zobrazení, dokážeme si jedno jednoduché lemma, které později využijeme.

Lemma 2.4.12. *Nechť $(X, \varrho_1), (Y, \varrho_2)$ jsou metrické prostory a zobrazení $f : X \rightarrow Y$ je bilipschitzovské. Pak $f : X \rightarrow f(X)$ je prosté.*

Důkaz. Podle předpokladu je f bilipschitzovské zobrazení, tedy existují konstanty $K_1, K_2 > 0$ takové, že

$$K_1\varrho_1(x, y) \leq \varrho_2(f(x), f(y)) \leq K_2\varrho_1(x, y), \quad x, y \in X.$$

Surjektivita zobrazení f je zřejmá z její definice. Nechť existuje dvojice různých bodů $x, y \in X$. Pak

$$0 < K_1\varrho_1(x, y) \leq \varrho_2(f(x), f(y)),$$

pročež $f(x) \neq f(y)$. □

V našem případě se dále omezíme, stejně jako předtím, pouze na prostor \mathbb{R}^n .

Věta 2.4.13. *Nechť jsou dány metrické prostory $(\mathbb{R}^n, \varrho_n)$ a $(\mathbb{R}^m, \varrho_m)$, kde ϱ_n, ϱ_m jsou metriky, $F \subseteq \mathbb{R}^n$ a zobrazení $f : F \rightarrow \mathbb{R}^m$. Platí:*

(i) *Je-li f lipschitzovské, pak*

$$\underline{\dim}_B f(F) \leq \underline{\dim}_B F \text{ a } \overline{\dim}_B f(F) \leq \overline{\dim}_B F.$$

(ii) *Je-li f bilipschitzovské, pak*

$$\underline{\dim}_B f(F) = \underline{\dim}_B F \text{ a } \overline{\dim}_B f(F) = \overline{\dim}_B F.$$

Důkaz. Máme tedy metrické prostory $(\mathbb{R}^n, \varrho_n)$, $(\mathbb{R}^m, \varrho_m)$, zobrazení $f : F \rightarrow \mathbb{R}^m$ a $F \subseteq \mathbb{R}^n$.

(i) Jako první si všimneme, že je-li $\mathcal{F} = \{F_1, F_2, \dots\}$ δ -pokrytí množiny F , kde $\delta > 0$, pak je jím i systém

$$\mathcal{F}' = \{F \cap F_1, F \cap F_2, \dots\}.$$

Podle předpokladu je f lipschitzovské, tzn. pro každé $x, y \in \mathbb{R}^n$ je

$$\varrho_m(f(x), f(y)) \leq K \varrho_n(x, y), \quad K > 0.$$

Speciálně tak platí i $\text{diam}(f(F \cap F_i)) \leq K \text{diam}(F \cap F_i)$ pro každé i , a tedy

$$\text{diam}(f(F \cap F_i)) \leq K \text{diam}(F \cap F_i) \leq K \text{diam } F_i \leq K\delta.$$

Z toho plyne, že $\mathcal{G} = \{f(F \cap F_1), f(F \cap F_2), \dots\}$ tvoří $K\delta$ -pokrytí množiny $f(F)$. Tedy máme, že $N_{K\delta}(f(F)) \leq N_\delta(F)$. Po úpravě

$$\frac{\ln N_{K\delta}(f(F))}{-\ln \delta} \leq \frac{\ln N_\delta(F)}{-\ln \delta}.$$

Tedy celkově

$$\begin{aligned} \overline{\dim}_B f(F) &= \limsup_{\delta \rightarrow 0} \frac{\ln N_{K\delta}(f(F))}{-\ln K\delta} = \limsup_{\delta \rightarrow 0} \frac{\ln N_{K\delta}(f(F))}{-\ln \delta} \cdot \frac{\ln \delta}{\ln K\delta} \\ &\leq \limsup_{\delta \rightarrow 0} \frac{\ln N_\delta(f(F))}{-\ln \delta} \cdot \frac{\ln \delta}{\ln K\delta} = \limsup_{\delta \rightarrow 0} \frac{\ln N_\delta(f(F))}{-\ln \delta} \\ &= \overline{\dim}_B F. \end{aligned}$$

Nerovnost pro $\underline{\dim}_B f(F)$ získáme analogicky.

(ii) Je-li f bilipschitzovské, pak podle lemmatu 2.4.12 je $f : F \rightarrow f(F)$ prostá, a tedy existuje inverzní zobrazení $f^{-1} : f(F) \rightarrow F$. Volme $u, v \in f(F)$ libovolně a položme $x = f^{-1}(u), y = f^{-1}(v)$. Pak

$$\begin{aligned} K_1 \varrho_n(x, y) &= K_1 \varrho_n(f^{-1}(u), f^{-1}(v)) \leq \varrho_m(f(f^{-1}(u)), f(f^{-1}(v))) \\ &= \varrho_m(u, v), \end{aligned}$$

neboli

$$\varrho_n(f^{-1}(u), f^{-1}(v)) \leq \frac{1}{K_1} \varrho_m(u, v),$$

přičemž K_1, K_2 jsou konstanty z definice. Tzn. f^{-1} je lipschitzovské.

Podle bodu (i) tedy platí

$$\begin{aligned} \underline{\dim}_B f(F) &= \underline{\dim}_B f^{-1}(f(F)) \leq \underline{\dim}_B f(F), \\ \overline{\dim}_B f(F) &= \overline{\dim}_B f^{-1}(f(F)) \leq \overline{\dim}_B f(F). \end{aligned}$$

Ovšem podle bodu (i) ovšem již víme, že také platí

$$\begin{aligned} \underline{\dim}_B f(F) &\leq \underline{\dim}_B F, \\ \overline{\dim}_B f(F) &\leq \overline{\dim}_B F. \end{aligned}$$

Z toho již plyne závěr tvrzení. □

(Převzato z [5, str. 36].)

Právě dokázaná věta 2.4.13 (konkrétně bod (ii)) nám ve své podstatě říká, že box-counting dimenze nějakého útvaru F je invariantní vůči libovolnému bilipschitzovskému zobrazení f . Tento výsledek se nám bude hodit dále v kapitole 4 u tzv. systémů iterovaných funkcí.

2.5 Hausdorffova míra a Hausdorffova dimenze

Způsobů, jak definovat dimenzi, je celá řada. Zatím jsme společně prozkoumali box-counting dimenzi (resp. některá její pojetí), avšak lze najít více způsobů, jak koncept dimenze uchopit¹⁴. Pravděpodobně asi nejstarším exemplářem svého druhu je tzv. *Hausdorffova dimenze* a s ní související *Hausdorffova míra*, které hrají ve fraktální geometrii velice podstatnou roli. Stále se však budeme zabývat pouze množinami v \mathbb{R}^n . Zmíněná dimenze a míra nesou jméno po německém matematikovi FELIXI HAUSDORFFOVI¹⁵.



Obrázek 2.6: Felix Hausdorff, 1868–1942 (Převzato z [15])

2.5.1 Definice Hausdorffovy míry

Definice 2.5.1. Nechť je dána množina $F \subseteq \mathbb{R}^n$ a $s \geq 0$. Pak pro každé $\delta > 0$ definujeme

$$\mathcal{H}_\delta^s(F) = \inf \left\{ \sum_{i=1}^{\infty} (\text{diam } F_i)^s \mid F \subseteq \bigcup_{i=1}^{\infty} F_i, \text{ diam } F_j \leq \delta \text{ pro } j \in \mathbb{N} \right\}.$$

Infimum se zde dělá přes všechna spočetná pokrytí F množinami o průměru nejvýše δ . Lze si všimnout, že pro $0 < \delta_1 < \delta_2$ je $\mathcal{H}_{\delta_1}^s(F) \geq \mathcal{H}_{\delta_2}^s(F)$. Jinými slovy funkce $\delta \mapsto \mathcal{H}_\delta^s(M)$ je nerostoucí. Pro menší δ počítáme infimum z „menší“ množiny, protože kritérium na přípustná pokrytí je „přísnější“. To znamená, že

$$\begin{aligned} \mathcal{H}_{\delta_1}^s(F) &= \inf \left\{ \sum_{U \in \mathcal{F}_1} (\text{diam } U)^s \mid \mathcal{F}_1 \text{ je } \delta_1\text{-pokrytí } F \right\} \\ &\geq \inf \left\{ \sum_{U \in \mathcal{F}_2} (\text{diam } U)^s \mid \mathcal{F}_2 \text{ je } \delta_2\text{-pokrytí } F \right\} = \mathcal{H}_{\delta_2}^s(F). \end{aligned}$$

Zároveň je z definice 2.5.1 zjevné, že $\mathcal{H}_\delta^s(F) \geq 0$.

¹⁴Některé další jsou sepsány např. v [5, str. 40].

¹⁵1868–1942

Definice 2.5.2 (Hausdorffova míra). Nechť $F \subseteq \mathbb{R}^n$ a $s \geq 0$. Pak pro množinu F definujeme *s-dimenzionální Hausdorffovu míru* jako

$$\mathcal{H}^s(F) = \lim_{\delta \rightarrow 0} \mathcal{H}_\delta^s(F).$$

Z přechodzího je zjevné, že limita v definici 2.5.2 vždy existuje, protože funkce $\delta \mapsto \mathcal{H}_\delta^s(F)$ je monomónní.

Bude dobré se přesvědčit, že je Hausdorffova míra mírou ve smyslu definice 2.2.4. Začneme však otázkou. *Na jaké σ -algebře je potřeba Hausdorffovu míru \mathcal{H}^s uvažovat?* Odpověď nám poskytují tzv. *borelovské množiny*, které jsou pojmenovány po francouzském matematikovi ÉMILE BORELOVI¹⁶. Borelovské množiny jsou zá-



Obrázek 2.7: Émile Borel, 1871–1956 (Převzato z [16])

kladním pojmem v tzv. *Deskriptivní teorii množin*. Pojdme si vysvětlit alespoň základ. Nechť (X, ρ) je metrický prostor. Pak borelovskými množinami nazýváme všechny množiny, které lze získat iteracemi operací spočetného sjednocení, průniku a doplnku otevřených množin z X . Označme systém takových množin jako \mathcal{G} . Na tomto základě pak definujeme tzv. σ -algebру borelovských množin na X :

$$\mathcal{B}(X) = \bigcap_{\substack{\mathcal{F} \supseteq \mathcal{G} \\ \mathcal{F} \text{ je } \sigma\text{-algebra}}} \mathcal{F}.$$

Jinými slovy, $\mathcal{B}(X)$ je nejmenší σ -algebra generovaná¹⁷ všemi otevřenými množinami z X .

Nás speciálně bude zajímat σ -algebra $\mathcal{B}(\mathbb{R}^n)$. Nejdříve si však dokážeme dvě pomocná lemmata.

¹⁶1871–1956

¹⁷Obecně σ -algebra \mathcal{A} je generovaná množinou X , když

$$\mathcal{A} = \bigcap_{\substack{\mathcal{F} \supseteq X \\ \mathcal{F} \text{ je } \sigma\text{-algebra}}} \mathcal{F}.$$

Tento fakt se někdy značí $\mathcal{A} = \sigma(X)$. Lze totiž ukázat, že průnik σ -algeber je opět σ -algebrou.

Lemma 2.5.3 (σ -subaditivita Hausdorffovy míry). *Nechť jsou dány množiny A_1, A_2, \dots , kde $A_i \subseteq X$ pro každé $i \in \mathbb{N}$. Pak pro každé $s \geq 0$ platí*

$$\mathcal{H}^s\left(\bigcup_{i=1}^{\infty} A_i\right) \leq \sum_{i=1}^{\infty} \mathcal{H}^s(A_i).$$

Důkaz. Nechť $s \geq 0$ a dále budiž dáno $\varepsilon > 0$. Pro každé $i \in \mathbb{N}$ a $\delta > 0$ mějme pokrytí

$$\mathcal{F}_i = \{F_{i,1}, F_{i,2}, \dots\}$$

množiny A_i takové, že platí

$$\sum_{j=1}^{\infty} (\text{diam } F_{i,j})^s \leq \mathcal{H}_{\delta}^s(A_i) + \frac{\varepsilon}{2^i}.$$

Systém $\bigcup_{i=1}^{\infty} \mathcal{F}_i$ tedy tvoří δ -pokrytí množiny $A = \bigcup_{i=1}^{\infty} A_i$. Celkově

$$\begin{aligned} \mathcal{H}_{\delta}^s\left(\bigcup_{i=1}^{\infty} A_i\right) &\leq \sum_{i,j \in \mathbb{N}} (\text{diam } F_{i,j})^s = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} (\text{diam } F_{i,j})^s \leq \sum_{i=1}^{\infty} \left(\mathcal{H}_{\delta}^s(A_i) + \frac{\varepsilon}{2^i} \right) \\ &= \sum_{i=1}^{\infty} \mathcal{H}_{\delta}^s(A_i) + \varepsilon. \end{aligned}$$

Limitním přechodem $\delta \rightarrow 0$ a aplikací Leviho věty¹⁸ dostáváme

$$\begin{aligned} \mathcal{H}^s\left(\bigcup_{i=1}^{\infty} A_i\right) &= \lim_{\delta \rightarrow 0} \mathcal{H}_{\delta}^s\left(\bigcup_{i=1}^{\infty} A_i\right) \leq \lim_{\delta \rightarrow 0} \sum_{i=1}^{\infty} \mathcal{H}_{\delta}^s(A_i) + \varepsilon = \sum_{i=1}^{\infty} \lim_{\delta \rightarrow 0} \mathcal{H}_{\delta}^s(A_i) + \varepsilon \\ &= \sum_{i=1}^{\infty} \mathcal{H}^s(A_i) + \varepsilon. \end{aligned}$$

□

Lemma 2.5.4. *Nechť (X, ϱ) je metrický prostor, kde $X \subseteq \mathbb{R}^n$, ϱ je metrika, a $A, B \subseteq X$ takové, že pro jejich vzdálenost platí $\varrho(A, B) > 0$. Pak pro každé $s \geq 0$ platí*

$$\mathcal{H}^s(A \cup B) = \mathcal{H}^s(A) + \mathcal{H}^s(B).$$

¹⁸Leviho věta o záměně pořadí limity a Lebesgueova integrálu říká, že je-li posloupnost nezáporných měřitelných funkcí $\{f_n\}_{n=1}^{\infty}$ neklesající, tj.

$$f_1 \leq f_2 \leq \dots$$

na prostoru (X, \mathcal{A}, μ) a zároveň $\lim_{n \rightarrow \infty} f_n(x) = f(x)$ pro každé $x \in X$, pak

$$\lim_{n \rightarrow \infty} \int_X f_n \, d\mu = \int_X \lim_{n \rightarrow \infty} f_n \, d\mu.$$

Zde je speciálně μ aritmetická míra, $X = \mathbb{N}$ a $f_n(i)$ lze volit např. $\mathcal{H}_{1/n}^s(A_i)$. Z Heineho věty víme, že

$$\lim_{n \rightarrow \infty} \mathcal{H}_{1/n}^s(A_i) = \lim_{\delta \rightarrow 0} \mathcal{H}_{\delta}^s(A_i) = \mathcal{H}^s(A_i).$$

Důkaz. Nerovnost $\mathcal{H}^s(A \cup B) \leq \mathcal{H}^s(A) + \mathcal{H}^s(B)$ je zřejmá ze σ -subaditivity Hausdorffovy míry (viz lemma 2.5.3).

Bez újmy na obecnosti předpokládejme, že $\mathcal{H}^s(A \cup B) < \infty$. Mějme libovolné $\varepsilon > 0$ a $\delta < \varrho(A, B)/2$. Zvolme δ -pokrytí $\mathcal{F} = \{F_1, F_2, \dots\}$ množiny $A \cup B$ takové, že

$$\sum_{i=1}^{\infty} (\text{diam } F_i)^s \leq \mathcal{H}_{\delta}^s(A \cup B) + \varepsilon.$$

Každá z množin F_i má neprázdný průnik s nejvýše jednou z množin A, B , tzn. z pokrytí \mathcal{F} lze vybrat dva disjunktní podsystémy \mathcal{F}_A a \mathcal{F}_B , přičemž $\bigcup \mathcal{F}_A \supseteq A$ a $\bigcup \mathcal{F}_B \supseteq B$. Tedy celkově s užitím předchozího lemmatu 2.5.3 máme

$$\begin{aligned} \mathcal{H}_{\delta}^s(A) + \mathcal{H}_{\delta}^s(B) &\leq \mathcal{H}_{\delta}^s\left(\bigcup_{F \in \mathcal{F}_A} F\right) + \mathcal{H}_{\delta}^s\left(\bigcup_{F \in \mathcal{F}_B} F\right) \\ &\leq \sum_{F \in \mathcal{F}_A} (\text{diam } F)^s + \sum_{F \in \mathcal{F}_B} (\text{diam } F)^s \\ &\leq \sum_{i=1}^{\infty} (\text{diam } F_i)^s \leq \mathcal{H}_{\delta}^s(A \cup B) + \varepsilon. \end{aligned}$$

Pro $\delta \rightarrow 0$ dostáváme

$$\mathcal{H}^s(A) + \mathcal{H}^s(B) \leq \mathcal{H}^s(A \cup B) + \varepsilon$$

□

Definice 2.5.5 (Vnější míra). Necht (X, \mathcal{A}) je měřitelný prostor. Zobrazení $\mu^* : \mathcal{A} \rightarrow \langle 0, \infty \rangle$ nazveme *vnější mírou* na \mathcal{A} , pokud platí:

- (a) $\mu^*(\emptyset) = 0$;
- (b) Pokud $A, B \in \mathcal{A}$ a $A \subseteq B$, pak $\mu^*(A) \leq \mu^*(B)$;
- (c) Je-li A_1, A_2, \dots posloupnost množin, kde $A_i \in \mathcal{A}$ pro každé $i \in \mathbb{N}$, pak

$$\mu^*\left(\bigcup_{i=1}^{\infty} A_i\right) \leq \sum_{i=1}^{\infty} \mu^*(A_i).$$

Vnější míra představuje zobecnění toho, co jsme měli možnost vidět již v sekci 2.3 týkající se Lebesgueovy míry¹⁹. Tu jsme definovali na základě tzv. *vnější Lebesgueovy míry* (viz definice 2.3.1), která sice sama o sobě míru nepředstavovala, nicméně při restrikci na „správný“ systém množin jsme konstatovali, že se již jedná o míru. Lze se přesvědčit, že vnější Lebesgueova míra λ_n^* je vnější mírou na $\mathcal{P}(\mathbb{R}^n)$ ve smyslu definice 2.5.5 výše. Podobné pozorování lze učinit i pro Hausdorffovu míru \mathcal{H}^s . Platnost podmínky (c) jsme dokázali v lemmatu 2.5.3 a o platnosti (a) a (b) se může čtenář velice snadno předvědčit. Tedy Hausdorffova míra na měřitelném prostoru $(\mathbb{R}^n, \mathcal{P}(R^n))$ je vnější mírou. Navíc pokud vnější míra μ^* splňuje tvrzení lemmatu 2.5.4, pak ji nazýváme *metrickou vnější mírou*.

¹⁹Jedná se slabší požadavek, tzn. každá míra je vnější mírou, opačné tvrzení však neplatí.

Při zavádění *lebesgueovské měřitelnosti* (viz definice 2.3.5) jsme zmínili tzv. Carathéodoryho kritérium. To jednoduše říkalo, že rozdelením libovolné množiny G pomocí pevně zvolené množiny A lze stanovit její míru jako součet měr dílčích částí, tzn. $G \cap A$ a $G \setminus A$. Tento koncept lze však rozšířit. Obecně jakákoli množina je μ -měřitelná, pokud splňuje Carathéodoryho kritérium.

Definice 2.5.6. Nechť X je množina, μ je vnější míra na $\mathcal{P}(X)$ a $A \subseteq X$. Množina A je μ -měřitelná, pokud pro každé $G \subseteq X$ platí

$$\mu(G) = \mu(G \cap A) + \mu(G \setminus A).$$

Nyní ukážeme platnost následující věty.

Věta 2.5.7. Nechť (X, ϱ) je metrický prostor, kde $X \subseteq \mathbb{R}^n$. Pak každá množina $A \in \mathcal{B}(X)$ je \mathcal{H}^s -měřitelná pro každé $s \geq 0$.

Důkaz. Není těžké si rozmyslet, že $\mathcal{B}(X)$ obsahuje též všechny uzavřené²⁰. Volme tedy uzavřenou množinu A , libovolnou množinu $G \subseteq X$ a $s \geq 0$. Ze σ -subadditivity plyne nerovnost

$$\mathcal{H}^s(G) \leq \mathcal{H}^s(G \cap A) + \mathcal{H}^s(G \setminus A).$$

Pro důkaz opačné nerovnosti definujeme posloupnost množin P_0, P_1, P_2, \dots následovně:

$$\begin{aligned} P_0 &= \{x \in G \mid \varrho(x, A) \geq 1\}, \\ P_i &= \left\{x \in G \mid \frac{1}{i+1} \leq \varrho(x, A) \leq \frac{1}{i}\right\}, \quad i \geq 1. \end{aligned}$$

Pro libovolnou dvojici množin z podposloupnosti P_0, P_2, P_4, \dots platí, že jejich vzdálenosti jsou kladné. Z faktu, že \mathcal{H}^s je metrická (viz lemma 2.5.4) a z monotonie plyne

$$\sum_{i=1}^m \mathcal{H}^s(P_{2i}) = \mathcal{H}^s\left(\bigcup_{i=0}^m P_{2i}\right) \leq \mathcal{H}^s(G)$$

pro všechna $m \in \mathbb{N}$. Podobně pro liché členy $\sum_{i=0}^m \mathcal{H}^s(P_{2i+1}) \leq \mathcal{H}^s(G)$. Tzn. řada $\sum_{i=0}^{\infty} \mathcal{H}^s(P_i)$ je konvergentní. Zároveň platí

$$\varrho\left(\bigcup_{i=0}^m P_i, G \cap A\right) > 0$$

pro každé $m \in \mathbb{N}$, tedy lze psát

$$\begin{aligned} \mathcal{H}^s(G \setminus A) &\leq \mathcal{H}^s\left(\bigcup_{i=0}^m P_i\right) + \mathcal{H}^s\left(\bigcup_{i=m+1}^{\infty} P_i\right) \\ &\leq \mathcal{H}^s(G) - \mathcal{H}^s(G \cap A) + \sum_{i=m+1}^{\infty} \mathcal{H}^s(P_i). \end{aligned}$$

První nerovnost plyne z faktu, že množiny P_i pokrývají $G \setminus A$, protože A je uzavřená množina. Pro $m \rightarrow \infty$ dostáváme

$$\mathcal{H}^s(G \setminus A) \leq \mathcal{H}^s(G) - \mathcal{H}^s(G \cap A)$$

²⁰Plyne z uzavřenosti na doplněk.

z čehož již plyne požadovaná nerovnost.

Lze ukázat, že všechny \mathcal{H}^s -měřitelné množiny (ve smyslu Carathéodoryho kritéria) tvoří σ -algebru. Zároveň nyní víme, že všechny uzavřené množiny jsou \mathcal{H}^s -měřitelné, tedy všechny borelovské množiny jsou měřitelné. \square

Důsledek 2.5.8. *Trojice $(X, \mathcal{B}(X), \mathcal{H}^s)$, kde X je libovolná množina a $s \geq 0$, tvoří prostor s mírou.*

Nyní již můžeme zobrazení \mathcal{H}^s nazývat mírou oprávněně. Pojdme se podívat na nějaké příklady.

Příklad 2.5.9. Pro $s = 0$ představuje zobrazení \mathcal{H}^s obyčejnou aritmetickou míru, tzn. pro konečnou množinu $A \subseteq \mathbb{R}^n$ je $\mathcal{H}^0(A) = |A|$. Toto není těžké předvést. Mějme n -prvkovou množinu $A = \{x_1, x_2, \dots, x_n\}$. Zvolíme-li

$$\delta < \frac{1}{2} \cdot \min \{\varrho_e(x_i, x_j) \mid 1 \leq i, j \leq n\},$$

pak pro δ -pokrytí $\mathcal{F} = \{F_1, F_2, \dots, F_n\}$ takové, že $x_i \in F_i$ pro každé i , máme

$$\sum_{i=1}^n (\text{diam } F_i)^0 = \sum_{i=1}^n 1 = n.$$

Není těžké si rozmyslet, že n je nejmenší počet množin o průměru nejvýše δ potřebných pro pokrytí množiny A . Zároveň pro libovolné $\varepsilon > 0$ je potřeba nejvýše n -koulí o poloměru $\varepsilon/2$ se středy v x_i pro pokrytí A . Tzn. $\mathcal{H}^0(A) = |A| = n$.

V rámci tohoto textu jsme se již zabývali jiným typem míry, a to tzv. *Lebesgueovou mírou* (viz sekce 2.3). Ta pro nás hrála důležitou roli v jednom z možných pojetí *box-counting dimenze* (viz sekce 2.4). Lze ukázat, že pro borelovskou množinu $F \subseteq \mathbb{R}^n$ je

$$\mathcal{H}^n(F) = \frac{1}{v_n} \lambda_n(F),$$

kde v_n je objem (míra) jednotkové koule v \mathbb{R}^n . Čtenář snad promine, že tento fakt zde ponecháme bez důkazu. [5, str. 45]

2.5.2 Stručně k vlastnostem Hausdorffovy míry

Na chvíli se ještě zastavíme u vlastností Hausdorffovy míry. Již jsme společně dokázali, že Hausdorffova míra je skutečně mírou, tzn. splňuje všechny základní vlastnosti, které jsme si představili ve větě 2.2.6 (viz sekce 2.2). V tomto ohledu tedy netřeba již nic dalšího dokazovat. Nicméně podobně jako v případě *box-counting dimenze* (viz podsekce 2.4.2) se i zde podíváme, jak se Hausdorffova míra chová vůči *lipschitzovským zobrazením*.

Věta 2.5.10. *Nechť $F \subseteq \mathbb{R}^n$ v metrickém prostoru (\mathbb{R}^n, ϱ) , kde ϱ je metrika, a zobrazení $f : F \rightarrow \mathbb{R}^n$ je lipschitzovské²¹ s konstantou $K > 0$. Pak pro každé $s \geq 0$ platí*

$$\mathcal{H}^s(f(F)) \leq K^s \mathcal{H}^s(F).$$

²¹Tvrzení lze zformulovat obecněji pro tzv. *hölderovská zobrazení*, tedy zobrazení f splňující

$$\varrho(f(x), f(y)) \leq K(\varrho(x, y))^\alpha,$$

Důkaz. Nechť $\mathcal{F} = \{F_1, F_2, \dots\}$ je δ -pokrytí F . Pak

$$\text{diam}(f(F \cap F_i)) \leq K \text{diam}(F \cap F_i) \leq K \text{diam } F_i,$$

což znamená, že $\mathcal{G} = \{f(F \cap F_1), f(F \cap F_2), \dots\}$ je $K\delta$ -pokrytí $f(F)$. Z toho plyne, že

$$\sum_{i=1}^{\infty} (\text{diam}(f(F \cap F_i)))^s \leq K^s \sum_{i=1}^{\infty} (\text{diam } F_i)^s$$

a tedy $\mathcal{H}_{K\delta}^s(f(F)) \leq K^s \mathcal{H}_{\delta}^s(F)$. Pro $\delta \rightarrow 0$ máme požadovaný výsledek. \square

(Převzato z [5, str. 46].)

Z toho speciálně plyne důsledek týkající se podobnosti.

Důsledek 2.5.11. *Nechť $F \subseteq \mathbb{R}^n$ v metrickém prostoru (\mathbb{R}^n, ϱ) a zobrazení $f : F \rightarrow \mathbb{R}^n$ je podobnost, tzn. existuje $K > 0$ takové, že pro každé $x, y \in F$ platí*

$$\varrho(f(x), f(y)) = K \varrho(x, y).$$

Pak pro každé $s \geq 0$ platí

$$\mathcal{H}^s(f(F)) = K^s \mathcal{H}^s(F).$$

Důkaz. K podobnosti f existuje inverzní zobrazení f^{-1} s koeficientem $L = 1/K$. Z věty 2.5.10 tedy plyne, že

$$\mathcal{H}^s(F) = \mathcal{H}^s(f^{-1}(f(F))) \leq \frac{1}{K^s} \mathcal{H}^s(f(F)),$$

nebo-li $\mathcal{H}^s(f(F)) \geq K^s \mathcal{H}^s(F)$. Opačnou nerovnost získáme aplikací věty 2.5.10 na zobrazení f . \square

2.5.3 Hausdorffova dimenze

Středobodem této sekce je tzv. *Hausdorffova dimenze*. Na úvod si dokážeme jedno jednoduché tvrzení týkající se Hausdorffovy míry.

Věta 2.5.12. *Nechť $0 \leq s < t < \infty$ a $F \subseteq X$. Pak platí:*

$$(i) \quad \mathcal{H}^s(F) < \infty \implies \mathcal{H}^t(F) = 0,$$

$$(ii) \quad \mathcal{H}^t(F) > 0 \implies \mathcal{H}^s(F) = \infty.$$

kde $\alpha \in (0, 1)$. Pak pro $F \subseteq \mathbb{R}^n$ platí

$$\mathcal{H}^{s/\alpha}(f(F)) \leq K^{s/\alpha} \mathcal{H}^s(F).$$

My si však vystačíme se speciálním případem.

Důkaz. Mějme δ -pokrytí $\mathcal{F} = \{F_1, F_2, \dots\}$ takové, že

$$\sum_{i=1}^{\infty} (\text{diam } F_i)^s \leq \mathcal{H}_{\delta}^s(F) + \varepsilon, \quad \varepsilon > 0.$$

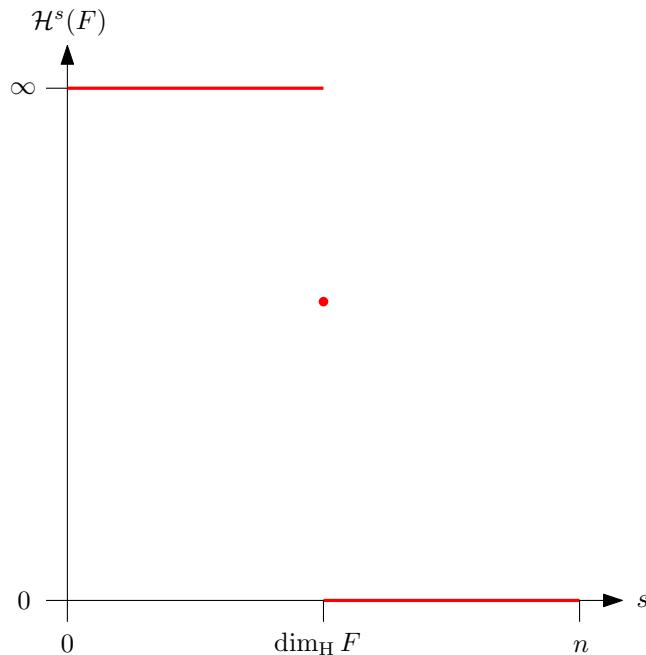
Pak

$$\mathcal{H}_{\delta}^t(F) \leq \sum_{i=1}^{\infty} (\text{diam } F_i)^t \leq \delta^{t-s} \sum_{i=1}^{\infty} (\text{diam } F_i)^s \leq \delta^{t-s} (\mathcal{H}_{\delta}^s(F) + \varepsilon).$$

Tzn. $\mathcal{H}_{\delta}^t(F) \leq \delta^{t-s} \mathcal{H}_{\delta}^s(F)$. Pro $\delta \rightarrow 0$ dostaneme body (i) a (ii). \square

(Převzato z [9, str. 68].)

Z věty 2.5.12 lze vidět, že Hausdorffova míra je zajímavá jen pro určitou hodnotu s . Pro „příliš velké“ s bude hodnota vždy 0, naopak pro „moc malé“ s bude jeho hodnota rovna ∞ (viz obrázek 2.8). Této kritické hodnotě s říkáme



Obrázek 2.8: Graf funkce $f(s) = \mathcal{H}^s(F)$, kde $F \subseteq \mathbb{R}^n$

Hausdorffova dimenze.

Definice 2.5.13 (Hausdorffova dimenze). Nechť $F \subseteq \mathbb{R}^n$. Hausdorffovou dimenzí²² množiny F nazveme hodnotu

$$\dim_H F = \inf \{s \geq 0 \mid \mathcal{H}^s(F) = 0\} = \sup \{s \geq 0 \mid \mathcal{H}^s(F) = \infty\}.$$

Hodnota $\mathcal{H}^s(F)$ se může pro různé množiny F splňující $s = \dim_H F$ lišit, tzn. může platit, že $\mathcal{H}^s(F) = \infty$, $\mathcal{H}^s(F) = 0$ a nebo se může jednat o konečné nenulové číslo, tj. $0 < \mathcal{H}^s(F) < \infty$.

Nyní se podívejme na příklad výpočtu. Podobně jako v případě box-counting dimenze, i zde budeme nezávisle určovat horní a dolní odhad.

²²Též někdy nazývaná *Hausdorffova-Bezikovičova dimenze*.

Příklad 2.5.14 (Sierpińského trojúhelník). V tomto případě se podíváme na dvě možnosti, jak dojít k výsledku. Celý Sierpińského trojúhelník si označme S .

- V k -té iteraci S_k , kde $k = 0, 1, 2, \dots$, vzniknou 3 nové trojúhelníky o obsahu $1/4$ obsahu původního trojúhelníka, tzn. jejich celkový počet je $t = 3^k$. Uvažíme-li δ -pokrytí

$$\mathcal{K} = \{K_\delta(x_1), K_\delta(x_2), \dots, K_\delta(x_t), \emptyset, \emptyset, \dots\},$$

kde $x_1, x_2, \dots, x_t \in S_k$ a $\delta \leq 2^{-k}/2 = 2^{-k-1}$, pak

$$\mathcal{H}_{2^{-k}}^s(S_k) \leq \sum_{i=1}^{3^k} (2^{-k})^s = 3^k 2^{-ks} = 1.$$

Pro $k \rightarrow \infty$ je $\mathcal{H}^s(S) \leq 1$. Poslední rovnost nastává právě pro $s = \ln 3 / \ln 2$.

Nyní ukážeme, že $\mathcal{H}^s(S) \geq 3^{-s} = 1/2$. Zvolme δ -pokrytí

$$\mathcal{F} = \{F_1, F_2, \dots\}$$

takové, že

$$2^{-k-1} \leq \text{diam } F_i < 2^{-k}, \quad (2.5)$$

kde $i \in \mathbb{N}$. Lze si rozmyslet, že každá z množin F_i má neprázdný průnik s nejvýše dvěma dílčími trojúhelníky S_k . Zvolíme-li $j \geq k$, pak každá z množin F_i má průnik maximálně s 3^{j-k} trojúhelníky v j -té iteraci, resp.

$$3^{j-k} = 3^j 2^{-ks} \leq 2^j 3^s (\text{diam } F_i)^s,$$

jak plyne z volby pokrytí \mathcal{F} v (2.5). Pokud navíc pro každé $i \in \mathbb{N}$ platí, že

$$3^{-j-1} \leq \text{diam } F_i,$$

pak každá z množin F_i má neprázdný průnik s nejvýše 3^j trojúhelníky. Tedy pro jejich počet platí

$$3^j \leq \sum_{i=1}^{\infty} 3^j 3^s (\text{diam } F_i)^s,$$

přičemž úpravou už získáme požadovanou nerovnost.

