

# **Synchronizace a komunikace pomocí deterministického chaosu v prostředí Mathematica**

Bc. Petra Adamcová

---

Diplomová práce  
2006



**Univerzita Tomáše Bati ve Zlíně**  
Fakulta aplikované informatiky

---

**Univerzita Tomáše Bati ve Zlíně**

**Fakulta aplikované informatiky**

**Ústav aplikované informatiky**

**akademický rok: 2005/2006**

## **ZADÁNÍ DIPLOMOVÉ PRÁCE**

**(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)**

**Jméno a příjmení: Bc. Petra ADAMCOVÁ**

**Studijní program: N 3902 Inženýrská informatika**

**Studijní obor: Informační technologie**

**Téma práce: Synchronizace a komunikace pomocí  
deterministického chaosu v prostředí Mathematica**

**Zásady pro vypracování:**

**Práce bude zaměřena na principy a využití přenosu informace pomocí deterministického chaosu. Náplní práce bude vytvoření programového balíku demonstrujícího přenos signálu pomocí chaotických signálů co by nosiče. :**

- 1. vypracovat přehled problematiky deterministického chaosu**
- 2. vybrat vhodné již řešené příklady**
- 3. vypracovat alternativní řešení pomocí sw Mathematica**
- 4. provést závěr**

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

Arrowsmith D.K., Place C.M., An Introduction to Dynamical Systems, Cambridge University Press, UK,

ISBN 0-521-316-50, 1990

Gleick J.1996, Chaos, Ando publishing, ISBN 80-86047-04-0, 1996

Hilborn R.C.1994, Chaos and Nonlinear Dynamics, Oxford University Press, ISBN 0-19-508816-8, 1994

Horák J., Krlín L.1996, Deterministický chaos, Academia, ISBN 80-200-0416-5, 1996

Marek M., Schreiber I., Chaotic Behaviour of Deterministic Dissipative Systems, Academia, ISBN 80-200-0186-7

Nicolis G.1982, Introduction to Nonlinear Science, Cambridge University Press, ISBN 0-521-46782-9, 1982

Palis J., Takens F.1993, Hyberbolicity & sensitive chaotic dynamics at homoclinic bifurcations, Cambridge University Press, ISBN 0-521-47572-4, 1993

Peitgen H.O., Jurgens H., Saupe D. 1992, Chaos and Fractals, New Frontiers of Science, Springer-Verlag 1992, ISBN 3-540-97903-4

Vedoucí diplomové práce:

**doc. Ing. Ivan Zelinka, Ph.D.**

Ústav aplikované informatiky

Datum zadání diplomové práce:

**14. února 2006**

Termín odevzdání diplomové práce:

**26. května 2006**

Ve Zlíně dne 14. února 2006

prof. Ing. Vladimír Vašek, CSc.

*pověřený děkan*



doc. Ing. Ivan Zelinka, Ph.D.

*ředitel ústavu*

## **ABSTRAKT**

Diplomová práce se zabývá tematikou synchronizace a řízení deterministického chaosu. Seznámíme se s historií a se základní problematikou této nové vědy chaos a ukážeme si synchronizaci dvou chaotických systémů. Pro synchronizaci použijeme již řešené příklady a alternativně je vypracujeme pomocí software Mathematica. Vzhledem k obtížnosti a šíři dané problematiky jsou v diplomové práci jmenovány základní informace a principy pro synchronizaci a řízení chaosu.

Klíčová slova: deterministický chaos, synchronizace, řízení, atraktor

## **ABSTRACT**

The master thesis is focused on synchronization and chaos control. We introduce history and basic issue of this new science. We show synchronization of two chaos systems. We use already exist example for synchronization. And this example we programme with help software Mathematica. Because theme is difficulty and complexity the master thesis is focused on basic information and principle of synchronization and chaos control.

Keywords: deterministic chaos, synchronization, control, attractor

Děkuji vedoucímu diplomové práce panu Doc. Ing. Ivanu Zelinkovi, Ph.D. za odborné vedení, cenné rady a připomínky, které mi poskytoval při řešení diplomové práce.

# OBSAH

<b>ÚVOD</b> .....	<b>10</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>11</b>
<b>1 HISTORIE CHAOSU</b> .....	<b>12</b>
1.1 PŮVOD SLOVA CHAOS.....	12
1.2 HISTORIE NOVÉ VĚDY CHAOS .....	13
1.2.1 Jules Henri Poincare (29. 4. 1854 – 17. 7. 1912).....	13
1.2.2 Edward Norton Lorenz (23. 5. 1917).....	14
1.2.3 Alexandr Michailovič Ljapunov (6. 6. 1857 – 3. 11. 1918) .....	17
1.2.4 Georg Ferdinand Ludwig Philipp Cantor (3. 3. 1845 – 6. 1. 1918).....	17
1.2.5 Oscar Rösler .....	19
1.2.6 Balthasar Van der Pol (1889 - 1959).....	20
1.2.7 René Thom (2. 9. 1923 – 25. 10. 2002) .....	21
1.2.8 Michell Jay Feigenbaum (19. 12. 1944).....	23
<b>2 ÚVOD DO TEORIE CHAOSU</b> .....	<b>25</b>
2.1 TEORIE CHAOSU .....	25
2.2 CHAOS.....	26
2.3 VÝSKYT DETERMINISTICKÉHO CHAOSU.....	27
2.4 CHAOTICKÝ POHYB .....	27
2.4.1 Citlivost na počáteční podmínky.....	27
2.4.2 Topologická tranzitivita .....	28
2.5 SYSTÉMY GENERUJÍCÍ DETERMINISTICKÝ CHAOS .....	28
2.5.1 Hamiltoniánské systémy .....	28
2.5.2 Disipativní systémy .....	29
2.6 ČASOVÉ ŘADY .....	29
2.7 ATRAKTORY .....	30
2.7.1 Co to je atraktor.....	30
2.7.2 Druhy atraktorů .....	31
2.7.2.1 Množina pevných bodů.....	32
2.7.2.2 Periodické a kvaziperiodické body .....	32
2.7.2.3 Chaotický atraktor .....	32
2.7.2.4 Podivný atraktor.....	32
2.7.3 Lorenzův atraktor .....	33
2.8 BIFURKACE .....	34
2.9 ZNÁZORNĚNÍ CHAOTICKÉHO CHOVÁNÍ .....	35
2.9.1 Bifurkační diagram.....	35
2.9.2 WEB diagram.....	36
2.10 FEIGENBAUMOVA KONSTANTA.....	37
2.10.1 Feigenbaumova konstanta delta .....	37
2.10.2 Feigenbaumova konstanta alfa .....	38

