VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# FRAKTÁLY V POČÍTAČOVÉ GRAFICE

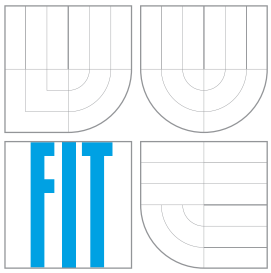
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN ŠELEPA

BRNO 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# FRAKTÁLY V POČÍTAČOVÉ GRAFICE
FRACTALS IN COMPUTER GRAPHICS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN ŠELEPA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. STANISLAV SUMEC, Ph.D.

BRNO 2007

# Abstrakt

V této práci se zabývám fraktály. První kapitola představuje úvod do problematiky, kterou se tato prácec zabývá. Druhá kapitola obsahuje základní pojmy z oblasti fráktálů a fraktální geometrie. Ve třetí kapitole je uvedena historie fraktálů a některé významné osobnosti z fraktální vědy. Kapitola čtvrtá obsahuje klasifikaci fraktálů dle různých kritérií. V této části práce také uvádím příklady fraktálů jednotlivých typů. V páté kapitole jsou uvedeny některé nejpoužívanější programy zabývající se vykreslováním fraktálů. Šestá kapitola je věnována demonsrtační aplikaci, kterou jsem v rámci této bakalářské práce vytvořil.



# Klíčová slova

fraktální geometrie, fraktál, Hausdorffova dimenze, soběpodobnost, atraktor, dynamický systém, jednodimezionální dynamický systém, dvoudimenzionální dynamický systém, dynamický systém v komplexní rovině, systém iterovaných funkcí, algoritmus náhodné procházky, deterministický algoritmus, upravený algoritmus náhodné procházky, algoritmus minima pixelů, L-systém, stochastický fraktál, Brownův pohyb, posun středního bodu, spektrální analýza



# Abstract

The goal of this work is to give introduction and specification of fractals. The first chapter presents the basics of fractal geometry. The second chapter maps the history of fractals and points out the most important people of fractal science. The third chapter presents fractal classification based on several criteria and gives basic examples. The fourth chapter is a summary of widely used applications for fractal creation. The last chapter describes the application that was made to demonstrate given algorithms and fractals mentioned in this thesis.



# Keywords

fractal geometry, fractal, Hausdorff dimension, self-similarity, attractor, dynamical system, one-dimensional dynamical system, two-dimensional dynamical system, dynamical system in complex plane, iterated function system, random walk algorithm, deterministic iteration algorithm, modified random walk algorithm, minimal plotting algorithm, L-system, stochastic fractal, Brownian motion, midpoint displacement method, spectral synthesis



# Citace

Jan Šelepa: Fraktály v počítačové grafice, bakalářská práce, Brno, FIT VUT v Brně, 2007

Fraktály v počítačové grafice

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Stanislava Sumce, Ph.D.

.....
Jan Šelepa
16. května 2007

*Zadání bakalářské práce/2541/2006/xselep00*

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií Akademický rok 2006/2007

# Zadání bakalářské práce

Řešitel: **Šelepa Jan**
Obor: Informační technologie
Téma: **Fraktály v počítačové grafice**
Kategorie: Počítačová grafika

Pokyny:

1. Prostudujte a popište historii a pozadí nauky o fraktálech.
2. Prostudujte a popište typické i méně známé algoritmy zobrazování fraktálních obrazců v počítačové grafice.
3. Prostudujte a popište použití fraktálů v praxi.
4. Prostudujte a popište existující a dostupné softwarové balíky pro výpočet fraktálů.
5. Navrhněte a implementujte program zobrazující vybrané fraktály.
6. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; prezentujte projekt plakátkem.

Literatura:

- Barnsley Michael: *"Fractals Everywhere"*, Academic Press Inc., 1988, ISBN 0-12-079062-9
- Barnsley Michael, Devaney R. L., Mandelbrot Benoit B., Peitgenn Heinz-Otto, Saupe Dietmar, Voss Richard: *"The Science of Fractal Images"*, Springer-Verlag, New York, 1988
- Mandelbrot Benoit B.: *"The Fractal Geometry of Nature"*, W. H. Freeman, New York; San Francisco, 1982, ISBN 0-7167-1186-9
- Peitgen Heinz-Otto, Richter Peter: *"The Beauty of Fractals"*, Springer-Verlag, New York, 1986, ISBN 0-387-15851-0

Při obhajobě semestrální části projektu je požadováno:

- dílčí řešení bodů 1.-4.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Sumec Stanislav, Ing., Ph.D.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2006
Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2
L.S.

doc. Dr. Ing. Pavel Zemčík
*vedoucí ústavu*

**LICENČNÍ SMLOUVA**
**POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

**1. Pan**

Jméno a příjmení: **Jan Šelepa**
Id studenta: 89016
Bytem: Prachatická 9, 370 05 České Budějovice
Narozen: 02. 08. 1983, České Budějovice
(dále jen "autor")

a

**2. Vysoké učení technické v Brně**

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

**Článek 1**
**Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Fraktály v počítačové grafice
Vedoucí/školitel VŠKP: Sumec Stanislav, Ing., Ph.D.
Ústav: Ústav počítačové grafiky a multimédií
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:
tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

**Článek 2**
**Udělení licenčního oprávnění**

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
   - ☐ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

**Článek 3**
**Závěrečná ustanovení**

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

..... .....Lelepa Jan.....

Nabyvatel Autor

# Obsah

# Kapitola 1

# Úvod

Fraktály jsou vizuálně a matematicky velice zajímavé útvary. Tyto útvary můžeme najít v mnoha jevech, dokonce i některé objekty z reálného světa mají fraktální vzhled. Fraktální strukturu můžeme vidět například u stromů nebo hor. Jevy a objekty, které mají fraktální strukturu začali zkoumat matematici a tak se zrodila fraktální geometrie jako vědecký obor. Od prvních objevů v tomto oboru se fraktální věda velmi rozvinula. V dnešní době již rozlišujeme několik typů fraktálů, které se liší způsobem vytváření. Fraktální útvary byly nalezeny již v mnoha vědních disciplínách a jejich použití se značně rozšířilo. Fraktální obrazce jsou zajímavé hlavně pro svoji nekonečnou členitost a to, že mohou vznikat i z velmi jednoduchých rovnic. Zajímavé na fraktálech je také to, že dodnes neexistuje přesná matematická definice, která by dostatečně přesně definovala všechny fraktály jako celek.

Tato práce se zabývá především rozdělením fraktálů do skupin, které se v průběhu studia fraktální geometrie utvořily. Také uvádí některé zajímavé příklady fraktálů, které tyto skupiny reprezentují. U každého konkrétního příkladu bude vždy uveden způsob jeho vytváření. K rozvoji fraktální geometrie jako samostatné vědní disciplíny velkou měrou přispěly počítače, pomocí nichž se mohou fraktály jednoduše počítat a také vykreslovat v různých podobách, což předtím šlo jen velmi obtížně. Dnes již existuje řada kvalitní programů, které se specializují na vytváření fraktálů. Některé z těchto programů jsou popsány i v této práci. Součástí mé bakalářské práce je také demonstrační aplikace, pomocí které se dají vykreslovat a upravovat některé fraktály. Proto, abychom mohli pracovat s fraktály, je nutné znát alespoň základní pojmy z problematiky. Výčet těchto základních pojmů jsem také uvedl v této práci. Cílem mé bakalářské práce je také poukázat na možnosti praktického využití fraktálů.

# Kapitola 2

# Teorie fraktálů

## 2.1 Fraktální geometrie

Geometrie, která je vyučována na školách, se zabývá pravidelnými obrazci, jako je čtverec, kružnice nebo přímka. Kolem nás v reálném světě však jsou nepravidelné útvary. Tyto útvary většinou vyjadřujeme aproximací jejich tvaru, tím však dochází ke zkreslení a ztrátě přesnosti. Pokud například budeme chtít změřit obvod nějakého ostrova můžeme toho docílit tak, že obejdeme celý ostrov s nějakým měřidlem, které má odpovídající délku. Tímto však získáme pouze přibližný obvod ostrova a pokud bychom si vzali měřidlo s kratší délkou vyšel by nám obvod toho samého ostrova o trochu větší, protože jsme zachytili více detailů. Teoreticky pokud bychom zmenšovali délku měřidla do nekonečna, pak by nám vyšel i obvod ostrova nekonečný. *Fraktální geometrie* se na rozdíl od klasické geometrie zabývá nepravidelností objektů.

## 2.2 Fraktál

Poprvé slovo fraktál použil *Benoit B. Mandelbrot*, který se snažil pro svůj objev vymyslet nějaké pojmenování a když listoval sešitem svého syna narazil na slovo *fractus*, z něhož je odvozené slovo *frangere*, což znamená rozlámat na nepravidelné kousky. Fraktál je tedy jakýkoliv geometrický nepravidelný útvar, který když rozdělíme tak v ideálním případě vznikne několik soběpodobných útvarů nezávislých na měřítku. Fraktály mají často ještě jiné zajímavé vlastnosti, jako například nekonečně malý obsah nebo, již zmiňovaný, nekonečně velký obvod.Dosud nejvýstižnější definici fraktálů uvedl právě pan Benoit B. Mandelbrot, který ho definoval takto: Fraktál je množina, jejíž hodnota Hausdorffovy dimenze přesahuje hodnotu dimenze topologické.

## 2.3 Topologická dimenze

Topologická dimenze je dimenze, kterou známe, a která určuje geometrický rozměr tělesa. Můžeme také říci, že topologická dimenze určuje minimální počet parametrů, kterými lze dané těleso popsat. Bod má topologickou dimenzi rovnu nule, přímka rovnu jedné, kružnice rovnu dvěma a krychle rovnu třem. V topologické dimenzi, že rozměry mohou být zadávány v různých jednotkách a na vlastnostech tělesa to nic nezmění.

## 2.4 Hausdorffova dimenze

Tuto dimenzi začali vědci studovat až se studiem fraktálu, protože do té doby jsme si vystačili s dimenzí topologickou. Hausdorffovu dimenzi asi nejlépe vysvětlíme na příkladu. Pokud například měříme délku přímky vyjde nám i při změně měřidla stejný výsledek. Ovšem vezmeme-li si jako příklad měření obvod ostrova, pak se výsledky liší podle délky použitého měřidla. Obvod ostrova tedy zabírá v rovině více místa než přímka, nezabírá však celou plochu. Z toho vyplývá, že jeho dimenze je větší než dimenze přímky, ale zároveň menší než dimenze plochy. Pokud ovšem dimenze přímky je rovna jedné a dimenze plochy rovna dvěma, pak musí být dimenze křivky znázorňující obvod ostrova neceločíselná. Toto neceločíselné číslo je Hausdorffova dimenze. Rozdíl mezi dimenzí Hausdorffovou a topologickou pak udává jak moc je určité těleso členité, nebo také jakou rychlostí roste délka, nebo odpovídající veličina v jiných rozměrech.

## 2.5 Soběpodobnost

Kromě dimenze se občas pro definici fraktálů používá pojem soběpodobnost nebo také invariance vůči změně měřítka. To znamená, že pokud se podíváme na určitý objekt v různých zvětšeních, nebo-li v různých měřítkách, bude objekt vypadat podobně. Tuto vlastnost mají skoro všechny fraktály kromě některých náhodných. V reálném světě můžeme soběpodobnost pozorovat například u mraků, rostlin nebo hor. Soběpodobné množiny vznikají opakováním sebe samých při určité transformaci. Transformace v tomto případě může být například změna velikosti nebo rotace. Tyto množiny jsou také invariantní vůči změně měřítka. Musíme si ovšem uvědomit, že soběpodobnost je podobnost objektu pouze při změně měřítka. V klasické geometrii totiž najdeme útvary, které jsou si podobné nebo jsou dokonce úplně totožné po provedení určité transformace. Pokud vezmeme například čtverec a budeme ho zrcadlit podle středu dostaneme úplně stejný čtverec. To samé se dá udělat i například s kružnicí. Pokud spolu s touto soběpodobností není i soběpodobnost při změně měřítka nemůžeme mluvit o definici fraktálu, ale o normálním geometrickém objektu.

## 2.6 Atraktor

Atraktor dynamického systému je množina bodů, do kterých systém směřuje. Nebo se atraktor dá vyjádřit jako množina hodnot, kterých může nabýt stavový vektor po dostatečně dlouhé době od inicializace systému. Atraktor se využívá například u systémů iterovaných funkcí, kde se většinou vykresluje právě jeho atraktor. Atraktory můžeme rozdělit do několika kategorii:

- Atraktor z pevných bodů
- Atraktor z periodických nebo kvaziperiodických bodů
- Chaotický atraktor
- Podivný atraktor

Je-li atraktor složen z pevných bodů, pak se systém v nekonečném čase ustálil v nějakém předem stanovitelném a spočitatelném stavu.

Pokud je atraktor tvořen periodickými nebo kvaziperiodickými body pak se systém v nekonečném čase ustálí tak, že osciluje mezi několika stavy. Tyto stavy mohou být buďto spočitatelné nebo nespočitatelné.

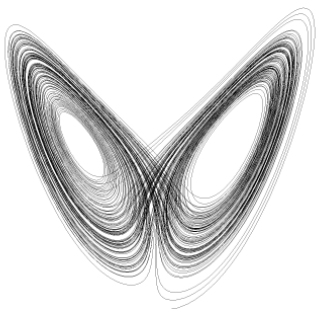
U chaotického atraktoru nelze s určitostí předpovědět, kde se daný systém ustálí, což ovšem neznamená, že by neměl svůj atraktor. Takovou vlastnost mají systémy, které jsou velmi citlivé na počáteční podmínky, nebo-li systémy, které i při malé změně počátečních podmínek vykazují velmi odlišné výsledky. Slovo „chaotický“ v tomto případě nepředstavuje náhodnost, protože se stále jedná o deterministické systémy.

Podivný atraktor je z hlediska fraktální geometrie nejzajímavější. Systém s tímto typem atraktoru může vzniknout, pokud je tento systém definován minimálně třemi navzájem souvisejícími diferenciálními rovnicemi. Atraktor tohoto systému bude vykazovat vlastnosti pravidelného, ale i chaotického atraktoru. Přesná matematická definice podivného atraktoru ještě neexistuje, protože definice, které existují nezahrnují všechny typy podivných atraktorů. Nejpoužívanější popis podivných atraktorů je takový, že mají podobné vlastnosti jako fraktály. Také platí, že všechny chaotické atraktory jsou zároveň i podivnými atraktory opačně to však neplatí.

Jedním z často probíraných příkladů dynamických systémů s podivným atraktorem je systém složený ze tří těles, kde známe počáteční podmínky jako jejich polohu, hmotnost, rychlost a směr pohybu. Úkolem je spočítat polohu těles v libovolném časovém okamžiku. Tento úkol se ukázal jako nevyřešitelný, neboť koncem 19. století dokázal francouzský matematik Henri Poincaré, že analytické řešení neexistuje. Není tedy možné jakýmkoliv způsobem určit aspoň přibližné chování vzájemně se ovlivňujících tří těles. Tento případ můžeme sledovat například v pásu asteroidů mezi Marsem a Jupiterem, kde obíhá skupina tří velkých asteroidů.

