

Využití deterministického chaosu pro aplikaci generátoru náhodných čísel v prostředí Mathematica a WebMathematica

The use of deterministic chaos for the application of the random numbers in an environment of Mathematica and WebMathematica

Jiří Barák

Bakalářská práce
2012



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jiří BARÁK**
Osobní číslo: **A08022**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Využití deterministického chaosu pro aplikaci generátoru náhodných čísel v prostředí Mathematica a WebMathematica**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Vytvořte v SW Mathematica aplikaci pro generování náhodných čísel pomocí chaotických systémů.
3. Převeďte vytvořenou aplikaci do prostředí WebMathematica.
4. Umístěte webovou prezentaci na server UTB.
5. Diskutujte a navrhněte možnosti využití pro kryptografické techniky či moderní metody softcomputingu.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **HOSTE, Jim. Mathematica DeMYSTiFied. McGraw-Hill Professional, 2008. 408 s. ISBN 978-0071591447.**
2. **RUSKEEPAA, Heikki. Mathematica Navigator: Mathematics, Statistics and Graphics, Third Edition. Academic Press, 2009. 1136 s. ISBN 978-0123741646.**
3. **STEJSKAL, J. Vytváříme WWW stránky pomocí HTML, CSS a JavaScriptu. Computer Press, 2006, ISBN: 80-251-0167-3.**
4. **GILMORE, R.; LEFRANC, M. The Topology of Chaos. Willey VCH, 2002. 518 s. ISBN 978-0-471-40816-1.**
5. **RESIG JOHN, Javascript a AJAX – Moderní programování webových aplikací, 2007. Computer Press. ISBN: 978-80-251-1824-5.**
6. **GLEICK, James. Chaos: vznik nové vědy. Ando Publishing, 1996. 350s. ISBN 80-86047-04-0.**
7. **HORÁK, Jiří. Deterministický chaos a jeho fyzikální aplikace. Academia, 2003. 437 s. ISBN 8020009108.**
8. **KOVÁČOVÁ, M. webMathematica. STU Bratislava, 2007. 178 s. ISBN 80-969562-1-3.**

Vedoucí bakalářské práce:

Ing. Roman Šenkeřík, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce:

24. února 2012

Termín odevzdání bakalářské práce:

8. června 2012

Ve Zlíně dne 24. února 2012

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Hlavní cílem této práce bylo vytvoření funkční aplikace pro generování náhodných čísel s využitím deterministického chaosu. V teoretické části je objasněno několik základních pojmů z oblasti deterministického chaosu, stability chaosu a jeho vizualizace. Jsou zde i základní informace o generování náhodných čísel. Praktická část obsahuje popis naprogramovaných systémů a popis celé aplikace.

Klíčová slova: deterministický chaos, atraktor, bifurkace, náhodná čísla, mathematica, webmathematica

ABSTRACT

The main topic of this thesis was to create a functional application generating random numbers using deterministic chaos. In the theoretical part is explained some of basic terms of deterministic chaos, chaos stability and its visualization. There are also basic information about generating random numbers. The practical part contains a description of programmed systems and a description of the entire application.

Keywords: deterministic chaos, attractor, bifurcation, random numbers, Mathematica, webMathematica

Na tomto místě bych rád poděkoval mému vedoucímu bakalářské práce Ing. Romanu Šenkeříkovi, Ph.D. za jeho podnětné připomínky, ochotu při konzultacích, podporu a trpělivost při psaní této bakalářské práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 TEORIE CHAOSU	11
1.1 VÝVOJ TEORIE CHAOSU	11
2 DETERMINISTICKÝ CHAOS	13
2.1 VIZUALIZACE A IDENTIFIKACE CHAOSU	13
2.1.1 Dynamické systémy.....	13
2.1.2 Atraktory	14
2.1.3 Fraktály.....	16
2.2 BIFURKAČNÍ ANALÝZA	18
2.2.1 Typy bifurkací	18
2.2.2 Bifurkační diagramy.....	19
2.3 STABILITA SYSTÉMU A LJAPUNOVY EXPONENTY.....	20
2.4 ROZDĚLENÍ DYNAMICKÝCH SYSTÉMŮ	21
3 GENEROVÁNÍ NÁHODNÝCH ČÍSEL	22
3.1 FYZIKÁLNÍ GENERÁTORY NÁHODNÝCH ČÍSEL	22
3.2 PSEUDONÁHODNÁ ČÍSLA	22
3.2.1 Lineární kongruentní generátor	23
3.3 GENERÁTORY NÁHODNÝCH ČÍSEL POMOCÍ TEORIE CHAOSU	24
II PRAKTICKÁ ČÁST	25
4 MATHEMATICA A WEBMATHEMATICA	26
4.1 MATHEMATICA.....	26
4.2 WEBMATHEMATICA.....	27
4.2.1 JavaServlet	28
4.2.2 Java Server Pages	28
5 TVORBA APLIKACE PRO GENEROVÁNÍ NÁHODNÝCH ČÍSEL	29
5.1 VYTVOŘENÉ DYNAMICKÉ SYSTÉMY V SOFTWARE MATHEMATICA.....	29
5.1.1 Non-invertibilní systémy.....	29
5.1.1.1 Logistic map.....	29
5.1.1.2 Cubic map	33
5.1.2 Disipativní systémy	33
5.1.2.1 Burgers map	34
5.1.2.2 Tinkerbell map.....	35
5.1.2.3 Delayed map	36
5.1.2.4 Henon map.....	36
5.1.2.5 Lozi map	36
5.1.2.6 Dissipative standard map.....	36
5.1.2.7 Ikeda map.....	36
5.1.2.8 Sinai map	37

5.1.2.9	Arnold's cat map.....	37
5.1.2.10	Chirikov map.....	37
5.1.3	Oscilátory	37
5.1.3.1	Damped driven pendulum oscillator	37
5.1.3.2	Ueda oscillator	38
5.1.3.3	Duffing-van der Pool oscillator.....	39
5.1.3.4	Hadley circulation	39
5.1.3.5	Chua's circuit.....	39
5.1.3.6	Henon-Heiles system	39
6	VYTVORENÍ WEBOVÉ PREZENTACE V PROSTŘEDÍ WEBMATHMATICA	40
6.1	PŘEVOD DO PROSTŘEDÍ WEBMAHTEMATICA	40
6.2	NÁHLED NA WEBOVOU PREZENTACI	41
7	MOŽNOSTI VYUŽITÍ GENERÁTORU NÁHODNÝCH ČÍSEL.....	44
	ZÁVĚR	45
	ZÁVĚR V ANGLIČTINĚ.....	46
	SEZNAM POUŽITÉ LITERATURY.....	47
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	49
	SEZNAM OBRÁZKŮ	50
	SEZNAM PŘÍLOH.....	51

ÚVOD

Slovní spojení „deterministický chaos“ se může na první pohled zdát zavádějící až nesmyslný. Většina lidí si pod pojmem „chaos“ představí stav neuspořádanosti, stav bez jakýchkoliv struktur. Slovo chaos je řeckého původu, kde označovalo nepředvídatelnost a neuspořádanost. V abstraktním smyslu slova se jedná o opak zákona a pořádku, naopak v matematice označuje toto slovo deterministické chování, to znamená chování citlivé na počáteční podmínky.

Tento deterministický chaos je možné využít při potřebě získání náhodných dat, ať už pro potřeby šifrování, počítačového modelování nebo v případě hazardních her.

I. TEORETICKÁ ČÁST

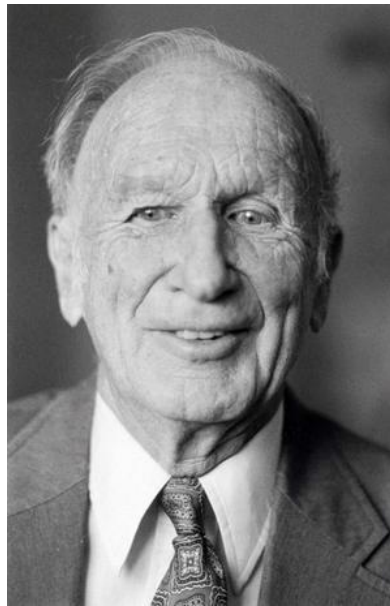
1 TEORIE CHAOSU

Podstata teorie je snaha popsat skutečně reálné fyzikální jevy, bez zanedbávání okolních vlivů. Tyto fyzikální modely jsou pak velice citlivé na počáteční podmínky tzn. malou změnou počátečního stavu na vstupu dostáváme velké rozdíly na výstupu. Díky tomu se tedy takové systémy jeví jako nahodilé, nepředvídatelné, chaotické. Chaotické systémy jsou nicméně předpověditelné, deterministické. Proto často hovoříme o deterministickém chaosu a uspořádané neuspořádanosti. [1]

1.1 Vývoj teorie chaosu

Základy teorie chaosu se poprvé objevují v roce 1900 ve studiích Henri Poincarého o problému pohybu tří objektů se vzájemnou gravitační silou, tzv. problému tří těles. Poincaré objevil, že mohou existovat orbity, které jsou neperiodické, a které nejsou ani neustále vzrůstající ani se neblíží k pevnému bodu.

Později až kolem roku 1960 se začaly objevovat první počítače bylo možné v rozvíjení teorie chaosu pokračovat. Jedním z největších průkopníků byl Edward Lorenz, americký matematik. Za druhé světové války začal pracovat jako meteorolog, a protože v sobě nezapřel matematické myšlení, snažil se popsat vývoj počasí jednoduchými rovnicemi. Ty potom důkladně zkoumal na počátku 60. let 20. století, když už byly k dispozici první počítače. Byly ale velmi pomalé, a tak když chtěl jednou svůj výpočet zopakovat, neprovedl ho celý, ale jenom od poloviny. Zadal hodnoty spočítané v polovině výpočtu jako počáteční, ovšem zaokrouhlil je o pár řádů. Myslel, že taková drobná nepřesnost nemůže nic způsobit, avšak mýlil se. Dosažené výsledky se značně lišily od těch původních. Objevil tak první z chaotických systémů. Tato objevená citlivost se stala známou jako „motýlí efekt“.



Obr. 1 Edward Lorenz (zdroj: internet)

Toto zjištění představovalo obrovskou ránu pro tehdejší meteorology, jelikož ti ve svých snahách směřovali k úplnému ovládnutí počasí.

Definice chaosu říká, že neuspořádanost může být jednoduše vyšším řádem složitosti, která vzniká ze zcela deterministických procesů, to znamená, že v rámci chaosu (součtu deterministických jevů) existuje současně jakási podivná organizovanost. [2]

2 DETERMINISTICKÝ CHAOS

Pro deterministický chaos je typické, že:

- je extrémě citlivý na počáteční podmínky či změny některých řídicích parametrů
- dynamický systém se musí nacházet v nestabilním stavu nebo v jeho blízkosti
- systém je globálně stabilní, ale lokálně nestabilní a nepředpovědatelný
- chyby v počátečních naměřených hodnotách exponenciálně narůstají, proto nelze vývoj dynamických systémů citlivých na počáteční podmínky využít pro předpovědi jejich chování

Tohle jsou nejdůležitější deterministicky formulované atributy chaosu. [2]

2.1 Vizualizace a identifikace chaosu

2.1.1 Dynamické systémy

Dynamické systémy jsou vytvářeny na konkrétních soustavách, mechanických, elektromechanických i obecně komplexních. Využíváme je k popisu chování a vlastností v závislosti na nezávislé veličině času t a to jak v diskrétních okamžicích, tak i pro spojitý čas. Jednou variantou je **evoluční rovnice v diskrétních časových okamžicích** t_k , $k = 1, 2, \dots$ bývá popisována soustavou algebraických rovnic (map) $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k)$, kde \mathbf{x} vektor konečné dimenze n , \mathbf{x}_k a \mathbf{x}_{k+1} jsou stavové vektory v časech. Funkce $\mathbf{f}(\dots)$ je vektorovou funkcí řádu n . [2]