- Druhá varianta výpočtu je sice méně rigorózní, avšak podstatně jednodušší. Sierpińského trojúhelník sestává ze tří kopií sebe samotného, přičemž každá z nich je obrazem původního obrazce S v podobnosti s koeficientem $K = 1/2$. Označme si dané části S_1, S_2 a S_3 , tj. $S = S_1 \cup S_2 \cup S_3$. Tedy podle σ -aditivity Hausdorffovy míry a důsledku 2.5.11 můžeme psát

$$\begin{aligned} \mathcal{H}^s(S) &= \mathcal{H}^s(S_1) + \mathcal{H}^s(S_2) + \mathcal{H}^s(S_3) \\ &= \left(\frac{1}{2}\right)^s \mathcal{H}^s(S) + \left(\frac{1}{2}\right)^s \mathcal{H}^s(S) + \left(\frac{1}{2}\right)^s \mathcal{H}^s(S) \\ &= 3 \left(\frac{1}{2}\right)^s \mathcal{H}^s(S). \end{aligned}$$

Budeme-li předpokládat, že $\mathcal{H}^s(S) < \infty$ (jak jsme již viděli z předchozího výpočtu, jedná se o netriviální předpoklad) pro $s = \dim_H S$, pak z rovnosti $1 = 3(1/2)^s$ lze dopočítat, že $s = \ln 3 / \ln 2$.

(Převzato a upraveno z [5, str. 53].)

Myšlenku druhého výpočtu z příkladu 2.5.14 ještě rozvedeme v kapitole 4, konkrétně v sekci 4.2 věnované systémům iterovaných funkcí.

Jako poslední zkusme postavit Hausdorffovu dimenzi proti box-counting dimenzi představenou v sekci 2.4. Vztah mezi nimi je docela jednoduchý a udává jej následující věta. Pro připomenucí doporučujeme se opětovně podívat na definici 2.4.2.

Věta 2.5.15. *Nechť $F \subseteq \mathbb{R}^n$ je neprázdná omezená množina. Pak platí*

$$\dim_H F \leq \underline{\dim}_B F \leq \overline{\dim}_B F$$

Důkaz. Nechť platí $1 < \mathcal{H}^s(F) = \lim_{\delta \rightarrow 0} \mathcal{H}_\delta^s(F)$ pro $s \geq 0$. Zvolme libovolné $\delta > 0$. Pak z omezenosti F existuje nějaké konečné pokrytí množiny, přičemž platí

$$1 < \mathcal{H}_\delta^s(F) \leq N_\delta(F)\delta^s.$$

(Pravá strana plyne z definice Hausdorffovy míry.) Úpravou této nerovnosti dostaneme

$$\begin{aligned} 0 &< \ln N_\delta(F) + s \ln \delta \\ s &< \frac{\ln N_\delta(F)}{-\ln \delta} \\ s &\leq \liminf_{\delta \rightarrow 0} \frac{\ln N_\delta(F)}{-\ln \delta}. \end{aligned}$$

Druhá nerovnost, jak již víme, plyne přímo z definice box-counting dimenze (viz 2.4.2). Pro případ $0 < \mathcal{H}_\delta^s(F) < 1$ stačí aplikovat vhodnou podobnost (obecně bilipschitzovská zobrazení nemění box-counting dimenzi množiny, jak víme z věty 2.4.13) na F a využít tvrzení 2.5.11. \square

(Převzato z [5, str. 50].)

K závěru poznamenejme, že mnoho „rozumných“ útvarů F je Hausdorffova dimenze $\dim_H F$ rovna box-counting dimenzi $\dim_B F$, ale obecná rovnost zde neplatí. Např. pro množinu $X = \mathbb{Q} \cap (0, 1)$ je $\dim_H X = 0$, neboť X je spočetná. Nicméně v případě box-counting dimenze je $\dim_B X = 1$. Rozmyšlení již ponecháme na čtenáři.

Pro více informací si dovolíme odkázat na článek [4] věnovaný přímo této otázce.

Kapitola 3

Hausdorffův metrický prostor

V této kratší kapitole se podíváme na některé výsledky související s *Hausdorffovým metrickým prostorem*, s nímž se ještě dále setkáme. Pro připomenutí záležitostí ohledně metrických prostorů celkově doporučujeme čtenáři se podívat do sekce 2.1 v kapitole 2. Pro další podrobnosti si dovolíme odkázat čtenáře na knihu [2, str. 71].

3.1 Hausdorffova metrika

Definice 3.1.1 (Hausdorffova metrika). Hausdorffovou metrikou (vzdáleností) nazýváme takové zobrazení $\varrho_H : \mathcal{P}(X)^2 \rightarrow \langle 0, \infty \rangle$, kde pro každé $A, B \in \mathcal{P}(X)$ platí

$$\varrho_H(A, B) = \inf \{ \delta > 0 \mid A \subseteq (B)_\delta, B \subseteq (A)_\delta \}.$$

Zde si nyní uvědomme, že nad standardním matrickým prostorem (X, ϱ) zavádíme metriku ϱ_H pracující se všemi podmnožinami X (měříme tedy vzdálenosti množin v X , nikoliv samotných bodů v X), kde δ -okolí množin $A, B \in \mathcal{P}(X)$ závisí na volbě původní metriky ϱ . Podobně jako v případě Lebesgueovy míry (viz sekce 2.3, definice 2.3.5) se i zde nabízí stejná otázka: *Co nás opravňuje nazývat Hausdorffovu metriku metrikou?* Odpovědí je, že zatím nic, neboť tato definice sama o sobě neimplikuje, že ϱ_H je metrika. V definici povolujeme totiž případ, kdy $\varrho_H(A, B) = \infty$, a pro příklady takových množin netřeba chodit daleko. Např. v \mathbb{R} , když budeme počítat Hausdorffovu vzdálenost množin $\{0\}$ a $\langle 0, \infty \rangle$, zjistíme, že $\varrho_H(\{0\}, \langle 0, \infty \rangle) = \infty$. Podobně, budeme-li počítat $\varrho_H(\emptyset, \{0\})$, dojdeme ke stejnemu výsledku. Ovšem i v případě některých neprázdných omezených množin si lze všimnout nesrovnalostí. Např. pro množiny $(0, 1)$ a $(0, 1)$ je Hausdorffova vzdálenost nulová, přestože dané množiny nejsou stejné.

Pokud se však omezíme jen na některé množiny, bude ϱ_H skutečně metrikou ve smyslu definice 2.1.1. Proto se dále zaměříme pouze na takové podmnožiny X , které jsou *neprázdné* a *kompaktní*.

Definice 3.1.2 (Hyperprostor). Systém všech neprázdných kompaktních podmnožin množiny X nazýváme *hyperprostor* a značíme jej $\mathbb{H}(X)$.

Tvrzení 3.1.3. Nechť $A, B \in \mathbb{H}(X)$. Pak $A \cup B \in \mathbb{H}(X)$.

Důkaz. Neprázdnost $A \cup B$ je zjevná. Sjednocení kompaktních množin je opět kompaktní. Máme-li totiž pokrytí $\mathcal{G} \supseteq A \cup B$, pak z kompaktnosti množin A, B víme, pro každou z nich lze vybrat konečné podpokrytí

$$\mathcal{U} = \{U_{i_1}, U_{i_2}, \dots, U_{i_k}\} \subset \mathcal{G}, \text{ resp. } \mathcal{V} = \{V_{j_1}, V_{j_2}, \dots, V_{j_\ell}\} \subset \mathcal{G},$$

kde $k, \ell \in \mathbb{N}$. Tedy $\mathcal{U} \cup \mathcal{V}$ tvoří konečné pokrytí $A \cup B$, resp.

$$\mathcal{G} \supseteq \mathcal{U} \cup \mathcal{V} \supseteq A \cup B.$$

□

Tvrzení 3.1.3 lze opět rozšířit indukcí

Důsledek 3.1.4. Pro $A_1, A_2, \dots, A_n \in \mathbb{H}(X)$ platí $\bigcup_{i=1}^n A_i \in \mathbb{H}(X)$.

Pro operaci průniku nebo rozdílu toto tvrzení již neplatí. Např. intervaly $\langle 0, 1 \rangle$ a $\langle 2, 3 \rangle$ jsou uzavřené a omezené, tedy (podle Heineho-Borelovovy věty 2.1.19) jsou kompaktní. Avšak jejich průnik je prázdný. Podobně pro rozdíl stačí uvážit $A \subseteq B$, tzn. $A \setminus B = \emptyset$.

Věta 3.1.5. Nechť (X, ϱ) je metrický prostor.. Pak Hausdorffova metrika ϱ_H je metrikou na $\mathbb{H}(X)$.

Důkaz. Zjevně pro každé $A, B \in \mathbb{H}(X)$ z definice platí $\varrho_H(A, B) \geq 0$ a $\varrho_H(A, B) = \varrho_H(B, A)$. Z kompaktnosti množin A, B plyne (podle věty 2.1.20), že A, B jsou omezené množiny, tedy $\varrho_H(A, B) < \infty$.

Pro $A = B$ platí, že pro každé $\delta > 0$ je splněno $A \subseteq (B)_\delta$ a zároveň $B \subseteq (A)_\delta$, tzn. $\varrho_H(A, B) = 0$. Opačná implikace také platí. Předpokládejme, že množiny A, B splňují $\varrho_H(A, B) = 0$. Pokud $x \in A$, pak pro každé $\delta > 0$ platí, že $x \in (B)_\delta$, tzn. $\varrho(x, B) = 0$, a z uzavřenosti B plyne $x \in B$. Tedy $A \subseteq B$ a analogicky platí i $B \subseteq A$, tzn. $A = B$.

Jako poslední je třeba ukázat platnost trojúhelníkové nerovnosti. Mějme množiny $A, B, C \in \mathbb{H}(X)$ a $\varepsilon > 0$. Pro libovolné $x \in A$ existuje $y \in B$ takové, že

$$\varrho(x, y) < \varrho_H(A, B) + \varepsilon.$$

Podobně pro y existuje $z \in C$ takové, že

$$\varrho(y, z) < \varrho_H(B, C) + \varepsilon.$$

Z toho plyne

$$\varrho(x, z) \leq \varrho(x, y) + \varrho(y, z) < \varrho_H(A, B) + \varrho_H(B, C) + 2\varepsilon,$$

kde pravou stranu nerovnosti označme δ . To znamená, že $A \subseteq (C)_\delta$ a zároveň $C \subseteq (A)_\delta$. Tedy

$$\varrho_H(A, C) \leq \varrho_H(A, B) + \varrho_H(B, C) + 2\varepsilon.$$

□

(Převzato z [2, str. 72].)

Definice 3.1.6 (Hausdorffův metrický prostor). Metrický prostor $(\mathbb{H}(X), \varrho_H)$ nazýváme *Hausdorffův metrický prostor* na metrickém prostoru (X, ϱ) .

Nyní si opět připomeňte něco z terminologie metrických prostorů. Posloupnost $\{x_n\}_{n=1}^\infty$ v metrickém prostoru (X, ϱ) , se nazývá *cauchyovská*, pokud

$$\forall \varepsilon > 0 \exists n_0 \in \mathbb{N} \forall n, m > n_0 : \varrho(x_n, x_m) < \varepsilon.$$

Obecně neplatí, že každá cauchyovská posloupnost je konvergentní, avšak pokud v X všechny takové posloupnosti jsou konvergentní, pak (X, ϱ) nazýváme *úplný metrický prostor*. O Hausdorffově metrickém prostoru (HMP) lze v tomto ohledu dokázat následující tvrzení.

Věta 3.1.7 (Úplnost HMP). *Je-li (X, ϱ) úplný metrický prostor, pak $(\mathbb{H}(X), \varrho_H)$ je také úplný metrický prostor.*

Důkaz věty je trochu delší, nicméně lze jej nalézt opět např. v knize [2, str. 72].

3.2 Kompaktní množiny a konvergence

Věta 3.2.1. *Nechť A_1, A_2, \dots je nerostoucí posloupnost množin, kde $A_i \in \mathbb{H}(X)$ pro každé $i \in \mathbb{N}$. Pak posloupnost A_1, A_2, \dots konverguje k množině*

$$A = \bigcap_{i=1}^{\infty} A_i$$

v Hausdorffově metrice.

Důkaz. Nechť je dáno $\varepsilon > 0$. Zjevně pro každé i je $A \subseteq A_i$, tzn. $A \subseteq (A_i)_\varepsilon$.

Nyní najdeme i_0 tak, že $\varrho_H(A, A_{i_0}) < \varepsilon$ pro každé $i \geq i_0$. Systém

$$\mathcal{F} = \{(A_i)_\varepsilon\} \cup \{X \setminus A_i \mid i \in \mathbb{N}\}$$

tvoří otevřené pokrytí množiny A_1 . Podle předpokladu je však A_1 kompaktní, tedy z \mathcal{F} lze vybrat konečné podpokrytí. Specificky existuje $i_0 \in \mathbb{N}$ takové, že systém

$$\mathcal{G} = \{X \setminus A_{i_0}\} \cup \{(A)_\varepsilon\}$$

pokrytím A_1 a $\mathcal{G} \subseteq \mathcal{F}$ (viz obrázek 3.1). Z toho plyne, že $A_{i_0} \subseteq (A)_\varepsilon$, a tedy pro všechna $i \geq i_0$ je $\varrho_H(A, A_i) < \varepsilon$. \square

V souvislosti s konvergencí v prostoru $(\mathbb{H}(X), \varrho_H)$ nebude na škodu si připomeneout jedno známé související tvrzení z matematické analýzy.

Věta 3.2.2. *Následující tvrzení jsou ekvivalentní:*

- (i) (X, ϱ) je úplný metrický prostor.



Obrázek 3.1: Ilustrace věty 3.2.1

(ii) Je-li A_1, A_2, \dots neklesající posloupnost uzavřených množin v X takových, že $\text{diam } A_i \rightarrow 0$ pro každé $i \in \mathbb{N}$, pak existuje $x \in X$ splňující

$$\bigcap_{i=1}^{\infty} A_i = \{x\}.$$

Věta 3.2.2 je v podstatě rozšířením Cantorova principu vnořených intervalů v \mathbb{R} . Speciálně z toho vyplývá, že i v Hausdorffově metrickém prostoru platí podmínka (ii), neboť z věty 3.1.7 víme, že $(\mathbb{H}(X), \varrho_H)$ tvoří úplný metrický prostor.

Věta 3.2.3. Nechť $(X, \varrho_1), (Y, \varrho_2)$ jsou metrické prostory a zobrazení $\{f_i\}_{i=1}^{\infty}$, kde $f_i : X \rightarrow Y$ pro každé $i \in \mathbb{N}$ a $f_j \Rightarrow f$ pro $j \rightarrow \infty$ na X (viz podsekce 2.1.4). k f. Pak posloupnost $f_1(X), f_2(X), \dots$ konverguje k $f(X)$ na metrickém prostoru $(\mathbb{H}(Y), \varrho_H)$.

Důkaz. Mějme $\varepsilon > 0$. Najdeme $i_0 \in \mathbb{N}$ tak, že $\varrho_H(f(X), f_i(X)) < \varepsilon$ pro $i \geq i_0$. Funkce f_1, f_2, \dots konvergují k f stejnomořně, tzn.

$$\exists i_0 \in \mathbb{N} \quad \forall i \geq i_0 \quad \forall x \in X : \varrho_2(f(x), f_i(x)) < \varepsilon.$$

Tzn. pro každé $x \in X$ a $i \geq i_0$ platí

$$\varrho_2(f(x), f_i(X)) \leq \varrho_2(f(x), f_i(x)) < \varepsilon$$

a též

$$\varrho_2(f_i(x), f(X)) \leq \varrho_2(f_i(x), f_i(x)) = \varrho_2(f(x), f_i(x)) < \varepsilon.$$

Z toho vyplývá, že

$$f(X) \subseteq (f_i(X))_{\varepsilon} \text{ a zároveň } f_i(X) \subseteq (f(X))_{\varepsilon},$$

neboli $\varrho_H(f(X), f_i(X)) < \varepsilon$. □

(Převzato z [2, str. 74])

Poznámka 3.2.4. Předpoklad stejnoměrné konvergence ve větě 3.2.3 je důležitý. Mějme např. metrický prostor $X = \langle 0, 1 \rangle$ se standardní eukleidovskou metrikou a definujme posloupnost funkcí

$$f_n : \langle 0, 1 \rangle \rightarrow \mathbb{R}, \quad f_n(x) = x^n.$$

Limitou je funkce f , kde

$$f(x) = \begin{cases} 0 & x \in \langle 0, 1 \rangle, \\ 1 & x = 1. \end{cases}$$

Posloupnost konverguje k funkci f pouze bodově, nikoliv stejnoměrně. Pro celý interval je $f(X) = \{0, 1\}$, nicméně

$$\varrho_H(\langle 0, 1 \rangle, \{0, 1\}) = \frac{1}{2}.$$

Tedy neplatí, že $f_n(X) \rightrightarrows f(X)$ v $(\mathbb{H}(\mathbb{R}), \varrho_H)$.

Kapitola 4

Klasifikace fraktálů

S kapitolami 2 a 3 jsme si v jistém rozsahu ukázali formální podstatu fraktální geometrie. Nicméně pravdou zůstává, že z fraktálních útvarů jako takových jsme si toho zatím moc neukázali (některým z nich jsme se věnovali v kapitole 1 v sekci 1.2 o soběpodobnosti). Proto se nyní podíváme na některé další příklady a především jejich podrobnější klasifikaci dle způsobu jejich vzniku. Na tuto kapitolu též volně navazuje kapitola 5, kde si fraktály trochu rozebereme z programovacího hlediska.

V této kapitole se zaměříme na tři hlavní metody klasifikace a konstrukce fraktálů – konkrétně na *L-systémy*, *systémy iterovaných funkcí (IFS)* a tzv. *Time Escape algoritmy*. Tyto přístupy představují základní postupy, pomocí kterých lze fraktální útvary nejen systematicky vytvářet, ale také analyzovat jejich strukturu a chování.

4.1 L-systémy

Uvedeme tuto kapitolu s ohledem na předchozí obsah trochu netradičně a od matematiky se (alespoň zdánlivě) na chvíli odkloníme. Podíváme se na fraktály, s jejichž způsobem popisu přišel v minulém století, konkrétně roku 1968, maďarský biolog ARISTID LINDENMAYER¹, a který (možná pro někoho i překvapivě) má základy především v informatice. [19, str. 2]

K popisu specifického druhu fraktálů lze využít znalosti z *teorie formálních jazyků* a *teorie automatů*, na jejímž počátku stál (mj.) britský matematik a informatik ALAN TURING². Ten ve svém článku *On Computable Numbers, with an Application to the Entscheidungsproblem* z roku 1936 zavedl koncept abstraktního stroje, dnes známého jako *Turingův stroj*. Jedná se o jednoduché zařízení s výpočetními schopnostmi již tehdy porovnatelnými se současnými počítači. Jeho princip fungování přitom není nikterak složitý. Pomineme-li nyní matematickou definici Turingova stroje, lze říci, že sestává ze tří hlavních částí:

1. **Oboustranně nekonečná páska.** Páska je rozdělena na políčka, z nichž každé může obsahovat některý symbol z předem známé abecedy (formálně

¹1925–1989

²1912–1954

vzato množiny) znaků.

2. **Čtecí/zapisovací hlava.** Hlavu si můžeme představit jako čtecí/zapisovací „okénko“, které stojí právě nad jedním políčkem pásky. V každém kroku stroj:

- přečte symbol pod hlavou,
- podle svého stavu a právě přečteného symbolu rozhodne, jakou akci provést.

3. **Řídicí jednotka.** Řídí chování stroje pomocí konečné množiny stavů

$$Q = \{q_0, q_1, \dots, q_n\}.$$

Přechodová funkce δ pro (ne nutně všechny) dvojice (stav, znak) udává:

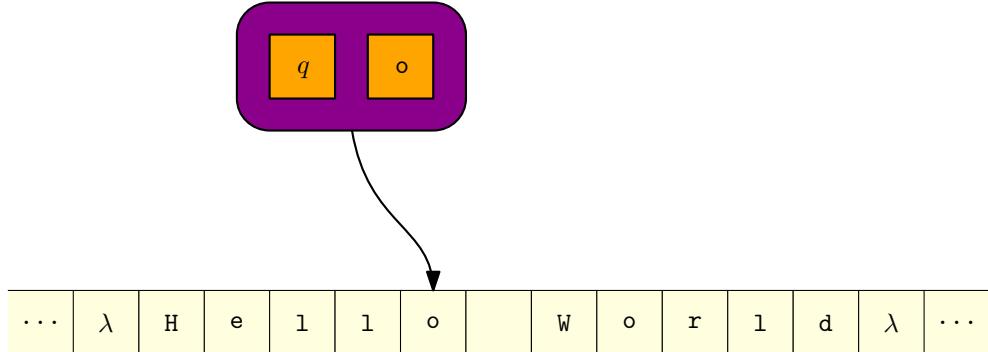
- jaký symbol má hlava na pásku zapsat (může přepsat stávající symbol nebo nechat stejný),
- do jakého stavu se stroj přepne,
- jak se posune hlava: doleva (L), doprava (R) nebo zůstane (N).

Výpočet pak probíhá takto:

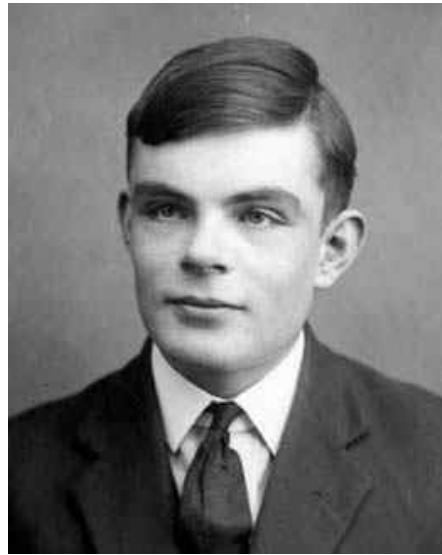
1. Stroj začíná ve stavu, který je stanovený jako počáteční. Na pásmu je vložen vstup (řetězec symbolů), zbytek pásky je prázdný.
2. Stroj vždy podle přechodové funkce stroje:
 - přečte, co je pod hlavou,
 - zapíše (případně přepíše) symbol,
 - přesune hlavu,
 - přejde do dalšího stavu.
3. Pokud stroj nemá definovaný přechod pro aktuální stav a čtený symbol na pásmu, zastaví se. Pokud stroj skončil ve stavu, který je označený jako přijímající, pak stroj slovo *přijal*, jinak jej *odmítá*.

Viz obrázek 4.1. V té době se Turing zabýval otázkou, kterou v roce 1928 položil známý německý matematik DAVID HILBERT (1862–1943). Je známá pod názvem „*Entscheidungsproblem*“³. Problémem bylo, zda existuje algoritmus, který o každém matematickém tvrzení je schopný rozhodnout (v konečném čase), zda je, či není pravdivé. Později Alan Turing tento problém přeformuloval takto: *Existuje program, který o jiném programu na vstupu rozhodne, zda se zastaví, či nikoliv?* David Hilbert byl ve svých vizích optimistický, avšak nakonec Alan Turing dokázal, že **takový algoritmus nemůže existovat**. Způsob, jakým Turing došel k onomu výsledku, byl v konečném důsledku vlastně až překvapivě jednoduchý

Čtecí/zapisovací hlava



Obrázek 4.1: Znázornění Turingova stroje



Obrázek 4.2: Alan Turing, 1912–1954 (Převzato z [14])

a existuje pro něj velké množství popularizačních materiálů⁴. Tím dal základ dnešní teoretické informatice. Turingův stroj, jakožto výpočetní model, stojí na samotném vrcholu hierarchie dalších výpočetních modelů⁵, které jsou však svojí výpočetní silou slabší. Později pak americký lingvista NOAM CHOMSKY⁶ popsal celkem 4 základní třídy tzv. *formálních jazyků*, které jsou dnes souhrnně známé pod názvem *Chomského hierarchie*. Za formální jazyk považujeme určitou množinu slov (retězců). Chomského hierarchie zařazuje každý jazyk do jedné ze tříd podle typu tzv. *formální gramatiky*, která slova z daného jazyka generuje. O gramatikách, a obecně určitém základu teorie formálních jazyků, si ještě budeme podrobněji povídат. O hlubší související teorii se čtenář může více dozvědět např. v knize [10].

³Anglicky *The Decision problem*, česky přeložitelné jako „rozhodovací problém“. Zde však poznamenejme, že onen český termín se používá i v související teorii složitosti a vyčíslitelnosti, má však podstatnějiný význam.

⁴Pokud by se chtěl čtenář dozvědět více o této problematice a teorii s ní související, doporučujeme např. knihu [10].

⁵Mezi ně patří jmenovitě tzv. *lineárně omezený automat*, *zásobníkový automat*, *deterministický* a *nedeterministický konečný automat*.

⁶1928 až do současnosti (2025)



Obrázek 4.3: David Hilbert, 1862–1943 (Převzato z [13])

4.1.1 Stručně k formálním jazykům a gramatikám

V úvodu této sekce jsme si již trochu přiblížili historické pozadí teorie formálních jazyků a automatů (tou se již dále zabývat nebudeme, nicméně bylo by přinejmenším neslušné ji zde alespoň nezmínit vzhledem k její přímé souvislosti s touto problematikou). Ačkoliv tento text si neklade za cíl seznámit čtenáře se všemi podrobnostmi, byly zde zmíněny určité termíny, jejichž význam bude dobré si objasnit, konkrétně

- *formální jazyk*
- a *formální gramatika*.

Definice 4.1.1. Množinu symbolů (znaků) Σ nazýváme *abeceda*.

- Libovolnou konečnou posloupnost znaků

$$w = a_1 a_2 \dots a_n,$$

kde $a_i \in \Sigma$ pro každé $1 \leq i \leq n$, nazýváme *slovo*, popř. *řetězec*.

- Prázdným slovem nazýváme slovo neobsahující žádné znaky, značíme ho λ .
- Délku slova w značíme $|w|$, tzn.

$$|w| = |a_1 a_2 \dots a_n| = n.$$

- Množinu všech slov délky n značíme Σ^n , tj.

$$\Sigma^n = \{a_1 a_2 \dots a_n \mid a_i \in \Sigma \text{ pro každé } i\}.$$

Speciálně $\Sigma^0 = \{\lambda\}$.

- Množinu všech slov v abecedě Σ značíme Σ^* , tzn.

$$\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n.$$

- Množinu všech neprázdných slov v abecedě Σ značíme Σ^+ , tzn.

$$\Sigma^+ = \Sigma^* \setminus \{\lambda\}.$$

- *Formálním jazykem* (nebo zkráceně jen *jazykem*) nazýváme libovolnou podmnožinu $L \subseteq \Sigma^*$.

Příklad 4.1.2. Některé příklady abeced:

- $\Sigma = \{0, 1\}$, tzv. *binární abeceda*.

$$\{0, 1\}^* = \{\lambda, 0, 1, 00, 01, 10, 11, 000, 001, \dots\}$$

- $\Sigma = \{a, b, c, \dots, z\}$, tj. všechna písmena anglické abecedy.

$$\{a, b, c, \dots, z\}^* = \{\lambda, a, b, c, \dots, z, aa, ab, ac, \dots\}$$

- Všechny znaky ASCII⁷ tvoří abecedu.

Definice 4.1.3 (Operace se slovy). Nechť Σ je libovolná abeceda a mějme slova $u, v \in \Sigma^*$, kde $u = u_1u_2\dots u_n$ a $v = v_1v_2\dots v_m$. Pak definujeme následující operace:

- *Zřetězení (konkantenace)* slov u a v je slovo

$$uv = u_1u_2\dots u_nv_1v_2\dots v_m.$$

Je celkem zjevné, že $|uv| = |u| + |v| = n + m$.

- Nechť $n \in \mathbb{N}$. Pak definujeme induktivně:

$$\begin{aligned} u^0 &= \lambda, \\ u^1 &= u, \\ u^n &= u^{n-1}u. \end{aligned}$$

Slovo u^n se nazývá *n-tá mocnina slova u*.

- *Obráceným slovem u* rozumíme slovo $u^R = u_nu_{n-1}\dots u_1$.

Poznámka 4.1.4. Speciálně pro prázdné slovo λ a libovolné slovo $u \in \Sigma^*$ platí $u\lambda = \lambda u = u$.

⁷Zkratka pro *American Standard Code for Information Interchange*. Stanovuje 128-bitové znakové kódování. Podotkněme, že ne všechny znaky jsou nutně tisknutelné a mají čistě informativní charakter, avšak to nás zde z formálního hladiska trápit vůbec nemusí.

Na základě definice 4.1.3 můžeme být nyní daleko konkrétnější při popisu některých slov a jazyků. Např.

$$L = \{0^n 1^n \mid n \in \mathbb{N}_0\}$$

značí jazyk všech slov obsahující znaky 0 a 1 ve tvaru

$$\lambda, 01, 0011, 000111, \dots$$

O jazycích celkově lze dokázat řadu zajímavých tvrzení, která se přímo opírají o již zmíněnou teorii automatů. V tomto ohledu si dovolíme však hodně záležitostí přeskočit. Jazyky lze popisovat několika různými způsoby, avšak nás bude zajímat popis pomocí formálních gramatik.

Definice 4.1.5 (Formální gramatika). *Formální gramatikou* (zkráceně jen *gramatikou*) nazýváme uspořádanou čtveřici $G = (V, T, P, S)$, kde

- $V \neq \emptyset$ je množina *neterminálů*⁸ (neterminálních symbolů),
- $T \neq \emptyset$ je množina *terminálů* (terminálních symbolů),
- $S \in V$ je *počáteční symbol*,
- a P je množina přepisovacích *pravidel*⁹ ve tvaru $\beta A \gamma \rightarrow \omega$ (čteme „řetězec $\beta A \gamma$ se přepíše na řetezec ω “), kde $A \in V$ a $\beta, \gamma, \omega \in (V \cup T)^*$. Tzn. levá strana každého pravidla obsahuje alespoň jeden neterminál.

Definice 4.1.6 (Odvození slova v gramatice). Mějme gramatiku $G = (V, T, P, S)$. Říkáme, že

- α se *přímo přepíše* na ω , píšeme $\alpha \Rightarrow_G \omega$ nebo jen $\alpha \Rightarrow \omega$, jestliže

$$\exists \beta, \gamma, \eta, \nu \in (V \cup T)^* : \alpha = \eta \beta \nu, \omega = \eta \gamma \nu \wedge (\beta \rightarrow \gamma) \in P.$$

- α se *přepíše* na ω , píšeme $\alpha \Rightarrow_G^* \omega$ nebo jen $\alpha \Rightarrow^* \omega$, pokud

$$\exists \beta_1, \beta_2, \dots, \beta_n \in (V \cup T)^* : \alpha = \beta_1 \Rightarrow \beta_2 \Rightarrow \dots \Rightarrow \beta_n = \omega.$$

Posloupnost β_1, \dots, β_n takovou, že $\beta_1 \Rightarrow \beta_2 \Rightarrow \dots \Rightarrow \beta_n$ nazýváme odvozením. Též říkáme, že gramatika G generuje slovo w , pokud $S \Rightarrow^* w$.

Terminální symboly gramatiky představují znaky koncové abecedy, nad níž uvažujeme výsledná slova. Neobsahuje-li tedy slovo na konci odvození žádný neterminál, jsme hotovi.

Formální gramatiky nám umožňují rekurzivní definici jazyka. Jinak řečeno popisuje, jak lze slova daného jazyka generovat. Pravidla z množiny P lze aplikovat v libovolném pořadí.

⁸Anglicky *variables*

⁹Formálně vzato se jedná o uspořádanou dvojici (α, ω) .

Poznámka 4.1.7. • Typicky platí, že pro neterminály používáme velká písma abecedy A, B, \dots, Z a pro terminály naopak malá písmena a, b, \dots, z . Pro slova (řetězce) budeme využívat buď malá písmena řecké abecedy, nebo malá písmena z konce abecedy, tj. \dots, w, y, x, z .

- Pokud máme více pravidel ve tvaru $\alpha \rightarrow \omega_i$, tj. se shodnou levou stranou, pak pro jejich zápis volíme tuto kompaktnější variantu:

$$\alpha \rightarrow \omega_1 \mid \omega_2 \mid \dots \mid \omega_n.$$

Příklad 4.1.8. Jazyk všech palindromů nad binární abecedou, tj.

$$L_{\text{pal}} = \{w \in \{0, 1\}^* \mid w = w^R\}$$

lze generovat gramatikou $G = (\{S\}, \{0, 1\}, P, S)$, přičemž P obsahuje následující pravidla:

$$S \rightarrow \lambda \mid 0 \mid 1 \mid 0S0 \mid 1S1.$$

Kupříkladu slovo $w = 011110$ lze odvodit takto:

$$S \Rightarrow 0S0 \Rightarrow 01S10 \Rightarrow 011S110 \Rightarrow 011\lambda110 = 011110.$$

Trochu praktičtěji zaměřený příklad nám poskytuje 4.1.9. Gramatiky se typicky využívají (vyjma fraktální geometrie) např. v komplátorech různých programovacích jazyků v rámci tzv. *syntaktické analýzy*, při níž se (jak název napovídá) kontroluje syntaktická správnost zápisu programu. Konkrétně se pro jednoduchost zaměříme pouze na matematické výrazy obsahující pouze operace $+$ a $*$ (násobení).

Příklad 4.1.9. Mějme gramatiku $G = (V, T, P, S)$, kde

$$V = \{E, I\}, T = \{a, b, (,), +, *\}, S = E$$

a pravidla jsou následující:

$$\begin{aligned} E &\rightarrow I \mid E + E \mid E * E \mid (E) \\ I &\rightarrow a \mid b \mid Ia \mid Ib. \end{aligned}$$

V takto definované gramatice lze vygenerovat např. slovo

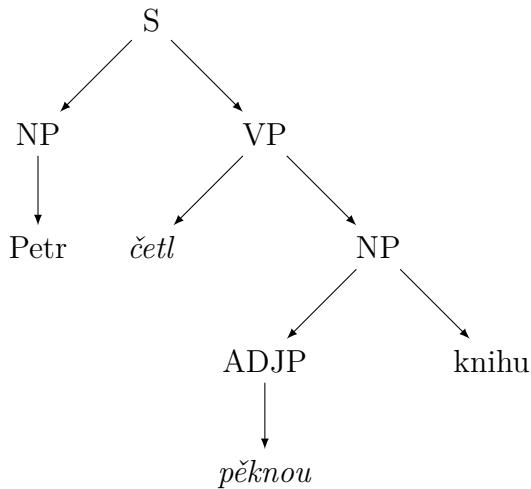
$$w = (a + b) + bb * (ab + bb)$$

následovně:

$$\begin{aligned} E &\Rightarrow E + E \Rightarrow (E) + E \Rightarrow (E) + E * E \Rightarrow (E) + E * (E) \\ &\Rightarrow (E + E) + E * (E) \Rightarrow (E + E) + E * (E + E) \\ &\Rightarrow^* (I + I) + I * (I + I) \Rightarrow^* (a + b) + Ib * (Ib + Ib) \\ &\Rightarrow^* (a + b) + bb * (ab + bb). \end{aligned}$$

S podobným přístupem se lze setkat i např. v lingvistice. Čtenář si možná ze základní školy vzpomíná na větný rozbor, v jehož rámci bylo úkolem určit pro zadanou větu její strukturu. Ve skutečnosti se nejedná o nic jiného, než využítí určité formální gramatiky $G = (V, T, P, S)$, jejíž pravidla v množině P jsou dány gramatickými pravidly (nyní v lingvistickém slova smyslu) českého jazyka. Pro příklad viz obrázek¹⁰ 4.4. Zároveň zde můžeme vidět souvislost formální gramatiky s gramatikou v lingvistice.