### 2.6.1 Lorenzův atraktor

První dynamický systém s podivným atraktorem, který byl podrobněji popsán je vidět na obrázku 2.1. Tento atraktor je pojmenován po americkém matematikovi a meteorologovi Edwardu Nortonovi Lorenzovi, který v roce 1963 při simulování vývoje počasí na velmi zjednodušeném modelu použil systém popsaný třemi diferenciálními rovnicemi. Na tomto modelu se také numericky i analyticky ověřila velká citlivost na počáteční podmínky, což Lorenz pojmenoval jako *motýlí efekt*. Pokud bychom vzali matematicky přesný model pak nastane situace, že každá smyčka bude mít unikátní dráhu, to znamená, že se nebude překrývat s žádnou jinou smyčkou. Další informace o tomto fraktálu lze nalézt na [6].



Obrázek 2.1: Lorenzův atraktor

## 2.7 Chaos

Pojmem chaos označujeme takovou vlastnost dynamického, ale i deterministického, systému, kdy nemůžeme s určitostí spočítat budoucí stavy daného systému. Tento chaos nastává můžeme pozorovat u systémů, které jsou velmi citlivé na počáteční podmínky. Pokud u takovýchto systémů zvolíme dvoje počáteční podmínky, které jsou nekonečně blízké budou se nám výsledky vycházející z těchto počátečních podmínek exponenciálně vzdalovat, což nakonec vede k nepředpovídatelnosti systému.

Velmi často se se jako příklad chaotického systému používá již zmiňovaný model počasí, kterým se zabýval Edward Norton Lorenz. Dalším systémem s chaotickým chováním je například systém, jehož součástí je kužel a na jeho vrcholu umístěná kulička. U tohoto systému pokud jen nepatrně změníme počáteční polohu kuličky velice to ovlivní směr pádu kuličky a tím i chování celého systému. Pro tyto systémy také platí, že jejich počáteční stav nelze přesně změřit, protože už samotné měření ovlivňuje tyto počáteční podmínky.

## 2.8 Uplatnění fraktálů

### 2.8.1 V různých vědních disciplínách

Princip opakování podobných tvarů ve zmenšené podobě je vidět prakticky u jakékoliv komplexní, složité struktury, která je vytvářena i pomocí velmi jednoduchých pravidel. Způsob, jakým probíhá větvení stromů či cév a žil v tělech živočichů nebo hromadění baktérií a řas v koloniích, se dá matematicky uspokojivě popsat pouze fraktální geometrií, pokusy o popis pomocí klasické geometrie byly buď neúměrně komplikované, a/nebo neodpovídaly realitě.

Jedním z vědních oborů, kde můžeme fraktály pozorovat je změna stavu některých materiálů. Pokud bychom studovali změnu stavu magnetického materiálu na materiál nemagnetický, zjistili bychom, že materiál se skládá z elementárních magnetů, které v závislosti na teplotě zda je materiál jako celek magnetický nebo ne. Při nízkých teplotách je materiál jako celek magnetický, ale pokud bychom ho zkoumali podrobněji pod mikroskopem, zjistili bychom, že existují malé skupiny neuspořádaných elementárních magnetů. Pokud tento materiál zahřejeme na vysoké teploty, pak je jako celek nemagnetický, ale opět se najdou místa, kde jsou elementární magnety uspořádány. Magnetismus je tedy závislý na měřítku pohledu v jakém se na materiál díváme. Pokud bychom upravovali teplotu materiálu tak, aby měnil svoji magnetickou vlastnost ať už z magnetického stavu do nemagnetického nebo naopak, nalezli bychom teplotu, při které se tato změna odehrává. Tato teplota se nazývá Curierova teplota. Při této teplotě jednak nemůžeme dokázat zda je materiál magnetický nebo ne a také při této teplotě vypadá materiál ve všech měřítkách stejně a má fraktální strukturu.

Fraktální strukturu můžeme pozorovat i při změně jiných stavů materiálu. Jedním z takových případů může být změna skupenství látky. Například změna vody na páru nebo tání ledu. Pokud bychom hleděli na hmotu z makroskopického hlediska, pak existují tři základní stavy a to: pevné, kapalné a plynné. Pokud ovšem začneme hmotu zkoumat blíže, zjistíme, že těchto stavů je daleko více a při přechodech mezi jednotlivými stavy můžeme pozorovat fraktální strukturu.

Dalším jevem, který má fraktální strukturu je Brownův pohyb. Tento pohyb vzniká chaotickým pohybem částic v hmotě a je způsoben nenulovou teplotou hmoty. Brownův pohyb lze dobře sledovat například tehdy, kdybychom vzali dvě různě zbarvené kapaliny a smíchali je dohromady. Také lze tento pohyb relativně věrohodně simulovat, pomocí

fraktální geometrie, na počítači.

Fraktální geometrie nachází své využití také v biologii a medicíně. Například při zkoumání krve se měří Hausdorffova dimenze jejích částeček. Dokonce někteří odborníci věří, že existují jistá spojitost mezi Hausdorffovou dimenzí povrchu mozku a inteligencí toho určitého jedince. Spojitost je v počtu a členitosti mozkových závitů, která má ovlivňovat určité vlastnosti mozku, jako například paměť nebo inteligenci.

### 2.8.2 V počítačové grafice

Fraktální geometrie však nachází své největší uplatnění v oboru počítačová grafika. Pomocí ní lze totiž fraktály zkoumat velmi podrobně. Počítačová grafika pak naopak využívá poznatky získané o fraktálech ve svůj prospěch.

Jak se stále zvyšuje nárok na kvalitu obrazu prezentovaného pomocí počítače je potřeba se zabývat postupy při modelování objektů z reálného světa. V podstatě existují tři možnosti jak vytvořit model nějakého skutečného objektu v počítači:

1. Použití nějakého modelovacího programu, například typu CAD. Tento způsob je vhodný při modelování technických a geometrických útvarů. Pokud bychom ovšem tímto způsobem chtěli modelovat přírodní útvary jako například hory nebo stromy, potřebovali bychom pro popis těchto objektů velké množství dat, a ani po dlouhém úsilí by nebyl výsledek uspokojující.

2. Přímé snímání objektu, který chceme zobrazit v počítači. Tento způsob je ovšem velmi omezený co se týká rozměrů skenovaného objektu. Trojrozměrné skenery totiž mají omezenou rozlišovací schopnost, tudíž skenovaný objekt nemůže být příliš malý, ale také nemůže být příliš velký, protože je fyzicky nemožné naskenovat třeba horu nebo dům. Dalším nedostatkem této metody je skenování velmi členitých objektů. Například skenování nějakého chlupatého zvířete. Po skenování vzniká velké množství dat, jelikož skenovaný objekt je v počítači reprezentován trojúhelníky, a aby bylo zobrazení věrohodné musí tyto trojúhelníky být velice malé musí jich být mnoho. Další nevýhodou je vysoká cena skenovacího zařízení.

3. Procedurální modelování. U tohoto způsobu se objekt nereprezentuje tvarem, ale způsobem jeho generování. Výhoda této metody spočívá v množství dat, které jsou potřeba zadat pro vytvoření objektu. Výsledný objekt se dá lehce změnit změnou počátečních podmínek. Při používání této metody je nutné správně vybrat metodu pro generování určitých typů objektů. Dále musíme vědět co který parametr té určité metody znamená a co ovlivňuje. Nevýhodou této metody je to, že není hned vidět výsledný objekt, ale musí se nechat vygenerovat a poté je možné jej upravovat změnou počátečních podmínek. Takto se dají vytvořit i různé animace, například růst stromu.

   U této metody se využívají i znalosti fraktální geometrie. Fraktály se používají při generování reálných objektů jako jsou rostliny, stromy, kameny, hory, mraky a podobně. Pro tento účel se fraktály hodí nejlépe a také mají ze všech metod nejlepší výsledky.

Fraktály se také mohou velmi efektivně použít pro generování textur. Výhodou takto vytvořených textur je jejich velmi malá náročnost na data potřebná pro vygenerování. Další výhodou tohoto způsobu je možnost změny velikosti výsledné textury bez jakéhokoliv zkreslení, které můžeme pozorovat u běžných bitmapových textur.

Fraktály lze také využít při animaci. Rozšíříme-li generování fraktálu o jeden rozměr, lze tento nadbytečný rozměr považovat za čas a tímto způsobem fraktál animovat. Tato metoda je použita při zobrazování ohně nebo pohybujících se mraků. Lze ji také kombinovat se systémem částic, což umožňuje animaci mnoha jevů, které nelze běžnými metodami zobrazit.

Fraktální geometrie se s úspěchem používá na rozpoznávání obrazu. V tomto oboru se zatím asi nejvíce rozvinulo rozpoznávání písma. Přesto však uspokojivé výsledky vykazuje pouze rozpoznávání strojového písma. Ani nejmodernější technologie totiž nejsou schopny rozeznat rukou psané písmo od různých lidí s různým typem rukopisů.

I při kompresi dat se dá použít fraktální geometrie. V tomto případě se používá systém iterovaných funkcí, což je jeden z typů fraktálů. Algoritmus, který se dnes používá pro kompresi dat byl představen nejprve jako algoritmus pro generování textur. Až po pozdějším zkoumání a rozboru se tato metoda ukázala jako vhodná pro kompresi dat. Metodu komprese dat pomocí fraktální geometrie využívá grafický formát FIF.

V neposlední řadě jsou fraktály velice zajímavé svými tvary, a proto si někteří matematici, kteří zkoumali fraktály, vytvářeli fraktály jen pro jejich estetičnost. V dnešní době existuje již velké množství programů, které generují fraktály různých tvarů a jejich jediný účel je pouze estetický.

# Kapitola 3

# Historie fraktálů

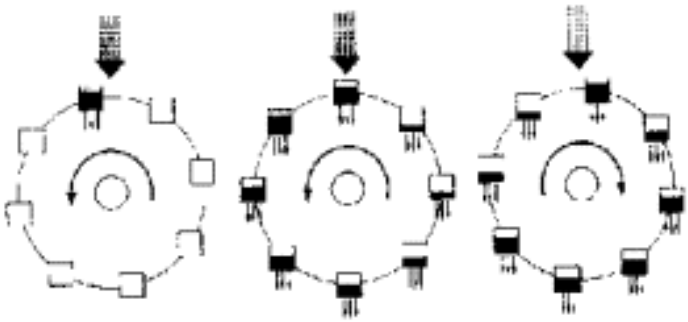
Kolem roku 1960 se někteří vědci zabývali počasím a jeho předpovědí. Jedním z těch, kteří se touto problematikou zabývali byl i Edward Lorenz. Fraktální geometrie je rozvíjena od 60. let 20. století. Dnes je to samostatná a již poměrně rozsáhlá vědní disciplína. K rozvoji tohoto oboru velmi výraznou měrou přispěl rozvoj osobních počítačů. Za zakladatele této disciplíny je považován objevitel fraktálů Benoit B. Mandelbrot. Tento polský vědec uvedl definici fraktálů, která se používá v podstatě dodnes, i přes to, že jeho definice není naprosto přesná. O definování fraktálů se pokoušeli i někteří vědci a matematici, ale jejich popis a definice fraktálů nebyla ani dost přesná ani tak všeobecná jako právě Mandelbrotova definice. Další velice výraznou postavou dějin fraktálů je Edward Norton Lorenz. Historie fraktálů je popsána i na [4].

## 3.1 Edward Norton Lorenz

Tento americký matematik a meteorolog byl jedním z mála vědců, kteří měli ke své práci a výzkumům k dispozici počítač. Snažil se vytvořit rovnice, které by popisovali chovaní počasí na naší planetě. Na tehdejších počítačích však byly výpočty i velmi jednoduchých rovnic poměrně zdlouhavé, a proto si Lorenz jednou výpočet zkrátil výpočtem pouze poloviny grafu. Počáteční hodnoty zvolil dle předchozího výpočtu. Ovšem když Lorenz srovnal právě vykreslený graf s již dříve vypočítaným grafem zjistil, že se tyto dva grafy po nějaké době rozcházejí. Nejprve si myslel, že tato chyba je způsobena chybou počítače, ale při zopakování celého procesu dostal úplně stejný výsledek. Nakonec zjistil, že tento úkaz je způsoben tím, že počítač pracuje s osmi desetinnými místy, ale on jich zadal pouze šest. Právě po této zkušenosti si uvědomil, že tento systém je velmi citlivý na počáteční podmínky, a že je velmi obtížné předpovídat chování tohoto systému na delší dobu dopředu. Tato vlastnost se projevuje u většiny chaotických systémů.

Další problém, kterým se Lorenz zabýval se týká chování vodního kola. Na tomto kole jsou připevněny nádoby, do kterých postupně přitéká a odtéká voda. Tento systém lze vidět na obrázku 3.1. Voda přitéká zeshora vždy do vrchní nádoby, ta vlivem gravitační síly roztočí kolo. Lorenz vyjádřil chování tohoto systému soustavou rovnic, zadal je do počítače a nechal si vykreslit výsledný graf. Očekával, že vodní kolo se bude otáčet buď jedním směrem, nebo se budou směry otáčení rovnoměrně střídat, jenže tento systém neudělal ani jedno. Při zvolených počátečních podmínkách se systém dostal do nestability a nebylo jasné jak se tento systém bude chvat v příštích několika okamžicích.

Lorenz a jeho poznatky ovšem v jeho době nebyly pochopeny a jeho práce zapomenuta



Obrázek 3.1: Vodní kolo, kterým se Lorenz zabýval

až do roku 1972, kdy jeden z Lorenzových článků objevil jistý matematik a ten jej věnoval Jamesi Yorkemu. Článek na něj velice zapůsobil a uvědomil si, že nelze na všechny věci nahlížet jako na lineární. Začal článek rozdávat svým kolegům. Jednu kopii také půjčil Stevu Smaleovi. Na tuto kopii napsal své jméno a adresu, aby se mu vrátila. Smale shledal článek velmi zajímavým a jako York rozdával jeho kopie. Ovšem na těchto kopiích bylo jméno a adresa Jamese Yorkeho, a proto si někteří lidé mysleli, že článek pochází právě od něj. Yorke ale není znám pouze touto příhodou, je to velmi významný vědec, který se zabýval chaosem a dynamickými systémy.

## 3.2 Benoit B. Mandelbrot

Narodil se 20. 1. 1924 ve Varšavě. I když se narodil v Polsku, žil již od svých dvanácti let ve Francii, kde studoval pod vedením Gastona Julii a také Paula Lévyho. Mandelbrot mimojiné publikoval Juliovu práci a tím ho velice zviditelnil. Podle Julia byly později také pojmenovány Juliovy množiny. Mandelbrot se zabýval matematikou, vyučoval fyziku a filozofii. Během své vědecké činnosti dosáhl nejvyššího akademického ocenění na Yalské univerzitě.

Jedno z Mandelbrotových zaměstnání bylo u firmy IBM, kde působil ve výzkumném a vývojovém oddělení, ovšem ne jako programátor. Zde studoval mimojiné rozložení chyb, které vznikaly na přenosové lince. Taková chyba je pokud se na začátku a na konci jednobitové přenosové linky objevily bity s převrácenou hodnotou. Taková chyba měla na první pohled náhodné rozložení, ale po důkladném prozkoumání se zjistilo, že chyba má určitou pravidelnost. Chyby se vyskytovali ve shlucích a při změně časového měřítka se objevovali podobné vzory.

