

# Fraktály a L-systémy

## Ročníková práce

Autor: David Dostal

Vedoucí práce: Radovan Daniel

2014/2015

## **Poděkování**

Děkuji svému konzultantovi, Radovanovi Danielovi, za odbornou konzultaci a trpělivost a své rodině za podporu.

## **Prohlášení**

Prohlašuji, že jsem tuto práci vypracoval samostatně a že jsem uvedl všechny použité prameny a literaturu

Datum

Podpis autora

## **Stanovisko vedoucího práce**

Souhlasím s předloženou podobou ročníkové práce.

Datum

Podpis vedoucího práce

## Obsah

1. Úvod.....	4
2. Teoretická část.....	5
2.1 Benoit Mandelbrot.....	5
2.2 Úvod do fraktální geometrie.....	8
2.3 Co je to fraktál?.....	9
2.3.1 Příklad fraktálu – Kochova křivka .....	9
2.3.1.1 Konstrukce.....	9
2.3.1.2 Vlastnosti.....	10
2.3.1.3 Délka Kochovy křivky.....	10
2.3.2 Fraktální dimenze .....	11
2.3.2.1 Nesnáze při měření pobřeží .....	11
2.3.2.2 Dimenze a měřítko .....	11
2.3.2.3 Výpočet fraktální dimenze .....	12
2.3.2.4 Hausdorffova dimenze dalších fraktálů .....	13
2.4 Dělení fraktálů.....	14
2.4.1 Deterministické a stochastické fraktály.....	14
2.4.2 Soběpodobné a soběpříbuzné fraktály .....	14
2.4.3 Dělení podle způsobu generování .....	15
2.4.3.1 Iterované funkční systémy .....	15
2.4.3.2 Time Escape Algoritmy .....	16
2.4.3.3 L-systémy.....	20
2.4.3.4 Náhodné fraktály, podivné atraktory a další.....	21
2.5 Využití.....	22
2.5.1 Fraktálové antény.....	22
2.5.2 Fraktálová komprese .....	22

2.5.3	Počítačově vytvořené krajiny .....	23
2.5.4	Medicína .....	24
2.5.5	Další využití .....	24
2.6	Lindenmayerovy systémy .....	25
2.6.1	Přepisování, iterace .....	25
2.6.2	Želví grafika.....	25
2.6.3	Souvislé L-systémy .....	27
2.6.4	Větvení.....	28
2.6.5	Další rozšíření.....	29
2.6.5.1	Vykreslování ve 3D .....	29
2.6.5.2	Stochastické L-systémy .....	29
2.6.5.3	Parametrické L-systémy .....	30
2.6.5.4	Podmíněné přepisování .....	31
2.6.5.5	Kontextové L-systémy .....	31
2.6.5.6	Další možná rozšíření .....	31
3.	Praktická část.....	32
4.	Umělecká část.....	33
5.	Závěr .....	34

# 1. Úvod

Před letními prázdninami (2014) jsme si měli vybrat téma ročníkové práce. Věděl jsem, že bych chtěl zpracovat práci na téma z přírodovědné oblasti (matematika, fyzika, chemie či biologie) nebo na téma z oblasti informatiky (především jsem chtěl něco naprogramovat) nebo zpracovat nějaké spíše umělecké téma (hudba, grafika.) Z toho jsem získal dvě možná témata: „Tvorba počítačové hry“ a „Nasazení redakčního systému Wordpress pro školy“. Nakonec vyhrálo nasazení redakčního systému pro školy a to hlavně z důvodu, že bych v praktické části pracoval na webových stránkách školy a dělal něco opravdu užitečného.

Přes letní prázdniny jsem ale zjistil, že by mně zpracování tohoto tématu vůbec nebavilo. A to byl jeden z nejdůležitějších požadavků, které jsem na ročníkovou práci měl. Mohli jsme naštěstí téma na začátku školního roku ještě změnit, tak jsem začal nad tématem přemýšlet znovu. Spolu s tátou jsme přišli na několik zajímavých témat, mezi kterými byla například „Překlopitelná tělesa“, jeden z chemických prvků, „Co je to čas?“, „Založení vlastního podniku“ anebo „Proč v Pardubicích neprší“. U některých z témat bylo složité vytvořit smysluplnou praktickou část, u jiných by zase nebylo co psát v teoretické části. V těchto dnech jsem se náhodou díval na dokument o fraktálech, který mě velmi zaujal. Téma fraktálů mi také umožňovalo jako praktickou část něco naprogramovat a nakonec jsem si toto téma vybral.

V teoretické části čtenáře nejprve seznámím s fraktály obecně (tzn. například s tím, co to fraktál vlastně je, jak se fraktály dělí a jak se dají generovat, kde nacházejí praktické využití). Více se pak zaměřím na jednu kategorii fraktálů, takzvané Lindenmayerovy systémy, které mne zaujaly svým jednoduchým principem a možností i přesto generovat složité rostlinné obrazce. Jako praktickou část jsem naprogramoval počítačovou aplikaci na generování L-systémů a naprogramoval jsem pokročilejší L-systém několika rostlin v online generátoru malsys.cz. Jako uměleckou část ročníkové práce jsem namaloval obraz inspirovaný fraktální tematikou a fraktály v přírodě – noční krajinu s fraktálovým stromem.

## 2. Teoretická část

### 2.1 Benoit Mandelbrot

Benoit B. Mandelbrot se narodil roku 1924 v Polsku do židovské rodiny. Již velmi brzy si oblíbil geometrii. Také hrál rád šachy. V šachách ale nevynikal díky logickému přemýšlení. Hru promýšlel geometricky.<sup>1</sup>

V roce 1936, když bylo Mandelbrotovi 11 let, začal v Německu sílit nacismus a on s rodinou emigroval do Francie. Několik let chodil do Lycée Rolin v Paříži. Pak se Mandelbrotovi při vypuknutí 2. světové války (1936) přestěhovali do Tulle ve Francii. Během války nad ním i jeho rodinou visela neustálá hrozba chudoby a obavy z udání a následného posláním na smrt. Strach ještě zesílil po zabití pařížské fyzičky Ziny Morhange, jeho blízké známé. Za války se učil hlavně samostudiem. Díky tomu se rozvinul jeho geometrický přístup k matematice.<sup>2</sup>

Na konci války (v roce 1944) se Mandelbrot navrátil do Paříže. Zde se v letech 1945-1947 učil na École Polytechnique. Mimo jiné zde studoval i pod Gastonem Juliem a Paulem Lévyem, kteří též významně přispěli k rozvoji fraktální geometrie.<sup>3</sup>

V letech 1947 až 1949 navštěvoval Kalifornský technologický institut. Po získání doktorátu matematiky na Pařížské univerzitě (1952) se znovu vrátil do Ameriky, kde studoval v Princetonu. 1955 se vrátil do Francie, kde si vzal za ženu Aliettu Kagan. Chvilku učil v Paříži jako profesor na Université Lille a v Národním výzkumném centru v Paříži. Byl ale nespokojený s příliš abstraktním rázem matematiky v tehdejší Francii a proto odešel do USA. Zde získal práci v IBM (1958).<sup>4</sup>

V IBM zkoumal náhodný šum způsobující poruchy v přenosu dat přes telefonní linku. Mandelbrot byl zvyklý dívat se na věci vizuálně. Zkoumal proto šum hlavně z hlediska tvarů, které vytvářel. Všiml si, že chyby nepřicházely zcela náhodně, jak se předpokládalo. Chyby nastávaly ve shlucích. Byla období zcela bez chyb a pak zase

---

<sup>1</sup> Guardian obituary. The MacTutor History of Mathematics archive [online]. 18 October 2010 [cit. 2015-02-27]. Dostupné z: [http://www-history.mcs.st-and.ac.uk/Obits2/Mandelbrot\\_Guardian.html](http://www-history.mcs.st-and.ac.uk/Obits2/Mandelbrot_Guardian.html)

<sup>2</sup> Benoit Mandelbrot. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: [http://en.wikipedia.org/wiki/Benoit\\_Mandelbrot](http://en.wikipedia.org/wiki/Benoit_Mandelbrot)

<sup>3</sup> FRAME, Michael. Benoit B. Mandelbrot: A Biographical Memoir. National Academy of Sciences [online]. © 2014 [cit. 2015-02-27]. Dostupné z: <http://www.nasonline.org/publications/biographical-memoirs/memoir-pdfs/mandelbrot-benoit.pdf>

<sup>4</sup> GREGOROVÁ, Dagmar. Stal jsem se konstruktérem geometrie. OSEL.cz [online]. 19.10.2010 [cit. 2015-02-27]. Dostupné z: <http://www.osel.cz/index.php?clanek=5346>

s mnoha chybami. Mandelbrot zde objevil určitou pravidelnost. Ať už se díval na výskyt chyb v rámci měsíce, dne i hodiny, poměr mezi časy bez chyb a obdobími s chybami byl stejný. Když se na období s chybami podíval zblízka, bylo vidět, že i tato část grafu je rozdělena na zase menší části s chybami a bez chyb. Tyto menší části byly překvapivě podobné grafu jako celku (například během měsíce podobná jako během týdne i během dvaceti minut). Při jakémkoli měřítku byla křivka podobná sama sobě. Tak udělal Mandelbrot první velký krok vstříc fraktálům.<sup>5</sup>

Mandelbrot postupně začal objevovat vlastnost „soběpodobnosti“ i v jiných oblastech. Zkoumal mraky, zjistil, že soběpodobnost lze nalézt i v celém vesmíru. Přišel na to, že ačkoli je cena bavlny během roku, měsíce i dne různá, křivky průběhu denního měsíčního i ročního prodeje jsou shodné.<sup>6</sup>

V roce 1967 napsal práci s názvem „Jak dlouhé je pobřeží Británie?“. Zjistil, že na tuto otázku nelze tak jednoduše odpovědět. Všiml si, že různá měření délky pobřeží se často velmi liší. Čím podrobnější byly mapy, tím delší se pobřeží zdálo být. Při přesnějším měření se berou v ohled i menší zákruty a zátoky, které se předtím zanedbaly, při ještě podrobnějším zkoumání i zátoky v zátocích a tak dále – teoreticky až ke každému kameni, každému zrnku písku.<sup>7</sup>

Mandelbrot si vzpomněl na práci Gastona Julia a Pierra Fatou. Již před lety ho jeho strýc Szolem Mandelbrojt upozornil na jejich dílo zabývající se takzvanými Juliovými množinami, které vycházejí z jednoduché rovnice  $z = z^2 + c$ . Mandelbrot mu však tehdy nevěnoval velkou pozornost a nepřišel zde na nic nového. Nyní se k této práci vrátil. Využil počítačů, s pomocí kterých mohl tuto rovnici jednoduše a rychle opakovat – použít její výsledek jako nový vstup. Následně výsledek na počítači nechal vykreslit. Když se Mandelbrot na něj podíval detailněji, viděl, že obraz obsahoval menší útvary, které vypadaly podobně jako celek. Fascinující bylo, že jak se člověk přesněji podíval na detail množiny, objevoval čím dál tím více podrobností, nové a nové útvary. A přeci se zde dala

---

<sup>5</sup> Fractal Geometry. IBM United States [online]. May 18, 2011 [cit. 2015-02-27]. Dostupné z: <http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/fractal/>

<sup>6</sup> Telegraph obituary. The MacTutor History of Mathematics archive [online]. © 2010 [cit. 2015-02-27]. Dostupné z: [http://www-history.mcs.st-and.ac.uk/Obits2/Mandelbrot\\_Telegraph.html](http://www-history.mcs.st-and.ac.uk/Obits2/Mandelbrot_Telegraph.html)

<sup>7</sup> Telegraph obituary. The MacTutor History of Mathematics archive [online]. © 2010 [cit. 2015-02-27]. Dostupné z: [http://www-history.mcs.st-and.ac.uk/Obits2/Mandelbrot\\_Telegraph.html](http://www-history.mcs.st-and.ac.uk/Obits2/Mandelbrot_Telegraph.html)

najít podobnost s celým obrazcem. Záleželo pouze na výkonu počítače, jak daleko mohl člověk takto pokračovat.<sup>8</sup>

Tento útvar – Mandelbrotova množina, vzdáleně připomínající tvary korálů, je krásným příkladem „fraktálu“. Pojem „fraktál“ použil Mandelbrot poprvé v roce 1975 v práci „Fraktály: tvar, náhoda a dimenze“. Z ní také vychází kniha shrnující jeho dosavadní objevy nesoucí název „The Fractal Geometry of Nature“ („Fraktální geometrie přírody“) vydané nejprve ve francouzštině a roku 1982 i v angličtině. Mandelbrot zde popisuje, v čem se fraktální geometrie liší od klasické, Euklidovské. Geometrie, jak ji známe my, dokáže přesně popsat předměty námi vytvořené. Těžko ale popisuje jevy v přírodě, které nejsou rovné a hladké, ale hrubé a nerovné. „Mraky nejsou koule, hory nejsou kužely, pobřeží nejsou kruhy, a kůra není hladká, ani blesky nejdou po přímé linii“<sup>9</sup>, napsal Mandelbrot. Tyto jevy, včetně mnoha dalších, lze popsat pomocí fraktální geometrie.<sup>10</sup>

Po zbytek svého života se Mandelbrot i nadále věnoval výzkumu fraktálů. Učil na Harvardu a na Massachusettském technologickém institutu a od roku 1999 pracoval jako profesor na univerzitě v Yale. Předpověděl finanční krizi v roce 2004. Za své úspěchy dostal mnohá ocenění. Benoit B. Mandelbrot zemřel 14. 10. 2010 ve stáří 85 let.<sup>11</sup>

---

<sup>8</sup> Fractal Geometry. IBM United States [online]. May 18, 2011 [cit. 2015-02-27]. Dostupné z: <http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/fractal/>

<sup>9</sup> MANDELBROT. *Fractal geometry of nature*. New York 1983, s. 1

<sup>10</sup> Telegraph obituary. The MacTutor History of Mathematics archive [online]. © 2010 [cit. 2015-02-27]. Dostupné z: [http://www-history.mcs.st-and.ac.uk/Obits2/Mandelbrot\\_Telegraph.html](http://www-history.mcs.st-and.ac.uk/Obits2/Mandelbrot_Telegraph.html)

<sup>11</sup> [http://www-history.mcs.st-and.ac.uk/Obits2/Mandelbrot\\_Telegraph.html](http://www-history.mcs.st-and.ac.uk/Obits2/Mandelbrot_Telegraph.html) (21. 1. 2014)