¹⁰S, NP, VP, ADJP představují neterminály příslušné gramatiky a jednotlivá slova (Petr, četl,



Obrázek 4.4: Příklad syntaktického stromu věty: „Petr četl pěknou knihu.“

4.1.2 Definice L-systému

V podsekci 4.1.1 jsme v podstatě viděli „rychloúvod“ do problematiky *formálních jazyků a gramatik*, kde jsme si především zavedli potřebnou terminologii a značení. L-systémy¹¹ se v kontextu formálních gramatik mírně, avšak podstatně, liší v tom, že pravidla jsou v jednom kroku aplikována paralelně.

Základní idea vychází z konceptu, který jsme si již představili v úvodní kapitole 1, a to *soběpodobnosti*. V každém dalším kroku jsme část útvaru nahradili jeho zmenšenou kopíí. Nešlo by toto nějak zachytit pomocí postupného přepisování, jako tomu bylo v případě formálních gramatik? Ovšem, že ano. Jako první se tedy podíváme na definici L-systému.

Definice 4.1.10 (L-systém). L-systémem nazveme uspořádanou trojici $G = (V, \omega, P)$, kde:

- V je abeceda,
- $\omega \in \Sigma^+$ je počáteční slovo zvané *axiom*,
- a P je množina pravidel tvaru $a \rightarrow \alpha$, kde $a \in V$ a $\alpha \in V^*$.
- Pro každé $a \in V$ platí $(a \rightarrow a) \in P$, pokud a není na levé straně žádného jiného pravidla.

Porovnejme na chvíli definici 4.1.10 s definicí formální gramatiky uvedenou výše (viz definice 4.1.5). U gramatik jsme rozlišovali tzv. neterminální a terminální symboly. U L-systému však tato dvojice druhů symbolů splývá, resp. nerozlišujeme mezi terminály a neterminály, neboť v tomto případě nebude účelem generovat slova nějakého konkrétního jazyka (tj. nemá ani smysl rozlišovat, které

pěknou, knihu) její terminály. Významy: S (*Sentence*), NP (*Noun Phrase*), VP (*Verb Phrase*), ADJP (*Adjective Phrase*).

¹¹Pojmenovány po Aristidu Lindenmayerovi.

symboly jsou součástí koncové abecedy). Dále začínáme rovnou s řetězcem symbolů $\omega \in V^+$. To lze v gramatice zařídit jednoduše tak, že zavedeme speciální neterminál T , který uvedeme jako počáteční, a pak přidáme pravidlo $T \rightarrow \omega$, přičemž ω , ani žádná jiná pravidla neobsahují neterminál T . V tomto ohledu tedy můžeme vidět, že L-systém je z pohledu definice gramatiky pouze zvláštní případ.

Jak už bylo uvedeno na začátku této podsekce, podstatný rozdíl oproti gramatickám nastává ve způsobu přepisování, resp. odvozování slov (pro připomenutí viz definice 4.1.6).

Definice 4.1.11 (Odvození slova v L-systému). Nechť je dán L-systém $G = (V, \omega, P)$.

- Slovo $\alpha = a_1 a_2 \dots a_n \in V^+$, kde $a_i \in V$, se *přímo přepíše* na $\omega = \omega_1 \omega_2 \dots \omega_n$, kde $\omega_i \in V^*$ pro každé $1 \leq i \leq n$, pokud

$$\forall i \in \mathbb{N}, 1 \leq i \leq n : (a_i \rightarrow \omega_i) \in P.$$

Píšeme $\alpha \Rightarrow_G \omega$ nebo jen $\alpha \Rightarrow \omega$.

- Slovo α se *přepíše* na ω , pokud

$$\exists \beta_1, \beta_2, \dots, \beta_n \in V^* : \alpha = \beta_1 \Rightarrow \beta_2 \Rightarrow \dots \Rightarrow \beta_n = \omega.$$

Píšeme $\alpha \Rightarrow_G^* \omega$ nebo jen $\alpha \Rightarrow^* \omega$.

Zatím jsme se moc nepozastavili nad poslední podmínkou v definici 4.1.10. Zajišťuje nám, že každý znak lze vždy přepsat na jiný řetězec. Zkuste si schválně rozmyslet, že právě díky této podmínce dává definice odvození slova 4.1.11 smysl.

Než se posuneme dál, zkusme se ještě věnovat různým variantám L-systémů. Definice 4.1.10, kterou jsme si uvedli, ve skutečnosti odráží pouze jeden ze speciálních případů. Tomuto typu se říká *bezkontextové L-systémy* (značené *OL-systémy*). Název vychází z faktu, že každý symbol přepisujeme zvlášť (připouštíme pouze pravidla ve tvaru $a \rightarrow \omega$, kde $a \in V$ a $\omega \in V^*$), tzn. nezáleží na ostatních znacích (tedy kontextu, v němž se nachází). Jinými typy L-systémů jsou naopak *kontextové*¹², u nichž pracujeme s pravidly ve tvaru $\gamma A \beta \rightarrow \gamma \omega \beta$, kde $\gamma, \beta, \omega \in V^*$ a $A \in V$.

Dále lze L-systémy rozlišovat podle toho, zda pro každý řetězec $\alpha \in V^*$ existuje právě jedno pravidlo $\alpha \rightarrow \omega$, nebo existuje více pravidel v P , které obsahují α na levé straně. Takový typům L-systémů se říká *deterministické*¹³, resp. *neterministické*. Pro deterministické bezkontextové L-systémy se pak užívá značení *DOL-systémy*, což je i případ, kterým se budeme dále zabývat. Pokud by čtenáře tato teorie zajímala více, doporučujeme knihu [19].

¹²Tyto typy L-systémů souvisí do jisté míry s již v úvodu zmíněnou Chomského hierarchií, v jejímž rámci jsou definovány podobně i bezkontextové a kontextové gramatiky.

¹³Tzn. je jednoznačně určeno, na jaký řetězec bude α přepsáno.

4.1.3 Želví grafika

Pro neznalého čtenáře se může zdát zarážející, že zde pracujeme celou dobu s řetězci znaků a nikoliv s geometrickými útvary. Zde si ukážeme, jak interpretovat výsledné řetězce pomocí tzv. *želví grafiky*.

Představme, že máme virtuální želvu v \mathbb{R}^2 , jejíž stav reprezentujeme pomocí trojice (x, y, α) , kde (x, y) jsou její pozice¹⁴ a α je orientovaný úhel udávající její směr. Dále máme pevně zadanou délku kroku $d > 0$ a přírůstek úhlu δ . Definujeme základní čtverečci symbolů, s níž budeme v rámci abecedy V pracovat. Symboly budou reprezentovat jednotlivé akce pro želvu. Jejich význam udává tabulka 4.1.

- F* Želva se přesune o vzdálenost d . Stav se změní na (x', y', α) , kde $x' = x + d \cos \alpha$ a $y' = y + d \sin \alpha$, a zároveň je nakreslena úsečka mezi body (x, y) a (x', y') .

f Želva se přesune o vzdálenost d bez kreslení úsečky.

+ Želva se otočí doleva o úhel δ . Nový stav bude $(x, y, \alpha + \delta)$.

- Želva se otočí doprava o úhel δ . Nový stav bude $(x, y, \alpha - \delta)$.

Tabulka 4.1: Význam symbolů v abecedě V pro želvu

Pracujeme tedy v základu s abecedou $V = \{F, f, +, -\}$ (později dodefinujeme ještě další symboly). Předpokládáme, že počáteční stav želvy je vždy $(0, 0, 0)$. Začneme s velice jednoduchou ukázkou, kde $\delta = \pi/6$ a krok d volíme libovolně. Pro začátek se podíváme na Kochovu křivku, kterou už jsme viděli v sekci 1.2.

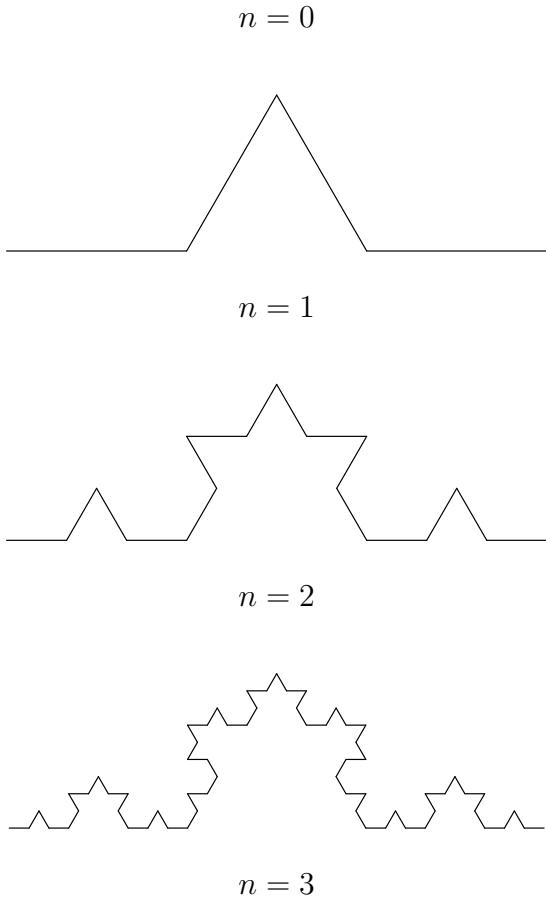
- Axiom $\omega = F$
 - Pravidla $P: F \rightarrow F - F - - - - F - F$

Iterováním tohoto L-systému postupně dostaneme slova:

Délka řetězce poroste exponenciálně¹⁵. Výslednou interpretaci daných řetězců můžeme vidět na obrázku 4.5. Všimněme si, že právě onen rekurzivní „charakter“ gra-

¹⁴Situaci lze pochopitelně zobecnit do \mathbb{R}^n . Ve vyšších dimenzích bychom museli zavést symboly pro rotaci okolo každé osy zvláště.

¹⁵Podobná situace nastává pro ostatní fraktály, které si zde ještě představíme. Generování vyšších iterací daných útváruj je proto často výpočetně náročné.



Obrázek 4.5: Interpretace vygenerovaných řetězců pro Kochovu křivku

matik, resp. L-systémů nám pomáhá dobře zachytit soběpodobnost (viz sekce 1.2) daných útvarů. Lindenmayer původně přišel s L-systémy kvůli modelování růstu a vývoje rostlin. Takový proces však není jednoduše popsatelný a hraje při něm roli jistá míra náhody. [19] I takové případy lze modelovat¹⁶. Pro zachování rozumného rozsahu tohoto textu se podíváme čistě na deterministické varianty. Na další typy fraktálů a jejich L-systémů jsou uvedeny v podsekci 4.1.4.

V případě modelování rostlin se nám hodí zavést ještě další dvojici symbolů. Mohly by nám stát, že se potřebujeme vrátit na určitou pozici, kde želva již stála. To by bylo možné vyřešit např. zpětným chodem želvy do původní pozice, nicméně velice špatně by se s tímto pracovalo v rámci L-systému. Proto si nejdříve vysvětlíme pojem tzv. *zásobníku*.

Koncept zásobníku je programátorem nejspíše velmi dobře známý, nicméně pro „neprogramátory“ si jej zde krátce vysvětlíme. Jedná se datovou strukturu pro dočasné ukládání dat, přičemž jeho charakteristikou je, že data, která jsou zařazena do zásobníku jako poslední jsou jako první odebrána. Pro tento způsob manipulace s daty se používá zkratka *LIFO*¹⁷ – *Last In, First Out*. V rámci zá-

¹⁶Konkrétně pomocí tzv. *stochastických L-systémů*, kde každému pravidlu přiřadíme pravděpodobnost aplikace v dalším kroku.

¹⁷Alternativním způsobem je *FIFO* – *First In, First Out*, který je využíván u tzv. *fronty*.

sobníku máme dvojici základních operací:

- *vložení prvku na vrchol zásobníku* (tzv. operace **push**),
- a *odebrání prvku z vrcholu zásobníku* (tzv. operace **pop**).

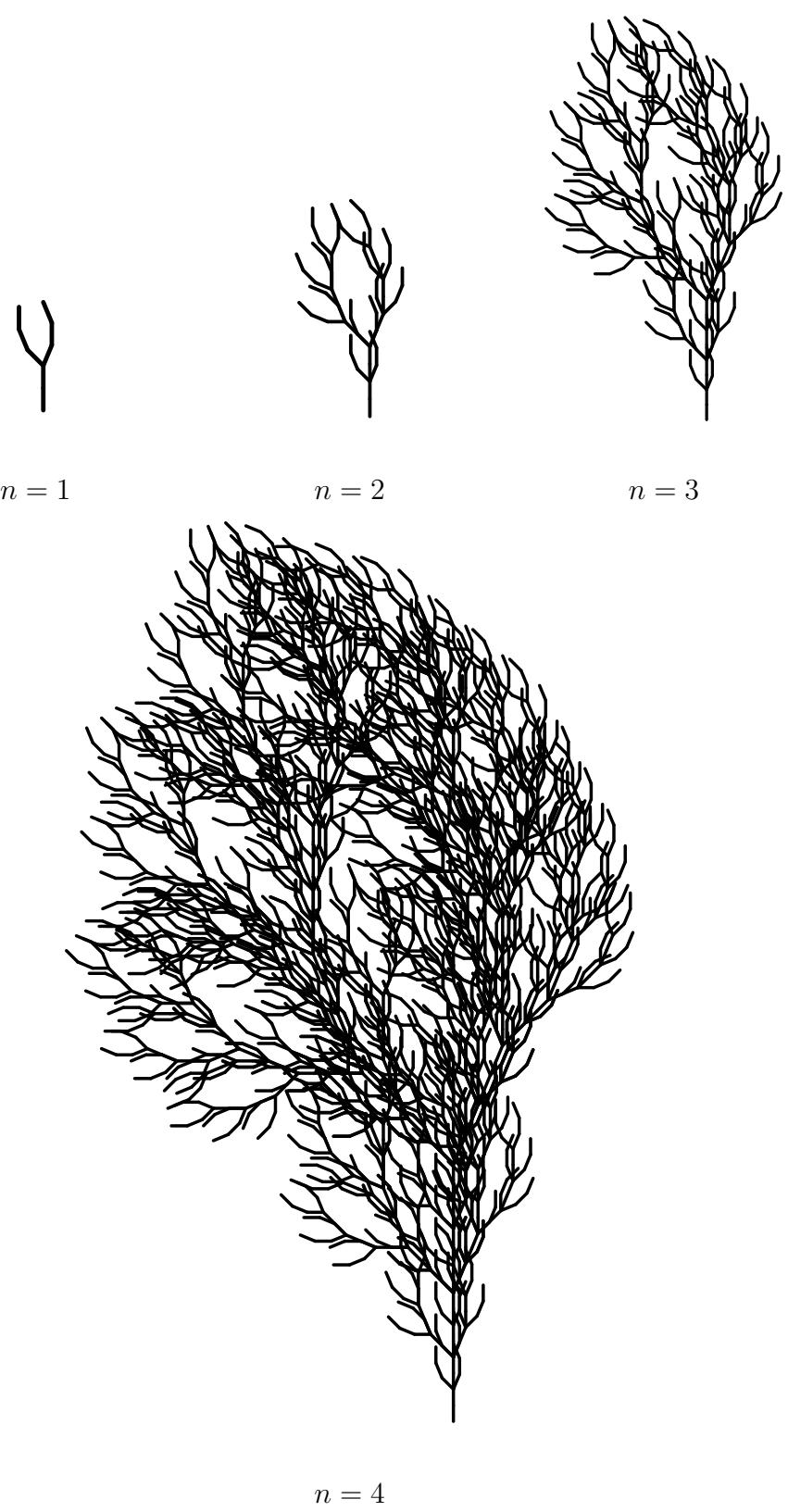
Pro znázornění zásobníku viz obrázek 4.7. Do abecedy V zavedeme nové znaky uvedené níže.

- [Uložení aktuálního stavu želvy (x, y, α) na zásobník.
-] Odebere stav uložený na vrcholu zásobníku a nastaví jej jako aktuální stav želvy.

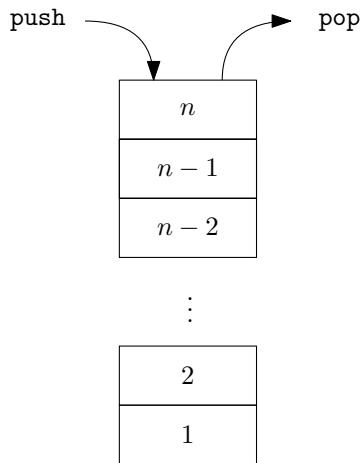
Tabulka 4.2: Význam symbolů v abecedě V pro želvu

Tato dvojice symbolů nám umožňuje využít želví grafiku pro vykreslení dalších fraktálních útvarů, neboť máme možnost si určitých místech ukládat stav želvy, k němuž se lze posléze vrátit. Uvažujme např. L-systém definovaný takto (viz obrázek 4.6):

- *Axiom:* $\omega = F$,
- *Pravidla P:* $F \rightarrow FF - [-F + F + F] + [+F - F - F]$,
- $\delta = \pi/8$.



Obrázek 4.6: Fraktální strom pro $n = 1, 2, 3, 4$

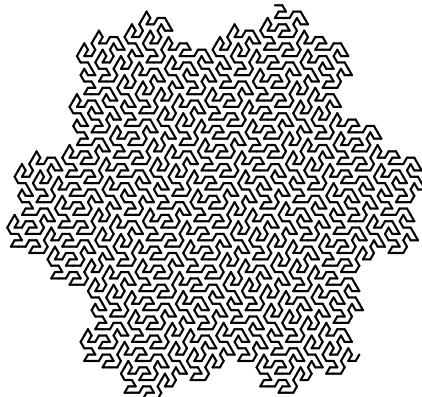


Obrázek 4.7: Znázornění zásobníku

4.1.4 Ukázky fraktálů a jejich L-systémů

Zde lze nalézt některé další ukázky fraktálů a jejich L-systémů. Všechny obrázky fraktálních útvarů, které jste zde viděli a ještě uvidíte jsou vygenerovány pomocí programu¹⁸ přiloženého k tomuto textu. Všechny tyto příklady jsou převatvy z knihy [19], kde lze též nalézt mnoho dalších.

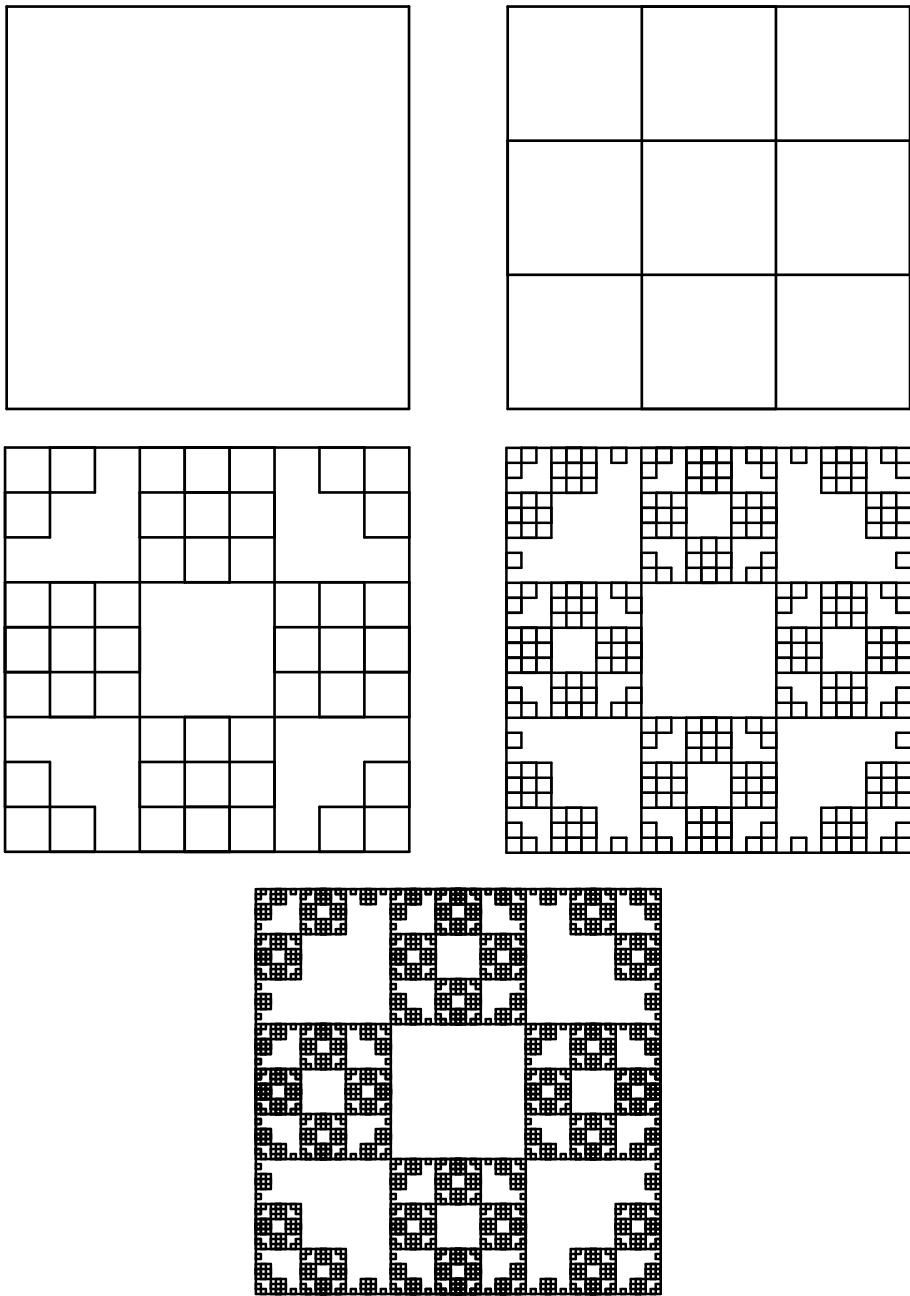
Zároveň zde ještě dodejme, že někdy se též můžeme setkat se situací, kdy potřebujeme více než jeden neterminál, tedy kromě F máme ještě jiné znaky. Obecně však platí, že jejich interpretace v želví grafice je stejná jako pro symbol F .



$$\begin{aligned}\omega &= L \\ L &\rightarrow L + R + +R - L - -LL - R + \\ R &\rightarrow -L + RR + +R + L - -L - R \\ \delta &= \pi/3 \\ n &= 4\end{aligned}$$

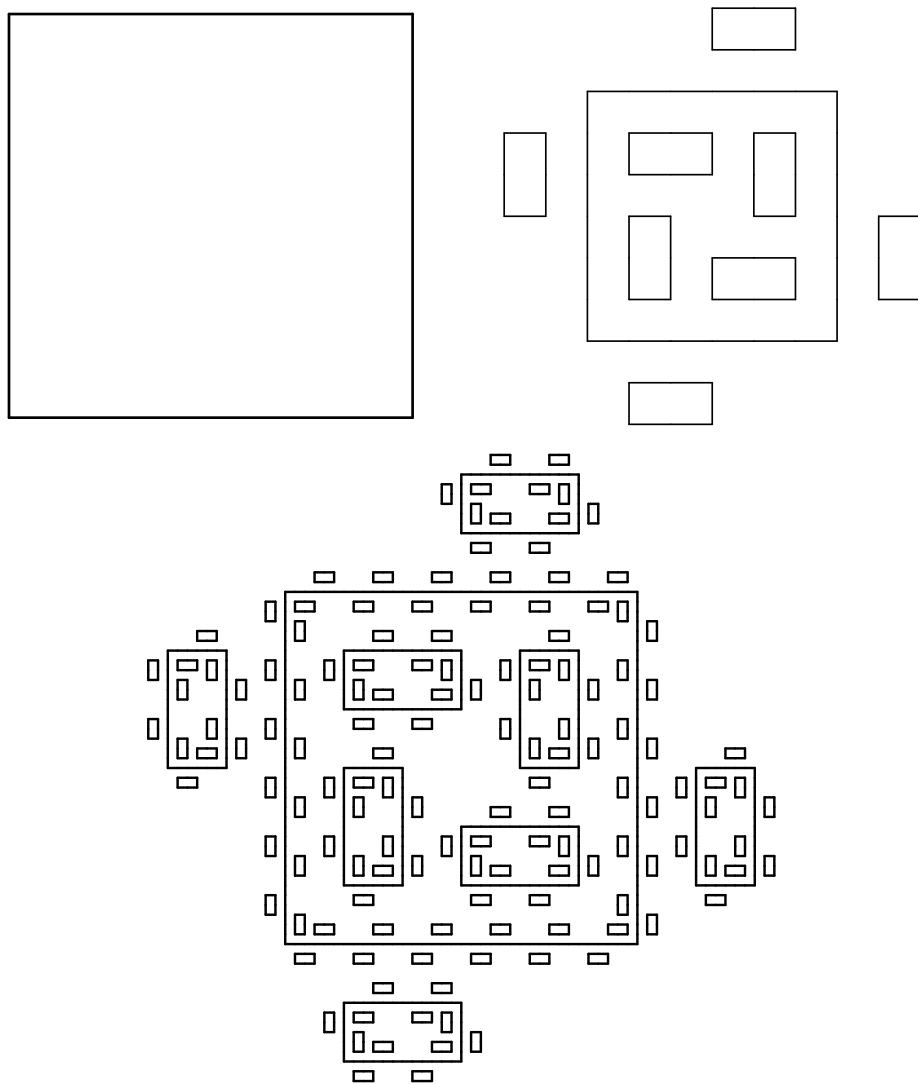
Obrázek 4.10: Prostor vyplňující křivka

¹⁸viz odkaz na GitHub repozitář: <https://github.com/D4vE0FF/Py-Fractal-Generator>



$$\begin{aligned}
 \omega &= F - F - F - F \\
 F &\rightarrow FF - F - F - F - FF \\
 \delta &= \pi/2
 \end{aligned}$$

Obrázek 4.8: Varianta Kochovy křivky



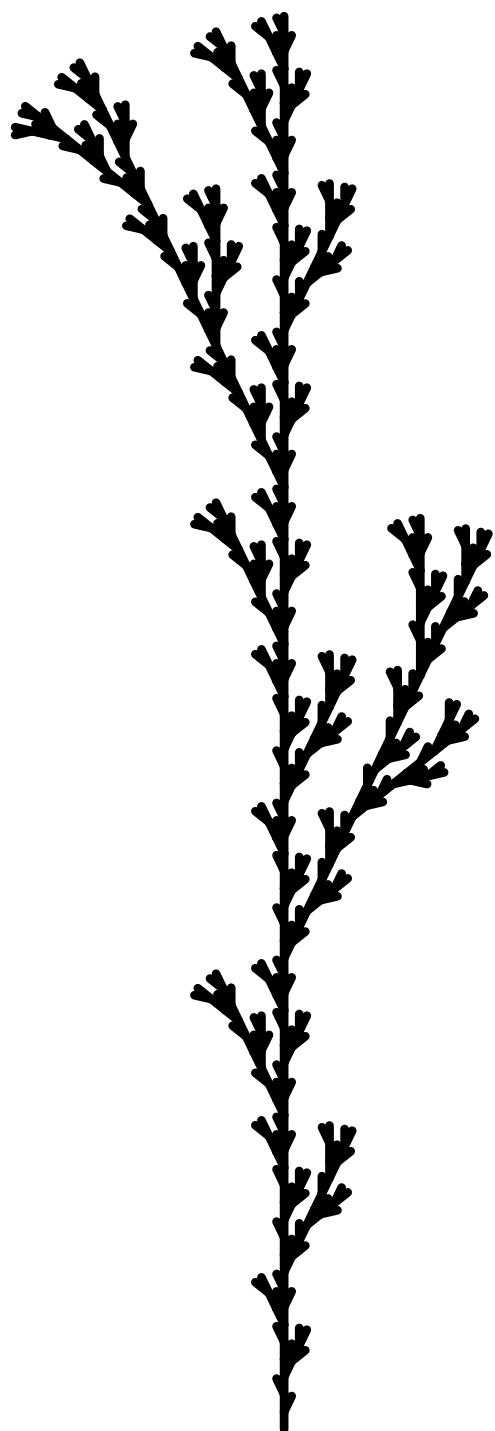
$$\omega = F + F + F + F$$

$$F \rightarrow F + f - FF + F + FF + Ff + FF - f + FF - F - FF - Ff - FFF$$

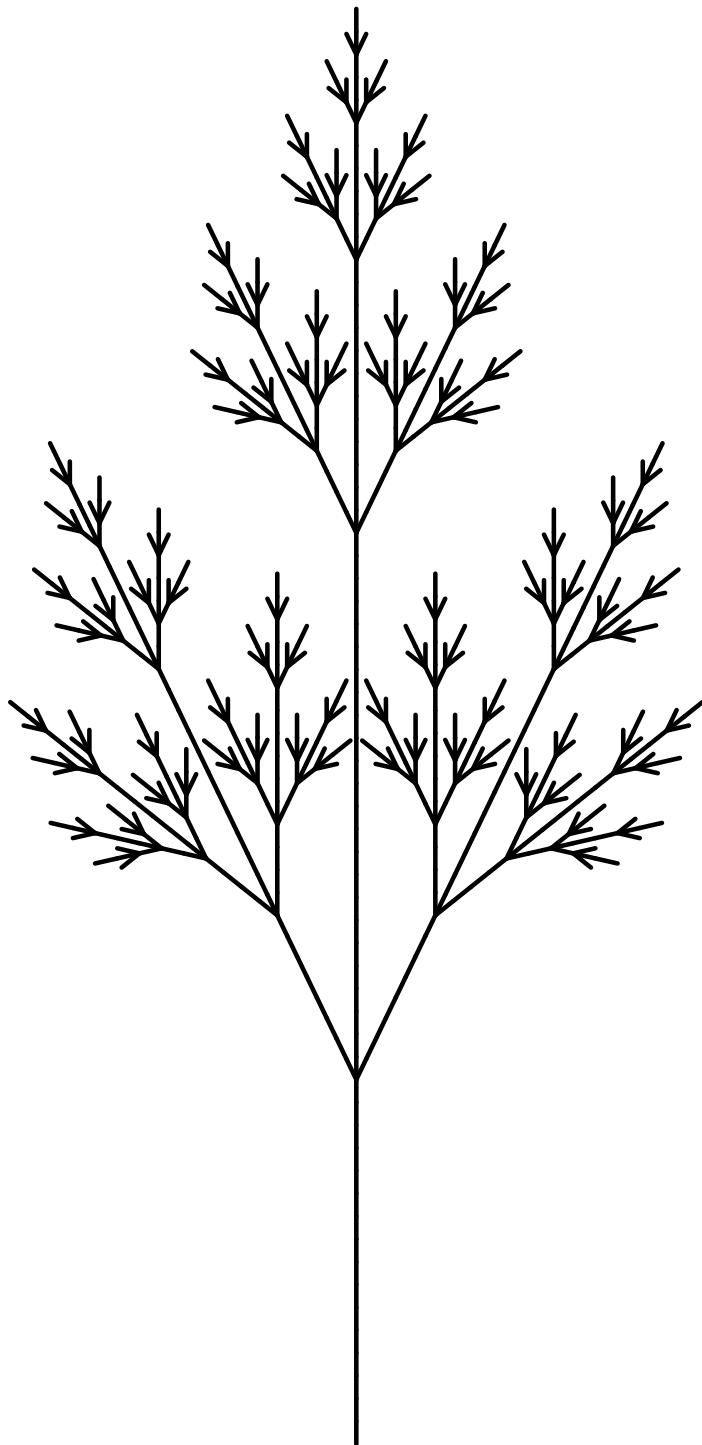
$$f \rightarrow ffffff$$

$$\delta = \pi/2$$

Obrázek 4.9: Jezera a ostrovy

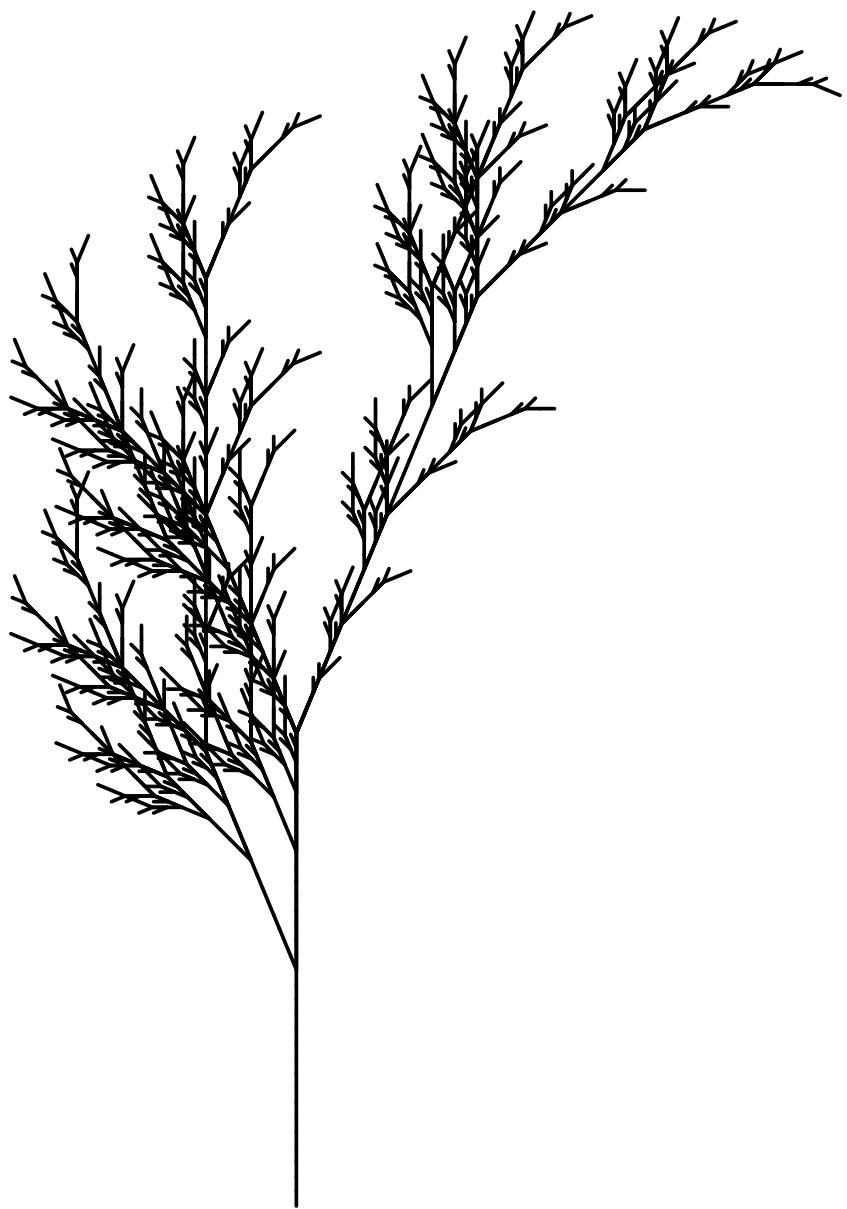

$$\begin{aligned}\omega &= F \\F &\rightarrow F[+F]F[-F]F \\ \delta &= 25,7^\circ \\ n &= 4\end{aligned}$$

Obrázek 4.11: Fraktální strom I



$$\begin{aligned}\omega &= X \\ X &\rightarrow F[+X][-X]FX \\ F &\rightarrow FF \\ \delta &= 25,7^\circ \\ n &= 4\end{aligned}$$

Obrázek 4.12: Fraktální strom II



$$\omega = X$$

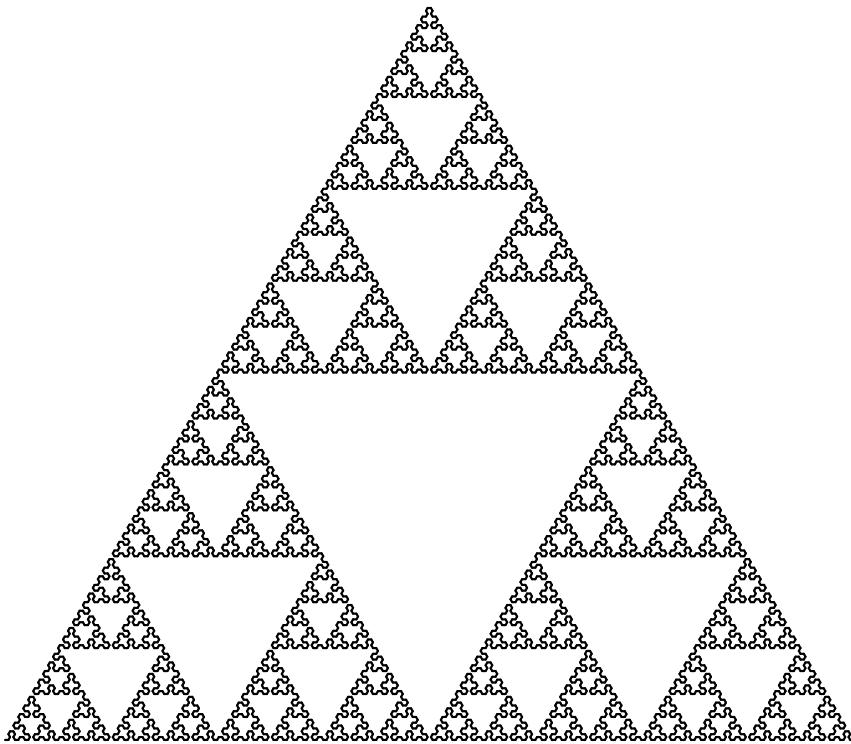
$$X \rightarrow F - [[X] + X] + F[+FX] - X$$

$$F \rightarrow FF$$

$$\delta = 22,5^\circ$$

$$n = 5$$

Obrázek 4.13: Fraktální strom III



$$\begin{aligned}
 \omega &= A \\
 A &\rightarrow B - A - B \\
 B &\rightarrow A + B + A \\
 \delta &= \pi/3 \\
 n &= 8
 \end{aligned}$$

Obrázek 4.14: L-systém pro Sierpiňského trojúhelník

4.2 Systém iterovaných funkcí

V předchozí části kapitoly jsme viděli, jak lze fraktální objekty efektivně popisovat pomocí L-systémů, u nichž struktura vzniká paralelním přepisováním symbolů a jejich vizualizace se provádí prostřednictvím želví grafiky. Tento přístup nám umožnil popsat určitou skupinu fraktálů, nicméně pro jiné fraktály by se nám hodil vhodnější popis jejich konstrukce. Např. Sierpiňského trojúhelník lze generovat pomocí L-systému, nicméně v sekci 1.2 o soběpodobnosti jsme si jej závaděli spíše pomocí opakování aplikace určitých geometrických transformací (ač jsme neuvedli jejich explicitní vyjádření). Ty si později zavedeme jako tzv. *systémy iterovaných funkcí*.