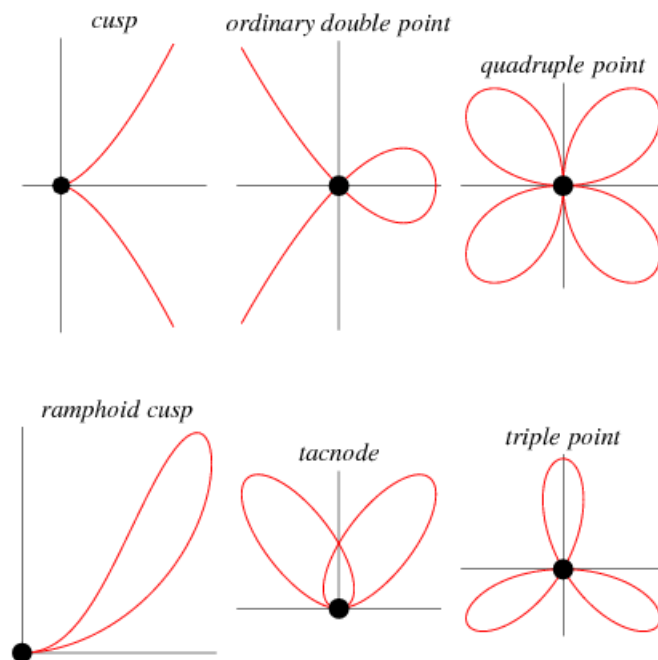
Řešení dynamických systémů v konstatních časových okamžicích představují rovnovážné stavy. Grafická zobrazení diferenciálních rovnic v rozšířeném stavovém prostoru $\mathbb{R}^n + \mathbb{R}^1$ představují integrální křivky, které nazýváme orbitami. Při různých počátečních podmínkách se může měnit vývoj fázových trajektorií a navíc mohou tyto soustavy obsahovat parametry, které se vlivem vnějších a vnitřních podmínek mohou měnit. Při určitých hodnotách těchto parametrů může dojít ke strukturálním změnám chování k tzv. **bifurkacím**. Objevuje se rozdělení fázového prostoru na oblasti stabilních a nestabilních počátečních stavů i rozdělení prostoru parametrů, ve kterých nedochází k bifurkacím a na hranice, kde se objevují.

2.1.2 Atraktory

Slouží ke grafické interpretaci dynamického chování soustav. Atraktor je množina stavů systému, do kterých systém směřuje s časem $t > t_0$. U disipativních soustav dojde po odeznění přechodové odezvy k některému z ustálených stavů:

- je-li buzení konstatní, bude ustálený stav pevným bodem ve stavovém prostoru

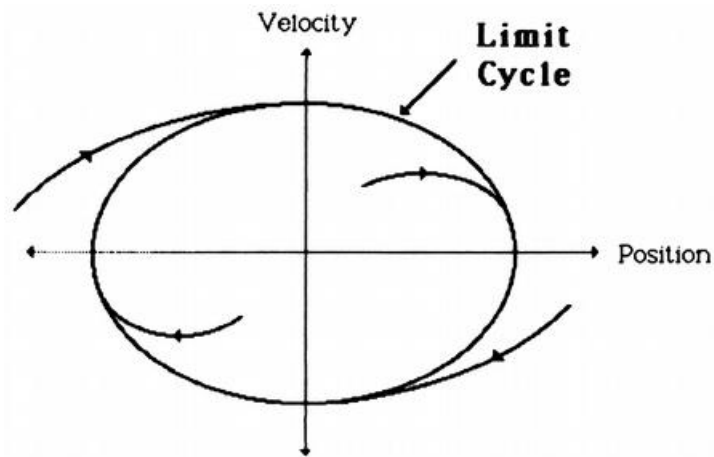
Jsou-li atraktorem dynamického systému pevné body, jedná se o nejjednodušší případ, kdy se systém v nekonečném čase ustálí v určitém stabilním stavu a v podstatě již dále nejde o dynamický systém. [3]



Obr. 2 Limitní body (zdroj: internet)

- je-li buzení periodické, bude ustálený stav uzavřenou křivkou (limitní cyklus)

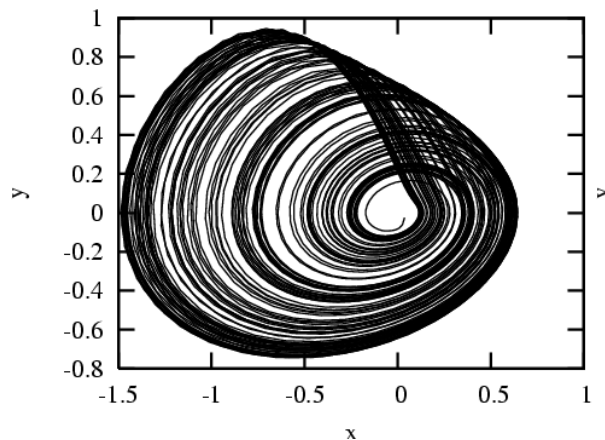
Jsou-li atraktorem periodické body, stále se jedná o jednoduchý případ, kdy se systém ustálí tak, že osciluje mezi několika stavy. Příkladem může být malé těleso ustálené na eliptické dráze planety.



Obr. 3 Limitní cyklus (zdroj: internet)

- stav chaosu reprezentuje chaotický atraktor, což je složitá limitní množina bodů, která se vyznačuje nestabilním chováním

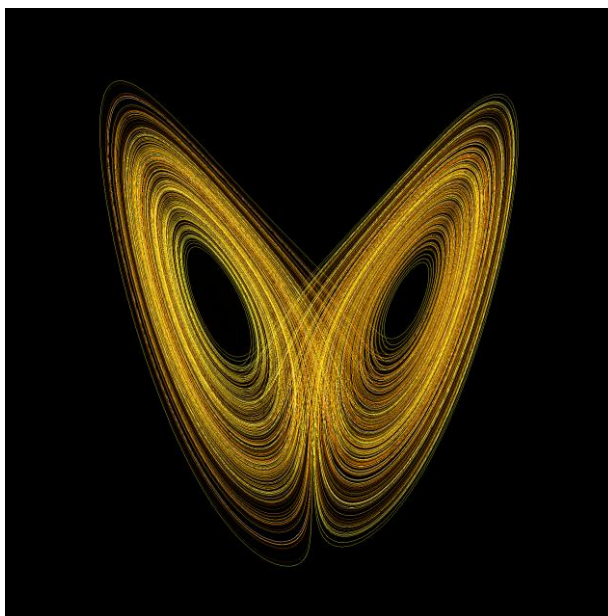
Je-li atraktor chaotický, nelze výsledný stav systému nijak dopředu předpovědět. Je to způsobeno extrémní citlivostí na počáteční podmínky. Chaotičnost v tomto případě neznamena náhodnost, protože jde o deterministické systémy.



Obr. 4 Chaotický atraktor (zdroj:internet)

- můžeme se setkat s „podivným atraktorem“ (strange atraktor), reprezentovaným složitým útvarem

Termín podivný atraktor není exaktně matematicky definován, ale považujeme za něj takový atraktor, který vykazuje stejné vlastnosti, jaké mají fraktály. [2]

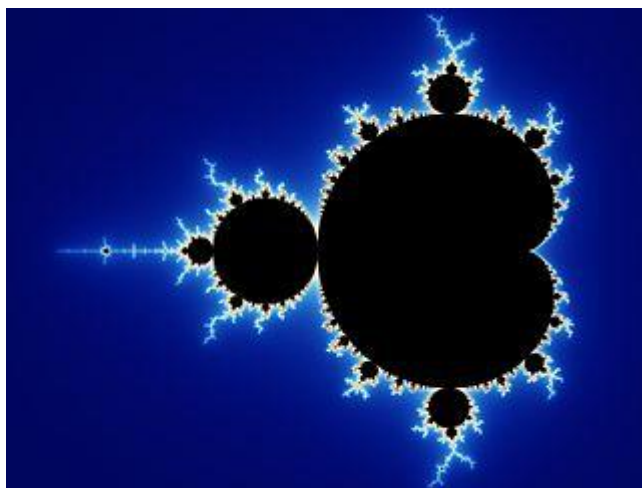


Obr. 5 Lorenzův „podivný“ atraktor (zdroj: internet)

2.1.3 Fraktály

Nejdříve je potřeba si objasnit pojem topologická dimenze. Tato dimenze určuje, v kolika na sebe kolmých směrech v prostoru můžeme pracovat a zobrazovat výsledky. Topologická dimenze je celým číslem. Například: bod má nulovou dimenzi, úsečka jednu, plošný útvar má dvě atd. Existuje zde ještě jedna dimenze Hausdorffova, někdy nazývaná též fraktální dimenze. Na rozdíl od topologické dimenze není fraktální dimenze celé číslo. Tato dimenze spolu s topologickou nám pomáhá určit, jak je útvar členitý. Pokud se tyto dimenze liší málo, je útvar málo členitý a obráceně.

Co je to tedy fraktál? Fraktál je množina, jejíž Hausdorffova dimenze je větší než topologická dimenze (B. Mandelbrot). Matematická definice tohoto pojmu zatím neexistuje. Existuje také obecná definice, *podle které je fraktál takový útvar, při jehož zvětšení dostaneme opět stejný obraz bez ohledu na měřítko (tzv. invariance vzhledem k měřítku)*. Od šedesátých let minulého století se intenzivně rozvíjí Fraktální geometrie jako vědní disciplína. Zde se začaly rozlišovat jednotlivé typy fraktálů, vhodné pro řešení zcela určitého typu problémů. Rozdělení fraktálů je následující: tzv. L-systémy, systémy iterovaných funkcí, dynamické systémy – chaos a tzv. nepravidelné fraktály – statisticky soběpodobné. [4]



Obr. 6 Ukázka fraktálů – Mandelbrotova množina (zdroj: internet)



Obr. 7 Juliovy množiny (autor: Petra Holbíková)

2.2 Bifurkační analýza

Pojem bifurkace označuje jev, při kterém dochází k velkým změnám vnitřního stavu sledovaného systému, a to v případě nepatrných změn na vstupních podmínkách. U mnoha běžných systémů k bifurkacím nedochází, vnitřní stavy těchto systémů se mění jen minimálně dokonce až lineárně. U některých systémů při dosažení kritických hodnot parametrů na vstupu dochází k prudké změně vnitřního stavu, která může dovést tyto systémy až do chaotického stavu. Bifurkaci lze také chápat jako náhlou změnu stability, která bezprostředně souvisí s chaosem, předchází mu.

Jinými slovy, chaos nastává v případě, kdy v soustavě začíná docházet k postupnému vzniku na sebe navazujících bifurkací. [2]

2.2.1 Typy bifurkací

Rozdělením bifurkací se zabývá bifurkační analýza, která analyzuje chování dynamických systémů v závislosti na změnách některých parametrů. Na první pohled je vidět, že tato analýza je velice rozsáhlá a z těchto důvodů se využívají různé speciální strategie.

Nejužívanější strategie je identifikace bifurkačních bodů, které v parametrickém prostoru oddělují oblasti, ve kterých se dynamický systém chová odlišně, speciálně jde o oblasti se stabilním a nestabilním chováním. V okolí bifurkačního bodu pak hovoříme o strukturální nestabilitě systému. [5]

Bifurkace můžeme rozdělit na dvě základní skupiny, na **lokální a globální bifurkace**.