## 2.2 Úvod do fraktální geometrie

„Proč je geometrie často popisována jako „studená“ a „suchá?“ Jeden z důvodů spočívá v její neschopnosti popsat tvar oblaku, hory, pobřeží, nebo stromu. Mraky nejsou koule, hory nejsou kužely, pobřeží nejsou kruhy, a kůra není hladká, stejně jako se blesk nepohybuje po přímé linii.“<sup>12</sup> – Benoit B. Mandelbrot

Euklidovská geometrie, tak jak ji známe ze škol, je perfektní k popisu hladkých útvarů. Dokáže popsat přímku, kruh, krychli, dovedeme s její pomocí spočítat objem válce či obsah čtverce. Podíváme-li se ale z okna, zjistíme, že svět není tvořen z dokonalých kvádrů, válců, jehlanů, koulí a tak podobně. Zkuste v přírodě najít útvar, který by šel přesně popsat pomocí klasické geometrie. Takovéto útvary se v přírodě vyskytují velmi zřídka, téměř vůbec. Vodní hladina za úplného bezvětří by mohla tvořit rovinu, měsíc viděný z dálky se nám může zdát jako kruh. Například strom za oknem bychom mohli popsat pomocí mnoha různých válců. Když se ale podíváme blíže, zjistíme, že je to dost nepřesné. Náš model tedy upravíme tak, aby zahrnoval každou větvíčku i každý list na větvích. Co nyní? Podíváme se přesněji a zjistíme, že každý list má na sobě složitý vzor, možná se podobající celému stromu. Zpozorujeme, že větve na stromu nejsou perfektně hladké, ale že jejich kůra je hrubá a vroubkovitá. Takto bychom mohli pořád pokračovat. Konečně, ani měsíc není perfektním kruhem a má na svém povrchu spoustu nerovností. A i při zdánlivém bezvětří můžeme na hladině jezera pozorovat malinkaté vlnky. Ukazuje se, že je těžké nenarazit při každém kroku na fraktální útvar. I u věcí vytvořených člověkem, zdánlivě rovných a hladkých, jako může být například protější panelák v ulici, lze najít nerovnosti. Stačí se podívat dostatečně přesně, třeba na omítku. Takovéto útvary, kdy nám již přestává stačit klasická euklidovská geometrie, se snaží popsat fraktální geometrie.<sup>13</sup>

---

<sup>12</sup> MANDELBROT. *Fractal geometry of nature*. New York 1983, s. 1

<sup>13</sup> SIXTA, Tomáš. Úvod do fraktálů a chaosu. ITnetwork.cz [online]. © 2015 [cit. 2015-02-27]. Dostupné z: <http://www.itnetwork.cz/fraktaly-a-chaos-eukleidovska-geometrie-uvod-svitani-chaosu>

## 2.3 Co je to fraktál?

Fraktál je množina nebo geometrický útvar, který lze charakterizovat následujícími typickými vlastnostmi:

- **Nekonečná členitost**

Fraktální útvary jsou nekonečně členité. Oproti klasickým geometrickým útvarům (jako například oproti přímce, která je při zvětšení stále stejná), můžeme u fraktálů při jakémkoli zvětšení vidět nekonečně podrobné obrazce.<sup>14</sup>

- **Soběpodobnost**

Jednotlivé části fraktálu se podobají fraktálu jako celku. Neustále se v něm opakují ty samé geometrické motivy. Fraktály mohou být úplně soběpodobné (složené z menších kopií sebe samého) nebo soběpříbuzné (jednotlivé části nejsou přesnými zmenšeninami celku, pouze se mu podobají).<sup>15</sup>

- **Nezávislost na měřítku**

Soběpodobné fraktály mají ještě další vlastnost – tzv. nezávislost na změně měřítka. Znamená to, že se při zvětšování fraktálu budou objevovat pořád ty samé útvary – při různém zvětšení bude vypadat pořád stejně.<sup>16</sup>

### 2.3.1 Příklad fraktálu – Kochova křivka

#### 2.3.1.1 Konstrukce

Kochovu křivku zkonstruujeme následovně. Začneme jednotkovou úsečkou. Ta je naším výchozím útvarem, tzv. iniciátorem. Dále potřebujeme tzv. generátor, což je útvar, kterým se nahradí iniciátor. Ten získáme rozdělením iniciátoru na třetiny a nahrazením prostřední třetiny dvěma úsečkami o délce  $\frac{1}{3}$ , které svírají úhel  $60^\circ$ . Nyní opakovaně nahrazujeme generátor iniciátorem. Nejprve získáme křivku složenou ze čtyř úseček o délce  $\frac{1}{3}$ . Každá z těchto úseček se stane iniciátorem pro další krok, ve kterém získáme 16 úseček o délce  $\frac{1}{9}$ . Po nekonečně mnoha opakováních (iteracích) získáme Kochovu křivku. Složením tří Kochových křivek do „hvězdy“ sestavíme Kochovu vločku.<sup>17</sup>

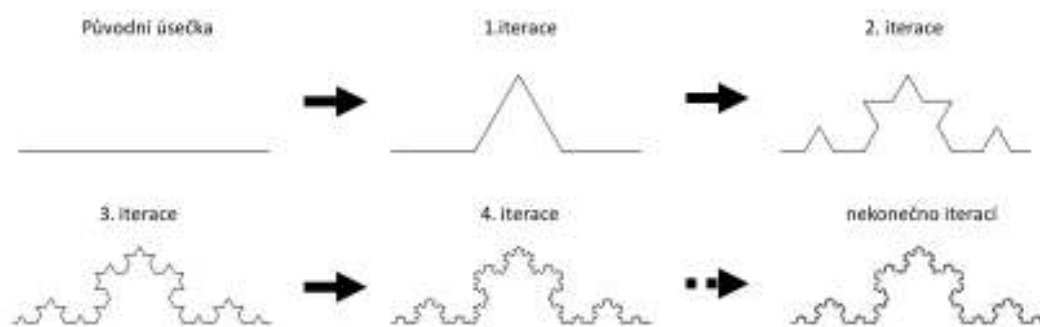
---

<sup>14</sup> ZELINKA, VČELAŘ a ČANDÍK. Fraktální geometrie: principy a aplikace. Praha 2006, strana 88

<sup>15</sup> ZELINKA, VČELAŘ a ČANDÍK. Fraktální geometrie: principy a aplikace. Praha 2006, strana 87-88

<sup>16</sup> ZELINKA, VČELAŘ a ČANDÍK. Fraktální geometrie: principy a aplikace. Praha 2006, strana 87

<sup>17</sup> HOTAŘ, Vlastimil. Fraktální geometrie a fraktály. Fraktální geometrie [online]. [2006] [cit. 2015-02-27]. Dostupné z: <http://www.ksr.tul.cz/fraktaly/geometrie.html>



Obr. 1: Konstrukce Kochovy křivky.

### 2.3.1.2 Vlastnosti

Na Kochově křivce můžeme dobře vidět zatím popsané vlastnosti fraktálů. Je složena ze čtyř (zmenšených a otočených) kopií sebe samé. A ty jsou zase složeny ze zmenšených kopií celé křivky. Kochova křivka je tedy nekonečně podrobná, soběpodobná a při různém měřítku vypadá stejně. Ani při velikém zvětšení nikde nenajdeme rovnou úsečku, vždy budou vidět pouze menší a menší zákruty. Z toho také plyne, že u Kochovy křivky nelze v žádném bodě určit derivaci<sup>18</sup> (určit právě jednu tečnu).

### 2.3.1.3 Délka Kochovy křivky

V první iteraci vytváření Kochovy křivky získáme čtyři nové úsečky o délce  $\frac{1}{3}$ . To odpovídá celkové délce  $4 \cdot \frac{1}{3} = \frac{4}{3}$ . Ve druhém kroku se z každé úsečky stanou čtyři nové. Počet úseček tedy bude  $4 \cdot 4 = 4^2$ . Délka úseček je oproti předchozí iteraci zase třetinová. To znamená, že délka každé z nich bude  $\frac{1}{3} \cdot \frac{1}{3} = \left(\frac{1}{3}\right)^2$ . Celková délka křivky je nyní  $\left(\frac{1}{3}\right)^2 \cdot 4^2 = \left(\frac{4}{3}\right)^2$ . Pro třetí iteraci je délka křivky  $\left(\frac{1}{3}\right)^3 \cdot 4^3 = \left(\frac{4}{3}\right)^3$ . A pro  $n$ -tou iteraci bude  $\left(\frac{4}{3}\right)^n$ . Protože je počet iterací nekonečný, lze říci, že celková délka Kochovy křivky  $L = \lim_{n \rightarrow \infty} \left(\frac{4}{3}\right)^n = +\infty$ . Délka Kochovy křivky je tedy nekonečná.<sup>19</sup>

<sup>18</sup> HOTAŘ, Vlastimil. Fraktální geometrie a fraktály. Fraktální geometrie [online]. [2006] [cit. 2015-02-27]. Dostupné z: <http://www.ksr.tul.cz/fraktaly/geometrie.html>

<sup>19</sup> JABLONSKI, Adrian. Grundlagen der fraktalen Geometrie mit iterierten Funktionensystemen (IFS). Quadsoft.org [online]. Juli 2011 [cit. 2015-02-27]. Dostupné z: <http://quadsoft.org/fraktale/>

## 2.3.2 Fraktální dimenze

### 2.3.2.1 Nesnáze při měření pobřeží

Běžné geometrické útvary mají celočíselnou dimenzi. Například úsečka má dimenzi 1, dimenze čtverce je 2 a krychle je trojrozměrná. Kdybychom tyto útvary měřili různě velikými „měřidly“, vždy bychom změřili stejnou celkovou velikost (délku, obsah, objem). Měřit přírodní útvary ale není tak jednoduché. Kdybychom třeba měřili délku pobřeží ostrova, zjistili bychom, že u stejného pobřeží můžeme při různých měřeních získat odlišné výsledky. Například při měření dvoumetrovou tyčí bychom naměřili mnohem větší délku, než kdybychom měřili po 100 metrech. Se zkracováním úseků, po kterých se měří, se naměřená délka zvětší o nové detaily. Přibudou nové zákruty, zohlední se více podrobností členitého pobřeží, které byly při méně podrobném měření ignorovány. Teoreticky, kdybychom zmenšovali měřidlo až do nekonečna, můžeme naměřit i nekonečnou délku pobřeží. Hranice pobřeží je křivka, tudíž má dimenzi 1. Je ale teoreticky nekonečně dlouhá a zabírá tak „více místa“, než třeba úsečka. Zároveň ani nevyplňuje žádnou plochu. Můžeme tušit, že pobřeží má kromě klasické (tzv. topologické) dimenze ještě jinou, „něco mezi“ dimenzí 1 a 2. Tím se dostáváme k Hausdorffově neboli fraktální dimenzi.<sup>20</sup>

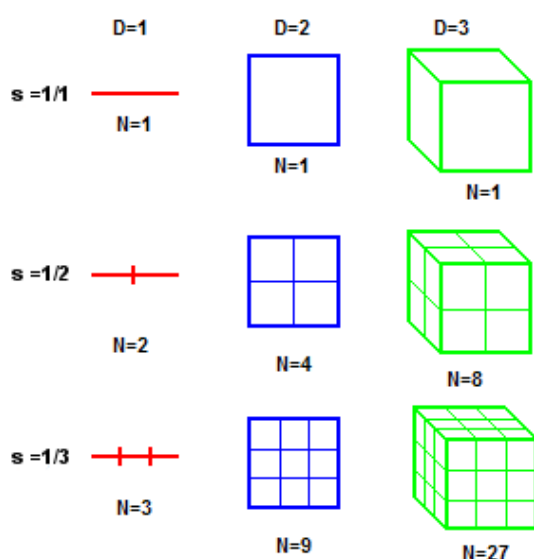
### 2.3.2.2 Dimenze a měřítko

Vraťme se ještě trochu a podívejme se na to, jak souvisí dimenze útvaru s měřítkem. Začneme úsečkou. Měřidlo (úsečka) o stejné délce se do ní vejde jednou. Použijeme-li měřící úsečku o poloviční délce, vejde se dvakrát. A když bude její délka třetinová, vejde se třikrát. Nyní zkusme to samé se čtvercem. Měřící čtverec o poloviční velikosti se do čtverce vejde čtyřikrát. Kdyby měl měřící čtverec třetinovou velikost, vešel by se devětkrát. Do krychle by se vešlo osm kostek o poloviční a dvacet sedm o třetinové velikosti. To, kolikrát se měřidlo do daného útvaru vejde (nebo také na kolik menších částí se útvar rozdělí), závisí na dimenzi daného útvaru, roste s mocninou jeho dimenze. U úsečky to byly první mocniny  $(1^1, 2^1, 3^1)$ , u čtverce druhé mocniny  $(1^2, 2^2, 3^2)$  a u krychle třetí mocniny  $(1^3, 2^3, 3^3)$ . Obecně platí vztah  $N = \left(\frac{1}{s}\right)^D$ , kde  $N$  je počet částí (kolikrát se měřidlo vejde do útvaru),  $s$  je velikost měřidla (velikost menších částí, na který

---

<sup>20</sup> ZELINKA, VČELARŠ a ČANDÍK. Fraktální geometrie: principy a aplikace. Praha 2006, strana 80-81

lze útvar rozdělit) a  $D$  je dimenze daného objektu. Například do krychle se kostky o poloviční velikosti vejdu  $N = \left(\frac{1}{1/2}\right)^3 = 8$ krát.<sup>21</sup>



Obr. 2: Závislost měřítka a dimenze.

### 2.3.2.3 Výpočet fraktální dimenze

Teď se podíváme na to, jak je to s fraktály. Začněme Kochovou křivkou. Úsečka o délce  $\frac{1}{3}$  se do ní vejde 4krát, úsečka dlouhá  $\frac{1}{9}$  pak 16krát. Když nyní dosadíme do vzorce  $N = \left(\frac{1}{s}\right)^D$ , získáme následující rovnici:  $4 = \left(\frac{1}{1/3}\right)^D$ . Po úpravě získáme  $4 = 3^D$ . Platí to, i když dosadíme ostatní čísla, třeba  $s = \frac{1}{9}$  a  $N = 16$ , vždy nám vyjde stejný poměr. Nyní obě strany rovnice zlogaritmujeme:  $\log 4 = D \cdot \log 3$ . Rovnici vydělíme  $\log 3$  a získáme Hausdorffovu dimenzi Kochovy křivky  $D = \frac{\log 4}{\log 3} \doteq 1,2619$ . Vidíme, že Hausdorffova dimenze Kochovy křivky je větší než jedna, což je její topologická dimenze, ale zároveň ještě není dvojrozměrná.<sup>22</sup>

Neceločíselná Hausdorffova dimenze je typickou vlastností fraktálních útvarů. Obecně se Hausdorffova dimenze spočítá jako  $D = \frac{\log N}{\log 1/s}$ . Nyní již jsme schopni pochopit matematickou definici fraktálu: „Fraktál je množina, jejíž Hausdorffova dimenze je (ostře) větší než dimenze topologická.“<sup>23 24</sup> Ukažme si, že to platí i pro další fraktály.