2.11	TEORIE KATASTROF .....	38
2.11.1	Katastrofa typu záhyb .....	39
2.11.2	Katastrofa typu pyramida .....	40
2.12	LOGISTICKÁ ROVNICE .....	41
2.13	PEVNÝ BOD .....	42
<b>3</b>	<b>ŘÍZENÍ A SYNCHRONIZACE DETERMINISTICKÉHO CHAOSU.....</b>	<b>43</b>
3.1	ÚVOD .....	43
3.2	ČASOVÝ CHAOS .....	44
3.3	ČASOPROSTOROVÝ CHAOS .....	45
3.3.1	Systémy časoprostorového chaosu.....	45
3.3.2	Logistické rovnice v paralelním spojení .....	45
3.3.3	Přenos informace.....	46
3.4	SAMOORGANIZACE.....	47
3.4.1	Historie .....	47
3.4.2	Bélousov – Žabotinského reakce .....	47
3.4.3	Entropie .....	48
3.4.4	Teorie samoorganizace.....	49
<b>4</b>	<b>SOMA ALGORITMUS .....</b>	<b>50</b>
4.1	SAMO-ORGANIZUJÍCÍ SE MIGRAČNÍ ALGORITMUS .....	50
4.2	PARAMETRY .....	50
4.2.1	Mass .....	50
4.2.2	Step.....	51
4.2.3	PRT .....	51
4.2.4	DIM .....	52
4.2.5	POPSIZE .....	52
4.2.6	Migrace .....	52
4.2.7	MINDIV .....	52
4.2.8	Specimen .....	52
4.3	PRINCIP ALGORITMU SOMA .....	53
4.3.1	Definice parametrů.....	53
4.3.2	Tvorba populace.....	53
4.3.3	Migrační kola .....	53
4.3.4	Testování naplnění ukončovacích parametrů.....	54
4.3.5	Stop .....	54
<b>5</b>	<b>DIFERENCIÁLNÍ EVOLUCE.....</b>	<b>55</b>
5.1	PARAMETRY .....	55
5.1.1	CR .....	55
5.1.2	NP.....	55
5.1.3	F.....	55
5.1.4	Generations .....	55
5.1.5	D.....	56
5.2	PRINCIP DIFERENCIÁLNÍ EVOLUCE.....	56
5.2.1	Stanovení parametrů.....	56

5.2.2	Tvorba populace.....	56
5.2.3	Započetí cyklu.....	56
5.2.4	Evoluční cyklus.....	56
5.2.5	Testování naplnění ukončovacích parametrů.....	56
5.2.6	Vyhodnocení.....	57
<b>II</b>	<b>PRAKTICKÁ ČÁST.....</b>	<b>58</b>
<b>6</b>	<b>SOFTWARE MATHEMATICA.....</b>	<b>59</b>
6.1	HISTORIE.....	59
6.2	MATHEMATICA.....	59
<b>7</b>	<b>SYNCHRONIZACE LORENZOVÝCH ATRAKTORŮ.....</b>	<b>60</b>
7.1	SYNCHRONIZACE.....	60
7.1.1	Věta o synchronizaci.....	60
7.1.2	Způsob synchronizace.....	60
7.2	POPIS PROGRAMU.....	60
7.2.1	Proměnná LorenzAttractor.....	60
7.2.2	Proměnná NLA.....	61
7.2.3	ParametricPlot3D.....	61
7.3	ZHODNOCENÍ VÝSLEDKŮ.....	62
7.3.1	Systemy.....	62
7.3.2	System A synchronizovaný, systém B řídicí.....	63
7.3.3	System A řídicí, systém B synchronizovaný.....	65
<b>8</b>	<b>SYNCHRONIZACE LORENZOVA A RÖSLEROVA ATRAKTORU.....</b>	<b>67</b>
8.1	POPIS PROGRAMU.....	67
8.1.1	Proměnná LorenzAttractor.....	67
8.1.2	Proměnná NLA.....	67
8.1.3	ParametricPlot3D.....	68
8.1.4	Plot.....	68
8.2	POKUS 1.....	69
8.2.1	3D grafy.....	69
8.2.2	Grafy podle jedné proměnné systému.....	70
8.3	POKUS 2.....	72
8.3.1	3D grafy.....	72
8.3.2	Grafy podle jedné proměnné systému.....	74
8.3.3	Účelová funkce pro jednotlivé řezy.....	75
8.4	ZHODNOCENÍ VÝVOJE EVOLUCE.....	78
8.4.1	Pro pokus 1.....	78
8.4.2	Pro pokus 2.....	79
<b>9</b>	<b>PŘENOS INFORMACE POMOCÍ ČASOPROSTOROVÉHO CHAOSU.....</b>	<b>82</b>
	<b>ZÁVĚR.....</b>	<b>83</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>84</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>86</b>
	<b>SEZNAM OBRÁZKŮ.....</b>	<b>87</b>



<b>SEZNAM PŘÍLOH.....</b>	<b>89</b>
---------------------------	-----------

## ÚVOD

Co znamená slovo chaos? Na význam tohoto slova můžeme nahlížet z několika pohledů. Označuje stav neuspořádanosti, zmatku, temnoty, stav, kde vládne nepředvídatelnost, neuchopitelnost, stav, který je blízký nicotě a neexistenci řádu, stav, který je nahodilý a bez vnitřních zákonitostí. Pro většinu lidí chaos představuje stav neuspořádanosti a zmatku.

Slovo chaos pochází z řeckého jazyka a označuje nepředvídatelnost. V klasickém období starověkého Řecka neznamenovalo nepořádek, ale Řekové chápali toto slovo jako primární prázdnotu, kosmický prostor. Nepřesným chápáním v období raného křesťanství se postupně význam slova měnil.

V metafyzickém smyslu se jedná o opak zákona pořádku. Slovo chaos se také používá k popisu raných stádií vesmíru ve smyslu neohraničeného prostoru a beztvare hmoty.

Ve druhé polovině dvacátého století se začíná obsah tohoto pojmu značně měnit. V matematice chaos znamená neperiodické deterministické chování, které je velice citlivé na počáteční podmínky, a s ním už souvisí teorie chaosu a další pojem, jako je řízení deterministického chaosu.

