

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta – Katedra fyziky

Nelineární fyzika a teorie chaosu

Diplomová práce

Vedoucí práce: RNDr. Petr Jelínek, Ph.D.

Autor: Bc. Tomáš Náhlík

Anotace:

Tato diplomová práce se zabývá tématem nelineární fyziky a teorie chaosu od jejího vzniku, přes hlavní osobnosti, až po její uplatnění v různých oborech. V této práci se také nachází část o fraktálech a fraktální geometrii. Součástí této práce jsou též zdrojové kódy různých příkladů.

Abstract:

This thesis deals with nonlinear physics and chaos theory from its beginning, through the main people to its application in various fields. This work has also part of fractals and fractal geometry. There are also source codes of various example.

Prohlašuji, že svoji diplomovou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním mé diplomové práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

V Českých Budějovicích dne

Podpis.....

Na tomto místě bych rád poděkoval svému vedoucímu diplomové práce RNDr. Petru Jelínkovi, Ph.D. za cenné rady a připomínky při zpracování mé práce.

Obsah

1. Úvod.....	6
Teoretická část	
2.1. Chaos a teorie chaosu.....	7
2.1.1. Historie teorie chaosu.....	7
2.1.2. Osobnosti teorie chaosu.....	7
2.1.3. Dynamické soustavy.....	9
2.1.4. Chaotické chování.....	11
2.1.5. Motýlí efekt.....	11
2.1.6. Atraktory.....	12
2.1.7. Logistická funkce.....	17
2.1.8. Dimenze.....	21
2.1.9. Ljapunovovy exponenty.....	26
2.1.10. Poincarého mapy.....	26
2.1.11. Bifurkace a bifurkační diagram.....	27
2.2. Fraktály a fraktální geometrie.....	32
2.2.1. L – Systémy.....	32
2.2.2. IFS.....	35
2.2.3. TEA.....	35
2.2.4. Náhodné fraktály.....	38
2.3. Praktické aplikace teorie chaosu.....	39
2.3.1. Samoorganizace.....	39
2.3.2. Ekonomie.....	40
2.3.3. Astronomie.....	40
2.3.4. Informatika.....	42
2.3.5. Elektronika.....	43
2.3.6. Fyzika.....	45
2.3.7. Biologie.....	48
2.3.8. Chemie.....	55
2.3.9. Teorie klimatu.....	56
Praktická část	
3. Dodatek A: Programování atraktorů.....	58
3.1. Lorenzův atraktor.....	58
3.2. Hénonův atraktor.....	59
3.3. Kings Dream.....	60
3.4. Rösslerův atraktor.....	61
4. Dodatek B: Programování fraktálů.....	62
4.1. Mandelbrotova množina.....	62
5. Dodatek C: Programování diagramů.....	63
5.1. Logistická funkce.....	63
5.2. Bifurkační diagram.....	64
5.3. Lotka – Volterra simulace.....	65
6. Závěr.....	66
7. Literatura.....	67
8. Seznam grafů.....	68
9. Seznam obrázků.....	69
10. Seznam tabulek.....	70

1. Úvod

Tématem této práce je nelineární fyzika a teorie chaosu. Je koncipována jako práce rešeršního charakteru.

V úvodu se seznámíme s počátky a historií teorie chaosu, s hlavními osobnostmi a důležitými pojmy. Na tuto část navazuje pojednání o fraktálech a fraktální geometrii, která je důležitou součástí teorie chaosu. Závěr teoretické části je věnován praktickým aplikacím teorie chaosu a fraktální geometrie. Je zde popsáno několik odvětví, kde se lze s touto teorií setkat.

Za teoretickou částí následuje část praktická, a to ve formě dodatků A-C, kde jsou autorem napsané zdrojové kódy atraktorů, fraktálů, diagramů a simulací různých oborech vědy a techniky. Podrobněji byla v této práci zkoumaná aplikace v biologii, konkrétně simulace Lotka – Volterra, predátor – kořist.

Teoretická část

2.1. Chaos a teorie chaosu

Slovo chaos pochází z řečtiny a v různých odvětvích má různý význam. Ve starém Řecku toto slovo znamenalo primární prázdnotu, nebo kosmický prostor. Počátkem křesťanství došlo k posunu významu ke slovu nepořádek. V metafyzice je opakem zákona a pořádku. V matematice a fyzice se jedná o popis chování jistých nelineárních dynamických systémů. Matematický chaos popisuje neperiodické dynamické chování.

2.1.1. Historie teorie chaosu

Počátky teorie chaosu vztahujeme již k Newtonovým teoriím, kdy prohlásil, že pokud dokonale známe popis jednotlivých částí systému, lze přesně předpovědět budoucnost tohoto systému. Roku 1890 Jules Henri Poincaré studoval problém tří těles. Jedná se o předpovídání systému třech částic vzájemně ovlivněných gravitační silou. Zjistil, že za určitých podmínek je pohyb neperiodický – ani se nevzdaluje a ani se nepřibližuje žádnému pevnému bodu.

Velký zlom v teorii chaosu nastal s příchodem počítačů, které usnadnily výpočty, protože teorie chaosu je převážně založena na opakovaném počítání stejných rovnic jen s různými hodnotami. Jednalo se především o výpočty Edwarda Lorenze z roku 1961, který se zabýval předpověďmi počasí. Jeho výpočty vedly k rozdílným výsledkům, přestože zadávaná data se jen nepatrně lišila. Tento výsledek později nazval jako motýlí efekt.

V dnešní době se teorie chaosu uplatňuje v mnoha odvětvích. Je to například předpovídání počasí, medicína, astronomie a různé jevy, při kterých vznikají turbulence.

2.1.2. Osobnosti teorie chaosu

Jules Henri Poincaré (1854 – 1912)

Byl francouzským filozofem, fyzikem – matematikem. Jeho hlavním přínosem pro teorii chaosu bylo jeho studium problému tří těles, při kterém jako první popsal chaotický dynamický systém. Také zavedl takzvané Poincarého mapy, což jsou řezy fázovým prostorem, které umožňují znázornit i vícerozměrné útvary v prostoru nižší dimenze.



Obrázek 1: J. H. Poincaré

Edward Norton Lorenz (1917 – 2008)

Byl matematikem a meteorologem. Teorii chaosu obohatil o pojem podivný atraktor a o termín motýlí efekt.



Obrázek 2: E. N. Lorenz

Mitchell Jay Feigenbaum (1944 – současnost)

Jako matematický fyzik se zabýval studiem turbulencí tekutin, odkud se dostal ke studiu chaosu. Jeho přínosem do teorie chaosu je studium logistického zobrazení, objev jevu zdvojování period a Feigenbaumovy konstanty alfa a delta.



Obrázek 3: M. J. Feigenbaum

Benoit Mandelbrot (1924 – současnost)

Je matematikem a zakladatelem fraktální geometrie. Je autorem pojmu fraktál.



Obrázek 4: B. Mandelbrot

Michel Hénon (1931 – současnost)

Je francouzským matematikem a astronomem. V teorii chaosu je znám především díky Hénonově atraktoru. Jedná se o jednoduchý diskretní dynamický systém, který vykazuje chaotické chování, je spojován s jistými pozorováními v astronomii.

Aleksandr Michajlovič Ljapunov (1857 – 1918)

Byl ruským matematikem a fyzikem. Jeho přínosem pro teorii chaosu bylo zavedení Ljapunovových exponentů (v některých případech se můžeme setkat s anglosaským přepisem v podobě Lyapunov, tedy např. Lyapunovových exponentů, apod.).



Obrázek 5: A. M. Ljapunov

Otto Rössler (1940 – současnost)

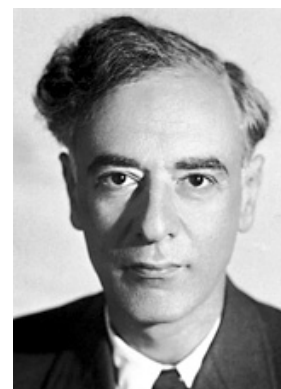
Jako německý biochemik studoval systémy chemických reakcí, při kterých se produkty zpětně mění na reaktanty. Při tomto studiu popisoval chemické reakce pomocí diferenciálních rovnic. Výsledkem tohoto popisu byl Rösslerův atraktor.



Obrázek 6: O. Rössler

Lev Davidovič Landau (1908 – 1968)

Byl ázerbajdžánským fyzikem, který přispěl k rozvoji mnoha oblastí teoretické fyziky. Do teorie chaosu se zapsal zejména studiem turbulence.



Obrázek 7: L. D. Landau

2.1.3. Dynamické soustavy

Dynamickou soustavu (systém) lze popsat pomocí stavového prostoru a dynamických podmínek. Souřadnice stavového prostoru popisují stav systému a dynamické podmínky změny stavů soustavy v čase.

Dynamické podmínky jsou obvykle zadány pomocí soustavy diferenciálních rovnic. Tyto rovnice jsou popisem změn stavového vektoru v čase.

Stavový prostor je množinou všech možných stavů, do nichž se může daný systém dostat. Existuje několik typů stavového prostoru. Těmi jsou stavové prostory konečné, nekonečné a spočítatelné. Konečný stavový prostor je takový prostor, který obsahuje konečné množství stavů. Takovým prostorem může být například balíček hracích karet. Spočítatelný stavový prostor obsahuje nekonečné množství stavů, ale každému stavu lze přiřadit jedno přirozené číslo. Nekonečný stavový prostor je prostor obsahující nekonečný počet stavů, jeho stavové proměnné jsou reálná čísla.

Další možností, jak popsat soustavu, je pomocí stupňů volnosti. Stupeň volnosti udává počet vzájemně nezávislých stavových proměnných. Na číselné ose je jeden stupeň volnosti, v rovině jsou dva, v prostoru tři.

Dynamické soustavy lze rozdělit do několika typů

- lineární
- nelineární
- deterministický
- stochastický

Lineární systém lze charakterizovat tak, že v něm lze uplatnit princip superpozice.

Princip superpozice:

$$f(x) = a, \quad (1)$$

$$f(y) = b, \quad (2)$$

pak:

$$f(x + y) = a + b. \quad (3)$$

V lineárním systému lze předpovědět jeho vývoj i do budoucnosti.

Nelineární systém lze popsat jako systém, ve kterém neplatí princip superpozice. Takový systém je nutné řešit pomocí diferenciálních rovnic a nelze zaručit jeho přesnou předpověď jeho budoucího vývoje.

U *deterministického systému* lze předpovědět, jak se bude v budoucnosti vyvíjet a je snadné ho popsat.

Stochastický systém je systém, kde nelze předpovědět jeho budoucnost [5]. Je náhodný a je nutné se spolehnout pouze na statické funkce.

2.1.4. Chaotické chování

Systém vykazující chaotické chování musí splňovat následující podmínky. Systém musí být citlivý na nastavení počátečních podmínek. To znamená, že i malá změna v počátečních podmínkách vyvolá velkou změnu ve vývoji systému. Příkladem je tzv. motýlí efekt. Další podmínkou je topologická tranzitivita. Funkce:

$$f : X \rightarrow X, \quad (4)$$

je topologicky tranzitivní, jestliže pro každé dvě otevřené množiny:

$$A, B \subset X \quad (5)$$

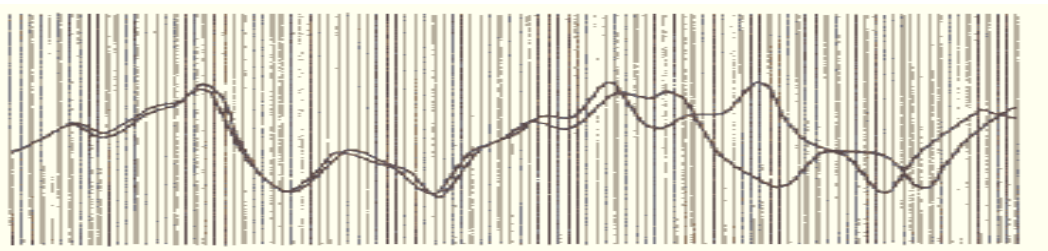
existuje přirozené n takové, že:

$$f^n(A) \cap B \neq \emptyset, \quad (6)$$

Poslední podmínkou je, že periodické orbity systému musí být husté [6, 7].

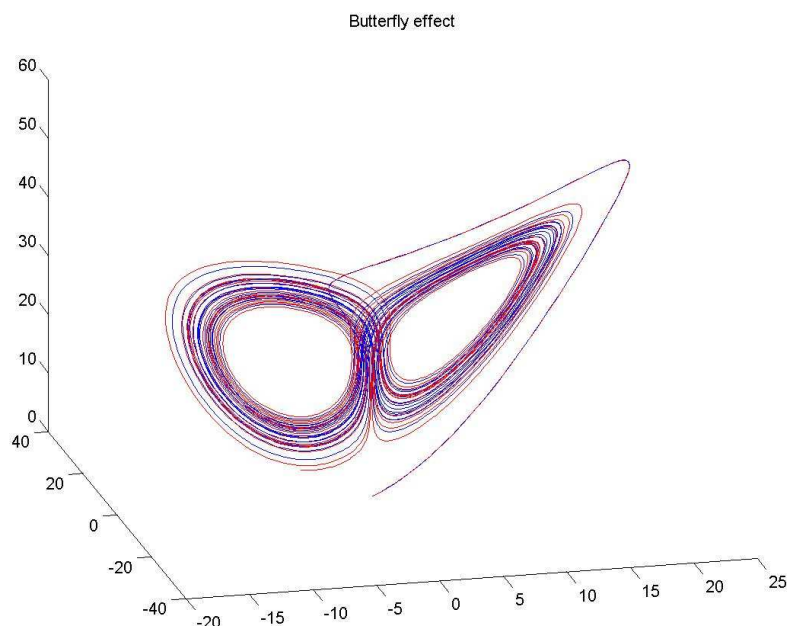
2.1.5. Motýlí efekt (efekt motýlích křídel, butterfly effect)

Termín motýlí efekt se vztahuje k citlivosti počátečních podmínek u chaotického systému. Jako první ho použil Edward Lorenz, a to při studiu atmosféry a pokusech o předpovídání počasí. Při zadávání dat do počítače napsal hodnotu zaokrouhlenou 0,506 místo 0,506127 [8]. I přesto, že odchylka byla malá, vedla k velmi rozdílným výsledkům. Na základě těchto zjištění došel k tomu, že třeba pouhé mávnutí motýlích křídel může naprosto změnit počasí [9].



Obrázek 8: Původní Lorenzův graf

Graf 1: Motýlí efekt - 2 Lorenzovy atraktory lišící se v počátečních podmínkách y-ové souřadnice pouze o 0,05



2.1.6. Atraktory

Vývoj dynamického systému se znázorňuje ve fázovém prostoru, kde každý jednotlivý rozměr představuje jednu stavovou proměnnou dynamického systému. V průběhu vývoje systému lze ve fázovém prostoru pozorovat buď křivku, pokud je čas spojitý, nebo body, pokud je čas diskrétní. Atraktorem se nazývá stav, do kterého daný dynamický systém směřuje.

Rozeznáváme několik typů atraktorů:

- pevné body
- periodické nebo kvaziperiodické body
- chaotické atraktory
- podivné atraktory

Je-li atraktorem *pevný bod*, jedná se o nejjednodušší případ. Příkladem takového atraktoru může být kulička v prohlubni, která se ustálí v nejnižším bodě.

Pokud jsou *periodické* nebo *kvaziperiodické body*, jedná se také o jednoduchý případ. Příkladem může být obíhání vesmírné družice.

Jestliže atraktor systému je *chaotický*, pak nelze vývoj daného systému vůbec předpovědět [7]. Tento systém vykazuje vysokou citlivost na počáteční podmínky. Takovým systémem

může být například kulička na vrcholu kopce. Nikdy nevíme, kterým směrem se bude kutálet dolů.

Podivný atraktor je z hlediska teorie chaosu nejdůležitějším atraktorem. Může vzniknout jen tehdy, je-li daný dynamický systém popsán nejméně třemi diferenciálními rovnicemi. Tento termín poprvé zavedl E. Lorenze a dosud není přesně matematicky definován. Obecně můžeme říci, že daný atraktor je podivný, pokud vykazuje stejné vlastnosti jako fraktály. Z tohoto tvrzení vyplývá, že každý podivný atraktor je fraktálem.

Jako příklad typických atraktorů lze uvést Lorenzův atraktor, Rösslerův atraktor, Hénonův atraktor.

Lorenzův atraktor

Jedná se o trojrozměrnou strukturu, kterou roku 1963 představil Edward N. Lorenz. Popisuje nelineární deterministický dynamický systém odvozený z vynuceného proudění v atmosféře. Je popsána třemi diferenciálními rovnicemi. Tato struktura je spojitá a nikde se neprotíná, viz graf 2.

$$\frac{dx}{dt} = \sigma(y - x), \quad (7)$$

$$\frac{dy}{dt} = x(\rho - z) - y, \quad (8)$$

$$\frac{dz}{dt} = x \cdot y - \beta \cdot z, \quad (9)$$

kde σ je Prandtlovo číslo, ρ Rayleighovo číslo, β nepojmenovaná konstanta, x , y , z jsou souřadnice [1, 2, 3].

Konstanty β , ρ , σ musí být kladná čísla.

Rösslerův atraktor

Rösslerův atraktor je uměle vytvořený systém, který generuje chaos na základě jednoduchých transformací. Jedná se o transformace stretch a fold (= rozprostření a ohyb), viz graf 3. Byl vytvořen roku 1976 a jeho základem je o třináct let starší Lorenzův atraktor.

Je také popsán třemi diferenciálními rovnicemi:

$$\frac{dx}{dt} = -y - z, \quad (10)$$

$$\frac{dy}{dt} = x + ay, \quad (11)$$

$$\frac{dz}{dt} = b + z(x - c), \quad (12)$$

kde x, y, z jsou souřadnice a a, b, c jsou parametry [5].

Hénonův atraktor (Hénonovo zobrazení)

Hénonův atraktor je jeden z dvojdimenzionálních diskrétních atraktorů. Je založen na dvou rovnicích – jedné kvadratické, druhé lineární. Na začátku generování tohoto zobrazení jsou vykreslovány body, které spolu zdánlivě nesouvisejí, viz graf 4.

$$x_{n+1} = y_n + 1 - ax_n^2, \quad (13)$$

$$y_{n+1} = bx_n, \quad (14)$$

kde x, y jsou souřadnice a a, b jsou parametry.