<sup>21</sup> ZELINKA, VČELAŘ a ČANDÍK. Fraktální geometrie: principy a aplikace. Praha 2006, strana 82-83

<sup>22</sup> MANDELBROT. *Fractal geometry of nature*. New York 1983, s. 36

<sup>23</sup> Fraktál. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: <http://cs.wikipedia.org/wiki/Frakt%C3%A1l>

### 2.3.2.4 Hausdorffova dimenze dalších fraktálů

**Sierpinského trojúhelník** vzniká rozdělením rovnostranného trojúhelníku na čtyři menší a vyjmutím prostředního trojúhelníku. To se znovu opakuje pro zbylé tři trojúhelníky. Topologická dimenze  $D_T$  Sierpinského trojúhelníku je 1, jelikož neobsahuje žádnou souvislou plochu (vyjmutím dalších a dalších trojúhelníků je plocha na konci nulová). Sierpinského trojúhelník si můžeme též představit jako nekonečně dlouhou křivku. S každou iterací se počet trojúhelníků (úseček) ztrojnásobí, jejich velikost je poloviční. Hausdorffova dimenze tedy bude  $D = \frac{\log 3}{\log 2} \doteq 1,585$ . Platí tedy  $D > D_T$ .<sup>25</sup>



Obr. 3: Konstrukce Sierpinského trojúhelníku.



Obr. 4: Sierpinského trojúhelník jako křivka.

**Kantorova množina** vzniká rozdělením úsečky na tři části, prostřední část je vyjmuta z množiny. Opakovaně vždy z každé další úsečky vyjmeleme prostřední třetinu. Topologická dimenze Kantorovy množiny  $D_t = 0$ . Obsahuje sice nekonečně mnoho bodů, po nekonečně mnoha iteracích ale již neobsahuje žádnou úsečku (ta by se ještě dále dělila). V každé iteraci získáme 2 nové úsečky o délce  $\frac{1}{3}$ . Hausdorffova dimenze Kantorovy množiny bude tím pádem  $D = \frac{\log 2}{\log 3} \doteq 0,631$ .<sup>26</sup>



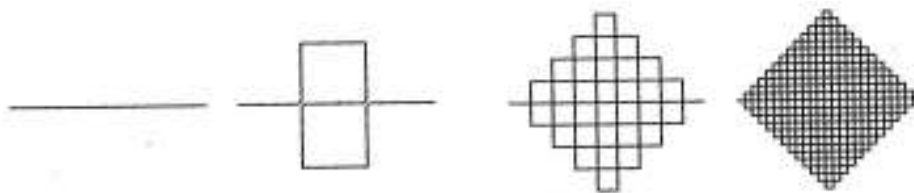
Obr. 5: Šest iterací Kantorovy množiny.

<sup>24</sup> MANDELBROT. *Fractal geometry of nature*. New York 1983, s. 15

<sup>25</sup> ZELINKA, VČELARĚ a ČANDÍK. *Fraktální geometrie: principy a aplikace*. Praha 2006, strana 85

<sup>26</sup> ZELINKA, VČELARĚ a ČANDÍK. *Fraktální geometrie: principy a aplikace*. Praha 2006, strana 85

**Fraktály vyplňující rovinu**, jako například obměna **Peanovy křivky**, jsou obzvláště zajímavé. Princip generování by měl být jasný z obrázku. V každém kroku se úsečka nahradí novou křivkou sestávající z devíti úseček o třetinové délce. Tím pádem získáme Hausdorffovu dimenzi  $D = \frac{\log 9}{\log 3} = 2$ . Popsali jsme si křivku, která má topologickou dimenzi 1 a přitom vyplňuje dvojrozměrnou plochu. Ne všechny fraktály tedy mají neceločíselnou Hausdorffovu dimenzi, vždy je ale větší než dimenze topologická.<sup>27</sup>



Obr. 6: Křivka, která vyplňuje plochu.

## 2.4 Dělení fraktálů

### 2.4.1 Deterministické a stochastické fraktály

Deterministické fraktály jsou nenáhodné, pravidelné a jsou dané přesnými pravidly. Při opakovaném generování je výsledek pokaždé stejný. Stochastické fraktály oproti tomu obsahují prvek náhody. Díky tomu může fraktál vypadat při opakovaném generování různě.<sup>28</sup>

### 2.4.2 Soběpodobné a soběpříbuzné fraktály

Soběpodobné fraktály jsou složeny ze zmenšených částí, které jsou přesnými kopiemi celého fraktálu. U soběpříbuzných fraktálů můžeme sice vidět podobné opakující se motivy, ale nelze již najít přesné zmenšeniny. Soběpodobnými fraktály jsou například Kochova vložka nebo Sierpinského trojúhelník, soběpříbuzná je třeba Mandelbrotova množina, u které při zvětšování můžeme nacházet stále nové a nové útvary.<sup>29</sup>

<sup>27</sup> MANDELBROT. *Fractal geometry of nature*. New York 1983, s. 62

<sup>28</sup> HOTAŘ, Vlastimil. Fraktální geometrie a fraktály. Fraktální geometrie [online]. [2006] [cit. 2015-02-27].

Dostupné z: <http://www.ksr.tul.cz/fraktaly/geometrie.html>

<sup>29</sup> ZELINKA, VČELARŠ a ČANDÍK. Fraktální geometrie: principy a aplikace. Praha 2006, strana 30

## 2.4.3 Dělení podle způsobu generování

### 2.4.3.1 Iterované funkční systémy

Iterované funkční systémy (zkráceně IFS) fungují na principu opakovaného použití transformací, jako je posunutí, rotace a změna měřítka (transformační funkce jsou jednoduché funkce, které pracují s x-ovou a y-ovou souřadnicí bodu, například posun doprava znamená přičítání k x-ové souřadnici). Uvedme si příklad na Sierpinského trojúhelníku. K jeho generování jsou potřeba tři transformační funkce. První funkce zmenší čtverec na polovinu jak ve směru osy x, tak i y:  $F_1(x; y) = (\frac{x}{2}; \frac{y}{2})$ . Druhá funkce má tento tvar:  $F_2(x; y) = (\frac{x+1}{2}; \frac{y}{2})$ . S původním čtvercem udělá to, že jej zmenší na polovinu a ještě ho posune o  $\frac{1}{2}$  doprava. Poslední funkce zmenší čtverec na polovinu a posune ho o  $\frac{1}{2}$  nahoru:  $F_3(x; y) = (\frac{x}{2}; \frac{y+1}{2})$ .<sup>30</sup>

IFS fraktály jsou často popisovány pomocí transformačních matic, které udávají jednotlivé transformace a se kterými program dokáže dobře pracovat. Nejčastější metodou generování IFS fraktálů je tzv. „hra chaosu“. Například Sierpinského trojúhelník lze vygenerovat následujícím způsobem. Nejprve zkonstruujeme trojúhelník. Náhodně zvolíme výchozí bod na jeho obvodu. Dále náhodně vybereme jeden z vrcholů trojúhelníka. V polovině mezi bodem a vrcholem zakreslíme bod, který se stane naším novým výchozím bodem. Několikrát opakujeme postup od vybrání náhodného vrcholu. Poté náhodně zvolíme nový výchozí bod na obvodu a vše opakujeme.<sup>31</sup>

---

<sup>30</sup> DRAVES, Scott a Erik RECKASE. The Fractal Flame Algorithm. RECKASE, Erik. The Flame Algorithm [online]. September 2003, November 2008 [cit. 2015-02-27]. Dostupné z: [http://flam3.com/flame\\_draves.pdf](http://flam3.com/flame_draves.pdf)

<sup>31</sup> ZELINKA, VČELARŠ a ČANDÍK. Fraktální geometrie: principy a aplikace. Praha 2006, strana 47-48

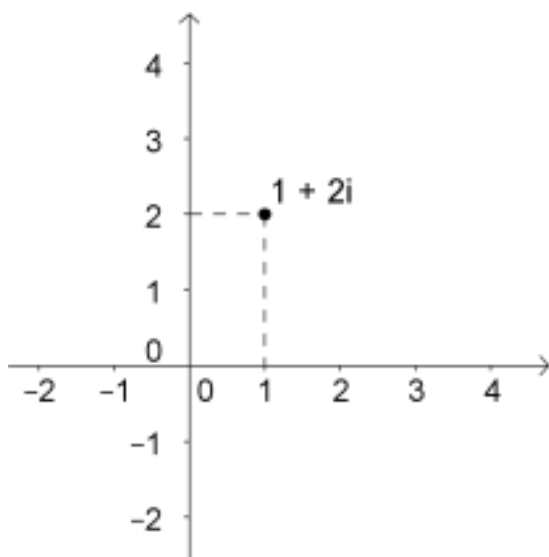


### 2.4.3.2 Time Escape Algoritmy

TEA fraktály se generují iterací rovnice s komplexními čísly. Nejprve si stručně popíšeme, co to komplexní čísla jsou.

#### **Komplexní čísla**

Komplexní číslo obsahuje dvě části: část reálnou a část imaginární. Zapisuje se ve tvaru  $a + bi$ , kde  $a$  je reálná a  $b$  imaginární část. Tato dvě čísla  $a$  a  $b$  jsou reálná čísla, která určují souřadnice bodu v komplexní rovině. Imaginární číslo  $i$  představuje  $\sqrt{-1}$ . Odmocninu ze záporného čísla sice nedokážeme spočítat, dokážeme s ní ale pracovat i přesto, že neznáme její číselnou hodnotu. S imaginárním  $i$  pracujeme jednoduše jako s proměnnou.<sup>32</sup>



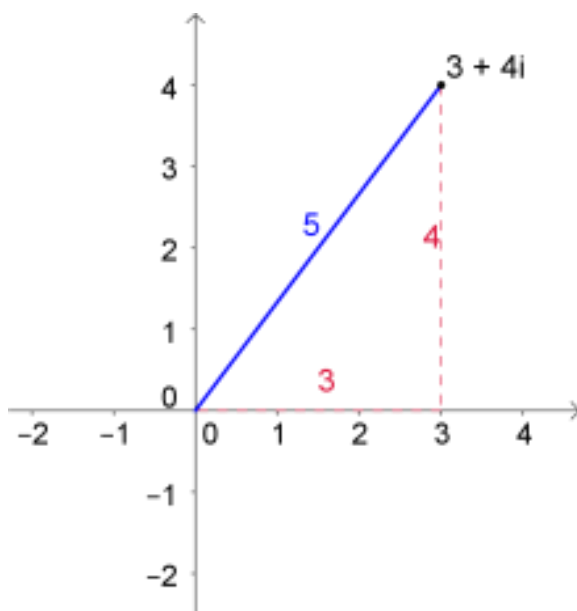
Obr. 7: Číslo  $1+2i$  v komplexní rovině.

---

<sup>32</sup> PATRZALEK, Edyta. Introduction to Fractal Geometry. Fractal.org: Centre for Fractal Design and Consultancy [online]. [2008] [cit. 2015-02-27]. Dostupné z: <http://www.fractal.org/Bewustzijns-Besturings-Model/Fractals-Useful-Beauty.htm>

## Mandelbrotova množina

Mandelbrotova množina je jedním z nejznámějších fraktálů a stala se symbolem pro fraktální geometrii. Je definována jako množina všech bodů  $C$  v komplexní rovině, pro které rovnice  $Z_{n+1} = Z_n + C$  nekonverguje k nekonečnu<sup>33</sup> (čísla  $Z$  i  $C$  jsou komplexní,  $Z_0 = 0$ ). Jednoduše se dá říci, že výsledek rovnice znovu dosazujeme za  $Z$  do původní rovnice. Pokud se výsledky postupně blíží k nekonečnu (rychle se zvětšují), není vybraný bod  $C$  součástí Mandelbrotovy množiny. Když ale hodnoty nebudou směřovat k nekonečnu, ale budou se blížit k nějakému bodu nebo se výsledky budou periodicky opakovat, bod  $C$  do Mandelbrotovy množiny patří. Jinými slovy, body patřící do Mandelbrotovy množiny při iteraci nikdy nepřekročí určitou vzdálenost od počátku (hranici). U Mandelbrotovy množiny je maximální vzdálenost od počátku 2, všechny vzdálenější body budou při iteraci rovnice vždy unikat k nekonečnu.<sup>34</sup> Vzdálenost bodu od počátku spočítáme jednoduše. Vezměme například komplexní číslo  $3 + 4i$ . Vzdálenost od počátku je přeponou pravoúhlého trojúhelníku, jehož odvěsny mají délky 3 a 4. Podle Pythagorovy věty  $a^2 + b^2 = c^2$  spočítáme délku přepony  $c$  jako  $\sqrt{a^2 + b^2}$ . V našem případě tedy bude vzdálenost od počátku  $\sqrt{3^2 + 4^2} = \sqrt{25} = 5$ .



Obr. 8: Vzdálenost bodu od počátku.

<sup>33</sup> DEVANEY, Robert L. Unveiling the Mandelbrot set. Plus.maths.org [online]. August 31, 2006 [cit. 2015-02-27]. Dostupné z: <http://plus.maths.org/content/unveiling-mandelbrot-set>

<sup>34</sup> ZELINKA, VČELARĚ a ČANDÍK. Fraktální geometrie: principy a aplikace. Praha 2006, strana 50

Nyní si ukážeme několik příkladů. Nejprve určíme bod  $C$ , pro který chceme zjistit, zda náleží Mandelbrotově množině.

$$C = 0,5 + 1,5i.$$

Na začátku bude vždy  $Z$  nulové.

$$Z_0 = 0.$$

V první iteraci bude  $Z_1 = Z_0^2 + C$ , tedy

$$Z_1 = 0^2 + (0,5 + 1,5i) = 0,5 + 1,5i.$$

Druhá iterace bude pro nás zajímavější.

$$\begin{aligned} Z_2 &= (0,5 + 1,5i)^2 + (0,5 + 1,5i) = \\ &= (0,5^2 + 2 \cdot 0,5 \cdot 1,5i + 1,5i^2) + (0,5 + 1,5i) = \\ &= 0,25 + 1,5i + 2,25i^2 + (0,5 + 1,5i). \end{aligned}$$

Dostali jsme druhou mocninu imaginárního  $i$ . Ve výsledku ale chceme mít jedno komplexní číslo, mimo jiné abychom mohli zjistit jeho vzdálenost od počátku. Co s tím?