## **I. TEORETICKÁ ČÁST**

## 1 HISTORIE CHAOSU

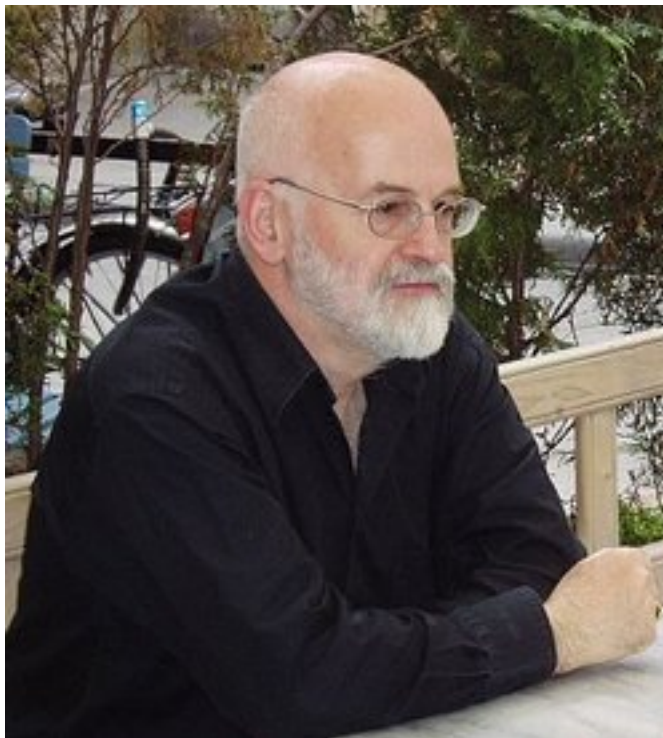
### 1.1 Původ slova chaos

Původ slova chaos nacházíme již v řecké mytologii. Hésiodos ve své básni Zrození bohů uvádí: „Zajisté, první vznikl Chaos, ale po něm Země ...“. Tím začíná řetěz zplození a zrození. Nejprve spontánně vznikl Chaos, po něm temná noc Nyx a věčná tma Erebos, z Ereba a Nykty vzniklo věčné světlo Aithér a světlý den Hémerá. Poté se zrodila země Gaia, podsvětní propast Tartaros a všeoživující láska Erós. Gaia zrodila nebe Úrana a moře Ponta. S Úranem měla 12 synů a dcer, obrovských Titánů. Úranos se zmocnil vlády nad světem, později ji musel přenechat svému synu Kronovi a ten zase svému synu Diovi. Hésiodos chápal chaos jako prázdný prostor, v němž se může rozšiřovat později vzniklé.

Ovidius ve své knize Proměny začíná rovněž stvořením světa, chápe Chaos jako „mohutnou masu, beztvárovou a pustou, nic než spočívající tíži, hromadu neslučitelných jader“.

Na chaos z řecké mytologie mnohem později volně navazují výskyty Chaosu ve fantasy literatuře. Americký autor fantasy a science fiction Roger Zelazny (13. května 1937 – 14. června 1995) napsal sérii knih Tajuplný Amber. V ní existují jen dvě skutečná místa (světy), a to Amber a Dvory Chaosu. Všechny ostatní, včetně naší Země, jsou jen stíny vytvářené napětím mezi nimi, mezi Řádem a Chaosem. Obyvatelé (potomci královské rodiny) obou skutečných světů mohou mezi stíny cestovat pomocí magie aktivované speciálním obrazem. Ambéřané tomu svému říkají Vzor, chaosané Logrus.

V sérii Zeměplocha anglického spisovatele Terryho Pratchetta (28. dubna 1948) je Chaos pátým jezdcem apokalypsy (ostatní jsou Smrt, Válka, Nákaza a Hlad) a disponuje mečem chladnějším než absolutní nula a jezdí válečným vozem.



Obr. 1 Terry Pratchett

## 1.2 Historie nové vědy chaos

### 1.2.1 Jules Henri Poincare (29. 4. 1854 – 17. 7. 1912)

Byl francouzský matematik, fyzik, astronom, filozof, člen Francouzské akademie věd (1908) a 35 dalších akademií. Je považován za jednoho z posledních univerzálních vědců. Podílel se mimo jiné na vytvoření speciální teorie relativity, jako první odhalil chaotické chování dynamických systémů. Svými současníky byl považován za největšího génia konce 19. století. Některé jeho myšlenky natolik předběhly svou dobu, že je vědci pochopili až dlouho po jeho smrti.

Od dětství byl krátkozraký, roztržitý, velmi nešikovný. Fyzickou neobratnost mu příroda nadměru vynahradila matematickým talentem. Své první práce publikoval v odborných časopisech, když mu bylo 19 let. V pětadvaceti letech získal doktorát a stal se profesorem na univerzitě v Caen. Za dva roky už přednášel matematiku na věhlasné pařížské Sorboně, která se stala jeho vědeckým domovem na celý život.



Obr. 2 Jules Henri Poincare

Kořeny teorie chaosu lze datovat k roku 1990, ve studiích Henri Poincarého o problému pohybu 3 objektů se vzájemnou gravitační silou, tzv. problému tří těles. Poincaré objevil, že mohou existovat orbity, které jsou neperiodické a které nejsou ani neustále vzrůstající a ani se neblíží pevnému bodu.

Je zakladatelem kvalitativní dynamiky, což je matematická teorie dynamických systémů. Od příjmení tohoto slavného matematika je odvozen název Poincarého plocha, která slouží k vizualizaci chování mnohorozměrných dynamických systémů.

### **1.2.2 Edward Norton Lorenz (23. 5. 1917)**

Edward Norton Lorenz je americký matematik a meteorolog působící v oblasti teorie chaosu, vynálezce pojmu podivný atraktor a termínu motýlí efekt.

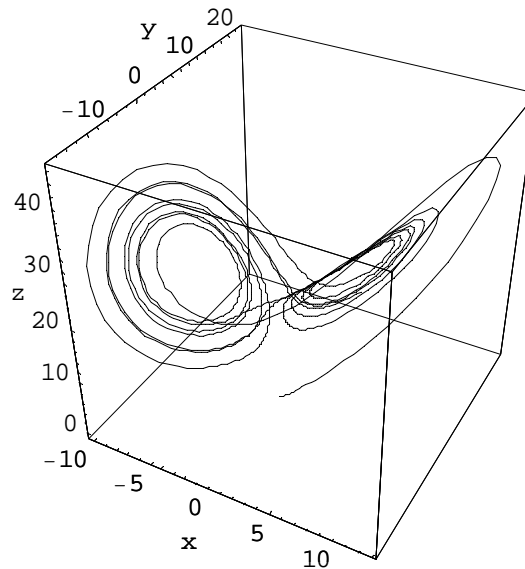
Edward N. Lorenz se narodil 23. května 1917 ve městě West Havenu ve státě Connecticut. Studoval matematiku na Dartmouth College v New Hampshiru a na Harvardské univerzitě v Cambridge, Massachusetts. Sloužil jako meteorolog pro letecké jednotky USA během 2. světové války. Po svém návratu z války studoval meteorologii na Massachusetts Institute of Technology. Na tomto institutu pak působil mnoho let jako profesor.



Obr. 3 Edward Norton Lorenz

Lorenz sestavil matematický model vynucené konvekce v atmosféře. Při studiu modelů počasí objevil, že počasí se ne vždy chová podle předpovědi. Malá odchylka v počátečních hodnotách proměnných v jeho primitivním počítačovém modelu počasí měla za následek velké rozdíly chování počasí. Tato citlivá závislost na počátečních podmínkách je známá jako motýlí efekt.