Podobné jevy, nezávislost na časovém měřítku a pravidelnost, objevil Mandelbrot i při studiu jiných oborů. Například při studiu vývoje cen na burze zjistil podobné znaky, což překvapilo jistě spoustu lidí. Když nalezl podobné znaky u takových, spolu nesouvisejících oborů, jen ho to podpořilo v dalším výzkumu. Při těchto výzkumech popsal Mandelbrot některé dnes již známé fraktály, tak však byly nazvány až později právě podle něj. Fraktály nacházel v živé i neživé přírodě. Podobné vlastnosti totiž nalezl také například u rozložení hmoty ve vesmíru.

V tomto období také vydává své knihy *Fraktály* a *Fraktální geometrie přírody*. V těchto knihách používá výklad, ve kterém bez mnoha podrobností porovnává zdánlivě nesouvisející jevy. Také si při výkladu místo složitých vzorců pomáhal analogií s jinými podobnými

jevy nebo také geometrií. Často ve svých knihách dával jen tvrzení a důkazy a rozbory nechával na jiných lidech. Takto se mohl věnovat novým zajímavým věcem, zatímco ostatní se zabývali jeho tvrzeními. Tímto způsobem výkladu se do značné míry lišil od zaběhnutého stylu výkladu definicí, tvrzením, větou a důkazem.

Benoit Mandelbrot je v dnešní době znám hlavně proto, že je po něm pojmenován jeden z nejslavnějších fraktálů. Mandelbrotova množina je dynamický systém, který leží v komplexní rovině. Tento fraktál byl poprvé prezentován v roce 1979 a po Mandelbrotovi ho pojmenoval na začátku 80. let 20.století Johnem Hubbardem. Mandelbrot nebyl první, kdo tento fraktál vytvořil, jako první ho však publikoval. Více informací o Mandelbrotovi lze nalézt na [5].

# Kapitola 4

# Typy fraktálů

## 4.1 Dynamické systémy

Dynamické systémy jsou typem fraktálů, které mají v praxi nejširší uplatnění. Dynamické systémy tvoří model, který je závislý na nezávislé veličině. Touto veličinou je většinou čas a od toho je také odvozen název dynamické systémy. Dynamický systém je určen počátečními podmínkami, podle kterých dynamický systém vypadá. Existují i dynamické systémy, které se ani v nekonečném čase neustálí ani nediverguje. Dynamický systém s takovými vlastnostmi mívá fraktální strukturu a je označován pojmem deterministický chaos.

Dynamický systém je determinován počátečními podmínkami a tyto podmínky popisují změnu systému v čase. Systém má stavový vektor, který určuje stav tohoto systému v určitém okamžiku. Počáteční podmínky jsou většinou zadány ve formě diferenciálních rovnic, které popisují změnu stavu systému v čase. Abychom mohli zjistit budoucí stav systému musíme spočítat diferenciální rovnice z počátečních podmínek a nahradit stavový vektor nově spočítaným vektorem.

Jako příklad dynamického systému s fraktální strukturou bych zde uvedl výpočet populačního růstu. Tento systém je zajímavý tím, že chování systému lze určit jediným parametrem. Podle hodnoty tohoto parametru totiž může být systém buďto ustálený, oscilující nebo chaotický. Při zkoumání populačního růstu byly objeveny konstanty, které platí obecně, nejen pro tento konkrétní příklad. I v komplexní rovině existují dynamické systémy s fraktální strukturou, nejznámějšími typy z této kategorie jsou Mandelbrotova a Juliova množina.

To, jaké má určitý systém chování, tedy zda je ustálený nebo ne, je velmi důležité při výpočtech nad tímto systémem. Pokud bychom totiž chtěli zjistit stav fraktálně dynamického systému v časovém okamžiku dále v budoucnosti museli bychom simulovat celý vývoj od začátku. Je to tak proto, že tento typ systémů je velice citlivý na počáteční podmínky, což znamená, že i sebemenší nepřesnost může vést k úplně jinému výsledku.

V počítačové grafice se mohou dynamické systémy i vhodně obarvit podle nějaké vlastnosti systému. U dynamických systému v komplexní rovině můžeme vzniklý fraktál obarvit podle počtu iterací, které bylo potřeba spočítat, abychom rozhodli zda určitý bod leží uvnitř nebo vně fraktálu. Výsledný vykreslený fraktál můžeme použít například jako texturu, kterou můžeme nanášet na prostorové objekty. Takto vytvořené textury zabírají velmi málo místa, je to proto, že se textury vygenerují podle počátečních podmínek. Výhodou je také to, že texturu můžeme libovolně zmenšovat nebo zvětšovat a výsledek nebude nijak zkreslen. Nevýhodou je ovšem velká náročnost na výpočet, pokud chceme vygenerovat texturu s velkým rozlišením, může to relativně zdržovat.

### 4.1.1 Jednodimenzionální dynamické systémy

Příkladem jednodimenzionálního dynamického systému je populační růst v uzavřeném prostoru. Při zkoumání zjednodušeného modelu tohoto systému vyšlo najevo, že růst populace v jednom časovém rozmezí závisí na pouze jednom předcházejícím stavu tohoto systému. Systém si tedy nepamatuje všechny své předchozí stavy. Růst populace v tomto systému začne klesat když celkový počet jedinců v prostoru systému dosáhne určité hodnoty. Pokud má naopak populace méně jedinců dochází k růstu této populace. Název tohoto dynamického systému je Verhulstův proces. Výpočet dalšího stavu systému se provádí výpočtem jednoduché rovnice, ve které se počítá s velikostí růstu a jedním předchozím stavem systému. Rovnice pro výpočet následujícího stavu systému vypadá takto:

$$x_{n+1} = G * x_n * (1 - x_n)$$

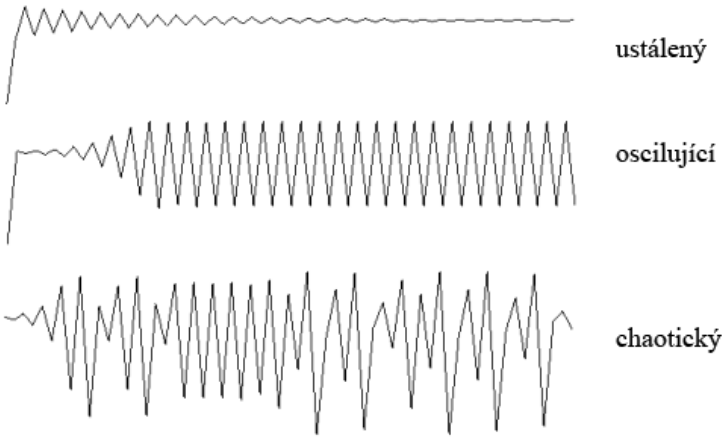
V tomto vzorci $x_n$ znamená počet jedinců v n-tém stavu systému a $G$ značí velikost růstu populace. Počet jedinců $x_n$ může nabývat hodnot od 0, což značí vymření populace, do 1, to znamená úplnou saturaci.

Zajímavé je sledovat vývoj systému pro různé hodnoty růstu populace $G$:

- $G < 2.0$ Pokud je hodnota růstu populace menší než 200%, pak se systém, tedy počet jedinců, po určité době ustálí na jedné hodnotě. Tato hodnota je atraktorem systému.
- $G = 2.0$ Když zvolíme hodnotu růstu rovnu 200% pak se systém neustálí, ale osciluje mezi dvěmi hodnotami. Tyto dvě hodnoty se střídají a jsou periodickým atraktorem takto nastaveného systému.
- $G = 2.45$ Při hodnotě růstu na 245% dochází k oscilaci systému mezi čtyřmi hodnotami, které opět tvoří atraktor systému.
- Pro některé hodnoty populačního růstu větší než 245%, například 300%, 345%, 354% a další, se systém dostává do oscilace mezi osmi, šestnácti, dvaatřiceti a tak dále stavy. Zkoumáním této posloupnosti byla objevena konstanta, která se nazývá Feigenbaumova konstanta. Tuto vlastnost bychom mohli najít u mnoha dynamických systémů, nejen jednorozměrných, a nazývá se zdvojování period.
- $G > 2.57$ Pokud hodnota růstu populace přesáhne 257% systém se stává chaotickým, což znamená, že nelze úspěšně předpovědět budoucí stav systému.

Příklad, který je zde uveden lze vyjádřit graficky například pomocí logistické mapy. Je to plošný graf, na kterém je přímo zobrazena funkce $f(x)$. Podle zvolených počátečních podmínek jakými jsou počáteční hodnota počtu jedinců a hlavně velikost populačního růstu může nastat několik možností. Graf se bude ustalovat, až zůstane na jedné hodnotě, nebo může oscilovat mezi několika stavy a nebo se systém bude chovat chaoticky. Všechny tyto možnosti jsou vidět na obrázku 4.1.

Dalším způsobem jak vyjádřit jednodimenzionální dynamický systém je bifurkační diagram. Zobrazení jaké nám poskytuje logistická mapa příliš neobjasňuje jak se systém chová při různých počátečních podmínkách. Pro tento účel je vhodný právě bifurkační diagram, na kterém se na horizontální osu nanáší hodnoty některého vstupního parametru, u příkladu, který je zde uveden se na tuto osu nanáší hodnota populačního růstu $G$, a na vertikální osu se nanáší stavy dynamického systému. Pokud bychom vzali příklad, který je uveden výše, pak tyto stavy jsou hodnoty $x_n$, kterých systém nabývá po určitém počtu kroků. Tyto hodnoty jsou vykreslovány jako body, graf tedy není spojitý, ale je tvořen izolovanými body. Příklad bifurkačního diagramu je vidět na obrázcích 4.2 a 4.3.



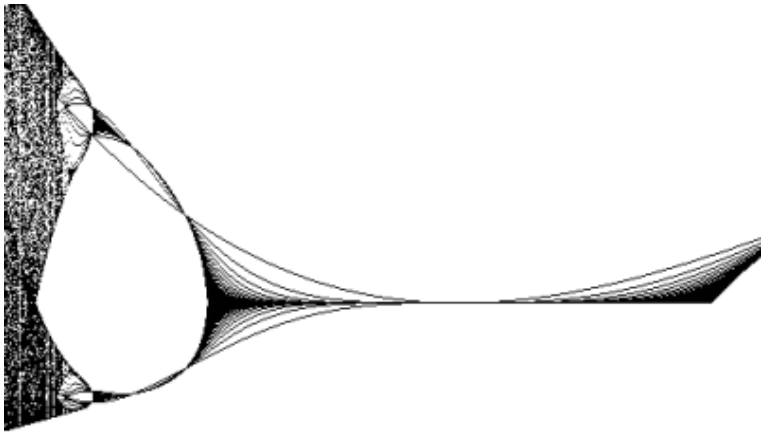

Obrázek 4.1: Logistická mapa pro systémy s různým chováním

Obrázek 4.2: Bifurkační diagram pro kladné hodnoty populačního růstu $G$

### 4.1.2 Dvoudimenzionální dynamické systémy

Vícedimenzionální systémy jsou z hlediska počítačové grafiky zajímavější než jednodimenzionální systémy. Pro vizualizaci vícedimenzionálních systémů je vhodné zobrazovat jejich *orbit*. Tento způsob je vlastně zobrazování stavových vektorů systémů, které chceme vykreslit. Pro některé vícedimenzionální systémy je je vhodné vykreslovat jejich mapu s tím, že vykreslí například počet iterací potřebných pro splnění určité podmínky. Vykreslení orbitu bychom mohli pomocí obecného jazyka definovat takto:

```
Zadání počátečních hodnot x0, y0
Zadání počtu iterací
Zadání hodnot všech parametrů p0 ... pn
for (n = 0 to počet_iterací)
  xn1 = rovnice1(xn, yn, zn, p0 ... pn)
  yn1 = rovnice2(xn, yn, zn, p0 ... pn)
  vykresli_bod(xn1, yn1)
end
```



Obrázek 4.3: Bifurkační diagram pro záporné hodnoty populačního růstu $G$

### Hénonův atraktor

Tento dvoudimenzionální dynamický systém, který je vidět na obrázku 4.4 patří k nejjednodušším systémům s podivným atraktorem. Rovnice, pomocí kterých se počítá následující stav tohoto systému objevil Michel Hénon, když se zabýval studiem pohybu astronomických těles. Hausdorffova dimenze tohoto systému je rovna přibližně 1,261. Rovnice pro výpočet tohoto systému vypadají následovně:

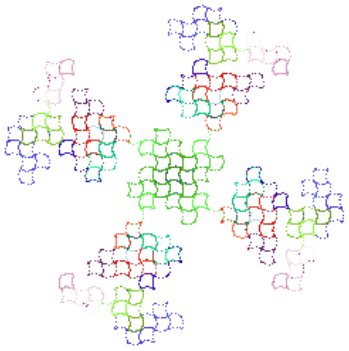
$$\begin{aligned} x_{n+1} &= 1 + y_n - a x_n^2 \\ y_{n+1} &= b x_n \end{aligned}$$

Obrázek 4.4: Hénonův atraktor

### Martin

Tento dynamický systém se poprvé objevil v časopise Computer Recreation, kde ho zveřejnil Alexander Keewatin Dewdney. Pro výpočet tohoto systému se používají dvě velmi jednoduché rovnice. Při zobrazení tohoto systému vznikne zajímavý obrazec, který je vidět na obrázku 4.5. Rovnice pro výpočet tohoto systému mají následující tvar:

$$\begin{aligned} x_{n+1} &= y_n - \sin x_n \\ y_{n+1} &= a - x_n \end{aligned}$$

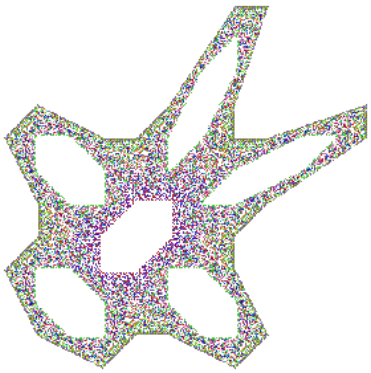


Obrázek 4.5: Dynamický systém Martin

### Gingerbreadman

Název tohoto dvoudimenzionálního dynamického systému s podivným atraktorem vznikl v programu Fractint. Vztahy pro výpočet tohoto systému jsou opět velmi jednoduché. Jediné co můžeme u tohoto systému změnit jsou počáteční parametry $x_0$ a $y_0$. Vztahy pro výpočet tohoto systému, který je možné vidět na obrázku 4.6, jsou zde:

$$\begin{aligned} x_{n+1} &= 1 - y_n + |x_n| \\ y_{n+1} &= x_n \end{aligned}$$



Obrázek 4.6: Dynamický systém Gingerbreadman

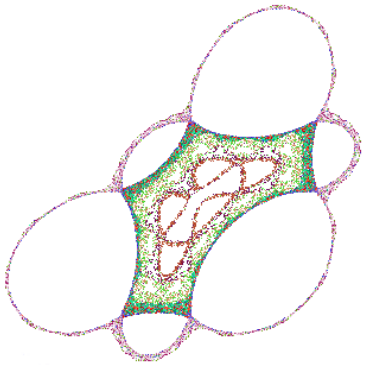
### Hopalong

Dalším typem dynamické systému, který je vidět na obrázku 4.7 je systém nazvaný Hopalong. Tento systém byl poprvé uveden A. K. Dewdneym v časopise Computer Recreations. Hopalong vytváří velice pěkné obrazce. Zajímavé na tomto systému je to, že při výpočtu používá podle hodnoty proměnné $x_n$ různého výpočtu. Ovlivnit tvar systému je možné změnou počátečních podmínek $x_0$ a $y_0$ a také změnou parametrů $a$, $b$, $c$. Celý výpočet se provádí podle následujícího návodu:
Pokud platí $x_n > 0$ pak

$$\begin{aligned} x_{n+1} &= y_n - \sqrt{|b * x_n - c|} \\ y_{n+1} &= a - x_n \end{aligned}$$