Víme, že  $i$  je  $\sqrt{-1}$ . Pak tedy  $i^2$  bude  $(\sqrt{-1})^2 = -1$ . Zbývá nám dopočítat rovnici:

$$\begin{aligned} Z_2 &= 0,25 + 1,5i + 2,25 \cdot (-1) + 0,5 + 1,5i = \\ &= -1,5 + 3i. \end{aligned}$$

Vzdálenost tohoto bodu od počátku bude  $\sqrt{(-1,5)^2 + 3^2} \doteq 2,598$ . Vzdálenost je větší než 2, komplexní číslo  $0,5 + 1,5i$  tedy nebude patřit do Mandelbrotovy množiny a tzv. escape-time (doba úniku) bude dvě iterace. Kdybychom pokračovali dále, bude  $Z_3 = -6,25 - 4,5i$ , vzdálenost od počátku bude přibližně 6,427, ve čtvrté iteraci pak  $Z_4 = 19,3125 + 57,75i$  a vzdálenost od počátku přibližně 60,894. Vidíme, že vzdálenost od počátku opravdu roste čím dál tím více.

Zkusme další bod.

$$C = 0,1 + 0,2i$$

$$Z_0 = 0$$

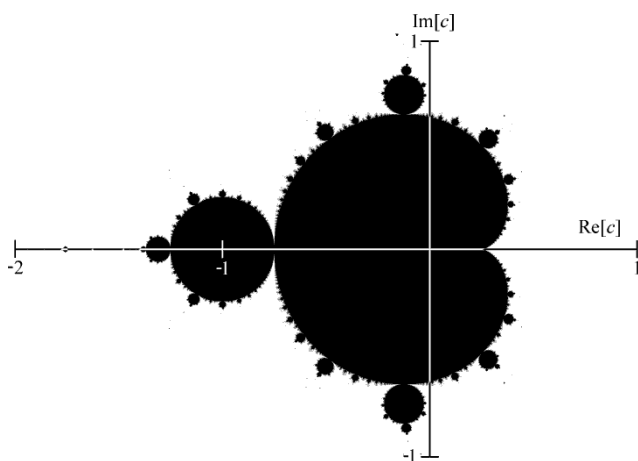
$$Z_1 = 0,1 + 0,2i$$

$$\begin{aligned} Z_2 &= 0,1^2 + 2 \cdot 0,1 \cdot 0,2i + (0,2i)^2 + (0,1 + 0,2i) = \\ &= 0,07 + 0,24i \end{aligned}$$

$$\begin{aligned} Z_3 &= 0,07^2 + 2 \cdot 0,07 \cdot 0,24i + (0,24i)^2 + (0,1 + 0,2i) = \\ &= 0,473 + 0,2336i \end{aligned}$$

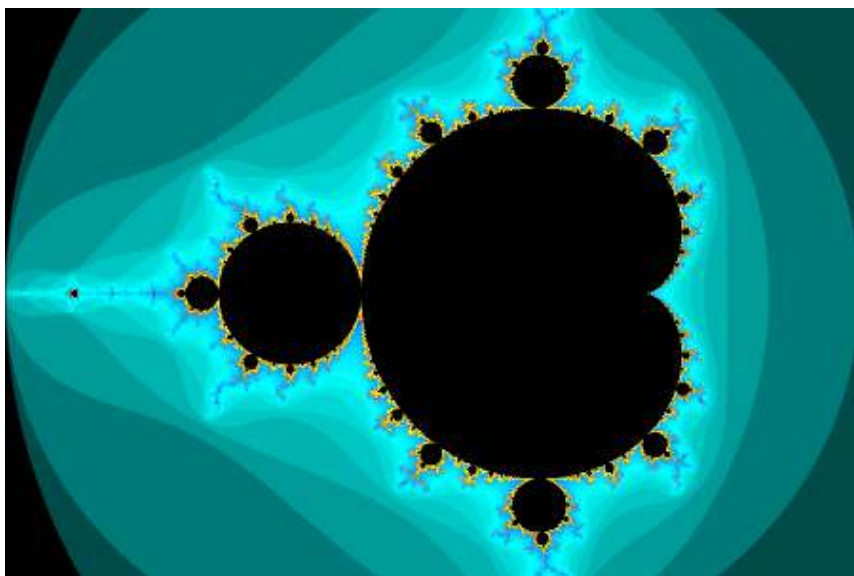
$$\begin{aligned} Z_4 &= 0,473^2 + 2 \cdot 0,473 \cdot 0,2336i + (0,2336i)^2 + (0,1 + 0,2i) = \\ &= 0,26916004 + 0,42i \end{aligned}$$

Takto bychom mohli počítat dále a dále. Zatím nemůžeme s jistotou říci, že při dalších iteracích bod nebude konvergovat k nekonečnu. Člověk ani počítač ale nedokážou počítat do nekonečna a proto se v praxi musí spokojit s odhadem a určit maximální počet iterací. Pokud se dosáhne maximálního počtu iterací a není překročena hraniční vzdálenost od počátku, bod je považován za součást Mandelbrotovy množiny. Čím více iterací provedeme, tím přesnější je odhad Mandelbrotovy množiny.



Obr. 9: Mandelbrotova množina.

Když vykreslujeme Mandelbrotovu množinu pomocí počítače, provede počítač výpočet pro každý bod (pixel) na obrázku. Body pak může obarvit podle toho, zda do množiny patří, nebo může třeba bodům mimo množinu přiřadit barvu podle toho, kolik iterací bylo třeba k překročení hraniční vzdálenosti.<sup>35</sup>

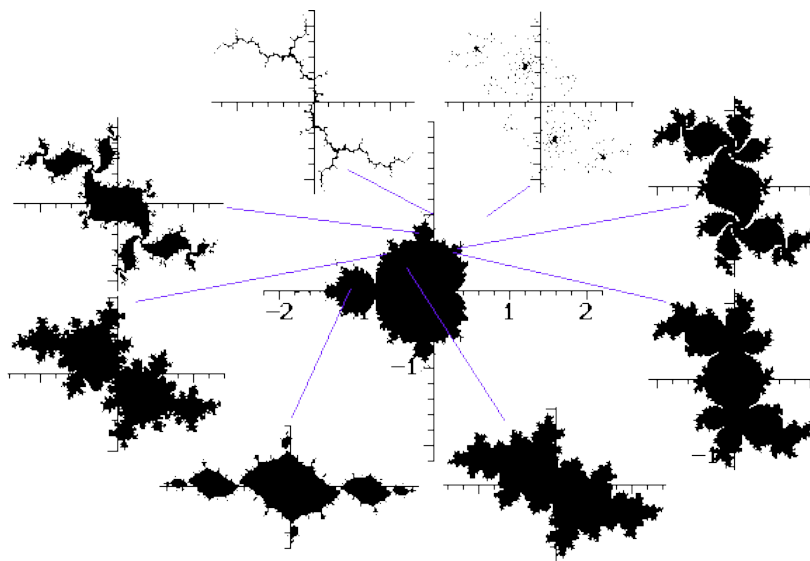


Obr. 10: Obarvení podle počtu iterací.

<sup>35</sup> PATRZALEK, Edyta. Introduction to Fractal Geometry. Fractal.org: Centre for Fractal Design and Consultancy [online]. [2008] [cit. 2015-02-27]. Dostupné z: <http://www.fractal.org/Bewustzijns-Besturings-Model/Fractals-Useful-Beauty.htm>

## Juliovy množiny

Dalšími TEA fraktály, které úzce souvisí s Mandelbrotovou množinou, jsou Juliovy množiny. Jejich rovnice je stejná jako pro Mandelbrotovu množinu, jen je  $c$  konstantní a pro každý bod se mění výchozí hodnoty  $Z_0$ . Pro různá  $c$  tak získáme odlišné Juliovy množiny. Je zajímavé, že pokud bod  $c$  je uvnitř Mandelbrotovy množiny, Juliova množina je tvořena pouze jednou, propojenou částí. Pokud vybereme bod mimo Mandelbrotovu množinu, získáme více oddělených částí. Mandelbrotovu množinu lze tím pádem definovat i jako množinu komplexních čísel  $C$ , pro která jsou odpovídající Juliovy množiny souvislé.<sup>36</sup>

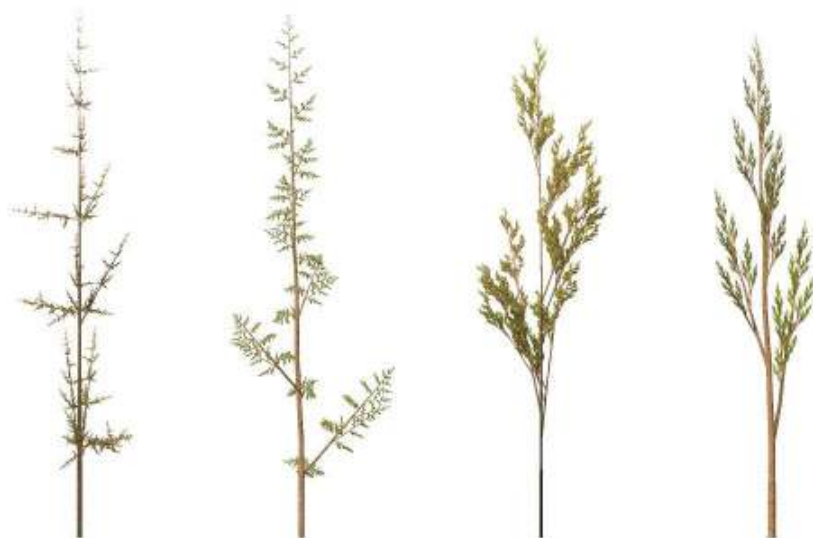


Obr. 11: Vztah Juliových množin a Mandelbrotovy množiny.

### 2.4.3.3 L-systémy

Lindenmayerovy systémy (zkráceně též L-systémy) jsou generovány pomocí přepisovacích pravidel, která určují čím (a za jakých podmínek) se má část fraktálu přepsat novou částí. Dokážou dobře modelovat větvičí se fraktály, jako jsou rostliny. Lze pomocí nich simulovat růst řas a chování buněk a dovedou generovat i některé klasické fraktály. Pomocí přepisovacích pravidel lze rostliny nechat „růst“, lze určit, kdy se má stoněk rozvětvit – rovný stoněk je nahrazen („přepsán“) rozvětveným. L-systémům bude ještě věnována celá kapitola mé ročníkové práce.

<sup>36</sup> PATRZALEK, Edyta. Introduction to Fractal Geometry. Fractal.org: Centre for Fractal Design and Consultancy [online]. [2008] [cit. 2015-02-27]. Dostupné z: <http://www.fractal.org/Bewustzijns-Besturings-Model/Fractals-Useful-Beauty.htm>



Obr. 12: Travniny vytvořené pomocí L-systémů.

#### 2.4.3.4 Náhodné fraktály, podivné atraktory a další

Náhodné fraktály jsou vytvářeny pomocí stochastických pravidel. Jejich chování je ovlivněno náhodou, často ale s určitým pravděpodobnostním rozložením. Patří sem třeba Brownův pohyb (náhodný pohyb např. částice pylu ve vodě). I když je pohyb náhodný, má mnohé fraktální vlastnosti, má fraktální dimenzi, je nekonečně podrobný atd. Náhodné fraktály jsou používány například i při vytváření počítačově generovaných krajin.

Podivné atraktory se generují několika funkcemi, takzvanými atraktory, které k sobě body „přitahují“. Charakteristické pro ně je, že i malá změna ve výchozích podmínkách může způsobit, že fraktál bude vypadat úplně jinak.<sup>37 38</sup>

---

<sup>37</sup> Strange Attractors. FractalFoundation.org [online]. © 2013 [cit. 2015-02-27]. Dostupné z: <http://fractalfoundation.org/OFC/OFC-7-1.html>

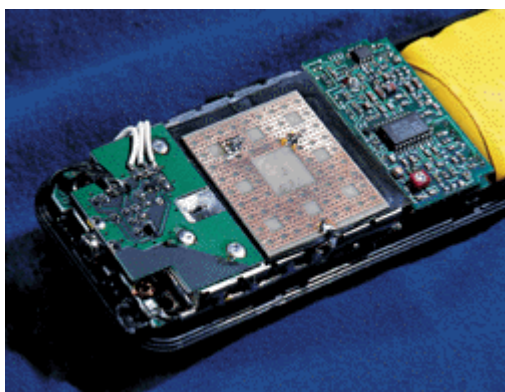
<sup>38</sup> Attractor. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: <http://en.wikipedia.org/wiki/Attractor>

## 2.5 Využití

Nejenže lze pomocí fraktálů vytvářet pěkné obrázky, existuje i široká škála využití v reálném životě. Nejspíše si ani neuvědomujeme, že v kapse s sebou neustále nosíme zařízení, která při své nejdůležitější funkci využívají fraktálové útvary.

### 2.5.1 Fraktálové antény

Od doby, co bylo vynalezeno rádio, řešili lidé jak dosáhnout co nejlepšího signálu. Obvykle to znamenalo zvětšit anténu. Tento problém byl vyřešen s pomocí fraktálů. Když se antény zakřivily do tvaru připomínajícím fraktál, mohly být mnohem delší při stejné velikosti. Namísto neustálého zvětšování stačilo ohnout drát na ještě menší úseky a drát tak pouze vyplňoval větší část plochy.<sup>39 40</sup>



Obr. 13: Fraktálová anténa v mobilním telefonu.