Lorenz prozkoumal tento model z matematického hlediska a publikoval své závěry v práci zvané Deterministické neperiodické proudění, ve které popsal relativně jednoduchý systém rovnic vedoucí k modelu nekonečné složitosti, Lorenzově atraktoru.



Obr. 4 Lorenzův atraktor

Jeho teorie chaosu popisuje “nedeterministické“ systémy, které jsou obecně popisovány a charakterizovány tzv. nelineárními diferenciálními rovnicemi, které používáme k modelování a k popisu.

Vše se stalo v roce 1961, kdy se Edward N. Lorenz snažil modelovat počasí. Lorenz použil počítač Royal McBee LPG-30 k výpočtu svého modelu simulující počasí. Stanovil soubor rovnic, pomocí nichž se iterací vypočítává působení atmosférických sil. Iteroval rovnice tak, že jejich řešení v každý okamžik bylo bráno jako počáteční podmínka do dalšího souboru výpočtů. Když byly tyto Lorenzovy rovnice naprogramovány do počítače, graficky vyjádřena řešení vypadal na jako meandr.

I když počítač zachovával přesnost výpočtu na šest desetinných míst, řešení rovnic byla tištěna s přesností jen na tři místa. V určitém okamžiku Lorenz spustil program znovu a jako počáteční podmínky použil jedno z předběžných řešení, objevujících se na předchozím výtisku. Nový tisk výsledků vypadal stejně jako starý, ale jen chvíli. Pak se výsledky začaly odchylovat od původní předpovědi a program brzy poskytl zcela novou předpověď budoucnosti. [18]



Značné rozdíly mezi oběma předpověďmi byly důsledkem rozdílu mezi čísly 0,506127 a 0,506. Rozdíl jen 0,000127 všechno změnil. Ukázalo se tedy, že zcela nepatrné změny počátečních podmínek vedou k dramaticky k odlišnému počasí a že i velice jednoduché systémy se mohou chovat chaoticky. Jinými slovy, malá změna počátečních podmínek vede k naprosto odlišnému výsledku.

### 1.2.3 Alexandr Michailovič Ljapunov (6. 6. 1857 – 3. 11. 1918)

Ruský matematik, jehož základní práce se týkaly diferenciálních rovnic, teorie potenciálu, stability řešení a teorie pravděpodobnosti. Na jeho počest je pojmenována Ljapunova stabilita: stabilita řešení diferenciálních rovnic vzhledem k perturbaci počáteční podmínky. Z fyzikálních problémů řešil podmínky stability rotující kapaliny.



Obr. 5 A. M. Ljapunov

Do nové vědy chaos přispěl svými tzv. Ljapunovy exponenty, které vyjadřují u chaotických systémů divergenci nebo-li rozbíhavost blízkých trajektorií. Exponenty jsou označovány znakem řecké abecedy  $\lambda$ .

### 1.2.4 Georg Ferdinand Ludwig Philipp Cantor (3. 3. 1845 – 6. 1. 1918)

Tento významný matematik se narodil 3 března roku 1845 v ruském Petrohradě. Otec byl Dán, matka Ruska. Základní školu navštěvoval v rodném Peterburgu, ale už v 11 letech se

s rodiči přestěhoval do Wiesbadenu v Německu, zde studoval na gymnáziu, potom na reálce v Darmstadtu, kterou v roce 1860 ukončil. Tady se projevil jeho matematické schopnosti. Pak začal studovat na polytechnice v Zürichu, ale přešel na univerzitu do Berlína. Tam se stal členem Matematické společnosti. Doktorát získal v roce 1867. Svou habilitační práci o teorii čísel dokončil v roce 1869 v Halle, kde byl o 3 roky později jmenován mimořádným profesorem. V letech 1879-1884 publikoval 6 článků, které se staly základem teorie množin. Ta výrazně ovlivnila další vývoj matematiky. Stejně průkopnické byly jeho názory na nekonečno. Jeho teorie spoluvytvořila základy reálných funkcí a topologie. V květnu roku 1884 podlehl Cantor těžké depresi, které se později opakovaly. V tomto období se odkláněl od matematiky k filozofii. Od roku 1899 začal v důsledku své nemoc vynechávat zimní semestry. Později strávil určitou dobu v sanatoriu pro duševně nemocné. V roce 1913 odešel do ústraní. O 4 roky později byl umístěn znovu do sanatoria. Tentokrát se už nevrátil.

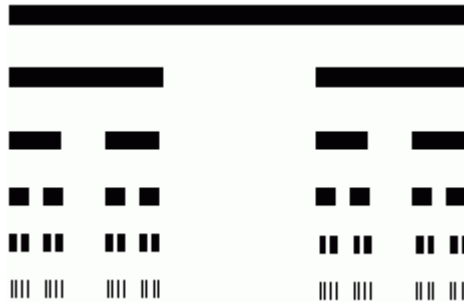


Obr. 6 Georg F. L. P. Cantor

K nové vědě chaos přispěl svým objevem Cantorovy množiny. Jelikož geometrická struktura chaotických atraktorů se nejednou projevuje jako Cantorova množina.

Sestrojíme ji z číselného intervalu  $\langle 0,1 \rangle$ . Když z intervalu  $\langle 0,1 \rangle$  znázorněného pomocí úsečky odstraníme její prostřední třetinu, tj. otevřený interval  $(1/3, 2/3)$ , zůstanou nám dva segmenty, a to úsečky, které jsou zobrazením sjednocení intervalů  $\langle 0, 1/3 \rangle \cup \langle 2/3, 1 \rangle$ .

Pokud dále opět odstraníme z těchto dvou úseček jejich prostřední třetiny, konkrétně intervaly  $(1/9, 2/9)$  a  $(7/9, 8/9)$ , zbudou nám čtyři segmenty, z kterých opět odstraníme jejich prostřední třetiny. Tímto způsobem pokračujeme až do nekonečna.



Obr. 7 Cantorova množina

Z původní viditelné úsečky zbylo tzv. Cantorovo mračno. Pokud ztotožníme toto mračno s číselnou množinou, která je vlastně na číselné ose a reprezentuje body původní úsečky, získáme Cantorovu množinu nebo také Cantorovo discontinuum.

Jelikož z původní úsečky nic nevidíme, míra nebo-li délka Cantorova discontinua je rovna 0. Co se týče otázky mohutnosti Cantorovy množiny, je naprosto stejná jako mohutnost původního intervalu  $\langle 0, 1 \rangle$ .

### 1.2.5 Oscar Rösler

O. Rösler byl původním povoláním lékař. Na poli chaosu se proslavil Röslerovým atraktorem, který představil světu 13 let po Lorenzově atraktoru. Röslerův atraktor vychází z Lorenzova atraktoru. Jedná se o umělý systém, který generuje podivný atraktor založený na nejjednodušších principech generace chaosu, a těmi jsou transformace rozprostření a ohyb.