Nejdříve se podíváme trochu více na matematickou podstatu. Hodně záležitostí jsme si již rozebrali v kapitole 3 o Hausdorffově metrickém prostoru, který zde bude hrát významnou roli. Pro související matematickou teorii, kterou zde dále budeme vykládat, doporučujeme knihu [1].

4.2.1 Kontrakce na Hausdorffově metrickém prostoru

V minulých kapitolách jsme se často zaměřovali na lipschitzovská a bilipschitzovská zobrazení. Těmto termínům a faktům s nimi souvisejícími jsme se krátce věnovali v podsekci 2.1.6. V tomto případě nás budou speciálně zajímat tzv. *kontrakce*.

Připomeňme, že lipschitzovskost rovnou implikuje spojitost zobrazení v libovolném metrickém prostoru (X, ϱ) . Ještě než však začneme, podíváme se na alternativní definici Hausdorffovy metriky, která se nám bude hodit.

Věta 4.2.1 (Alternativní definice Hausdorffovy metriky). *Nechť (X, ϱ) je metrický prostor. Pro každé $A, B \in \mathbb{H}(X)$ platí*

$$\varrho_H(A, B) = \max \left\{ \sup_{x \in A} \varrho(x, B), \sup_{y \in B} \varrho(y, A) \right\}.$$

Důkaz. Nechť je dáno $\varepsilon > 0$ takové, že $\varepsilon \geq \varrho_H(A, B)$. Pak $A \subseteq (B)_\varepsilon$ a $B \subseteq (A)_\varepsilon$, tzn.

$$\varepsilon \geq \max \left\{ \sup_{x \in A} \varrho(x, B), \sup_{y \in B} \varrho(y, A) \right\}.$$

Naopak zvolíme-li $0 < \varepsilon \leq \varrho_H(A, B)$, pak určitě platí alespoň jedna z nerovností:

$$\varepsilon \leq \sup_{x \in A} \varrho(x, B) \quad \text{nebo} \quad \varepsilon \leq \sup_{y \in B} \varrho(y, A).$$

Tedy

$$\varepsilon \leq \max \left\{ \sup_{x \in A} \varrho(x, B), \sup_{y \in B} \varrho(y, A) \right\}.$$

Z toho dostáváme závěr tvrzení. □

Jako první se podíváme na trojici pomocných lemmat.

Lemma 4.2.2. *Nechť $f : X \rightarrow X$ je spojisté zobrazení v metrickém prostoru (X, ϱ) . Pak pro každé $S \in \mathbb{H}(X)$ platí $f(S) \in \mathbb{H}(X)$.*

Důkaz. Nechť $S \in \mathbb{H}(X)$. Zjevně platí $f(S) \neq \emptyset$. Pro důkaz kompaktnosti $f(S)$ uvažujme posloupnost $\{y_n\}_{n=1}^\infty$, kde $y_i \in f(S)$ pro každé $i \in \mathbb{N}$. Nyní definujeme posloupnost $\{x_n\}_{n=1}^\infty$ tak, aby $x_j \in S$ a navíc $f(x_j) = y_j$ pro každé $j \in \mathbb{N}$. Protože S je kompaktní, existuje posloupnost indexů $\{n_k\}_{k=1}^\infty$ taková, že $x_{n_k} \rightarrow x \in S$. Tzn., že $\{f(x_{n_k})\}_{k=1}^\infty = \{y_{n_k}\}_{k=1}^\infty$ je podposloupností posloupnosti $\{y_n\}_{n=1}^\infty$ a ze spojitosti f plyne

$$y_{n_k} = f(x_{n_k}) \rightarrow f(x) = y.$$

Tedy i $f(S)$ je kompaktní. □

V následujícím lemmatu uvažujeme zobrazení \hat{f} definované pro $A \in \mathbb{H}(X)$ předpisem $\hat{f}(A) = \{f(x) \mid x \in A\} = f(A)$. Protože nemůže dojít k nedorozumění, dohodneme se, že místo \hat{f} budeme psát opět f .

Lemma 4.2.3. Nechť $f : X \rightarrow X$ je kontrakce na metrickém prostoru (X, ϱ) s faktorem $0 < K < 1$. Pak $f : \mathbb{H}(X) \rightarrow \mathbb{H}(X)$ je kontrakce na Hausdorffově metrickém prostoru s faktorem K .

(Převzato z [1, str. 79].)

Důkaz. Z předchozího lemmatu 4.2.2 víme, že $f(S) \in \mathbb{H}(X)$ pro každé $S \in \mathbb{H}(X)$. Mějme množiny $A, B \in \mathbb{H}(X)$. Pak

$$\begin{aligned}\varrho(f(A), f(B)) &= \inf \{\varrho(f(x), f(y)) \mid x \in A, y \in B\} \\ &\leq \inf \{K \cdot \varrho(x, y) \mid x \in A, y \in B\} \\ &= K \cdot \inf \{\varrho(x, y) \mid x \in A, y \in B\} \\ &= K \cdot \varrho(A, B).\end{aligned}$$

Tedy celkově použitím věty 4.2.1

$$\begin{aligned}\varrho_H(f(A), f(B)) &= \inf \{\delta > 0 \mid f(A) \subseteq (f(B))_\delta \wedge f(B) \subseteq (f(A))_\delta\} \\ &= \max \left\{ \sup_{x \in A} \varrho(f(x), f(B)), \sup_{y \in B} \varrho(f(y), f(A)) \right\} \\ &= \max \left\{ \sup_{x \in A} \inf_{y \in B} \varrho(f(x), f(y)), \sup_{y \in B} \inf_{x \in A} \varrho(f(y), f(x)) \right\} \\ &\leq K \max \left\{ \sup_{x \in A} \inf_{y \in B} \varrho(x, y), \sup_{y \in B} \inf_{x \in A} \varrho(y, x) \right\} \\ &= K \varrho_H(A, B).\end{aligned}$$

□

(Převzato a upraveno z [1, str. 79].)

Lemma 4.2.4. Pro každé množiny $A, B, C, D \in \mathbb{H}(X)$, kde (X, ϱ) je metrický prostor, platí

$$\varrho_H(A \cup B, C \cup D) \leq \max \{\varrho_H(A, C), \varrho_H(B, D)\}.$$

Důkaz. Budeme vycházet z alternativní definice Hausdorffovy metriky, jejíž ekvivalence s původní definicí jsme dokázali výše (viz věta 4.2.1). Pro $\varrho(x, A \cup B)$ platí následující odhad:

$$\varrho(x, A \cup B) = \min \left\{ \inf_{y \in A} \varrho(x, y), \inf_{z \in B} \varrho(x, z) \right\} \leq \inf_{y \in A} \varrho(x, y) = \varrho(x, A).$$

Z toho pak máme

$$\sup_{x \in A} \varrho(x, C \cup D) \leq \sup_{x \in A} \inf_{y \in C} \varrho(x, y) = \varrho(x, A) \leq \varrho_H(A, C),$$

a tedy použitím analogického odhadu $\sup_{x \in B} \varrho(x, C \cup D) \leq \varrho_H(B, D)$ dostáváme

$$\sup_{x \in A \cup B} \varrho(x, C \cup D) \leq \max \{\varrho_H(A, C), \varrho_H(B, D)\}.$$

Stejný odhad lze získat i pro $\sup_{x \in C \cup D} \varrho(x, A \cup B)$:

$$\begin{aligned}\sup_{x \in C \cup D} \varrho(x, A \cup B) &= \max \left\{ \sup_{x \in C} \varrho(x, A \cup B), \sup_{x \in D} \varrho(x, A \cup B) \right\} \\ &\leq \max \{ \varrho_H(C, A), \varrho_H(D, B) \} = \max \{ \varrho_H(A, C), \varrho_H(B, D) \}.\end{aligned}$$

Tedy celkově lze psát

$$\begin{aligned}\varrho_H(A \cup B, C \cup D) &= \max \left\{ \sup_{x \in A \cup B} \varrho(x, C \cup D), \sup_{x \in C \cup D} \varrho(x, A \cup B) \right\} \\ &\leq \max \{ \varrho_H(A, C), \varrho_H(B, D) \}.\end{aligned}$$

□

Dvojice lemmat 4.2.2 a 4.2.3 nám v podstatě říká, že obrazem kompaktní množiny v kontraktivním zobrazení je opět kompaktní množina a že „kontraktivita“ zobrazení definovaného na libovolném metrickém prostoru (X, ϱ) se zachovává na hyperprostoru. Tento výsledek se nám bude později hodit, neboť jak již bylo zmíněno na začátku, některé fraktály lze konstruovat pomocí opakování aplikace určitých geometrických transformací. Jak lze nejspíše z dosavadního výkladu tušit, budeme pracovat právě s kontrakcemi.

Definice 4.2.5 (Systém iterovaných funkcí). *Systém iterovaných funkcí*, zkráceně IFS (z anglického *iterated function system*), na metrickém prostoru (X, ϱ) je konečná množina kontrakcí

$$\{\psi_i : X \rightarrow X \mid 1 \leq i \leq n\}$$

s faktory $K_i \in \langle 0, 1 \rangle$. Kontraktivním faktorem IFS je číslo

$$K = \max \{K_i \mid 1 \leq i \leq n\}.$$

Zatím není zcela zjevné, proč definujeme pro IFS kontraktivní faktor jako maximum z faktorů všech kontrakcí v něm obsažených (byť to může působit do jisté míry intuitivně). Odpověď na tuto otázku nám poskytne následující věta 4.2.6.

Věta 4.2.6. *Nechť $\{\psi_1, \psi_2, \dots, \psi_n\}$ je IFS na metrickém prostoru (X, ϱ) s kontraktivním faktorem $0 < K < 1$. Pak zobrazení $\Psi : \mathbb{H}(X) \rightarrow \mathbb{H}(X)$ pro $A \in \mathbb{H}(X)$ definované předpisem*

$$\Psi(A) = \bigcup_{i=1}^n \psi_i(A)$$

je kontrakce na $(\mathbb{H}(X), \varrho_H)$ s faktorem K .

(Převzato z [1, str. 81].)

Důkaz. Důkaz tvrzení lze provést indukcí podle n . V případě, kdy $n = 1$ je situace triviální. Pro $n = 2$ zvolme množiny $A, B \in \mathbb{H}(X)$. Pak

$$\begin{aligned}\varrho_H(\psi_1(A) \cup \psi_2(A), \psi_1(B) \cup \psi_2(B)) &\leq \max \{ \varrho_H(\psi_1(A), \psi_1(B)), \varrho_H(\psi_2(A), \psi_2(B)) \} \\ &\leq \max \{ K_1 \varrho_H(A, B), K_2 \varrho_H(A, B) \} \\ &\leq K \varrho_H(A, B),\end{aligned}$$

kde druhá nerovnost plyne z lemma 4.2.4. Předpokládejme nyní, že $n > 2$ a tvrzení platí pro $n - 1$. Ukážeme, že zobrazení Ψ definované předpisem $\Psi(A) = \bigcup_{i=1}^n \psi_i(A)$ je kontrakce. Mějme opět množiny $A, B \in \mathbb{H}(X)$. Pak

$$\begin{aligned}\varrho_H(\Psi(A), \Psi(B)) &= \varrho_H\left(\left(\bigcup_{i=1}^{n-1} \psi_i(A)\right) \cup \psi_n(A), \left(\bigcup_{i=1}^{n-1} \psi_i(B)\right) \cup \psi_n(B)\right) \\ &\leq \max \left\{ \varrho_H\left(\bigcup_{i=1}^{n-1} \psi_i(A), \bigcup_{i=1}^{n-1} \psi_i(B)\right), \varrho_H(\psi_n(A) \cup \psi_n(B)) \right\} \\ &\stackrel{\text{I.P.}}{\leq} \max \{K \varrho_H(A, B), K_n \varrho_H(A, B)\} = K \varrho_H(A, B).\end{aligned}$$

□

Značení 4.2.7. Dodejme, že dále v textu budeme používat značení $\Psi^{\circ n}$ definované induktivně:

$$\begin{aligned}\Psi^{\circ 0} &= \text{id}, \\ \Psi^{\circ n} &= \Psi \circ \Psi^{\circ(n-1)}, \quad n \in \mathbb{N}.\end{aligned}$$

Tzn. pro libovolnou množinu $B \in \mathbb{H}(X)$ je

$$\Psi^{\circ 0}(B) = B \quad \text{a} \quad \Psi^{\circ n}(B) = \Psi(\Psi^{\circ(n-1)}(B)).$$

S kontrakcemi se pojí známá věta z matematické analýzy, která se nazývá *Banachova věta o pevném bodě* (viz 4.2.9).

Definice 4.2.8 (Pevný bod). Bod $x \in X$ se nazývá *pevným bodem* zobrazení $f : X \rightarrow X$, pokud $f(x) = x$.

Věta 4.2.9 (Banachova věta o pevném bodě). *Nechť (X, ϱ) je úplný metrický prostor a zobrazení $f : X \rightarrow X$ je kontrakce. Pak existuje právě jeden pevný bod $x \in X$ zobrazení f . Navíc volíme-li $x_0 \in X$ libovolně a $x_n = f(x_{n-1})$ pro každé $n \in \mathbb{N}$, pak $x_n \rightarrow x$.*

Důkaz. Podle předpokladu je f kontrakce s faktorem K . Zvolme $x_0 \in X$ a dále pro každé $n \in \mathbb{N}$ položme $x_n = f(x_{n-1})$. Ukážeme, že takto definovaná posloupnost bodů má limitu. Podle předpokladu je (X, ϱ) úplný metrický prostor, tedy stačí ukázat, že posloupnost $\{x_n\}_{n=1}^\infty$ je cauchyovská. Pro vzdálenosti dvou po sobě jdoucích členů platí

$$\begin{aligned}\varrho(x_1, x_2) &= \varrho(f(x_0), f(x_1)) \leq K \varrho(x_0, x_1), \\ \varrho(x_2, x_3) &= \varrho(f(x_1), f(x_2)) \leq K \varrho(x_1, x_2) \leq K^2 \varrho(x_0, x_1), \\ \varrho(x_3, x_4) &= \varrho(f(x_2), f(x_3)) \leq K \varrho(x_2, x_3) \leq K^3 \varrho(x_0, x_1), \\ &\vdots \\ \varrho(x_i, x_{i+1}) &= \varrho(f(x_{i-1}), f(x_i)) \leq K \varrho(x_{i-1}, x_i) \leq K^i \varrho(x_0, x_1).\end{aligned}$$

Pro odhadnutí vzdálenosti dvojice členů, které nejdou nutně bezprostředně po sobě použijeme trojúhelníkovou nerovnost. Volme $n, m \in \mathbb{N}$, přičemž $m > n$. Pak lze psát

$$\begin{aligned}\varrho(x_n, x_m) &\leq \sum_{i=n}^{m-1} \varrho(x_i, x_{i+1}) \leq \sum_{i=n}^{m-1} K^i \varrho(x_0, x_1) = K^n \varrho(x_0, x_1) \sum_{i=1}^{m-n-1} K^i \\ &= K^n \varrho(x_0, x_1) \cdot \frac{1 - K^{n-m}}{1 - K}.\end{aligned}$$

Výraz na pravé straně má pro $n \rightarrow \infty$ limitu 0, tedy posloupnost $\{x_n\}_{n=1}^\infty$ je cauchyovská a má limitu $x \in X$. Dále ze spojitosti funkce f (neboť je lipschitzovská) plyne

$$x = \lim_{n \rightarrow \infty} x_{n+1} = \lim_{n \rightarrow \infty} f(x_n) = f(x),$$

tedy $x \in X$ je pevným bodem f . Jednoznačnost pevného bodu x lze ukázat sporem. Pokud by existoval další pevný bod $y \neq x$, pak

$$\varrho(x, y) = \varrho(f(x), f(y)) \leq K \varrho(x, y).$$

Protože $\varrho(x, y) > 0$, musí být $K \geq 1$, což je spor s předpokladem, že f je kontrukce. \square

Speciálně pro zobrazení Ψ definované ve větě 4.2.6 nám Banachova věta 4.2.9 nejen říká, že má právě jeden pevný bod $A \in \mathbb{H}(X)$, tzn. $\Psi(A) = A$, na úplném metrickém prostoru, ale zároveň udává způsob, jak daný pevný bod nalézt. Stačí opakováně iterovat dané zobrazení.

Definice 4.2.10 (Atraktor). Pevný bod $A \in \mathbb{H}(X)$ zobrazení Ψ definovaného ve větě 4.2.6 pro libovolné IFS se nazývá *atraktor* tohoto IFS.

Z Banachovy věty speciálně plyne, že atraktor A libovolného IFS lze určit jako následující limitu:

$$A = \lim_{n \rightarrow \infty} \Psi^{on}(B),$$

kde $B \in \mathbb{H}(X)$. Zajímavostí je fakt, že výsledný atraktor A je zcela nezávislý na volbě množiny B . Tento fakt si ještě přiblížíme v podsekci 4.2.2.

V dalším textu budeme pracovat především s afinními zobrazeními v \mathbb{R}^n se standardní eukleidovskou metrikou. Připomeňme si zde, že affiním zobrazením rozumíme jakékoli zobrazení $f : X \rightarrow X$ takové, že

$$f(x) = \mathbf{A}x + b,$$

kde $b \in X$ a \mathbf{A} je matice. V našem případě vždy o affiní bijekci, čili \mathbf{A} bude vždy regulární.

4.2.2 Fraktály generované pomocí IFS

V předešlé podsekci 4.2.1 jsme si osvojili některé důležité poznatky týkající se kontrukcí a Hausdorffova metrického prostoru (viz kapitola 3). Na některá dokázaná tvrzení se zde budeme odkazovat.

Protože pracujeme s fraktálními útvary v rovině, tj. \mathbb{R}^2 , affinní zobrazení budou mít tvar

$$\psi \begin{pmatrix} x \\ y \end{pmatrix} = \underbrace{\begin{pmatrix} a & b \\ c & d \end{pmatrix}}_{\mathbf{A}} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}. \quad (4.1)$$

Připomeňme, že matice \mathbf{A} je regulární. Podíváme se na některé fraktály, které jsme již viděli v úvodní kapitole 1 v sekci 1.2. Jiný způsob generování těchto a mnohých jiných fraktálních útvarů nám poskytuje L-systémy, které jsme si ilustrovali v sekci 4.1.

Znovu se podívejme na jeden z nejznámějších fraktálů z této kategorie – Sierpińského trojúhelník. Jeho konstrukce lze docílit pomocí IFS $\{\omega_1, \omega_2, \omega_3\}$, přičemž $\omega_1, \omega_2, \omega_3 : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ a jednotlivé kontrakce jsou definovány následujícími předpisy:

$$\begin{aligned} \omega_1 \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \\ \omega_2 \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1/2 \\ 0 \end{pmatrix}, \\ \omega_3 \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1/4 \\ \sqrt{3}/4 \end{pmatrix}. \end{aligned}$$

Tedy zobrazení $\Omega : \mathbb{H}(\mathbb{R}^2) \rightarrow \mathbb{H}(\mathbb{R}^2)$ definujeme jako

$$\Omega(A) = \omega_1(A) \cup \omega_2(A) \cup \omega_3(A),$$

kde $A \in \mathbb{H}(\mathbb{R}^2)$. Připomeňme, že takto definované Ω je (podle věty 4.2.6) kontrakce a tedy z Banachovy věty o pevném bodě plyne (viz věta 4.2.9), že má právě jeden atraktor, tj. množinu $A \in \mathbb{H}(\mathbb{R}^2)$ takovou, že platí $A = \Omega(A)$. Kontraktivní faktor tohoto (ani jiného) IFS nás vyloženě zajímá nemusí, avšak není těžké jej dopočítat. Např. zde je celkem zjevné, že $K = 1/2$. Užitečnější je však pro nás fakt, že atraktor A lze určit jako

$$A = \lim_{n \rightarrow \infty} \Omega^{\circ n}(B),$$

kde $B \in \mathbb{H}(\mathbb{R}^2)$. V našem případě je počátečním útvarem rovnostranný trojúhelník T , jehož vrcholy mají souřadnice

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1/2 \\ \sqrt{3}/2 \end{pmatrix}.$$

Prvních několik iterací zobrazení Ω si lze prohlédnout na obrázku 4.15. Sierpińského trojúhelník je tedy atraktorem zobrazení Ω .

Zajímavostí je, že atraktor daného IFS je zcela nezávislý na volbě počátečního útvaru. Ač je tedy běžné, že v případě Sierpińského trojúhelníka začínáme s rovnostranným trojúhelníkem, je ve skutečnosti zcela lhostejně¹⁹, s jakým útvarem začneme (viz obrázek 4.16).

¹⁹Z formálního hlediska je třeba, aby se jednalo o kompaktní neprázdnou množinu, nicméně v \mathbb{R}^n stačí pro tento účel uvažovat všechny neprázdné množiny, které jsou *uzavřené a omezené* (viz věta 2.1.19)

Vypisovat explicitně dané kontrakce je sice formálně žádoucí, nicméně dosti nepraktické. Dále tedy budeme hodnoty jednotlivých koeficientů a, b, \dots, f (viz (4.1)) zapisovat do tabulky. U Wacława Sierpińského se ještě chvíli zdržíme, neboť

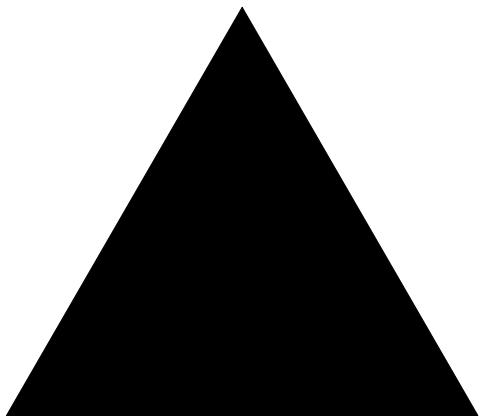
Kontrakce	a	b	c	d	e	f
ω_1	1/2	0	0	1/2	0	0
ω_2	1/2	0	0	1/2	1	0
ω_3	1/2	0	0	1/2	1/4	$\sqrt{3}/2$

Tabulka 4.3: Koeficienty IFS Ω pro Sierpińského trojúhelník

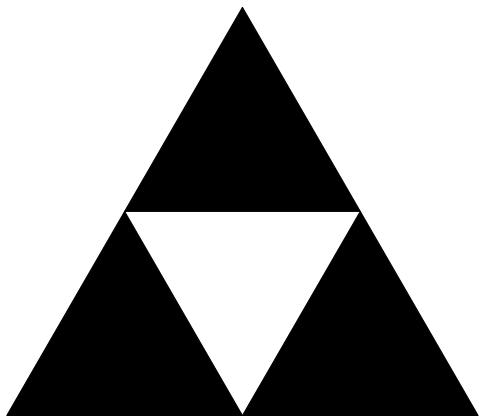
princip konstrukce jeho pravděpodobně nejznámenšího fraktálu lze přenést i na další útvary. Jiným takovým příkladem je tzv. *Sierpińského koberec* jehož IFS si označíme Φ (viz tabulka 4.4). I v tomto případě (podobně jako u Sierpińského trojúhelníka) lze začít s libovolným útvarem. Pro ukázku viz obrázek 4.17.

Kontrakce	a	b	c	d	e	f
φ_1	1/3	0	0	1/3	0	0
φ_2	1/3	0	0	1/3	0	1/3
φ_3	1/3	0	0	1/3	0	2/3
φ_4	1/3	0	0	1/3	1/3	2/3
φ_5	1/3	0	0	1/3	1/3	0
φ_6	1/3	0	0	1/3	2/3	0
φ_7	1/3	0	0	1/3	2/3	0
φ_8	1/3	0	0	1/3	2/3	0

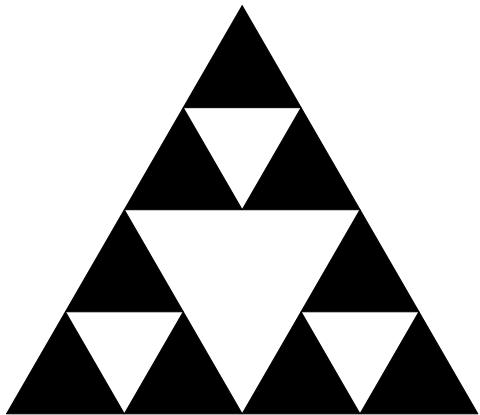
Tabulka 4.4: Koeficienty IFS Φ pro Sierpińského koberec



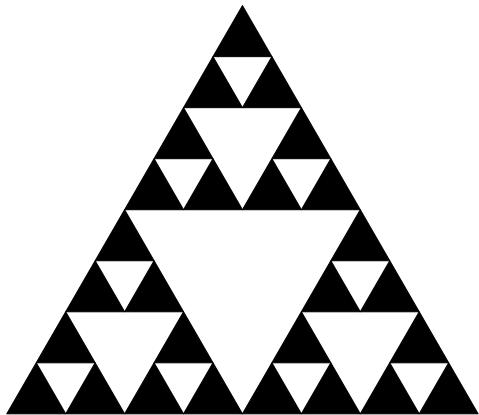
$n = 0$



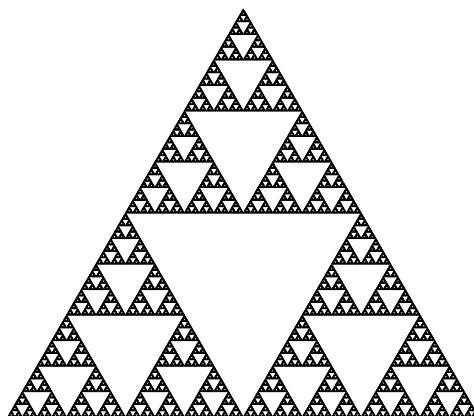
$n = 1$



$n = 2$

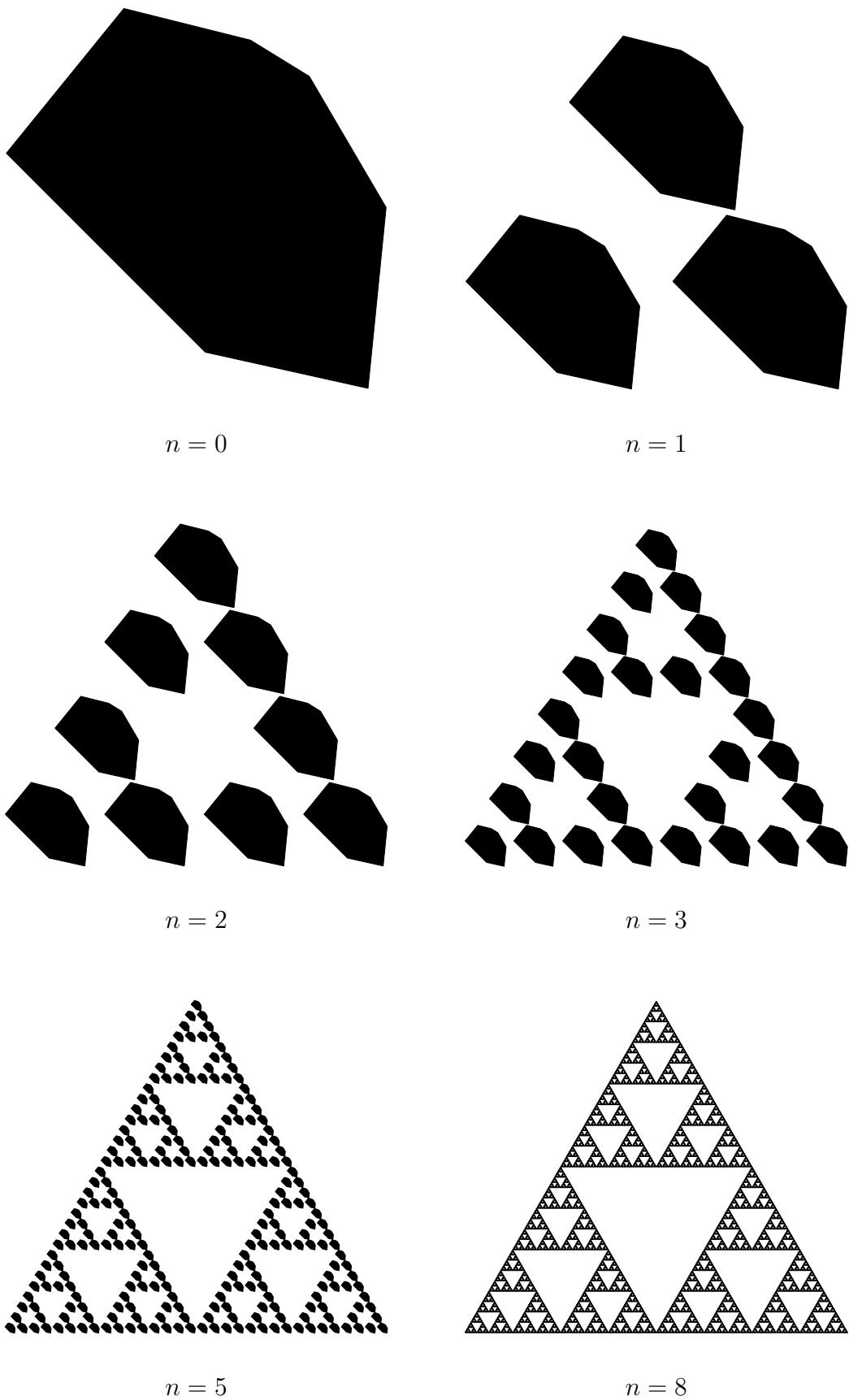


$n = 3$

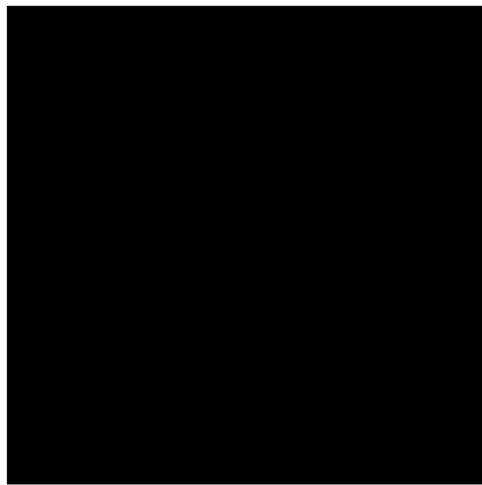


$n = 8$

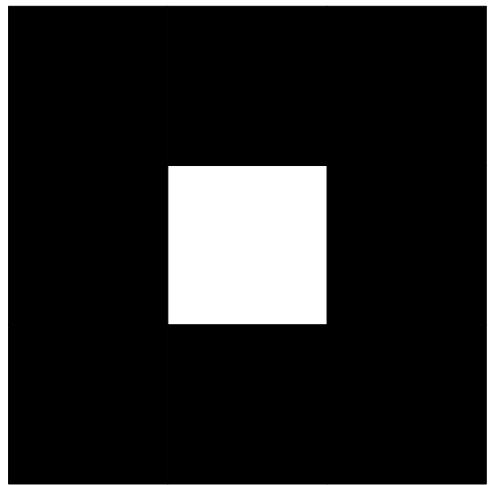
Obrázek 4.15: Iterace zobrazení Ω (Sierpińského trojúhelník)



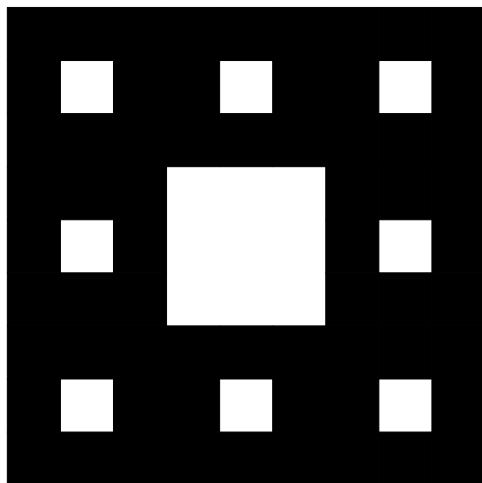
Obrázek 4.16: Iterace zobrazení Ω s jiným počátečním útvarem B



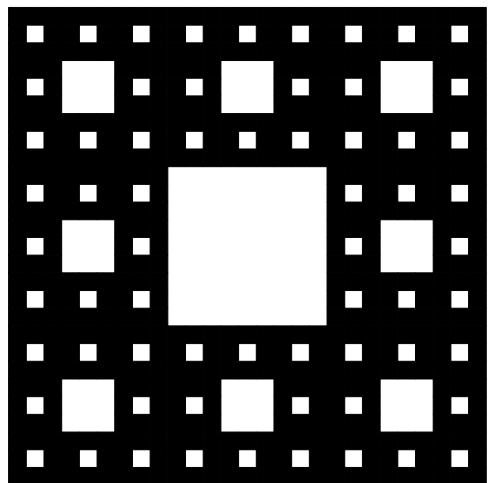
$n = 0$



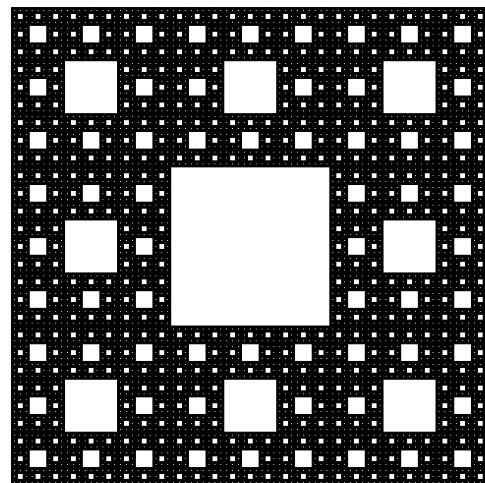
$n = 1$



$n = 2$



$n = 3$

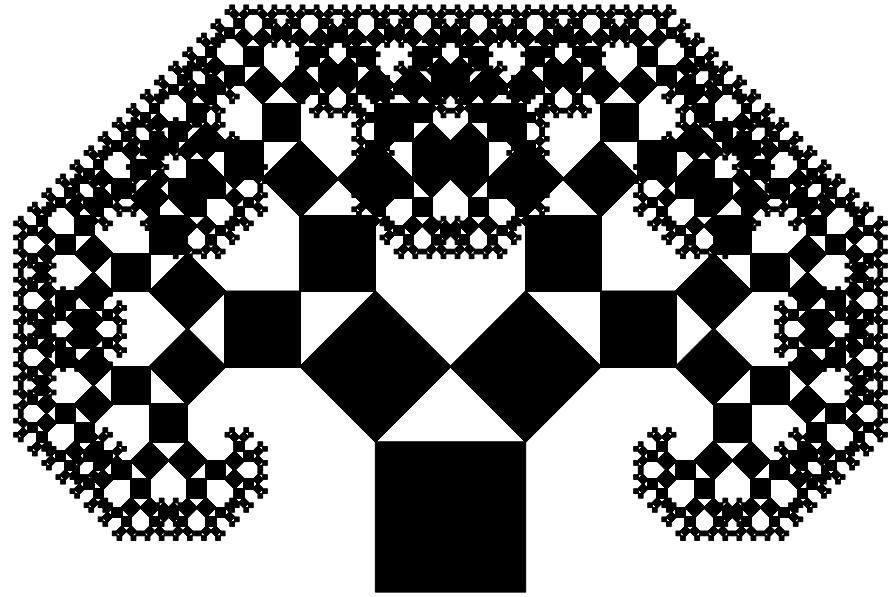


$n = 5$

Obrázek 4.17: Iterace zobrazení Φ (Sierpiňského koberec)

4.2.3 Další fraktály a jejich IFS

V této podsekci se podíváme na některé další příklady fraktálů a jejich IFS. I zde jsou všechny obrázky vygenerovány pomocí přiloženého programu²⁰.