### 2.5.2 Fraktálová komprese

Fraktálová komprese obrázků využívá faktu, že jdou najít fraktály téměř všude, tedy i například na fotografii. Algoritmus rozdělí obrázek na části a snaží se najít části co nejpodobnější. Výhodou fraktálové komprese je, že se obrázky dají libovolně zvětšovat s mnohem menší ztrátou kvality, než je tomu u klasických formátů. To lze využít i pro dodatečné zvětšení fotografie. Nevýhodou oproti klasickým formátům je trochu delší doba komprese.<sup>41 42</sup>

---

<sup>39</sup> FRAME, Michael, Benoit MANDELBROT a Nial NEGER. Fractal Antennas. MANDELBROT. Yale University [online]. © 2015 [cit. 2015-02-27]. Dostupné z:

<http://classes.yale.edu/fractals/panorama/ManuFractals/FractalAntennas/FractalAntennas.html>

<sup>40</sup> Fractal Applications. FractalFoundation.org [online]. © 2013 [cit. 2015-02-27]. Dostupné z:

<http://fractalfoundation.org/OFC/OFC-12-2.html>

<sup>41</sup> GUPTA, Ashish. Fractal Image Compression. Digital Image Analysis [online]. [2003] [cit. 2015-02-27].

Dostupné z:

[http://www.cs.northwestern.edu/~agupta/\\_projects/image\\_processing/web/FractalImageCompression/](http://www.cs.northwestern.edu/~agupta/_projects/image_processing/web/FractalImageCompression/)

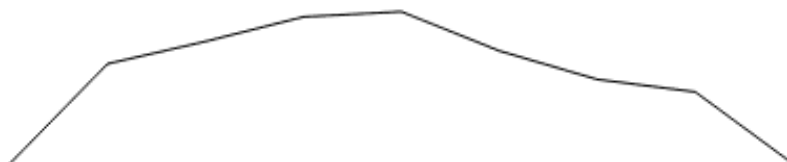
### 2.5.3 Počítačově vytvořené krajiny

Počítačové generování krajin je dnes ve filmovém i videoherním průmyslu zcela běžné a ušetří grafikům stovky hodin práce. Pomocí fraktálů lze vytvářet modely terénu o nekonečné podrobnosti – hornaté i kopcovité krajiny, řeky, ostrovy, oblohu, stromy, traviny, neznámé planety a tak dále.



Obr. 14: Tato krajina nikdy neexistovala.

Popíši zde nejjednodušší variantu pro vytvoření náhodného modelu krajiny, pro jednoduchost ve 2D. Začneme přímkou. Tu rozdělíme na polovinu a prostřední bod posuneme o náhodnou vzdálenost směrem nahoru nebo dolů. Tento postup opakujeme pro každou z nově vzniklých úseček. Různě velikým rozsahem náhodné vzdálenosti pro posunutí bodu můžeme dosáhnout různé kvality krajiny – můžeme tím ovlivnit, zda bude krajina spíše kopcovitá nebo hornatá.<sup>43</sup>



Obr. 15: Takto by mohla vypadat třetí iterace.

<sup>42</sup> ZELINKA, VČELAŘ a ČANDÍK. Fraktální geometrie: principy a aplikace. Praha 2006, strana 103-115

<sup>43</sup> Generating Random Fractal Terrain. Game Programmer [online]. © 1996, 1997 [cit. 2015-02-27]. Dostupné z: <http://www.gameprogrammer.com/fractal.html>



#### 2.5.4 Medicína

V medicíně se využití fraktálů teprve začíná rozmáhat. Vyvíjí se například postup, pomocí kterého by šla rakovina odhalit mnohem dříve. Dnešní technologie jako CT a MRI umožňuje pořizování velmi podrobných snímků. Problémem je, že pokud má člověk odhalit rakovinu ještě ve velmi brzkém stádiu, kdy je nádor velmi malý, musí při vysokém rozlišení analyzovat velké množství dat. Řešením je tuto úlohu přenechat počítači, který je mnohokrát rychlejší. Abychom dokázali počítači popsat rozdíl mezi zdravou a nezdravou tkání, můžeme využít toho, že nádor má obvykle vyšší fraktální dimenzi než zdravá tkáň.<sup>44</sup>

#### 2.5.5 Další využití

Dále se fraktály využívají kupříkladu k analýze cen na trhu, při výrobě lan, která dokážou udržet i několikatunovou konstrukci, jakou je most, jako potisk na trička, k měření množství vstřebaného oxidu uhličitého v lesích, při plánování měst a dopravy a k nespočetným dalším účelům.

---

<sup>44</sup> Fractal Applications. FractalFoundation.org [online]. © 2013 [cit. 2015-02-27]. Dostupné z: <http://fractalfoundation.org/OFC/OFC-12-4.html>

## 2.6 Lindenmayerovy systémy

Lindenmayerovy systémy, nazývané též zkráceně L-systémy byly objeveny v roce 1968 maďarsko-nizozemským biologem Aristidem Lindenmayerem.<sup>45</sup> Lindenmayer se v té době zabýval studiem růstu různých druhů řas a L-systémy měly původně sloužit pouze k modelování růstu rostlin na úrovni buněk. Později byly rozšířeny i na složitější rostlinné struktury a ke generování i jiných než rostlinných fraktálních struktur.<sup>46</sup>

### 2.6.1 Přepisování, iterace

Při generování využívají L-systémy opakované nahrazování jednoduchých útvarů pomocí množiny přepisovacích pravidel. Nejčastěji jsou Lindenmayerovy systémy reprezentovány pomocí řetězců znaků. Na začátku máme výchozí řetězec, tzv. axiom. Přepisovací pravidla nám říkají, jakými znaky je nahrazen znak z původního řetězce.

Uvedme si příklad. Výchozím axiomem bude  $A$ . Přepisovací pravidlo  $A \rightarrow AB$  říká, že každý znak  $A$  nahradíme znaky  $AB$ . Druhé pravidlo  $B \rightarrow A$  nahradí písmeno  $B$  písmenem  $A$ . Důležité je, že se pravidla aplikují současně. Jinak by druhé pravidlo způsobilo, že ve výsledku budou pouze samá  $A$ .<sup>47</sup>

V první iteraci z původního  $A$  získáme  $AB$ . Druhé pravidlo zatím žádný znak nepřepíše, jelikož v řetězci zatím není přítomno žádné  $B$ , které by se dalo nahradit. V druhé iteraci se z řetězce  $AB$  stane  $ABA$ .  $A$  bylo nahrazeno  $AB$  a z  $B$  se stalo  $A$ . Dalším opakováním bychom získali  $ABAAB$ , pak  $ABAABABA$  a tak dále.

### 2.6.2 Želví grafika

Nejčastější způsob interpretace symbolů vygenerovaných pomocí přepisovacích pravidel je tzv. „želví grafika“. Každý ze symbolů je pro pomyslnou želvu určitým příkazem, jako je například „jdi o krok dopředu“ nebo „otoč se doprava“. Želva s sebou má štětec, kterým zaznamenává svou cestu. Stav želvy je dán pozicí želvy a směrem, kterým se želva dívá. Lze zapsat jako uspořádanou trojici  $(x, y, \alpha)$ , která udává souřadnice želvy v rovině a úhel jejího otočení vzhledem k ose  $x$ . Zde jsou základní povely, které želva zná:

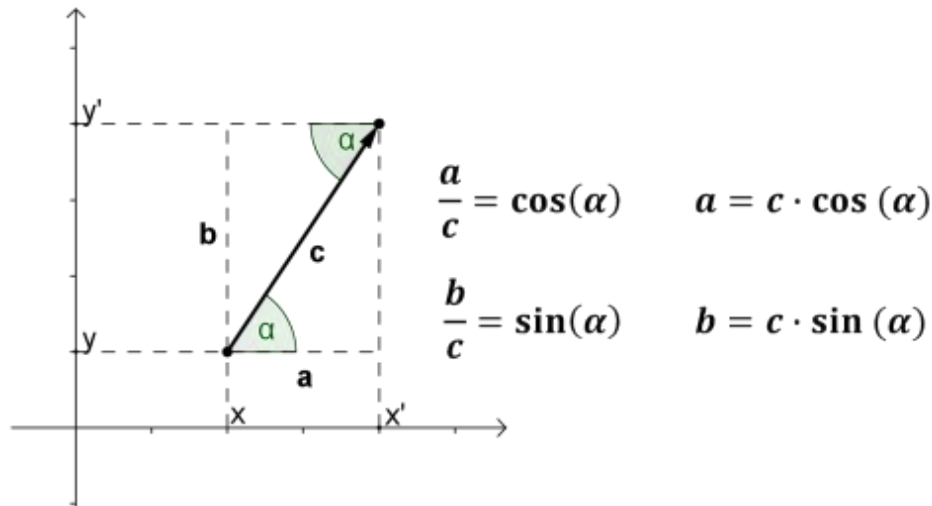
---

<sup>45</sup> L-system. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: <http://en.wikipedia.org/wiki/L-system>

<sup>46</sup> PRUSINKIEWICZ a LINDENMAYER. The algorithmic beauty of plants. New York 1990, strana 1

<sup>47</sup> PRUSINKIEWICZ a LINDENMAYER. The algorithmic beauty of plants. New York 1990, strana 3-4

- **F** Písmeno **F**, nebo i jiné velké písmeno abecedy říká želvě „jdi o krok dopředu“, přičemž želva za sebou štětcem zanechává stopu, tj. nakreslí čáru od její původní do nové pozice. Nová pozice želvy  $(x', y', \alpha')$  je nyní  $(x + c \cdot \cos \alpha, y + c \cdot \sin \alpha, \alpha)$ , kde  $c$  je délka kroku želvy.



Obr. 16: Výpočet nové pozice želvy.

- **f** Malé písmeno abecedy posune želvu o krok dopředu, jen za sebou želva tentokrát nebude kreslit čáru.
- **+** Otočí želvu proti směru hodinových ručiček o úhel  $\delta$ . Nový stav želvy tedy bude  $(x, y, \alpha + \delta)$ .
- **-** Otočí želvu po směru hodinových ručiček. Nový stav želvy bude  $(x, y, \alpha - \delta)$ .<sup>48</sup>

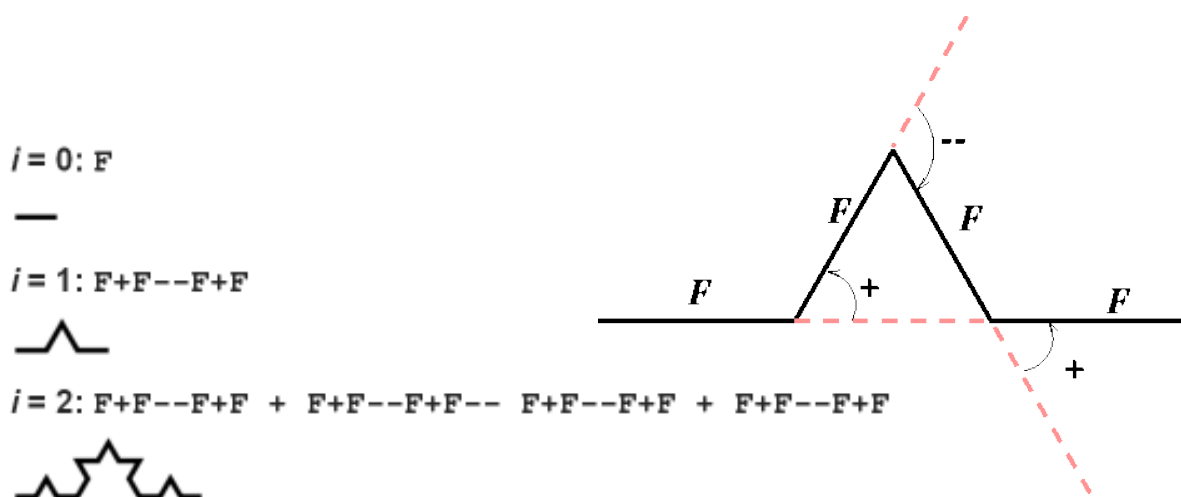
Ukažme si, jak napsat želví grafikou třeba písmeno F. Nejprve půjdeme dva kroky rovně nahoru (**FF**). Poté se otočíme doprava (**-**) a uděláme jeden krok (**F**), úhel otáčení je  $90^\circ$ . Znovu se otočíme po směru hodinových ručiček (**-**), tentokrát však čáru kreslit nebudeme (**f**). Zbývá nám otočka po směru hodinových ručiček (**-**) a poslední krok (**F**) a naše F je hotové. Celou cestu želvy zapíšeme jako **FF - F - f - F**.

<sup>48</sup> PRUSINKIEWICZ a LINDENMAYER. The algorithmic beauty of plants. New York 1990, strana 6-7

### 2.6.3 Souvislé L-systemy

Nyní, když spojíme dohromady generování pomocí prepisovacích pravidel a dokážeme symboly interpretovat pomocí želví grafiky, můžeme začít modelovat fraktály.

Axiomem pro generování Kochovy křivky je jediný krok  $F$ . Na něj aplikujeme prepisovací pravidlo  $F \rightarrow F + F - -F + F$ . Úhel otočení je  $60^\circ$ . Prepisovací pravidlo říká, že nahrazujeme každou úsečku (protože jiné než  $F$  v tomto případě nemáme) krokem rovně, poté dalším o  $60^\circ$  otočeným, otočíme se o  $120^\circ$  doprava, nakreslíme další úsečku a poslední nakreslíme otočenou o  $60^\circ$  doleva. Můžeme vidět, že L-systemy dokážou popsat postup vytváření Kochovy křivky vlastně velmi podobným způsobem, jak jsme si popisovali v jedné z prvních kapitol.<sup>49 50</sup>



Obr. 17: Kochova křivka jako L-systém.

<sup>49</sup> PRUSINKIEWICZ a LINDENMAYER. The algorithmic beauty of plants. New York 1990, strana 11-17

<sup>50</sup> L-systém. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: <http://cs.wikipedia.org/wiki/L-syst%C3%A9m>

## 2.6.4 Větvení

Nyní již dokážeme vytvářet pomocí L-systémů fraktály, které jsou tvořeny jednou souvislou linkou. Kdybychom ale chtěli vytvářet větvící se struktury, museli bychom nejprve vytvořit jednu větev, poté se vrátit na místo větvení a odtud pak kreslit větev další. Člověk je naštěstí líný a proto nám tuto práci usnadnil. K želvím příkazům nám přibudou další dva znaky.

- [ Hranatá závorka říká želvě „zapamatuj si aktuální pozici“, přesněji „ulož svůj stav na zásobník“. Zásobník si můžeme představit například jako sloupek talířů – talíř, který jsme položili na hromadu jako poslední, z ní vezmeme jako první.
- ] Uzavírající hranatá závorka znamená pro želvu „vrať se na svou poslední zapamatovanou pozici“, neboli „načti stav ze zásobníku“.<sup>51</sup>

Tím můžeme jednoduše vytvářet větvící se struktury. Jednoduše uzavřeme větev do hranatých závorek. Tím se před vytváření větve uloží stav želvy na zásobník a po dokončení větve se může zase vrátit zpět.