Obr. 8 Oscar Rösler

Röslerovo jméno začalo být spojováno s jednoduchým atraktorem ve tvaru stuhy, svinuté do věnečku se záhybem. Rösler si ale také představoval atraktory z vyšších dimenzí jako ohýbání a stlačování stavového prostoru, které se skutečně stalo klíčem ke konstrukci podivných atraktorů.

Rösler zastával názor, že tyto tvary představují princip samoorganizace v přírodě (vznik disipativních struktur díky složitým a vzájemně souvisejícím procesům). Představoval si punčochu k měření větru, do které se chytil vítr. Vítr je v pasti a musí "proti své vůli" konat něco užitečného. Podle něj princip samoorganizace spočíval v tom, že příroda dělá něco proti své vůli, a díky tomu se zaplétá do sebe, a dává tak vzniknout kráse. [19]

### 1.2.6 Balthasar Van der Pol (1889 - 1959)

Tento holandský elektroinženýr jako jeden z prvních studoval experimentálně dynamiku v laboratoři ve 20. a 30. letech minulého století. Zkoumal elektrické obvody s vakuovými lampami a objevil, že mohou mít stabilní oscilace, dnes nazývané limitními cykly.

Van der Pol zjistil při experimentech, že pokud je takovýto obvod „řízen“ signálem, jehož frekvence je blízká frekvenci limitního cyklu, pak frekvence periodické odezvy tohoto dynamického systému je „strhována“ k frekvenci řídicího signálu. [19]

V roce 1927 publikoval se svým kolegou Van der Markem v časopise článek o nepravidelném šumu, který pozoroval pro některé budící frekvence. Z rekonstrukce jeho pokusů dnes víme, že objevil deterministický chaos.

Sestrojil mnoho elektronkových modelů lidského srdce a na nich studoval jeho dynamiku a stabilitu. Snažil se o řízení srdeční činnosti externím signálem, jelikož jeho cílem bylo stabilizovat srdeční aritmie. V roce 1935 byl Balthasar Van der Pol oceněn medailí IRE za obrovský přínos v oblasti elektronických obvodů a šíření elektromagnetických vln.

### **1.2.7 René Thom (2. 9. 1923 – 25. 10. 2002)**

René Thom, jeden z velkých matematiků 20. století, se narodil v roce 1923 ve městě jménem Montbéliard. Na lyceu byl premiantem nadaným nejen pro exaktní vědy, ale i pro literaturu. O tom, že se nakonec zabýval matematikou, rozhodla náhoda. Stalo se to roku 1939, kdy se blížila válka a s matematikou se dalo dostat spíše k dělostřelectvu než k pěchotě. Matematiku studoval na École normale supérieure, byl žákem Henri Cartana a zpočátku se věnoval teorii funkcí více komplexních proměnných. Ve Štrasburku pak navštěvoval seminář Charlese Ehresmanna, v němž se studovala algebraická topologie, fibrování prostory a kohomologie. Ze spojení těchto matematických teorií povstala myšlenka zkoumat místo analytických zobrazení, pouze zobrazení diferencovatelná. První práce v tomto směru publikoval v roce 1946 a v roce 1954 se mu podařilo rozřešit jeden z tehdejších problémů, a to problém kobordizmů. V roce 1958 pak za tento výsledek převzal z rukou Heinze Hopfa na matematickém kongresu v Edinburku Fieldsovu medaili.



Obr. 9 René Thom

I on přispěl svou troškou do mlýna. Thomova teorie katastrof těsně souvisí s teorií chosu. Vše začalo roku 1961, kdy byl Thom v Bonnu a navštívil přírodovědné muzeum v poppelsdorfském zámku. Nešel tam původně z nějakého přírodovědného zájmu, ale proto, že tam matematici z Bonnu pořádali recepci. Když se procházel sály muzea, uviděl sádrový model gastrulace žabího vajíčka, a tu se mu náhle objevila asociace se singularitami, které právě studoval. Model totiž vypadal jako rozvinutí jedné ze základních singularit. K tomuto nápadu se potom vrátil, začal se zabývat morfogenezí a postupně opouštěl čistou matematiku.

Výsledkem byla kniha *Strukturální stabilita a morfogeneze*, kterou René Thom napsal v letech 1966 – 1967. Rukopis poslal americkému nakladatelství Benjamin. Toto nakladatelství však projevilo značně malý zájem o text, který byl těžký a zároveň bizarní. Rukopis tam zůstal ležet asi do roku 1971, kdy při nějakých změnách v nakladatelství zjistili, že kniha dosud nevyšla. Mezitím rukopis knihy koloval v opisech a kopiích. Tímto způsobem se s ní seznámil i matematik E. C. Zeeman, který pro celou Thomovu teorii vymyslel daleko prodejnější název, a to *teorie katastrof*.

Po matematickém kongresu ve Vancouveru roku 1974 se teorie katastrof dostala do médií a stala se módní teorií. Mezitím ovšem kniha vyšla francouzsky v roce 1972, anglický překlad byl vydán roku 1975 a druhé rozšířené vydání 1977. Zároveň byla teorie katastrof některými matematiky kritizována za to, že uvedené základní pojmy „jako bifurkace, byly známy již před Thomem, a to například u J. H. Poincarého. Právě tak byl už znám pojem strukturální stability u ruského matematika A. Andronova. Thom to nepopíral, a dokonce ani to, že „jeho“ větu dokázal někdo jiný. Podstatné však bylo použití těchto zčásti starých,

Thomem však dále rozvinutých teorií, na vysvětlení diskontinuit v přírodě, na problém vzniku a zániku tvarů, na morfologii v nejširším pojetí.

### 1.2.8 Michell Jay Feigenbaum (19. 12. 1944)

Michell Jay Feigenbaum se narodil 19. 12. 1944 ve Philadelphii v USA. Je to matematický fyzik, jehož práce v teorii chaosu vedla k objevu Feigenbaumovi konstanty.

Feigenbaum je syn imigrantů z Polska a SSSR. Studoval elektroinženýrství na vysoké škole v New Yorku. V roce 1964 pokračoval ve studiu na Massachusetts Institute of Technology v doktorátu z elektroinženýrství. Přešel ale ke studiu fyziky se zaměřením na obecnou relativitu. V roce 1970 získal doktorát za práci o disperzních vztazích podle Francisova zákona.

Po jeho krátkém působení na Cornellově univerzitě a Viržinském polytechnickém institutu dostal nabídku k dlouhodobější pozici v Los Alamoské národní laboratoři ke studiu turbulence. Ačkoliv se této skupině nepodařilo postoupit v řešení nepoddajné teorie turbulence tekutin, Feigenbauma to přivedlo ke studiu chaotických zobrazení.