- a) Lokální bifurkace mohou být analyzovány na základě posuzování lokálních změn stability periodických orbit. Patří sem následující bifurkace:
 - bifurkace sedlo – uzel
 - bifurkace typu vidlice
 - transkritická bifurkace
 - Hopfova bifurkace
 - Neimarkova bifurkace

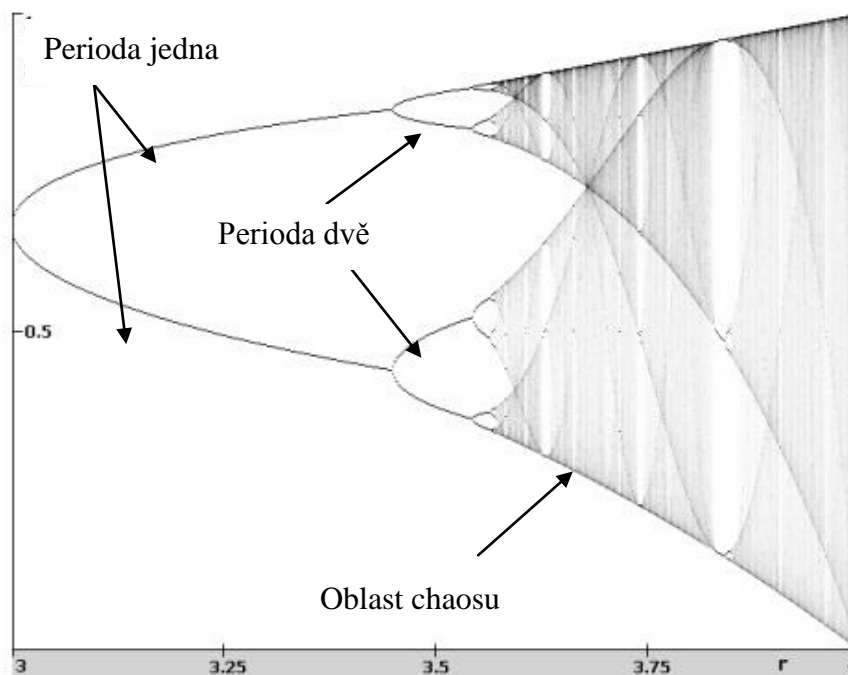
b) Globální bifurkace se objevují tehdy, když sousedící oblasti, charakterizující vlastnosti systému si navzájem překáží. Pokud tyto změny v topologii trajektorií zahrnují velký prostor, nazýváme je globálními. Příklady těchto bifurkací jsou:

- homoklinická bifurkace, kde si překáží limitní cyklus se sedlovým bodem
- heteroklinická bifurkace, kde si překáží limitní cyklus se dvěma a více sedlovými body
- bifurkace s neohrazenou periodou
- globální bifurkace se mohou objevovat jako chaotické atraktory

2.2.2 Bifurkační diagramy

Chaos nastává v případě, že v systému začíná docházet k velkému počtu na sebe navazujících bifurkací. Grafickým znázorněním těchto jevů je **bifurkační diagram**, který představuje závislost limitních stavů systému na některém z řídicích parametrů. Příkladem bifurkačního diagramu může být grafické znázornění logistické rovnice popisující vývoj živočichů.

$$x(k+1) = a \cdot x(k) \cdot (1 - x(k))$$



Obr. 8 Bifurkační diagram populační rovnice (zdroj: internet, úprava: Jiří Barák)

Kde a je řídicí parametr představující míru nelinearity a $k = 0, 1, 2, \dots$ je diskrétní tok. Na diagramu jsou vidět oblasti řešení: limitní cykly s periodou jedna, oblasti s limitní periodou dvě i oblast chaosu. Podoblasti v diagramu vypadají podobně a zdá se, že tvoří jakýsi celek. Oblast začínající druhou bifurkací vypadá stejně jako oblast s třetí bifurkací a tak dále. Rozdíly mezi následujícími body bifurkace se zmenšují.

2.3 Stabilita systému a Ljapunovovy exponenty

Reálné systémy se vyznačují tím, že objem svého stavového prostoru se zmenšuje a směřuje ke konečné podobě nazývané atraktor dynamického systému. Vývoj a chování dynamických systémů můžeme popsat evoluční rovnicí

$$\dot{x} = f(x_1, x_2, \dots, x_n) \rightarrow i = 1, 2, \dots, n,$$

kde x_i je i -ta stavová veličina a $f_i(\dots)$ je jí odpovídající nelineární funkce. Dynamický systém má atraktor, pokud je splněna podmínka

$$\sum_{i=1}^n \frac{\partial f_i}{\partial x_i} < 0$$

Pokud je systém v nestabilním stavu, tak se jeho dvě blízké trajektorie vzdalují rychleji než polynomiálně. Bude-li vzdálenost dvou bodů na trajektoriích r , můžeme změnu této hodnoty vyjádřit vztahem

$$r(t) = r_0 e^{\Lambda t}$$

kde $r_0 = r(0)$ je počáteční hodnota vzdálenosti r a Λ je tzv. lokální Ljapunovův exponent. Ve vícerozměrném stavovém prostoru je definováno tzv. globální spektrum Ljapunovových exponentů, kdy každé stavové veličině odpovídá jeden, pro ně platí vztah

$$\Lambda_i = \lim_{t \rightarrow \infty} \frac{1}{t} \ln \frac{r_i(t)}{r_i(0)},$$

kde i je index stavové veličiny jedné dimenze. Známe-li hodnoty veličin Λ_i , můžeme hodnotit stabilitu chování dynamického systému.

Podmínka stability je následující: **dynamický systém má stabilní chování, pokud jsou všechny Ljapunovovy exponenty nekladné, je-li alespoň jeden exponent kladný, systém se stává chaotickým.**

Pokud má systém více kladných Ljapunovových exponentů, nazýváme jej hyperchaotickým.

Výpočet těchto koeficientů je obtížný, ale jejich znalost jednoznačně určuje chaotické chování systému. [5]

2.4 Rozdělení dynamických systémů

Základním rozdělením dynamických systémů je na spojité a iterované. Dalším rozdělením je dělení na Hamiltonovské systémy a disipativní systémy.

Hamiltonovské systémy se vyznačují dynamikou, která ve fázovém prostoru zachovává objem, na rozdíl od systémů disipativních. Uzavřené systémy hamiltonovského typu mohou modelovat pouze úzký výsek přírodních jevů.

3 GENEROVÁNÍ NÁHODNÝCH ČÍSEL

Generátor náhodných čísel je zařízení nebo procedura generující náhodná nebo jen zdánlivě náhodná čísla. Parametrem pro generování náhodných čísel většinou bývá počáteční hodnota, od které čísla začínají, dále pak maximální hodnota, kterou nesmí překročit, někdy také maximální krok mezi těmito čísly.

Pro generování náhodných čísel se v informatice používají dvě metody. Prvním je generování pseudonáhodných čísel tzv. PRN generator a druhým je Generátor pravých náhodných čísel tzv. TRN generator, který vypisuje náhodnost z fyzikálních jevů.[6]

3.1 Fyzikální generátory náhodných čísel

Fyzikální generátor může být založen na mnoha principech, ale vždy se jedná o měření jevu, který se zdá být náhodný. Jedním z nich je chování atomárních jevů, jež sledujeme pomocí kvantové mechaniky, další variantou je sledování rozpadu radioaktivního zdroje, kde jsou časové intervaly rozpadu zcela nepředpověditelné.

Jedním ze zajímavých generátorů byl Lavarand, který používal snímky lávové lampy pro generování skutečně náhodných čísel. Extrahoval data z obrázků lávové lampy a ty pak použil jako počáteční podmínky pro pseudonáhodný generátor. Ačkoliv Lavarand používá pseudonáhodný generátor čísel, jedná se o pravý generátor náhodných čísel, protože se zde stále využívá nahodilosti z lávové lampy. [7], [8]

3.2 Pseudonáhodná čísla

Generátory pseudonáhodných čísel se využívají k mnoha rozmanitým účelům. Pomocí výpočetní techniky se dají v krátkých časech vygenerovat poměrně dlouhé posloupnosti „náhodných čísel“. Tato čísla jsou deterministicky vypočtena z hodnoty nazývané klíč. Nevýhodou těchto generátorů je, že trpí periodicitou. To znamená, že po určité době se tato sekvence čísel začne opakovat a tím se snižuje kvalita rozdělení četnosti čísel. Jedním z nejpoužívanějších algoritmů je lineární kongruentní generátor.

3.2.1 Lineární kongruentní generátor

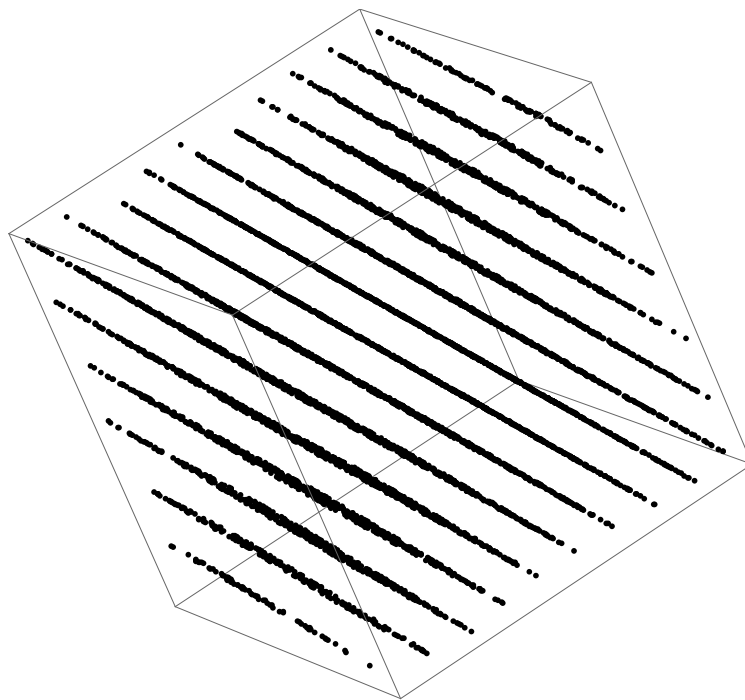
Lineární kongruentní generátor je jeden z nejjednodušších generátorů pseudonáhodných čísel. Princip tohoto generátoru spočívá ve vztahu

$$x_{i+1} = (ax_i + c) \bmod m,$$

kde mod znamená zbytek po celočíselném dělení, a, c, m jsou vhodně zvolené konstanty. Počáteční nastavení x_0 se nazývá „random seed“ – náhodné semínko. Generátor generuje čísla s rovnoměrným rozložením v intervalu

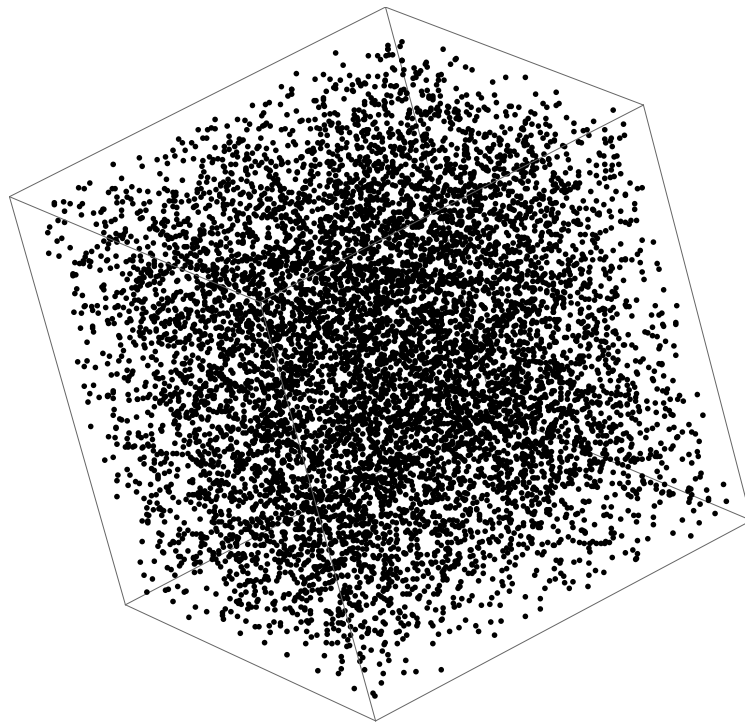
$$0 \leq x_i < m.$$

Jelikož je počet čísel v tomto intervalu omezen, začne se nejpozději po m vygenerovaných číslech opakovat stejná posloupnost – perioda generátoru. Větším problémem je nerovnoměrné zaplnění prostoru při špatné volbě parametrů, tím se proslavil generátor RANDU, který se používal v 70. letech 20. století. Jeho problémem bylo, že body byly rozloženy v 15 rovinách a zbytek prostoru zůstal neobsazen (viz obrázek 8).



Obr. 9 RANDU se špatně zvolenými parametry (zdroj: internet)

Při trošku odlišně zvolených parametrech, kdy bylo pouze opraveno $m = 2^{39}$, je výsledek daleko uspokojivější. [9]



Obr. 10 RANDU s opraveným parametrem $m = 2^{39}$ (zdroj: Jiří Barák)

3.3 Generátory náhodných čísel pomocí teorie chaosu

Dynamické systémy při určitých hodnotách parametrů vykazují chaotické chování, které bylo popisováno výše. Pro generování náhodných čísel pomocí teorie chaosu bude použito několik těchto známých chaoticky se chovajících systémů, kdy nedokážeme předpovědět výsledek pro dané počáteční podmínky. Tyto počáteční podmínky pro nás budou vstupním parametrem pro vygenerování náhodných čísel.