Kings Dream

Atraktory neslouží pouze k osvětlení matematických nebo fyzikálních vlastností, ale mohou mít také estetickou funkci. Jedním z nich je Kings Dream, který byl vytvořen čistě za estetickým účelem, viz graf 5.

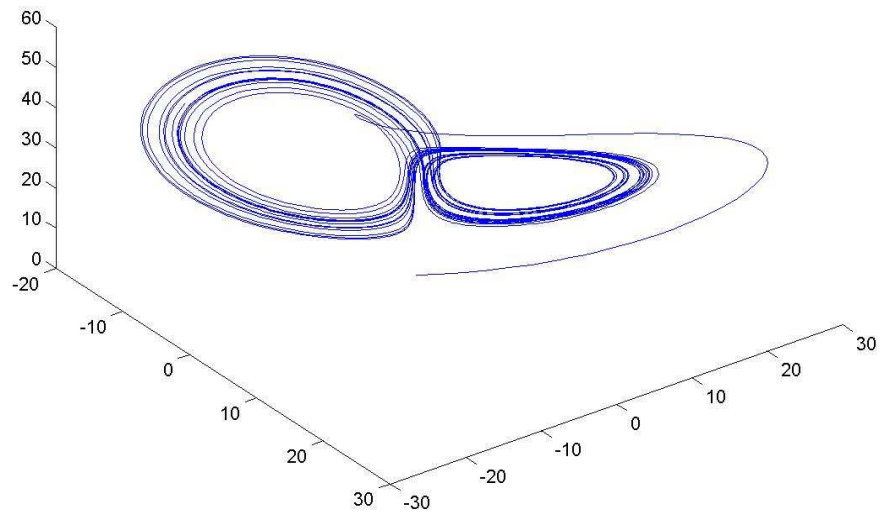
$$x_{n+1} = \sin(y_n \cdot B) + C \sin(x_n \cdot B), \quad (15)$$

$$y_{n+1} = \sin(x_n \cdot A) + D \sin(y_n \cdot A), \quad (16)$$

kde x, y jsou souřadnice a A, B, C, D jsou parametry.

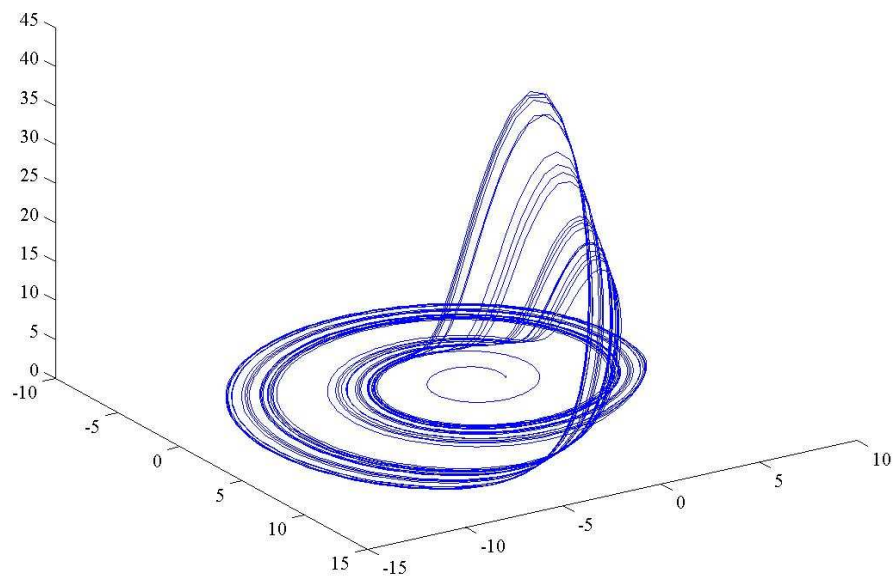
Graf 2: Lorenz Atraktor

Lorenz atraktor

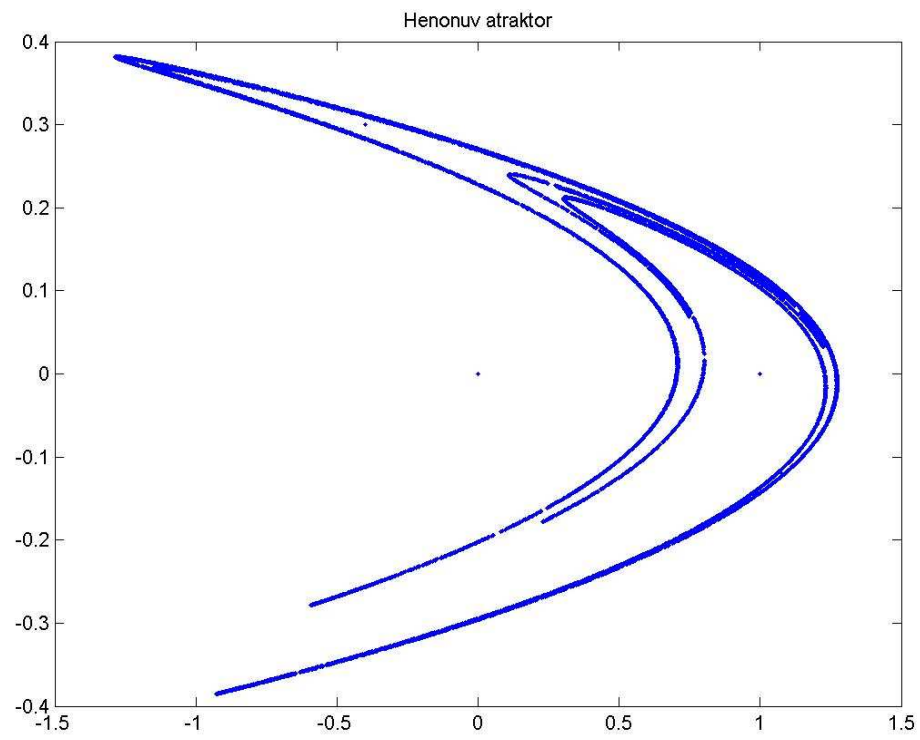


Graf 3: Rössler Atraktor

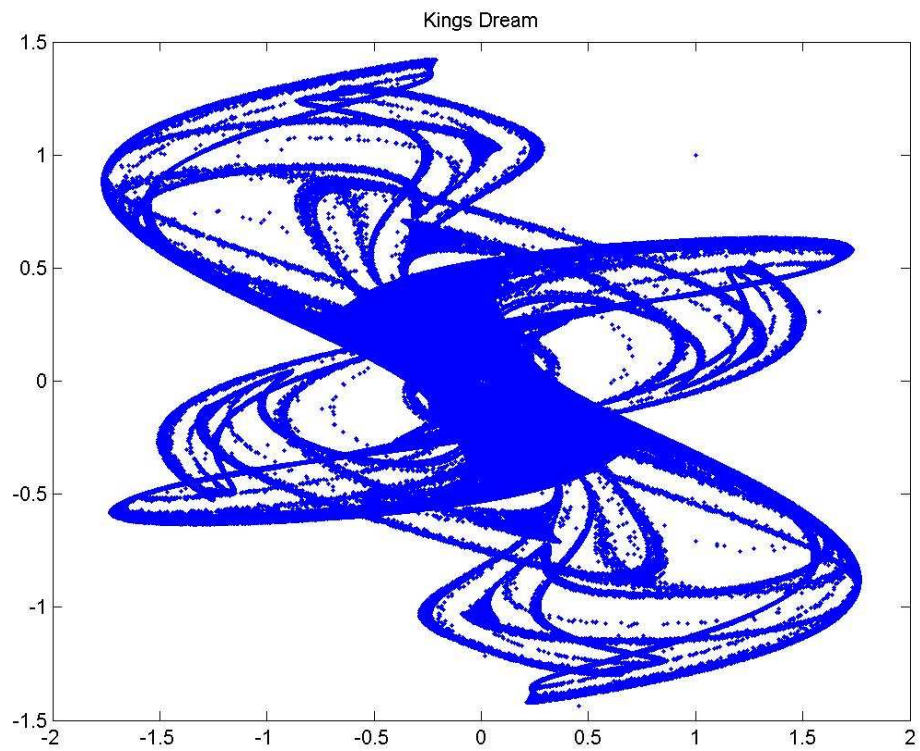
Rössler atraktor



Graf 4: Hénonův Atraktor



Graf 5: Kings Dream



2.1.7. Logistická funkce (logistická křivka)

Logistická funkce je dalším případem chaotického chování. Tato funkce popisuje růst nějaké množiny a při určitých parametrech může dojít k chaotickému chování.

Logistická funkce je matematicky definovaná jako:

$$P(t; a, m, n, \tau) = a \frac{1 + me^{-\frac{t}{\tau}}}{1 + ne^{-\frac{t}{\tau}}}, \quad (17)$$

kde P je velikost; a, m, n, τ jsou reálné parametry.

Důležitým případem je logistická funkce s parametry $a = 1, m = 0, n = 1, r = 1$:

$$P(t) = \frac{1}{1 + e^{-t}}, \quad (18)$$

jejím grafem je sigmoida a je řešením nelineární diferenciální rovnice:

$$\frac{dP}{dt} = P(1 - P), \quad (19)$$

s okrajovou podmínkou $P(0) = 0,5$.

Logistická rovnice se uplatňuje v mnoha oblastech vědy jako je biologie (modelování růstu populace), ekonomie, fyzika (Fermi – Diracovo rozdělení), chemie (reakční kinetika samokatalyzujících reakcí), medicína (růst nádorů), statistika a pravděpodobnost, umělé neuronové sítě a další. Podle oboru, kde se používá, se může lišit její vyjádření.

Chaotické chování lze nejlépe ukázat na biologické aplikaci této funkce. Užívá se zde ve tvaru:

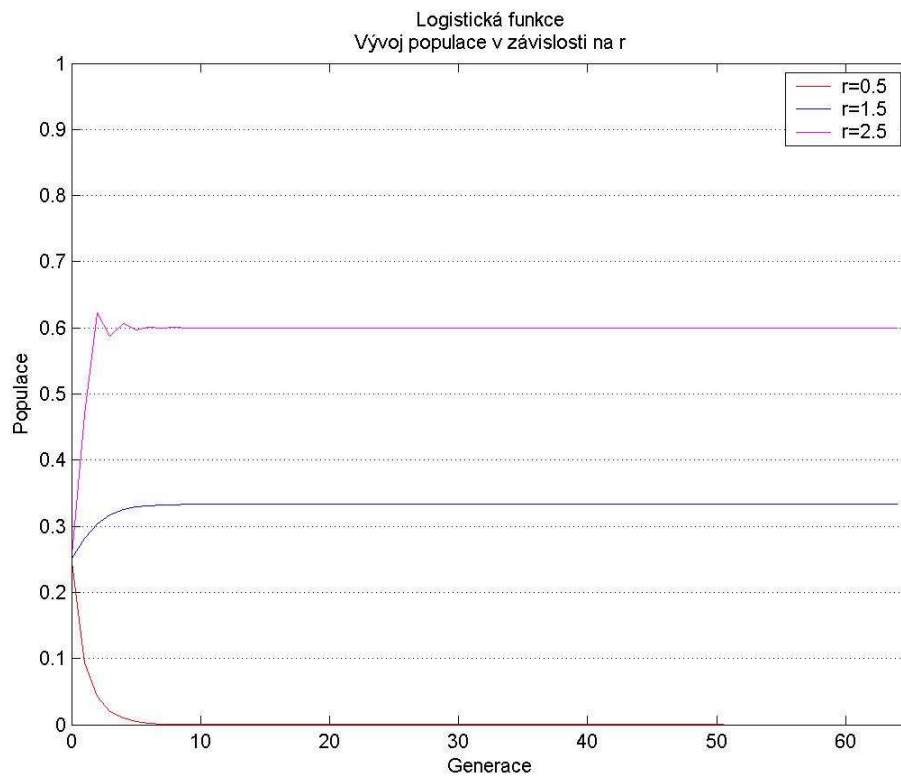
$$x_{n+1} = rx_n(1 - x_n), \quad (20)$$

kde x_n je populace v roce n a má hodnotu mezi 0 a 1, přičemž 0 znamená mrtvou populaci a 1 odpovídá stavu, kdy populace dosáhne svého maxima a plného nasycení prostředí (nemá se kam dále rozmnožovat). Parametr r je řídicí parametr a definuje rychlost růstu.

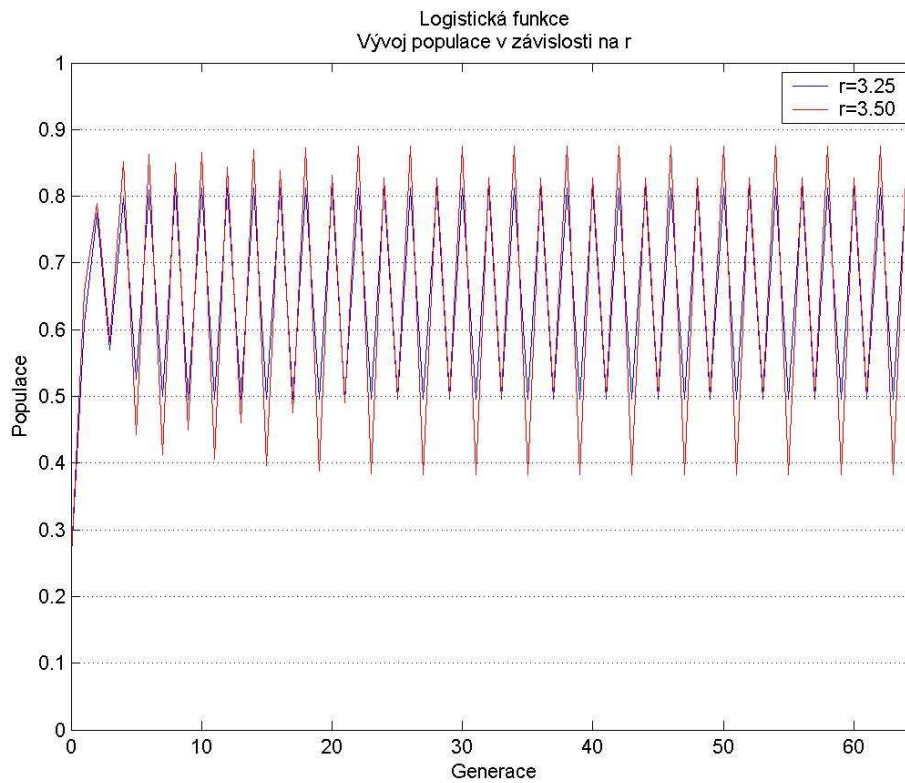
Tabulka 1: Logistická funkce

Parametr r	Stav populace	Graf
$0 < r < 1$	Populace vymře	Graf 6
$1 < r < 2$	Populace se ustálí na hodnotě $\frac{r-1}{r}$	Graf 6
$2 < r < 3$	Populace bude zpočátku oscilovat mezi dvěma hodnotami, později se ustálí na hodnotě $\frac{r-1}{r}$	Graf 6
$3 < r < 1 + \sqrt{6}$	Populace bude oscilovat mezi dvěma hodnotami a bude záviset na r	Graf 7
$3,449 < r < 3,540$	Populace bude oscilovat mezi čtyřmi hodnotami	Graf 7
$3,54 < r < 3,57$	V tomto intervalu bude populace vykazovat chování známé jako zdvojnásobování periody (přechod k chaotickému chování). Populace bude oscilovat mezi počtem hodnot osm, šestnáct, třicet dva ...	Graf 8
$r = 3,57$	Zde je počátek chaosu. Populace neosciluje mezi hodnotami, ale chaoticky nabývá různých hodnot	Graf 8
$3,57 < r < 4,00$	Pro většinu hodnot r je systém chaotický, přesto lze najít hodnoty r , pro které bude systém oscilovat mezi třemi, šesti, dvanácti hodnotami... Vyskytují se zde i oscilace mezi pěti, deseti hodnotami... Tato místa lze nazvat „ostrovy stability“	Graf 8
$r > 4$	Hodnota populace nebude v intervalu $\langle 0;1 \rangle$, nabývá nesmyslných hodnot	Graf 9

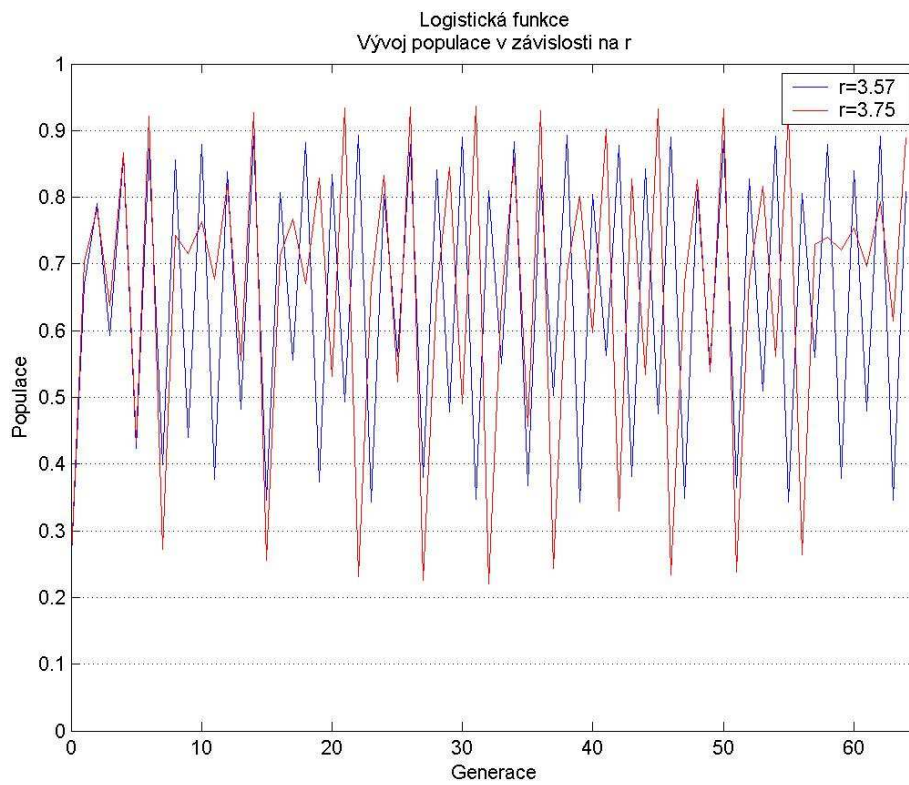
Graf 6: Logistická rovnice pro $r = 0,5; 1,5; 2,5$