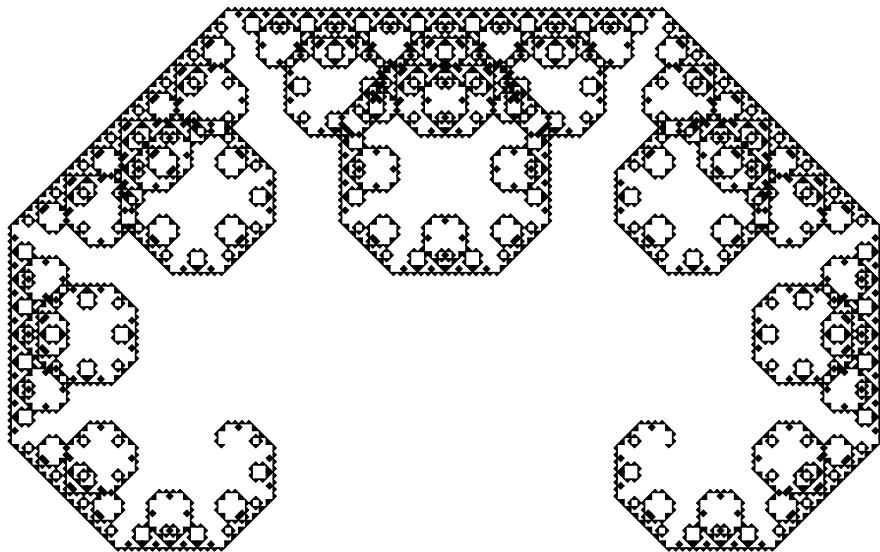
$$n = 10$$

Zobrazení	a	b	c	d	e	f
ψ_1	1/2	-1/2	1/2	1/2	0	1
ψ_2	1/2	1/2	-1/2	1/2	1/2	3/2
ψ_3	1	0	0	1	0	0

Obrázek 4.18: Pythagorův strom

Zde upozorněme však na fakt, že u Pythagorova stromu na obrázku 4.18 zobrazení ψ_3 je identita, která není kontrakcí, čili na takový systém nelze aplikovat větu 4.2.6. Toto zobrazení je zde však čistě z estetických důvodů, aby původní obrazec nezměnil svoji pozici v další iteraci. Systém $\{\psi_1, \psi_2\}$ již představuje IFS s jednoznačným atraktorem.

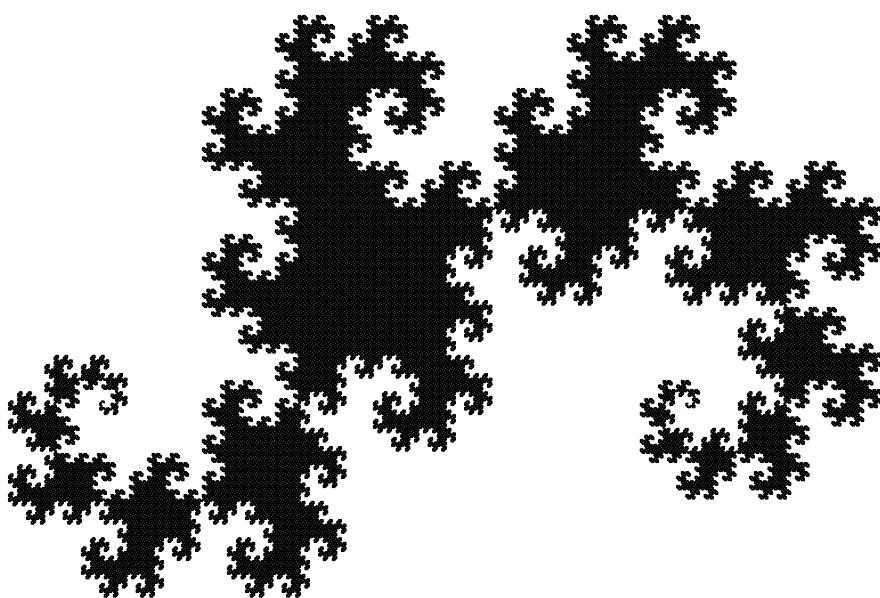
²⁰viz odkaz na GitHub repozitář: <https://github.com/D4vE0FF/Py-Fractal-Generator>



$n = 12$

Kontrakce	a	b	c	d	e	f
ψ_1	$1/2$	$-1/2$	$1/2$	$1/2$	0	0
ψ_2	$1/2$	$1/2$	$-1/2$	$1/2$	$1/2$	$1/2$

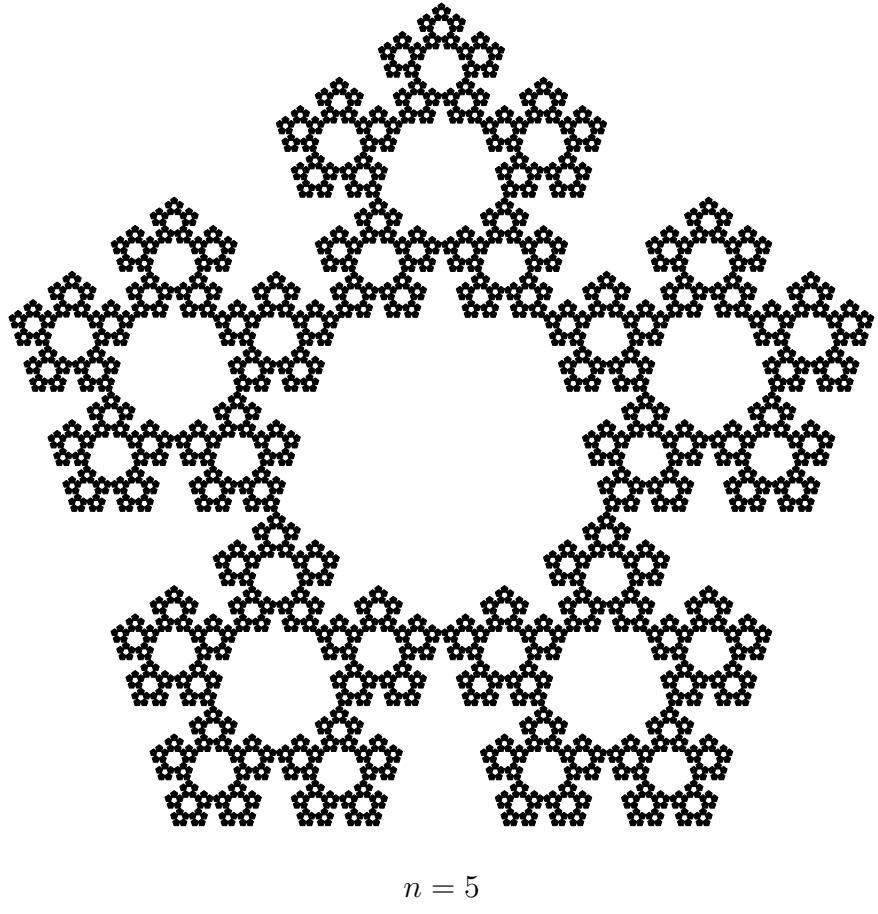
Obrázek 4.19: Lévyho drak



$n = 15$

Kontrakce	a	b	c	d	e	f
ψ_1	$1/2$	$-1/2$	$1/2$	$1/2$	0	0
ψ_2	$-1/2$	$-1/2$	$1/2$	$-1/2$	1	0

Obrázek 4.20: Dračí křivka



Kontrakce	a	b	c	d	e	f
ψ_1	0,382	0	0	0,382	0	0
ψ_2	0,382	0	0	0,382	0,618	0
ψ_3	0,382	0	0	0,382	0,809	0,588
ψ_4	0,382	0	0	0,382	0,309	0,951
ψ_5	0,382	0	0	0,382	-0,191	0,588

Obrázek 4.21: Sierpińského pětiúhelník

4.2.4 IFS a výpočet dimenze

Pojďme se ještě vrátit k tématu, které jsme již probírali v úvodní kapitole 1 a dále jsme se mu hlouběji věnovali v kapitole 2 – dimenzi. V rámci tohoto textu jsme si ukázali dva základní typy, a to *box-counting dimenzi* a *Hausdorffovu dimenzi*. V této části navážeme na některé výsledky, k nimž jsme dospěli v podsekcích 2.5.2 a 2.5.3.

Ohlédněme se zpět za příkladem 2.5.14, kde jsme počítali Hausdorffovu dimenzi Sierpińského trojúhelníka celkově dvěma způsoby. První vycházel přímo z její definice, zatímco druhý byl podstatně jednodušší, neboť pracoval s netriviálním předpokladem, že míra daného útvaru je konečná. Z onoho příkladu však nejspíše tušíme, že počítat Hausdorffovu dimenzi z definice je dosti nepraktické. Proto si zkusíme druhý způsob výpočtu trochu přiblížit.

Než se však pustíme do dalšího výkladu, připomeňme si některé nám již známé

výsledky. V sekci 2.4.2 jsme dokázali větu 2.4.13, která říká, že box-counting dimenze je invariantní vůči lipschitzovskému a bilipschitzovskému zobrazení. Tedy speciálně zobrazení $\Psi : \mathbb{H}(X) \rightarrow \mathbb{H}(X)$ z věty 4.2.6 definované předpisem

$$\Psi(B) = \bigcup_{i=1}^n \psi_i(B),$$

kde $B \in \mathbb{H}(X)$ a $\{\psi_1, \psi_2, \dots, \psi_n\}$ je IFS, je lipschitzovské. Horní box-counting dimenze kterékoliv iterace $\Psi^{on}(B)$ je tedy shora omezená horní box-counting dimenzí počátečního útvaru B , což plyne z věty 2.4.10 bodu (iii) (resp. důsledku 2.4.11). Budeme-li tedy počítat např. box-counting dimenzi Cantorova diskontinua (viz příklady 2.4.5 a 2.4.9), lze ihned odhadnout, že jeho dimenze bude menší než 1. Konkrétně jsme došli k výsledku

$$\frac{\ln 2}{\ln 3} \approx 0,630929 \dots < 1.$$

Dále na konci sekce 2.5 (konkrétně v podsekci 2.5.3) jsme dokázali fakt (viz věta 2.5.15), že pro neprázdné omezené množiny je Hausdorffova dimenze vždy shora omezena dolní box-counting dimenzí. Pro mnoho útvarů jsme sice jejich box-counting dimenzi, resp. Hausdorffovu dimenzi explicitně nepočítali, ale i tak nám to dává alepoň určitou představu o výsledku.

Tyto dosavadní výsledky jsou jistě hezké, ale přesto nám poskytují pouze odhady daných dimenzí. Nicméně v případě IFS, kterým je tato sekce věnována, si zformulujeme poměrně silné tvrzení, které nám v tomto ohledu podstatně zjednoduší práci. Nejdříve si ale zavedeme jiný termín, který budeme potřebovat.

Definice 4.2.11 (Open set condition). Nechť je dán IFS $\{\psi_1, \psi_2, \dots, \psi_n\}$. Řekneme, že kontrakce ψ_i , kde $i \in \mathbb{N}$, splňují tzv. *open set condition*²¹, pokud existuje neprázdná otevřená množina V taková, že platí:

- (a) $V \supseteq \Psi(V)$
- (b) a množiny $\psi_1(V), \psi_2(V), \dots, \psi_n(V)$ jsou po dvou disjunktní.

(Převzato z [4, str. 139].)

V případě Cantorova diskontinua je jeho IFS $\{\gamma_1, \gamma_2\}$ dán následujícími předpisy:

$$\begin{aligned}\gamma_1(x) &= \frac{1}{3}x, \\ \gamma_2(x) &= \frac{1}{3}x + \frac{2}{3}.\end{aligned}$$

Vezmeme-li jako otevřenou množinu otevřený interval $I = (0, 1)$, pak

$$\gamma_1(I) = \left(0, \frac{1}{3}\right) \text{ a } \gamma_2(I) = \left(\frac{2}{3}, 1\right).$$

Tedy určitě platí

$$I \supseteq \left(0, \frac{1}{3}\right) \cup \left(\frac{2}{3}, 1\right)$$

²¹Do češtiny bychom tento název mohli volně přeložit jako „Podmínka existence otevřené množiny“, avšak radši se budeme držet oficiálního názvu.

a zároveň $\gamma_1(I) \cap \gamma_2(I) = \emptyset$. Celkově tedy IFS $\{\gamma_1, \gamma_2\}$ splňuje open set condition. Podobně si lze uvědomit, že např. IFS pro Sierpiňského trojúhelník též splňuje open set condition, neboť při volbě rovnostranného trojúhelníka T jako počátečního útvaru lze za hledanou otevřenou množinu vzít vnitřek T° . Společně s tímto konceptem máme již nástroje k formulaci následující věty.

Věta 4.2.12. *Nechť (X, ρ) je metrický prostor a dále nechť je dán IFS*

$$\{\psi_1, \dots, \psi_n\}$$

splňující open set condition, kde kontrakce ψ_i má faktor r_i pro každé $1 \leq i \leq n$. Je-li $F \in \mathbb{H}(X)$ atraktorem IFS $\{\psi_1, \dots, \psi_n\}$, pak

$$\dim_H F = \dim_B F = s,$$

kde s splňuje rovnost

$$\sum_{i=1}^n r_i^s = 1.$$

Navíc platí $0 < \mathcal{H}^s(F) < \infty$.

Poznamejme, že je-li splněn předpoklad věty výše, pak box-counting dimenze útvaru F je vždy definovaná. S prominutím si dovolíme opět důkaz vynechat, neboť je poměrně pracný, avšak lze jej nalézt v knize [4, str. 140]. Tato věta nejen ospravedlňuje náš předešlý „zjednodušený“ výpočet Hausdorffovy dimenze v příkladu 2.5.14, ale dokonce nám jej podstatně zlehčuje. Pojdme se proto podívat na některé další příklady výpočtu.

Příklad 4.2.13 (Sierpiňského koberec). IFS pro Sierpiňského koberec S jsme si uvedli v tabulce 4.4. Lze si snadno rozmyslet, že tento útvar splňuje open set condition a že každá z jeho kontrakcí má faktor $r_i = 1/3$ pro $i = 1, 2, \dots, 8$. Tedy podle věty 4.2.12 lze psát

$$\sum_{i=1}^8 r_i^s = 8 \left(\frac{1}{3}\right)^s = 1.$$

Odtud již jednoduchou úpravou získáme výsledek

$$s = \dim_H S = \frac{\ln 8}{\ln 3} \approx 1,892789\dots$$

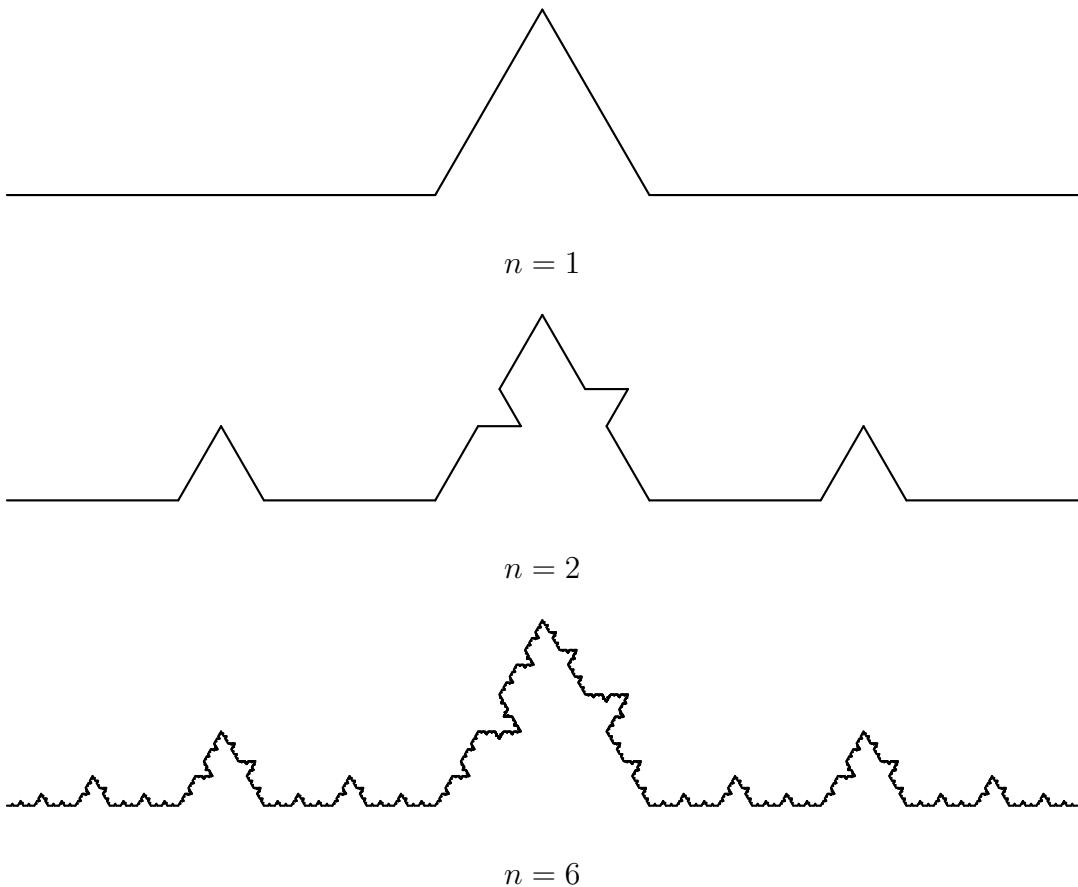
V případě ostatních dosud prezentovaných fraktálů lze podobným způsobem dojít k výsledku. Nicméně jejich IFS mají společnou jednu vlastnost, a sice tu, že všechny jejich kontrakce mají stejný faktor, což nám dosti usnadňuje celý výpočet. Pojdme se proto podívat na fraktál, kde jsou faktory kontrakcí příslušného IFS různé.

Příklad 4.2.14 (Modifikovaná Kochova křivka). Modifikujeme Kochovu křivku tak, že strany rovnostranného trojúhelníka sestrojného nad úsečkou bude mít

stranu obecné délky $0 < a < 1$. Dále navíc upravíme orientaci sestrojených trojúhelníků. Definujme IFS $\{\kappa_1, \kappa_2, \kappa_3, \kappa_4\}$ následovně:

$$\begin{aligned}\kappa_1 \begin{pmatrix} x \\ y \end{pmatrix} &= \frac{1-a}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \\ \kappa_2 \begin{pmatrix} x \\ y \end{pmatrix} &= \frac{1-a}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{1+a}{2} \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \\ \kappa_3 \begin{pmatrix} x \\ y \end{pmatrix} &= a \begin{pmatrix} \cos(\pi/3) & -\sin(\pi/3) \\ \sin(\pi/3) & \cos(\pi/3) \end{pmatrix} + \frac{1-a}{2} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= a \begin{pmatrix} 1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & 1/2 \end{pmatrix} + \frac{1-a}{2} \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \\ \kappa_4 \begin{pmatrix} x \\ y \end{pmatrix} &= a \begin{pmatrix} \cos(2\pi/3) & -\sin(2\pi/3) \\ \sin(2\pi/3) & \cos(2\pi/3) \end{pmatrix} + \frac{1+a}{2} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= a \begin{pmatrix} -1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{pmatrix} + \frac{1+a}{2} \begin{pmatrix} 1 \\ 0 \end{pmatrix}.\end{aligned}$$

Pro představu viz obrázek 4.22. Označíme-li si opět faktory pro $\kappa_1, \kappa_2, \kappa_3$ a κ_4 po



Obrázek 4.22: Modifikovaná Kochova křivka pro $a = 1/5$

řadě r_1, r_2, r_3, r_4 , pak je zjevně

$$r_1 = r_2 = \frac{1-a}{2}, \quad r_3 = r_4 = a.$$

Tedy Hausdorffova dimenze, resp. box-counting dimenze s je řešením rovnice

$$2a^s + 2 \cdot \left(\frac{1-a}{2}\right)^s = 1.$$

Tu obecně algebraicky řešit nelze s výjimkou $a = 1/3$, což by byl případ standardní Kochovy křivky. Pro $a = 1/5$ lze numericky dopočítat přibližné řešení $s \approx 1,1601$.

(Převzato a upraveno z [4, str. 142].)

Příklady výpočtů pro ostatní prezentované fraktální útvary si může čtenář vyzkoušet sám, neboť nejsou nikterak složité. V některých případech může být obtížnější určit kontraktivní faktor daných zobrazení. Asi nejfektivější způsob (nechceme-li faktor počítat přímo z definice) je přes tzv. *SVD rozklad*²² a *singulární čísla*²³ matice \mathbf{A} . Tím se zde ale již zabývat nebudeme (pro podrobnosti však doporučujeme knihu [6]).

4.3 Time Escape algoritmy

Pokud se čtenář dostal až do této části, mohl si všimnout, že jednu kategorii fraktálů jsme zatím zcela vynechali. Přitom právě ta je z velké části zodpovědná za popularitu, které se těší toto odvětví matematiky. Jedná se zejména o *Mandelbrotu množinu* (viz obrázek 4.23) pojmenovanou po samotném zakladateli fraktální geometrie. Přitom její definice není v konečném důsledku nikterak složitá.

S pravděpodobně nejznámějším fraktálem však souvisí dvojice širších termínů, se kterými začneme, jimiž jsou

- *Juliovy množiny*
- a *Fatouovy množiny*.

Jsou pojmenované po francouzských matematicích GASTONU JULIOVI²⁴ a PIERRU FATOUVI²⁵. Pro jejich studium se však budeme muset navštívit svět komplexních čísel.

Lze nejspíše předpokládat, že se čtenář s komplexními čísly již setkal. Nebudeme se tedy společně hlouběji zabývat naprostými základy. Pouze si stručně připomeňme značení.

- *Komplexním číslem* rozumíme číslo $z = a + bi$, kde $a, b \in \mathbb{R}$ a $i^2 = -1$. Množinu komplexních čísel, jak je zvykem, budeme značit \mathbb{C} .
- *Komplexně sdruženým číslem* k číslu $z = a + bi$ rozumíme číslo

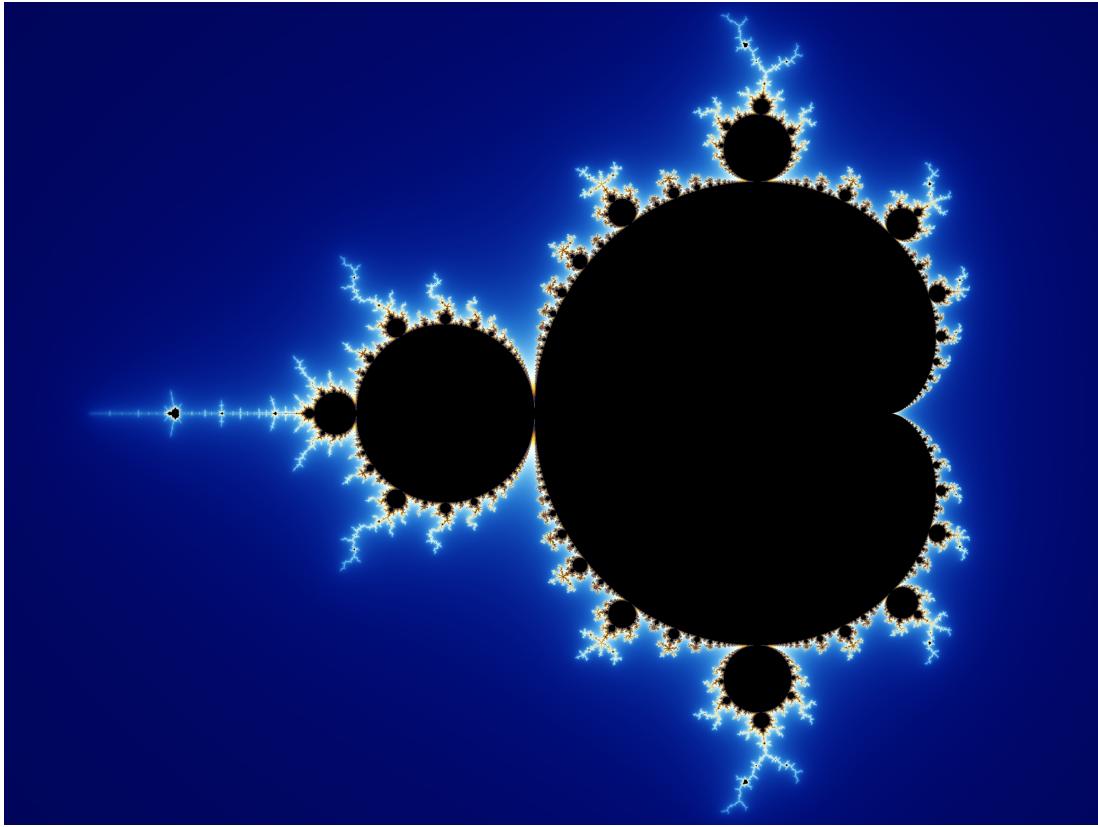
$$\bar{z} = a - bi.$$

²²Zkratka pro *Singular Value Decomposition*

²³Ta lze rovněž získat pomocí vlastních čísel matice \mathbf{A} .

²⁴1893–1978

²⁵1878–1929



Obrázek 4.23: Mandebrotova množina (Převzato z Wikipedia Commons, viz [odkaz](#))

- *Absolutní hodnotou komplexního čísla $z = a + bi$ rozumíme vzdálenost od počátku, tj.*

$$|z| = \sqrt{a^2 + b^2}.$$

V této sekci budeme především pracovat s komplexními polynomiálními funkcemi $f : \mathbb{C} \rightarrow \mathbb{C}$, tzn. funkcemi ve tvaru

$$f(z) = \sum_{i=1}^n a_i z^i = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0, \quad \text{kde } a_i \in \mathbb{C}$$

Brzy zjistíme jejich důležitou roli.

4.3.1 Juliovy a Fatouovy množiny

Většinový výklad v této části je převzat z knihy [4, str. 235], avšak řadu věcí si opět dovolíme přeskočit. Dále ještě připomeňme značení, které jsme zavedli již v části 4.2.1. Obecně n -tou iteraci funkce $f : \mathbb{C} \rightarrow \mathbb{C}$ budeme značit

$$f^{\circ n}(z) = (f \circ f^{\circ(n-1)})(z) = f(f^{\circ n}(z))$$

kde $z \in \mathbb{C}$. Speciálně $f^{\circ 0} = \text{id}$ a $f^{\circ 1} = f \circ \text{id} = f$.

Definice 4.3.1 (Juliova a Fatouova množina). Mějme polynomiální funkci $f : \mathbb{C} \rightarrow \mathbb{C}$. Pak definujeme

(a) vyplněnou Juliovu množinu polynomiální funkce f

$$K(f) = \{z \in \mathbb{C} \mid f^{\circ n}(z) \not\rightarrow \infty\}.$$

(b) Juliovu množinu polynomiální funkce f

$$J(f) = \partial K(f).$$

(c) Fatouovu množinu polynomiální funkce f

$$F(f) = \mathbb{C} \setminus J(f).$$

Co nám definice 4.3.1 vlastně říká? U pevně zadané funkce $f(z) = \sum_{i=1}^n a_i z^i$ pro zadaný bod $z \in \mathbb{C}$ zkoumáme, zda je posloupnost²⁶ jejích postupných iterací $\{f^{\circ n}(z)\}_{n=1}^{\infty}$ omezená. Později uvidíme, že Juliova množina má v typickém případě charakter fraktálu.

Příklad 4.3.2. • Uvažujme funkci $f(z) = z^k$. Pak pro její n -tou iteraci platí $f^{\circ n}(z) = z^{k^n}$. Tedy

$$K(f) = \{z \in \mathbb{C} \mid |z| < 1\},$$

resp.

$$J(f) = \{z \in \mathbb{C} \mid |z| = 1\}.$$

Juliova množina funkce f tvoří jednotkovou kružnici se středem v počátku Gaussovy roviny, resp. vyplněná Juliová množina $K(f)$ tvoří jednotkový kruh.

- Pro $f(z) = z^2 + c$, kde $c \in \mathbb{C}$ je jedna z možných variant množiny $K(f)$ znázorněna na obrázku 4.24a.
- Ještě jiný příklad pro složitější polynom $f(z) = z^3 - z^2$ lze vidět na obrázku 4.24b.
- Pro příklady Juliových množin zadaných polynomiálních funkcí viz obrázky 4.25a a 4.25b.

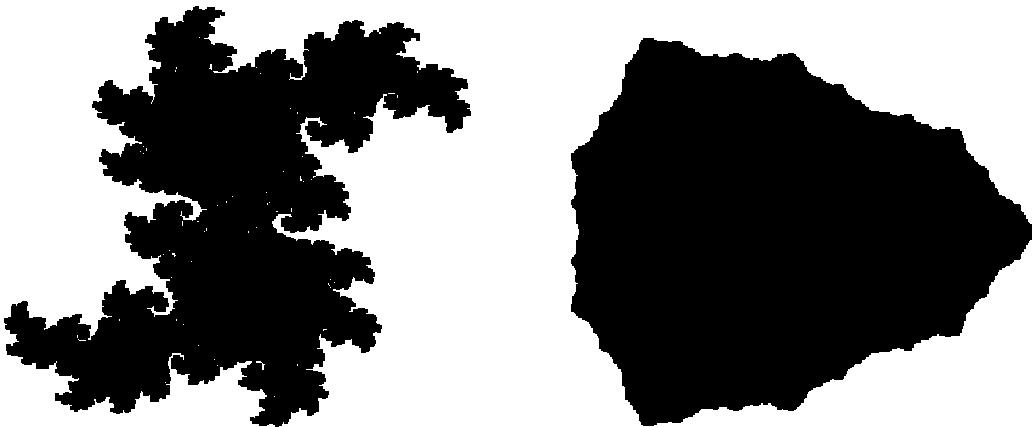
Jak jsme již konstatovali, u polynomiální funkce f nás pro zadaný bod $z \in \mathbb{C}$ pouze zajímá, zda posloupnost postupných iterací je omezená.

V souvislosti se zmíněnými záležitostmi si nyní dokážeme několik základních poznatků. Řadu dalších však vynecháme, neboť jsou příliš technické, především ty využívající znalostí z komplexní analýzy. Začneme tvrzením, které se nám později bude hodit při generování tohoto typu fraktálů.