Obr. 10 Feigenbaum

Mnoho matematických zobrazení obsahujících jednoduchý lineární parametr začne vykazovat pro hodnoty parametru určité oblasti zcela náhodné chování, které je známé jako

chaos. Jakmile se parametr blíží této oblasti, zobrazení vykazuje bifurkace pro přesné hodnoty parametru. Nejprve existuje jeden stabilní bod, pak pro určitou hodnotu parametru začne řešení oscilovat mezi dvěma body, pro další hodnotu parametru dojde k nové bifurkaci a řešení začne oscilovat mezi čtyřmi body atd.

V roce 1975 Feigenbaum s použitím svého počítače HP-65 objevil, že poměr rozdílu mezi dvěma následnými hodnotami parametru, ve kterých dochází k bifurkaci, se postupně blíží ke konstantě přibližně 4,6692. Feigenbaum poskytl matematický důkaz tohoto faktu a ukázal, že stejné chování a stejná konstanta se objevuje v široké třídě matematických funkcí vedoucích k chaosu. Byl to jeden z prvních kroků vedoucích k popisu chaosu. Tento poměr konvergence je nyní znám jako Feigenbaumova konstanta.

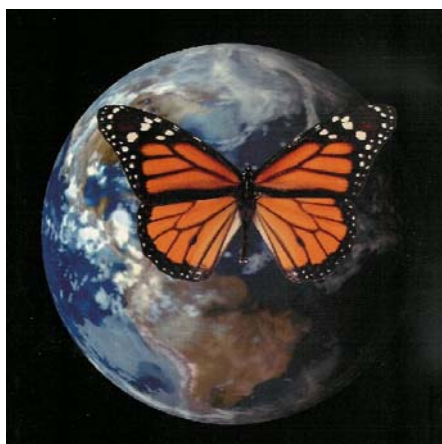


## 2 ÚVOD DO TEORIE CHAOSU

### 2.1 Teorie chaosu

Tradičně je chaos definován jako nepořádek či úplný zmatek. Chaos znamená neexistenci řádu či struktury. Jako příklad chaosu se vám může vybavít například vodopád nebo počasí. Ale začneme-li uvažovat o chaotických systémech, bude mít tato definice poněkud jiný význam. Místo naprosté neuspořádanosti nalezneme strukturu, která se řídí přesně danými pravidly. Tedy náš příklad vodopády nejsou vůbec neuspořádané systémy, jsou zcela deterministické, i když extrémně složité. Stejně tak i počasí je deterministické. Zahrnuje tolik složitých vzorců, že je musíme v současné době považovat za dlouhodobě nepředpověditelné.

V souvislosti s teorií chaosu se můžeme často setkat s pojmem “motýlí efekt”. Znamená to, že máchnutí křídel motýla na jednom konci Zeměkoule může na druhém konci způsobit větnou smršť. Tedy i malá nepřesnost v měření vstupních hodnot může znamenat, že naše předpověď bude naprosto rozdílná od skutečnosti. Tedy kdyby motýl nemávl svými křídly, trajektorie systému by mohla být zcela odlišná. S naprostou přesností nemůžeme měřit nic. Tyto jevy se nazývají chaotické.



Obr. 11 Motýlí efekt

Objevem “motýlího efektu” se proslavil Edward N. Lorenz, který prezentoval svůj objev v 60. letech minulého století. Motýlí efekt ukazuje nemožnost dlouhodobé předpovědi počasí, a tím na jeho chaotičnost. Možná předpověď počasí je 2 až 3 dny.

Termínem chaos je označena taková vlastnost nějakého dynamického a současně deterministického systému, při jejíž platnosti je nemožné vypočítat budoucí stav systému. Chaos nastává u těch dynamických systémů, které vykazují velkou citlivost na počáteční podmínky. V takových systémech se při volbě minimálně dvou nekonečně blízkých počátečních bodů, které reprezentují počáteční podmínky systému, se tyto dva body posléze exponenciálně vzdalují, takže budoucí stav systému není možné žádným způsobem předpovědět.

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 Chaos

V matematice a fyzice se teorie chaosu zabývá chováním jistých nelineárních dynamických systémů, které za jistých podmínek vykazují jev známý jako chaos, nejvýznamněji charakterizovaný citlivostí na počáteční podmínky. V důsledku této citlivosti se chování těchto fyzikálních systémů, vykazujících chaos, jeví jako náhodné, i když model systému je deterministický v tom smyslu, že je dobře definovaný a neobsahuje žádné náhodné parametry.

Vlastní výraz „deterministický chaos“ plyne z faktu, že systémy, které jej produkují, lze modelovat, a tím, že se v těchto modelech nevyskytuje náhodnost. [20]

## 2.3 Výskyt deterministického chaosu

Chaotické systémy jsou například atmosféra, ekonomie, vývoj populace, solární systém, tektonika zemských desek, turbulence tekutin. Často se setkáváme s chaosem v elektronických obvodech..

Výskyt chaosu můžeme očekávat u systémů, které obsahují vhodnou nelinearitu, nebo pokud mezi spolupracujícími systémy existuje nelineární vazba.

Samotný výskyt chaosu ještě nemusí znamenat, že se v daném systému děje něco špatného. Mnohdy to pouze znamená, že je příslušný systém „na cestě“ ke kvalitnějšímu uspořádání. [20]

## 2.4 Chaotický pohyb

Abychom mohli klasifikovat chování systému jako chaotické, musí mít systém následující vlastnosti:

- citlivost na počáteční podmínky
- musí být topologický tranzitivní
- jeho periodické orbity musí být husté

### 2.4.1 Citlivost na počáteční podmínky

Citlivost na počáteční podmínky znamená, že dvě blízké trajektorie ve fázovém prostoru se s rostoucím časem exponenciálně rozbíhají. Nebo-li malá změna v počátečních podmínkách vede po čase k velmi odlišnému výsledku. Systém se chová identicky, jen když jeho počáteční konfigurace je naprosto stejná. Příkladem takové citlivosti je motýlí efekt. Citlivost k počátečním podmínkám se dá kvantifikovat Ljapunovým exponentem.

### 2.4.2 Topologická tranzitivita

Aplikace transformace na libovolný daný interval  $I_1$  ho roztahuje až do doby, kdy překryje libovolný další interval  $I_2$ .

Tranzitivita, husté periodické body a citlivost na počáteční podmínky můžeme rozšířit na libovolný metrický prostor. J. Banks a jeho kolegové dokázali v roce 1992, že v nastavení obecného metrického prostoru tranzitivita a husté periodické podmínky implikují citlivost na počáteční podmínky.