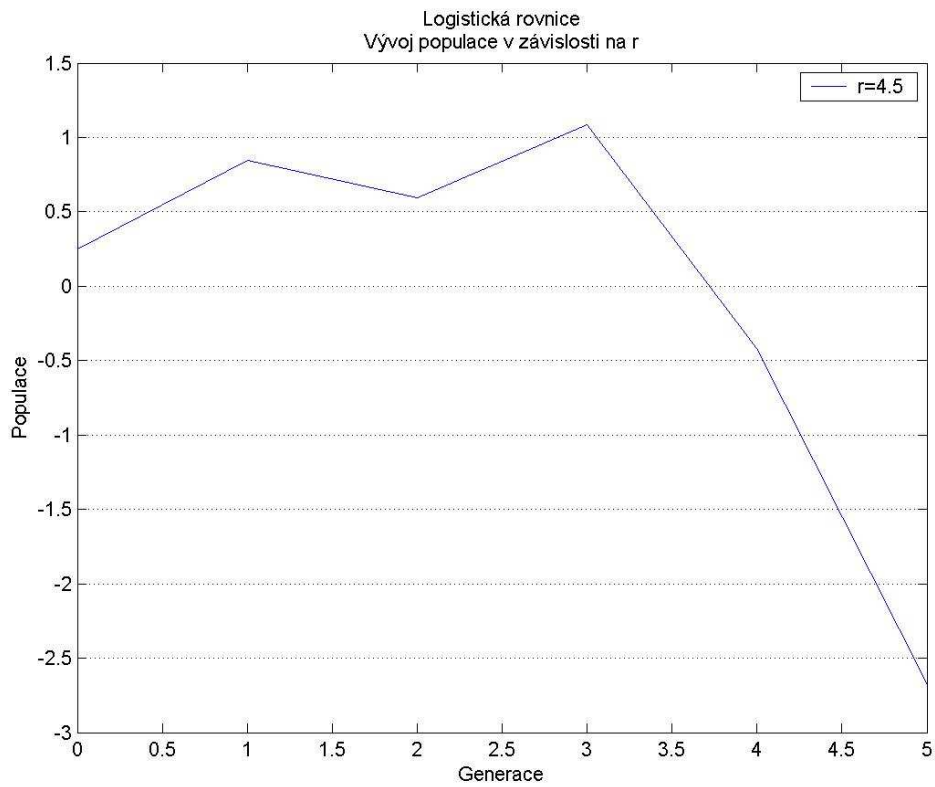
Graf 7: Logistická rovnice pro $r = 3,25; 3,50$



Graf 8: Logistická rovnice pro $r = 3,57; 3,75$



Graf 9: Logistická rovnice pro $r = 4, 5$



2.1.8. Dimenze

Topologická dimenze

K tomu, abychom mohli definovat pojem fraktál, je nejprve nutné říct, co je to dimenze. Dimenze je počet vektorů báze daného prostoru, neboli počet souřadnic, které potřebujeme k jednoznačnému určení bodu v prostoru. Můžeme také říct, že je to počet rozměrů daného objektu. Např.: dimenze roviny je 2, potřebujeme dvě souřadnice k určení bodu. Dimenze krychle je 3, má tři rozměry: šířka, výška, hloubka.

Je-li hodnota dimenze celočíselná, pak řekneme, že toto číslo je topologickou dimenzí daného objektu. Topologická dimenze popisuje geometricky hladké objekty.

Fraktální dimenze

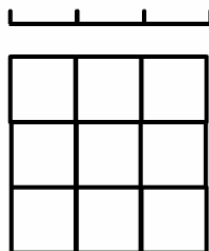
Fraktální dimenze říká, jak moc je daný útvar členitý. Existuje několik typů fraktálních dimenzí:

- Hausdorffova dimenze (Hausdorff dimension)
- mřížková dimenze (box – counting dimension)
- informační dimenze
- korelační dimenze
- Ljapunovova dimenze
- Kolmogorovova entropie (Measure – Theoretic, Kolmogorov – Sinai, KS entropy)

Hausdorffova dimenze (Hausdorff dimension)

Tato dimenze je definována pouze pro soběpodobné struktury. Pro soběpodobné útvary platí, že pokud je pozorujeme při jakémkoli měřítku, vidíme stále se opakující charakteristický tvar.

Příklad: Hausdorffova dimenze: úsečka, čtverec



Obrázek 9: Hausdorffova dimenze

Tabulka 2: Hausdorffova dimenze

Objekt	Počet částí a	Faktor zmenšení s
Úsečka	2	1/2
	3	1/3
	4	1/4
Čtverec	$4 = 2^2$	1/2
	$9 = 3^2$	1/3
	$16 = 4^2$	1/4

$$a = \frac{1}{s^D}, \quad (21)$$

kde D je dimenze.

$$s^D a = 1, \quad (22)$$

$$D = \frac{\log a}{\log \frac{1}{s}}, \quad (23)$$

takto lze spočítat dimenzi libovolného soběpodobného objektu.

Mřížková dimenze (box – counting dimension, Minkowski – Bouligand dimension)

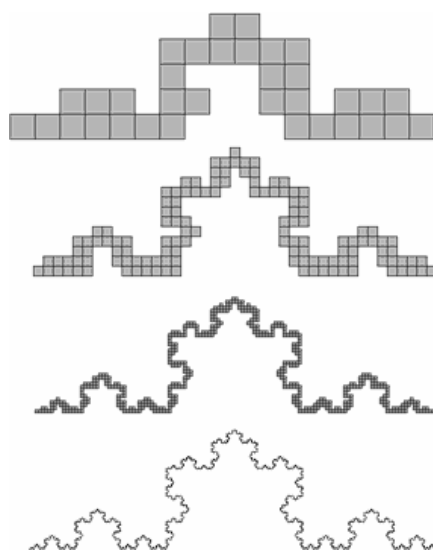
Mřížkovou dimenzi lze – narozdíl od Hausdorffovy dimenze – určovat pro libovolný útvar. Pro ustanovení této dimenze je nutné objekt vložit do mřížky a spočítat počet neprázdných čtverců mřížky.

Tato dimenze je definována jako:

$$\dim_{box}(S) = \lim_{\varepsilon \rightarrow 0} \frac{\log N(\varepsilon)}{\log \left(\frac{1}{\varepsilon} \right)}, \quad (24)$$

kde S je množina (křivka fraktálu), $N(\varepsilon)$ je počet neprázdných čtverců mřížky, ε velikost strany čtverce mřížky.

Příklad: mřížková dimenze: Kochova křivka [2]

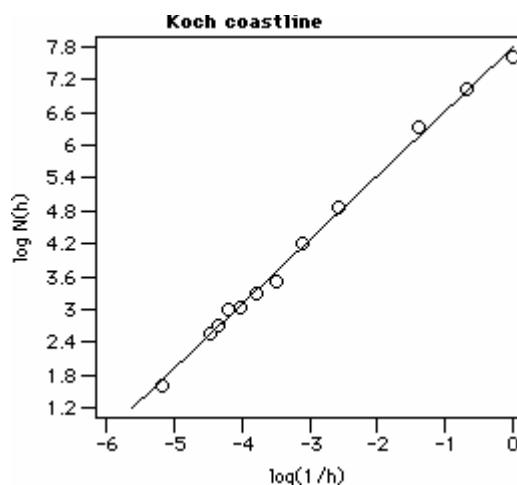


Obrázek 10: Kochova křivka - mřížková dimenze

Tabulka 3: Kochova křivka - mřížková dimenze

Kochova křivka						
log (1/h)	0	-0.69315	-1.38629	-2.56495	-3.09104	-3.49651
log N(h)	7.60837	7.04054	6.32972	4.85981	4.21951	3.52636
log (1/h)	-3.78419	-4.00733	-4.18965	-4.34381	-4.47734	-5.17615
log N(h)	3.29584	3.04452	2.99573	2.70805	2.56495	1.60944

Graf 10: Kochova křivka - mřížková dimenze



Informační dimenze

Mřížková dimenze nepopisuje přesně, jak jsou dané čtverce v mřížce pokryté, poskytuje pouze informaci, zda daný čtverec obsahuje část křivky. Nelze rozlišit, jestli je to jeden

bod, nebo větší část křivky. Tento problém řeší informační dimenze, která je definována na základě přirozené míry a informační funkce.

Informační funkce:

$$I \equiv -\sum_{i=1}^N P_i(\varepsilon) \ln[P_i(\varepsilon)], \quad (25)$$

kde P_i je přirozená míra.

Potom je informační dimenze definována jako:

$$d_{\text{inf}} \equiv -\lim_{\varepsilon \rightarrow 0^+} \frac{I}{\ln(\varepsilon)}, \quad (26)$$

pokud daná limita existuje.

Korelační dimenze

Korelační dimenze je jednou z nejpoužívanějších dimenzí. Je definována jako:

$$d_2 = \lim_{r \rightarrow 0} \frac{\ln \sum_i p_i^2}{\ln r}, \quad (27)$$

kde p_i^2 udává pravděpodobnost toho, že dva náhodně zvolené body zkoumaného objektu leží v i -té množině o velikosti r .

Ljapunovova dimenze

Ljapunovova dimenze d_L je definována na základě Ljapunovových exponentů. Pro její výpočet je nejprve nutné seřadit Ljapunovovy exponenty sestupně podle velikosti:

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \quad (28)$$

Poté je začneme sčítat, dokud součet je větší nebo roven nule:

$$\sum_{i=1}^k \lambda_i \geq 0, \quad (29)$$

kde k je největší přirozené číslo, pro které platí daná nerovnice. Pokud takové k neexistuje, dimenze je rovna 0. Jestliže $k = m$, potom $d_L = m$ a pokud $k < m$, pak:

$$d_L = k + \frac{\sum_{i=0}^k \lambda_i}{|\lambda_{k+1}|}. \quad (30)$$

Kolmogorovova entropie (Measure – Theoretic, Kolmogorov – Sinai, KS entropie)

Kolmogorovova entropie má jistý vztah k Ljapunovovým exponentům. K její definici je nutné fázový prostor pokrýt disjunktními množinami o velikosti r . Jednotlivé množiny pojmenujme i_1, i_2, \dots, i_N . $P(i_1, i_2, \dots, i_N)$ značí sdruženou pravděpodobnost toho, že se trajektorie dynamického systému nalézá v čase t_1 v množině i_1 , v čase:

$$t_2 = t_1 + \Delta t, \quad (31)$$

v množině i_2 , atd. a v čase:

$$t_N = t_1 + (N-1)\Delta t, \quad (32)$$

v množině i_N .

Kolmogorovova entropie je pak definována jako [10]:

$$K = - \lim_{\Delta t \rightarrow 0} \lim_{r \rightarrow 0} \lim_{N \rightarrow \infty} \frac{1}{\Delta t N} \sum_{i_1, i_2, \dots, i_N} p(i_1, i_2, \dots, i_N) \ln p(i_1, i_2, \dots, i_N). \quad (33)$$

Hodnota K	Vlastnosti KS entropie
$K = 0$	pro periodické, kvaziperiodické či stacionární systémy
$K \rightarrow \infty$	pro čistě náhodné systémy
$0 < K < \infty$	pro deterministické chaotické systémy
$K \leq \sum_{i, \lambda_i > 0} \lambda_i$	Pasinova nerovnost (vztah k Ljapunovovým exponentům)

Tabulka 4: Vlastnosti KS entropie

2.1.9. Ljapunovovy exponenty

Ljapunovovy exponenty vyjadřují chování dynamického systému, ať už diskrétního, nebo spojitého, v čase. Jsou měřítkem, jestli blízké orbity konvergují, nebo divergují. Jestliže hodnota Ljapunovova exponentu je záporná, blízké orbity konvergují (jejich vzdálenost s časem klesá), pak systém není citlivý k počátečním podmínkám. Tato hodnota je charakteristická pro disipativní nebo nekonzervativní systém – jako je tlumený harmonický oscilátor. Je-li hodnota Ljapunovova exponentu rovna nule, pak systém je konzervativní a vykazuje Ljapunovovskou stabilitu. Pokud je Ljapunovův exponent kladný, pak blízké dráhy divergují (jejich vzdálenost s časem roste), daný systém je nestabilní, vykazuje citlivost k počátečním podmínkám a lze o něm prohlásit, že je chaotický.

Maximální Ljapunovův exponent je matematicky vyjádřen jako:

$$\lambda = \lim_{t \rightarrow \infty} \frac{1}{t} \ln \frac{|\delta Z(t)|}{|\delta Z_0|}. \quad (34)$$

2.1.10. Poincarého mapy (mapy prvního návratu)

Poincarého mapa je zobrazení fázového diagramu vyšší dimenze v dimenzi nižší. Jedná se o řez nebo projekci vícerozměrného útvaru, kdy jedna či více proměnných je konstantní. Je to jediný způsob, jak lze takovýto útvar znázornit [11].

Poincarého mapa je jeden z nejdůležitějších nástrojů pro zkoumání vícerozměrných systémů. Bývá také označována jako mapa prvního návratu. Posloupnosti Poincarého mapy, které připomínají trajektorie, jsou složeny z bodů. Tyto body znázorňují místa, kdy trajektorie fázového prostoru znovu protne rovinu řezu. Mapu prvního návratu lze označit

za analogii fázového portréту, kdy trajektorii nahradíme posloupností bodů, které spolu souvisejí, pomocí zobrazení P . Jestliže označíme bod Poincarého mapy x_0 , pak platí:

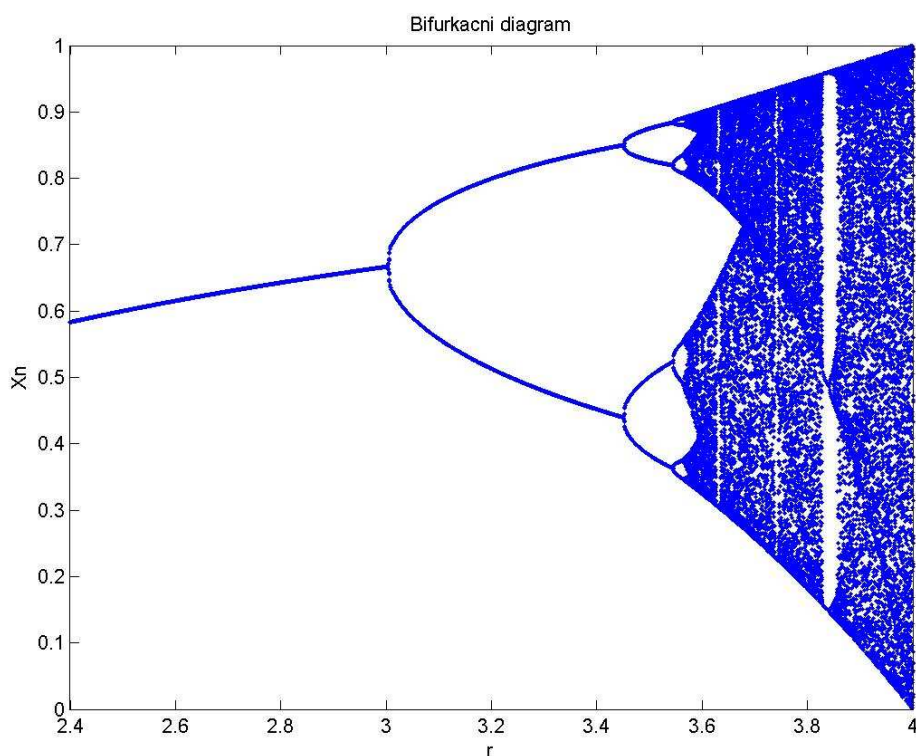
$$x_1 = P(x_0), \quad (35)$$

$$x_2 = P(x_1) = P(P(x_0)) = P^2(x_0) \Rightarrow x_k = P^k(x_0). \quad (36)$$

2.1.11. Bifurkace a bifurkační diagram

Jev, kdy v systému při malé změně řídicího parametru dojde k náhlé a výrazné změně vnitřního stavu systému, nazýváme bifurkací. Pro studium tohoto jevu slouží bifurkační diagram, do kterého se na osu x vynáší hodnota řídicího parametru a na osu y hodnota vnitřního stavu [12].

Graf 11: Bifurkační diagram logistické funkce



Existuje několik základních typů bifurkací:

Lokální bifurkace:

- flip bifurcation (zdvojování periody = period doubling)
- fold bifurcation (sedlo – uzel = saddle – node)
- transcritical bifurcation
- pitchfork bifurcation
- Andronov – Hopf bifurcation (Hopfova bifurkace)

Globální bifurkace:

- homoclinic bifurcation
- heteroclinic bifurcation
- infinite – period bifurcation
- blue sky catastrophe

Lokální bifurkace

Flip bifurkace (Zdvojnásobování periody)

Tato bifurkace je také známá jako zdvojování periody. Jedná se o bifurkaci diskrétního dynamického systému, kdy se změní chování systému tak, že má dvojnásobnou periodu než původní systém. V dalších krocích pak může docházet ke zdvojení již dvojnásobné periody, dochází tak k přechodu od řádu k chaosu. Opačným procesem je bifurkace půlení periody, kdy dochází k přechodu od chaosu k nastolení řádu [7].

Chování, při kterém dochází ke zdvojnásobení periody, vykazuje například funkce:

$$f(x) = \mu - x - x^2. \quad (37)$$

Se zdvojnásobováním periody souvisí princip univerzality a Fiegenbaumova konstanta δ . Princip univerzality říká, že pokud v různých systémech dochází ke zdvojnásobování periody, rozestupy mezi jednotlivými přechody jsou ve všech systémech stejné a jejich vzdálenost je dána Fiegenbaumovou konstantou. Tato konstanta je iracionální číslo (její desetinný rozvoj je nekonečný, neperiodický a nelze jej vyjádřit zlomkem) a její přibližná hodnota je $\delta = 4.6692010609102990\dots$ [13].

Fold bifurkace (bifurkace sedlo – uzel)

Známa také jako bifurkace sedlo – uzel nebo tangenciální bifurkace. Při této bifurkaci dochází ke srážce a anihilaci dvou pevných bodů. Pevný bod je bod, který je pomocí dané funkce zobrazen sám na sebe:

$$f(x) = x. \quad (38)$$

Název bifurkace sedlo – uzel se obvykle používá u spojitých dynamických systémů a bifurkace fold se používá u diskretních systémů.

Tento typ bifurkace vykazuje například funkce [14]:

$$\frac{dx}{dt} = r + x^2. \quad (39)$$

Transkritická bifurkace

Transkritická bifurkace je typ lokální bifurkace, pro který je charakteristické, že rovnovážný bod má vlastní hodnotu, jejíž reálná část prochází nulou. V okolí bifurkace se nachází dva body, jeden stabilní pevný bod a jeden nestabilní bod. Pokud se tyto body setkají, jejich stabilita se prohodí.