Lemma 4.3.3. Nechť je dána polynomiální funkce $f : \mathbb{C} \rightarrow \mathbb{C}$ stupně $n \geq 2$. Pak platí následující:

- (i) Existuje $r \in \mathbb{R}$ takové, že pokud platí $|z| \geq r$, pak $|f(z)| \geq 2|z|$.

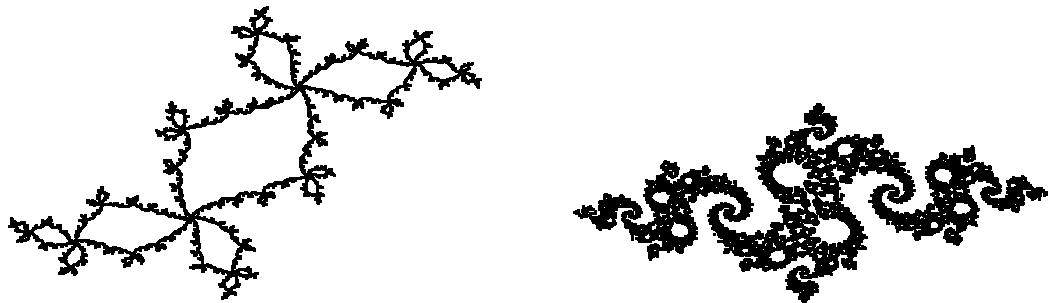
²⁶Poznamenejme, že limitu posloupnosti komplexní čísel zde chápeme (stejně jako v ostatních případech) jako limitu posloupnosti v metrickém prostoru, jmenovitě (\mathbb{C}, ρ_e) .



(a) $f(z) = z^2 + 0.35 + 0.35i$

(b) $f(z) = z^3 - z^2$

Obrázek 4.24: Příklady approximace $K(f)$



(a) $f(z) = z^2 + 0,7885 \cdot e^{i \cdot 9\pi/16}$

(b) $f(z) = z^2 - 0,8 + 0,156i$

Obrázek 4.25: Příklady approximace $J(f)$

(ii) Navíc pokud existuje $m \in \mathbb{N}$ takové, že platí-li $|f^{\circ m}(z)| \geq r$, pak $f^{\circ k}(z) \rightarrow \infty$ pro $k \rightarrow \infty$.

Důkaz. Mějme polynomiální funkci

$$f(z) = \sum_{i=1}^n a_i z^i$$

kde $z \in \mathbb{C}$ volíme libovolně. Volme $r \in \mathbb{R}$ takové, že je-li splněno $|z| \geq r$, pak platí nerovnost $\frac{1}{2}|a_n||z|^n \geq 2|z|$ a zároveň

$$|a_{n-1}||z|^{n-1} + |a_{n-2}||z|^{n-2} + \cdots + |a_1||z| + |a_0| \leq \frac{1}{2}|a_n||z|^n.$$

Předpokládejme tedy, že $|z| > r$. Pak z trojúhelníkové nerovnosti plyne

$$\begin{aligned} |f(z)| &= \left| \sum_{i=0}^n a_i z^i \right| \geq |a_n||z|^n - \sum_{i=0}^{n-1} |a_i||z|^i \geq |a_n||z|^n - \frac{1}{2}|a_n||z|^n \\ &= \frac{1}{2}|a_n||z|^n \geq 2|z|. \end{aligned}$$

Pokud navíc platí, že existuje $m \in \mathbb{N}$ takové, že $|f^{\circ m}(z)| \geq r$, pak indukcí lze odvodit, že pro $m+k$, kde $k \in \mathbb{N}$, platí

$$|f^{m+k}(z)| \geq 2^k |f^{\circ m}(z)| \geq r,$$

neboli $f^{\circ k}(z) \rightarrow \infty$. □

(Převzato z [4, str. 237].)

V čem je lemma 4.3.3 užitečné? V podstatě nám říká, že pro test „chování“ posloupnosti $\{f^{\circ k}(z)\}_{n=1}^{\infty}$ stačí kontrolovat, zda absolutní hodnota obrazu v k -té iteraci není vyšší, než nějaké r , jehož existenci nám toto lemma zaručuje. Pokud ano, je již jasné, že posloupnost bude divergovat. Toho využijeme hlavně při generování fraktálů (viz kapitola 5).

Algoritmem pro generování tohoto typu fraktálů se proto také říká *Time Escape*, neboť pro každý bod z po určité době (resp. pevném počtu iterací) algoritmus zastaví a zkонтroluje, zda $|z| > r$. Podle výsledku buď daný bod zařadí do zkoumané množiny, nebo ne.

Příklad 4.3.4. Vezměme si funkci $f(z) = z^2 + c$, kde $c \in \mathbb{C}$. Podle věty 4.3.3 hledáme vhodné $r \in \mathbb{R}$, aby pro $|z| > r$ fungovalo, že $|z^2 + c| \geq |z|^2 - |c| \geq 2|z|$. Tedy řešíme nerovnici

$$r^2 - |c| \geq 2r.$$

Výpočtem diskriminantu $D = 4 + 4|c|$ a aplikací známého vzorce dostaneme

$$r \geq \frac{-2 + 2\sqrt{1 + |c|}}{2} = -1 + \sqrt{1 + |c|}.$$

Je celkem jasné, že záporný kořen zde nemá smysl. Zároveň však potřebujeme zajistit, aby platilo

$$\begin{aligned} r^2 - |c| &\geq 2r, \\ r^2 - 2r &\geq |c| \geq 0, \\ r(r - 2) &\geq 0. \end{aligned}$$

To je splněno pro $r \in (2, \infty)$. Pro test divergence posloupnosti $\{f^{\circ k}(z)\}_{n=1}^{\infty}$ stačí kontrolovat, zda pro nějaké k platí, že

$$|f^{\circ k}(z)| \geq r.$$

Pro funkce tohoto tvaru se často volí právě $r = 2$.

Věta 4.3.5. Nechť f je komplexní polynomální funkce. Pak

- (i) $K(f)$ a $J(f)$ jsou neprázdné kompaktní množiny,
- (ii) $J(f) \subseteq K(f)$,
- (iii) $(J(f))^{\circ} = \emptyset$.

Důkaz. Z lemmatu 4.3.3 víme, že existuje $r \in \mathbb{R}$ takové, že pokud pro nějaké $m \in \mathbb{N}$ platí $|f^{\circ m}(z)| \geq r$, pak $f^{\circ k} \rightarrow \infty$. Tedy množiny $K(f)$ a $J(f)$ jsou obsaženy v kouli o poloměru r umístěné v počátku, tj.

$$K(f), J(f) \subseteq B_r(0).$$

Volme $z \in \mathbb{C} \setminus K(f)$, tzn. $f^{\circ k}(z) \rightarrow \infty$. Tedy nutně existuje $m \in \mathbb{N}$ takové, že $|f^{\circ m}(z)| > r$. Protože však f je spojitá funkce, existuje $\varepsilon > 0$ takové, že pro každé $w \in B_\varepsilon(z) \setminus \{z\}$ platí $f^{\circ k}(w) \rightarrow \infty$, a tedy $w \in \mathbb{C} \setminus K(f)$. To znamená, že množina $\mathbb{C} \setminus K(f)$ je otevřená, nebo-li $K(f)$ je uzavřená množina (totéž platí i pro $J(f)$). Podle Heineho-Borelovovy věty 2.1.19 jsou tedy $K(f)$ a $J(f)$ kompaktní²⁷. Podle základní věty algebry má rovnice $f(z) = z$ alespoň jedno řešení, označme jej z_0 . Tedy $f^{\circ k}(z_0) = z_0$, kde $k \in \mathbb{N}$, z čehož plyne, že $z_0 \in K(f)$, tzn. $K(f) \neq \emptyset$.

Dále si zvolme $z_1 \in \mathbb{C} \setminus K(f)$ a uvažujme funkci $l(\lambda) = \lambda z_0 + (1 - \lambda)z_1$, kde $0 \leq \lambda \leq 1$. Tedy l představuje spojnici bodů z_0 a z_1 . Položme

$$\lambda' = \sup \{ \lambda \in \langle 0, 1 \rangle \mid l(\lambda) \in K(f) \}.$$

Tzn. musí ležet na její hranici, tedy $l(\lambda') \in \partial K(f) = J(f)$. Z toho plyne, že i $J(f)$ je neprázdná množina. Tím jsou dokázány body (i) a (ii).

Poslední bod (iii) dokážeme sporem. Pokud by platilo $(J(f))^\circ \neq \emptyset$, pak by existovala neprázdná otevřená množina $U \subseteq J(f) \subseteq K(f)$. To by znamenalo, že $U \subseteq (K(f))^\circ$ a zároveň má U neprázdný průnik s hranicí $K(f)$, což je spor. \square

(Převzato z [4, str. 237].)

Zde poznamenejme, že body (ii) a (iii) jsou ve skutečnosti speciálními případy obecnějších tvrzení 2.1.21 a 2.1.22.

Věta 4.3.6. *Je-li f libovolná komplexní polynomická funkce, pak $J(f)$ je invariantní vůči f . Tzn.*

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

Důkaz. Nechť $z \in J(f)$. Z bodu (ii) tvrzení 4.3.5 plyne, že $f^{\circ k}(z) \not\rightarrow \infty$ (protože $z \in K(f)$). Najdeme posloupnost $\{\omega_n\}_{n=1}^\infty$, takovou, že $\omega_n \rightarrow z$ a zároveň $f^{\circ k}(\omega_n) \rightarrow \infty$ pro všechna $n \in \mathbb{N}$ (tzn. $\{\omega_n\}_{n=1}^\infty$ je posloupnost v $\mathbb{C} \setminus K(f)$). Ze spojitosti funkce f plyne, že $f(\omega_n) \rightarrow f(z)$, přičemž $f^{\circ k}(f(z)) = f^{\circ(k+1)}(z) \not\rightarrow \infty$ a $f^{\circ k}(f(\omega_n)) = f^{\circ(k+1)}(\omega_n) \rightarrow \infty$ pro každé $n \in \mathbb{N}$. Z toho plyne, že $f(z) \in J(f)$ (protože $f^{\circ k}(\omega_n) \rightarrow \infty$ pro každé $n \in \mathbb{N}$, tzn. $f(\omega_n)$ neleží ve vnitřku $K(f)$) a tedy $f(J(f)) \subseteq J(f)$. Dále rovněž máme

$$J(f) \subseteq f^{-1}(f(J(f))) \subseteq f^{-1}(J(f)),$$

kde první inkluze platí pro obecné zobrazení a druhá plyne z předchozího, tj. $f(J(f)) \subseteq J(f)$.

Druhou inkluzi $f^{-1}(J(f)) \subseteq J(f)$ ukážeme obdobně. Najdeme z_0 takové, že $f(z_0) = z \in J(f)$ a následně definujme posloupnost $\{\omega_n\}_{n=1}^\infty$ jako výše, tj.

²⁷Heineho-Borelovu větu jsme sice formulovali pro \mathbb{R}^n , avšak $\mathbb{C} \simeq \mathbb{R}^2$.

$\omega_n \in \mathbb{C} \setminus K(f)$ pro každé $n \in \mathbb{N}$ a zároveň $\omega_n \rightarrow z$. Poté můžeme nalézt posloupnost $\{v_n\}_{n=1}^{\infty}$, tak, aby platilo $f(v_n) = \omega_n$ pro každé $n \in \mathbb{N}$, tzn.

$$f(v_n) = \omega_n \rightarrow z = f(z_0).$$

Zároveň však platí $f^{\circ k}(z_0) = f^{\circ(k-1)}(z) \not\rightarrow \infty$ a $f^{\circ k}(v_n) = f^{\circ(k-1)}(\omega_n) \rightarrow \infty$, tedy $z_0 \in J(f)$ (argument stejný jako výše). Z toho plyne, že $f^{-1}(J(f)) \subseteq J(f)$, neboli $J(f) \subseteq f(f^{-1}(J(f))) \subseteq f(J(f))$. \square

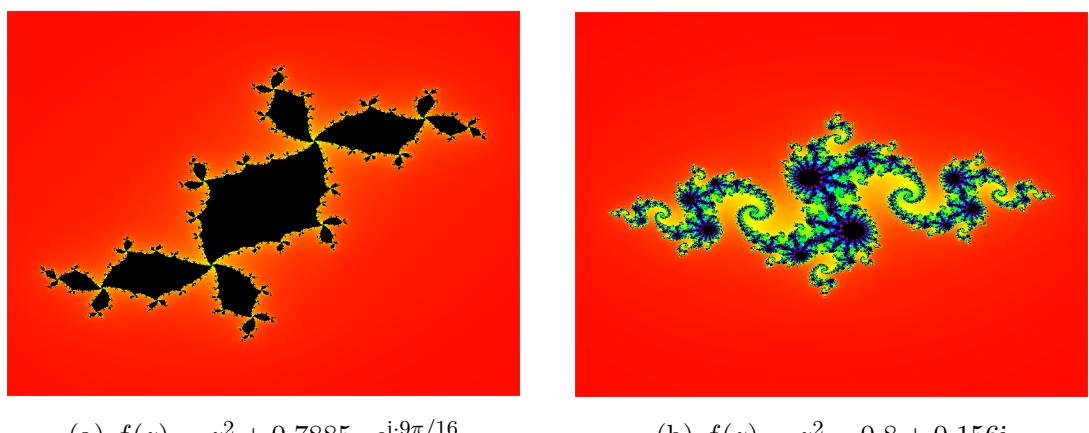
Věta 4.3.7. Nechť f je libovolná polynomiální funkce. Pak $J(f) = J(f^{op})$ pro libovolné $p \in \mathbb{N}$.

Důkaz. Z lemmatu 4.3.3 plyne, že $f^{ok}(z) \rightarrow \infty$ právě tehdy, když $(f^{\circ p})^{ok}(z) = f^{\circ kp}(z) \rightarrow \infty$. Tedy $K(f) = K(f^{\circ p})$, z čehož plyne $J(f) = J(f^{\circ p})$. \square

(Převzato z [4, str. 238].)

Pro základní přehled nám tyto vlastnosti budou stačit. Pro další informace doporučujeme se podívat do citované literatury.

Vraťme se k obrázkům 4.24 a 4.25. Vzhledem k rozšířenosti tohoto typu fraktálů (a to mnohdy i mezi širší matematickou zainteresovanou veřejností) si čtenář možná mohl všimnout, že zatímco v těchto případech jsme body naležící příslušné (vyplňené) Juliově množině vybarvili černou barvou, na obrázku 4.23 je zbarvení podstatně sofistikovanější²⁸. Příklady takového způsobu znázornění si lze prohlédnout na obrázcích 4.26a a 4.26b. Obrázky jsou opět vygenerovány pomocí přiloženého programu.



Obrázek 4.26: Příklady zbarvení bodů pro různé $K(f)$

Barvu jednotlivým bodům $z \in \mathbb{C}$ přidělujeme podle toho, kolik iterací dané polynomiální funkce f proběhlo, než hodnota $|f^{\circ k}(z)|$ překročila zadanou mez r (viz lemma 4.3.3). Vždy volíme pevný počet iterací funkce f , který aplikujeme na každý bod \mathbb{C} (resp. na nějakou její podmnožinu). Podle zvoleného počtu se pak bude odvíjet i přesnost naší approximace, avšak je třeba zvážit i výpočetní náročnost, která (celkem očekávatelně) roste se zvyšujícím se počtem iterací. Tomuto tématu (a mnohým dalším) se budeme věnovat v kapitole 5.

²⁸A leckdo by jej nejspíše shledal i hezčím.

4.3.2 Mandelbrotova množina

Mandelbrotova množina patří patrně mezi jedny z nejznámějších fraktálů a stala se oblíbeným motivem nemalého množství publikací a popularizačních materiálů. Její mimořádně složitě vypadající vzhled a charakter přitom popisuje až překvapivě jednoduché pravidlo. Nejdříve si zavedeme následující funkci:

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

kde $c \in \mathbb{C}$. Tento konkrétní tvar polynomální funkce jsme již viděli v minulé podsekci, kde jsme se zabývali tzv. *Juliovými množinami*. Ty tvořily hranici množiny všech $z \in \mathbb{C}$ takových, že posloupnost $\{f_c^{\circ k}(z)\}_{k=1}^{\infty}$ je omezená. Ani zde se od původní myšlenky nevzdálíme, avšak budeme nyní zkoumat proměnnou c .

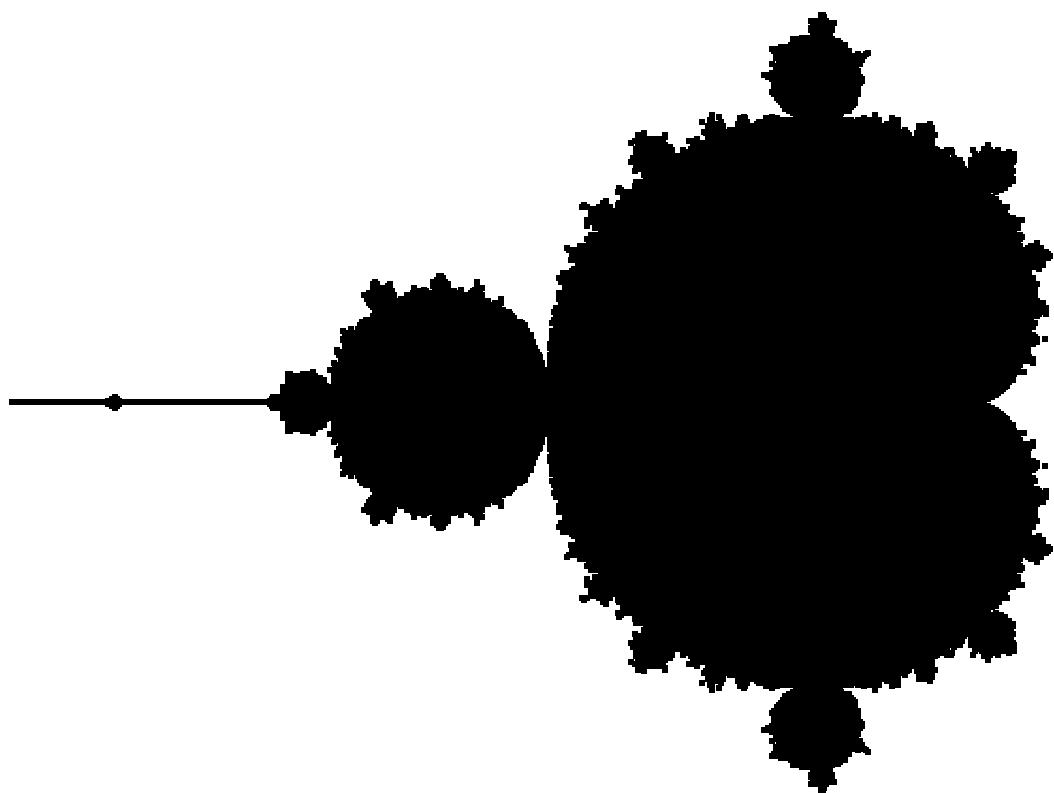
Definice 4.3.8 (Mandelbrotova množina). *Mandelbrotovu množinu* definujeme jako

$$\mathfrak{M} = \left\{ c \in \mathbb{C} \mid \left\{ f_c^{\circ k}(0) \right\}_{k=1}^{\infty} \not\rightarrow \infty \right\}.$$

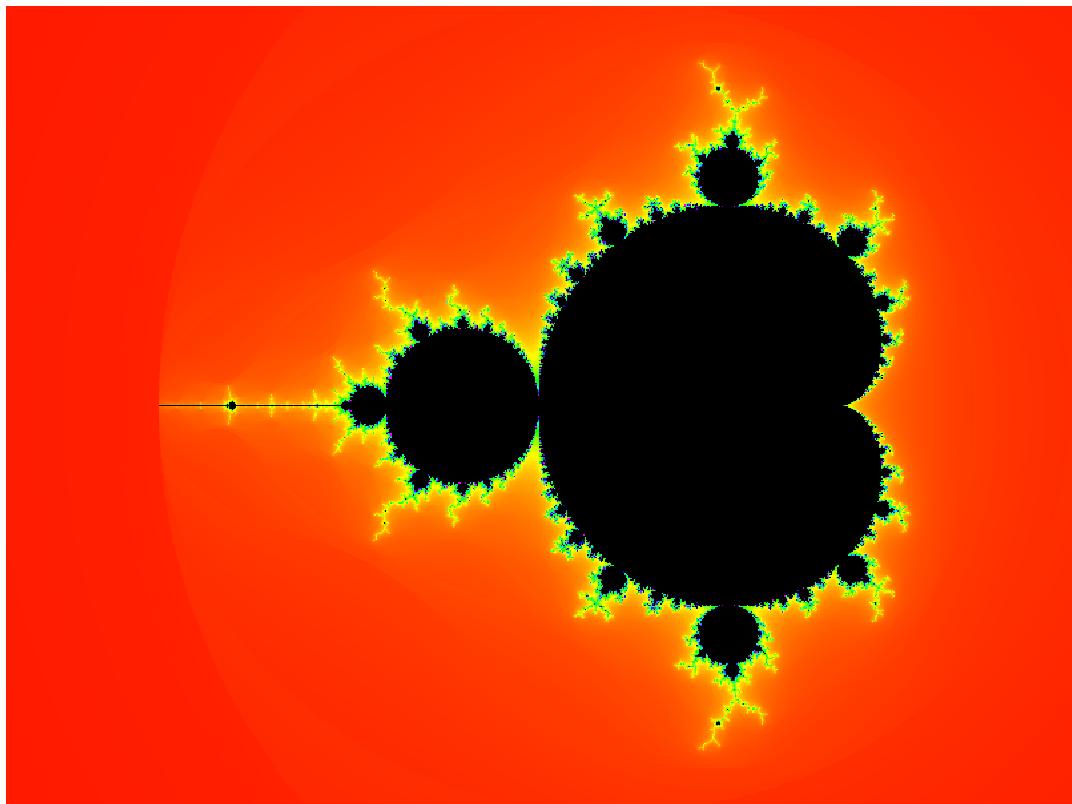
Upozorněme explicitně na fakt, že iterace vždy začínají pevně v bodě $z = 0$ a \mathfrak{M} je množina právě všech parametrů c , pro které tyto iterace nedivergují do nekonečna. Lze se setkat i s jinými ekvivalentními definicemi. Např. \mathfrak{M} lze charakterizovat i takto:

$$\mathfrak{M} = \{c \in \mathbb{C} \mid J(f_c) \text{ je souvislá}\}.$$

Důkaz je opět delší, takže se znovu odkážeme na knihu [4, str. 245]. Jak je to s dimenzí? Tu lze odhadnout velice snadno. Zjevně platí, že $\dim_H \mathfrak{M} \leq 2$. Zároveň však platí i druhá nerovnost $\dim_H \mathfrak{M} \geq 2$, neboť \mathfrak{M} obsahuje jako podmnožinu kouli $B_r(0)$, jejíž Hausdorffova dimenze je 2. To zároveň silně podtrhuje naše konstatování neexistence jednotné definice termínu „fraktál“, což jsme krátce rozebírali v sekci 1.4. Některé definice říkají, že za fraktální útvar považujeme soběpodobný útvar, jehož Hausdorffova dimenze je neceločíselná, avšak zde vidíme, že v tom případě bychom Mandelbrotovu množinu nemohli považovat za fraktál.



Obrázek 4.27: Znázornění aproximace množiny \mathfrak{M}



Obrázek 4.28: Barevné znázornění approximace množiny \mathfrak{M}

Kapitola 5

Generování fraktálů

V této poslední kapitole navážeme na znalosti z kapitoly předchozí. Proto zde opět čtenáři doporučujeme se podívat na její obsah (vyjma sekcí týkajících se matematického základu, který zde nebude tak podstatný). Předložíme zde stručný teoretický rozbor algoritmů pro generování fraktálních útvarů, i na jejich praktickou implementaci. Než tak však učiníme, krátce zmíníme způsob, jakým budeme rozbory vůbec provádět.

Vzorový způsob implementace všech zde zmíněných algoritmů lze nalézt v připojeném programu.

5.1 Stručně k zápisu programů

V době psaní tohoto textu existuje mnoho programovacích jazyků a nejspíše lze bezpečně předpokládat, že budou přibývat další. Nikoho tak nejspíše nepřekvapí, že vzhledem k současnému (dosti rychlému) vývoji v oblasti informatiky mnoho jazyků, které dříve byly považovány za nové a inovativní, postupně zastaraly a jiné pro mnoho jedinců dokonce upadly v zapomnění. Avšak jiné si naopak svoji pozici drží dodnes. Pro účely tohoto textu byl v rámci praktických ukázek, které uvidíte, zvolen jazyk **Python**, neboť jeho syntaxe není složitá¹ a zároveň tak není příliš obtížné si v mnoha případech domyslet význam jednotlivých příkazů². Zároveň však poznamenejme, že stejně jako v případě matematické části tohoto textu, i zde budeme pracovat s předpokladem, že čtenář je seznámen se základními koncepty programování a algoritmizace všeobecně. Nebudeme se zde tedy řešit, co je to proměnná, pole (resp. v Pythonu seznam), funkce, podmínky nebo základy objektově orientovaného programování.

Zároveň bychom neměli zapomínat na zájemce používající jiné programovací jazyky. Proto kromě praktických ukázek si prezentované algoritmy uvedeme i pomocí tzv. *pseudokódu*. Pseudokód nepředstavuje sám o sobě žádnou formu progra-

¹Složitost programovacího jazyka je, z pochopitelných důvodů, dosti subjektivní pojem, neboť závisí i na zkušenostech programátora.

²Samozřejmě nelze v tomto ohledu mluvit za každého (potenciálního) čtenáře. Pokud by tak kdykoliv vznikla nějaká nejasnost ohledně významu použitých příkazů, lze se podívat na stránky oficiální dokumentace jazyka Python: <https://docs.python.org>.

movacího jazyka. Jedná se o čistě abstraktní popis, který lze však s minimálním úsilím přepsat do libovolného programovacího jazyka. Jednoduchým příkladem pseudokódu je např. 5.1.1.

Algoritmus 5.1.1: Ukázkový pseudokód (hledání minima)

Vstup: Seznam čísel x_1, x_2, \dots, x_n .

```
1 max ←  $x_1$ ;
2 for  $i = 1, 2, \dots, n$  do
3   if  $x_i > \text{max}$  then
4     max ←  $x_i$ 
5 return max
```

Nejdříve si vysvětleme některé použité značení:

- symbol \leftarrow používáme pro operaci přiřazení hodnoty (místo standardního $=$),
- naopak symbol $=$ budeme používat ve smyslu porovnávání hodnot,
- indexy (u řetězců a polí) v rámci pseudokódu budeme počítat od jedné.

Implementace zmíněného algoritmu např. právě v jazyce Python si čtenář může prohlédnout u programu 5.1.2. Pochopitelně se v konkrétní implementaci mohou

```
1 def findMax(numbers: list) -> int:
2     maximum = numbers[0]
3     for i in range(len(numbers)):
4         number = numbers[i]
5         if number > maximum:
6             maximum = number
7     return maximum
```

Program 5.1.2: Možná implementace algoritmu 5.1.1

vyskytovat různé odchylky. Např. v programu 5.1.2 využíme proměnnou `number`, kterou bychom jistě mohli vypustit a pracovat přímo se seznamem `numbers`, resp. jeho i -tým prvkem `numbers[i]`. Nebo bychom například nemuseli program vůbec zapisovat jako funkci, či bychom mohli např. jinak pojmenovat proměnné (viz program 5.1.3). To jsou však v celkovém kontextu pouhé drobnosti. V rámci textu se ovšem budeme snažit držet jednotné konvence, tedy že programy budeme vždy psát jako funkce/procedury/metody a proměnné budeme pojmenovávat vždy v angličtině, nebot je to při programování zkrátka zvyklost.

5.2 Implementace L-systémů a želví grafiky

Na základní princip L-systémů jsme se podívali v části 4.1. Jejich implementace je do jisté míry přímočará. Řešení je potřeba rozdělit na dvě části: implementace *samotného L-systému* a *želví grafiky*.

```

1 count = int(input("Enter number count: "))
2 if (count <= 0): exit(0)
3
4 numbers = [int(input("Enter a number: "))]
5 for _ in range(count)
6
7 max_value = numbers[0]
8 for i in range(len(numbers)):
9     if numbers[i] > max_value: max_value = numbers[i]
10
11 print(f"Max value: {max_value}")

```

Program 5.1.3: Jiná možná implementace algoritmu 5.1.1

5.2.1 L-systémy

L-systém nepředstavuje nikterak složitou matematickou strukturu. Z definice (viz 4.1.10) je potřeba znát pouze používané *neterminály*, *axiom* a seznam přepisovacích pravidel. O to jednodušší je situace, započítáme-li fakt, že neterminální symboly (resp. jejich význam) v případě námi používaných L-systémů jsou pevně dané, tedy není třeba je explicitně uvádět v definici. L-systém lze tak implementovat jako jednoduchou třídu s atributy `word` obsahující aktuální slovo po k -té iteraci a slovník pravidel `rules` (viz program 5.2.1). Slovní pravidel `rules` má

```

1 class LSystem:
2     def __init__(self, axiom: str, rules: dict) -> None:
3         self._word = axiom
4         self._rules = rules

```

Program 5.2.1: Konstruktor třídy pro L-systém

jednoduchou strukturu. Klíče tvoří levé strany pravidel a k nim přiřazené hodnoty naopak tvoří pravé strany pravidel. Jeho vzhled může vypadat např. takto:

```

rules = {
    "X": "F-[X]+X]+F[+FX]-X",
    "F": "FF"
}

```

Poměrně zásadní pro nás však bude především metoda pro aplikaci jednotlivých pravidel. Pro další výklad si však zavedeme pohodlnější zápis řetězců, který je v programování zcela běžný.

Definice 5.2.1. Nechť $\alpha = x_1x_2 \dots x_n$ je slovo nad libovolnou abecedou $\Sigma \neq \emptyset$. Pak pro každé $1 \leq i \leq n$ definujeme $\alpha[i] = x_i$.

Myšlenka je velice intuitivní. Obecně máme-li řetězec w po m -té iteraci a množinu přepisovacích pravidel $P \subseteq \{a \rightarrow \alpha \mid a \in V, \alpha \in V^*\}$, kde V je abeceda, pak stačí pro každý znak $w[i]$, kde $1 \leq i \leq n$, pouze zkontovalovat, zda není na levé straně nějakého pravidla v P . Pokud ano, dojde k aplikaci příslušného

pravidla³. Viz pseudokód 5.2.2. Implementace je, vzhledem k dostupným funk-

Algoritmus 5.2.2: Algoritmus pro k -tou iteraci slova w

Vstup: Množina pravidel P , slovo w , číslo $k \in \mathbb{N}$

```
1  $w_{\text{prev}} \leftarrow \lambda, w_{\text{new}} \leftarrow w;$ 
2 for  $m = 1, 2, \dots, k$  do
3    $w_{\text{prev}} \leftarrow w_{\text{new}};$ 
4    $w_{\text{new}} \leftarrow \lambda;$ 
5   for  $i = 1, 2, \dots, |w_{\text{prev}}|$  do
6     if existuje pravidlo tvaru  $(w_{\text{prev}}[i] \rightarrow \alpha) \in P$  then
7        $w_{\text{new}} \leftarrow w_{\text{new}}[1] \dots w_{\text{new}}[i-1]\alpha;$ 
8     else
9        $w_{\text{new}} \leftarrow w_{\text{new}}w[i];$ 
10 return  $w_{\text{new}};$ 
```

Výstup: Slovo w_{new} odvozené po k iteracích ze slova w

cím v Pythonu, až překvapivě jednoduchá. O tom se čtenář může přesvědčit sám v případě kódu 5.2.3. Pojdme si stručně rozebrat použité funkce, resp. metody,

```
1 def iterate(self, iteration_count: int) -> None:
2   for _ in range(iteration_count):
3     self._word = self._word.translate(str.maketrans(self.
4       _rules))
```

Program 5.2.3: Implementace algoritmu 5.2.2

v programu 5.2.3.

- `str.maketrans` vytvoří ze zadaného slovníku překladovou tabulkou pro metodu `translate`. Její struktura odpovídá slovníku obsahující dvojice (*Unicode hodnota, znak*).
- `translate` nahradí každý ze znaků řetězcem uvedeným v překladové tabulce.

Tímto způsobem lze implementovat třídu, kde vygenerujeme příslušný řetězec na základě poskytnutých přepisovacích pravidel, a který následně budeme interpretovat pomocí želví grafiky.

5.2.2 Želví grafika

Druhou částí je naprogramování želví grafiky. Nyní pracujeme se scénářem, že máme vygenerovaný příslušný řetězec znaků w , jehož znaky chceme interpreto-

³Technicky vzato jsme z formálních důvodů v definici L-systému 4.1.10 přidali i pravidla tvaru $a \rightarrow a$, aby nedošlo k situaci, že pro a neexistuje pravidlo. Avšak z hlediska praktické implementace toto není překážkou, neboť v případě absence takového pravidla jednoduše symbol přeskočíme.

vat. Za účelem zjednodušení se pokusíme striktně oddělit samotnou *geometrickou interpretaci řetězce* od jeho *grafické interpretace*.

Pro připomenutí významů jednotlivých symbolů doporučujeme se znovu podívat do tabulek 4.1 a 4.2. Nejdříve si však ujasněme, jaké informace si potřebujeme o želvě uchovávat.

- Vzdálenost d , o kterou se želva při každém kroku posune,
- aktuální pozice želvy (x, y) ,
- úhel $\alpha \in \langle 0, 2\pi \rangle$ udávající směr želvy
- přírůstek úhlu δ ,
- seznam nakreslených úseček prezentovaných uspořádanými čtvericemi

$$(x_0, y_0, x_1, y_1),$$

kde (x_0, y_0) a (x_1, y_1) jsou souřadnice počátečního, resp. koncového bodu.

Podobně jako v případě L-systému, i zde můžeme želvu reprezentovat jako třídu (viz program 5.2.4).

```

1 class Turtle:
2     def __init__(self, step: float, position: Vector = Vector
3                  (0, 0), angle: float = 0) -> None:
4         self._position = position
5         self._step = step
6         self._angle = (angle % 360) * math.pi / 180
7         self._pen_down = False
8         self._lines = []
9
10        self._x_min, self._y_min = position.x, position.y
11        self._x_max, self._y_max = position.x, position.y

```

Program 5.2.4: Konstruktor třídy pro želvu

V tomto případě zvolíme při implementaci želvy následující strategii. Představíme si ji tak, že na sobě připevněné pero a budeme si pouze pamatovat, zda je či není v danou chvíli položeno na plátně. Pokud ano a želva provede krok vpřed, nakreslí za sebou úsečku.

Všechny atributy jsou vysvětleny níže.

- `self._position` uchovává pozici želvy (x, y) ,
- `self._step` reprezentuje velikost kroku d ,
- `self._angle` je počáteční úhel otočení želvy přepočítaný v radiánech,
- `self._pen_down` udává, zda je pero položeno na plátně.
- `self._lines` ukládá seznam dosud nakreslených úseček.