Tento fakt vedl Bau-Sen Du z Matematického institutu, Academia Sinica, Taiwan k definici silnější verze citlivé závislosti, a to k extrémně citlivé závislosti. Extrémně citlivá závislost není důsledkem tranzitivity a hustých periodických bodů. Tato závislost znamená, že blízké body se oddělují a konvergují nekonečně často, což je častokrát právě případ chaotických dynamických systémů.

## 2.5 Systémy generující deterministický chaos

Systémy generující deterministický chaos se dělí na systémy:

- Hamiltoniánské
- disipativní

### 2.5.1 Hamiltoniánské systémy

Hamiltoniánské systémy jsou typické tím, že u nich “nedochází” k disipaci energie. Tohle musíme brát s rezervou, jelikož každý systém časem ztrácí energii. Pokud tedy systém “neztrácí” energii, je tím myšleno v rámci krátkého časového období, kdy je prováděno pozorování, a ztráta je tedy zanedbatelná.

Tyto systémy bylo možné řešit bez pomoci výpočetní techniky, což byla obrovská výhoda.

### 2.5.2 Disipativní systémy

U disipativních systémů dochází ke ztrátě energie nebo-li k disipaci. Tyto systémy jsou popsány diferenciálními rovnicemi, které umožňují syntézu jejich řízení.

Typickými příklady disipativních systému je Lorenzův atraktor, který je generován třemi diferenciálními rovnicemi, a logistická rovnice.

Struktura těchto rovnic je přesně známa, a i přesto tato sestava jednoduchých diferenciálních rovnic generuje chaotické chování bez ohledu na počáteční podmínky. [20]

## 2.6 Časové řady

Teorie chaosu přistupuje ke zkoumání časových řad zcela odlišným způsobem než statistika. Existuje několik metod, které umožňují rozhodnout, zda jsou data náhodná nebo zda vykazují určitou neviditelnou závislost. Toto zjištění je podstatné pro další analýzu a případnou predikci. [18] Náhodné hodnoty jsou nepředpověditelné a neobsahují žádnou strukturu, tedy rovnoměrně vyplňují prostor o libovolném rozměru. Neexistuje žádná vazba mezi jednotlivými hodnotami časové řady.

Data, u kterých se prokáže existence jakési vnitřní struktury, mají zcela odlišný charakter. Existují zde určité vazby mezi hodnotami. A to například, že předchozí hodnoty určují hodnoty následující. To má obrovský význam pro předpovídání dalšího vývoje.

Avšak i pro chaotická data platí, že jsou předvídatelná pouze ve velmi malém časovém okamžiku, jelikož potom dochází k jejich velké divergenci. Pro chaosy nízkého řádu platí, že jsou srozumitelné a předpověditelné. Chaos vysokého stupně se jeví jako náhodný, protože mu nerozumíme. Budeme-li se na to dívat tímto způsobem, můžeme chaos definovat jako děj, kterému nerozumíme. To, co je chaotické, není determinováno povahou dějů, které sledujeme, ale nízkou úrovní našeho chápání.

Na nejnižší z nich najdeme konstantnost – stálost. Na vyšší úrovni se setkáme s periodickými oscilacemi a na ještě vyšší úrovni najdeme vzorce, kterým nerozumíme, které však jsou přesto úplně deterministické (protože vychází z deterministických procesů) a které by snad byly také předpověditelné, kdybychom lépe rozuměli dějům, které jsou v nich skryty [18].

## 2.7 Atraktory

### 2.7.1 Co to je atraktor

Jedním způsobem vizualizace chaotického pohybu je vytvoření fázového diagramu. V takovém diagramu je čas implicitní a každá osa reprezentuje jednu dimenzi stavu. Křivka se nazývá orbit.

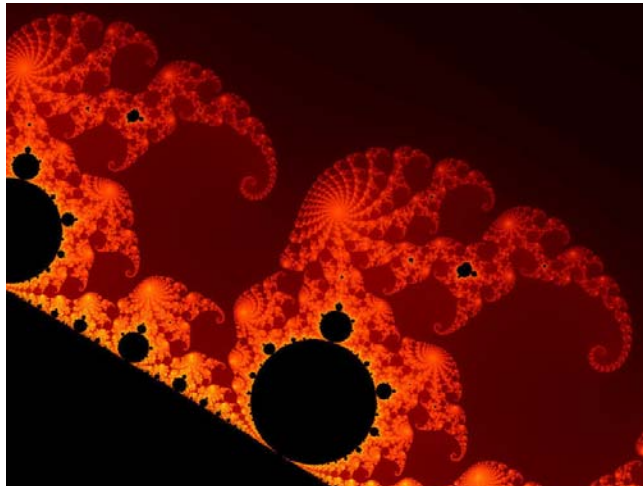
Mnohokrát je ve fázových diagramech vidět, že většina stavových trajektorií se přibližuje a omotává nějakou obecnou limitu. Systém končí ve stejném pohybu pro všechny počáteční stavy v oblasti okolo tohoto pohybu tak, jako by byl systém k tomuto pohybu přitahován. Tento cílový pohyb je nazýván atraktor. Atraktory jsou velmi často spojovány disipativními systémy, kde některý prvek spotřebovává energii. V libovolném bodě fázového diagramu se stav systému mění určitým deterministickým způsobem.

Slovo atraktor je odvozeno z anglického slovesa “to attract“, což znamená přitahovat. Atraktor dynamického systému je množina stavů, do kterých systém směřuje. Jinými slovy se jedná o množinu hodnot, kterých může nabývat stavový vektor dynamického systému po dostatečně dlouhém časovém úseku od počátečního impulsu, respektive od chvíle, kdy je systém inicializován.

Atraktorem je jakákoliv množina bodů, která má zjevnou tendenci směřovat k stabilnímu stavu nebo se v něm trvale pohybovat. Jedná se tedy o popis systému, který směřuje do jednoho určitého stabilního stavu.

Ne všechny atraktory jsou body. Některé atraktory jsou jednoduchými smyčkami, jiné jsou složitějšími dvojitými smyčkami, pro ty je potřeba více než dva stupně volnosti. Další atraktory jsou skutečnými fraktály, to jsou tak zvané podivné atraktory.

Termín fraktál poprvé použil v roce 1975 matematik Benoît Mandelbrot. Tento pojem pochází z latinského slova fractus, tedy rozbitý. Fraktál je geometrický objekt, který je soběpodobný. To znamená, že pokud daný útvar pozorujeme v jakémkoliv měřítku, v jakémkoliv rozlišení, pozorujeme stále opakující se určitý charakteristický tvar.



Obr. 12 Detail Mandelbrotovy množiny, jednoho z nejnámějších fraktálů

Ale zpět k atraktorům. Systémy s atraktory ve tvaru smyčky vykazují periodický pohyb. Systémy se složitěji rozdělenými smyčkami vykazují kvaziperiodický pohyb. A systémy s podivnými atraktory vykazují chaotické chování.