II. PRAKTICKÁ ČÁST

4 MATHEMATICA A WEBMATHEMATICA

Mathematica je výpočetní software, který se používá například ve vědeckých, technických nebo matematických odvětvích. Tvůrcem jejího konceptu byl Stephen Wolfram a vyvinuta byla ve Wolfram Research v Champaign ve státě Illinois.

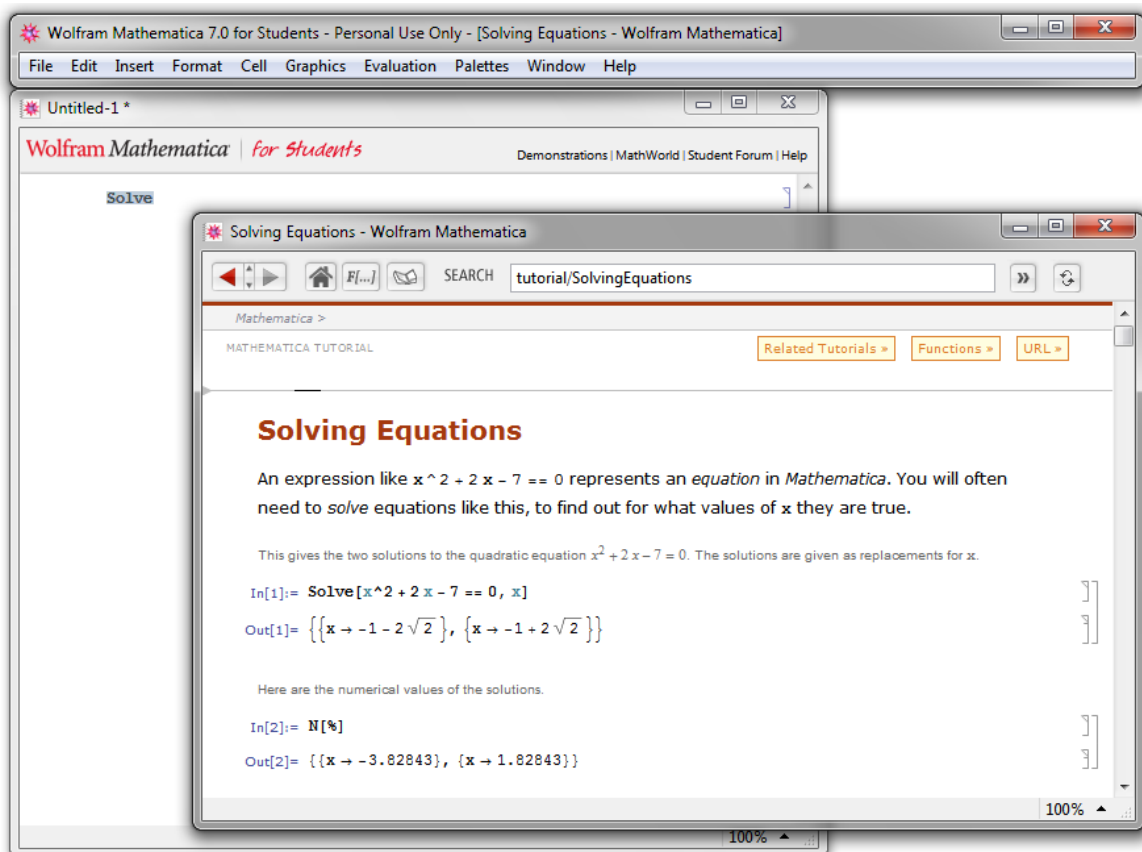
4.1 Mathematica

Mathematica je rozdělena do dvou částí: jádro a „front end“. Jádro se stará o interpretaci kódu a vrací výsledky výrazů. Část front end poskytuje GUI, který umožňuje editaci notebook dokumentů. Gui podporuje většinu funkcí textových procesorů, ale je tam jen jedna úroveň funkce zpět. Notebooky jsou strukturovány pomocí buněk, které umožňují rozdělení dokumentu a automatické indexování.

Mezi základní vlastnosti softwaru mathematica patří:

- základní a speciální funkce matematické knihovny
- maticové a manipulační nástroje
- práce s komplexními čísly a symbolické výpočty
- 2D a 3D vizualizace
- řešitelé pro systémy rovnic
- nástroje pro vizualizaci a analýzu grafů
- a další...

Pro zobrazení aplikace napsané v mathematice existuje několik možností. Jednou z možností je Mathematica Player Pro, zde je možné spustit jakoukoliv aplikaci, ale neumožňuje editaci a vytváření kódu. Další variantou je Wolfram CDF player, který je určen pro spuštění aplikací uložených v Computable Document Format. Tento CDF player umí i otevřít běžné notebooky, ale nespustí je. Poslední možností zpracování aplikace je webMathematica.[10]

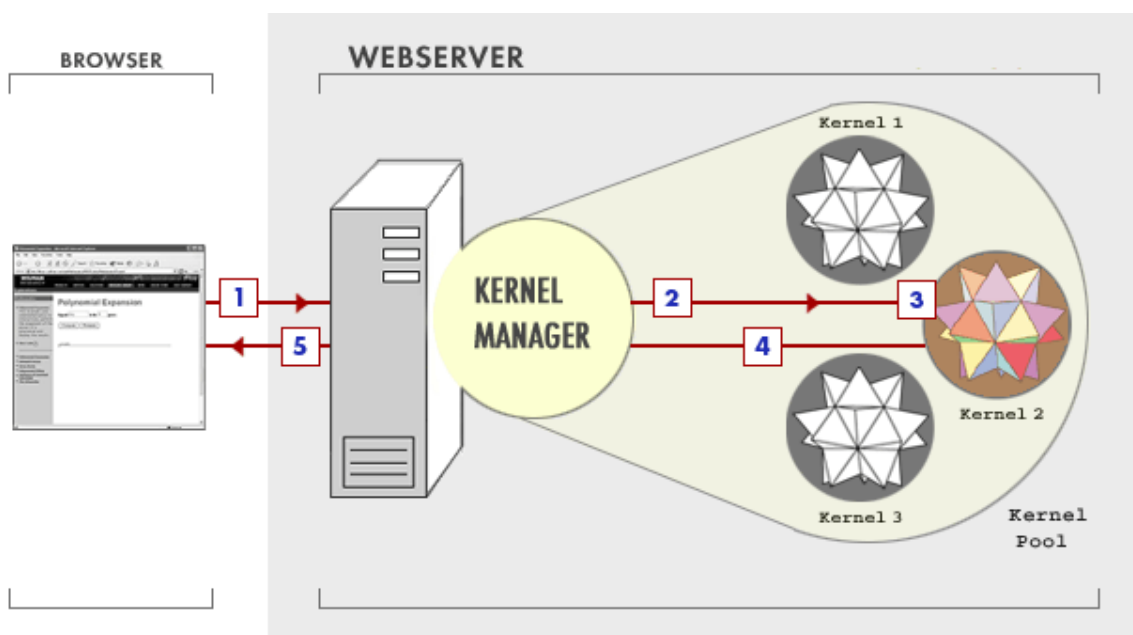


Obr. 11 vývojové prostředí Mathematica (zdroj: internet)

4.2 WebMathematica

Produkt, který slouží k tvorbě dynamických webových stránek, využívajících funkce matematiky. WebMathematica je založena na dvou java technologiích. JavaServlet a Java Server Pages.

Princip funkčnosti Webmathematicy je následující. V prvním kroku pošle webový prohlížeč požadavek na Webmathematica server (1). Poté si server rezervuje Mathematica kernel, výpočetní jádro, z „bazénu“ (2). Mathematica kernel zpracuje kalkulace a vrátí výsledky serveru (3). Poté se kernel vrátí do bazénu a webMathematica server vrátí výsledky prohlížeči (4 a 5). [10]



Obr. 12 Princip komunikace WebMathematicy (zdroj: Wolfram research)

4.2.1 JavaServlet

Java Servlet jsou součástí platformy J2EE od firmy SUN. Jsou to programové komponenty běžící na straně serveru, napsané v pomoci jazyku Java.

Servlety jsou načítány a spouštěny pomocí JVM aplikačním serveru. Odpovědi těchto servletů je potom v většině případů HTML kód. [11]

4.2.2 Java Server Pages

Java Server Pages (dále jen JSP) je nástroj pro psaní dynamických HTML stránek založený na jazyce Java, funkčností velmi podobný PHP. Jedná se vlastně o HTML stránky, do kterých je pomocí speciálních značek, v případě WebMathematicy značkami `<msp:evaluate>`, vložen kód v Javě, který se provádí při vyřizování dotazu na straně serveru. Hlavní odlišností je kompilace stránek do tzv. servletů (viz. kapitola 4.2.1), které pak komunikují s Web serverem. JSP by se tedy daly také charakterizovat jako nástroj na psaní servletů. Každá HTML stránka je také JSP stránkou, pro odlišení se jim dává přípona `.jsp`. [12]

5 TVORBA APLIKACE PRO GENEROVÁNÍ NÁHODNÝCH ČÍSEL

5.1 Vytvořené dynamické systémy v software Mathematica

Hlavní myšlenkou vytvoření generátoru čísel bylo využití citlivosti na počáteční podmínky těchto dynamických systému, které při určitých hodnotách řídicích parametrů vykazují chaotické chování.

Prvním krokem při tvorbě generátoru náhodných čísel bylo třeba vytvořit samotnou aplikaci v softwaru Mathematica. Tato část pak bude v konečném kódu stránek, zpracovaných ve Webmathematice, obstarávat samotné výpočty.

Generátory náhodných čísel byly rozděleny do tří kategorií. První kategorií jsou jednodimenzionální mapy (non-invertibilní systémy), druhou pak tvoří disipativní systémy a třetí skupinou jsou oscilátory.

Výstupem těchto naprogramovaných bloků budou samotná generovaná čísla, bifurkační diagram pro jednodimenzionální mapy, ostatních případech to bude atraktor systému, dále pak na výstupu bude graf zobrazující tzv. mapu systému a graf zobrazující průběh funkce. V neposlední řadě byl zařazen na výstup i histogram zobrazující četnost výskytu generovaných čísel v intervalech, jelikož tento údaj o rozložení čísel je důležitý pro hodnocení generátoru čísel.

5.1.1 Non-invertibilní systémy

Samotná mapa je graf zobrazující závislost x_{n+1} na x_n . Existuje celá řada jednodimenzionálních map. Například Logistic map, dobře známá jako logistická rovnice, Sin map, Cubic map, Tent map, Gauss map a spousta dalších. Pro naprogramování byla vybrána Logistic map a Cubic map.

5.1.1.1 Logistic map

Již dobře známá logistická rovnice, jelikož tato rovnice byla první, která odhalila dosud nepředpokládané chaotické chování. Rovnice má tento tvar

$$x_{n+1} = Ax(1 - x_n).$$

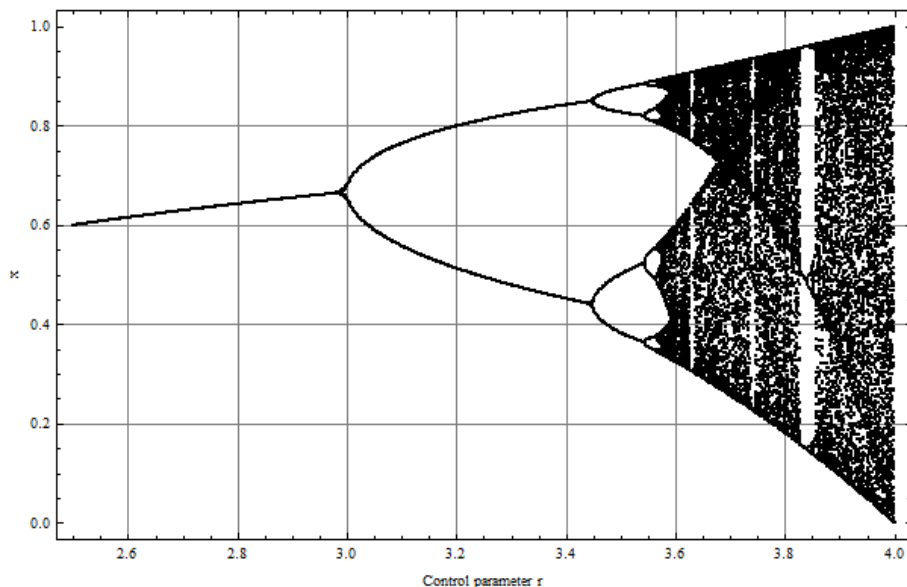
Jako první se tato rovnice v mathematice zkompilevala, aby mohla později být volána jako funkce s parametry x a A .

```
Logistic = Compile[{{x, _Real}, {A, _Real}}, A x (1 - x)]
CompiledFunction[{{x, A}, A x (1 - x), -CompiledCode-]
```

Obr. 13 kompilace logistické rovnice (autor: Jiří Barák)

Poté byly zpracovány jednotlivé bloky výstupu. První byl naprogramován bifurkační diagram. V tomto případě byly použity hodnoty paramteru A z intervalu $< 2.5, 4 >$ s krokem 0.002. Blok funguje tak, že pomocí funkce nestlist a table vytvoří tabulku o 300 hodnotách, které funkce nestlist vygenerovala dosazováním paramteru A a počáteční podmínky $x_{\text{start}} = 0.1$, z této tabulky pak vytvoří bifurkační diagram.