Tuto bifurkaci vykazuje například funkce [14]:

$$\frac{dx}{dt} = rx - x^2. \quad (40)$$

Tato rovnice je stejná jako rovnice pro logistickou funkci, ale na rozdíl od logistické funkce mohou r a x nabývat jak záporných, tak i kladných hodnot.

V reálném životě se můžeme s transkritickou bifurkací setkat v podobě rovnováhy mezi zákazníkem a výrobcem. Tuto rovnováhu lze vyjádřit rovnicí:

$$\frac{dx}{dt} = rx(1-x) - px, \quad (41)$$

kde $rx(1-x)$ je růst (obnova) suroviny a px je spotřeba suroviny.

Pitchfork bifurkace (*pitchfork = vidle*)

Tento typ bifurkace se rozděluje na dva typy – superkritickou (supercritical) a subkritickou (subcritical). Pitchfork bifurkace se objevuje u symetrických systémů.

Superkritický typ [14]:

$$\frac{dx}{dt} = rx - x^3, \quad (42)$$

pro $r < 0$ je stabilní rovnováha v $x = 0$, pro $r > 0$ je nestabilní rovnováha v $x = 0$ a dvě stabilní rovnováhy v $x = \pm\sqrt{r}$.

Subkritický typ [14]:

$$\frac{dx}{dt} = rx + x^3, \quad (43)$$

pro $r < 0$ je v $x = 0$ stabilní rovnováha a dvě nestabilní v $x = \pm\sqrt{-r}$, pro $r > 0$ je nestabilní rovnováha v $x = 0$.

Andronov – Hopfova bifurkace

Hopfova bifurkace se – stejně jako pitchfork bifurkace – rozděluje na subkritickou a superkritickou. V této lokální bifurkaci pevný bod dynamického systému ztrácí svou stabilitu jako dvojice komplexních konjugovaných vlastních hodnot linearizace kolem pevného bodu přes imaginární osu v komplexní rovině.

Rozhodnutí, jestli je bifurkace subkritická nebo superkritická, lze provést na základě prvního Ljapunovova koeficientu.

Andronov – Hopfovu bifurkaci lze zapsat jako:

$$\frac{dz}{dt} = z(\lambda + b|z|^2), \quad (44)$$

kde z a b jsou komplexní parametry

$$b = \alpha + i\beta, \quad (45)$$

kde α je první Ljapunovův koeficient.

Pro $\alpha < 0$ je Andronov – Hopfova bifurkace superkritická, pro $\alpha > 0$ je Andronov – Hopfova bifurkace subkritická

Globální bifurkace

Homoclinic bifurkace

Homoklinická bifurkace je globální bifurkace, která se často objevuje při kolizi periodické orbity a sedlového bodu. Také se vyskytuje jako superkritická a subkritická.

Heteroclinic bifurkace

Heteroklinická bifurkace je globální bifurkace, která zahrnuje heteroklinický cyklus (= invariantní množina ve fázovém prostoru dynamického systému, topologický kruh rovnovážných bodů spojující heteroklinické orbity.) Vyskytuje se ve dvou typech – jako rezonanční a transversální bifurkace. Oba typy souvisí se změnou stability heteroklinického cyklu. Resonanční heteroklinická bifurkace souvisí s vytvořením a zánikem periodické orbity. Transverzální heteroklinická bifurkace se vyskytuje při změně stability heteroklinického cyklu. Objeví se, když reálná část vlastní hodnoty jednoho z rovnovážných bodů cyklu prochází nulou.

Infinite – period bifurkace

Globální bifurkace, která se vyskytuje v případech, kdy se objeví dva pevné body na limitním cyklu. Pokud limita parametru dosáhne jisté kritické hodnoty, rychlost oscilace se sníží a perioda se přiblíží k nekonečnu. Bifurkace infinite – period se objevuje v této kritické hodnotě. Za kritickou hodnotou se od sebe dva pevné body na limitním cyklu vzdalují, narušují oscilaci a vytvářejí dva sedlové body.

Blue Sky katastrofa

Blue Sky katastrofa je typ bifurkace periodických orbit. Popisuje druh chování stabilního řešení množiny diferenciálních rovnic, které se mohou pozvolna měnit. Tato bifurkace je charakterizována jak periodou, tak i délkou orbity, která se blíží nekonečnu s kontrolním parametrem blížícím se konečné bifurkační hodnotě. Přičemž orbita stále zůstává v ohraničené části fázového prostoru beze ztráty stability před bifurkačním bodem.

2.2. Fraktály a fraktální geometrie

Za zakladatele fraktální geometrie je považován francouzský matematik Benoît Mandelbrot, který roku 1975 poprvé použil termín fraktál.

Neexistuje žádná matematicky přesná definice pojmu fraktál. Nejblíže k tomu má asi původní Mandelbrotova definice:

Fraktál je množina, jejíž Hausdorffova dimenze je větší než dimenze topologická [15].

Tato definice říká, že fraktál je takový útvar, jehož dimenze je neceločíselná. Existují ovšem také výjimky, kdy za fraktály považujeme i útvary mající celočíselnou dimenzi.

Fraktál lze také obecně definovat pomocí jeho vlastností:

Fraktál je geometrický útvar, který lze rozdělit na části, přičemž tyto části jsou (alespoň přibližně) zmenšené kopie celého útvaru. Fraktál je soběpodobný a nezávislý na měřítku [15].

Tato vlastnost fraktálu se nazývá invariance vůči změně měřítka.

Další obecnou definicí založenou na vlastnostech fraktálů je:

Fraktál je nekonečně členitý útvar.

Fraktály se rozdělují do několika typů [16]:

- L – systémy
- IFS (Iterated Function Systems)
- TEA (polynomické fraktály)
- náhodné fraktály

2.2.1. L – Systémy (Lindenmayerovy systémy)

L – systémy, také často nazývané fraktální křivky, byly poprvé použity v roce 1968 botanikem Aristidem Lindenmayerem k simulaci vývoje mnohobuněčných organismů. Jejich název pochází z anglického LOGO – like turtle. Jedná se o programovací jazyk, ve kterém je místo kurzoru želva (anglicky turtle), kterou uživatel ovládá pomocí základních příkazů. Těmito příkazy jsou pouze pohyb želvy vpřed, vzad, otáčení vlevo, vpravo a schování a vyjmutí ze zásobníku [17]. Za použití těchto příkazů lze generovat různé fraktály.

L – systém je definován jako:

$$G = \{V, S, \omega, P\}. \quad (46)$$

V je abeceda L – systému obsahující množinu symbolů, které jsou nahrazovány použitím pravidel generátoru. Abeceda jsou tedy proměnné. S množina symbolů, které zůstávají konstantní, neaplikuje se na ně generátor. Množina S jsou konstanty. ω je iniciátor, počáteční podmínky fraktálu. P je generátor, což jsou transformační pravidla, která se aplikují na abecedu systému. Tato pravidla mají levou a pravou stranu, kde levá strana se nazývá predecessor – předchůdce, a pravá je nazývána successor – následník.

L – systémy lze rozdělit do několika skupin podle toho, jak se příslušný fraktál generuje. L – systémy se nejčastěji rozdělují na

- parametrické
- kontextově senzitivní (endogenní)
- enviromentálně senzitivní

Pomocí těchto fraktálů lze generovat útvary podobné rostlinám a jiným přírodním útvarům. Díky těmto vlastnostem jsou využívány v biologii, geologii a jiných přírodních vědách.

Příklad: L – Systémy [1]:

Růst řas (původní Lindenmayerův L – systém)

Abeceda V : A, B

Konstanty S : žádné

Iniciátor ω : A

Generátor P : ($A \rightarrow AB$), ($B \rightarrow A$)

0. krok: A
1. krok: AB
2. krok: ABA
3. krok: ABAAB
4. krok: ABAABABA
5. krok: ABAABABAABAAB
6. krok: ABAABABAABAABABAABABA
7. krok: ABAABABAABAABABAABAABABAABAABABAABAAB

Růst posloupnosti je stejný jako růst řas.

Fibonacciho posloupnost

Abeceda V : A, B

Konstanty S : žádné

Iniciátor ω : A

Generátor P : $(A \rightarrow B), (B \rightarrow AB)$

0. krok: A
1. krok: B
2. krok: AB
3. krok: BAB
4. krok: ABBAB
5. krok: BABABBAB
6. krok: ABBABBABABBAB
7. krok: BABABBABABBABBABABBAB

Počet prvků v každém kroku odpovídá jednotlivým číslům ve Fibonacciho posloupnosti: 1 1 2 3 5 8 13 21.....

Cantorova množina (Cantorovo discontinuum)

Abeceda V : A, B

Konstanty S : žádné

Iniciátor ω : A

Generátor P : $(A \rightarrow ABA), (B \rightarrow BBB)$

0. krok: A
1. krok: ABA
2. krok: ABABBBABA
3. krok: ABABBBABABBBBBBBBBBABABBBABA

A znamená kreslení s posunem vpřed, B znamená pouze posun vpřed.



Obrázek 11: Cantorova množina

Obdobným způsobem lze zkonstruovat také například Sierpinského trojúhelník, či rostlinám podobné fraktály.

2.1.2. IFS (Iterated function system)

Systém iterovaných funkcí je matematická metoda k vytváření fraktálů. Takto vytvořené fraktály jsou vždy soběpodobné. Ačkoli ze své podstaty mohou mít IFS fraktály libovolnou dimenzi, obvykle se modelují jako 2D objekty. IFS byly poprvé představeny Johnem Hutchinsonem v roce 1981.

Existují dva způsoby vytváření IFS fraktálů. Jeden je stochastický a druhým způsobem je metoda založena na prosté iteraci.

Stochastický IFS fraktál je generován od jednoho bodu, na který aplikujeme množinu transformačních pravidel. Transformační pravidlo se z množiny vybere na základě náhody, kdy jednotlivá pravidla z množiny mají přiřazenou svojí pravděpodobnost. Součet pravděpodobností všech pravidel je 1. Takto se vypočítá nový bod, který se vezme jako počátek pro další transformaci. Tento postup bývá také nazýván jako chaotická hra (chaotic game).

Metoda prosté iterace pro generování IFS fraktálů se skládá ze základního obrazce a jeho transformovaných kopií. To znamená, že pokud chceme vytvořit IFS fraktál, vezmeme nějaký základní obrazec, uděláme jeho kopii, kterou transformujeme, a následně oba obrazce sloučíme. V dalším kroku totéž zopakujeme.

2.1.3. TEA (polynomické fraktály)

Jedná se o fraktály, které jsou popsány nějakou polynomickou funkcí; patří sem například Mandelbrotova množina a Juliovy množiny. Konstrukce těchto fraktálů je založena na mapování přitažlivosti pro různé oblasti nelineárního systému.

Principem výpočtu je testování různých hodnot dosazováním do rovnice. Na tento výsledek opět aplikujeme stejné pravidlo. Polynomické fraktály jsou množiny komplexních čísel, pro které hodnoty funkce postupně konvergují ke konečné hodnotě. Jestliže atraktor viditelně směřuje do oblasti přitažlivosti, je výpočet ukončen a příslušný bod obarven v závislosti na počtu iterací. Pokud atraktor do této oblasti nesměřuje ani po maximálním počtu iterací, příslušný bod obarvíme implicitní barvou.

Přestože rovnice těchto fraktálů jsou často jednoduché, jejich výpočet je náročný, závislý na výpočetní síle a počtu iterací. Čím kvalitnější chceme výsledek, tím větší nároky klademe na výpočet. Výpočet nelze nijak urychlit, protože bez dosažení daného bodu do

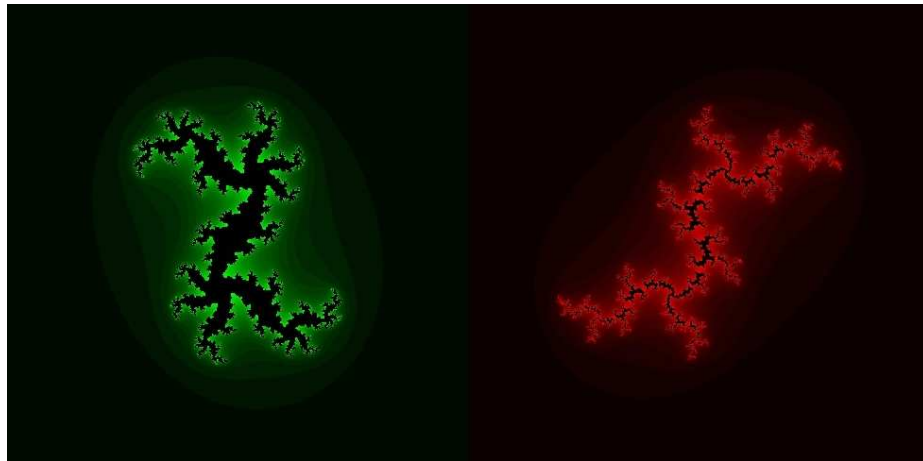
rovnice a provedení příslušného počtu iterací nelze rozhodnout, zda daný bod spadá do oblasti přitažlivosti.

Juliovy množiny

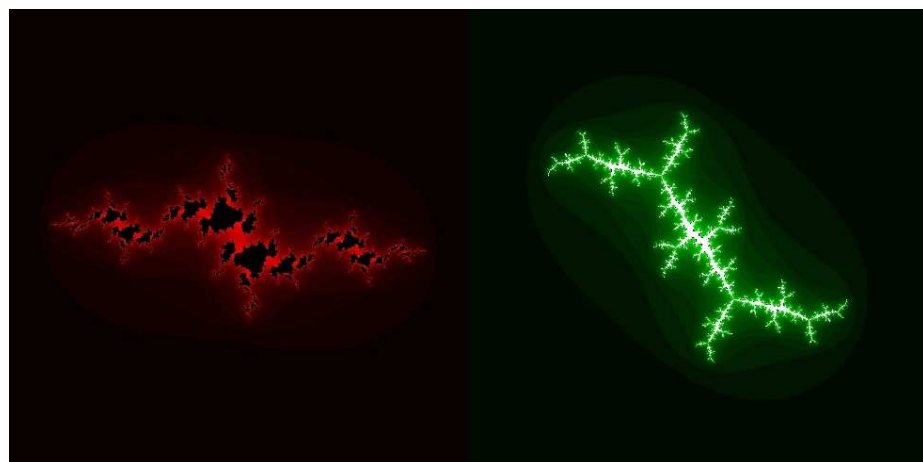
Jejich autory jsou Gaston Julia a Pierre Fatou, kteří popsali jejich konstrukci roku 1918 [18]. Základem je rovnice:

$$z_{i+1} = z_i^2 + c, \quad (47)$$

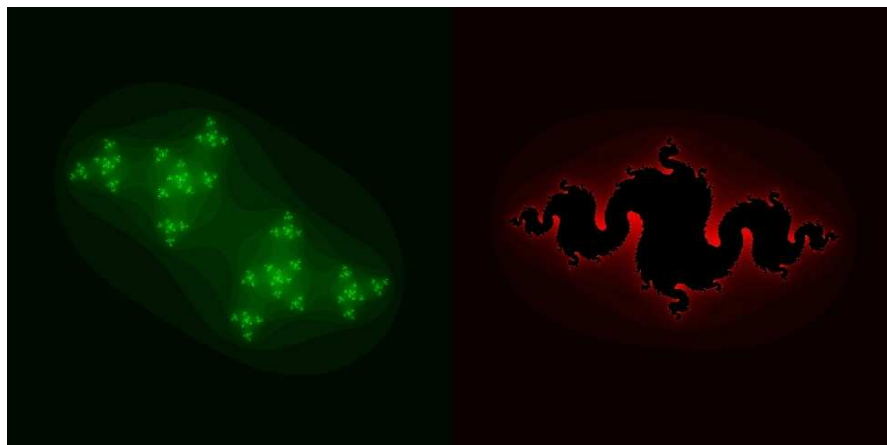
kde z a c jsou komplexní čísla, přičemž c je parametr charakterizující danou Juliovu množinu. Podle toho předpisu ověříme, jestli hodnota z nediverguje. Při praktickém výpočtu stačí, jestliže $|z| > 2$.



Obrázek 12: Juliovy množiny pro $c = 0, 40 + 0, 32i$ (zelená) a $c = 0, 0 - 0, 8i$ (červená)



Obrázek 13: Juliovy množiny pro $c = - 1, 10 + 0, 25i$ (červená) a $c = - 0, 12 + 0, 90i$ (zelená)



Obrázek 14: Juliovy množiny pro $c = -0,45 + 0,25i$ (zelená) a pro $c = -0,80 + 0,13i$ (červená)

Mandelbrotova množina

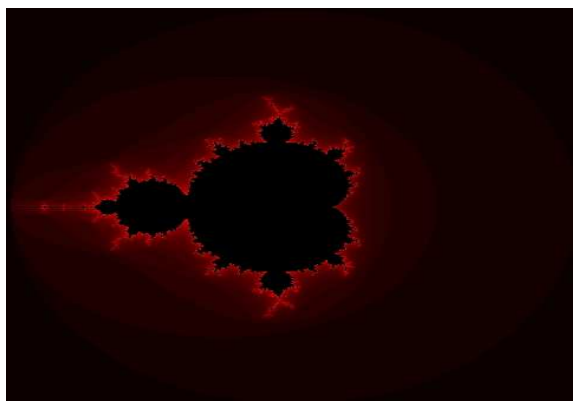
První, kdo ji popsal, byl Pierre Fatou roku 1905. Díky složitosti výpočtů ji však nikdy neviděl [18]. Až Mandelbrot, po němž je pojmenována, ji poprvé vykreslil. Mandelbrotova množina je katalogem množin Juliových. Každému bodu Mandelbrotovy množiny odpovídá Juliova množina s parametrem daným souřadnicemi tohoto bodu.

Je definována jako množina komplexních čísel c , pro které [7]:

$$\lim_{n \rightarrow \infty} |z_n| \neq \infty. \quad (48)$$

Pro z_n platí, $z_0 = 0$ a:

$$z_{n+1} = z_n^2 + c. \quad (49)$$



Obrázek 15: Mandelbrotova množina

2.1.4. Náhodné fraktály

Většina fraktálů je určitým způsobem symetrická a obrazce, jež tímto způsobem vzniknou, neodpovídají příliš reálným objektům, i když se jim často podobají. To je dáno tím, že se v živé přírodě nevyskytuje příliš souměrnost. Abychom dostali obrazy lépe odpovídající skutečnosti, je nutné obohatit generování fraktálů o určitý prvek náhody. Tato nahodilost bude mít vliv na tvar výsledného fraktálu i jeho Hausdorffovu dimenzi. Existuje několik způsobů, jak lze generovat náhodné fraktály:

- simulace Brownova pohybu
- metoda přesouvání středního bodu
- spektrální syntéza

Pro generování náhody lze využít guassovský generátor, či generátor bílého šumu.