Pokud platí $x_n \leq 0$ pak

$$\begin{aligned} x_{n+1} &= y_n + \sqrt{|b * x_n - c|} \\ y_{n+1} &= a - x_n \end{aligned}$$



Obrázek 4.7: Dynamický systém Hopalong

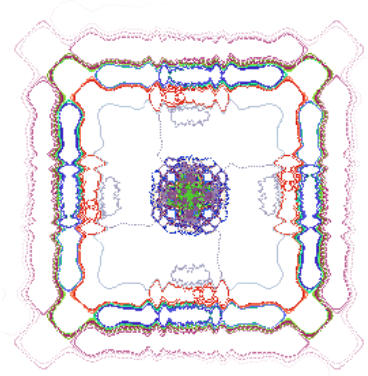
### Chip

Tento dynamický systém s podivným atraktorem použil poprvé Michael Peters ve svém programu HOP. Stejně jako v předchozím příkladě je výpočet tohoto systému závislí na znaménku parametru $x_n$. Výpočet fraktálního obrazce z obrázku 4.8 vypadá následovně:
Pokud platí $x_n > 0$ pak

$$\begin{aligned} x_{n+1} &= y_n - \cos\sqrt{\log|b * x_n - c|} * \arctan\sqrt{\log|c * x_n - b|} \\ y_{n+1} &= a - x_n \end{aligned}$$

Pokud platí $x_n \leq 0$ pak

$$\begin{aligned} x_{n+1} &= y_n + \cos\sqrt{\log|b * x_n - c|} * \arctan\sqrt{\log|c * x_n - b|} \\ y_{n+1} &= a - x_n \end{aligned}$$



Obrázek 4.8: Dynamický systém Chip

### Kamtorus

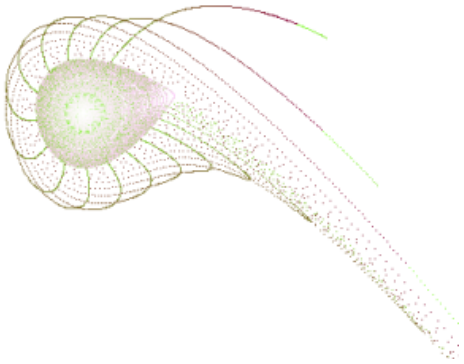
Tento systém nazvaný Kamtorus, který je vidět na obrázku 4.9, je zajímavý v tom, že se nevykresluje pouze jeden orbit, ale hned několik na sobě nezávislých orbitů. Výpočet každého orbitu probíhá podle stejných vzorců i parametrů, mění se pouze počáteční podmínky pro každý orbit. Vztahy pro výpočet tohoto systém jsou zde:

$$\begin{aligned} x_{n+1} &= x_n \cos\alpha + (x_n x_n - y_n)\sin\alpha \\ y_{n+1} &= x_n \sin\alpha - (x_n x_n - y_n)\cos\alpha \end{aligned}$$

Počáteční podmínky:

$$\begin{aligned} x_0 &= k/3 \\ y_0 &= k/3 \end{aligned}$$

Hodnota $k$ je konstantou, která platí pro jeden orbit.

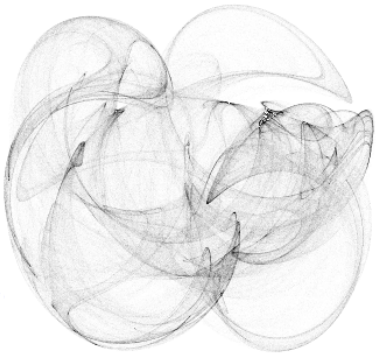


Obrázek 4.9: Dynamický systém Kamtorus

### Pickover

Tento dynamický systém byl pojmenován po svém autorovi Cliffordu Pickoverovi, což fyzik a matematik, který vydal kromě jiného také knihy, ve kterých se zabývá fraktální geometrií. Systém, který nese jeho jméno se počítá pomocí třech rovnic. V těchto rovnicích jsou $a$, $b$, $c$ a $d$ parametry. U tohoto systému se ignoruje z-ová souřadnice, která se používá pouze pro výpočet. Rovnice pro výpočet systému,který je vidět na obrázku 4.10, vypadají následovně:

$$\begin{aligned} x_{n+1} &= \sin ay_n - z_n \cos bx_n \\ y_{n+1} &= z_n \sin cx_n - \cos dy_n \\ z_{n+1} &= \sin x_n \end{aligned}$$



Obrázek 4.10: Dynamický systém Pickover

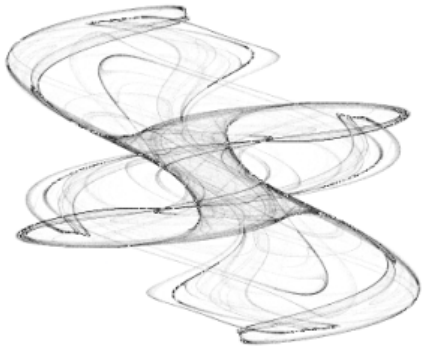
### Latoocarfian

Tento dynamický systém s podivným atraktorem je vlastně pouze zobecněním předchozího dynamického systému Pickover. Zobecnění spočívá v použití obecných funkcí, které ovšem musí mít definiční obor v reálných číslech. Tyto funkce jsou v rovnicích označeny $f_1$ až $f_4$. Rovnice pro výpočet tohoto systému na obrázku 4.11:

$$\begin{aligned} x_{n+1} &= f_1(by_n) + cf_2(bx_n) \\ y_{n+1} &= f_3(ax_n) + df_4(ay_n) \end{aligned}$$

## 4.1.3 Dynamické systémy v komplexní rovině

U dynamických systému je důležité studium jejich chování v čase, to znamená jak mění svůj stav. Pokud bychom chtěli vidět všechny stavy, ve kterých se systém nachází po určitý časový úsek, máme na výběr ze dvou možností. Buď animací postupně všech stavů, nebo můžeme najednou zobrazit celý stavový prostor. Jak jsem již uvedl stav dříve, stav systému je definován stavovým vektorem. Pokud stavový vektor obsahuje jednu, dvě nebo tři složky můžeme je použít pro definici pozice bodu při vykreslování stavu systému.



Obrázek 4.11: Dynamický systém Latoocarfian

Když stavový vektor obsahuje více složek pak můžeme zbývající složky použít pro definici barvy nebo intenzity bodu definovaného prvními složkami vektoru. Tento způsob byl použit u předchozích příkladů. Místo orbitů dynamického systému můžeme vykreslit mapu dynamického systému. Mapa je chápána jako rastrový obrázek, kde barva každého pixelu je definována stavem dynamického systému v bodě. Souřadnice tohoto bodu jsou mapovány do pixelů. V tomto rastrovém obrázku pak tedy můžeme každému pixelu přiřadit pouze a jen jednu komplexní hodnotu. Způsob, který se nejvíce používá při zobrazování dynamických systémů v komplexní rovině je zobrazení podle počtu iterací, který je potřeba, než se rozhodne zda platí podmínka určitého systému.

### Juliovy množiny

Juliovy množiny nesou jméno francouzského matematika Gastona Julii, který se od roku 1917 zabýval analýzou chování komplexní paraboly. Touto problematikou se začal později zabývat i Pierre Fatou. Na společnou práci těchto dvou mužů se prakticky zapomnělo až do 80. let 20. století, kdy se stejnou problematikou začal zabývat jejich žák Benoit B. Mandelbrot. Po Gastonu Juliovi mají jméno Juliovy množiny a Fatouův prach, což je speciální ty Juliových množin, dostal jméno po Pierrovi Fatouovi. Někteří vědci tomuto speciálnímu typu Juliových množin říkají také Cantorův prach.

Výpočet Juliových množin vychází právě z funkce komplexní paraboly a je založen na její postupné iteraci:

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

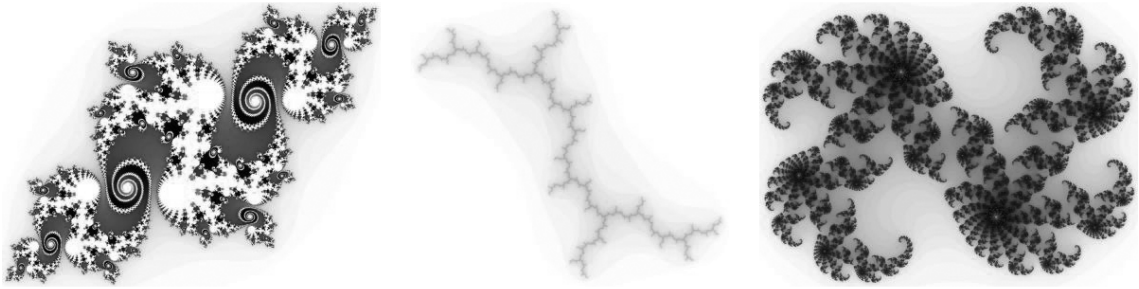
V této rovnici $z_n$ a $c$ leží v komplexní rovině. Počáteční podmínka je $z_0$. Hodnota $c$ je stejná pro celý výpočet a velikost této hodnoty není větší než 2, protože jinak by došlo k divergenci. Juliovy množiny můžeme definovat jako množinu všech $z_0$, pro které posloupnost $z_n$ nediverguje.

Existuje nekonečné množství Juliových množin, které jsou odlišeny pomocí hodnoty $c$, která se používá při výpočtu. Juliovy množiny můžeme rozdělit do tří kategorii, jak je vidět na obrázku 4.12:

1. Výsledný obraz Juliovy množiny tvoří spojitý útvar, to znamená že jakékoliv dva body v této verzi Juliovy množiny můžeme spojit křivkou, která leží celá uvnitř Juliovy množiny. Obraz tvoří jeden útvar, takže neexistují žádné oddělené části.

2. Body, které leží uvnitř Juliovy množiny jsou spojité, ale celkově nemají žádnou plochu. Výsledný obraz vypadá jako křivka. Obraz tvoří opět jen jeden útvar.

3. Body tvořící Juliovu množinu jsou zcela izolovány od všech ostatních bodů. Žádný bod nemá vedle sebe jiný bod, který by tvořil Juliovu množinu. Této katergorii se říká Fatouův nebo také Cantorův prach.




Obrázek 4.12: Typy Juliových množin

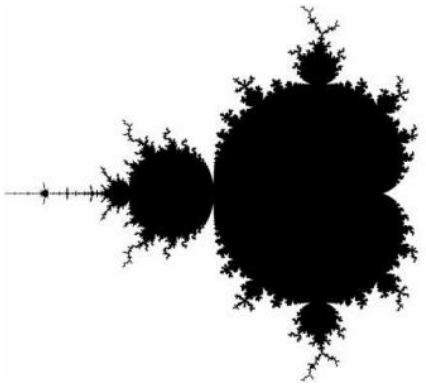
### Mandelbrotova množina

Mandelbrotova množina, kterou můžeme vidět na obrázku 4.13, má jako základ opět relativně jednoduchou rovnici. Výzkumem této rovnice se podrobně zabývali Gaston Julia a Pierre Fatou. Ovšem na jejich práci navázal až na přelomu 70. a 80. letech 20. století vědec jménem Benoit B. Mandelbrot. Ten používal ke své práci, tehdy ještě nerozšířený, počítač. Byl to právě Mandelbrot, který jako první vytvořil a zveřejnil obrazy Mandelbrotovy množiny. I když prvenství Mandelbrota není tak úplně pravdivé, protože zhruba rok předtím než Mandelbrot zveřejnil svůj objev se touto problematikou zabývali pánové Brooks a Matelski, kteří svoje studium zveřejnili až o tři roky později. Ovšem Mandelbrot byl ve svém zkoumání přesnější a šel více do hloubky. Proto také v roce 1982 John Hubbard pojmenoval tento fraktál po Mandelbrotovi.

Při vykreslování Mandelbrotovy množiny na tehdejší výpočetní technice vznikl samozřejmě obrázek s malým rozlišením, který byl pouze černobílý. Ovšem i tak vzbudil velký zájem jak u odborníků tak u laiků. Černobílé obrázky rozlišovaly pouze zda posloupnost v daném bodě konverguje nebo diverguje, tedy zda bod patří do množiny nebo ne. Později s rozvojem techniky se začali do obrázků přidávat barvy, asi nejpoužívanějším způsobem obarvení je podle počtu iterací než se rozhodne zda posloupnost konverguje nebo diverguje.

To zda určitý bod do Mandelbrotovy množiny patří nebo ne zjistíme tak, že porovnáme hodnotu $z_n$ s hodnotou 2. Pokud platí, že $z_n > 2$ pak posloupnost diverguje bod pro který jsme posloupnost počítali do Mandelbrotovy množiny nepatří. Pokud $z_n < 2$ pak provádíme výpočet dál. Před výpočtem si stanovíme maximální počet iterací, které chceme provádět a pokud provedeme výpočet tolikrát na kolik máme nastavenou hodnotu maximálního počtu iterací, pak prohlásíme, že bod do množiny patří. To znamená, že můžeme s určitostí pouze označit ty body, které do množiny nepatří a o těch bodech, u kterých na zadaném počtu iterací tuto skutečnost nezjistíme prohlásíme že do množiny patří. To je ovšem nepřesné,

protože kdybychom provedli o jednu iteraci více pak zjistíme několik dalších bodů, které do Mandelbrotovy množiny nepatří. Proto, čím více iterací, tím přesnější je výsledný obraz.



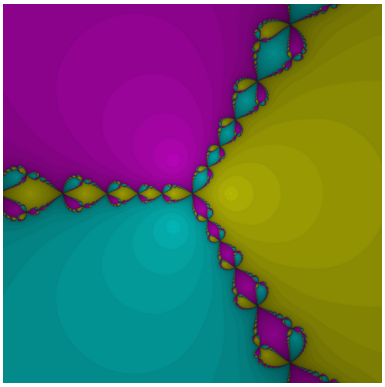
Obrázek 4.13: Mandelbrotova množina

Mandelbrotova množina existuje pouze jedna a to proto, že pro každý počítaný bod je hodnota $z_0$ nastavena na nulu a mění se pouze hodnota parametru $c$, která se nastaví podle souřadnic právě počítaného bodu. Oproti tomu Juliovy množiny mají nekonečně mnoho možností. Při výpočtu Juliových množin se libovolně mění hodnota parametru $c$ a každé hodnotě tohoto parametru odpovídá právě jedna Juliova množina. Při svém studiu Gaston Julia zjistil, že proto, aby mohl určit zda je nějaká Juliova množina spojitá je možné provést iterace komplexní nuly, což je takzvaný orbit nuly. Pokud se však podíváme na Mandelbrotovu množinu, zjistíme, že to není nic jiného než množina bodů, pro které orbit nuly nediverguje. Z toho vyplývá, že Mandelbrotova množina je skutečnosti mapou všech Juliových množin. Takže pokud vezmeme souřadnice jakéhokoliv bodu, který leží uvnitř Mandelbrotovy množiny a dosadíme je místo parametru $c$ do výpočtu Juliovy množiny pak vznikne spojitá Juliova množina.