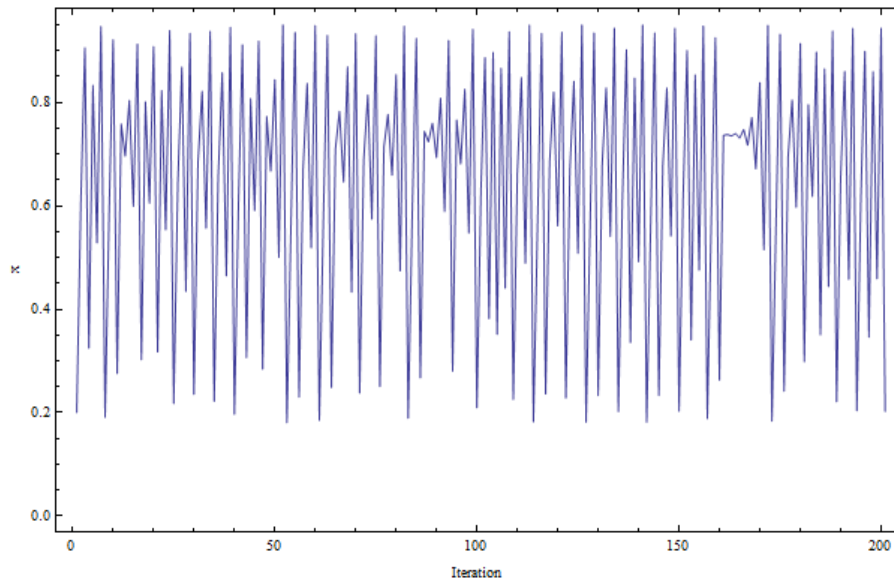
```
LogisticBifData = Table[Take[NestList[{A, Logistic[#1[[2]], A]} &, {0, xstart}, 300], -100], {A, 2.5, 4, 0.002}];
lq = ListPlot[Flatten[LogisticBifData, 1], PlotRange -> All, PlotJoined -> False, Axes -> {True, False},
  GridLines -> Automatic, Frame -> True, PlotStyle -> {RGBColor[0, 0, 0], PointSize[0.003]},
  FrameLabel -> {"Control parameter r", "x"}, ImageSize -> 600]
```



Obr. 14 bifurkační diagram Logistické rovnice (autor: Jiří Barák)

Pro další pokračování se používá hodnota paramteru $A = 3.8$. Dalším krokem bylo vytvoření grafu zobrazující průběh funkce pro hodnotu x . Zde byla opět použita vygenerovaná data z logistické rovnice pomocí funkce nestlist a opět vykreslena v jiném grafu.

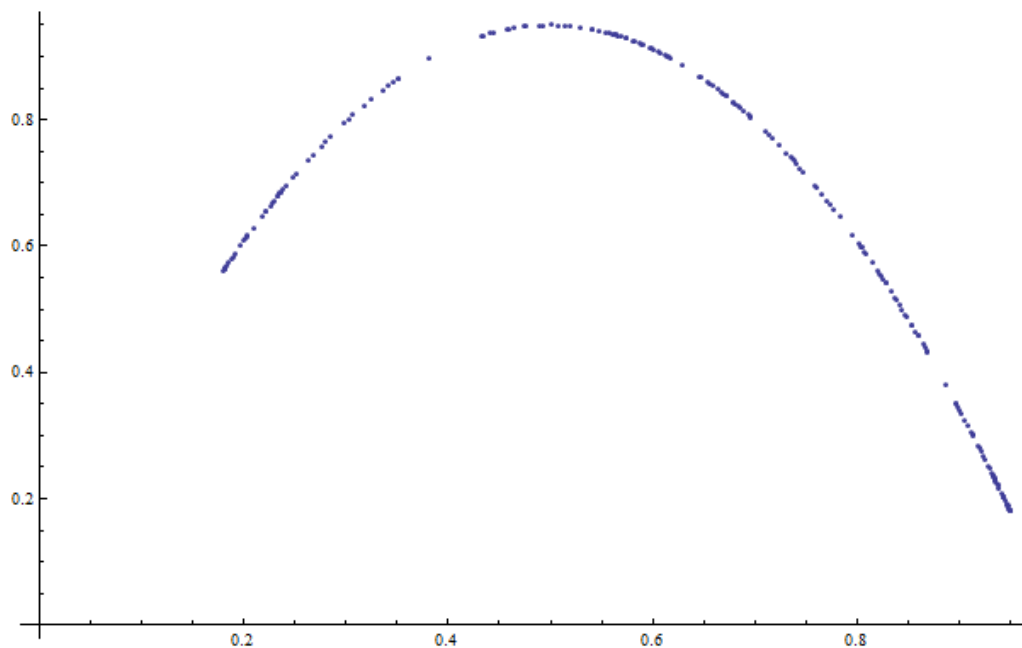
```
lq|= ListLinePlot[LogisticData, PlotRange -> All, Frame -> True, FrameLabel -> {"Iteration", "x"}, ImageSize -> 600]
```



Obr. 15 graf průběhu funkce pro hodnotu x (autor: Jiří Barák)

Vytvoření mapy logistické rovnice, tedy grafu závislosti následující hodnoty na předchozí hodnotě x , bylo vytvořeno pomocí dvousloupcové tabulky sestavené z vygenerovaných hodnot, přičemž v jednom sloupci je původní hodnota a v druhém hodnota z ní vypočtená. Tabulka se vykreslila do grafu zobrazujícím tuto závislost.

```
LogisticData2 = Table[{LogisticData[[i]], LogisticData[[i + 1]]}, {i, 1, Length[LogisticData] - 1, 1};  
ListPlot[LogisticData2, AxesOrigin -> {0, 0}, ImageSize -> 600]
```



Obr. 16 mapa Logistické rovnice (autor: Jiří Barák)

Generování samotných náhodných čísel bylo vytvořeno ve dvou funkcích pro generování reálných a celých náhodných čísel v požadovaném intervalu. Funkce si vygenerují z rovnice data a ta potom převedou do požadovaného intervalu, popřípadě funkce RndInt ještě číslo převede na Integer. Tato funkce vytvoří vždy jen jedno číslo, proto se musí pro vygenerování počtu požadovaných čísel zavolat vícekrát. Při výpočtech používá proměnnou NP, která určuje maximální velikost generovaného čísla, a proměnnou pocet, která určuje počet generovaných čísel.

```
A = 3.8;
LogisticData = NestList[Logistic[#1, A] &, xStart, 200];

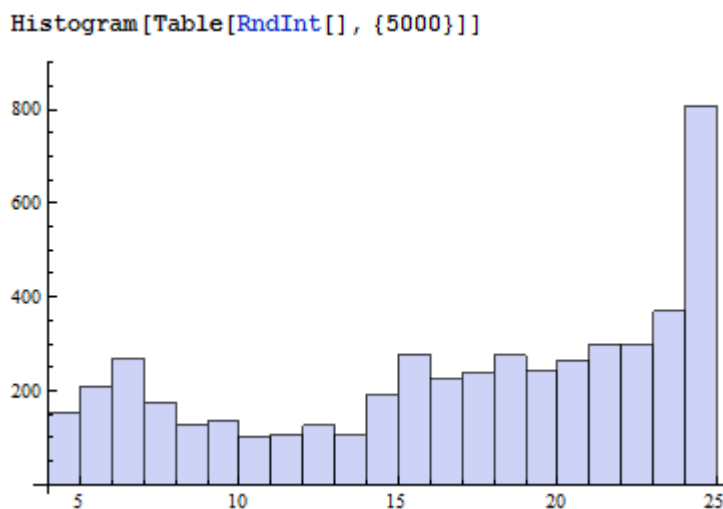
MaxVal = Max[Abs[LogisticData][[All]]];

RndInt[] := Module[{Value}, Index = Index + 1; If[Index > Dimensions[LogisticData][[1]],
  LogisticData = NestList[Logistic[#1, A] &, xStart, MaxRun]; MaxVal = Max[Abs[LogisticData][[All]]]; Index = 2;
  reevaluace = reevaluace + 1;
  Value = IntegerPart[Abs[LogisticData][[Index]] / MaxVal NP]; If[Value == 0, Value = Value + 1]; Return[Value]
]

RndReal[] := Module[{Value}, Index = Index + 1; If[Index > Dimensions[LogisticData][[1]],
  LogisticData = NestList[Logistic[#1, A] &, xStart, MaxRun]; MaxVal = Max[Abs[LogisticData][[All]]]; Index = 2;
  reevaluace = reevaluace + 1;
  Value = Abs[LogisticData][[Index]] / MaxVal NP; If[Value == 0, Value = Value + 1]; Return[Value]
]
```

Obr. 17 funkce pro generování náhodného čísla (autor: Jiří Barák)

Posledním výstupním blokem byl histogram, který se vygeneroval pomocí funkce histogram z tabulky, která byla vytvořena voláním fce RndInt nebo RndReal a parametru pocet.



Obr. 18 histogram vygenerovaných čísel z logistické rovnice (autor: Jiří Barák)

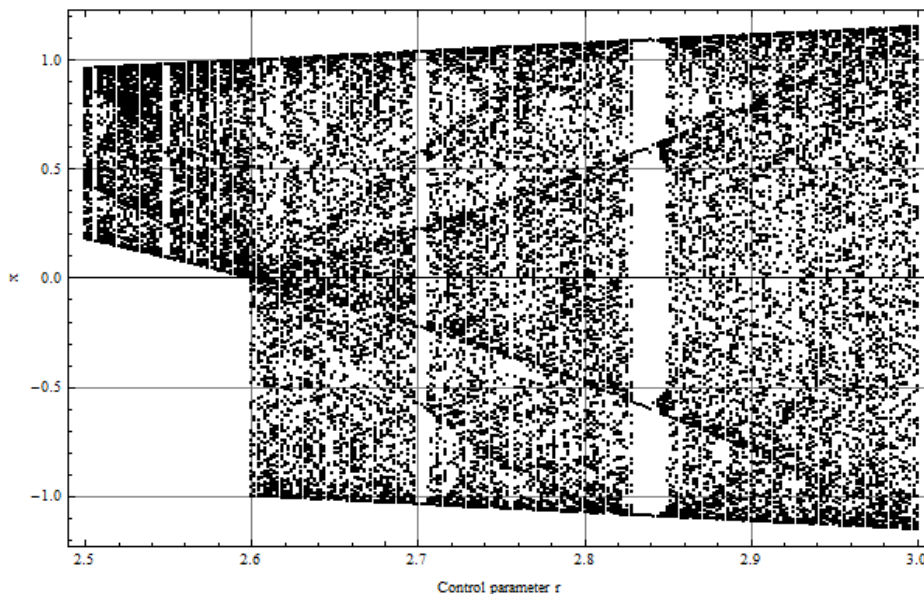
5.1.1.2 Cubic map

Tato mapa má tvar rovnice

$$x_{n+1} = Ax(1 - x_n^2).$$

Kód byl hodně podobný předchozí logistické rovnici. Vytvořil jsem novou funkci cubic zkompileváním rovnice. Použil jsem parametr A z intervalu $\langle 2.5, 3 \rangle$ s krokem 0.002. Pro další výpočty pak hodnotu $A=3$, $x_{start}=0.1$.

```
AFrom = 2.5;
ATo = 3;
AStep = .002;
CubicBifData = Table[Take[NestList[{A, Cubic[#1[[2]], A]} &, {0, xStart}, 300], -100], {A, AFrom, ATo, AStep}];
lq = ListPlot[Flatten[CubicBifData, 1], PlotRange -> All, PlotJoined -> False, Axes -> {True, False},
GridLines -> Automatic, Frame -> True, PlotStyle -> {RGBColor[0, 0, 0], PointSize[0.003]},
FrameLabel -> {"Control parameter r", "x"}, ImageSize -> 600]
```



Obr. 19 bifurkační diagram Cubic map (autor: Jiří Barák)

5.1.2 Disipativní systémy

Tyto systémy se liší od non-invertibilních tím, že mají počáteční podmínky dvě x a y , tedy se jedná o soustavu dvou rovnic. Navíc se obvykle liší i větším počtem parametrů. Dalším podstatným rozdílem je nahrazení bifurkačního diagramu atraktorem, grafem závislosti hodnoty y na x .