V konstruktoru třídy `Turtle` se navíc nachází soukromé atributy

```
self._x_min, self._x_max, self._y_min a self._y_max.
```

Ty nám budou sloužit pro pozdější vykreslování výsledného útvaru. Průběžně si v nich budeme uchovávat minimální, resp. maximální souřadnici x a y ze všech dosud vygenerovaných úseček. Tyto informace se nám budou hodit pro záverečnou manipulaci s výsledným obrazcem.

Jednotlivé akce želvy lze v rámci třídy naprogramovat poměrně snadno. V případě otočení se jedná o jednoduchou práci s úhly. Totéž lze říci i o posunutí želvy ve směru její orientace, neboť její novou pozici stanovíme pomocí vzorce

$$(x', y') = d \cdot (\cos \varphi, \sin \varphi),$$

kde φ je úhel otočení želvy. Navíc v závislosti na tom, zda je pero položeno na plátně, buď s pohybem želvy současně vytvoříme novou úsečku, nebo pouze posuneme želvu. Pseudokód si lze prohlédnout v ukázce 5.2.5 a výslednou implementaci společně s přepočítáváním minimálních a maximálních souřadnic v programu 5.2.6.

Algoritmus 5.2.5: Posunutí želvy ve směru a nakreslení úsečky

Vstup: Pozice želvy (x, y) , úhel φ , délka kroku d , seznam úseček L

- 1 $x' \leftarrow d \cos(\varphi), y' \leftarrow d \sin(\varphi);$
- 2 **if** je pero položeno na plátně **then**
- 3 | Do seznamu L přidej úsečku $\ell = (x, x', y, y');$
- 4 **return** $(x', y', \ell);$

Výstup: Nová pozice želvy s nakreslenou úsečkou (x', y', ℓ)

```
1 def forward(self) -> None:
2     prev = self._position
3     self._position += self._step * Vector(math.cos(self._angle),
4                                         math.sin(self._angle))
5     if self._x_min > self._position.x: self._x_min = self._position.x
6     if self._y_min > self._position.y: self._y_min = self._position.y
7     if self._x_max < self._position.x: self._x_max = self._position.x
8     if self._y_max < self._position.y: self._y_max = self._position.y
9
10    if self._pen_down:
11        self._lines.append([prev, self._position])
```

Program 5.2.6: Implementace kroku želvy

5.2.3 Zásobník

S želví grafikou jsme kromě základní čtveřice symbolů zavedli i dvojici symbolů [a], které složily pro uložení aktuálního stavu želvy na vrchol zásobníku.

Implementace zásobníku je typicky jednou ze základních programovacích úloh a nejspíše není nikterak složité si jeho implementaci rozmyslet. Pro její vzorovou ukázku viz 5.2.7.

```
1 class Stack:
2     def __init__(self, items: list = []) -> None:
3         self._items = items
4
5     def push(self, item: object) -> None:
6         self._items.insert(0, item)
7
8     def pop(self) -> object:
9         item = self._items[0]
10        del self._items[0]
11        return item
```

Program 5.2.7: Implementace zásobníku

5.2.4 Vykreslení obrazce pomocí knihovny `tkinter`

Záverečná část implementace želví grafiky spočívá již pouze v samotné implementaci řetězce, který jsme vygenerovali pomocí našeho L-systému (viz podsekce 5.2.1). Začneme však záležitotí, která se spíše týká estetické stránky, a to manipulací s výsledným obrazcem.

Existují různé knihovny pro práci s grafickým rozhraním v jazyce Python. My si pro jednoduchost zvolíme knihovnu `tkinter`. S prominutím zde nebudeme rozebírat její kompletní obsah⁴, využijeme pouze některé funkce. Uvedeme si však alespoň nutný základ, který budeme potřebovat.

Základem je vytvoření grafického okna, do něhož budeme vykreslovat výsledné obrazce. K tomu je potřeba zavolat konstruktor `Tk()`. Dále pomocí metod `geometry` a `title` lze nastavit jeho šířku a titulek.

Do samotného okna však ještě nelze nic vykreslovat. K tomu je potřeba vytvořit v okně plátno (anglicky *canvas*) pomocí třídy `Canvas`. Do okna jej následně vložíme zavoláním metody `pack()` (viz ukázka 5.2.8).

Okno nakonec spustíme pomocí metody `mainloop()`, jejíž volání skočí až po jeho uzavření.

Knihovna `tkinter` nabízí mnoho dalších funkcí a toto nebyl ani zdaleka základ v pravém slova smyslu, nicméně bude to pro naše potřeby dostačující.

Problém, s nímž se však nyní potřebujeme vypořádat, je zarovnání obrázku na

⁴Pokud by čtenáře zajímal dokumentace, lze ji nalézt např. zde: <https://docs.python.org/3/library/tkinter.html#module-tkinter>.

```

1 import tkinter
2
3 window = tkinter.Tk()
4 window.geometry("1280x720")
5 window.title("Ukazkové okno")
6
7 canvas=tkinter.Canvas(window, width=600, height=800)
8 canvas.pack()
9
10 window.mainloop()

```

Program 5.2.8: Základní práce s knihovnou `tkinter`

střed grafického okna, neboť počátek souřadnic je umístěn v levém horním rohu plátna, tedy celé plátno tvoří její **první kvadrant**. To však znamená, že např. bod (x, y) , kde např. $x < 0$, nebude viditelný. Pro tento účel jsme si však uchovávali minimální a maximální hodnoty obou souřadnic. Díky nim můžeme následně provést posunutí celého obrazce do požadované polohy.

Střed výsledného (zatím nevykresleného) obrazce dokážeme jednoduše spočítat. Známe-li minimální, resp. maximální hodnoty obou souřadnic, které označíme $x_{\min}, x_{\max}, y_{\min}$ a y_{\max} , pak střed (x_c, y_c) určíme jako

$$(x_c, y_c) = \left(\frac{x_{\max} + x_{\min}}{2}, \frac{y_{\max} + y_{\min}}{2} \right).$$

Pokud budeme tedy chtít posunout střed obrazce obecně do bodu (x, y) , pak vektor posunutí bude jednoduše $\vec{v} = (x, y) - (x_c, y_c)$. Každou z úseček, kterou jsme vygenerovali pomocí želví grafiky, stačí posunout o spočítaný vektor \vec{v} (viz algoritmus 5.2.9). Implementaci si lze opět prohlédnout v ukázce programu 5.2.10.

Nyní jsme již zcela připraveni provést závěrečné vykreslení obrazce. Stačí pro-

Algoritmus 5.2.9: Algoritmus pro posun obrazce

Vstup: Seznam úseček L , nový střed (x, y)

- 1 $(x_c, y_c) \leftarrow ((x_{\max} + x_{\min})/2, (y_{\max} + y_{\min})/2);$
- 2 $\vec{v} \leftarrow (x, y) - (x_c, y_c);$
- 3 $L' \leftarrow \emptyset;$
- 4 **foreach** úsečka $\ell = (x_0, y_0, x_1, y_1)$ v seznamu L **do**
- 5 $(x'_0, y'_0) \leftarrow (x_0, y_0) + \vec{v};$
- 6 $(x'_1, y'_1) \leftarrow (x_1, y_1) + \vec{v};$
- 7 Do L' přidej úsečku $\ell' = (x'_0, y'_0, x'_1, y'_1);$
- 8 **return** $L';$

Výstup: Posunuté úsečky L'

cházet vygenerovaný řetězec znak po znaku a simulovat pohyb želvy, která nám bude do příslušného seznamu ukládat postupně vygenerované úsečky. Následně náš obrazec zarovnáme na střed okna a každou z úseček vykreslíme na plátno (viz algoritmus 5.2.11 a ukázka implementace 5.2.12 společně s vykreslením obrazce).

```

1 def center_to(self, xc: float, yc: float) -> None:
2     lines_center = Vector((self._x_min + self._x_max) // 2, (
3         self._y_min + self._y_max) // 2)
4     translation_vector = Vector(xc, yc) - lines_center
5
6     for line in self._lines:
7         line[0] += translation_vector
8         line[1] += translation_vector

```

Program 5.2.10: Posunutí středu obrazce do zvoleného bodu

Algoritmus 5.2.11: Simulace pohybu želvy

Vstup: Vygenerované slovo w , pozice želvy (x, y) , úhel otočení φ , inkrementace úhlu δ , délka kroku d

```

1 Založ prázdný zásobník  $Z$ ;
2  $L \leftarrow \emptyset$ ;
3 for  $i = 1, 2, \dots, |w|$  do
4   if  $w[i] = +$  then  $\varphi \leftarrow \varphi + \delta$ ;
5   if  $w[i] = -$  then  $\varphi \leftarrow \varphi - \delta$ ;
6   if  $w[i] = f$  then
7      $(x, y) \leftarrow (x + d \cos \delta, y + d \sin \delta)$ ;
8   if  $w[i] = [$  then
9     Na vrchol zásobníku  $Z$  ulož stav želvy  $(x, y, \varphi)$ ;
10  if  $w[i] = ]$  then
11    Z vrcholu zásobníku  $Z$  odstraň stav  $(x_t, y_t, \varphi_t)$ ;
12     $(x, y) \leftarrow (x_t, y_t)$ ;
13     $\varphi \leftarrow \varphi_t$ ;
14  else
15     $(x', y') \leftarrow (x + d \cos \varphi, y + d \sin \varphi)$ ;
16    Do seznamu  $L$  přidej úsečku  $(x, y, x', y')$ ;
17     $(x, y) \leftarrow (x', y')$ ;
18 return  $(x, y, \varphi, L)$ ;

```

Výstup: Stav želvy a seznam vzniklých úseček (x, y, φ, L)

5.3 Implementace IFS

Systémy iterovaných funkcí a k nim související teorii jsme si vyložili již v části 4.2. Podobně jako v případě L-systémů, i zde budeme postupovat přímo z definice. Konkrétně jsme si definovali IFS jako množinu kontrakcí

$$\{\psi_i : X \rightarrow X \mid 1 \leq i \leq n\},$$

přičemž jsme následně zkoumali a pracovali se zobrazením Ψ daným předpisem

$$\Psi(B) = \bigcup_{i=1}^n \psi_i(B), \quad B \in \mathbb{H}(X).$$

```

1 lsystem = LSystem(axiom, rules)
2 lsystem.iterate(iteration_count)
3
4 # Simulate turtle movement
5 stack = Stack()
6 for char in lsystem.word:
7     if char == '+':
8         turtle.rotate(angle)
9     elif char == '-':
10        turtle.rotate(-angle)
11    elif char == 'f':
12        turtle.pen_down = False
13        turtle.forward()
14    elif char == '[':
15        stack.push((turtle.position, turtle.angle))
16    elif char == ']':
17        state = stack.pop()
18        turtle.position = state[0]
19        turtle.angle = state[1]
20    else:
21        turtle.pen_down = True
22        turtle.forward()
23
24 # Center the figure
25 turtle.center_to(window_width // 2, window_height // 2)
26
27 # Draw figure
28 for line in turtle.lines:
29     canvas.create_line(line[0].x, line[0].y, line[1].x, line
30 [1].y)

```

Program 5.2.12: Implementace algoritmu 5.2.11 s vykreslením

O zobrazení Ψ jsme následně dokázali, že se též jedná o kontrakci (viz věta 4.2.6).

Kontrakce, s nimiž jsme pracovali, byly tzv. *affinní zobrazení* v \mathbb{R}^2 , tedy jejich předpis byl ve tvaru

$$f(x, y) = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} ax + by + e \\ cx + dy + f \end{pmatrix}. \quad (5.1)$$

V konečném důsledku si tedy stačilo uchovat pouze koeficienty a, b, \dots, f . A toho přesně využijeme i zde. Daná affinní zobrazení budeme reprezentovat seznamem uspořádaných šestic

$$(a, b, c, d, e, f).$$

Dále potřebujeme již pouze znát počáteční obrazec. Ačkoliv bychom mohli jistě různými sofistikovanými způsoby reprezentovat celou řadu množin, my se omezíme pouze na mnohoúhelníky⁵, neboť jejich reprezentace je velmi jednoduchá. Stačí si pamatovat pozice jejich vrcholů

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n).$$

I zde provedeme implementaci IFS pomocí třídy (viz ukázka 5.3.1). Po vzoru

⁵V konečném důsledku, jak již víme z Banachovy věty 4.2.9, počáteční obrazec nehraje žádnou roli pro atraktor zobrazení Ψ .

```

1  from copy import deepcopy
2
3  class IFS:
4
5      def __init__(self, starting_figure: list, tr_coefs: list = []
6          []) -> None:
7          self._figures = [starting_figure]
8          self._total_iterations = 0
9
10         # Min/max coordinates (used for centering)
11         self._x_min, self._y_min, self._x_max, self._y_max =
12             0, 0, 0, 0
13         self.__update_min_max_coords()
14
15         self._transformations = set()
16         for tpl in tr_coefs:
17             def transformation(point, tpl=deepcopy(tpl)):
18                 return Vector(
19                     tpl[0]*point.x + tpl[1]*point.y + tpl[4],
20                     tpl[2]*point.x + tpl[3]*point.y + tpl[5]
21                 )
22             self._transformations.add(transformation)
23
24     def __update_min_max_coords(self) -> None:
25         self._x_min = min(point.x for figure in self._figures
26         for point in figure)
27         self._y_min = min(point.y for figure in self._figures
28         for point in figure)
29         self._x_max = max(point.x for figure in self._figures
30         for point in figure)
31         self._y_max = max(point.y for figure in self._figures
32         for point in figure)

```

Program 5.3.1: Konstruktor pro třídu IFS

třídy `Turtle`, kterou jsme si ukázali v minulé sekci 5.2, i zde si budeme průběžně aktualizovat minimální a maximální hodnoty souřadnic pro pozdější manipulaci s obrazcem. Dále zde máme dvojici důležitých atributů:

- `self._figures` uchovává všechny vygenerované obrazce po obecně k -té iteraci jako seznam usporádaných n -tic vrcholů.
- `self._transformations` ukládá zadané kontrakce $\psi_1, \psi_2, \dots, \psi_m$ jako first-class funkce. Výpočet probíhá podle (5.1).

Co je to *first-class funkce*⁶? Jedná se o koncept práce s funkcemi jakožto standardními objekty, na které se lze odkazovat. V praxi to znamená možnost předávat funkce jako parametry, používat je jako návratové hodnoty, nebo ukládat je do proměnných. To se nám v tomto případě velmi hodí, neboť tyto funkce potřebujeme vytvářet až za běhu programu v závislosti na zadaných koeficientech affiních zobrazení.

⁶Též se lze někdy setkat s českým označením funkce nebo obecněji objekty *první kategorie*.

V této části se zaměříme pouze na generování výsledného obrazce v jednotlivých iteracích. V tomto ohledu bude potřeba si uchovávat nově vygenerované útvary do nějaké struktury. To vše je shrnuto v algoritmu 5.3.2. Speciálně, pokud bychom

Algoritmus 5.3.2: k -tá iterace IFS

Vstup: IFS $\{\psi_1, \dots, \psi_n\}$, množina útvarů \mathcal{F} , číslo $k \in \mathbb{N}$

- 1 **for** $i = 1, 2, \dots, k$ **do**
- 2 $\mathcal{F}' \leftarrow \emptyset;$
- 3 **foreach** $F \in \mathcal{F}$ **do**
- 4 **foreach** $\psi \in \{\psi_1, \dots, \psi_n\}$ **do**
- 5 $\mathcal{F}' \leftarrow \mathcal{F}' \cup \{\psi(F)\};$
- 6
- 7 $\mathcal{F} \leftarrow \mathcal{F}';$
- 8 **return** $\mathcal{F};$

Výstup: Nová množina útvarů F

chtěli vygenerovat k -tou z počátečního obrazce F_0 , stačí algoritmus zavolat na množinu $\{F_0\}$. Praktickou implementaci si lze prohlédnout v ukázce 5.3.3. Posu-

```

1  def iterate(self, iterations: int) -> None:
2      for _ in range(iterations):
3          figures_new = []
4
5          for figure in self._figures:
6              for tr in self._transformations:
7                  figure_new = []
8                  for point in figure: figure_new.append(tr(
9                      point))
10
11                 figures_new.append(figure_new)
12
self._figures = figures_new

```

Program 5.3.3: Implementace algoritmu 5.3.2 ve třídě IFS

nutí obrazce na střed si již může čtenář samostatně rozmyslet. Provedení by však bylo obdobné jako v případě L-systémů.

5.4 Implementace Time Escape algoritmů

Poslední kategorie fraktálních útvarů tvorily tzv. *Juliovy množiny*, u nichž jsme si jednoduše vysvětlili, že jejich generování probíhá pomocí tzv. *Time Escape algoritmů*. Jejich princip lze nastínit následovně: na vstupu zadáme nějakou komplexní polynomiální funkci f a dále čísla $m \in \mathbb{N}_0$ a $r \in \mathbb{R}$. Číslo m bude sloužit jako horní hranice počtu iterací, který pro každý bod v určité omezené oblasti komplexní roviny provedeme (to lze pochopitelně dospecifikovat, avšak teď to není úplně podstatné). Pokud v kterékoli iteraci nastane, že $|f^{\circ k}(z)| > r$, pak bod vyloučíme ze zkoumané množiny. Naopak v případě, že pro každé $0 \leq k \leq m$ je $|f^{\circ k}(z)| \leq r$, pak prohlásíme, že bod naleží Juliově množině.

5.4.1 Aproximace vyplněné Juliovy množiny

Podívejme se na tento algoritmus trochu blíže v pseudokódu 5.4.1. Nejspíše ní-

Algoritmus 5.4.1: Generování vyplněné Juliovy množiny při pevném počtu iterací

Vstup: Komplexní polynomiální funkce f , maximální počet iterací $m \in \mathbb{N}_0$, číslo $r \in \mathbb{R}$, konečné množiny $X \subset \langle x_{\min}, x_{\max} \rangle$ a $Y \subset \langle y_{\min}, y_{\max} \rangle$

```

1  $K \leftarrow \emptyset;$ 
2 foreach  $(a, b) \in X \times Y$  do
3    $z \leftarrow a + bi;$ 
4    $t \leftarrow id;$ 
5   for  $k = 0, 1, \dots, m$  do
6      $t \leftarrow t \circ f;$ 
7     if  $|t(z)| > r$  then
8       pokračuj další iterací vnějšího cyklu;
9    $K \leftarrow K \cup \{z\};$ 
10 return  $K;$ 

```

Výstup: Aproximace vyplněné Juliovy množiny $K(f)$

koho nepřekvapí, že při vyšších hodnotách čísla m obdržíme lepší odhad Juliovy množiny příslušné polynomiální funkci f . Avšak vždy je potřeba zvážit náročnost výpočtu.

Znázornění (vyplněné) Juliovy množiny lze provést různými způsoby. Prezentovaný algoritmus 5.4.1 pouze určuje pro každý zvolený bod z , zda naleží, či nenaleží množině J . Avšak čtenář zajímající se o tuto partii matematiky již nejspíše viděl poměrně známý způsob vyobrazení těchto množin s barevným rozlišováním bodů. Tuto záležitost jsme již zmínili ke konci části 4.3.1, avšak jeho podstatné aspekty jsou především algoritmické povahy a tedy teprve v této kapitole je více rozvedeme. K tomu se však dostaneme později.

Praktická implementace Time Escape algoritmů bude podstatně složitější, neboť se musíme vypořádat s konverzí samotného polynomu a rovněž vyřešit způsob vzorkování vybrané části komplexní roviny. Dále se budeme držet realizace pomocí třídy (viz ukázka 5.4.2). Pojdme si konstruktor 5.4.2 opět rozebrat.

- `self._x_count` a `self._y_count` udávají počet bodů, které budeme zkoumat, ve směru reálné a imaginární osy. Jejich hodnoty jsou závislé na velikosti kroku `step`, kterou konstruktor přijímá jako parametr. Tedy např. s krokem 1 v rámci intervalu $\langle -1, 2 \rangle$ budeme zkoumat celkem 4 body.
- V seznamu `self._iter_counts` si budeme pro každý bod uchovávat, kolik iterací zadání funkce bylo potřeba, než hodnoty posloupnosti překročily zadanou mez (tedy $|f^{ok}(z)| > r$). Ten využijeme především později při určování barev jednotlivých bodů.
- `self._sequence` uchovává předpis polynomiální funkce, kterou budeme iterovat, jako řetězec. Předpisy budeme zadávat standardní syntaxí v Pythonu,

```

1  class TEA:
2      def __init__(self, width: int, height: int, sequence: str,
3          step: int = 1, escape_radius: int = 2, bounds: tuple, var
4          : str, explore_var: str):
5              self._x_count, self._y_count = width // step, height
6              // step
7
8              self._iter_counts = [[0 for _ in range(self._x_count)]]
9              for _ in range(self._y_count):
10                 self._width, self._height = width, height
11                 self._sequence = sequence
12                 self._var = var
13                 self._explore_var = explore_var
14                 self._total_iterations = 0
15                 self._escape_radius = escape_radius
16
17                 x_min, x_max, y_min, y_max = bounds
18
19                 x_vals = [x_min + step * (x_max - x_min) * j / width
20                 for j in range(self._x_count + 1)]
21                 y_vals = [y_min + step * (y_max - y_min) * i / height
22                 for i in range(self._y_count + 1)]
23                 self._complex_grid = [[x + 1j * y for x in x_vals] for
24                 y in y_vals]
25
26                 self._point_last_values = [[0 for _ in range(self.
27                     _x_count)] for _ in range(self._y_count)]

```

Program 5.4.2: Konstruktor třídy TEA

tzn. např. pro Mandelbrotovu množinu, kde $f_c(z) = z^2 + c$, bychom předpis napsali

$$z^{**2} + c.$$

Levou stranu „ $f(z)$ “ pochopitelně psát netřeba.

- `self._var` udává, která proměnná v `self._sequence` je argumentem zadané funkce (typicky z).
- `self._explore_var` uchovává proměnnou, za níž budeme dosazovat hodnoty zkoumaných bodů. Pro Juliovu množinu se typicky jedná přímo o argument zadané funkce f , ale např. pro Mandelbrotovu množinu je to c .
- `self._escape_radius` reprezentuje číslo r , tedy hranici absolutní hodnoty čísla z , po jejímž překročení prohlásíme posloupnost iterací za divergentní.
- Parametr `bounds` specifikuje část komplexní roviny, z níž budeme zkoumat vybrané body. Jedná se o datový typ `tuple`, v našem případě uspořádanou čtveřici $(x_{\min}, x_{\max}, y_{\min}, y_{\max})$.
- `self._complex_grid` uchovává všechny body ze zadané části komplexní roviny jako komplexní čísla, tedy v Pythonu datový typ `complex`.

- Do seznamu `self._point_last_values` budeme ukládat hodnoty $f^{ok}(z)$ pro zadané z , kde k je číslo poslední prozkoumané iterace (tzn. buď jsme u bodu prozkoumali maximální počet zadaných iterací, nebo výpočet skončil dříve kvůli překročení povolené absolutní hodnoty).

Je vidět, že atributů zde máme poměrně hodně. Zkusme si tedy nejdříve vyjasnit, jak bychom mohli pomocí těchto informací implementovat algoritmus pro iterování zadané polynomiální funkce. Již jsme si uvedli asi nejjednodušší možnost v rámci algoritmu 5.4.1. Jak by se ale situace změnila, když si budeme uchovávat počty iterací, které proběhly, než jsme překročili zadanou absolutní hodnotu r nebo dosáhly maximálního počtu m ? Označme si takové pole např. T . Podívejme se na algoritmus 5.4.3. Pro implementaci algoritmu 5.4.3 si však budeme muset

Algoritmus 5.4.3: Generování vyplněné Juliové množiny pomocí pole iterací T

Vstup: Komplexní polynomiální funkce f , maximální počet iterací $m \in \mathbb{N}_0$, číslo $r \in \mathbb{R}$, konečné množiny $X \subset \langle x_{\min}, x_{\max} \rangle$ a $Y \subset \langle y_{\min}, y_{\max} \rangle$

- 1 $K \leftarrow \emptyset;$
- 2 Vytvoř prázdné pole T ;
- 3 **foreach** $(x, y) \in X \times Y$ **do**
- 4 $z \leftarrow x + yi$;
- 5 $t \leftarrow \text{id}$;
- 6 **for** $k = 0, 1, \dots, m$ **do**
- 7 $t \leftarrow t \circ f$;
- 8 $T[x, y] \leftarrow k$;
- 9 **if** $|t(z)| > r$ **then** opust cyklus;
- 10 **if** $T[x, y] = m$ **then** $K \leftarrow K \cup \{z\}$;
- 11 **return** K ;

Výstup: Aproximace vyplněné Juliové množiny K

rozmyslet, jak budeme pomocí řetězce s předpisem pro funkci f (tj. atributu `self._sequence`) počítat její funkční hodnoty. Jistě se nabízí možnost vytvořit funkci pro vyhodnocení obecného matematického výrazu. Ač se jedná o poměrně hezké programovací cvičení a čtenář si jej může vyzkoušet, my si poradíme trochu jinak – funkcí `eval`. Funkce `eval` jednoduše vyhodnotí zadáný výraz a za proměnné dosadí hodnoty podle slovníku poskytnutého v příslušném parametru či parametrech.

Tím se nám situace podstatně zlehčuje. Implementaci si může čtenář prohlédnout v ukázce 5.4.4.

5.4.2 Aproximace Juliové množiny

S množinou $K(f)$ se rovněž pojí i její hranice $\partial K(f)$. Co kdybychom chtěli vykreslit pouze hraniční body? Otázka se příliš nevzdaluje té původní, akorát je navíc

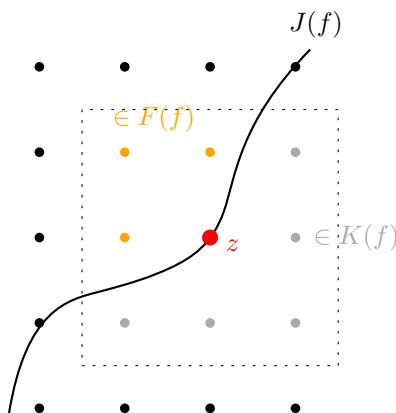
```

1 def iterate(self, iterations: int) -> None:
2     for i in range(self._y_count):
3         for j in range(self._x_count):
4             # Initialize variables
5             vars_dict = {self._var: 0, self._explore_var: self
6             ._complex_grid[i][j]}
7
8             # Iterate
9             for k in range(1, iterations + 1):
10                 try:
11                     # Evaluate the next value in the sequence
12                     vars_dict[self._var] = eval(self._sequence
13                     , {"math": math}, vars_dict)
14
15                     # Check for escape condition
16                     self._iter_counts[i][j] = k
17                     if abs(vars_dict[self._var]) > self.
18                         _escape_radius:
19                         break
20                 except OverflowError:
21                     self._iter_counts[i][j] = k
22                     break
23
24             self.point_last_values[i][j] = vars_dict[self._var]
25

```

Program 5.4.4: Implementace algoritmu 5.4.3

potřeba o každém bodu $K(f)$ rozhodnout, zda je hraniční, či nikoliv⁷. Přitom myšlenka řešení se vůbec nemusí příliš vzdalovat matematickému pojedání hranice. Tu jsme si definovali pro libovolnou množinu M jako množinu takových bodů, které ve svém libovolně malém okolí obsahují alespoň jeden bod, který M náleží a alespoň jeden, který jí nenáleží. Toho bychom však mohli využít. Stačí zkонтrolovat všechny sousední body pro každé $z \in K(f)$, zda alespoň jeden z nich nenáleží $F(f)$. Pokud ano, pak z prohlásíme za hraniční bod (viz obrázek 5.1).



Obrázek 5.1: Ilustrace hraničního bodu

⁷Všimněte si, že zde se nám náramně hodí výsledek z věty 4.3.5, specificky bod (ii), kde jsme dokázali, že $J(f) \subseteq K(f)$. Náš návrh je tedy v tomto ohledu oprávněný.

Na základě této ideje si lze již poměrně snadno domyslet postup při generování hranice. Pracujeme opět s předpokladem, že oblast komplexní roviny představuje diskrétní množinu bodů, které tvoří mřížku (tedy každá dvojice sousedních bodů má stejnou vzdálenost).

Algoritmus 5.4.5: Generování Juliových množin J

Vstup: Komplexní polynomiální funkce f , konečné množiny
 $X \subset \langle x_{\min}, x_{\max} \rangle$ a $Y \subset \langle y_{\min}, y_{\max} \rangle$

- 1 $J \leftarrow \emptyset;$
- 2 **foreach** $(x, y) \in X \times Y$ **do**
- 3 $z \leftarrow x + yi;$
- 4 **if** existují sousední body $w \in K(f)$ a $w' \in F(f)$ bodu z **then**
- 5 $J \leftarrow J \cup \{z\};$
- 6 **return** $J;$

Výstup: Aproximace Juliových množin J

```

1  h_px = len(iter_counts)
2  w_px = len(iter_counts[0])
3
4  # List od inner points
5  inside = [
6      [iter_counts[y][x] == max_iterations for x in range(w_px)]
7      for y in range(h_px)
8 ]
9
10 # Determine boundary points
11 boundary_mask = [[False]*w_px for _ in range(h_px)]
12 for y in range(h_px):
13     for x in range(w_px):
14         if inside[y][x]:
15             for dx, dy in ((1,0), (-1,0), (0,1), (0,-1)):
16                 nx, ny = x+dx, y+dy
17                 if 0 <= nx < w_px and 0 <= ny < h_px:
18                     if not inside[ny][nx]:
19                         boundary_mask[y][x] = True
20                         break

```

Program 5.4.6: Implementace algoritmu 5.4.5

5.4.3 Přiřazování barev

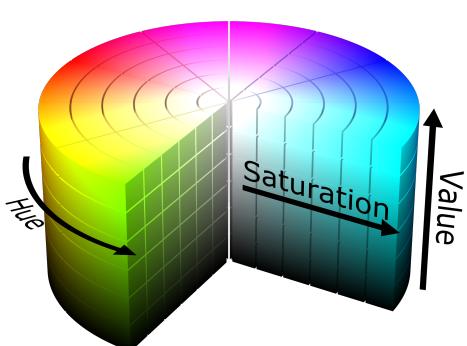
Jako poslední nás čekají barvy a jejich přiřazování jednotlivým bodům. Již jsme měli možnost vidět několik příkladů obrázků s fraktály, v nichž byly body barevně zvýrazněny. Zevrubně bychom mohli říci, že čím blíž se bod necházel hranici útvaru, tím výraznější byla jeho barva.

Stručně k barevnému modelu HSV

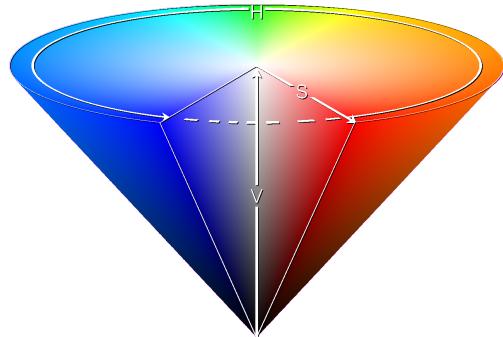
Ukážeme si zde dvojici základních možností, jak lze přiřazovat barvy daným bodům. K tomu však budeme potřebovat pracovat s barevným modelem *HSV* (*Hue*, *Saturation*, *Value*). S prominutím si zde opět odustíme delší vysvětlování a budeme předpokládat, že se čtenář již s modelem HSV někdy setkal (podobně jako např. s RGB nebo s CMYK používaným u tiskáren). Avšak základem, jak název napovídá, je trojice následujících složek:

- **hue** (česky *odstín*),
- **saturation** (česky *saturace*, nebo též *sytost*)
- a **value** reprezentující hodnotu jasu (tj. podílu bílé barvy).

Celkově tento model vychází přímo z vnímání barev lidským okem. Různé možnosti znázornění HSV modelu si lze prohlédnout na obrázcích 5.2a a 5.2b.



(a) Válcová reprezentace



(b) Kuželová reprezentace

Obrázek 5.2: Grafické znázornění HSV modelu (Převzato z Wikipedia Commons, viz *odkaz*)

Existují pochopitelně metody pro převody mezi jednotlivými barevnými modely, ty však pro nás zde nejsou relevantní. Jednotlivé barvy budeme reprezentovat jako uspořádané trojice (H, S, V) , kde $0 \leq H, S, V \leq 1$.

Lineární interpolace barev

Patrně nejjednodušším způsobem pro přiřazení barev jednotlivým bodům je tzv. *lineární interpolaci* odstínu při pevně zvolené saturaci a jasu. Obecně jsou-li zadány body v rovině (x_0, y_0) a (x_1, y_1) , pak lineární interpolace přiřazuje každému $x \in (x_0, x_1)$ souřadnici $y = f(x)$ takovou, že (x, y) leží na spojnici bodů (x_0, y_0) a (x_1, y_1) . Toto lze vyjádřit poměrně jednoduchým vzorcem:

$$f(x) = y_0 + (x - x_0) \frac{y_1 - y_0}{x_1 - x_0}.$$

Pojďme si nyní rozmyslet náš případ. Bodům z komplexní roviny budeme přiřazovat barvu podle počtu iterací, kterého jsme při výpočtu dosáhli, než začala

samočná postupnost absolutních hodnot iterací funkce divergovat (popř. oscilovat). Zde mohou tedy nastat celkově dva případy. Počet iterací pro pevně zvolený bod z si označme k a maximální počet iterací si označme m .

- Pokud $k < m$, pak přiřadíme bodu z barvu na základě lineární interpolace odstínu, přičemž saturaci a jas volíme pevně, označme S_0, V_0 .
- Pokud $k = m$, pak bodu z přiřadíme černou barvu, tedy prohlásíme jej za bod naležící zkoumané vyplněné Juliově množině.

Hodnoty odstínu H nabývají hodnot z intervalu $\langle H_{\min}, H_{\max} \rangle$, přičemž

$$0 \leq H_{\min}, H_{\max} \leq 1$$

a hodnoty k jsou z množiny $\{0, 1, 2, \dots, m\}$. Tedy celkový předpis pro lineární interpolaci odstínu bude

$$H = H_{\min} + k \cdot \frac{H_{\max} - H_{\min}}{m}$$

a tudíž pro výslednou barvu bodu z bude platit

$$(H, S, V) = \begin{cases} (H_{\min} + k(H_{\max} - H_{\min})/m, S_0, V_0) & k < m, \\ (0, 0, 0) & k = m. \end{cases}$$

Pro ukázku viz obrázky 5.3 a 5.4 s hranicí Mandebrotovy množiny.

Pochopitelně stejně jako odstín lze interpolovat i zbylé dvě složky. Není třeba se omezovat pouze nutně na odstín.