### Newton

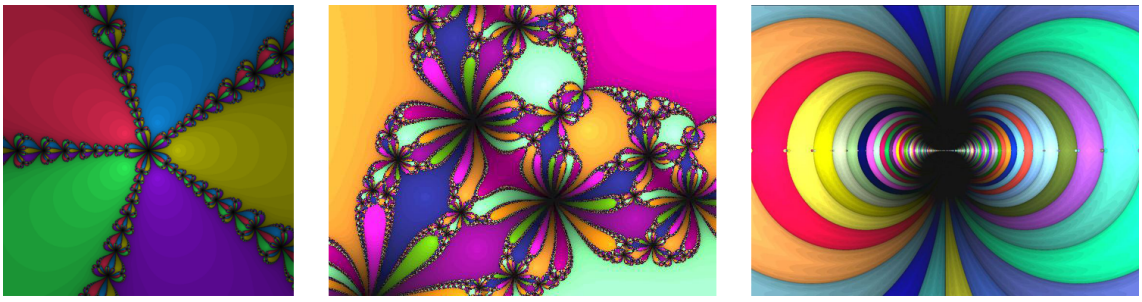
U tohoto typu fraktálů se pro výpočet používá, jak již název napovídá, Newtonova iterační metoda. Pokud vezmeme rovnici $z^3 - 1 = 0$, můžeme spočítat všechny její kořeny. Tyto kořeny se nachází v komplexní rovině. Pokud bychom do rastrového obrázku položeného do komplexní roviny vykreslovali body $z_0$ a barevně odlišili ke kterému kořenu se řešení dostalo, získáme tím mapu atraktorů. Řešení, které bychom si mohli myslet, že vznikne by mohlo vypadat asi tak, že komplexní rovina se rozdělí do tří stejně velkých částí, které budou rozděleny nějakými pravidelnými hranicemi a že tyto tři části se budou setkávat v komplexní nule. Toto se však nestane. Výsledný obraz totiž vytvoří fraktální obrazec, který můžeme vidět na obrázku 4.14. Na tomto obrázku je použita právě metoda obarvování pixelů podle příslušnosti bodu k určitému kořenu. Každá barva odpovídá jednomu kořenu, podle toho jak jsou k němu body přitahovány. Na obrázku opravdu vznikly tři oblasti, které jsou stejně velké, ale hranice mezi tvoří fraktální strukturu, pokud bychom totiž přiblížili obrázek v oblasti přechodu, viděli bychom podobné obrazce původnímu útvaru. To, že hranice tvoří fraktální strukturu značí, že Newtonova iterační metoda, použitá na komplexní rovnice, je značně citlivá na počáteční podmínky.

Ovšem nemusíme se omezovat pouze na tuto jednu rovnici. Newtonovu metodu můžeme použít i na jiné rovnice a výsledky budou také zajímavé. S rostoucím počtem kořenů bude



Obrázek 4.14: Newtonův fraktál

ve výsledném obrázku stále více barev. Možností je opravdu mnoho. Na obrázku 4.15 jsou vidět příklady pro různé polynomy.




Obrázek 4.15: Newtonovy fraktály pro různé polynomy

### Barnsley

Barnsley ve své knize *Fractals Everywhere* uvedl tři nové fraktály. Tyto fraktály jsou v podstatě modifikací Mandelbrotovy množiny a ke každé z těchto modifikací existuje její Juliova varianta. Původní vzorec Mandelbrotovy množiny je změněn a je přidána podmínka pro způsob výpočtu.

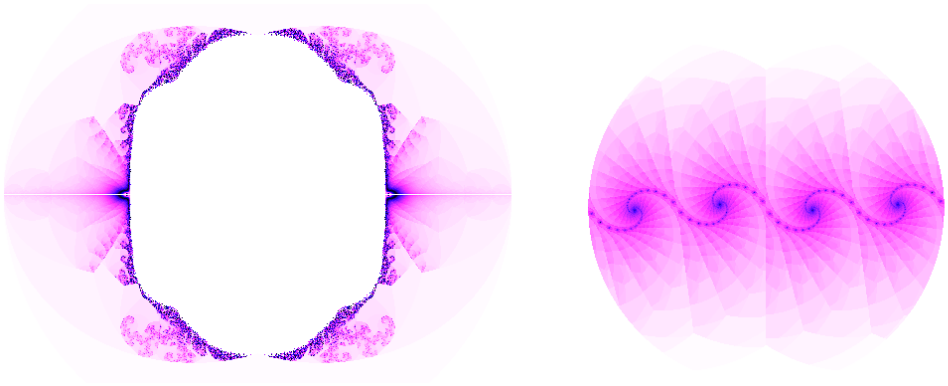
V podmínce pro první typ Barnsleyho fraktálu se vyskytuje reálná složka $z_n$, což jsem označil přidaným indexem $r$. Postup pro výpočet prvního typu Barnsleyho fraktálu, který můžeme spolu s jednou z jeho Juliových variant vidět na obrázku 4.16, je zde:
Pokud $z_{rn} \geq 0$ pak

$$z_{n+1} = c * (z_n - 1)$$

Pokud $z_{rn} < 0$ pak

$$z_{n+1} = c * (z_n + 1)$$

Druhá verze Barnsleyho fraktálu má oproti první variantě jinou podmínku pro zvolení vzorce výpočtu. Zde se zohledňují i imaginární složky jak $z_n$, tak $c$. Tuto imaginární



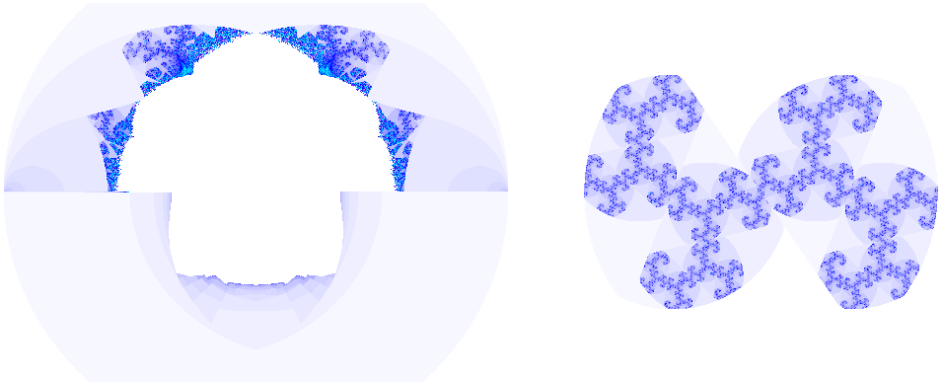
Obrázek 4.16: První typ Barnsleyho fraktálu s jednou z jeho Juliových variant

složku jsem označil indexem $i$. Obrázek 4.17 ukazuje druhou verzi Barnsleyho fraktálu spolu s jednou z jeho Juliových variant. Výpočet probíhá takto:
Pokud $z_{nr} * c_i + c_r * z_{ni} \geq 0$ pak

$$z_{n+1} = c * (z_n - 1)$$

Pokud $z_{nr} * c_i + c_r * z_{ni} < 0$ pak

$$z_{n+1} = c * (z_n + 1)$$



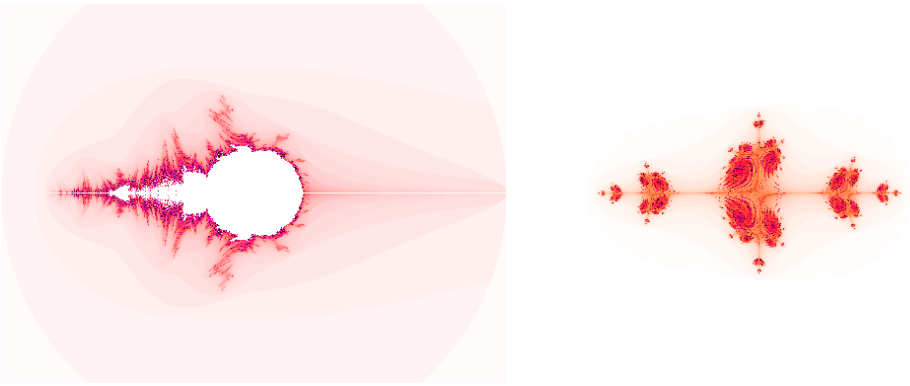
Obrázek 4.17: Druhý typ Barnsleyho fraktálu s jednou z jeho Juliových variant

Třetí fraktál, který Barnsley představil ve své knize má stejnou podmínku pro výběr vzorce jako typ první, ale má složitější výpočet další iterace. Třetí typ Barnsleyho fraktálu spolu s jednou ze svých Juliových variant je vidět na obrázku 4.18. Následuje postup výpočtu:
Pokud $z_{rn} \geq 0$ pak

$$z_{n+1} = z_{rn}^2 - z_{in}^2 - 1 + i * 2 z_{rn} z_{in}$$

Pokud $z_{rn} < 0$ pak

$$z_{n+1} = z_{rn}^2 - z_{in}^2 - 1 + c_r z_{rn} + i * (2 z_{rn} z_{in} + c_i z_{rn})$$



Obrázek 4.18: Třetí typ Barnsleyho fraktálu s jednou z jeho Juliových variant

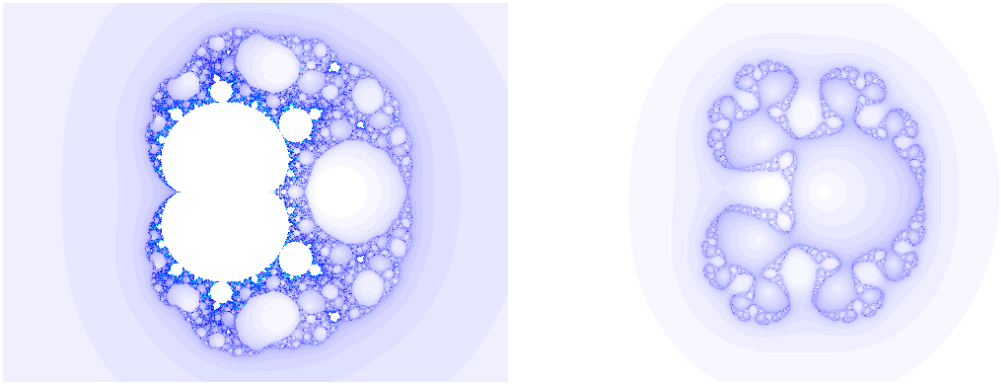
### Magnet

Další typ dynamického systému v komplexní rovině byl objeven při zkoumání magnetických vlastností některých materiálů. Při studiu změny těchto materiálů z magnetických na nemagnetické. Pokud se na materiál podíváme s velkým zvětšením, zjistíme, že se skládá z elementárních magnetů. Když jsou tyto magnety uspořádané, pak je materiál magnetický, pokud magnety uspořádány nejsou, pak materiál magnetický není. To zda budou elementární magnety uspořádány nebo nikoliv záleží na teplotě materiálu. Při nízkých teplotách jsou elementární magnety uspořádány a materiál jako celek je magnetický, pokud je však teplota materiálu vyšší, pak elementární magnety uspořádány nejsou a materiál jako celek tedy magnetický není. Ovšem uspořádanost či neuspořádanost není tak úplně jednoznačná. I přitom když je materiál jako celek magnetický můžeme na něm najít místa, kde magnety uspořádány nejsou a naopak i přes to, že je materiál nemagnetický můžeme na něm najít uspořádané skupiny elementárních magnetů. Magnetismus je tedy závislý na tom jak se na materiál díváme. Materiál mění svůj stav z magnetického na nemagnetický a naopak při určité teplotě. Tato teplota se nazývá Curierova teplota a při této teplotě také nelze určit zda je materiál magnetický nebo nikoliv. Při této teplotě také tvoří elementární magnety fraktální strukturu. Fraktály Magnet vznikly právě při studiu magnetických vlastností materiálů. Stejná fraktální struktura byla však nalezena i u změny skupenství látek. Obecně existují tři skupenství látek a to pevné, kapalné a plynné, ale pokud budeme materiál v průběhu změn sledovat podrobněji, zjistíme, že je jich daleko více. Fraktální strukturu můžeme pozorovat opět při přechodech mezi jednotlivými stavy.

První verze dynamického systému s podivným atraktorem, který leží v komplexní rovině se počítá a vykresluje podobně jako Mandelbrotova množina a podobně jako u Mandelbrotovy množiny i zde existují Juliovy verze tohoto fraktálu. Každý bod v rastrovém obrázku dostane určité komplexní číslo podle jeho souřadnic a testuje se u něj zda posloupnost s počátkem v tomto bodu vede ke konvergenci nebo divergenci a podle toho se mu přiřadí barva. Vzorec pro výpočet Magnetu je složitější než u Mandelbrotovy množiny. Také podmínka divergence je jiná než u Mandelbrotovy množiny. Tam byla podmínka $z_n > 2$, zde se podmínka mění na $z_n > 100$ a více. Také je zde přidána podmínka pro zjištění konvergence k fixnímu bodu. Výpočet první verze fraktálu Magnet, který je spolu s jednou

z jeho Juliových variant vidět na obrázku 4.19, je založen na následujícím vzorci:

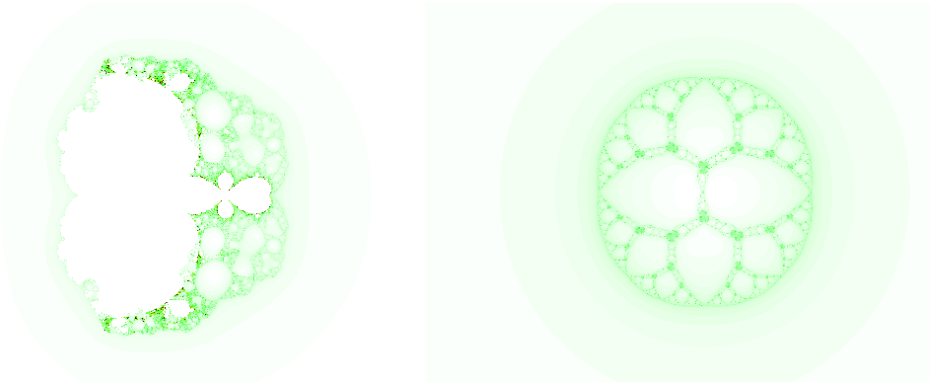
$$z_{n+1} = (\frac{z_n^2 + (c-1)}{2z_n + (c-2)})^2$$



Obrázek 4.19: První typ fraktálu Magnet s jednou z jeho Juliových variant

Druhá verze fraktálu Magnet se počítá úplně stejně, ale s použitím jiného vztahu pro výpočet další iterace. Na obrázku 4.20 můžeme vidět druhou verzi fraktálu Magnet opět s jednou z jeho Juliových variant. Vzorec pro výpočet je zde:

$$z_{n+1} = (\frac{z_n^3 + 3(c-1)z_n + (c-1)(c-2)}{3z_n^2 + 3(c-2)z_n + (c-1)(c-2) + 1})^2$$