Jako příklad větvící se struktury si ukážeme jednoduchý strom. Strom roste tím způsobem, že v každé iteraci vyroste větev na dvojnásobnou délku a na konci se rozvětví. Rozvětvují se vždy jen krajní (mladé) větve. Nekrajní (starší) části větve pouze rostou.

Mladé větve budou reprezentovány písmenem  $M$ , staré větve písmenem  $S$ . Strom začne růst z jediné mladé větvičky  $M$ , která je axiomem. V každé iteraci se z mladé větve stane stará a vyroste z ní jedna mladá větev doleva (úhel otočení je  $45^\circ$ ) a druhá doprava  $M \rightarrow S[+M][-M]$ . Růst starších větví zapíšeme jako  $S \rightarrow SS$ .



Obr. 18: Růst stromu.

<sup>51</sup> PRUSINKIEWICZ a LINDENMAYER. The algorithmic beauty of plants. New York 1990, strana 24

## 2.6.5 Další rozšíření

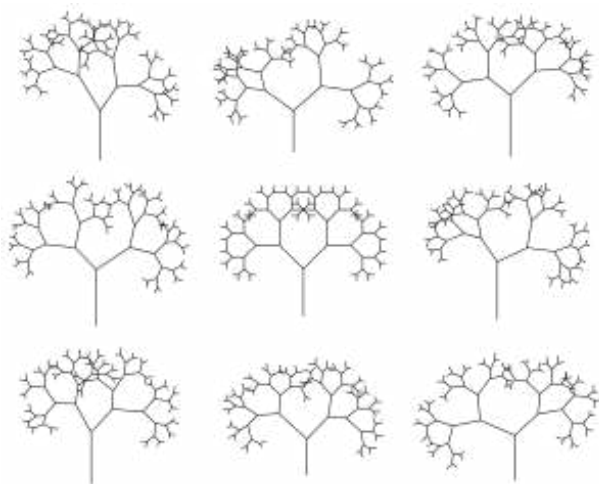
### 2.6.5.1 Vykreslování ve 3D

Výhodou L-systémů je jejich snadná rozšiřitelnost o nové příkazy. Tak lze jednoduše příkaz |, který znamená „čelem vzad“ neboli otočení o 180°. Dále lze L-systémy rozšířit i pro generování fraktálu v prostoru.

- ^ „Otoč se nahoru“ o předem stanovený úhel.
- & Otočení dolů o daný úhel.
- / Otočení dopředu podél vlastní osy (želva by nezměnila směr, kterým by se dívala, ale otočila by se např. na stranu nebo na záda).
- \ Otočení podél vlastní osy doprava (po směru hodinových ručiček při pohledu zezadu na želvu)<sup>52</sup>

### 2.6.5.2 Stochastické L-systémy

Stochastické L-systémy přidávají do přepisování anebo vykreslování prvek náhody. Nejjednodušším způsobem je při vykreslování náhodně o trochu měnit úhel a délku kroku (např. o  $\pm 20\%$ ). Vygenerované symboly tak budou vždy stejné, ale výsledný obraz bude pokaždé o něco jiný.<sup>53 54</sup>



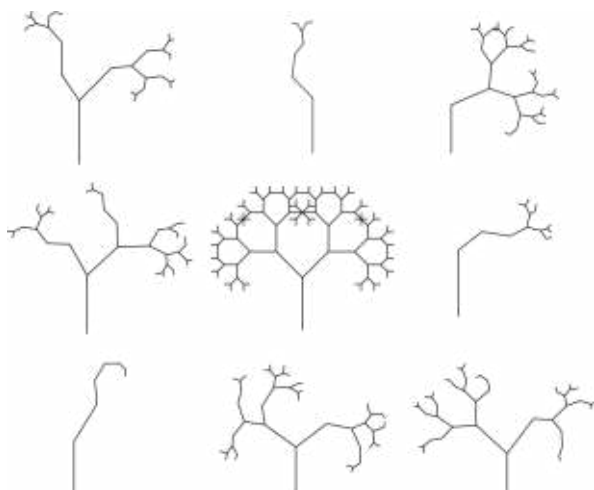
Obr. 19: Náhodná délka a úhel.

<sup>52</sup> PRUSINKIEWICZ a LINDENMAYER. The algorithmic beauty of plants. New York 1990, strana 18-21

<sup>53</sup> PRUSINKIEWICZ a LINDENMAYER. The algorithmic beauty of plants. New York 1990, strana 28

<sup>54</sup> L-systém. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: <http://cs.wikipedia.org/wiki/L-syst%C3%A9m>

Další možností je přidat náhodu již při samotném generování, kdy každému pravidlu přiřadíme určitou váhu, která určuje, s jakou pravděpodobností vůči ostatním se pravidlo vybere. Pravděpodobnostní poměr zapisujeme nad šipku v daném přepisovacím pravidle.<sup>55 56</sup>



Obr. 20: Náhodné přepisovací pravidlo.

### 2.6.5.3 Parametrické L-systémy

Parametrické L-systémy přidávají možnost přidat ke znakům v L-systému parametry, které mohou být při interpretaci želví grafikou použity například pro určení délky čáry, její tloušťky, úhlu rotace a podobně. Jednotlivé parametry zapisujeme za daný znak do kulatých závorek, oddělujeme je čárkou. Parametrem může být číslo, proměnná, nebo matematický výraz.<sup>57</sup>

Přepisovací pravidlo může vypadat například takto:  $F(x, y) \rightarrow G(x/2) F(8, y)$ . Toto přepisovací pravidlo by například z axiomu  $F(4, 3)$  v první iteraci vytvořilo  $G(2) F(8, 3)$ . V další iteraci bychom získali  $G(2) G(4) F(8, 3)$ .

Zvykem bývá při interpretaci brát první parametr jako délku kroku, u znaků + a – jako úhel rotace. Interpretace dalších parametrů je již různá. Pokud parametr nezapišeme, použije se výchozí hodnota, která by se použila i u bezparametrických L-systémů.<sup>58</sup>

<sup>55</sup> PRUSINKIEWICZ a LINDENMAYER. The algorithmic beauty of plants. New York 1990, strana 28

<sup>56</sup> L-systém. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: <http://cs.wikipedia.org/wiki/L-syst%C3%A9m>

<sup>57</sup> PRUSINKIEWICZ a LINDENMAYER. The algorithmic beauty of plants. New York 1990, strana 41-42

<sup>58</sup> PRUSINKIEWICZ a LINDENMAYER. The algorithmic beauty of plants. New York 1990, strana 46

#### 2.6.5.4 Podmíněné přepisování

Parametrické L-systémy lze ještě dále rozšířit o podmíněné přepisování. Pokud se podmínka vyhodnotí jako pravdivá, přepisovací pravidlo se použije, v opačném případě se přepisovací pravidlo přeskočí a zůstane původní symbol. Podmínky se zapisují k symbolu před šipkou. Za symbolem (a jeho parametry) následuje dvojtečka, poté podmínka. V podmínkách můžeme kromě aritmetických operátorů a porovnávacích operátorů (<, >, =) použít i logické operátory &, | a ! (and, or a not).<sup>59 60</sup>

Přepisovací pravidlo s podmínkou může vypadat například následovně:  $F(x, y): x \geq 3 \ \& \ y \neq 0 \rightarrow GG$ . Pravidlo lze přečíst jako: „pokud  $x$  je větší nebo rovno 3 a zároveň  $y$  není rovno 0, tak přepiš  $F(x, y)$  za  $GG$ “.

#### 2.6.5.5 Kontextové L-systémy

Kontextové L-systémy nám umožňují při označení znaku, který má být přepsán brát ohledy i na jeho sousedy. Na levého souseda vybraného znaku ukazujeme pomocí značky <, na pravého pomocí >.<sup>61</sup>

Pravidlo kontextového systému by mohlo vypadat třeba takto:  $F > G \rightarrow FGF$ . Toto pravidlo přepíše všechna  $F$  následovaná písmenem  $G$ .

#### 2.6.5.6 Další možná rozšíření

L-systémy jdou samozřejmě rozšiřovat dále a dále. Lze například přidat simulaci gravitace, která jako i v přírodě táhne větve stromů směrem dolů. Nebo lze L-systémy rozšířit například tak, aby reagovaly na prostředí, například dva stromy blízko u sebe by rostly méně na sobě přivrácené straně. Také lze přidat vykreslování obrázků, nebo ve 3D modelů, díky kterým můžeme třeba na konec stonku jednoduše přidat květ nebo list. Záleží na konkrétních potřebách, o co člověk L-systémy dále obohatí.

---

<sup>59</sup> PRUSINKIEWICZ a LINDENMAYER. The algorithmic beauty of plants. New York 1990, strana 41-42

<sup>60</sup> L-systém. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: <http://cs.wikipedia.org/wiki/L-syst%C3%A9m>

<sup>61</sup> PRUSINKIEWICZ a LINDENMAYER. The algorithmic beauty of plants. New York 1990, strana 30





## 4. Umělecká část

U umělecké části jsem dlouho nevěděl, co bych vlastně mohl udělat. Kresba některých klasických fraktálů mi přišla moc jednoduchá. Naopak třeba malba fraktálů TEA by byla velmi složitá. Stejně tak i kreslení vlastních vymyšlených fraktálů je moc jednoduché a malba fraktálového charakteru inspirovaná komplexními fraktály zase moc složitá. Navíc jsem chtěl tvořit spíše něco konkrétnějšího než abstraktní obrazce. Došel jsem tedy k tomu, že nebudu malovat přímo fraktály, ale že namaluji obraz, který je fraktály inspirován. Obzvláště mne zajímají přírodní fraktály a fraktálové rostliny, došel jsem tedy k tomu, že namaluji obraz, ve kterém budou rostliny fraktálového charakteru. Začal jsem nejprve několika náčrty a poté větší kresbou uhlím. Pak jsem podle vzorů namaloval na papír formátu A3 podklad akvarelovými barvami. Nakonec jsem pastelkami namaloval strom a doladil několik detailů.



Obr. 22: Dokončená umělecká část.

## 5. Závěr

V ročníkové práci jsme si řekli o tom, co je to fraktál a popsali jeho typické vlastnosti. Pak jsme se podívali na život Benoita B. Mandelbrota – člověka, který je po právu nazýván otcem fraktální geometrie, a který velmi přispěl k rozvoji fraktální geometrie. Vysvětlili jsme si, jakými různými způsoby lze fraktály generovat a jaké využití nacházejí v každodenním životě. Trochu podrobněji jsme se pak podívali na skupinu fraktálů zvaných Lindenmayerovy systémy, pomocí kterých lze především dobře popisovat rostliny.

Doufám, že se mi podařilo čtenáři nejen sdělit spoustu informací z oblasti fraktální geometrie, ale hlavně ukázat, že fraktály nejsou jen zajímavé matematické množiny nebo barevné obrazce, ale že dokážou pomocí jednoduchých pravidel popisovat i tak složité útvary, jako například rostliny, které se vyskytují v přírodě všude kolem nás a že fraktály nacházejí mnohem větší praktické využití, než si možná někteří mysleli.

Mně dala ročníková práce opravdu mnoho. Měl jsem možnost věnovat se fraktálům, které jsem předtím znal pouze okrajově více do hloubky, obzvláště mne zaujaly fraktály v přírodě a jejich modelování. Trochu jsem se zdokonalil i v praktických dovednostech programování a hlavně jsem se naučil, jak se má psát odborná práce a jak se pracuje se zdroji a odbornou literaturou. Pro mě to nejdůležitější jsou ale zkušenosti, které jsem díky ročníkové práci získal. Ze začátku jsem vůbec nekomunikoval s vedoucím a psaní ročníkové práce jsem (především ze strachu, že mi to nepůjde) pořád odkládal. Ročníkovou práci jsem začal psát až ke konci prosince a bylo to poprvé, co jsem opravdu narazil díky tomu, že něco dělám na poslední chvíli. Nakonec mě psaní ročníkové práce bavilo, a měl jsem chuť jí dále a dále rozšiřovat a vylepšovat. I praktické části bych se vlastně chtěl věnovat ještě více a program dále rozšířit.

## Seznam použitých pramenů a literatury

### Použitá literatura

- MANDELBROT, Benoit B. *The fractal geometry of nature*. Vyd. 3. New York: Freeman and company, 1983, 468 s. ISBN 07-167-1186-9.
- PRUSINKIEWICZ, Przemyslaw a Aristid LINDENMAYER. *The algorithmic beauty of plants*. New York: Springer-Verlag, 1990, xii, 228 p. ISBN 35-409-7297-8.  
Dostupné z: <http://algorithmicbotany.org/papers/abop/abop.pdf>
- ZELINKA, Ivan, František VČELARĚ a Marek ČANDÍK. *Fraktální geometrie: principy a aplikace*. 1. vyd. Praha: BEN, 2006, 159 s. ISBN 80-730-0191-8.