Simulace Brownova pohybu vytváří fraktální obrazce, jejichž Hausdorffova dimenze je úměrná změně velikosti při jedné iteraci. Tato metoda se nehodí pro generování 3D objektů a používá se především pro generování toků řek.

Metoda přesouvání středního bodu je typickou metodou generování fraktálů. Používá se jak pro generování plošných, tak i prostorových objektů. Často je tato metoda využívána pro modelování krajiny, protože nastavením maximální odchylky středního bodu lze vyvářet různorodé povrchy.

Spektrální syntéza je metoda založená na Fourierových řadách. Její výhodou je, že s její pomocí můžeme vytvářet fraktály s požadovanou Hausdorffovou dimenzí. A to tak, že náhodně vygenerujeme Fourierovy obrazy se spektrální hustotou, která je úměrná zvolené Hausdorffově dimenzi. Na tomto obrazu provedeme inverzní Fourierovu transformaci, jejímž výsledkem je cílový fraktální objekt.

2.3. Praktické aplikace teorie chaosu

2.3.1. Samoorganizace

Samoorganizaci lze definovat jako jev, kdy se samostatné jednotky začnou chovat jako celek. Samoorganizační schopnost souvisí s tou částí teorie chaosu, která zahrnuje kvantovou mechaniku. Tento celek pak může vykazovat chaotické chování, nebo vytvářet fraktální struktury.

Hlavní projevy samoorganizace v biologii:

- balení proteinů a jiných biomakromolekul (folding)
- tvorba micel a membrán s dvojvrstvou lipidů
- homeostáze – schopnost udržení stabilního vnitřního prostředí
- vytváření struktur u sociálních organismů – včely, mravenci, termity, někteří savci...
- kolektivní chování – hejna ptáků a ryb
- původ života na Zemi
- vytvoření biosféry vhodné pro život

Hlavní projevy samoorganizace v chemii:

- samovolné vytváření molekul a monovrstev
- reakčně – difúzní systémy a oscilující chemické reakce
- autokatalytické reakce
- tekuté a koloidní krystaly
- vytváření micel

Hlavní projevy samoorganizace ve fyzice:

- spontánní magnetizace a růst krystalů, tekuté krystaly v klasické fyzice
- lasery, supravodivost v kvantové fyzice
- formování struktur v astrofyzice a kosmologii – formování hvězd a galaxií [4]
- vytváření struktur v termodynamických systémech daleko od rovnovážného bodu

Samoorganizace se projevuje také například v ekonomii, v lidském chování, v matematice, kybernetice a mnoha dalších oblastech.

2.3.2. Ekonomie

Ekonomii lze považovat za velký otevřený systém, který je ovlivňován fluktuacemi jak vnitřními, tak vnějšími. Je-li systém v blízkosti rovnovážného stavu, lépe odolává výkyvům a jeho chování je cyklické. Pokud je systém vychýlen, může dojít k zesílení – místo potlačení nežádoucích efektů – a systém může nabývat nových stavů. Protože ekonomika jako systém obsahuje nelineární vazby, může dojít k posunu chování až do chaosu.

Prvně si toho všiml Benoit Mandelbrot, když se zabýval cenou bavlny. Cena bavlny měla být ovlivněna v dlouhém období skutečným stavem trhu a tím, v jakém stavu je textilní průmysl a mezinárodní obchod. V krátkém období měla vykazovat více či méně náhodné výkyvy. Skutečná data však vykazovala odlišné chování, vyskytovalo se zde mnoho velkých skoků a jejich poměr k těm malým byl menší, než se původně předpokládalo. Zdroj dat byl pro výzkum vynikající, protože trh s bavlnou byl centralizovaný, a tudíž měl také centralizované záznamy, které sahaly až sto let do historie [19]. Nehledal, zda mají malé či velké výkyvy, jistou pravidelnost nebo strukturu, ale snažil se najít jistý řád v rámci všech dat. Po zadání dat do počítače a jejich studiu nečekaně objevil jistou symetrii. Jednotlivé ceny měly náhodný charakter, ale pokud pozoroval křivky, zjistil, že průběhy denních a měsíčních změn si jistým způsobem odpovídaly.

S chaosem v ekonomii souvisí také jev, který se nezývá ekonomická bublina. Ekonomická bublina se také často označuje jako spekulativní, tržní, cenová, nebo finanční bublina. Je to jev, který nastane, pokud se začne ve velkém obchodovat se zbožím, jehož cena neodpovídá reálné hodnotě, což vede ke zvýšení jeho poptávky a zvyšování ceny. Poté, co cena dosáhne nesmyslných hodnot, bublina praskne a následuje prudký propad ceny. Bublina má za následek nevhodné umístění prostředků a prudký pokles cen může znehodnotit velké množství bohatství a poškodit celou ekonomiku. Toto nastalo například ve 30. letech 20. století.

2.3.3. Astronomie

Snaha o přesnou předpověď oběžných drah těles vedla od prostých Newtonových zákonů přes poruchový počet až k aplikaci teorie chaosu. Zpočátku se zdálo, že pokud budeme počítat pouze s Newtonovými rovnicemi, bude výsledek dostatečně přesný. Po několika výpočtech a pozorování bylo zjištěno, že některá tělesa se nechovají tak, jak by podle výpočtů měla, protože se nejedná o prostý problém pohybu dvou těles kolem společného těžiště. Henri Poincaré zjistil, že problém tří těles již není možný přesně analyticky vyřešit, natož potom spočítat pohyby celé Sluneční soustavy. Díky rozložení hmotnosti bylo možné

zavést působení ostatních těles jako poruchy a celý problém redukovat na systém dvou těles a poruchy v jejich oběžných drahách. S postupujícím vývojem počítačů došlo ke zpřesňování výpočtů oběžných drah těles a byly zjištěny velké rozdíly mezi drahami různých planet. A to jak v jejich sklonu k ekliptice, tak i v jejich excentricitě. Některé planety obíhají téměř po kružnicích, jiné mají velkou výstřednost. Další rozdílnou vlastností byl sklon rotační osy. U většiny planet je 23° až 29° , ale najdou se i výjimky, kde je sklon osy 82° a dokonce i 98° . Existují velké rozdíly i u dob rotace a mezi počtem přirozených družic jednotlivých planet. Znaky chaosu nejsou jen u jednotlivých planet, ale i v pásu planetek mezi Marsem a Jupiterem. Vyskytují se zde pásy, kde se žádné planetky nevyskytují, jsou to takzvané Kirkwoodovy mezery. Pozorováním těchto mezer bylo zjištěno, že je za ně zodpovědný Jupiter, který tyto mezery čistí. Pokud planetka má oběžnou dobu, která je v poměru s oběžnou dobou Jupiteru, dochází k rezonanci a planetka je vystřelena ze své dráhy.

První aplikace chaosu v astronomii pochází od amerického astronoma J. Wisdoma, který se v roce 1983 ve své studii zabýval přenosem hmoty z Kirkwoodových mezer do blízkosti Země. O rok později objevil další důkaz chaosu ve Sluneční soustavě, a to v podobě Hyperionu, měsíce Saturnu, který je chaoticky ovlivňován gravitačním působením Titanu, největšího Saturnova měsíce.

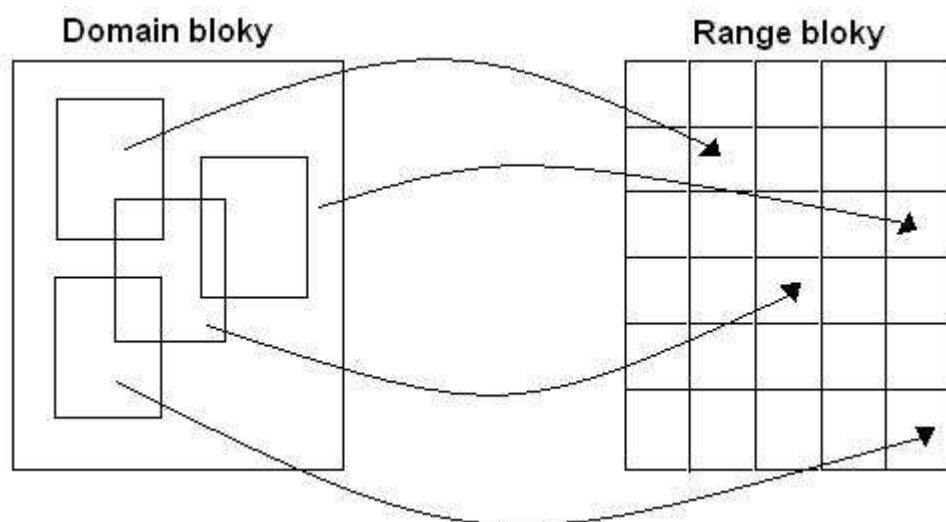
V roce 1985 byl spuštěn digitální planetostroj, počítač vytvořený pracovníky MIT (Massachusetts Institute of Technology). S jeho pomocí byly provedeny výpočty oběžných drah na 200 miliónů let zpět a až na 840 miliónů let dopředu. Bylo zjištěno kolísání sklonu oběžných drah k ekliptice. Dále došlo k chaotickým změnám dráhových parametrů, u Pluta v řádu 20 miliónů let a u vnitřních planet už v časovém horizontu 5 miliónů let. V horizontu 10 miliónů let je řád vystřídán chaosem. Pokud změny oběžných parametrů vyneseme do grafu, objevíme jisté periodické oscilace, které nápadně připomínají například průběh populační křivky.

Astronomií se také zabývali matematici Kolgomorov, Arnold a Moser, kteří zkoumali stabilitu nelineárních systémů. Výsledkem jejich práce je matematický teorém KAM, který udává dvě podmínky na stabilitu oběžných drah planet. V první podmínce se říká, že pokud má být oběžná dráha dvou těles stabilní, musí být vliv třetího tělesa velmi malý. Druhá podmínka udává stabilitu i za působení třetího tělesa, a to tak, že oběžné doby nesmí být v jednoduchém poměru, aby nedocházelo k rezonanci [20]. Potvrzením toho jsou již výše zmíněné Kirkwoodovy mezery.

2.3.4. Informatika

Komprese obrazu

Komprese obrazu je založena na znalostech IFS fraktálů. Princip komprese spočívá v rozdělení obrazu na dva typy bloků. Prvním typem jsou range bloky, které se nepřekrývají, a druhým typem jsou domain bloky, které se mohou překrývat a jsou podobné range blokům.



Obrázek 16: Fraktální komprese obrazu

Domain bloky se vyhledávají také transformované. Povolené transformace jsou rotace o 0° , 90° , 180° a 270° a překlopení podél horizontální a vertikální osy a podél obou diagonál. Přípustná je také změna kontrastu nebo jasu.

Fraktálová komprese probíhá v několika krocích [21]:

1. Segmentace obrazu – rozdělení obrazu na range bloky
2. Vytvoření domain bloků
3. Vyhledávání – pro každý range blok nalezneme nejvíce podobný domain blok

Pro vyhledávání existuje několik různých postupů – od prostého postupného procházení až po vyhledávání ve spirále od umístění range bloku. Výstupem jsou vždy transformační pravidla pro jednotlivé bloky.

Dekomprese je oproti kompresi velice jednoduchá. Jako základ se použije libovolný obrázek nebo černá plocha a poté se aplikují uložená transformační pravidla. Dekomprese je omezena pouze rozlišovací schopností zobrazovacího zařízení a lze ji provádět do nekonečna. Při vysokém počtu iterací při dekompresi může dojít k jevu, kterému se říká

fraktální interpolace. To znamená, že se v dekomprimovaném obraze objeví detaily, které se v původním vůbec nevyskytovaly.

Tato komprese dosahuje nejlepších výsledků na přírodních obrazech, naopak nejhorších při použití pro obrázky obsahující kresby a text. Takovéto části mohou být zdeformované nebo úplně zmizí.

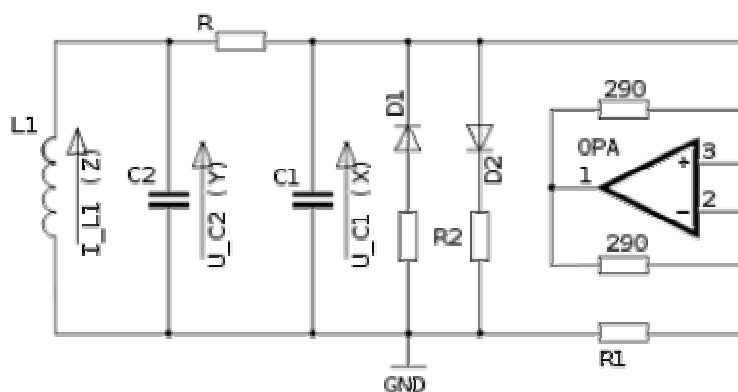
Modelování krajiny a přírodních objektů

Pro počítačové modelování krajiny, či jiných útvarů, se využívá náhodných fraktálů a metody přesouvání středního. Při modelování rostlin je vhodnější využívat IFS fraktály.

2.3.5. Elektronika

Chua obvod

Jednoduchý elektronický obvod, který vykazuje chaotické chování. Byl představen v roce 1983 Leonem O. Chua.



Obrázek 17: Chua obvod [1]

Aby autonomní obvod sestavený ze základních prvků – odporu, kondenzátoru a cívky, vykazoval chaotické chování, musí splňovat tři podmínky.

1. Alespoň jeden z prvků musí být nelineární
2. Alespoň jeden z rezistorů musí měnit svůj odpor v závislosti na napětí tak, aby udržel konstantní proud
3. Musí obsahovat nejméně tři prvky skladující energii

Chování tohoto obvodu lze popsat třemi nelineárními diferenciálními rovnicemi:

$$\frac{dx}{dt} = \alpha[y - x - f(x)], \quad (50)$$

$$\frac{dy}{dt} = x - y + z, \quad (51)$$

$$\frac{dz}{dt} = -\beta \cdot y, \quad (52)$$

kde $x(t)$, $y(t)$ jsou hodnoty napětí na kondenzátorech C1 a C2, $z(t)$ je hodnota proudu procházejícího cívkou L1, $f(x)$ popisuje nelinearitu rezistoru a α, β jsou parametry závislé na prvcích obvodu.

Komunikační systémy

Existují dva hlavní důvody pro používání chaotického signálu v komunikačních systémech. Jedním je, že chaotickým signálem lze nahradit normální šifrování, protože na něj lze namodulovat vysílaný signál. Kdokoli tento signál zachytí, uvidí jen chaotické šumové signály, kdežto autorizovaný přijímač dokáže odfiltrvat chaotický signál a obnovit původní informaci. Druhým důvodem je použití náhodných či pseudonáhodných signálů pro zvětšení šířky přenosového pásma. Takovéto signály jsou méně náchylné k interferenci a na vliv zpoždění signálu v bezdrátové komunikaci. Používá se několik přístupů:

Modulování chaotického signálu

Pro modulování chaotického signálu lze použít dva přístupy. Buď můžeme realizovat modulátor jako vysokofrekvenční chaotický systém a ten poté ovlivňovat signálem, nebo použijeme nízkofrekvenční chaotický systém, který smícháme se signálem a za pomoci modulátoru výsledný signál převedeme na vyšší frekvenci vhodnou pro přenos.

Chaotické maskování

Princip chaotického maskování spočívá v tom, že smícháme chaotický signál a signál nesoucí informaci a poté ho obnovíme za pomoci synchronizace. Pokud umístíme dva totožné chaotické systémy do přijímače a vysílače, lze informaci obnovit pouhým rozdílem signálů. Toto lze použít v případě, že signál nebude nijak významně ovlivněn šumem.

Chaotické klíčování

Při použití chaotického klíčování bude jeden z parametrů chaotického systému řízen informačním signálem. V nejjednodušším případě bude řídicí signál binární a bude přepínat mezi dvěma vektory parametrů. V přijímači budeme synchronizovat signál s oběma vektory a zjištěná synchronizace nám řekne, který bit jsme přijali.

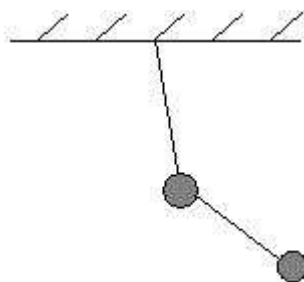
Diferenční chaotické klíčování

Aby byla odstraněna nutnost synchronizace, byla vyvinuta metoda diferenčního chaotického klíčování. Tato metoda vysílá modulovaný signál společně s nedomulovaným. Při odeslání jednoho bitu se do přenosového pásma vyšle ještě jeden signál, a to tak, že měl-li bit hodnotu 1, vyšle se jeho kopie. Pokud byla hodnota bitu 0, vyšle se kopie sečtená s hodnotou -1 . Pro získání původního signálu jsou v přijímači obě části jednoduše porovnány.

2.3.6. Fyzika

Dvojité kyvadlo

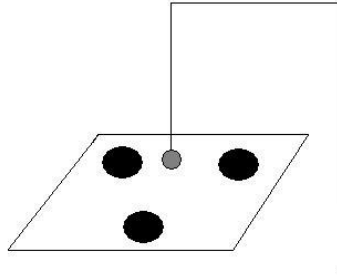
Dvojité kyvadlo je typickým příkladem chaotického systému. Složíme ho ze dvou jednoduchých kyvadel tak, že jejich konce volně spojíme, aby se mohla pohybovat. Pohyb dvojitého kyvadla je nutné popisovat diferenciálními rovnicemi a je velice citlivý na počáteční podmínky.



Obrázek 18: Dvojité kyvadlo

Kyvadlo s magnety

Dalším poměrně jednoduchým příkladem chaotického chování je kyvadlo, které bude ovlivněno třemi magnety v podložce. Z určitých míst kyvadlo vždy skončí u jednoho z magnetů, ale existují místa, kde bude kyvadlo vykazovat chaotické chování a my nebudeme schopni předpovědět, u kterého z magnetů skončí.



Obrázek 19: Kyvadlo s magnety

Turbulence

Turbulence, svatý grál fyziky, se také někdy označuje jako problém s rodokmenem, protože tento jev zkoumali všichni slavní fyzikové. Slavný fyzik Werner Heisenberg, který studoval kvantovou teorii, prý na smrtelném loži řekl, že bude mít na Boha dvě otázky. Proč relativita a proč turbulence. A na to prohlásil: „Opravdu věřím, že na první otázku by mohl mít odpověď.“ [19].