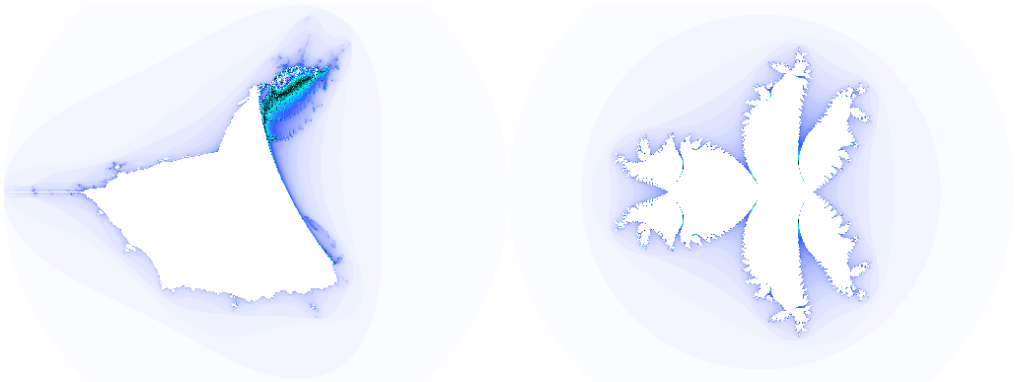
Obrázek 4.20: Druhý typ fraktálu Magnet s jednou z jeho Juliových variant

### Phoenix

Dynamický systém v komplexní rovině nazvaný Phoenix se poprvé objevil v časopise IEEE Transactions on Circuits and Systems. Vzorec, kterým se fraktál Phoenix počítá se skládá ze dvou částí. Druhá část vzorce plní v podstatě jen pomocnou funkci pro výpočet fraktálu. Na počátku výpočtu zda bod patří nebo nepatří do množiny se nastaví hodnoty $z_0$ na na souřadnice počítaného bodu a hodnotu $y_0$ na komplexní nulu. Fraktál Phoenix a jednu

z jeho Juliových variací lze vidět na obrázku 4.21. Vzorec pro výpočet dynamického systému Phoenix vypadá následovně:

$$\begin{aligned} z_{n+1} &= z_n^2 + c_r + c_i y_n \\ y_{n+1} &= z_n \end{aligned}$$



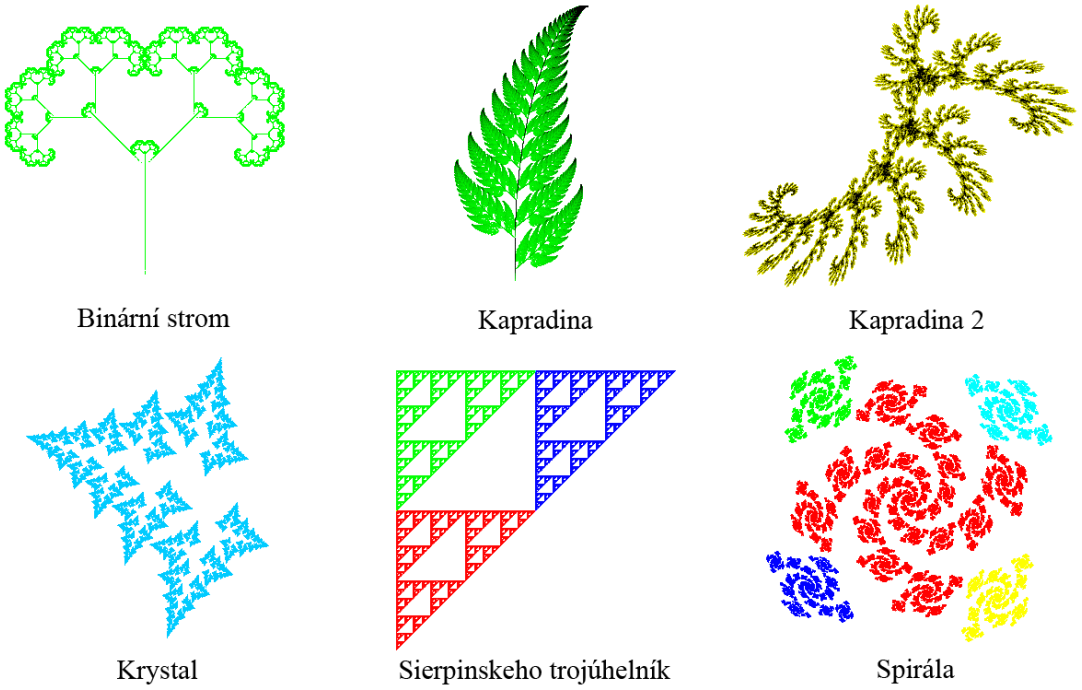
Obrázek 4.21: Fraktál Phoenix s jednou z jeho Juliových variant

## 4.2 Systémy iterovaných funkcí

Pro systémy iterovaných funkcí se používá zkratka *IFS*. Tato zkratka je odvozena z anglického názvu *Iterated Function System*. Prvními vědci, kteří se zabývali IFS byli Demko a Barnsley, kteří také jako první vydali publikace o tomto typu fraktálů [2] a [3]. Systémy iterovaných funkcí se vytváří generativní metodou. Těchto metod existuje více a budou popsány dále v této práci. Systémy iterovaných funkcí jsou definovány sadou transformací a každá z těchto transformací má určitou pravděpodobnost, že bude použita. Některé základní fraktály typu systému iterovaných funkcí jsou vidět na obrázku 4.22.

### 4.2.1 Využití v počítačové grafice

Systémy iterovaných funkcí nachází nejedno uplatnění v oboru počítačová grafika. Asi nejvýznamnější je fraktální komprimace rastrových obrázků. Touto problematikou se zabýval Michael Barnsley. Tato metoda je vhodná pro komprimaci rastrových obrázků, na nichž jsou zobrazeny reálné předměty. Každá transformace je definována osmi čísly, z nichž šest čísel udává posun bodu a zbylá dvě čísla definují jas a kontrast. Metodu fraktální komprimace používá také grafický formát FIF. V porovnání například s formátem JPEG dosahuje stejných, většinou však lepších výsledků z pohledu objemu dat a kvality výsledného obrazu. Formát FIF se však moc nerozšířil, protože fraktální komprimace, kterou tento formát využívá, je výkonově náročný. Dalším důvodem proč formát FIF s fraktálovou kompresí není rozšířen je fakt, že pan Barnsley a jeho spolupracovníci si nechali metodu fraktální komprese patentovat. To znamenalo zavedení poplatků za používání formátu FIF a to vedlo k výběru jiné metody pro kompresi rastrových dat. Velkou výhodou formátu FIF je možnost zvětšení obrazu, aniž by byla vidět jakákoliv ztráta kvality, což o formátu JPEG rozhodně říci nemůžeme.




Obrázek 4.22: Základní fraktály typu IFS

Ať už to byl Demko nebo Barnsley, oba ve svých publikacích vysvětlovali jak používat systémy iterovaných funkcí pro generování rastrových obrazů, které se dnes používají i jako textury v počítačových hrách. Dnes již existuje poměrně velké množství nástrojů pro vytváření rastrových obrázku pomocí systému iterovaných funkcí. Tyto nástroje jsou buď jako samostatné programy, nebo mohou být ve formě nadstaveb na již existující grafické programy.

V počítačové grafice se systémy iterovaných funkcí používají také pro generování trojrozměrných objektů. Pomocí systémů iterovaných funkcí vygenerujeme body v prostoru, které pak podle potřeby můžeme použít jako neorientované částice nebo jako orientované plošky. Systémy iterovaných funkcí se používají pro generování modelů stromů, či jiných rostlin, zde mají systémy iterovaných funkcí dobré výsledky.

### 4.2.2 Algoritmus náhodné procházky

Algoritmus náhodné procházky patří k nejjednodušším způsobům jak generovat systémy iterovaných funkcí. Tento algoritmus spočívá v tom, že se zvolí určitý bod, v rovině nebo v prostoru. Na tom, kde se tento počáteční bod zvolí nezáleží, protože systém se po několika transformacích dostane do svého atraktoru. Většinou se volí první bod v počátku souřadnic. Poté se na tento bod aplikují transformace, které jsou předem zadány. To, která transformace se použije je určeno náhodně podle pravděpodobnosti jakou má každá transformace zadanou. Po provedení transformace se tento bod vykreslí. Výsledný obraz se potom skládá ze všech vygenerovaných bodů. Vzhledem k tomu, že pozice prvního bodu je zvolena libovolně nevykresluje se prvních pár transformací, protože nevíme zda se nacházíme v atraktoru systému. Po prvních pár transformací se systém sám dostane do atraktoru a potom

v něm již zůstane. Jelikož je výsledný fraktál tvořen z vygenerovaných bodů, je zapotřebí provést velké množství iterací. Čím větší počet iterací, tím detailnější bude výsledný obraz, naopak pokud použijeme příliš málo iterací může se stát, že některé části fraktálu se vůbec nevygenerují.

### 4.2.3 Deterministický algoritmus

Deterministický algoritmus se od algoritmu náhodné procházky odlišuje tím, že na vygenerované body se aplikují všechny transformace, které jsou zadány. Tím pádem se nemusí počítat s pravděpodobnostmi jednotlivých transformací a tento algoritmus, jak již název napovídá, je deterministický.

Deterministický algoritmus začíná stejně jako algoritmus náhodné procházky zvolením počátečního bodu. Tento bod můžeme opět zvolit libovolně a volí se bod v počátku souřadnic. Poté se na tento bod aplikují postupně všechny transformace, které jsou zadány. Vznikne tolik nových bodů, kolik je zadáno transformací. Poté se na všechny nově vzniklé body aplikují postupně všechny zadané transformace a takto se postupuje tolikrát, kolik je zadaný počet iterací, které se mají provést. Počet bodů, které jsou v každé iteraci vygenerovány stoupá exponenciálně, proto není nutné provádět tolik iterací jako u algoritmu náhodné procházky.

### 4.2.4 Upravený algoritmus náhodné procházky

Upravený algoritmus náhodné procházky se snaží spojit základní algoritmus náhodné procházky, ze kterého vychází, s deterministickým algoritmem. Postup generování touto metodou začíná opět u zvolení počátečního bodu. Volí se opět libovolně, většinou v počátku souřadnic. Oproti normálnímu algoritmu náhodné procházky se ovšem na tento bod aplikují všechny transformace, které jsou zadány. Mohlo by se zdát, že se tedy spíše jedná o deterministický algoritmus, avšak pro výpočet další iterace se nepoužijí všechny vygenerované. Zvolí se pouze některé, minimálně jeden, na kterých se opět uplatní všechny transformace postupně. Výběr bodů, na kterých se budou dále provádět zadané transformace se dělá pomocí pravděpodobností zadaných u jednotlivých transformací. Při tomto způsobu generování fraktálního obrazce se v jednom kroku vygeneruje více bodů než tomu bylo u základního algoritmu náhodné procházky avšak jejich počet se nezvyšuje exponenciálně, jak tomu bylo u deterministického algoritmu.

### 4.2.5 Algoritmus minima pixelů

Algoritmus minima pixelů využívá toho, že obraz, který generujeme se skládá z konečného počtu pixelů. Nesnaží se proto vygenerovat fraktál naprosto přesně, jak to dělají předchozí algoritmy a k čemu by byl potřeba nekonečný počet bodů a tudíž nekonečný čas pro výpočet. Ve skutečnost je totiž možné vykreslit jen tu část fraktálu která je potřeba a která reprezentuje fraktál s ohledem na rozlišení co nejlépe. Tento algoritmus tedy pracuje přímo s pixely a ne s bezrozměrnými body v ploše. Bezrozměrných bodů je totiž i v omezeném prostoru nekonečně mnoho, protože ať vezmeme jakékoliv dva tyto body, mezi nimi leží nekonečně mnoho bezrozměrných bodů, které mají různé souřadnice. Pokud ovšem pracujeme s pixely, pak se nám na omezenou plochu vejde omezený počet pixelů.

Algoritmus minima pixelů se liší už v tom, že počáteční bod, ze kterého budeme vycházet se nevybírá libovolně. Ke každé transformaci je nejprve dopočten počáteční bod. Tyto vypočtené body, jsou poté použity jako počáteční body celého algoritmu. Na tyto body

se postupně aplikují všechny transformace. Nově vzniklé body se porovnají s již vygenerovanými a pokud je daný pixel již obarvený, vygenerovaný bod se zahazuje. Pokud je nově vygenerovaný bod na pixelu, který není obarvený, pak se obarví a provádí se na něm znovu všechny transformace. U tohoto způsobu generování se neomezuje počet iterací, které se mají provést, protože výpočet skončí jakmile se vyčerpají všechny body, na kterých se mají provádět transformace. Algoritmus je také deterministický, protože se provádí všechny transformace na každý vygenerovaný bod, který je na neobarveném pixelu.

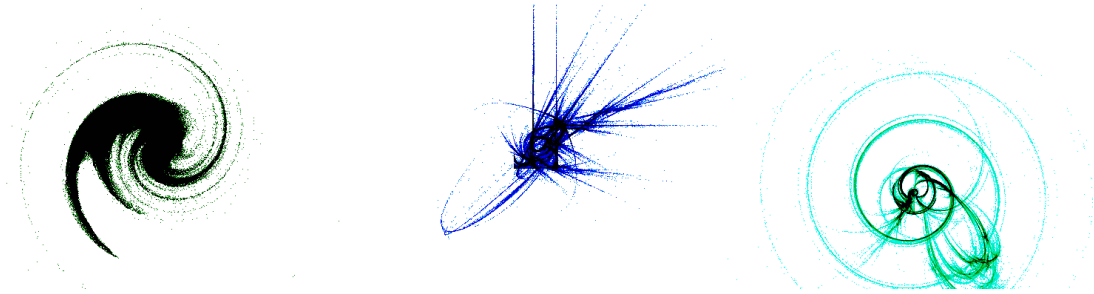
Výhodou tohoto algoritmu je tedy to, že každý pixel v obrázku bude obarven pouze jednou a pokud jednou pixel takto obarvíme, už se pro něj podruhé nebudou počítat žádné transformace. Z toho vyplývá, že tento algoritmus musí skončit v konečném čase. V nejhorším případě budeme pro každý pixel počítat všechny transformace, ale maximálně jednou, takže se nemůže algoritmus dostat do smyčky. Nevýhodou tohoto algoritmu je ovšem nutnost hledání zda nový vygenerovaný pixel není již obarvený. Na ploše by to nebyl problém, avšak nově vygenerované pixely, pro které se mají dále provádět transformace, jsou uloženy ve frontě a jsou přeházeny v pořadí v jakém byly generovány, proto může být hledání zdlouhavé. Tím se doba výpočtu značně prodlouží. Další nevýhodou je to, že nemůžeme generovat pouze určitou výseč fraktálu. Tento algoritmus totiž bude nutně počítat i s pixely, které nejsou vidět. Tuto nevýhodu má i deterministický algoritmus.

### 4.2.6 Flame

Tento druh systémů iterovaných funkcí, jehož příklady můžeme vidět na obrázku 4.23 se od základních typů liší tím, že má některé funkce navíc. Základní algoritmus se používá stejný jako u základního typu a to algoritmus náhodné procházky. Kde se na bod aplikuje náhodná transformace a výsledný bod je obarven v rastrovém obrázku. Rozdílem, který odlišuje základní systémy iterovaných funkcí od fraktálů Flame je použití i nelineárních transformací. U základního typu se používali funkce pouze lineární. Díky těmto transformacím vznikají graficky daleko zajímavější útvary než u základního typu. Dalším velkým rozdílem je symetrie, kterou fraktály typu Flame používají. Symetrii rozeznáváme dvou typů a to zrcadlovou a rotační. Symetrie můžeme dosáhnout prakticky dvěma způsoby. První způsob spočívá v tom, že po vypočtení souřadnic nového bodu se nevykreslí pouze jeden bod, ale několik bodů, které jsou zrcadlově nebo rotačně symetrické. Druhý způsob symetrie se zajistí už při výběru funkcí použitých pro výpočet nového bodu. Tento způsob je lepší, protože nevytváří zcela symetrický obraz, což vypadá příliš uměle. Symetrie je z toho způsobu na obrázku vidět, ale není úplně dokonalá. Funkce pro vytvoření symetrie se vybírají celkem snadno. Středová symetrie potřebuje funkci, která zajistí, aby se bod orotoval kolem počátku souřadnic. Symetrie zrcadlová pak potřebuje funkci, se změnou znaménka jedné z os souřadnic.