5.1.2.1 Burgers map

Tato mapa má tvar rovnic

$$\begin{aligned} X_{n+1} &= aX_n - Y_n^2 \\ Y_{n+1} &= bY_n - Y_n X_n \end{aligned}$$

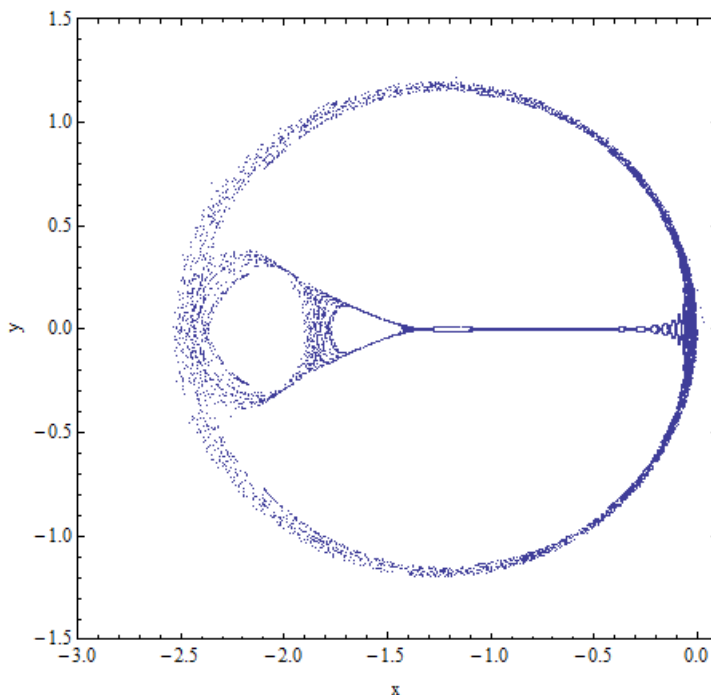
Byla vytvořena kompilovanou funkcí pojmenovaná Burgers.

```
Burgers = Compile[{{x, _Real}, {y, _Real}, {A, _Real}, {B, _Real}}, {A * x - y^2, B * y + x * y}]
CompiledFunction[{x, y, A, B}, {A x - y^2, B y + x y}, -CompiledCode-]
```

Obr. 20 kompilovaná funkce Burgers (autor: Jiří Barák)

Použité parametry při výpočtech měli hodnoty $A=0.75$, $B=1.75$. Celkově byl zdrojový kód velice podobný tvorbě logistické rovnice. Na výstup byl přidán blok průběhu funkce pro hodnotu Y , tento blok se bude nacházet pak u všech ostatních naprogramovaných disipativních systémů. Bifurkační diagram byl nahrazen atraktorem systému, který se vytvoří obdobně.

```
BurgersPlotData = NestList[Burgers[#1[[1]], #1[[2]], A, B] &, {xStart, yStart}, 20000];
ListPlot[BurgersPlotData, BaseStyle -> {FontWeight -> "Plain", FontColor -> RGBColor[0, 0, 0], FontSize -> 12},
Axes -> False, Joined -> False, GridLines -> None, PlotRange -> {{-3, 0.1}, {-1.5, 1.5}}, Frame -> True,
ImageSize -> 450, AspectRatio -> Automatic, PlotStyle -> PointSize[0.001], FrameLabel -> {"x", "y"}]
```



Obr. 21 atraktor Burgers map (autor: Jiří Barák)

5.1.2.2 Tinkerbell map

Tato mapa má tvar rovnic

$$\begin{aligned} X_{n+1} &= X_n^2 - Y_n^2 + aX_n + bY_n \\ Y_{n+1} &= 2X_n Y_n + cX_n + dY_n \end{aligned}$$

Kompilovaná funkce pojmenovaná Tinkerbell.

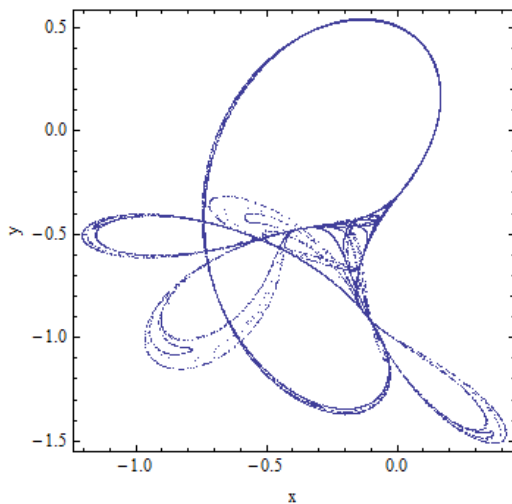
```
Tinkerbell = Compile[{{x, _Real}, {y, _Real}, {a, _Real}, {b, _Real}, {c, _Real}, {d, _Real}},
  {x^2 - y^2 + a*x + b*y, 2*x*y + c*x + d*y}]
CompiledFunction[{x, y, a, b, c, d}, {x^2 - y^2 + a*x + b*y, 2*x*y + c*x + d*y}, -CompiledCode-]
```

Obr. 22 kompilovaná funkce Tinkerbell (autor: Jiří Barák)

Použité parametry při výpočtech a=0.9, b=-0.6, c=2, d=0.5.

Vykreslený atraktor:

```
TinkerbellPlotData = NestList[Tinkerbell[#1[[1]], #1[[2]], a, b, c, d] &, {xStart, yStart}, 20000];
ListPlot[TinkerbellPlotData, BaseStyle -> {FontWeight -> "Plain", FontColor -> RGBColor[0, 0, 0], FontSize -> 12},
  Axes -> False, Joined -> False, GridLines -> None, PlotRange -> All, Frame -> True, ImageSize -> 350, AspectRatio -> 1,
  PlotStyle -> PointSize[0.001], FrameLabel -> {"x", "y"}]
```



Obr. 23 atraktor Tinkerbell map (autor: Jiří Barák)

5.1.2.3 Delayed map

Tato mapa má tvar rovnic

$$\begin{aligned} X_{n+1} &= AX_n(1 - Y_n) \\ Y_{n+1} &= X_n \end{aligned}$$

Použitý parametr při výpočtech $A=0.75$.

5.1.2.4 Henon map

Tato mapa má tvar rovnic

$$\begin{aligned} X_{n+1} &= 1 - aX_n^2 + bY_n \\ Y_{n+1} &= X_n \end{aligned}$$

Použité parametry při výpočtech $A=1.4, B=0.3$.

5.1.2.5 Lozi map

Tato mapa má tvar rovnic

$$\begin{aligned} X_{n+1} &= 1 - a|X_n| + bY_n \\ Y_{n+1} &= X_n \end{aligned}$$

Použité parametry při výpočtech $A=1.7, B=0.5$.

5.1.2.6 Dissipative standard map

Tato mapa má tvar rovnic

$$\begin{aligned} X_{n+1} &= X_n + Y_{n+1} \pmod{2\pi} \\ Y_{n+1} &= bY_n + k \sin X_n \pmod{2\pi} \end{aligned}$$

Použité parametry při výpočtech $B=0.6, k=8.8$.

5.1.2.7 Ikeda map

Tato mapa má tvar rovnic

$$\begin{aligned} X_{n+1} &= \gamma + \mu(X_n \cos \phi - Y_n \sin \phi) \\ Y_{n+1} &= \mu(X_n \sin \phi + Y_n \cos \phi) \end{aligned}$$

Použité parametry při výpočtech $\alpha = 6, \beta = 0.4, \gamma = 1, \mu = 0.9$.

5.1.2.8 Sinai map

Tato mapa má tvar rovnic

$$\begin{aligned} X_{n+1} &= X_n + Y_n + \delta \cos 2\pi Y_n \pmod{1} \\ Y_{n+1} &= X_n + 2Y_n \pmod{1} \end{aligned}$$

Použité parametry při výpočtech $\delta = 1$.

5.1.2.9 Arnold's cat map

Tato mapa má tvar rovnic

$$\begin{aligned} X_{n+1} &= X_n + Y_n \pmod{1} \\ Y_{n+1} &= X_n + kY_n \pmod{1} \end{aligned}$$

Použité parametry při výpočtech jsou $k=2$.

5.1.2.10 Chirikov map

Tato mapa má tvar rovnic

$$\begin{aligned} X_{n+1} &= X_n + Y_{n+1} \pmod{2\pi} \\ Y_{n+1} &= Y_n + k \sin X_n \pmod{2\pi} \end{aligned}$$

Použité parametry při výpočtech jsou $k=1$.

5.1.3 Oscilátory

Jsou zadány diferenciálními rovnicemi. Tyto rovnice se vyřeší pomocí funkce NDSolve, která vygeneruje data. Dále je postup zpracování dat stejný jako u logistické rovnice.

5.1.3.1 Damped driven pendulum oscillator

Tato mapa má tvar rovnic

$$\begin{aligned} dx / dt &= y \\ dy / dt &= -\sin x - by + A \sin \Omega t \end{aligned}$$

```

krok = 0.5;
XStart = 0;
YStart = 0;

Attractor = {
x'[t] == y[t],
y'[t] == -Sin[x[t]] - b y[t] + A Sin[Ω t],
x[0] == XStart, y[0] == YStart}

{x'[t] = y[t], y'[t] = 0.75 Sin[t Ω] - Sin[x[t]] - b y[t], x[0] = 0, y[0] = 0}

NLA = NDSolve[
Attractor /. {b → 0.05, A → 0.6, Ω → 0.7}, {x, y}, {t, 0, MaxRun}, MaxSteps → 10^6
];

```

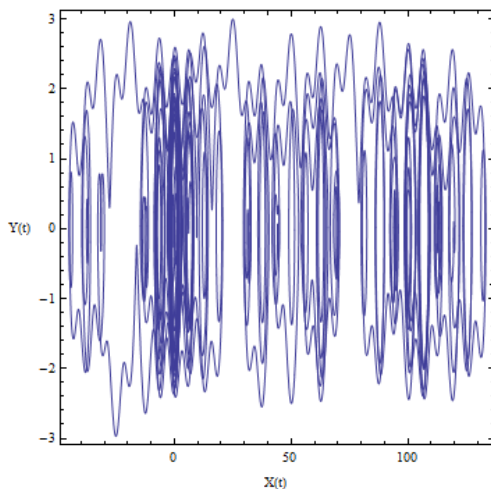
Obr. 24 dif. rovnice vyřešeny fci NDSolve (autor: Jiří Barák)

```

lq = ListLinePlot[ChaoticData[All], PlotRange → All, Frame → True, FrameLabel → {"Iterace", "x"},
ImageSize → 450, PlotLabel → Style["Graf průběhu hodnoty X", FontSize → 18]];

ParametricPlot[Evaluate[{x[t], y[t]} /. NLA], {t, 0, 1000}, PlotPoints → 10000, TextStyle → {FontWeight → "Plain", FontColor →
RGBColor[0, 0, 0],
FontSize → 10}, Axes → False,
Frame → True, ImageSize → 350, AspectRatio → 1, RotateLabel → False, FrameLabel → {"X(t)", "Y(t)"}, PlotRange → All, Frame → True]

```



Obr. 25 atraktor Damped driven pendulum oscilator (autor: Jiří Barák)

5.1.3.2 Ueda oscilator

Tato mapa má tvar rovnic

$$\begin{aligned} dx/dt &= y \\ dy/dt &= -x^3 - by + A \sin \Omega t \end{aligned}$$

Použité parametry při výpočtu $b=0.005$, $A=7.5$, $\Omega=1$.