### Internetové zdroje

- Attractor. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: <http://en.wikipedia.org/wiki/Attractor>
- Benoit Mandelbrot. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: [http://en.wikipedia.org/wiki/Benoit\\_Mandelbrot](http://en.wikipedia.org/wiki/Benoit_Mandelbrot)
- Benoit Mandelbrot. *NNDB* [online]. © 2014 [cit. 2015-02-27]. Dostupné z: <http://www.nndb.com/people/752/000022686/>
- DEVANEY, Robert L. Unveiling the Mandelbrot set. *Plus.maths.org* [online]. August 31, 2006 [cit. 2015-02-27]. Dostupné z: <http://plus.maths.org/content/unveiling-mandelbrot-set>
- DRAVES, Scott a Erik RECKASE. The Fractal Flame Algorithm. RECKASE, Erik. *The Flame Algorithm* [online]. September 2003, November 2008 [cit. 2015-02-27]. Dostupné z: [http://flam3.com/flame\\_draves.pdf](http://flam3.com/flame_draves.pdf)
- FIŠER, Marek. *L-systems online* [online]. Prague, 2012 [cit. 2015-02-27]. Dostupné z: <http://www.malsys.cz/Download/MarekFiser-LsystemsOnline.150208.v101.pdf>.  
Bakalářská práce. Charles University in Prague.
- Fractal antenna. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: [http://en.wikipedia.org/wiki/Fractal\\_antenna](http://en.wikipedia.org/wiki/Fractal_antenna)

- Fractal Applications. *FractalFoundation.org* [online]. © 2013 [cit. 2015-02-27].  
Dostupné z: <http://fractalfoundation.org/OFC/OFC-12-4.html>
- Fractal Applications. *FractalFoundation.org* [online]. © 2013 [cit. 2015-02-27].  
Dostupné z: <http://fractalfoundation.org/OFC/OFC-12-2.html>
- Fractal compression. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: [http://en.wikipedia.org/wiki/Fractal\\_compression](http://en.wikipedia.org/wiki/Fractal_compression)
- Fractal dimension. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: [http://en.wikipedia.org/wiki/Fractal\\_dimension](http://en.wikipedia.org/wiki/Fractal_dimension)
- Fractal Geometry. *IBM United States* [online]. May 18, 2011 [cit. 2015-02-27].  
Dostupné z: <http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/fractal/>
- Fractal landscape. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: [http://en.wikipedia.org/wiki/Fractal\\_landscape](http://en.wikipedia.org/wiki/Fractal_landscape)
- Fractals in Science, Engineering and Finance (Roughness and Beauty). *MIT Video* [online]. 2001-11-27 [cit. 2015-02-27]. Dostupné z: <http://video.mit.edu/watch/fractals-in-science-engineering-and-finance-roughness-and-beauty-9893/>
- Fractals. *The Math Images Project* [online]. July 22, 2008, 19 December 2012 [cit. 2015-02-27]. Dostupné z: <http://mathforum.org/mathimages/index.php/Field:Fractals>
- Fraktál. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: <http://cs.wikipedia.org/wiki/Frakt%C3%A1l>
- FRAME, Michael, Benoit MANDELBROT a Nial NEGER. Fractal Antennas. MANDELBROT. *Yale University* [online]. © 2015 [cit. 2015-02-27]. Dostupné z: <http://classes.yale.edu/fractals/panorama/ManuFractals/FractalAntennas/FractalAntennas.html>
- FRAME, Michael. Benoit B. Mandelbrot: A Biographical Memoir. *National Academy of Sciences* [online]. © 2014 [cit. 2015-02-27]. Dostupné z:

- <http://www.nasonline.org/publications/biographical-memoirs/memoir-pdfs/mandelbrot-benoit.pdf>
- Generating Random Fractal Terrain. *Game Programmer* [online]. © 1996, 1997 [cit. 2015-02-27]. Dostupné z: <http://www.gameprogrammer.com/fractal.html>
  - GREGOROVÁ, Dagmar. Stal jsem se konstruktérem geometrie. *OSEL.cz* [online]. 19.10.2010 [cit. 2015-02-27]. Dostupné z: <http://www.osel.cz/index.php?clanek=5346>
  - Guardian obituary. *The MacTutor History of Mathematics archive* [online]. 18 October 2010 [cit. 2015-02-27]. Dostupné z: [http://www-history.mcs.st-and.ac.uk/Obits2/Mandelbrot\\_Guardian.html](http://www-history.mcs.st-and.ac.uk/Obits2/Mandelbrot_Guardian.html)
  - GUPTA, Ashish. Fractal Image Compression. *Digital Image Analysis* [online]. [2003] [cit. 2015-02-27]. Dostupné z: [http://www.cs.northwestern.edu/~agupta/\\_projects/image\\_processing/web/FractalImageCompression/](http://www.cs.northwestern.edu/~agupta/_projects/image_processing/web/FractalImageCompression/)
  - HOTAŘ, Vlastimil. Fraktální geometrie a fraktály. *Fraktální geometrie* [online]. [2006] [cit. 2015-02-27]. Dostupné z: <http://www.ksr.tul.cz/fraktaly/geometrie.html>
  - Iterated function system. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: [http://en.wikipedia.org/wiki/Iterated\\_function\\_system](http://en.wikipedia.org/wiki/Iterated_function_system)
  - JABLONSKI, Adrian. Grundlagen der fraktalen Geometrie mit iterierten Funktionensystemen (IFS). *Quadsoft.org* [online]. Juli 2011 [cit. 2015-02-27]. Dostupné z: <http://quadsoft.org/fraktale/>
  - L-system. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: <http://en.wikipedia.org/wiki/L-system>
  - L-systém. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: <http://cs.wikipedia.org/wiki/L-syst%C3%A9m>

- Mandelbrot set. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-02-27]. Dostupné z: [http://en.wikipedia.org/wiki/Mandelbrot\\_set](http://en.wikipedia.org/wiki/Mandelbrot_set)
- O'CONNOR, J. J. a E. F. ROBERTSON. Mandelbrot biography. *The MacTutor History of Mathematics archive* [online]. © July 1999, 2014 [cit. 2015-02-27]. Dostupné z: <http://www-history.mcs.st-and.ac.uk/Biographies/Mandelbrot.html>
- PATRZALEK, Edyta. Fractals: Useful Beauty. *Elika Kurniadi* [online]. October 10, 2011 [cit. 2015-02-27]. Dostupné z: <https://elidakurniadi.wordpress.com/2011/10/10/fractal-geometry/>
- PATRZALEK, Edyta. Introduction to Fractal Geometry. *Fractal.org: Centre for Fractal Design and Consultancy* [online]. [2008] [cit. 2015-02-27]. Dostupné z: <http://www.fractal.org/Bewustzijns-Besturings-Model/Fractals-Useful-Beauty.htm>
- Rotations: the how's and why's. *Pete's QBASIC / QuickBasic Site* [online]. [2004] [cit. 2015-02-27]. Dostupné z: <http://www.petesqbsite.com/sections/tutorials/tuts/relsoft3d/Chapter2/Chapter2.htm>
- SIXTA, Tomáš. Úvod do fraktálů a chaosu. *ITnetwork.cz* [online]. © 2015 [cit. 2015-02-27]. Dostupné z: <http://www.itnetwork.cz/fraktaly-a-chaos-eukleidovska-geometrie-uvod-svitani-chaosu>
- Strange Attractors. *FractalFoundation.org* [online]. © 2013 [cit. 2015-02-27]. Dostupné z: <http://fractalfoundation.org/OFC/OFC-7-1.html>
- Telegraph obituary. *The MacTutor History of Mathematics archive* [online]. © 2010 [cit. 2015-02-27]. Dostupné z: [http://www-history.mcs.st-and.ac.uk/Obits2/Mandelbrot\\_Telegraph.html](http://www-history.mcs.st-and.ac.uk/Obits2/Mandelbrot_Telegraph.html)
- WAHL, Bernt Rainer. Calculating Fractal Dimension. *Fractal Explorer* [online]. © 2014 [cit. 2015-02-27]. Dostupné z: [http://www.wahl.org/fe/HTML\\_version/link/FE4W/c4.htm](http://www.wahl.org/fe/HTML_version/link/FE4W/c4.htm)

## Seznam zdrojů ilustrací

- Obr. 1: Konstrukce Kochovy křivky. .... 10  
<http://quadsoft.org/fraktale/koch.png>; upraveno (14. 1. 2015)
- Obr. 2: Závislost měřítka a dimenze. .... 12  
<http://upload.wikimedia.org/wikipedia/commons/4/4d/Fractaldimensionexample.PNG>; upraveno (15. 3. 2015)
- Obr. 3: Konstrukce Sierpinského trojúhelníku. .... 13  
[http://upload.wikimedia.org/wikipedia/commons/0/05/Sierpinski\\_triangle\\_evolution.svg](http://upload.wikimedia.org/wikipedia/commons/0/05/Sierpinski_triangle_evolution.svg); upraveno (2. 3. 2015)
- Obr. 4: Sierpinského trojúhelník jako křivka. .... 13  
[http://upload.wikimedia.org/wikipedia/commons/a/ae/Arrowhead\\_curve\\_1\\_through\\_6.png](http://upload.wikimedia.org/wikipedia/commons/a/ae/Arrowhead_curve_1_through_6.png) (2. 3. 2015)
- Obr. 5: Šest iterací Kantorovy množiny. .... 13  
[http://upload.wikimedia.org/wikipedia/commons/5/56/Cantor\\_set\\_in\\_seven\\_iterations.svg](http://upload.wikimedia.org/wikipedia/commons/5/56/Cantor_set_in_seven_iterations.svg) (2. 3. 2015)
- Obr. 6: Křivka, která vyplňuje plochu. .... 14  
<https://cedevriginoushoplist.files.wordpress.com/2012/11/peano-curve.jpg>  
(2. 3. 2015)
- Obr. 8: Číslo  $1+2i$  v komplexní rovině. .... 16  
Vytvořeno v programu GeoGebra
- Obr. 9: Vzdálenost bodu od počátku. .... 17  
Vytvořeno v programu GeoGebra
- Obr. 10: Mandelbrotova množina. .... 19  
[http://upload.wikimedia.org/wikipedia/commons/5/56/Mandelset\\_hires.png](http://upload.wikimedia.org/wikipedia/commons/5/56/Mandelset_hires.png)  
(2. 3. 2015)
- Obr. 11: Obarvení podle počtu iterací. .... 19  
<http://rosettacode.org/mw/images/2/2a/Mandelbrot-Javascript.png>; upraveno  
(2. 3. 2015)
- Obr. 12: Vztah Juliových množin a Mandelbrotovy množiny. .... 20  
<http://ocw.upm.es/matematica-aplicada/introduccion-a-los-sistemas-dinamicos/contenidos/IMAGES/juliaenmandel.gif> (2. 3. 2015)
- Obr. 13: Travniny vytvořené pomocí L-systémů. .... 21



	<a href="http://upload.wikimedia.org/wikipedia/commons/a/af/Fractal_weeds.jpg">http://upload.wikimedia.org/wikipedia/commons/a/af/Fractal_weeds.jpg</a> (2. 3. 2015)	
Obr. 14:	Fraktálová anténa v mobilním telefonu. ....	22
	<a href="http://classes.yale.edu/fractals/panorama/ManuFractals/FractalAntennas/FracTenna1.gif">http://classes.yale.edu/fractals/panorama/ManuFractals/FractalAntennas/FracTenna1.gif</a> (24. 5. 2015)	
Obr. 15:	Tato krajina nikdy neexistovala. ....	23
	<a href="http://upload.wikimedia.org/wikipedia/commons/4/40/BlueRidgePastures.jpg">http://upload.wikimedia.org/wikipedia/commons/4/40/BlueRidgePastures.jpg</a> (24. 5. 2015)	
Obr. 16:	Takto by mohla vypadat třetí iterace. ....	23
	<a href="http://www.gameprogrammer.com/fractal/mpd3.gif">http://www.gameprogrammer.com/fractal/mpd3.gif</a> (24. 5. 2015)	
Obr. 17:	Výpočet nové pozice želvy..... Vytvořeno v programu GeoGebra	26
Obr. 18:	Kochova křivka jako L-systém.....	27
	<a href="http://w3-o.cs.hm.edu/~ruckert/compiler/www.math.okstate.edu/img109.gif">http://w3-o.cs.hm.edu/~ruckert/compiler/www.math.okstate.edu/img109.gif</a> (24. 1. 2015)	
Obr. 19:	Růst stromu. ....	28
	<a href="http://en.wikipedia.org/wiki/L-system#mediaviewer/File:Graftal0.png">http://en.wikipedia.org/wiki/L-system#mediaviewer/File:Graftal0.png</a> <a href="http://en.wikipedia.org/wiki/L-system#mediaviewer/File:Graftal1.png">http://en.wikipedia.org/wiki/L-system#mediaviewer/File:Graftal1.png</a> <a href="http://en.wikipedia.org/wiki/L-system#mediaviewer/File:Graftal2.png">http://en.wikipedia.org/wiki/L-system#mediaviewer/File:Graftal2.png</a> <a href="http://en.wikipedia.org/wiki/L-system#mediaviewer/File:Graftal3.png">http://en.wikipedia.org/wiki/L-system#mediaviewer/File:Graftal3.png</a> <a href="http://en.wikipedia.org/wiki/L-system#mediaviewer/File:Graftal4.png">http://en.wikipedia.org/wiki/L-system#mediaviewer/File:Graftal4.png</a> <a href="http://en.wikipedia.org/wiki/L-system#mediaviewer/File:Graftal7.png">http://en.wikipedia.org/wiki/L-system#mediaviewer/File:Graftal7.png</a> ; upraveno (25. 1. 2015)	
Obr. 20:	Náhodná délka a úhel.....	29
	<a href="http://cs.wikipedia.org/wiki/L-syst%C3%A9m#mediaviewer/File:RandomizedPythagorasLsystem.svg">http://cs.wikipedia.org/wiki/L-syst%C3%A9m#mediaviewer/File:RandomizedPythagorasLsystem.svg</a> (26. 1. 2015)	
Obr. 21:	Náhodné přepisovací pravidlo.....	30
	<a href="http://cs.wikipedia.org/wiki/L-syst%C3%A9m#mediaviewer/File:PythagorasStochasticLsystem.svg">http://cs.wikipedia.org/wiki/L-syst%C3%A9m#mediaviewer/File:PythagorasStochasticLsystem.svg</a> (26. 1. 2015)	

## **Seznam příloh**

- Příloha 1: Modely rostlin – L-systémy
- Příloha 2: Program na generování L-systémů – zdrojové kódy