Do 70. let 20. století existovala ve fyzice teorie turbulence, kterou vytvořil Lev D. Landau. Tehdy označil turbulenci jako mnoho navzájem nesouhlasných rytmů. Nejprve se má objevit jeden a se zvyšující se rychlostí další a další. V roce 1973 se fyzikové Swiney a Gollub pokusili tuto teorii dokázat. Sestrojili válec, který se otáčel uvnitř válce, mezeru naplnili tekutinou a pozorovali, co se stane, když se bude zvyšovat rychlost rotace. Nejprve se objevila jedna frekvence. Podle teorie turbulence se měla postupně objevit další, ale tu nebyli schopni rozlišit a místo toho nastal chaos.

Jiný experiment s turbulencí provedl Libchaber. Vzal si uzavřenou nádobu naplněnou heliem, kterou zespodu zahříval. Helium sledoval dvěma teplotními sondami. Jak se nádoba ohřívala, tekutina začala proudit a za pomoci sond se začal vykreslovat graf. Při bližším pohledu zjistil, že graf odpovídá Lorenzovým rovnicím. Libchaber odhalil mnohem složitější strukturu než předpokládal.

Kapající kohoutek

Kapající kohoutek je dalším systémem, kde lze pozorovat teorii chaosu v praxi. Na počátku si vezmeme obyčejný vodovodní kohoutek, ze kterého postupně necháme odkapávat vodu. Pokud budeme kapky pozorovat, zjistíme, že kapou v pravidelných intervalech, máme diskrétní deterministický systém. Budeme – li postupně zvyšovat průtok, dojde k tomu, že kapky budou kapat nejprve po dvojicích a s dalším zvyšováním průtoku budou intervaly nepravidelné, projeví se zde chaotické chování a diskrétní systém postupně přejde na

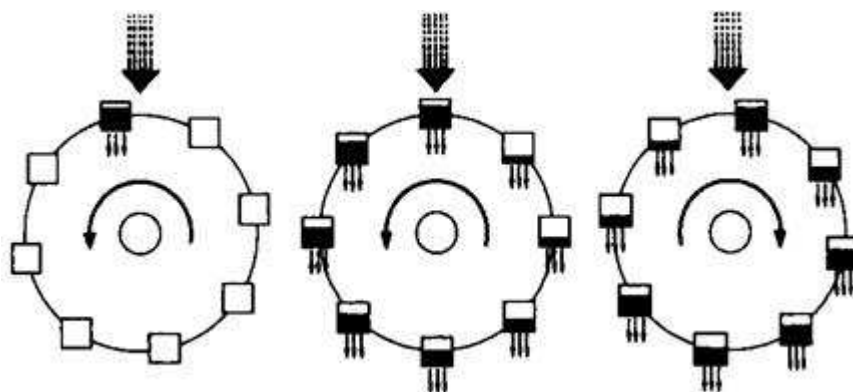
dynamický. Z původních kapek se stane laminární proudění a chaos zmizí. Pokud budeme i nadále zvyšovat průtok laminárního proudění, může přejít k přechodu na turbulentní proudění a znovu se projeví chaotické chování.

Shaw a Packard z University of Kalifornie udělali pokus, kdy vodu z kohoutku nechali kapat na mikrofon připojený k počítači. Každý dopad kapky byl zaznamenán jako cvaknutí. Zaznamenaná cvaknutí museli nějak zpracovat. Změřili intervaly mezi nimi a dostali sérii intervalů. Tato řada měla asi 5000 členů. Provedli rekonstrukci atraktoru pomocí principu posunutých časových řad a nechali jej počítačem vykreslit. Ve vykresleném obraze pozorovali strukturu, jež měla podkovovitý tvar – typický pro jednoduchý systém natahování a skládání (stretch a fold), a nápadně se podobala Hénonově atraktoru.

Lorenzovo vodní kolo

Vodní kolo je dalším chaotickým systémem. Jeho přesným popisem jsou Lorenzovy rovnice a jeho chování vystihuje Lorenzův atraktor (viz. kapitola Atraktory).

Lorenzovo vodní kolo vypadá stejně jako vodní kolo, které pohánělo například mlýny, jen místo lopatek jsou zde umístěny nádoby na vodu, které mají děravé dno. Principem tohoto kola je objasnit jeho chování při různých rychlostech přítoku vody.



Obrázek 20: Lorenzovo vodní kolo [22]

Chování systému závisí na hnací síle, v tomto případě na množství přitékající vody. Existuje několik typů chování vodního kola. Pokud je přítok malý, kolo nepřekoná tření, zůstane stát a všechna voda odteče. Zvýšíme-li množství přitékající vody, kolo se bude otáčet jedním směrem. Pokud ještě o něco zvýšíme přítok, může se systém kola dostat v důsledku nelineárních vazeb do chaosu, kdy se bude kolo otáčet střídavě jedním i druhým

směrem, přičemž toto chování nebude vykazovat žádnou pravidelnost a nebudeme schopni předpovědět, kdy se jakým směrem bude otáčet.

Byla také zjištěna podobnost chování Lorenzova kola a konvekčního proudění.

2.3.7. Biologie

Modelování populace

V biologii se teorie chaosu uplatňuje především v samoorganizaci (viz. výše) a v modelování vývoje populace pomocí logistické (populační) funkce. Logistická funkce se používá buď v základním tvaru:

$$x_{n+1} = rx_n(1 - x_n), \quad (53)$$

nebo v různých obměnách, kde vystupuje limitační faktor prostředí K (nosná kapacita prostředí):

$$x_{n+1} = rx_n \left(1 - \frac{x_n}{K}\right). \quad (54)$$

Lze případně modelovat i složitější systémy, kde se vyskytuje vazba predátor – kořist rovnice Lotka – Volterra:

$$\frac{dH}{dt} = rH - aPH, \quad (55)$$

$$\frac{dP}{dt} = bPH - mP, \quad (56)$$

kde H je množství kořisti, P počet predátorů, r je růst kořisti, a je útočnost predátora, b přeměna kořisti na biomasu predátora, m je mortalita predátora.

Sudé grafy vyjadřují velikosti populace v průběhu času, liché grafy popisují závislost predátora na kořisti.

Grafy 12 a 13 ukazují stabilní vývoj populací. Došlo k rovnováze a ani jeden z druhů nevyhyne.

Grafy 14 a 15 ukazují vývoj populace, když se kořist rychle množí. Dochází zde k rychlým a vysokým výkyvům v obou populacích.

Grafy 16 a 17 ukazují vývoj populací, když se kořist množí pomalu a lovec bez kořisti rychle vymírá. Dochází zde k téměř úplnému vyhynutí predátora.

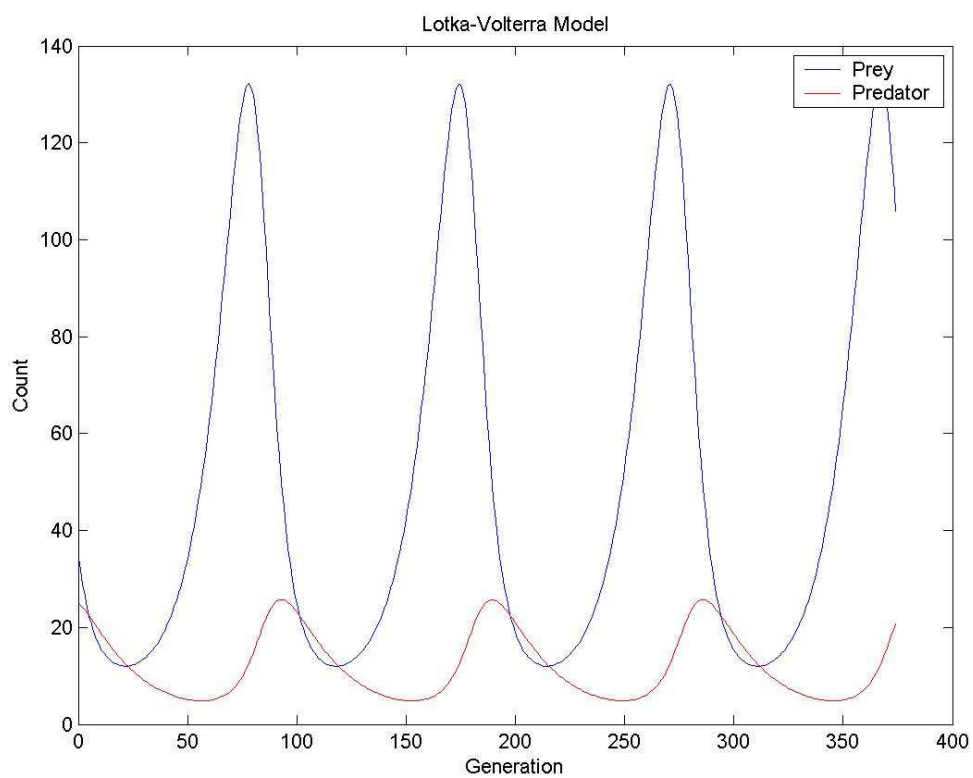
Grafy 18 a 19 ukazují vývoj populace, když je predátora víc než kořisti a ta se zároveň pomalu množí. Dojde k úplnému vyhynutí obou populací, protože lovec vybijí kořist a pak vyhladoví.

Grafy 20 a 21 ukazují vývoj stabilních populací, když lovec potřebuje větší množství kořisti k tomu, aby přežil. Obě populaci přežijí.

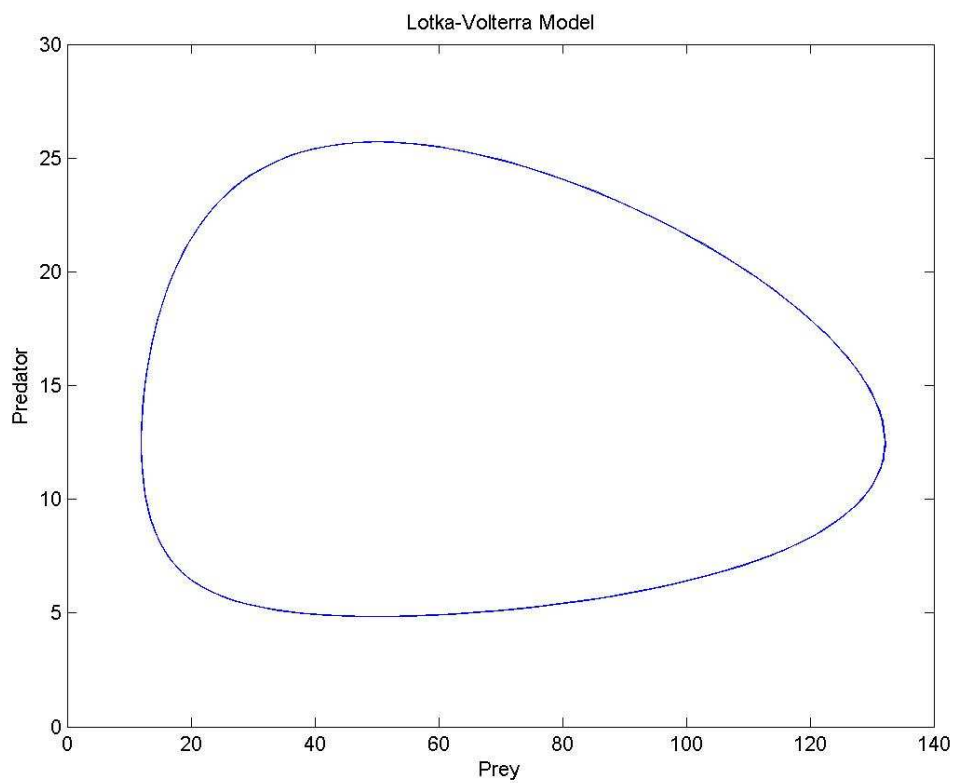
Tabulka 5: Populační model Lotka – Volterra

Model populace Lotka – Volterra						
číslo grafu	r	a	b	m	h_0	p_0
12, 13	0,01	0,08	0,01	0,05	35	25
14, 15	0,75	0,03	0,01	0,2	50	15
16, 17	0,05	0,01	0,05	0,25	15	15
18, 19	0,01	0,05	0,05	0,01	15	25
20, 21	0,05	0,01	0,01	0,3	50	2

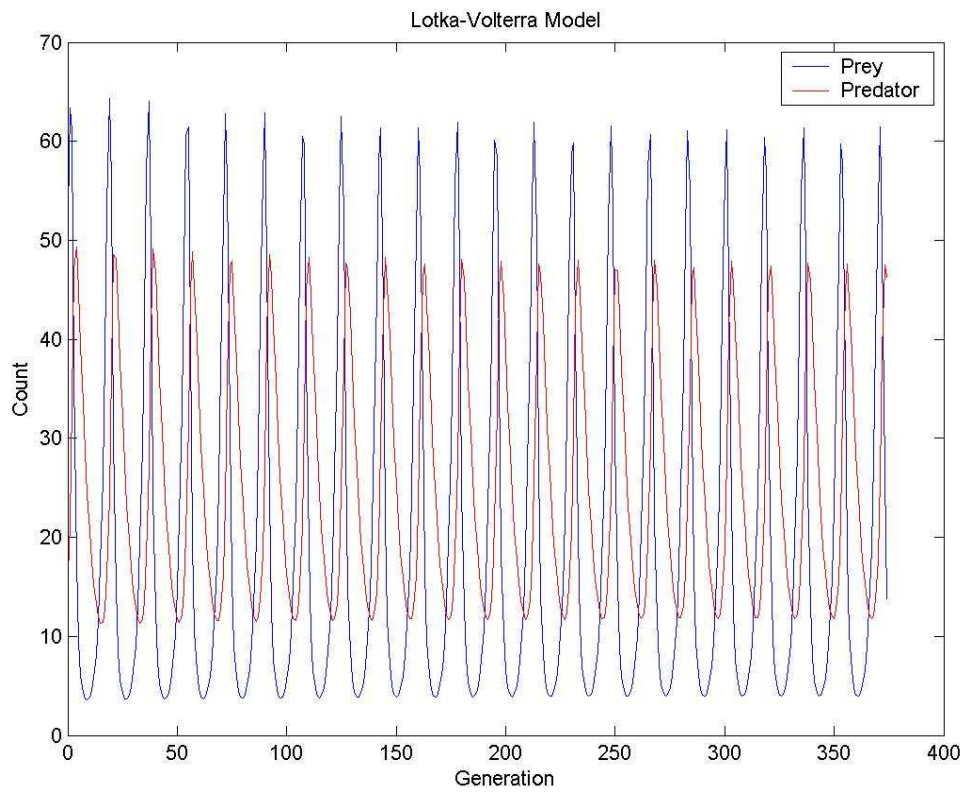
Graf 12: Lotka – Volterra - Simulace lovec – kořist 1a



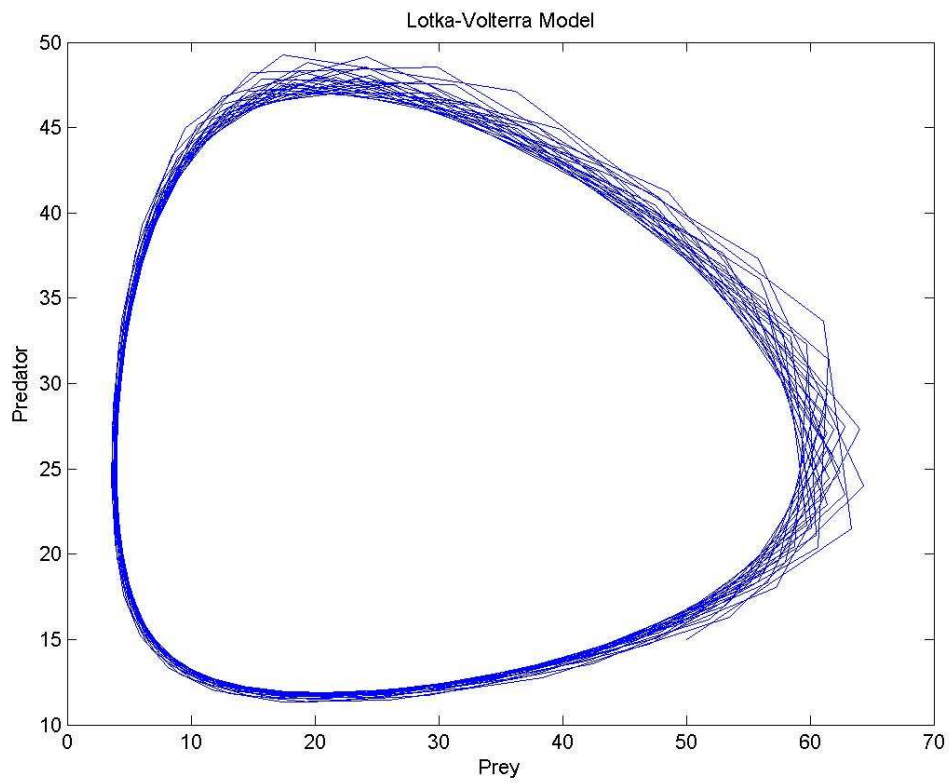
Graf 13: Lotka – Volterra - Simulace lovec – kořist 1b



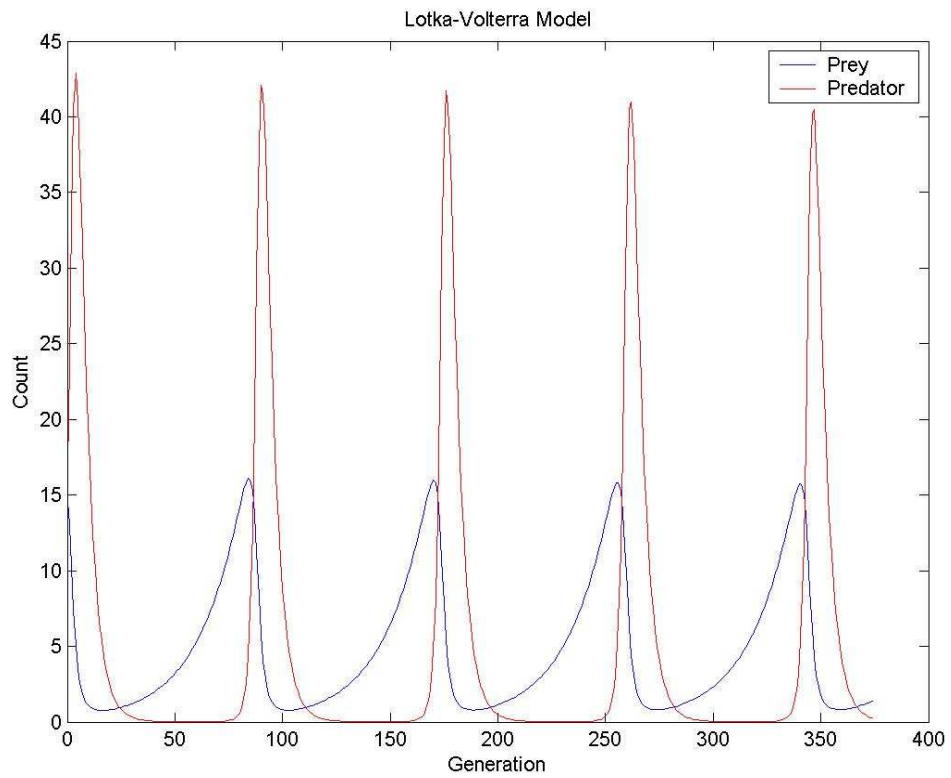
Graf 14: Lotka – Volterra - Simulace lovec – kořist 2a