5.1.3.3 *Duffing-van der Pool oscillator*

Tato mapa má tvar rovnic

$$\begin{aligned} dx / dt &= y \\ dy / dt &= \mu(1 - \gamma x^2)y - x^3 + A \sin \Omega t \end{aligned}$$

Použité parametry při výpočtu $\mu=0.2$, $\gamma=8$, $A=0.35$, $\Omega=1.02$.

5.1.3.4 *Hadley circulation*

Tato mapa má tvar rovnic

$$\begin{aligned} dx / dt &= -y^2 - z^2 - ax + aF \\ dy / dt &= xy - bxz - y + G \\ dz / dt &= bxy + xz - z \end{aligned}$$

Použité parametry při výpočtu $a=0.25$, $b=4$, $F=8$, $G=1$.

5.1.3.5 *Chua's circuit*

Tato mapa má tvar rovnic

$$\begin{aligned} dx / dt &= \alpha[y - x + bx + \frac{1}{2}(a - b)(|x + 1| - |x - 1|)] \\ dy / dt &= x - y + z \\ dz / dt &= -\beta y \end{aligned}$$

Použité parametry při výpočtu $\alpha=9$, $\beta=100/7$, $a=8/7$, $b=5/7$.

5.1.3.6 *Henon-Heiles system*

Tato mapa má tvar rovnic

$$\begin{aligned} dx / dt &= v \\ dy / dt &= w \\ dv / dt &= -x - 2xy \\ dw / dt &= -y - x^2 + y^2 \end{aligned}$$

Použité parametry při výpočtu $v=0$, $w=0.316076$.

6 VYTVOŘENÍ WEBOVÉ PREZENTACE V PROSTŘEDÍ WEBMATHEMATICA

6.1 Převod do prostředí webMahtematica

Server pracuje se soubory s příponou jsp. Byly tedy vytvořeny HTML stránky pro každý systém jedna, do kterého byly mezi tagy `<msp:evaluate>` `</msp:evaluate>` vloženy jednotlivé naprogramované bloky generátorů.

Prvním krokem při tvorbě jsp souboru bylo vytvoření formuláře pro zadávání počátečních podmínek, rozsahu čísel, jejich typu a počtu. Tento formulář předává proměnné metodou post, tedy mimo URL adresu.

```

        <form action="index.jsp" method="post">
Zadejte počet generovaných čísel (1-5000):
<br>
<input type="text" name="varpocet" value="1000" size="4" />
<br><br>

Zadejte maximální hodnotu generovaného čísla:<br> (max 6 cifer)
<br>
<input type="text" name="varNP" value="1024" size="6" />
<br><br>

Zadejte počáteční podmínky  $x_{start}$ :
<br>
<input type="text" name="varXstart" value="0.1" size="15" />
<br><br>

</div>
Zvolte typ generovaných čísel:
<br>
Real:
<input type="radio" name="varTyp" value="1" checked="checked">
<br>
Integer:
<input type="radio" name="varTyp" value="0">
<br><br>

<br><br>
<p align="center"><input type="image" src="tlacitko.png" value="Odeslat"/></p>

```

Obr. 26 ukázka formuláře na zadávání vstupů (autor: Jiří Barák)

Dále bylo třeba udělat přetypování proměnných pro webmathematicu, o to se stará tento blok kódu.

```

<msp:evaluate>

pocet=If[ MSPValueQ[ $\$\$$ varpocet],MSPToExpression[ $\$\$$ varpocet]];
NP=If[ MSPValueQ[ $\$\$$ varNP],MSPToExpression[ $\$\$$ varNP]];
typ=If[ MSPValueQ[ $\$\$$ varTyp],MSPToExpression[ $\$\$$ varTyp]];
xstart=If[ MSPValueQ[ $\$\$$ varXstart],MSPToExpression[ $\$\$$ varXstart]];
</msp:evaluate>

```

Obr. 27 přetypování proměnných pro webmathematicu (autor: Jiří Barák)

Teď již bylo možné vkládat jednotlivé výstupní bloky generátoru do zdrojového kódu stránky, který byl vytvořen pomocí tabulek a buňek a nastylován přímo v kódu.

```

<td width="510" valign="top">
  <div align="center">
    <center>
      <table border="0" cellpadding="2" width="100%" cellspacing="1">
        <tr>
          <td bgcolor="#2696B8" height="40">
            <p align="center"><b><font face="Verdana" size="2" color="FFFFFF">Logistic map</font></b></p>
            <p align="center"><b><font face="Verdana" size="1" color="FFFFFF"> $x_{n+1} = \lambda x_n(1-x_n)$ </font></b></p>
          </td>
        </tr>
        <tr>
          <td height="11">
            <div align="center">
              <table border="0" cellpadding="5" width="100%">
                <tr>
                  <td><b><font face="Verdana" size="2">Bifurkační diagram</font></b></td>
                </tr>
                <tr>
                  <td colspan="2"><p align="left">
                    <!--BIFURKACNI DIAGRAM-->
                    <div>
                      <msp:evaluate>

                        If[MSPValueQ[ $\$\$$ varXstart],
                          Logistic = Compile[{{x, _Real}, {A, _Real}}, A x (1 - x)];
                          LogisticBifData = Table[Take[NestList[{A, Logistic[#1[[2]], A]} &, {0, xstart}, pocet], -100], {A,2..4},
                          Axes -> {True, False}, GridLines -> Automatic, Frame -> True,
                          PlotStyle -> PointSize[0.001],
                          FrameLabel -> {"Řidící parametr  $\lambda$ ", "x"}, ImageSize -> 450,PlotLabel -> Style["Bifurkační diagram",
                          MSPShow[graflq]
                        ]
                      </msp:evaluate>
                    </div>
                  </td>
                </tr>
              </table>
            </div>
          </td>
        </tr>
      </table>
    </center>
  </div>

```

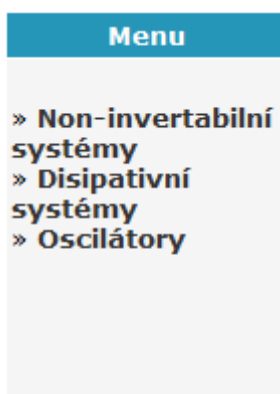
Obr. 28 blok pro vykreslení bifurkačního diagramu (autor: Jiří Barák)

6.2 Náhled na webovou prezentaci

Webové stránky jsou přístupné na adrese :

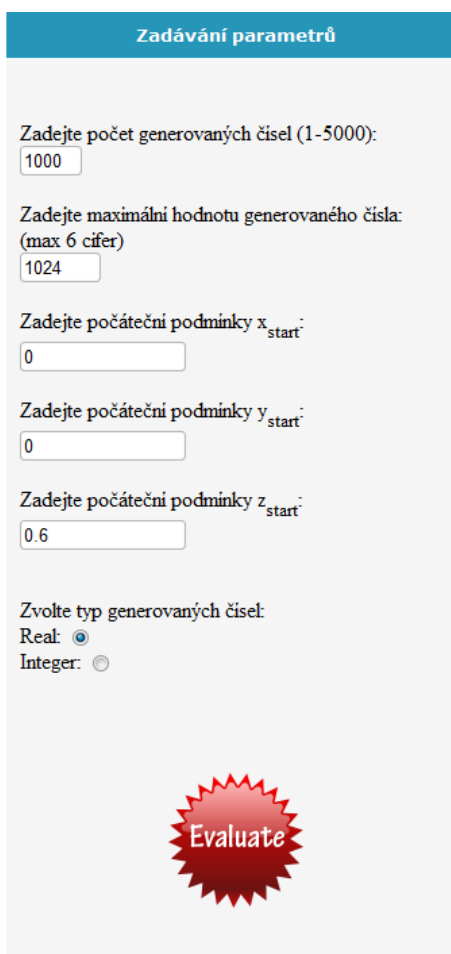
<http://mathematica.fai.utb.cz:8080/webMathematica/Barak/index.jsp>

Na těchto stránkách si uživatel v levé horní části zvolí skupinu systémů jež chce použít, tedy noninvertibilní, disipativní nebo oscilátory.



Obr. 29 volba skupiny dynamických systémů (autor: Jiří Barák)

Poté si uživatel v níže položeném menu zvolí konkrétní dynamický systém, jenž chce použít. Dále na pravé straně zadá počet vygenerovaných čísel, maximální hodnotu generovaného čísla, počáteční podmínky a zvolí typ generovaných čísel. Poté spustí generátor kliknutím na tlačítko evaluate.



Zadávání parametrů

Zadejte počet generovaných čísel (1-5000):

Zadejte maximální hodnotu generovaného čísla:
(max 6 cifer)

Zadejte počáteční podmínky x_{start} :

Zadejte počáteční podmínky y_{start} :

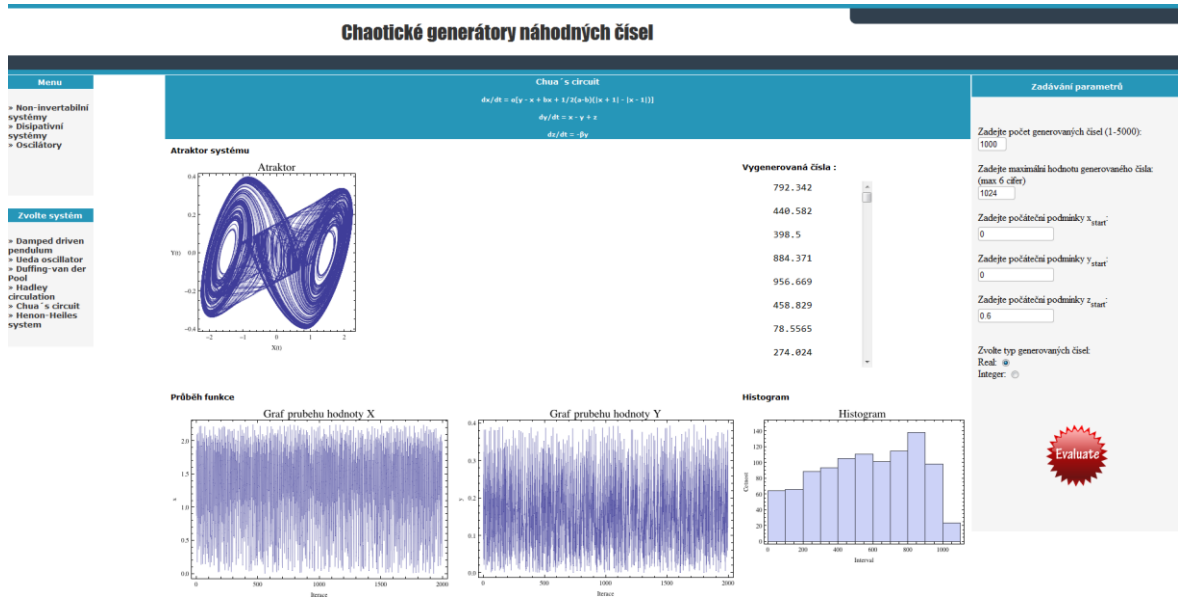
Zadejte počáteční podmínky z_{start} :

Zvolte typ generovaných čísel:
Real:
Integer:

Evaluate

Obr. 30 formulář pro zadávání parametrů (autor: Jiří Barák)

Po kliknutí se po chvíli vykreslí vypočtené údaje. V levém horním rohu se objeví bifurkační diagram nebo atraktor systému, napravo od něj vygenerovaná čísla. Dole pak graf průběhu funkce a pod vygenerovanými čísly histogram četnosti těchto čísel.