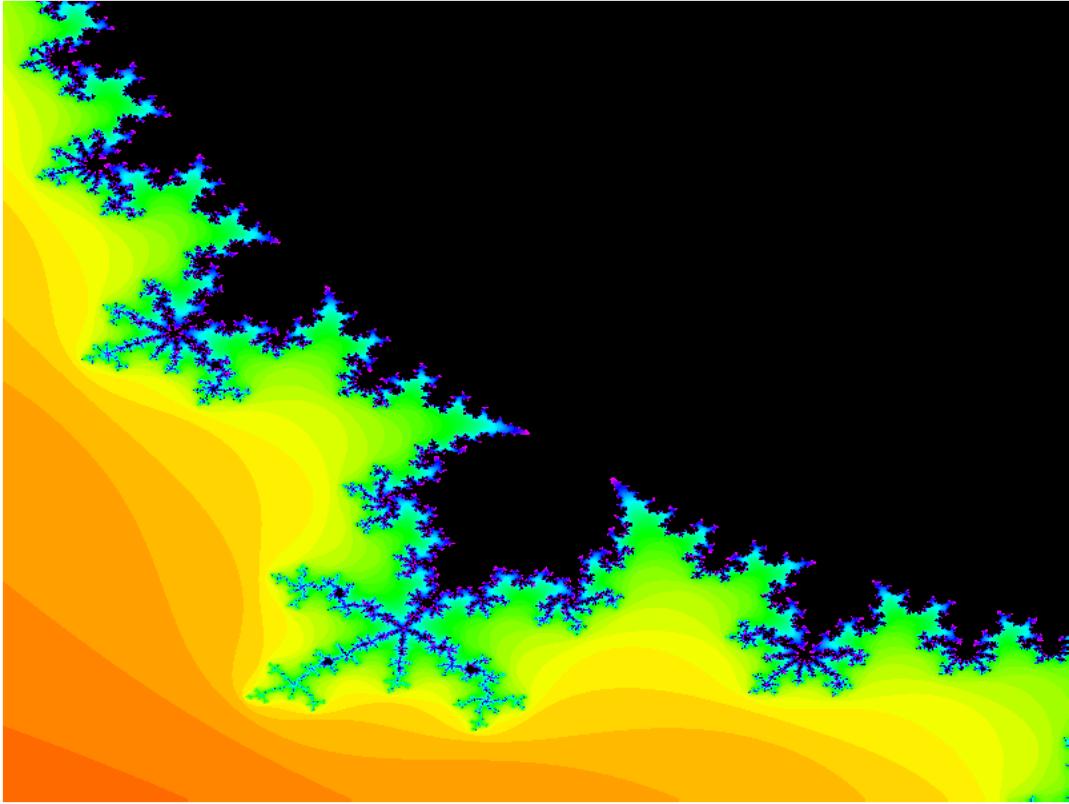
$$\begin{aligned} H &= H_{\min} + k \cdot \frac{H_{\max} - H_{\min}}{m}, \\ S &= S_{\min} + k \cdot \frac{S_{\max} - S_{\min}}{m}, \\ V &= V_{\min} + k \cdot \frac{V_{\max} - V_{\min}}{m}. \end{aligned}$$

Pokud bychom chtěli využít sofistikovanější barvení, mohli bychom též využít obecný interpolační polynom. Ten lze např. určit pomocí tzv. *Lagrangeovy interpolace*, kdy při známých funkčních hodnotách $f(x_0), f(x_1), \dots, f(x_n)$ pro navzájem různá x_0, x_1, \dots, x_n lze sestavit interpolační polynom $L_n(x)$ pomocí vzorce

$$L_n(x) = \sum_{i=0}^n f(x_i) \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{x - x_j}{x_i - x_j}.$$

Lze si všimnout, že takto definovaná funkce L_n prochází všemi zadanými body. Pro libovolné x_ℓ , kde $0 \leq \ell \leq n$, skutečně platí

$$\begin{aligned} L_n(x_\ell) &= \sum_{i=0}^n f(x_i) \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{x - x_j}{x_i - x_j} = \sum_{i=0}^n f(x_i) \frac{x_\ell - x_\ell}{x_\ell - x_j} \prod_{\substack{0 \leq j \leq n \\ j \neq i, j \neq \ell}} \frac{x_\ell - x_j}{x_i - x_j} \\ &= \sum_{i=0}^n f(x_i) \delta_{i\ell} = f(x_\ell), \end{aligned}$$



Obrázek 5.3: Mandebrotova množina pomocí lineární interpolace odstínu pro hodnoty $H_{\min} = 0$; $H_{\max} = 0,87$; $S_0 = V_0 = 1$; $m = 50$

kde δ_{ij} je Kroneckerovo delta⁸.

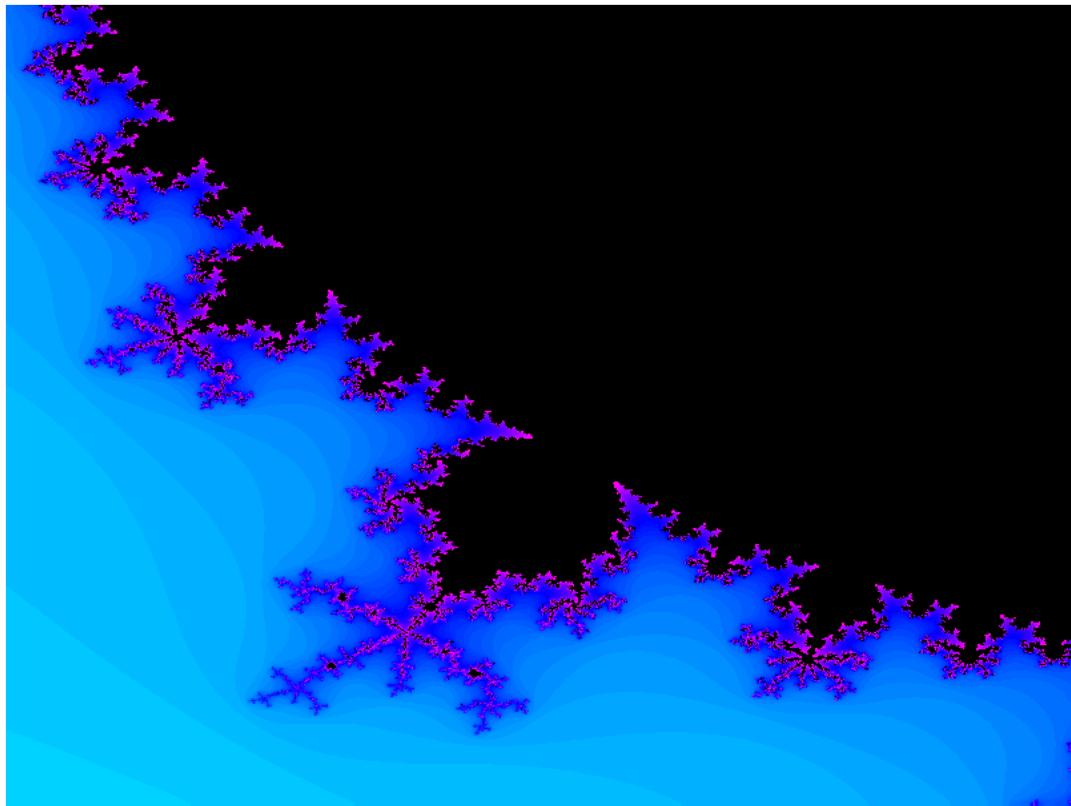
Hladké zbarvení

Ačkoliv bychom se s výsledkem pomocí prosté lineární interpolace (viz obrázky 5.3 a 5.4) mohli spokojit, lze si všimnout, že přechody mezi jednotlivými barvami (coby approximacemi vyplněné Juliovy množiny pro zvolenou polynomiální funkci) jsou velmi ostré. Je tomu tak z důvodu, že ze zvolené lineární interpolace vzniká konečná posloupnost hodnot H (funkce tedy není spojitá). Tento problém bychom mohli vyřešit zvýšením maximálního počtu iterací, tedy v konečném důsledku by mezi barvami nebyly takové rozdíly (pro porovnání s obrázkem 5.3 viz obrázek 5.5, kde $m = 100$). Nicméně zvýšením maximálního počtu iterací se podstatně zvyšuje náročnost výpočtu. Zkusme se tedy místo toho více zaměřit na způsob výpočtu výsledného odstínu. Často se při lineární interpolaci nepoužívá přímo počet iterací k , nýbrž číslo

$$k_s = k + \frac{\log(\log |f^{\circ k}(z)|)}{\log 2}.$$

⁸Kroneckerovo delta se definuje jako

$$\delta_{ij} = \begin{cases} 1 & i = j, \\ 0 & i \neq j. \end{cases}$$



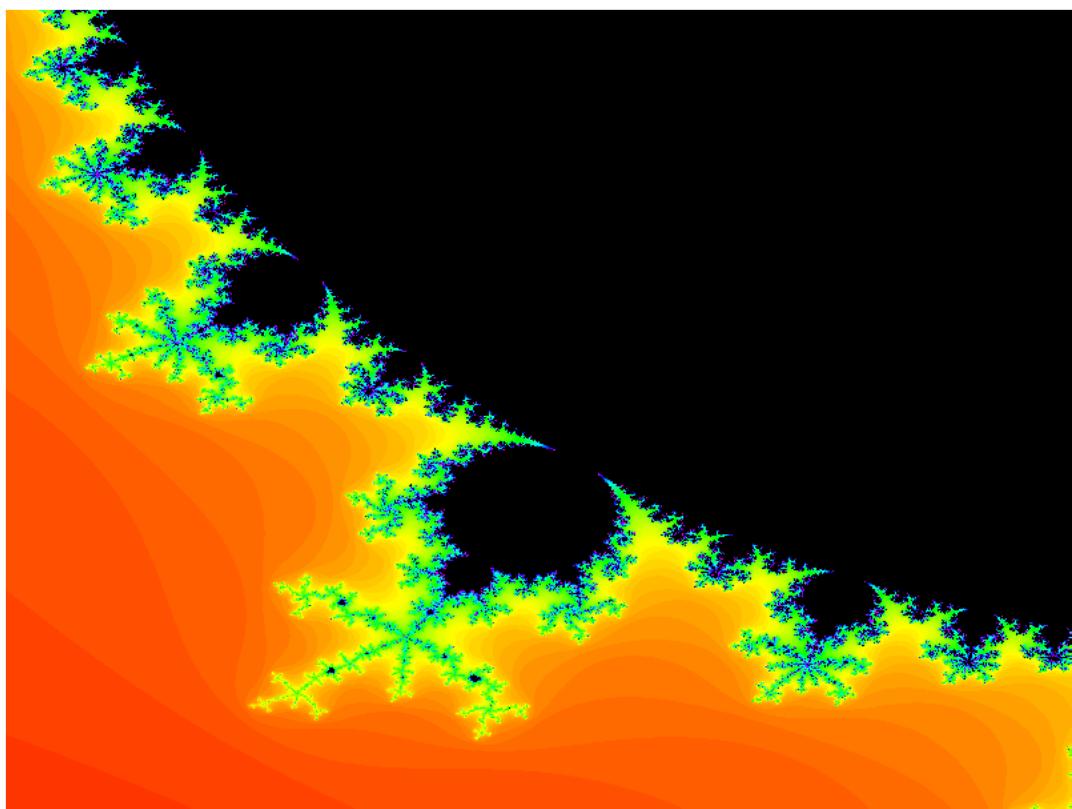
Obrázek 5.4: Mandebrotova množina pomocí lineární interpolace odstínu pro hodnoty $H_{\min} = 0,5$; $H_{\max} = 0,861$; $S_0 = 1$; $V_0 = 0,5$; $m = 50$

Tento vzorec si zde nebudeme odvozovat, avšak vychází z tzv. *potenciálové funkce* definované jako

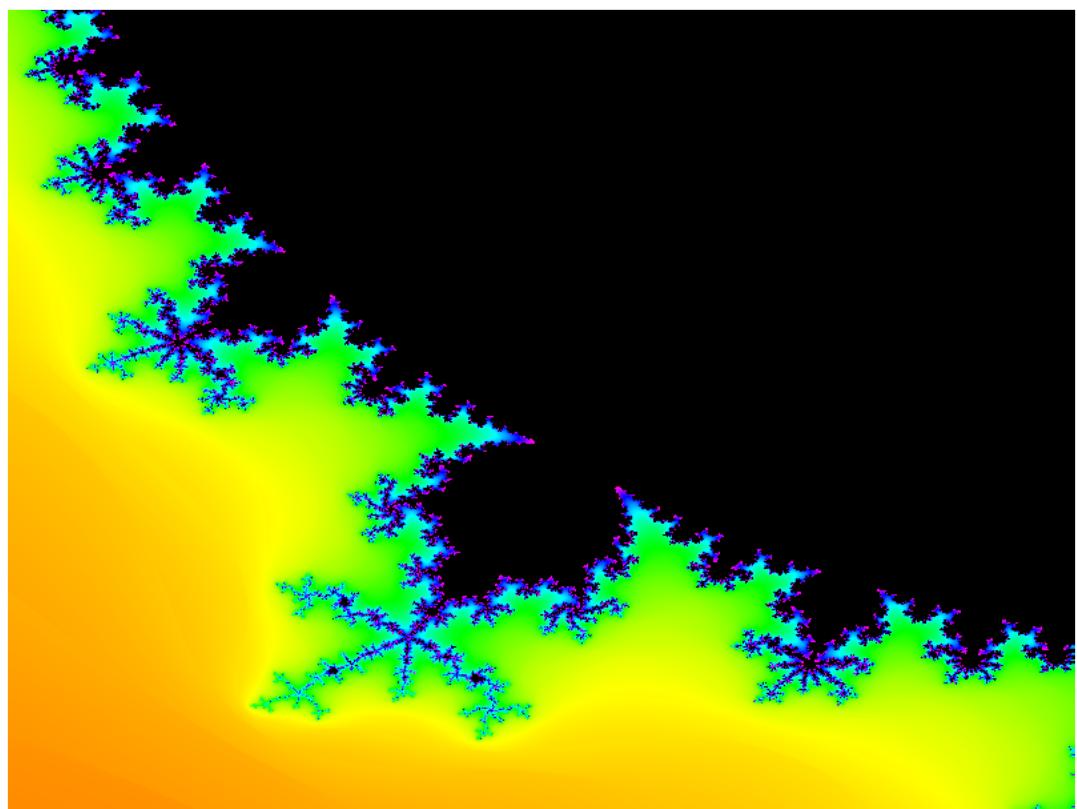
$$\varphi(z) = \lim_{k \rightarrow \infty} \frac{\log |f^{\circ k}(z)|}{d^k},$$

kde $d = \deg f$. V případě zájmu si čtenář může k této záležitosti vyhledat více informací.

Díky zohlednění absolutní hodnoty $|f^{\circ k}(z)|$ budou přechody mezi barvami již daleko jemnější. To je možno posoudit na obrázku 5.6.



Obrázek 5.5: Mandebrotova množina pomocí lineární interpolace odstínu pro hodnoty $H_{\min} = 0$; $H_{\max} = 0,87$; $S_0 = V_0 = 1$; $m = 100$



Obrázek 5.6: Mandebrotova množina pomocí hladkého zbarvení pro hodnoty $H_{\min} = 0$; $H_{\max} = 0,87$; $S_0 = V_0 = 1$; $m = 50$

Závěrem

Tento text celkově shrnuje některé základní poznatky z oblasti fraktální geometrie. Za zmínku stojí především koncepty soběpodobnosti a fraktální dimenze, základy teorie míry, L-systémy, systémy iterovaných funkcí (IFS) a Time Escape algoritmy, především se zaměřením na Juliovy množiny a jako poslední programová ukázka implementace některých algoritmů pro generování zmíněných typů fraktálů.

Text byl psán s cílem, aby se v něm orientovali jak čtenáři bez hlubšího matematického vzdělání – hlavně díky intuitivněji zaměřenému vysvětlení v kapitolách 1, 4 a 5, tak i pokročilejší zájemci, kteří ocení matematicky náročnejší obsah v kapitole 2 věnované teorii míry a dimenzím. V kapitole 5 byla vyvinuta snaha, co nejvíce vyjít vstříc širokému okruhu programátorů, a to důrazem na obecnost prezentovaných algoritmů. Tím je eliminována nutnost přímé znalosti jazyka Python, a čtenář si i tak může odnést něco užitečného pro vlastní projekty.

Velkou roli v práci hrají ilustrace, z nichž většina byla vytvořena ve vektorovém programu Ipe nebo pomocí programu na generování fraktálů, který je přiložen k práci. Lze jej nalézt na portálu GitHub⁹, kde je možné si přímo vyzkoušet generování fraktálních obrazců a seznámit se s praktickou implementací algoritmů uvedených v kapitole 5.

Text tedy poskytuje přehled hlavně teoretických aspektů fraktálů a jejich popisu. Je dobré podotknout, že matematická rozmanitost této problematiky je opravdu široká a většinu teorie tak ani nebylo možné zde ve smysluplné míře obsáhnout¹⁰. Tím spíš by se však slušelo přinejmenším zmínit její praktický aspekt. Fraktály se v našem světě vyskytují až překvapivě často a jsou všude okolo nás. Ukazuje se, že jejich praktické využití je velmi rozsáhlé. Například v oblasti kódování dat se fraktální metody uplatňují při efektivní kompresi obrázků, neboť dokáží jednoduchým způsobem popsat složité struktury. Ve sféře neuronových sítí se fraktální principy využívají jak v návrhu specifických architektur, tak při tvorbě optimalizačních algoritmů. V počítačové grafice jsou fraktály používány pro realistickou simulaci přírodních jevů, jako jsou například horské útvary, mraky nebo složité povrchy. V tomto ohledu si ještě naposledy dovolíme čtenáře odkázat na další literaturu, konkrétně knihu [22], která je přednostně věnována využití této oblasti geometrie.

Fraktály přitom nejsou pouze matematickými abstrakcemi – jejich struktury jsou hojně pozorovatelné i v přírodě, např. ve větvení stromů, ve struktuře rostlin, ve

⁹viz odkaz <https://github.com/D4vE0FF/Py-Fractal-Generator>

¹⁰Pro hlubší studium se čtenář může podívat do seznamu citované literatury

formacích horských hřebenů, v říčních sítích nebo v dynamice proudění kapalin. Tato všudypřítomnost fraktálních vzorů v přírodě potvrzuje hluboké propojení matematiky se světem kolem nás, což činí fraktální geometrii nejen atraktivní, ale také vysoce relevantní oblastí výzkumu a poznání.

Seznam použité literatury

- [1] BARNESLEY, M. F. (1993). *Fractals everywhere*. Academic Press Professional, Boston, 2nd ed. ISBN 0-12-079061-0.
- [2] EDGAR, G. (2008). *Measure, topology, and fractal geometry*. Springer, New York, 2nd ed. ISBN 978-0-387-74748-4.
- [3] ENGELKING, R. (1989). *General topology*. Heldermann, Berlin, rev. and completed ed. ISBN 3-88538-006-4.
- [4] FALCONER, K. J. (1989). Dimensions and measures of quasi self-similar sets. *Proceedings of the American Mathematical Society*, **106**(2). doi: 10.1090/S0002-9939-1989-0969315-8. URL <https://doi.org/10.1090/S0002-9939-1989-0969315-8>. MSC: Primary 58F12; Secondary 28A75.
- [5] FALCONER, K. J. (2014). *Fractal geometry*. John Wiley & Sons, Ltd, Chichester, 3rd edition. ISBN 978-1-119-94239-9.
- [6] HLADÍK, M. (2019). *Lineární algebra (nejen) pro informatiky*. Matfyzpress, Praha, vydání první. ISBN 978-80-7378-392-1.
- [7] LUKEŠ, J. a MALÝ, J. (2013). *Measure and integral*. Matfyzpress, Praha, 3rd ed. ISBN 978-80-7378-253-5.
- [8] MANDELBROT, B. B. (1983). *Fractal geometry of nature*. Freeman and company, New York, vyd. 3. ISBN 978-0716711865.
- [9] MATTILA, P. (1995). *Geometry of Sets and Measures in Euclidean Spaces*. Cambridge University Press, Cambridge, 1st ed. ISBN 0-521-46576-1.
- [10] MOTWANI, R., HOPCROFT, J. E. a ULLMAN, J. D. (2003). *Introduction to automata theory, languages, and computation*. Pearson Addison Wesley, Upper Saddle River, 2nd ed. ISBN 0321210298.
- [11] NETUKA, I. (2014). *Základy moderní analýzy*. Matfyzpress, Praha, 1. vyd. ISBN 978-80-7378-277-1.
- [12] NETUKA, I. (2016). *Integrální počet*. Matfyzpress, Praha, první vydání. ISBN 978-80-7378-334-1.
- [13] O'CONNOR, J. J. (Citováno 17. dubna 2025). Mactutor history of mathematics archive. [online]. Dostupné z: <https://mathshistory.st-andrews.ac.uk/Biographies/Hilbert/>.

- [14] O'CONNOR, J. J. (Citováno 17. dubna 2025). Mactutor history of mathematics archive. [online]. Dostupné z: <https://mathshistory.st-andrews.ac.uk/Biographies/Turing/>.
- [15] O'CONNOR, J. J. (Citováno 19. března 2025). Mactutor history of mathematics archive. [online]. Dostupné z: <https://mathshistory.st-andrews.ac.uk/Biographies/Hausdorff/>.
- [16] O'CONNOR, J. J. (Citováno 29. března 2025). Mactutor history of mathematics archive. [online]. Dostupné z: <https://mathshistory.st-andrews.ac.uk/Biographies/Borel/>.
- [17] O'CONNOR, J. J. (Citováno 4. března 2025). Mactutor history of mathematics archive. [online]. Dostupné z: <https://mathshistory.st-andrews.ac.uk/Biographies/Vitali/>.
- [18] PEITGEN, H.-O., JÜRGENS, H. a SAUPE, D. (2004). *Chaos and Fractals*. Springer, New York, 2nd edition. ISBN 978-1-4684-9396-2.
- [19] PRUSINKIEWICZ, P. a LINDENMAYER, A. (1990). *The Algorithmic Beauty of Plants*. Springer-Verlag, New York. ISBN 978-1-4613-8476-2.
- [20] ROYDEN, H. L. a FITZPATRICK, P. M. (2010). *Real analysis*. Prentice Hall, Boston, 4th ed. ISBN 978-0-13-511355-4.
- [21] VERNER, J. (Citováno 4. března 2025). Lebesgueovsky neměřitelné množiny. [online]. Dostupné z: <https://ktiml.mff.cuni.cz/~verner/download/nemeritelne.pdf>.
- [22] ZELINKA, I., VČELAŘ, F. a ČANDÍK, M. (2006). *Fraktální geometrie*. BEN - technická literatura, Praha, vydání 1. ISBN 80-7300-191-8.
- [23] ŠARMANOVÁ, P. (1996). *Malý průvodce historií integrálu*. Prometheus, Praha, 1. vyd. ISBN 80-7196-038-1.
- [24] ŠÁRKA VORÁČOVÁ, CSACHOVÁ, L., HÁJKOVÁ, V., HROMADOVÁ, J., MORAVCOVÁ, V., RICHTER, J., SURYNKOVÁ, P., ŠAROUNOVÁ, A., ŠAROUN, J., ŠRUBAŘ, J., ŠTAUBEROVÁ, Z. a TICHÝ, V. (2022). *Atlas geometrie*. Academia, Praha, vydání 2. ISBN 978-80-200-3336-9.

Seznam obrázků

1.1	Příklad mapy pobřeží se spojnicí bodů A a B	6
1.2	Odhad délky pobřeží, kde $n = 10$ při zvoleném ε	6
1.3	Část pobřeží od bodu A v menším měřítku	7
1.5	Princip Archimédovy metody	8
1.6	Aproximace obvodu kružnice pomocí pravidelného šestnáctiúhelníku	8
1.7	Prvních pět iterací Kochovy křivky	9
1.8	První iterace Kochovy křivky „uvnitř“ druhé v menším měřítku .	10
1.9	První čtyři iterace Sierpiňského trojúhelníka	11
1.10	Nultá a první iterace Kochovy vločky	12
1.11	Čtvrtá iterace Kochovy vločky	13
1.12	Rozdelení první iterace Kochovy vločky	13
1.13	Nově vzniklé trojúhelníky v druhé iteraci	14
1.14	Nultá až čtvrtá iterace Cantorova diskontinua	15
1.15	Znázornění délek vyjmutých úseků	15
1.16	Soběpodobnost čtverce	16
1.17	Úsečka rozdelená na šest stejných částí	16
1.18	Krychle rozdelená na 27 stejných částí	17
1.19	Trojúhelník T rozdelený na trojúhelníky T_1, \dots, T_4	19
1.20	Různé možnosti (pod)pokrytí úsečky	22
1.21	Možné (pod)pokrytí čtverce	23
2.1	Ilustrace k důkazu věty 2.2.6	38
2.2	Vnitřní a vnější Jordanova-Panova míra množiny M	39
2.3	Ilustrace měřitelnosti množiny A	42
2.4	Ilustrace věty 2.4.4	46
2.5	Aproximace box-counting dimenze pobřeží Velké Británie	47
2.6	Felix Hausdorff, 1868–1942	55
2.7	Émile Borel, 1871–1956	56
2.8	Graf funkce $f(s) = \mathcal{H}^s(F)$, kde $F \subseteq \mathbb{R}^n$	62
3.1	Ilustrace věty 3.2.1	68
4.1	Znázornění Turingova stroje	73
4.2	Alan Turing, 1912–1954	73
4.3	David Hilbert, 1862–1943	74
4.4	Příklad syntaktického stromu věty: „Petr četl pěknou knihu.“	78
4.5	Interpretace vygenerovaných řetězců pro Kochovu křivku	81
4.6	Fraktální strom pro $n = 1, 2, 3, 4$	83
4.7	Znázornění zásobníku	84
4.10	Prostor vyplňující křivka	84

4.8	Varianta Kochovy křivky	85
4.9	Jezera a ostrovy	86
4.11	Fraktální strom I	87
4.12	Fraktální strom II	88
4.13	Fraktální strom III	89
4.14	L-systém pro Sierpiňského trojúhelník	90
4.15	Iterace zobrazení Ω (Sierpiňského trojúhelník)	98
4.16	Iterace zobrazení Ω s jiným počátečním útvarem B	99
4.17	Iterace zobrazení Φ (Sierpiňského koberec)	100
4.18	Pythagorův strom	101
4.19	Lévyho drak	102
4.20	Dračí křivka	102
4.21	Sierpiňského pětiúhelník	103
4.22	Modifikovaná Kochova křivka pro $a = 1/5$	106
4.23	Mandelbrotova množina	108
4.24	Příklady aproximace $K(f)$	110
4.25	Příklady aproximace $J(f)$	110
4.26	Příklady zbarvení bodů pro různé $K(f)$	113
4.27	Znázornění aproximace množiny \mathfrak{M}	115
4.28	Barevné znázornění aproximace množiny \mathfrak{M}	116
5.1	Ilustrace hraničního bodu	132
5.2	Grafické znázornění HSV modelu	134
5.3	Mandelbrotova množina pomocí lineární interpolace odstínu pro hodnoty $H_{\min} = 0; H_{\max} = 0,87; S_0 = V_0 = 1; m = 50$	136
5.4	Mandelbrotova množina pomocí lineární interpolace odstínu pro hodnoty $H_{\min} = 0,5; H_{\max} = 0,861; S_0 = 1; V_0 = 0,5; m = 50$	137
5.5	Mandelbrotova množina pomocí lineární interpolace odstínu pro hodnoty $H_{\min} = 0; H_{\max} = 0,87; S_0 = V_0 = 1; m = 100$	138
5.6	Mandelbrotova množina pomocí hladkého zbarvení pro hodnoty $H_{\min} = 0; H_{\max} = 0,87; S_0 = V_0 = 1; m = 50$	139

Seznam tabulek

1.1	Hodnoty dimenze d pro různé útvary	17
1.2	Porovnání fraktálních dimenzí d_k různých objektů	20
1.3	Porovnání fraktální a topologické dimenze útvarů	23
2.1	Odhady λ_3 pro vybrané útvary	49
4.1	Význam symbolů v abecedě V pro želvu	80
4.2	Význam symbolů v abecedě V pro želvu	82
4.3	Koeficienty IFS Ω pro Sierpiňského trojúhelník	97
4.4	Koeficienty IFS Φ pro Sierpiňského koberec	97

Rejstřík

- δ -mříž, 28
- δ -pokrytí, 31
- σ -aditivita, 36
- σ -algebra, 33
- σ -subaditivita, 36
- n -rozměrný kvádr, 39
- abeceda, 74
- aditivita, 36
- afinní zobrazení, 126
- algoritmus, 141
 - Time Escape, 111, 128, 141
- aritmetická míra, 35, 60
- atraktor, 95
- automat
 - deterministický, 73
 - lineárně omezený, 73
 - nedeterministický, 73
 - zásobníkový, 73
- axiom, 78, 119
- barevný model, 134
- barevný model HSV, 134
- Benoît Mandelbrot, 5
- bezkontextový L-systém, 79
- bilipschitzovské zobrazení, 53
- bod
 - pevný, 94
- Bolzanova-Cauchyova podmínka, 29
- borelovská množina, 56
- box-counting dimenze, 17, 25, 43, 44,
 - 103
 - dolní, 44
 - horní, 44
- Cantorovo diskontinuum, 14
- Cantorův prach, 14
- Cantorův princip, 68
- Carathéodoryho kritérium, 59
- cauchyovská posloupnost, 67
- Chomského hierarchie, 73, 79
- Constantin Carathéodory, 42
- deterministický L-systém, 79
- dimenze
 - box-counting, 17, 43, 44, 103
 - fraktální, 17, 141
 - Hausdorffova, 61, 62, 103
 - Hausdorffova-Bezikovičkova, 62
 - Lebesgueova pokrývací, 21
 - Minkowského, 25
 - Minkowského-Bouligandova, 25
 - topologická, 21
- Diracova míra, 35
- diskontinuum
 - Cantorovo, 14
- diskrétní metrický prostor, 27
- DOL-systém, 79
- dolní box-counting dimenze, 44
- délka slova, 74
- Entscheidungsproblem, 72
- eukleidovská geometrie, 5, 6
- faktor, 93
 - kontraktivní, 93
- Fatouova množina, 107, 109
- FIFO, 81
- first-class funkce, 127
- formální
 - gramatika, 74, 76
 - jazyk, 74, 75
- fraktál, 21
- fraktální dimenze, 17, 141
- fraktální geometrie, 5, 141
- fronta, 81
- funkce
 - first-class, 127
 - potenciálová, 137
 - první kategorie, 127

- funkce první kategorie, 127
- geometrie
 - eukleidovská, 5, 6
 - fraktální, 141
- Giuseppe Vitali, 42
- grafické rozhraní, 123
- gramatika, 74, 76
- Hausdorffova dimenze, 61, 62, 103
- Hausdorffova míra, 56
- Hausdorffův metrický prostor, 65, 67, 90
 - Hausdorffova metrika, 65
- Henri Lebesgue, 21
- horní box-counting dimenze, 44
- hranice množiny, 28
- HSV, 134
- hyperprostor, 65
- holderovské zobrazení, 60
- IFS, 141
- interpolace
 - Lagrangeova, 135
 - lineární, 134
- iterated function system, 93
- jazyk, 74, 75
 - programovací, 117
- jev
 - náhodný, 35
- Jordanova-Peanova míra, 39
- Juliova množina, 107, 109, 114, 141
- Kochova křivka, 9, 12
- Kochova vločka, 12
- kompilátor, 77
- komplexní číslo, 107
- Komplexně sdružené číslo, 107
- konkantenace, 75
- kontrakce, 32, 91
- kontraktivní faktor, 93
- kontraktivní zobrazení, 93
- konvergence
 - bodová, 31
 - stejnoměrná, 31, 68, 69
- kritérium
 - Carathéodoryho, 59
- kvádr, 28
 - objem kvádru, 39
- křivka
 - Kochova, 9, 12
- L-systém, 71, 78, 118, 141
 - deterministický, 79
 - nedeterministický, 79
 - stochastický, 81
- Lagrangeova interpolace, 135
- Lebesgueova míra, 25
 - n -rozměrná, 40
 - vnější, 40
- Lebesgueova pokrývací dimenze, 21
- lebesgueovská měřitelnost, 59
- LIFO, 81
- limes inferior, 29
- limes superior, 29
- limita
 - funkce v bodě, 30
 - posloupnosti, 29
- limita posloupnosti, 29
- lineární interpolace, 134
- lipschitzovské zobrazení, 53, 60
- Mandelbrotova množina, 107, 114
- Mandelbrot, 5
- Mandelbrotova množina, 22
- metrická míra, 58
- metrický prostor, 26
 - diskrétní, 27
 - Hausdorffův, 65, 67, 90
 - úplný, 30, 67
- metrika, 26
- množina
 - borelovská, 56
 - Fatouova, 107, 109
 - Juliova, 107, 109, 114, 141
 - kompaktní, 31
 - lebesgueovsky měřitelná, 42
 - Mandelbrotova, 107
 - Mandelbrotova, 22, 114
 - omezená, 27
 - otevřená, 21, 27
 - uzavřená, 27
 - vyplněná Juliova, 109
- mocnina slova, 75
- model
 - barevný, 134
 - monotonie míry, 36
 - míra, 33, 35, 39

aritmetická, 35, 60
Diracova, 35
Hausdorffova, 56
Jordanova-Peanova, 39
metrická, 58
vnější, 58
měřitelnost
lebesgueovská, 59
měřitelný prostor, 25, 33
nedeterministický L-systém, 79
neterminál, 76, 119
neterminální symbol, 76
náhodný jev, 35
objekty první kategorie, 127
objem kvádru, 28
odvození slova, 76
OL-systém, 79
open set condition, 104
otevřená množina, 21
pevný bod, 94
podobnost, 61
podpokrytí, 31
pokrytí, 31
 otevřené, 31
pole, 117
potenciálová funkce, 137
prach
 Cantorův, 14
pravděpodobnost, 35
pravidlo, 76
princip
 Cantorův, 68
programovací jazyk, 117
proměnná, 117
prostor
 metrický, 26
 měřitelný, 33
 s mírou, 35
prostor s mírou, 25, 35
prázdné slovo, 74
průměr množiny, 27
pseudokód, 117
Python, 117
přepisovací pravidlo, 76
rovnostranný trojúhelník, 10
rozhraní
grafické, 123
seznam, 117
Sierpińského koberec, 97
Sierpińského trojúhelník, 10
singulární číslo, 107
slovo, 74
soběpodobnost, 8, 78, 141
stejnoměrná konvergence, 69
stochastický L-systém, 81
stroj
 Turingův, 71
SVD roklad, 107
symbol, 74
syntaktická analýza, 77
systém, 33
 iterovaných funkcí, 64, 71
systém iterovaných funkcí, 64, 71, 90, 93, 141
teorie míry, 25, 141
terminál, 76
terminální symbol, 76
The Decision problem, 73
Time Escape algoritmus, 71, 111, 128, 141
tkinter, 123
topologická dimenze, 21
trojúhelník
 rovnostranný, 10
 Sierpińského, 10
Turingův stroj, 71
Unicode, 120
uzávěr množiny, 28
Vitali, 42
Vitaliho konstrukce, 42
vločka
 Kochova, 12
vnitřek množiny, 28
vnější míra, 58
vyplněná Juliova množina, 109
vzdálenost
 bodu od množiny, 27
 množin, 27
zjemnění, 21, 31
znak, 74
zobrazení

afinní, 126
bilipschitzovské, 32, 53
hölderovské, 60
kontraktivní, 91, 93
lipschitzovské, 32, 53, 60
zásobník, 81
zřetězení, 75

úplný metrický prostor, 67

řetězec, 74
želví grafika, 80, 118, 120