## Příloha 1: Modely rostlin – L-systémy

### Kapradina

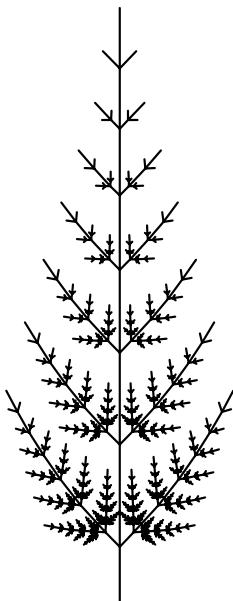
```
lssystem StochasticPlant {
  let li = 0.9; //length increase
  let wi = 0.8; //width increase
  let bi = 0.4; //branch increase
  let angle = 45;
  let td = 0.5; //decrease first branch part length
  let width = 0.2;

  set iterations = 7;
  set initialAngle = 90;
  set symbols axiom = A(10, 1, 1);
  set tropismVector = {0, 1, 0};
  set tropismCoefficient = 2;

  rewrite A(l, w, f) where f==0
  to F(l*li, w*wi)
  [+A(l*li*bi, w*wi*bi, 1)]
  [-A(l*li*bi, w*wi*bi, 1)]
  A(l*li, w*wi, 0);
  rewrite A(l, w, f) where f==1
  to F(l*li*td, w*wi*td)
  [+A(l*li*bi, w*wi*bi, 1)]
  [-A(l*li*bi, w*wi*bi, 1)]
  A(l*li, w*wi, 0);

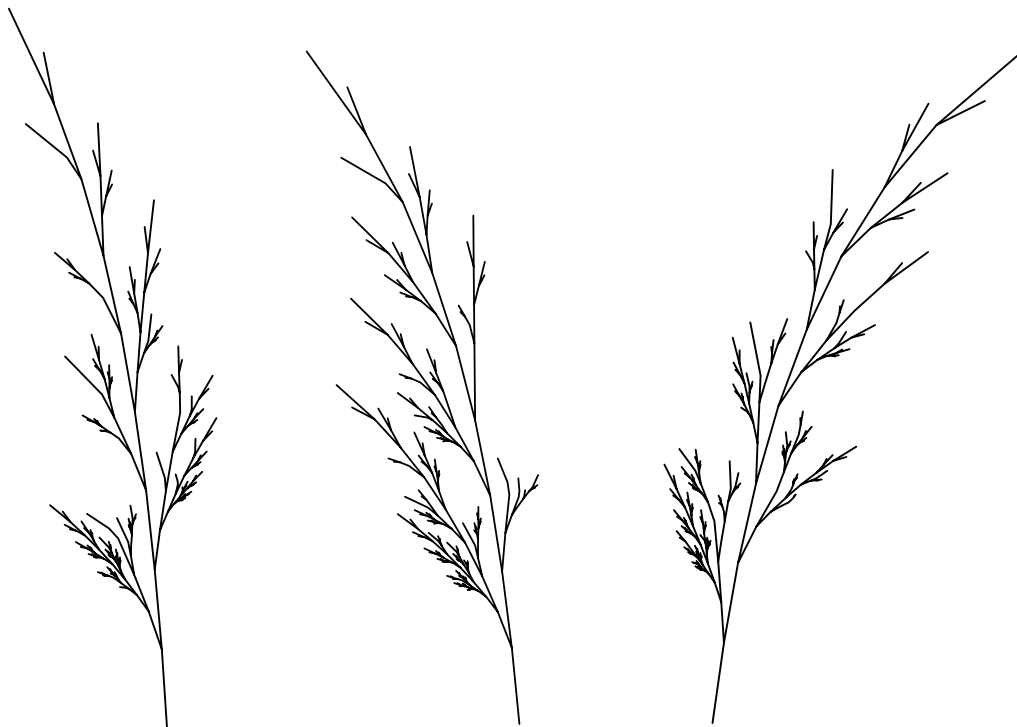
  interpret A(l, w, f) as DrawForward(l, width);
  interpret B(l, w) as DrawForward(l, width);
  interpret F(l, w) as DrawForward(l, width);
  interpret + as TurnLeft(angle);
  interpret - as TurnLeft(-angle);
  interpret [ as StartBranch();
  interpret ] as EndBranch();
}

process StochasticPlant with SvgRenderer;
```



## Stochastické traviny

```
lsystem Grass {  
  let angle = 14;  
  let thickness = 0.1;  
  
  set iterations = 8;  
  set initialAngle = random(80, 110);  
  set tropismVector = {0, -1, 0};  
  set tropismCoefficient = 2;  
  set symbols axiom = A(10);  
  
  rewrite A(x) where x<10 to F(x*0.75)[+A(x*0.5)]B(x) weight 5  
    to F(x/2)[+F(x)] weight 1;  
  rewrite A(x) to F(x*0.75)[+A(x*0.5)]B(x);  
  
  rewrite B(x) where x<10 to F(x*0.75)[-B(x*0.5)]A(x) weight 5  
    to f weight 1;  
  rewrite B(x) to F(x*0.75)[-B(x*0.5)]A(x);  
  
  interpret F(x) as DrawForward(x, thickness);  
  interpret A(x) as DrawForward(x, thickness);  
  interpret B(x) as DrawForward(x, thickness);  
  interpret + as TurnLeft(angle);  
  interpret - as TurnLeft(-angle);  
  interpret [ as StartBranch();  
  interpret ] as EndBranch();  
}  
  
process Grass with SvgRenderer;
```



## Strom

```
fun randomize (number, maxDistance) {
  return number + random(0, maxDistance) - random(0, maxDistance);
}

lsystem Tree {

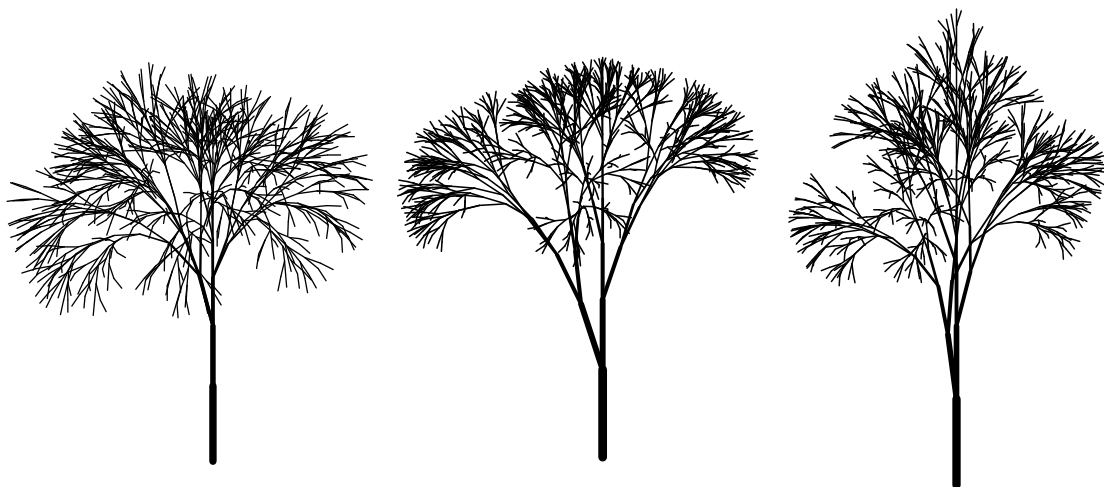
  let tlm = 0.8;    //trunkLength multiplier
  let blm = 0.8;    //branchLength multiplier
  let wm = 0.7;     //widthMultiplier
  let angle = 20;
  let rotation = 137.5;
  let rran = 20;    //rotation randomness

  set iterations = 10;
  set initialAngle = 90;
  set tropismVector = {0, -1, 0};
  set tropismCoefficient = 3;
  set symbols axiom = /(random(0, 360)) A(10, 1);

  rewrite A(l, w) to F(l, w)
    /(randomize(rotation, rran))[+B(l*blm, w*wm)]
    A(l*tlm, w*wm);
  rewrite B(l, w) to F(l, w)
    /(randomize(rotation, rran))[+C(l*blm, w*wm)]
    B(l*blm, w*wm);
  rewrite C(l, w) to F(l, w)
    /(randomize(rotation, rran))[-B(l*blm, w*wm)]
    C(l*blm, w*wm);

  interpret F(l, w) as DrawForward(l, w);
  interpret A(l, w) as DrawForward(l, w);
  interpret B(l, w) as DrawForward(l, w);
  interpret + as TurnLeft(angle);
  interpret - as TurnLeft(-angle);
  interpret /(r) as Roll(r);
  interpret [ as StartBranch();
  interpret ] as EndBranch();
}

process Tree with SvgRenderer;
```



## Příloha 2: Program na generování L-systémů – zdrojové kódy

### Turtle.cs

```
using System;
using System.Collections.Generic;
using System.Drawing;

namespace LSystem
{
    class Turtle
    {
        private Position position;
        private Bitmap buffer;
        private Stack<Position> savedPositions;

        public Turtle(Position start, Size canvasSize)
        {
            position = start;
            position.angle += 180;
            buffer = new Bitmap(canvasSize.Width, canvasSize.Height);
            savedPositions = new Stack<Position>();
        }

        public void TurnRight(float angle)
        {
            position.angle += angle;
        }

        public void TurnLeft(float angle)
        {
            TurnRight(-angle);
        }

        public void TurnBackwards()
        {
            position.angle += 180;
        }

        public void BeginBranch()
        {
            savedPositions.Push(position);
        }

        public void EndBranch()
        {
            position = savedPositions.Pop();
        }

        public void MoveForward(float length)
        {
            position.x += length * (float)Math.Cos(position.angle * Math.PI / 180);
            position.y += length * (float)Math.Sin(position.angle * Math.PI / 180);
        }

        public void DrawForward(float length)
        {
            Position oldPosition = position;
            MoveForward(length);
            DrawLine(oldPosition, position);
        }
    }
}
```

```

    public Bitmap GetImage()
    {
        return buffer;
    }

    private void DrawLine(Position from, Position to)
    {
        using (Graphics g = Graphics.FromImage(buffer))
        {
            g.DrawLine(Pens.Black, from.x, from.y, to.x, to.y);
        }
    }
}

```

### RewritingRule.cs

```

namespace LSystem
{
    struct RewritingRule
    {
        public RewritingRule(char toRewrite, string rewriteWith)
        {
            this.toRewrite = toRewrite;
            this.rewriteWith = rewriteWith;
        }
        public char toRewrite;
        public string rewriteWith;
    }
}

```

### Program.cs

```

using System;
using System.Windows.Forms;

namespace LSystem
{
    static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new FrmLSystem());
        }
    }
}

```

### Position.cs

```

namespace LSystem
{
    struct Position
    {
        public Position(float angle, float x, float y)
        {
            this.angle = angle;
            this.x = x;
            this.y = y;
        }
        public float angle, x, y;
    }
}

```

## LSystemParser.cs

```
using System;
using System.Text.RegularExpressions;

namespace LSystem
{
    class LSystemParser
    {
        private char ruleSeparator;
        private char rulePartsSeparator;

        public LSystemParser(char ruleSeparator, char rulePartsSeparator)
        {
            this.ruleSeparator = ruleSeparator;
            this.rulePartsSeparator = rulePartsSeparator;
        }

        public LSystem Parse(string axiom, string rules)
        {
            LSystem parsedLSystem;
            if (AxiomValid(axiom))
            {
                parsedLSystem = new LSystem(axiom,
                    getRewritingRules(
                        SplitRules(rules, ruleSeparator), rulePartsSeparator));
            }
            else
            {
                throw new ArgumentException(
                    "The axiom \"" + axiom + "\" is not valid.");
            }
            return parsedLSystem;
        }

        private RewritingRule[] getRewritingRules(string[] rules, char separator)
        {
            RewritingRule[] rewritingRules = new RewritingRule[rules.Length];
            for (int i = 0; i < rules.Length; i++)
            {
                char toRewrite = rules[i][0];
                string rewriteWith = rules[i].Split(separator)[1];
                RewritingRule r = new RewritingRule(toRewrite, rewriteWith);
                rewritingRules[i] = r;
            }
            return rewritingRules;
        }
    }
}
```



```

private string[] SplitRules(string rules, char separator)
{
    rules = rules.Trim();
    rules = rules.Remove(rules.Length - 1);
    string[] individualRules = rules.Split(separator);
    for (int i = 0; i < individualRules.Length; i++)
    {
        string rule = individualRules[i].Trim();
        if (RuleValid(rule))
        {
            individualRules[i] = rule;
        }
        if (!RuleValid(rule))
        {
            throw new ArgumentException(
                "The rule \"" + rule + "\" is not valid.");
        }
    }
    return individualRules;
}

private bool AxiomValid(string axiom)
{
    return Regex.IsMatch(axiom, @"^\w\+-\[\|\]{1,}$");
}

private bool RuleValid(string rule)
{
    return Regex.IsMatch(rule, @"^\w\+-\[\|\]{1}=\w\+-\[\|\]{1,}$");
}
}
}
}

```

## LSystemManager.cs

```
using System.Drawing;

namespace LSystem
{
    class LSystemManager
    {
        private LSystem lSystem;
        private LSystemParser parser;
        private Turtle turtle;

        public LSystemManager()
        {
            parser = new LSystemParser(';','=');
        }

        public Bitmap GetImage(string commands, float turnAngle, float stepLength,
            float x, float y, int width, int height)
        {
            turtle = new Turtle(new Position(90f, x, y), new Size(width, height));
            foreach (char c in commands)
            {
                if (c == '+')
                    turtle.TurnRight(turnAngle);
                else if (c == '-')
                    turtle.TurnLeft(turnAngle);
                else if (c == '[')
                    turtle.BeginBranch();
                else if (c == ']')
                    turtle.EndBranch();
                else if (c == '|')
                    turtle.TurnBackwards();
                else if (c.ToString().ToUpper() == c.ToString())
                    turtle.DrawForward(stepLength);
                else if (c.ToString().ToLower() == c.ToString())
                    turtle.MoveForward(stepLength);
            }
            return turtle.GetImage();
        }

        public string Process(string axiom, string rules, int iterations)
        {
            lSystem = parser.Parse(axiom, rules);
            for (int i = 0; i < iterations; i++)
            {
                lSystem.Iterate();
            }
            return lSystem.getText();
        }
    }
}
```

## LSystem.cs

```
namespace LSystem
{
    class LSystem
    {
        private string axiom;
        private string lSystem;
        private RewritingRule[] rules;

        public LSystem(string axiom, RewritingRule[] rules)
        {
            this.axiom = axiom;
            this.rules = rules;
            lSystem = axiom;
        }

        public void Iterate()
        {
            string lSystemBuffer = "";
            foreach (char c in lSystem)
            {
                string rewriteWith = c.ToString();
                foreach (RewritingRule r in rules)
                {
                    if (r.toRewrite == c)
                    {
                        rewriteWith = r.rewriteWith;
                    }
                }
                lSystemBuffer += rewriteWith;
            }
            lSystem = lSystemBuffer;
        }

        public string getText()
        {
            return lSystem;
        }
    }
}
```

## FrmLSystem.cs

```
using System;
using System.Windows.Forms;

namespace LSystem
{
    public partial class FrmLSystem : Form
    {
        private LSystemManager manager = new LSystemManager();

        public FrmLSystem()
        {
            InitializeComponent();
        }

        private void btnProcess_Click(object sender, EventArgs e)
        {
            try
            {
                tbxLsystem.Text = manager.Process(
                    tbxAxiom.Text, tbxRules.Text, (int)nudIterations.Value);
            }

            catch (ArgumentException ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void btnDraw_Click(object sender, EventArgs e)
        {
            pbxTurtle.Image = manager.GetImage(tbxLsystem.Text, (float)nudAngle.Value,
                (float)nudLength.Value, (float)nudPositionX.Value,
                (float)nudPositionY.Value, pbxTurtle.Width, pbxTurtle.Height);
        }
    }
}
```