## 4.3 L-systémy

L-systémy jsou popsány pomocí regulárních nebo bezkontextový gramatik. Na studiu těchto fraktálů se největší měrou podíleli Aristid Lindenmayer a Przemyslav Prusinkiewicz. Název tohoto systému je odvozen od programovacího jazyka LOGO. V tomto programovacím jazyce se dá pomocí jednoduchých příkazů ovládat takzvaná želva, se kterou se dají kreslit obrazce složené z úseček. Tvorba L-systémů spočívá v přepisování řetězů podle pravidel, která jsou buď zadána předem nebo se mohou také měnit v průběhu vytváření fraktálu. Každému symbolu z řetězce je přiřazen určitý geometrický význam. V případě jazyka LOGO



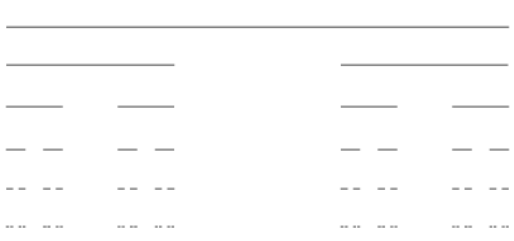
Obrázek 4.23: Fraktály typu Flame

jsou to příkazy pro pohyb a otočení. Vytvářené obrazce začnou být zajímavé po zavedení rekurze, což odpovídá iteraci. L-systémy se využívají pro generování přírodních útvarů, jako jsou stromy nebo rostliny.

### 4.3.1 Cantorova množina

Cantorova množina je jedním z vůbec nejjednodušších fraktálů. Při tvorbě tohoto fraktálu se začíná konstrukcí úsečky s jednotkovou délkou. Tato úsečka se poté rozdělí na tři stejné části. Prostřední část úsečky se odebere, takže nám vzniknou dvě úsečky z nichž každá má třetinovou délku oproti úsečce původní. V dalším kroku se vezmou nově vzniklé úsečky a aplikuje se na ně stejný postup. Takto se pokračuje nekonečněkrát. Poté co se provede nekonečný počet iterací vzniknou izolované body. Takto vzniklý útvar však tvoří pouze část celého fraktálu. Abychom dostali celý fraktál je potřeba vzniklý obraz nekonečněkrát zkopírovat na přímce, která má stejnou orientaci jako původní úsečka. Vzdálenost na přímce mezi těmito kopiemi, při délce původní úsečky jedna, musí být dva.

Pomocí Cantorovy množiny bylo upozorněno na některé vlastnosti algebry a geometrie, které se do té doby ignorovali. Cantorova množina je také důležitá pro svoji jednoduchost, protože se na ní snadno vysvětlují pojmy o fraktálech. Tuto množinu si lze také snadno nakreslit pouze na papír, což by například u mandelbrotovy množiny bylo nemožné.



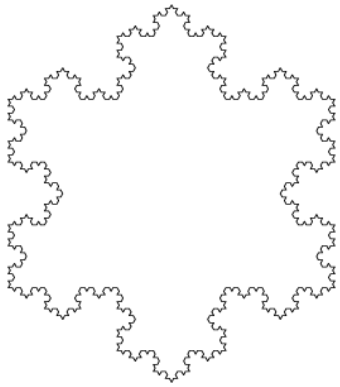
Obrázek 4.24: Cantorova množina

### 4.3.2 Kochova křivka

Pro vytvoření Kochovy křivky musíme podobně jako u Cantorovy množiny vykreslit jednotkovou úsečku. Tuto úsečku následně rozdělíme na tři stejné díly. Prostřední část této rozdělené úsečky nahradíme dvěma rameny rovnoramenného trojúhelníku. Každé z ramen má délku jako odstraněná část. Vzniklý obrazec, má tedy o třetinu větší délku. N všechny takto vzniklé části se znovu aplikuje rozdělení a nahrazení za ramena. Postup vypadá jako na obrázku 4.25.


Obrázek 4.25: Kochova křivka

Pokud změníme počáteční podmínku tak, aby se v první iteraci vykreslil trojúhelník místo přímky, vznikne nám Velice známá Kochova vločka, kterou můžeme vidět i na obrázku 4.26.



Obrázek 4.26: Kochova vločka

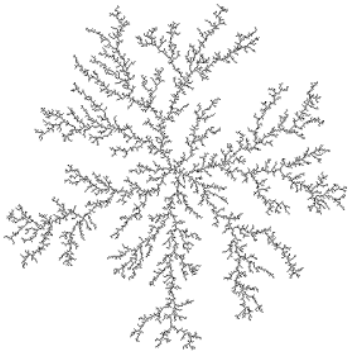
## 4.4 Stochastické fraktály

Fraktály stochastické vnášejí do generování náhodu. Tyto fraktály totiž nejsou tak úplně soběpodobné, jsou totiž jen soběpříbuzné. Při praktickém použití na počítači se ovšem náhoda musí nahradit pouze pseudonáhodou, kterou můžeme na počítači získat. Stochastické fraktály mají ze všech fraktálů nejlepší výsledky, při generování přírodních útvarů. Pokud totiž vezmeme obrazy přírodních útvarů vygenerovaných pomocí L-systémů nebo systémů iterovaných funkcí, zjistíme, že výsledkem jsou symetrické stromy, či jiné rostliny. Ovšem pokud se podíváme do přírody, neuvidíme ani jeden symetrický strom. Strom v přírodě je nepravidelný, jeho větve dorůstají do rozdílných délek a šířek. Proto zavádíme do procesu generování fraktálů náhodu, která nám má věrněji simulovat například růst

rostlin. Podle toho, jak náhodu zapojíme do generování, nám vyjde výsledný obraz a také se podle toho odvíjí Hausdorffova dimenze objektu. Pro generování čísel na počítačích se používají generátory pseudonáhodných čísel.

### 4.4.1 Brownův pohyb

Pokud bychom simulovali Brownův pohyb, vytvářeli bychom tím fraktální obrazec. Brownův pohyb je jev, při kterém jsou částečky nějaké hmoty, pevné, kapalné nebo plynné, roznášeny postupně v jiné hmotě. Například pokud bychom sledovali jak se navzájem promíchávají dvě různě barevné kapaliny nalité do jedné sklenice. Členitost a Hausdorffovu dimenzi můžeme upravovat změnou absolutní velikosti změny. Simulace Brownova pohybu se používá při vytváření modelů například řek. Tato metoda není příliš vhodná pro generování trojrozměrných modelů, protože výsledný model se skládá z izolovaných bodů a grafické vybavení dnešních počítačů je uzpůsobené pro práci s polygony. Fraktál vygenerovaný touto metodou je vidět na obrázku 4.27



Obrázek 4.27: Stochastický fraktál vytvořený simulací Brownova pohybu

### 4.4.2 Posun středního bodu

Metoda posunu středního bodu, která se dá použít jak na plochu tak na prostor, je velmi rozšířenou metodou. Tato metoda se s úspěchem používá v počítačové grafice pro generování přírodní krajiny. To jaký typ krajiny bude vygenerován lze měnit maximální možnou odchylkou při posunu středního bodu. Můžeme generovat krajinu od rovinaté až po skalní útvary. Maximální odchylka středního bodu nám také mění Hausdorffovu dimenzi výsledného útvaru.

### 4.4.3 Spektrální analýza

Další druh stochastických fraktálů je založen na výpočtu Fourierovy řady. Při spektrální analýze se náhodně vygenerují Fourierovy obrazy, jejichž spektrální hustota odpovídá Hausdorffově dimenzi. Poté se provede inverzní Fourierova transformace na vygenerované koeficienty. Výsledkem tohoto postupu je fraktální obrazec. Touto metodou se dají generovat opět krajiny všech různých tvarů. Výhodou této metody je možnost přímého určování Hausdorffovy dimenze a tím i členitosti výsledné krajiny.

# Kapitola 5

# Software

## 5.1 Fractint

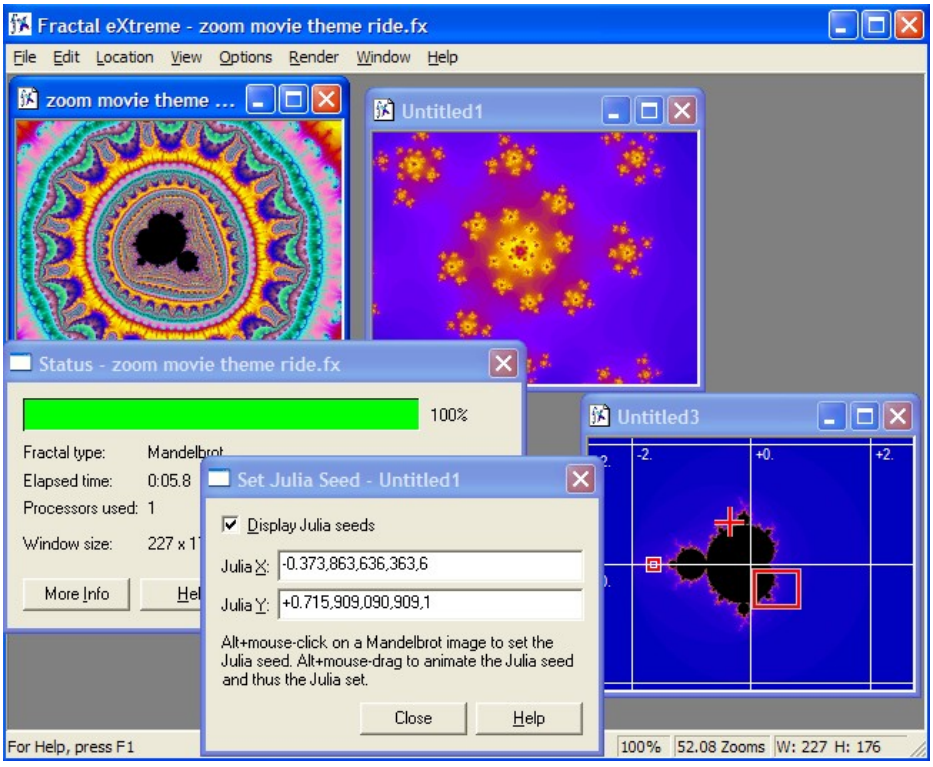
Fractint je jedním z nejznámějších programů, které zobrazují fraktály. Obsahuje v sobě velké množství různých typů fraktálů. Výsledný obraz lze nastavit množstvím nastavení. Výhodou tohoto programu je, že pracuje s celými čísly, ne jako většina programů tohoto typů, které pracují s čísly v plovoucí řádové čárce. To znamená, že není potřeba specializovaný koprocesor pro provádění operací v plovoucí řádové čárce. Zobrazuje fraktály dvourozměrné a také některé trojrozměrné. Fractint má různá nastavení pro mnoho grafických adaptérů. Tento program původně vytvořil Bert Tyler, ale do dnešní doby se na tomto programu podílelo již mnoho lidí. Program je freeware.

## 5.2 Fractal eXtreme

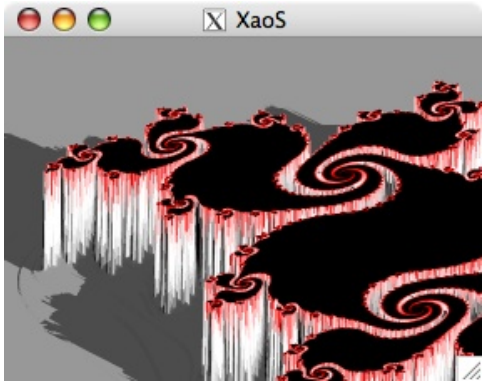
Program Fractal eXtreme používá při vykreslování víceprůchodovou metodu, která umožňuje i pro pomalejší počítače prozkoumávat fraktály. Zobrazují se totiž stále přesnější aproximace vybraného fraktálu. Tato aplikace používá okna pro zobrazování fraktálů a proto je možné zobrazovat více fraktálů najednou. Je zde možnost vytváření vlastních palet pomocí vestavěného editoru. Lze tvořit animace průletu jednotlivými fraktály. Fractal eXtreme má podporu pro více procesorů. Tento program je placený. Ukázka z programu Fractal eXtreme je na obrázku 5.1.

## 5.3 XaoS

Program XaoS je velice dobře optimalizovaná a přibližování ve fraktálu je plynulé. Program v sobě obsahuje výukové animace, které podávají vysvětlení jak o vlastnostech fraktálů, tak demonstrují možnosti programu. Program má široký výběr fraktálů, které může vykreslit. XaoS umožňuje úpravu fraktálu poté co byl vygenerován, například pseudo-3D projekce nebo antialiasing. XaoS má také velké množství obarvovacích metod jak pro pixely které jsou uvnitř fraktálu, tak pro pixely, které do fraktálu nepatří. Z programu se mohou ukládat obrázky s vytvořenými fraktály. Také se dá uložit přímo nastavení fraktálu a později ho vyvolat. Je možnost také ukládat animace průletu fraktálem. XaoS je multiplatformní. Ukázka programu XaoS je vidět na obrázku 5.2.




Obrázek 5.1: Ukázka programu Fractal eXtreme




Obrázek 5.2: Ukázka programu XaoS

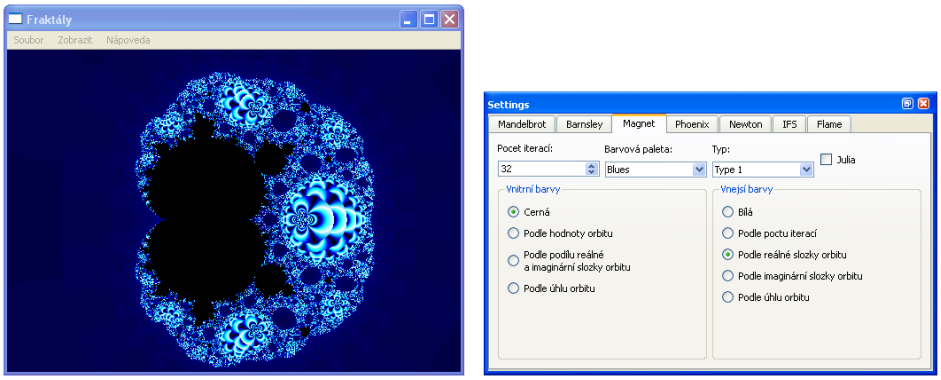
# Kapitola 6

# Demonstrační aplikace

Aplikace, kterou jsem vytvořil, demonstruje některé typy fraktálních obrazců. V této aplikaci je možné interaktivně měnit nastaví parametrů zobrazovaného fraktálu. Měnit lze maximální počet iterací, které se mají vykonat, dále pak můžeme změnit obarvení fraktálů. Dá se změnit sada použitých barev, v programu je to nazváno barvová paleta, ale také se dá změnit styl obarvování jednotlivých fraktálů. Barvové palety se dají měnit u všech fraktálů a styly obarvování jsou různé pro různé typy fraktálů. Dále se u většiny fraktálů může měnit podtyp, u Mandelbrotovy množiny se místo typu mění mocnina $z_n$ ve výpočtu tohoto fraktálu. U typů fraktálů, které mají sobě odpovídající Juliovy varianty se dá zobrazit i takto jejich možnost. U základních typů fraktálů systémů iterovaných funkcí a u fraktálů Flame je možnost ovlivnění barev ještě navíc v povolování či zakazování určité barevné složky a také ve změně barevného přírůstku. Další možností jak interaktivně ovlivnit grafický výsledek je přibližování a oddalování.