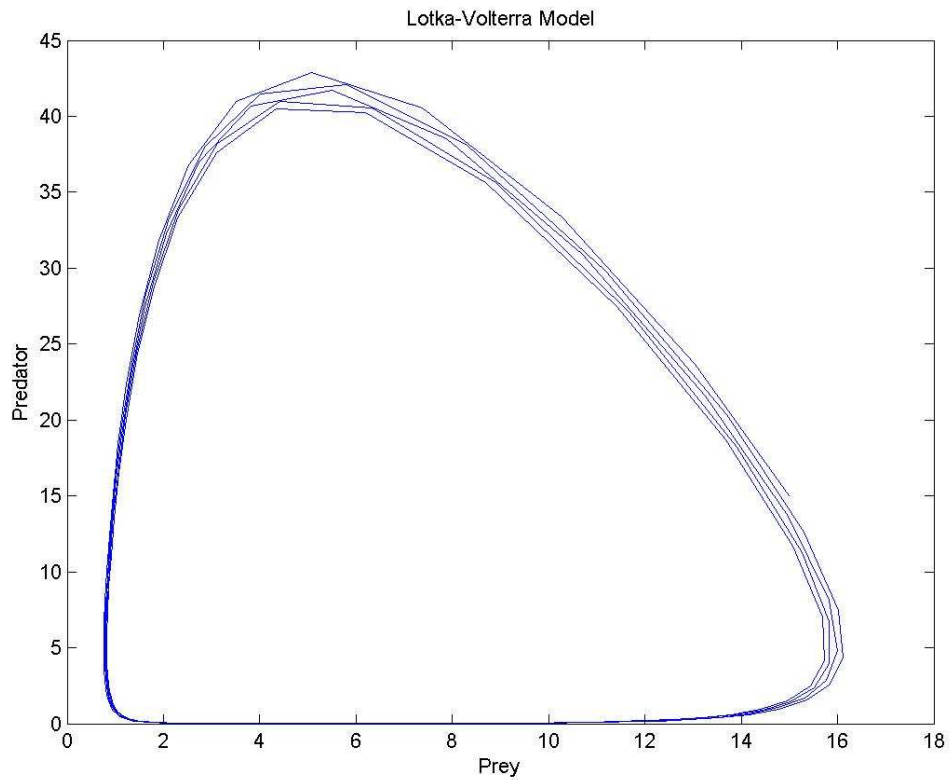
Graf 15: Lotka – Volterra - Simulace lovec – kořist 2b



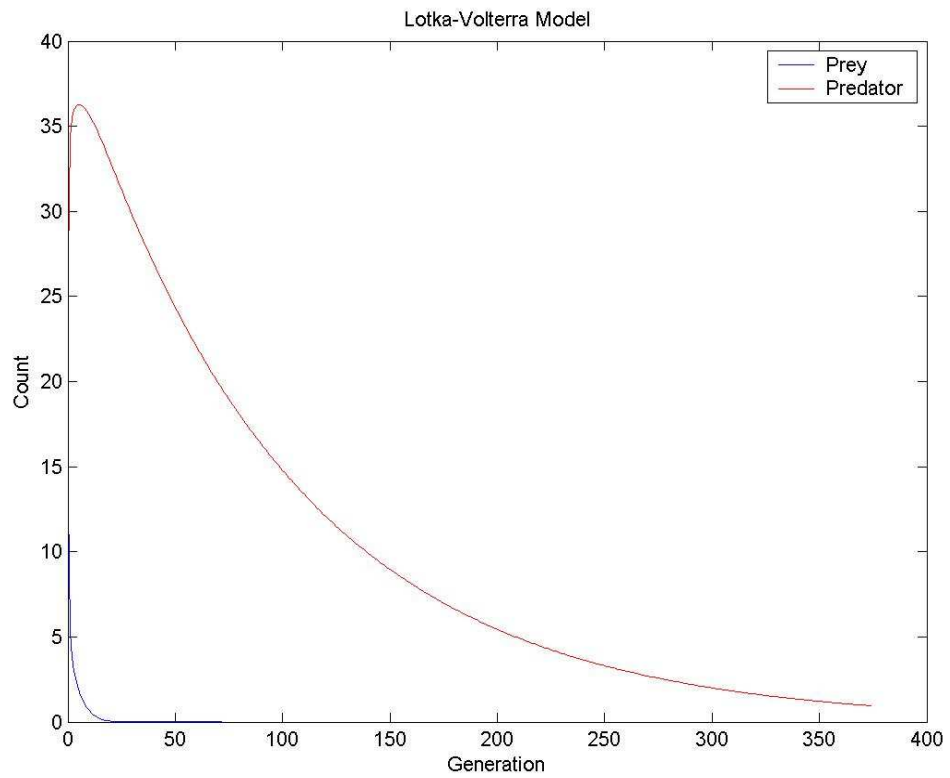
Graf 16: Lotka – Volterra - Simulace lovec – kořist 3a



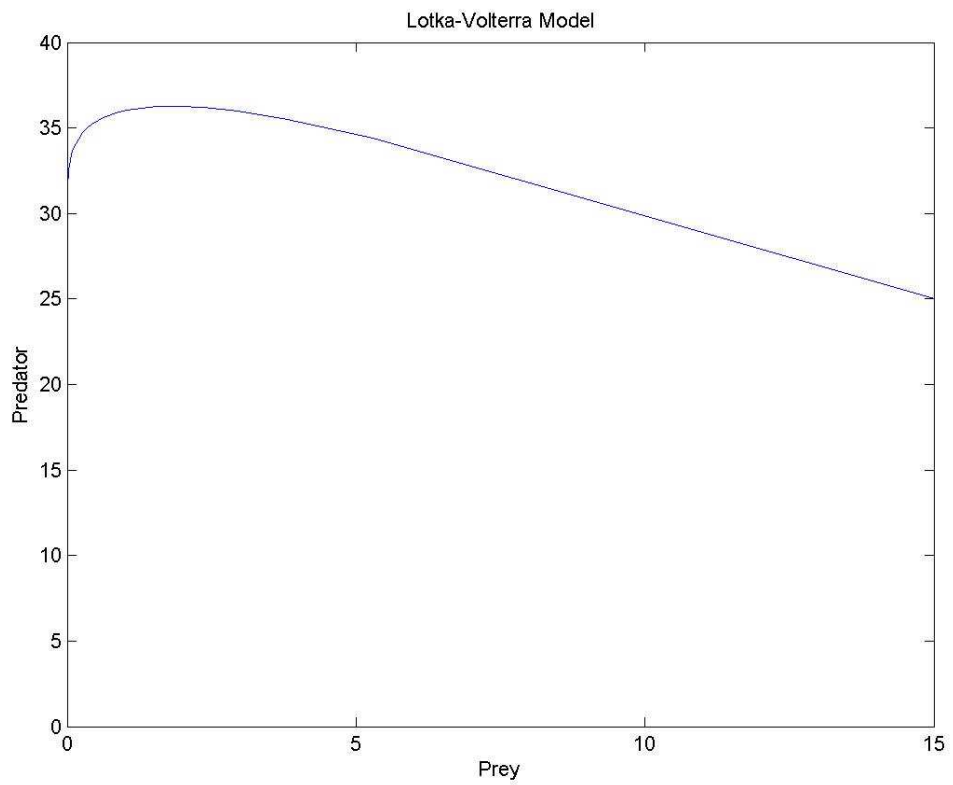
Graf 17: Lotka – Volterra - Simulace lovec – kořist 3b



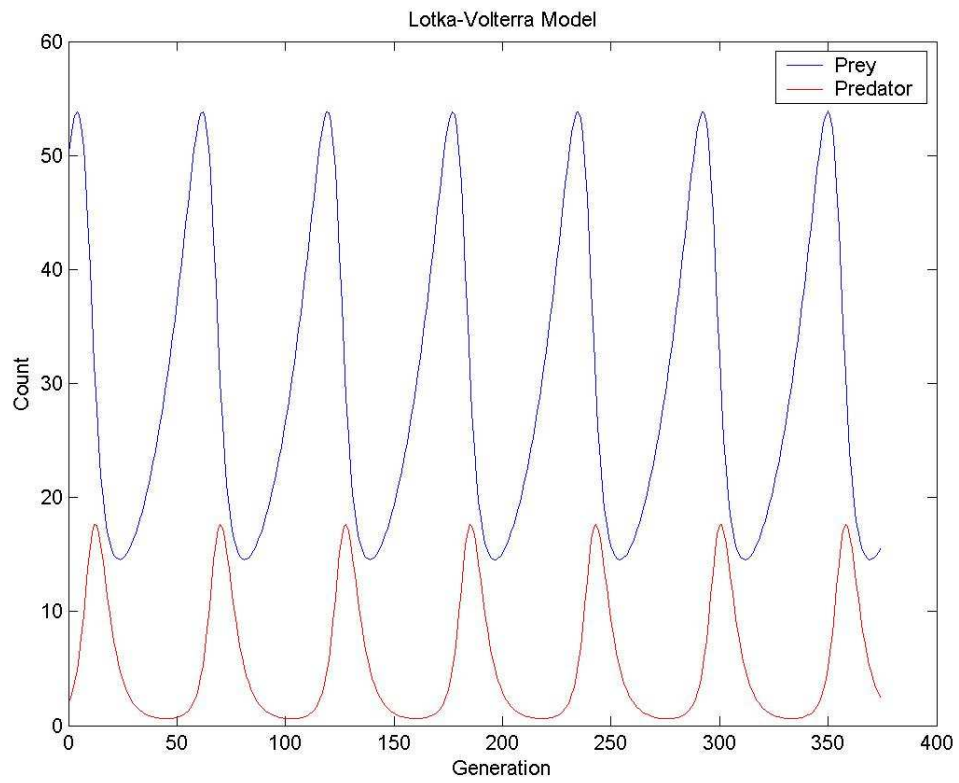
Graf 18: Lotka – Volterra - Simulace lovec – kořist 4a



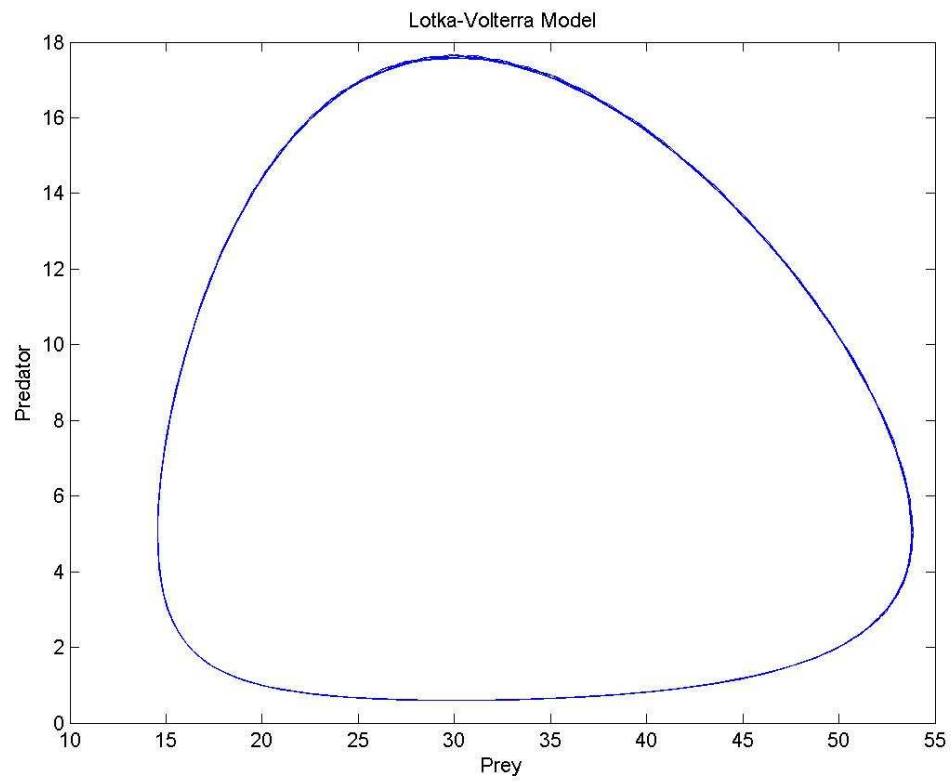
Graf 19: Lotka – Volterra - Simulace lovec – kořist 4b



Graf 20: Lotka – Volterra - Simulace lovec – kořist 5a

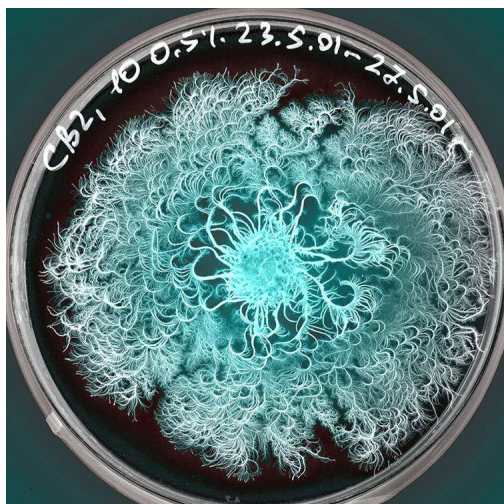


Graf 21: Lotka – Volterra- Simulace lovec – kořist 5b



Kolonie bakterií

Kolonie bakterií jsou při bližším pozorování ukázkou přírodních fraktálů a samoorganizace.



Obrázek 21: Kolonie bakterií [23]

Hejna ryb

Pohyb ryb v hejnu není dán žádným vedoucím jedincem, ale řídí se třemi jednoduchými pravidly – soudržností hejna, zařazením a oddělením. Jedná se o projev kolektivního chování na základě samoorganizace [24].

Srdce

Srdeční arytmie je stav, kdy srdce začne bít chaoticky. Při zkoumání EKG bylo zjištěno, že před infarktem srdce přechází od normálního rytmu za principu zdvojování periody k arytmií. V moderní medicíně se již průběh EKG podrobuje fraktální analýze [24].

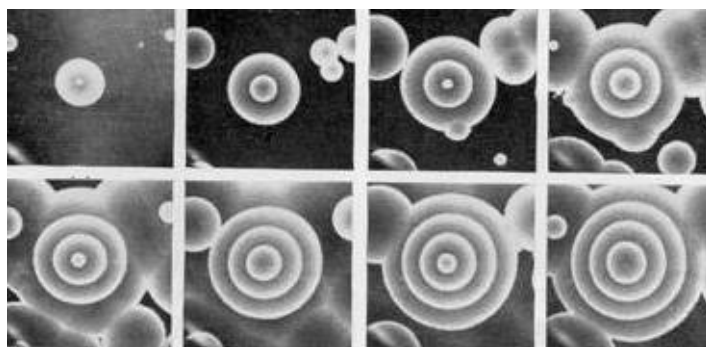
Výzkum v této oblasti je zajímavý, protože by se nám mohlo v budoucnu dařit předpovídat různé poruchy srdečního rytmu, včetně infarktů.

2.3.8. Chemie

Příkladem chaotického chování v chemii jsou oscilující reakce. Reakce Bělousov – Žabotinskij jsou jednou ze tříd chemických reakcí, které jsou typickým příkladem nerovnovážené termodynamiky, jejímž důsledkem je nelineární chemický oscilátor.

Jako první objevil chemické oscilace v roce 1926 William Bray, když rozkládal peroxid vodíku na kyslík a vodík pomocí jódového katalyzátoru. O 42 let později ruský chemik

Alexander Bělousov pozoroval periodické barevné oscilace ve směsi kyseliny citrónové, sírové, bromidu draselného a soli ceru. V roce 1963 Anatolij Žabotinskij pozměnil trochu recept a místo soli ceru dal do směsi soli železa. V tomto pozměněném pokusu se střídala červená barva s modrou. A pokud byla směs rozlita do tenké vrstvy, daly se na ní pozorovat kruhové a spirálové vlny.



Obrázek 22: reakce Bělousov – Žabotinskij

Při pozdějších pokusech se měřila koncentrace iontů bromu a za pomoci matematické analýzy se určilo frekvenční spektrum koncentrací a z něj se určily frekvence oscilací. Po vynesení do grafu byla patrná geometrie podivného atraktoru.

2.3.9. Teorie klimatu

V teorii klimatu se teorie chaosu uplatňuje především v oblasti předpovědí. Ať se jedná o předpovědi krátkodobé, dlouhodobé, nebo i předpovědi vývoje klimatu na desetiletí dopředu.

Jeden ze zakladatelů předpovědi počasí Wilhelm Bjerknes stanovil podmínky roku 1904, podle kterých je možná předpověď počasí. Tou první je znalost výchozích podmínek a tou druhou používání základních fyzikálních zákonů. Roku 1922 Lewis Fry Richardson sestavil první numerickou předpověď.

První novodobé pokusy o předpověď počasí s pomocí počítače vedly k vytvoření teorie chaosu. Jednalo se vcelku o náhodu. Edward Lorenz si chtěl zjednodušit práci a místo, aby zadal výpočet od začátku, zapsal do počítače data z předchozího výpočtu a spustil pokus. Po nějaké době zkontroloval výsledky a zjistil, že nové a původní výsledky se lišily. Při prozkoumání došel k závěru, že za to mohlo zaokrouhlování čísel.