Obr. 31 celkový náhled na webovou prezentaci (autor: Jiří Barák)

7 MOŽNOSTI VYUŽITÍ GENERÁTORU NÁHODNÝCH ČÍSEL

Náhodné číslo je číslo vygenerované procesem, který má nepředpověditelný výsledek a jeho průběh nelze přesně reprodukovat. Což je u našeho generátoru dosaženo velkou citlivostí na počáteční podmínky, tudíž se dá říci, že vytvořený generátor je generátor pravých náhodných čísel.

Oblastí, kde by bylo možné uplatnit generátor náhodných čísel, je celá řada, simulace a modelování, náhodné losování, hazardní hry. V oblasti kryptologie je možnost použití například při generování kryptografických klíčů, generování čísel výplní pro ztížení dešifrování zprávy pomocí kryptoanalýzy. Vyžadovaná kvalita náhodnosti se liší od způsobu použití. Jednou z možností použití je použití generátoru u Vernamovy šifry.

Vernamova šifra nebo také one-time pad je jednoduchý šifrovací postup patentovaný v roce 1917 Gilbertem Vernamem. Spočívá v posunu každého znaku zprávy o náhodně zvolený počet míst v abecedě. To se prakticky rovná náhradě zcela náhodným písmenem a na tomto faktu je založen důkaz, že Vernamova šifra je v principu nerozluštitelná. Vezmeme jednotlivá písmena tajné zprávy a každé z nich posuneme o několik pozic v abecedě. Například první písmeno je posunuto o 5 pozic, druhé o 1, třetí o 14, čtvrté o 24, další o 9, 0, 3, 9, 19. Když při posouvání překročíme konec abecedy, pokračujeme od jejího začátku. Ke zpětnému rozluštění je třeba znát sadu čísel neboli posunů, bez znalosti je rozluštění zprávy nemožné. Náš generátor je vhodný k tomuto šifrování, díky možnosti nastavení počtu vygenerovaných čísel, protože tento počet čísel musí odpovídat počtu znaků ve zprávě. Spojením generátoru a Vernamovy šifry se dostáváme k „nerozluštitelné“ šifře, pokud ovšem neznáme zcela přesně počáteční podmínky generátoru.

ZÁVĚR

Hlavním cílem této bakalářské práce bylo popsat deterministický chaos v dynamických systémech. Dále pak získané poznatky aplikovat v prostředí Mathematica a WebMathematica při generování náhodných čísel.

První část této bakalářské práce objasňuje základní pojmy z teorie chaosu, obsahuje také malinké nakouknutí do historie této teorie. Dále se věnuje podmínkám vzniku chaotického chování, vizualizaci detrmnistického chaosu a jeho stabilitě.

Začátek samotné tvorby generátoru náhodných čísel probíhal v prostředí Mathematica. Bylo zde potřeba vytvořit jednotlivé bloky kódu, které budou zpracovávat uživatelem zadané počáteční podmínky, vygenerují náhodná čísla a vykreslí pro zvolený systém výstupní grafy. Tyto bloky se starají o vykreslení bifurkačního diagramu pro non-invertibilní systémy, atraktoru systému pro disipativní systémy a oscilátory. Dále pak vykreslují grafy průběhu funkcí jak pro hodnotu x tak y . U non-invertibilních map byly, kvůli absenci hodnoty y , vytvořeny bloky kódu vykreslující mapu systému, nebo-li graf závislosti X_{n+1} na X_n . Dalším vykreslovaným blokem byl histogram, který zobrazuje četnost rozdělení vygenerovaných čísel v intervalech. Samotné vygenerování probíhalo ve dvou funkcích pro celá a reálná čísla. Tyto bloky bylo třeba vytvořit pro všechny použité dynamické systémy zvlášť.

Kód byl poté převeden do webových souborů s příponou jsp pomocí softwaru webMathematica. Převedení spočívalo ve vytvoření html formuláře, který se postará o předání uživatelem zadaných hodnot. Pro funkci generátoru byly nutné zadat počáteční podmínky pro generování čísel, typ čísel, počet vygenerovaných čísel a maximální hodnotu generovaného čísla.

Formulář a jednotlivé bloky kódu byly pak vsazeny do vytvořené html stránky, která byla umístěna na server UTB.

Cíle práce byly splněny. Byla sepsána rešerše týkající se daného tématu a byl vytvořen funkční generátor náhodných čísel převedený do webových stránek.

ZÁVĚR V ANGLIČTINĚ

The aim of this thesis was to describe the deterministic chaos in dynamic systems. Then apply the knowledge in an environment of Mathematica and webMathematica to generate random numbers.

The first part of this work explains the basic concepts of chaos theory, and also contains a tiny peek into the history of this theory. It also discusses the conditions of chaotic behavior, visualization deterministic chaos and his stability.

creation of a random number generator ran in an environment of Mathematica. There was a need to create individual blocks of code that will process the user-specified initial conditions, generate random numbers and draws for the selected system output graphs. These blocks are responsible for rendering bifurcation diagram for non-invertible systems, the system attractor for dissipative systems and oscillators. Furthermore, the graphs depict the function of both the value of x and y . For non-invertible maps were due to the absence of y -values, create blocks of code that illustrate the map system, or if the graph of X_{n+1} , X_n . Another block visualized histogram that shows the frequency distribution of numbers generated at intervals. The actual generation went out in two functions for the integers and real numbers. These blocks need to be established for all the dynamical systems separately.

The code was then converted into web files with jsp using webMathematica software. The transfer was to create html form that will handle the transfer of user-entered values. For the function generator was necessary to specify initial conditions for generating numbers, type numbers, the number of generated numbers and the maximum value of generated numbers.

The form and the individual blocks of code were then put in the generated html page that has been placed on the server UTB.

Work objectives have been met. Research has been written on the topic and was created function a random number generator and then was converted into web pages.

SEZNAM POUŽITÉ LITERATURY

- [1] HEJTMÁNEK, Martin. Teorie chaosu. Martin Hejtmánek -- Teorie chaosu [online].[cit.2012-05-15].Dostupné:
<http://kmlinux.fjfi.cvut.cz/~hejtmmar/chaos.php>.
- [2] KRATOCHVÍL, Ctirad a Pavel HERIBAN. Dynamické systémy a chaos. Vyd. 2., rozš. Brno: Vysoké učení technické v Brně, 2010, 229 s. ISBN 978-80-214-4152-1.
- [3] HORÁK, Jiří. *Deterministický chaos a jeho fyzikální aplikace*. Vyd. 1. Praha: Academia, 2003, 437 s., viii s. barev. obr. příl. 1539. ISBN 80-200-0910-8.
- [4] HORÁK, Jiří, Ladislav KRLÍN a Aleš RAIDL. Deterministický chaos a podivná kinetika. Vyd. 1. Praha: Academia, 2007, 164 s., 16 s. obr. příl. ISBN 978-80-200-1531-0.
- [5] KRATOCHVÍL, Ctirad. Bifurkace a chaos v technických soustavách a jejich modelování. Vyd. 1. Brno: ÚT AVČR v Praze, pobočka Brno, 2008, 108 s. ISBN 978-80-87012-20-8.
- [6] MIKLE, Ondrej. Slabý generátor náhodných čísel umožňuje faktorizovat RSA moduly. [online]. [cit. 2012-06-01]. Dostupné z:
<http://blog.nic.cz/2012/02/16/slaby-rng-faktorizacia-rsa/>
- [7] Generating random numbers. Www.randomnumbers.info [online]. 2004 [cit. 2012-06-04]. Dostupné z:
<http://www.randomnumbers.info/content/Generating.htm>
- [8] CURT NOLL, Landon, Simon COOPER a Mel PLEASANT. LavaRnd [online]. [cit. 2012-06-05]. Dostupné z: <http://www.lavarnd.org/>
- [9] YOUNG, Peter. Randu: a bad random number generator. s. 3. Dostupné z:
<http://physics.ucsc.edu/~peter/115/randu.pdf>
- [10] WOLFRAM RESEARCH. Wolfram Knowledge Base [online]. [cit. 2012-06-06]. Dostupné z: <http://support.wolfram.com/kb/>

- [11] BRANICKÝ, Marek. Java Servlets - predstavenie technológie. In: [online]. 2003 [cit. 2012-06-06]. Dostupné z: <http://interval.cz/clanky/java-servlets-predstavenie-technologie/>
- [12] BRANICKÝ, Marek. JavaServer Pages pro všechny. In: [online]. 2002 [cit. 2012-06-06]. Dostupné z: <http://interval.cz/clanky/jaserver-pages-pro-vsechny/>
- [13] ŠENKEŘÍK, Roman. Optimal control of deterministic chaos: Optimální řízení pomocí deterministického chaosu : doctoral thesis summary. Zlín: Tomas Bata University in Zlín, 2008. 30 s. ISBN 978-80-7318-781-1.
- [14] HOSTE, Jim. Mathematica demystified. New York: McGraw-Hill, c2009, xv, 386 p. McGraw-Hill "Demystified" series. ISBN 00-715-9144-3.
- [15] MAREK, Jan. Kryptografie.eu || Vše o kryptologii [online]. 2011 [cit. 2012-06-06]. Dostupné z: <http://kryptografie.eu/cl-8.htm>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PRN	Pseudo Random Number
TRN	True Random Number
GUI	Graphical user interface
JSP	Java Server Pages
J2EE	Java 2 Enterprise Edition
SUN	Sun Microsystems
JVM	Java Virtual Machine
HTML	HyperText Markup Language
PHP	Hypertext Preprocessor
URL	Uniform Resource Locator

SEZNAM OBRÁZKŮ

Obr. 1 Edward Lorenz (zdroj: internet)	12
Obr. 2 Limitní body (zdroj: internet)	14
Obr. 3 Limitní cyklus (zdroj: internet).....	15
Obr. 4 Chaotický atraktor (zdroj: internet)	15
Obr. 5 Lorenzův „podivný“ atraktor (zdroj: internet).....	16
Obr. 6 Ukázka fraktálů – Mandelbrotova množina (zdroj: internet)	17
Obr. 7 Juliovy množiny (autor: Petra Holbíková).....	17
Obr. 8 Bifurkační diagram populační rovnice (zdroj: internet, úprava: Jiří Barák).....	19
Obr. 9 RANDU se špatně zvolenými parametry (zdroj: internet)	23
Obr. 10 RANDU s opraveným parametrem $m = 2^{39}$ (zdroj: Jiří Barák).....	24
Obr. 11 vývojové prostředí Mathematica (zdroj: internet)	27
Obr. 12 Princip komunikace WebMathematicy (zdroj: Wolfram research).....	28
Obr. 13 kompilace logistické rovnice (autor: Jiří Barák).....	30
Obr. 14 bifurkační diagram Logistické rovnice (autor: Jiří Barák)	30
Obr. 15 graf průběhu funkce pro hodnotu x (autor: Jiří Barák).....	31
Obr. 16 mapa Logistické rovnice (autor: Jiří Barák)	31
Obr. 17 funkce pro generování náhodného čísla (autor: Jiří Barák).....	32
Obr. 18 histogram vygenerovaných čísel z logistické rovnice (autor: Jiří Barák)	32
Obr. 19 bifurkační diagram Cubic map (autor: Jiří Barák).....	33
Obr. 20 kompilovaná funkce Burgers (autor: Jiří Barák)	34
Obr. 21 atraktor Burgers map (autor: Jiří Barák).....	34
Obr. 22 kompilovaná funkce Tinkerbell (autor: Jiří Barák)	35
Obr. 23 atraktor Tinkerbell map (autor: Jiří Barák).....	35
Obr. 24 dif. rovnice vyřešeny fcí NDSolve (autor: Jiří Barák).....	38
Obr. 25 atraktor Damped driven pendulum oscilator (autor: Jiří Barák).....	38
Obr. 26 ukázka formuláře na zadávání vstupů (autor: Jiří Barák).....	40
Obr. 27 přetytování proměnných pro webmathematicu (autor: Jiří Barák)	41
Obr. 28 blok pro vykreslení bifurkačního diagramu (autor: Jiří Barák)	41
Obr. 29 volba skupiny dynamických systémů (autor: Jiří Barák).....	42
Obr. 30 formulář pro zadávání parametrů (autor: Jiří Barák).....	42
Obr. 31 celkový náhled na webovou prezentaci (autor: Jiří Barák).....	43

SEZNAM PŘÍLOH

P I: CD se zdrojovými kódy