Tato aplikace je vytvořena v programovacím jazyce C++ s použitím knihovny Qt. Tato knihovna umožňuje dělat multiplatformní software, takže tento program je možné přeložit a spustit i na linuxu. Pro přeložení tohoto programu je nutné mít nainstalovánu knihovnu Qt verze 4.0 a vyšší. Tento program využívá OpenGL prostředí, které je v knihovně Qt poskytováno. Fraktální obrazce se při výpočtu ukládají do takzvané pixmapy, která se po skončení výpočtu vykreslí. Výpočty fraktálů mohou být časově velmi náročné, obzvláště při zadání většího počtu iterací. Dokud je aplikace spuštěna, pak si každá záložková karta pamatuje svoje nastavení zvlášť. Při změně podtypu fraktálu se nastavení karty nemění, pouze se nově vykreslovaný podtyp zobrazí v základní poloze se základním přiblížením. Stejně tak je tomu při zobrazení Juliových variant. Vygenerované obrázky lze uložit do grafického formátu targa. Nastavení pro výpočet a vykreslování fraktálů je uděláno v samostatném okně, které lze přichytit k oknu, kam se vykresluje, což je nastaveno od spuštění program, dále lze okno uvolnit a pohybovat s ním nezávisle na vykreslovacím okně, toto je vidět na obrázku 6.1, a také lze toto okno s nastavení úplně schovat.

Při výpočtu složitějšího obrazce se může stát, že počítač přestane reagovat. Je to proto, že je zaměstnán výpočtem fraktálu, v tom případě se musí počkat než celý výpočet dokončí. Proto doporučuji jakékoliv změny fraktálů: změna měřítka, barev a podobně, dělat s nastaveným malým počtem iterací a až poté iterace zvýšit pro lepší kvalitu. Také pokud je schovaná část okna a chceme jej posunout na obrazovku, pak při posouvání musí fraktál překreslovat a pokud je nastaven větší počet iterací může to způsobit, že počítač nebude na chvíli odpovídat. Program je dělaný pro rozlišení nejméně 1024x768.



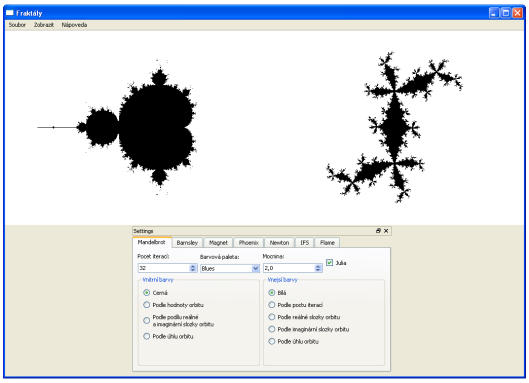

Obrázek 6.1: Aplikace a volné okno s nastavením

## 6.1 Zoom

Všechny fraktální obrazce, které tato aplikace nabízí lze přibližovat a oddalovat. Přibližování je nastaveno na levé tlačítko myši. Funguje tak, že pokud klikneme kamkoliv do okna s fraktálem, pak se jako nový střed zobrazení vezme bod, na který bylo kliknuto a obraz se zvětší o 10%. To samé platí o oddalování, které je nastaveno na pravé tlačítko myši, akorát se po přesunutí na nový střed zobrazení obraz oddálí rovněž o 10%. Po zmáčknutí prostředního tlačítka myši se obraz vrátí na původní zvětšení a pozici.

## 6.2 Juliova varianta

Některé fraktály mají svoje Juliovy varianty. Zobrazení těchto variant lze dosáhnout zatrhnutí políčka *Julia*. Po zaškrtnutí tohoto políčka se okno programu rozšíří o další pixmapu. Na levé polovině je vidět Mandelbrotova verze a v pravé části je Juliova verze. Mandelbrotova verze je zobrazena proto, že je mapou všech Juliových verzí fraktálu. Proto pokud klikneme na levé tlačítko myši v levé půlce okna změníme tím parametry pro výpočet Juliovy verze a ta se změní podle nově nastavených parametrů. Na pravé polovině okna funguje přibližování jak je uvedeno výše. Jak vypadá demonstrační aplikace se zapnutou Juliovou variantou je vidět na obrázku 6.2.



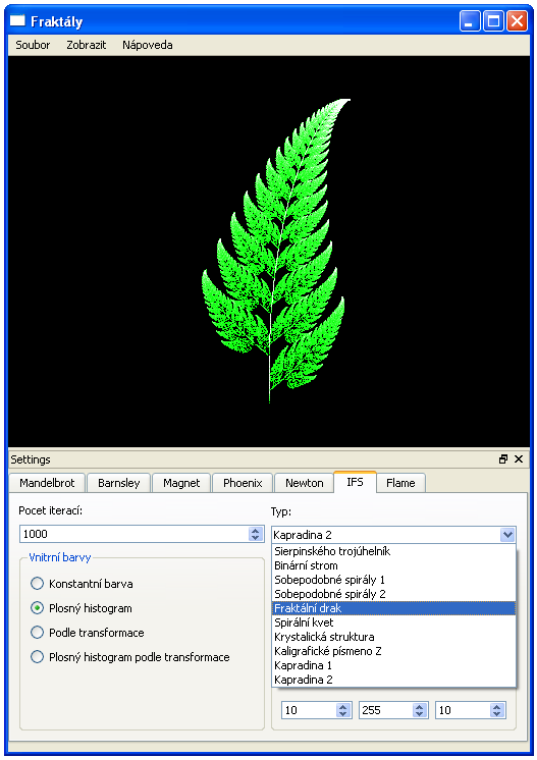
Obrázek 6.2: Aplikace se zapnutou Juliovou variantou

## 6.3 Počet iterací

Pro všechny fraktály lze stanovit hranici počtu iterací, které se mají pro výpočet fraktálu provést. Tento parametr nejvíce ovlivňuje dobu výpočtu fraktálu. Proto doporučuji provádět veškeré změny fraktálu při nízkých hodnotách maximálního počtu iterací a až po nastavení všech ostatních možností zvětšovat počet iterací.

## 6.4 Podtyp

U většiny fraktálu lze měnit jejich podtyp. U fraktálů v komplexní rovině, tedy Mandelbrot, Barnsley, Magnet a Newton znamená změna podtypu úpravu vzorce pro výpočet fraktálu. U IFS a Flame znamená změna podtypu použití jiných transformací. U Mandelbrotovy množiny je výběr podtypu nahrazen zvolením mocniny pro $z_n$ pro výpočet fraktálu. Výběr typu je vidět na obrázku 6.3.




Obrázek 6.3: Aplikace a výběr podtypu

## 6.5 Barvy

Ovlivňování barev lze v aplikace dělat různými způsoby. Lze buďto měnit barvové palety, které použijí dále lze měnit styl obarvování a u IFS a Flame fraktálů lze měnit jednotlivé barevné složky. Různé nastavení barev u fraktálů je vidět na obrázku 6.4.

*Barvové palety* v podstatě mění sadu barev, která se má použít při vykreslování fraktálu. Palety jsou pro všechny fraktály stejné.

Obarvení *podle počtu iterací* je nejpoužívanější metodou. Barva z barvové palety se vybere podle iterace v jaké se rozhodlo o tom, že bod do fraktálu nepatří.

Zbarvení *podle reálné složky orbitu* a *podle imaginární složky orbitu* se dosáhne tak, že po skončení všech iterací se podle reálné respektive imaginární složky vybere z barvové palety příslušná barva.

Obarvení *podle hodnoty orbitu* znamená, že po proběhnutí maximálního počtu iterací se spočítá $|z_n|$ a podle ní se z vybrané barvové palety vybere příslušná barva.

Barva *podle podílu reálné a imaginární složky orbitu* se získá po provedení všech iterací podělením reálné a imaginární složky orbitu $\frac{z_{rn}}{z_{in}}$ a poté se podle výsledku vybere z barvové palety příslušná barva.

Barva *podle úhlu orbitu* se získá vypočtením arctan $\frac{z_{rn}}{z_{in}}$ po provedení všech iterací výpočtu. Podle úhlu se následně z barvové palety vybere příslušná barva.

Obarvení *podle příslušnosti ke kořenu* se podle toho k jakému kořenu systém dojde při výpočtu toho určitého bodu. Pro tento typ obarvení jsou pevně stanoveny barvy, proto na ně nefunguje změna barvové palety.

Zvolení barvy *podle počtu iterací i příslušnosti ke kořenu* se dělá tak, že se vybere barva podle iterace, ve které výpočet skončil a poté se tato barva ještě upraví podle toho k jakému kořenu výpočet došel.

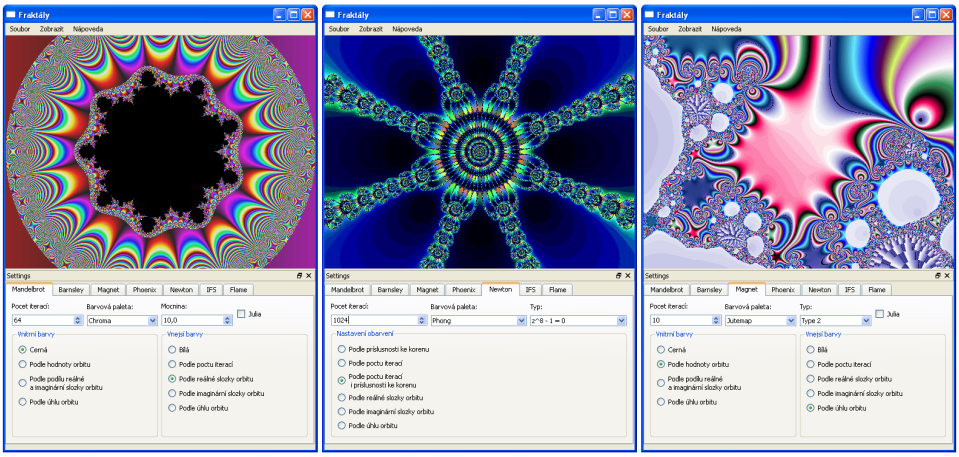
*Plošný histogram* je metoda obarvení taková, při které se barva na pixelu, na jehož souřadnice se bod vygeneroval, zesvětlí. Výsledný obrazec je tedy světlejší na těch místech kam se častěji generují body.

Obarvení *podle transformace* znamená, že bod je obarven podle toho, která transformace na něj byla použita.

*Plošný histogram podle transformace* je metoda, která kombinuje předešlé způsoby. Opět je zobrazován plošný histogram, ovšem barva, která se zesvětluje je vybrána podle použité transformace.

U některých typů obarvení lze povolovat a zakazovat různé *složky barev*. Tímto lze určit, které barevné složky se budou podílet na výsledné barvě.

Změnou *přírůstků barev* lze definovat o kolik se pixel zesvětlí při tvorbě plošného histogramu.




Obrázek 6.4: Aplikace s různým nastavením barev

# Kapitola 7

# Závěr

Cílem této práce bylo vysvětlit základní pojmy z fraktální geometrie, které jsem později použil pro podrobnější popis jednotlivých fraktálních obrazců. V mé bakalářské práci jsem také uvedl v jakých vědních disciplínách je možné se s fraktály setkat a kde je možné fraktály použít. Asi největší uplatnění mají fraktály v počítačové grafice.

Dalším tématem uvedeným v této práci je fraktální historie. Tam jsem mimo jiné uvedl, že fraktální geometrie jako vědní disciplína je velmi mladý obor, který se stále ještě rozvíjí. Z významných osobností, které se zabývaly fraktály bych vyzdvihl Mandelbrota, který je považován za objevitele fraktálů a tím také zakladatele vědy o fraktálech.

V nejobsáhlejší části této práce se zabývám jednotlivými typy fraktálů. Nejzákladnější rozdělení fraktálu je do čtyř skupin. Tyto skupiny se od sebe liší způsobem, jakým se jednotlivé typy fraktálů vytvářejí. U každého typu se dále zabývám podrobnějším rozdělením a jejich vlastnostmi. V této části jsem také uvedl příklady některých fraktálů, které reprezentují jednotlivé typy popsané v této kapitole.

Další část mé bakalářské práce je věnována programům, pomocí nichž se dají vykreslovat fraktální obrazce. Existuje již velké množství programů pro práci s fraktály. Některé programy jsou vytvořeny jen za účelem vizualizace a úpravy fraktálních obrazců. Tyto programy plní pouze estetickou funkci. Ovšem existují i programy, které se používají pro tvorbu modelů nebo textur. Příklady některých nejpoužívanějších programů jsem uvedl i ve své práci.

Výsledkem mé bakalářské práce je také demonstrační aplikace, která umožňuje vykreslovat některé fraktály. Jsou to vybrané fraktály z těch, které jsou popsané v této práci. U fraktálů, které mají svoje Juliovy varianty, je možné vykreslit i tyto jejich varianty. Implementovaným fraktálům lze měnit některé vlastnosti a barevné provedení. Výsledný obrázek vygenerovaný touto aplikací lze uložit na disk.

Prostor pro budoucí vývoj této práce vidím v prozkoumávání dalších typů fraktálů. V této práci jsem se zabýval především jednorozměrnými a dvourozměrnými fraktály, proto bych dále vyzkoušel fraktály vícerozměrné. Další možností by bylo vytváření trojrozměrných modelů fraktálů, které by se dali použít i v jiných aplikacích. Dále je možné do aplikace přidělat více možností obarvování a nastavení více parametrů, kterými se ovlivňuje výsledný vygenerovaný obrazec. Vytváření animací z přibližování, oddalování a pohybu po fraktálu je dalším možným budoucím vylepšením demonstrační aplikace.

# Literatura


[1] Tišnovský Pavel, Seriál fraktály v počítačové grafice[online]. rev. 1. května 2007. Dostupné na URL: http://www.root.cz/serialy/fraktaly-v-pocitacove-grafice/

[2] Barnsley Michael: „Fractals Everywhere“, Academic Press Inc., 1988, ISBN 0-12-079062-9

[3] Barnsley Michael, Devaney R. L., Mandelbrot Benoit B., Peitgenn Heinz-Otto, Saupe Dietmar, Voss Richard: „The Science of Fractal Images“, Springer-Verlag, New York, 1988

[4] Martin Hinner, Historie fraktálů[online]. rev. 29. ledna 2007. Dostupné na URL: http://martin.hinner.info/math/Fraktaly/historie.php

[5] Wikipedia, Benoit Mandelbrot - Wikipedia, The Free Encyclopedia[online]. rev. 4. května 2007. Dostupné na URL: http://en.wikipedia.org/wiki/Beno%C3%AEt_Mandelbrot

[6] Wikipedia, Lorenz attractor - Wikipedia, The Free Encyclopedia[online]. rev. 9. května 2007. Dostupné na URL: http://en.wikipedia.org/wiki/Lorenz_attractor