Největší oříškem při předpovědi počasí je ona znalost počátečních podmínek, protože jak Edward Lorenz dokázal, je výpočet atmosféry velmi citlivý a i nepatrná nepřesnost

v měření může mít za následek velké odlišnosti ve vývoji počasí. Další problémy v přesné předpovědi způsobuje velké množství proměnných a efekt, kdy dochází v průběhu výpočtů k zesílení nepřesností, které mohou pak ovlivňovat celkový charakter počasí.

I přes tyto složité podmínky se snažíme o stanovení co nejpřesnější předpovědi. Předpověď probíhá v cyklu o několika krocích. První krokem je pozorování atmosféry. To se děje za pomoci pozemních stanic, meteorologických balónů i družic. Dalším krokem je určení fyzikálních zákonů, pomocí kterých bude výpočet probíhat. Následně je nutné vytvořit nějaký matematický model, který bude předlohou pro počítačový model. Počítačový model zahrnuje nelineární diferenciální rovnice a také rozlišení, s jakým se bude počítat. Čím přesnější předpověď je vyžadována, tím lepší musí být rozlišení a s tím také rostou nároky na výpočetní techniku. Po vlastním výpočtu se výsledek porovná s realitou a začne se znovu od začátku. Dnes se pro přesnější modely počasí využívá základní vlastnosti chaosu, kterou je citlivost k počátečním podmínkám. Vezme se nějaký model počasí, naměřené hodnoty a spustí se výpočet. Zároveň s tím se ale spustí několik výpočtů i s mírně pozměněnými počátečními podmínkami. Celý soubor výsledků se poté vezme a porovná, pokud se zde nevyskytuje žádná výrazná odlišnost, můžeme považovat danou předpověď za poměrně přesnou.

Praktická část

3. Dodatek A: Programování atraktorů

3.1. Lorenzův atraktor

Programovací jazyk: Pascal

```
program Lorenz;
uses CRT;
var i,dx,dy,dz,x,y,z:real;
    soubor:text;
const sigma=20;
    beta=8/3;
    ro=28;
    dt=0.005;
procedure Vypocet;
begin
    x:=(x-
sigma*x*dt+sigma*y*dt);
    y:=(y+ro*x*dt-y*dt-
z*x*dt);
    z:=(z-beta*z*dt+x*y*dt);
end;
procedure Zapis;
begin
    Writeln(soubor,x,' ',y,'
',z);
end;

begin
assign(soubor,'lorenz.txt');
rewrite(soubor);
x:=0.1;
y:=0;
z:=0;
i:=0;
repeat
    i:=i+1;
    Vypocet;
    Zapis;
until i>6000;
close(soubor);
end.
```

Programovací jazyk: Java

```
import java.io.*;
public class Lorenz {

    static File soubor= new
File("Lorenz.txt");
    static double x,y,z;
    static final int
SIGMA=20;
    static final double
BETA=8/3;
    static final int RO=28;
    static final double
dt=0.005;

    public static void
Vypocet() throws Exception{
        x=0.1;
        y=0;
        z=0;
        FileWriter fw = new
FileWriter(soubor);
        fw.write(x+" "+y+"
+z+"\n");
        for(int
i=0;i<6000;i++){
            x=(x-
SIGMA*x*dt+SIGMA*y*dt);
            y=(y+RO*x*dt-
y*dt-z*x*dt);
            z=(z-
BETA*z*dt+x*y*dt);
            fw.write(x+"
+y+" "+z+"\n");
        }
        fw.close();
    }

    public static void
main(String[] args) throws
Exception{
        Vypocet();
    }
}
```

3.2. Hénonův atraktor

Programovací jazyk: Pascal

```
program Henon;
uses Graph,CRT;

var GD,GM:Integer;
    x,y,xn,yn:Real;
    soubor:text;
const
GP='D:\programy\BP\Bp\bgi';
(* Cesta ke grafice *)

procedure Fraktal;
begin
    x:=0;
    y:=0;
    repeat
        writeln(soubor,x,'
',y);
        putpixel(round(x*120+320),round(y*200+240),yellow);
        xn:=x;
        yn:=y;
        x:=yn+1-(1.4*xn*xn);
        y:=0.3*xn;
        delay(1);
    until keypressed;
end;

Begin

assign(soubor,'henon.txt');
rewrite(soubor);
GD:=Detect;
InitGraph(GD,GM,GP);
Fraktal;
CloseGraph;
close(soubor);
end.
```

Programovací jazyk: Java

```
import java.io.*;

public class Henon {

    static File soubor= new
File("Henon.txt");

    static double x,y,xn,yn;

    public static void
Vypocet() throws Exception{
        x=0;
        y=0;
        FileWriter fw = new
FileWriter(soubor);
        fw.write(x+"
+y+"\"n");
        for(int
i=0;i<14000;i++){
            xn=x;
            yn=y;
            x=yn+1-
(1.4*xn*xn);
            y=0.3*xn;
            fw.write(x+"
+y+"\"n");
        }

        fw.close();
    }

    public static void
main(String[] args) throws
Exception{
        Vypocet();
    }
}
```

3.3. Kings Dream

Programovací jazyk: Pascal

```
program KingsDream;
uses Graph,CRT;

var GD,GM:Integer;
    x,y,xn,yn:Real;
    soubor:text;

const A=-0.966918;
      B=2.879879;
      C=0.765145;
      D=0.744728;

GP='d:\programy\BP\bp\bgi';
(* Cesta ke grafice *)
procedure Fraktal;
begin
  x:=1;
  y:=1;
  repeat
    writeln(soubor,x, '
',y);

    putpixel(round(x*90+320),rou
nd(y*90+ 240),yellow);
    xn:=x;
    yn:=y;

    x:=sin(yn*B)+C*sin(xn*B);

    y:=sin(xn*A)+D*sin(yn*A);
  until keypressed;
end;

Begin

assign(soubor, 'kingsdre.txt'
);
rewrite(soubor);
GD:=Detect;
InitGraph(GD,GM,GP);
Fraktal;
CloseGraph;
close(soubor);
end.
```

Programovací jazyk: Java

```
import java.io.*;

public class KingsDream {

    static File soubor= new
File("KingsDream.txt");

    static double x,y,xn,yn;
    static final double A =
-0.9669108;
    static final double B =
2.879879;
    static final double C =
0.765145;
    static final double D =
0.744728;

    public static void
Vypocet() throws Exception{
        x=1;
        y=1;
        FileWriter fw = new
FileWriter(soubor);
        fw.write(x+"
"+y+"
");
        for(int
i=0;i<600000;i++){
            xn=x;
            yn=y;

            x=Math.sin(yn*B)+C*Math.sin(
xn*B);

            y=Math.sin(xn*A)+D*Math.sin(
yn*A);

            fw.write(x+"
"+y+"
");
        }
        fw.close();
    }

    public static void
main(String[] args) throws
Exception{
        Vypocet();
    }
}
```

3.4. Rösslerův atraktor

Programovací jazyk: Java

```
import java.io.*;

public class Rossler {

    static File soubor= new File("Rossler.txt");

    static double x,y,z,x2,y2,z2;
    static final double A=0.2;
    static final double B=0.2;
    static final double C=5.7;
    static final double dt=0.04;

    public static void Vypocet() throws Exception{
        x=1;
        y=1;
        z=1;
        FileWriter fw = new FileWriter(soubor);
        fw.write(x+" "+y+" "+z+"\n");
        for(int i=0;i<6000;i++){
            x2 = x-y*dt-z*dt;
            y2 = y+x*dt+A*y*dt;
            z2 = z+B*dt+x*z*dt-C*z*dt;
            x=x2;
            y=y2;
            z=z2;
            fw.write(x+" "+y+" "+z+"\n");
        }

        fw.close();
    }

    public static void main(String[] args) throws Exception{
        Vypocet();
    }
}
```

4. Dodatek B: Programování fraktálů

4.1. Mandelbrotova množina

Programovací jazyk: Pascal

```
program Mandel;
uses graph,crt;

var x,y,i,j,GD,GM:Integer;
    cr,ci,zr,zi:Real;
    soubor:text;

const ITER=255;
      N=2;
      GP='d:\programy\BP\bp\bgi'; (* Cesta ke grafice *)

procedure Fraktal;
begin
  for x:=0 to 640 do
    begin
      for y:=0 to 480 do
        begin
          cr:=(x-320)/100;
          ci:=(y-240)/100;
          zr:=cr;
          zi:=ci;
          for i:=0 to ITER do
            begin
              cr:=cr*cr-ci*ci+zr;
              ci:=2*cr*ci+zi;
              if(sqrt(cr*cr+ci*ci)>2) then break;
            end;
            putpixel(x,y,i);
            Writeln(soubor,x,' ',y,' ',i);
          end;
        end;
      end;
    end;

begin
  assign(soubor,'mandel.txt');
  rewrite(soubor);
  GD:=Detect;
  InitGraph(GD,GM,GP);
  Fraktal;
  Readln;
  CloseGraph;
  close(soubor);
end.
```

5. Dodatek C: Programování diagramů

5.1. Logistická rovnice

Programovací jazyk: Pascal

```
program Logisticka_fce;
uses CRT;

var i:integer;
    xn1,xnpj1,xn2,xnpj2,xn3,xnpj3:real;
    soubor:text;

const r1=0.5;
      r2=1.5;
      r3=2.5;

procedure Vypocet;
begin
  for i:=1 to 64 do
  begin
    xnpj1:=r1*xn1*(1-xn1);
    xn1:=xnpj1;
    xnpj2:=r2*xn2*(1-xn2);
    xn2:=xnpj2;
    xnpj3:=r3*xn3*(1-xn3);
    xn3:=xnpj3;
    WriteLn(soubor,i,' ',xn1,' ',xn2,' ',xn3);
  end;
end;

begin
  assign(soubor,'logfcel.txt');
  rewrite(soubor);
  xn1:=0.25;
  xn2:=0.25;
  xn3:=0.25;
  WriteLn(soubor,0,' ',xn1,' ',xn2,' ',xn3);
  Vypocet;
  close(soubor);
  Write('Vypocet ukoncen');
  ReadLn;
end.
```

5.2. Bifurkační diagram

Programovací jazyk: Pascal

```
program Bifurkacni_Diagram;
uses CRT;

var i,r:integer;
    xn,xnpj:real;
    soubor:text;

procedure Vypocet;
begin
  for r:=2400 to 4000 do
  begin
    for i:=1 to 200 do
    begin
      xnpj:=(r/1000)*xn*(1-xn);
      xn:=xnpj;
      if i>=150 then WriteLn(soubor,r/1000,xn);
    end;
  end;
end;

begin
  assign(soubor,'bifdiag.txt');
  rewrite(soubor);
  xn:=0.25;
  Vypocet;
  close(soubor);
  Write('Vypocet ukoncen');
  ReadLn;
end.
```


5.3. Lotka-Volterra simulace

Programovací jazyk: Java

```
import java.io.*;

public class Lotka {
    static File soubor = new File("lotka.txt");
    static final double R = 0.1;
    static final double A = 0.008;
    static final double B = 0.001;
    static final double M = 0.05;
    static final double H0 = 35;
    static final double P0 = 25;
    static double ht, dht, ht05, dht05;
    static double pt, dpt, pt05, dpt05;
    static int i;

    public static void Vypocet() throws Exception{
        FileWriter fw = new FileWriter(soubor);
        ht=H0;
        pt=P0;
        fw.write("0"+" "+ht+" "+pt+"\n");
        for(i=1;i<375;i++){
            dht=R*ht-A*pt*ht;
            dpt=B*pt*ht-M*pt;

            ht05=ht+0.5*dht;
            pt05=pt+0.5*dpt;

            dht05=R*ht05-A*pt05*ht05;
            dpt05=B*pt05*ht05-M*pt05;

            ht=ht+dht05;
            pt=pt+dpt05;
            fw.write(i+" "+ht+" "+pt+"\n");
        }
        fw.close();
    }

    public static void main(String[] args) throws Exception{
        Vypocet();
    }
}
```

6. Závěr

Cílem této práce bylo provést rešerši literatury a jiných dostupných zdrojů o nelineární fyzice a teorii chaosu. Součástí této práce bylo i zaměření na praktické aplikace. Teorie chaosu je poměrně nové a rychle se rozvíjející odvětví teoretické fyziky, tudíž nachází uplatnění v mnoha různých vědních oborech. Lze se s jejími důsledky setkat i v praktickém životě.

V rámci praktické části bylo naprogramováno několik aplikací ve dvou různých jazycích. Aby bylo možné porovnat výsledky, byly jako výstup zvoleny textové soubory. Z těchto souborů byly vytvořeny grafy pomocí programu Matlab. V praktické části byla také podrobněji zkoumána biologická aplikace na populačním modelu Lotka – Volterra.

K vykreslení grafů byl použit program Matlab 6.5, pro programování Pascal 7.0 a NetBeans 6.1 (Java), pro vykreslení Mandelbrotovy množiny a Juliových množin byl použit program EasyFractal 1.0.

7. Literatura

- [1] Horák J., Krlín L., Raidl A.: Deterministický chaos a jeho fyzikální aplikace, Academia, Praha, 2003
- [2] Horák J., Krlín L., Raidl A.: Deterministický chaos a matematické modely turbulence, Academia, Praha, 1996
- [3] Smith, L.: Chaos – A very short introduction, Oxford University Press, Oxford, 2007
- [4] Gerhard O., E.: Stochastic Stellar orbits in perturbed integrable galaxy potential, Astron., Astrophys. 151, 279-296, (1985)
- [5] Nelineární dynamické systémy, <http://www.wood.mysteria.cz>
- [6] Navajo – otevřená encyklopedie, <http://teorie-chaosu.navajo.cz>
- [7] Wikipedia, free encyclopedia, <http://www.wikipedia.org>
- [8] Chaos Theory: A Brief Introduction, <http://www.imho.com/grae/chaos/chaos.html>
- [9] Chaos theory, <http://library.thinkquest.org/3493/noframes/chaos.html>
- [10] Aleš Raidl, <http://www.chaos.estranky.cz>
- [11] Deterministické systémy – Obsah,
<http://www.fce.vutbr.cz/studium/materialy/Dynsys/Default.htm>
- [12] Root.cz, <http://www.root.cz>
- [13] The Chaos hypertextbook, <http://hypertextbook.com/chaos/>
- [14] Wolfram MathWorld, <http://mathworld.wolfram.com>
- [15] Výpočet plochy Mandelbrotovy množiny,
<http://www.elektrorevue.cz/clanky/01022/01.htm>
- [16] Fraktály a choas, <http://chaos.fraktaly.sweb.cz>
- [17] <http://www.fit.vutbr.cz/~tisnovpa/publikace/diplomka/doc/>
- [18] Jemný úvod do fraktálů, <http://martin.hinner.info>
- [19] 21. století, <http://www.21stoleti.cz>
- [20] Zélos – Chaos ve sluneční soustavě, <http://www.astro.cz>
- [21] Fraktální komprese, <http://www.prikryl.cz>
- [22] Poznámky k chaosu, <http://herodes.feld.cvut.cz/nonlin/chaos/chaos.php>
- [23] <http://physicalsciences.ucsd.edu>
- [24] Deterministický chaos: Princip a aplikace, <http://www.hungry-lord.wz.cz>

8. Seznam grafů

Graf 1:	Motýlí efekt.....	12
Graf 2:	Lorenzův atraktor.....	15
Graf 3:	Rösslerův atraktor.....	15
Graf 4:	Hénonův atraktor.....	16
Graf 5:	Kings Dream.....	16
Graf 6:	Logistická rovnice pro $r = 0,5; 1,5; 2,5$	19
Graf 7:	Logistická rovnice pro $r = 3,25; 3,50$	19
Graf 8:	Logistická rovnice pro $r = 3,57; 3,75$	20
Graf 9:	Logistická rovnice pro $r = 4,5$	20
Graf 10:	Kochova křivka – mřížková dimenze.....	23
Graf 11:	Bifurkační diagram logistické funkce.....	27
Graf 12:	Lotka – Volterra Simulace lovec – kořist 1a.....	50
Graf 13:	Lotka – Volterra Simulace lovec – kořist 1b.....	50
Graf 14:	Lotka – Volterra Simulace lovec – kořist 2a.....	51
Graf 15:	Lotka – Volterra Simulace lovec – kořist 2b.....	51
Graf 16:	Lotka – Volterra Simulace lovec – kořist 3a.....	52
Graf 17:	Lotka – Volterra Simulace lovec – kořist 3b.....	52
Graf 18:	Lotka – Volterra Simulace lovec – kořist 4a.....	53
Graf 19:	Lotka – Volterra Simulace lovec – kořist 4b.....	53
Graf 20:	Lotka – Volterra Simulace lovec – kořist 5a.....	54
Graf 21:	Lotka – Volterra Simulace lovec – kořist 5b.....	54

9. Seznam obrázků

Obrázek 1:	J. H. Poincaré.....	7
Obrázek 2:	E. N. Lorenz.....	8
Obrázek 3:	M. J. Feigenbaum.....	8
Obrázek 4:	B. Mandelbrot.....	8
Obrázek 5:	A. M. Ljapunov.....	9
Obrázek 6:	O. RöSSLer.....	9
Obrázek 7:	L. D. Landau.....	9
Obrázek 8:	Původní Lorenzův graf.....	11
Obrázek 9:	Hausdorffova dimenze.....	21
Obrázek 10:	Kochova křivka – mřížková dimenze.....	23
Obrázek 11:	Cantorova množina.....	34
Obrázek 12:	Juliovy množiny pro $c = 0,4 + 0,32i$ a pro $c = 0 - 0,8i$	36
Obrázek 13:	Juliovy množiny pro $c = -1,1 + 0,25i$ a pro $c = -0,12 + 0,9i$	36
Obrázek 14:	Juliovy množiny pro $c = -0,45 + 0,25i$ a pro $c = -0,8 + 0,13i$	37
Obrázek 15:	Mandelbrotova množina.....	37
Obrázek 16:	Fraktální komprese obrazu.....	42
Obrázek 17:	Chua obvod.....	43
Obrázek 18:	Dvojitě kyvadlo.....	45
Obrázek 19:	Kyvadlo s magnety.....	46
Obrázek 20:	Lorenzovo vodní kolo.....	47
Obrázek 21:	Kolonie bakterií.....	55
Obrázek 22:	Reakce Bělousov – Žabotinskij.....	56

10. Seznam tabulek

Tabulka 1:	Logistická funkce.....	18
Tabulka 2:	Hausdorffova dimenze.....	22
Tabulka 3:	Kochova křivka – mřížková dimenze.....	23
Tabulka 4:	Vlastnosti KS entropie.....	26
Tabulka 5:	Populační model Lotka – Volterra.....	49