### 2.7.2 Druhy atraktorů

- množina pevných bodů
- množina periodických bodů
- množina kvaziperiodických bodů

- chaotický atraktor
- podivný atraktor

### **2.7.2.1 Množina pevných bodů**

Nejjednodušším případem atraktoru dynamického systému jsou pevné body. Systém se v nekonečném čase ustálí v nějakém stabilním stavu, který je možné dopředu vypočítat.

### **2.7.2.2 Periodické a kvaziperiodické body**

Dynamický systém se po určité době ustálí tak, že osciluje mezi několika stavy. Ty mohou být buď spočítatelné nebo nespočítatelné. V případě spočítatelných stavů se může například jednat o astabilní klopný obvod. V druhém případě to může být například dráha osamělé planety okolo Slunce.

### **2.7.2.3 Chaotický atraktor**

Máme-li chaotický atraktor, znamená to, že výsledný stav systému nelze dopředu předpovídat. To může být způsobeno mimo jiné tím, že je systém velice citlivý na počáteční podmínky. Chaotičnost v tomto případě neznamená náhodnost, jelikož stále hovoříme o systémech deterministických.

### **2.7.2.4 Podivný atraktor**

Chaotický pohyb vede k tomu, co je známé jako podivný atraktor. Podivné atraktory vykazují velkou složitost a velkolepé detaily.

Podivné atraktory se objevují jak ve spojitých dynamických systémech, jak je Lorenzův systém, tak i v některých diskrétních systémech, jako je Hénonovo zobrazení. Jiné diskrétní dynamické systémy mají odpuzující strukturu nazývanou Juliovy množiny, která tvoří



hranici mezi oblastmi přitažlivosti pevných bodů. Juliovy množiny lze pokládat za podivné odpuzovače. Jak podivné atraktory, tak i Juliovy množiny mají typickou fraktální strukturu.

Termín podivný atraktor není ještě přesně matematicky definován. Tento typ atraktoru může vzniknout tehdy, je-li systém popsán minimálně třemi navzájem souvisejícími diferenciálními rovnicemi. V případě, že funkce použité v rovnicích není možné invertovat, může být rovnic i méně. Všechny chaotické atraktory jsou současně podivnými atraktory, opačná implikace však neplatí.

Poincaré-Bendixsonova věta říká, že podivný traktor může ve spojitém dynamickém systému vzniknout jen tehdy, má-li tři nebo více dimenzí. Avšak žádné takové omezení neplatí pro diskrétní systémy, které vykazují podivné atraktory ve dvou nebo v jednodimezionálních systémech.

### 2.7.3 Lorenzův atraktor

Lorenzův atraktor, zavedený Edwardem Lorenzem v roce 1963, je nelineární trojdimenzionální deterministický dynamický systém odvozený ze zjednodušených rovnic vynucené konvekce v atmosféře. [14] Pro jistou množinu parametrů systém vykazuje chaotické chování a zobrazuje to, co se dnes nazývá podivný atraktor. Toto bylo dokázáno W. Tuckerem v roce 2001.

Lorenzův atraktor je generován třemi diferenciálními rovnicemi:

$$\frac{dx}{dt} = \sigma(y - x) , \quad (1)$$

$$\frac{dy}{dt} = x(r - z) - y , \quad (2)$$

$$\frac{dz}{dt} = xy - bz , \quad (3)$$

kde  $\sigma$  je Prandlovo číslo a  $r$  je Reynoldsovo číslo,  $r, \sigma, b > 0$ , obvykle  $\sigma = 10$ ,  $b = 8/3$  a  $r$  se mění.



Obr. 13 Graf trajektorie Lorenzova systému pro hodnoty  $r = 28$ ,  $\sigma = 10$ ,  $b = 8/3$

## 2.8 Bifurkace

Termínem bifurkace se značí jev, při kterém dochází k velkým změnám vnitřního stavu ve sledovaném systému v případě, že se vstupní parametry nepatrně změní. U některých systémů k bifurkacím nedochází, protože se s postupným zvyšováním či snižováním hodnot vstupních parametrů systém mění pouze nepatrně. Ve značně jednoduchých případech se dokonce jedná o lineární závislost na vstupních parametrech.

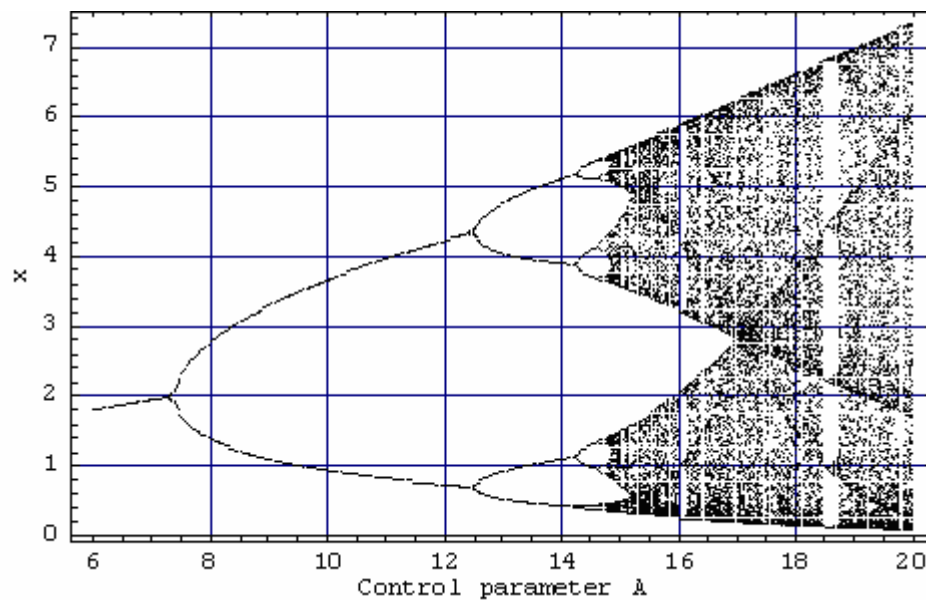
Existují však systémy, u nichž po dosažení určitých kritických hodnot na vstupu dochází k prudké změně vnitřního stavu. Může se například jednat o vznik turbulence v potrubí při dosažení určité rychlosti proudění kapaliny či o fázový přechod látky při změně teploty.

Bifurkace do značné míry souvisí s chaosem. Chaos nastává v případě, že v systému začíná docházet k velkému množství na sebe navazujících bifurkací.

## 2.9 Znázornění chaotického chování

### 2.9.1 Bifurkační diagram

Bifurkační diagram slouží pro znázornění chaotického chování. V tomto diagramu jsou na horizontální ose naneseny hodnoty vstupního parametru a na vertikální ose jsou vyneseny hodnoty jednoho z vnitřních stavů, jichž systém v po sobě následujících krocích nabývá.

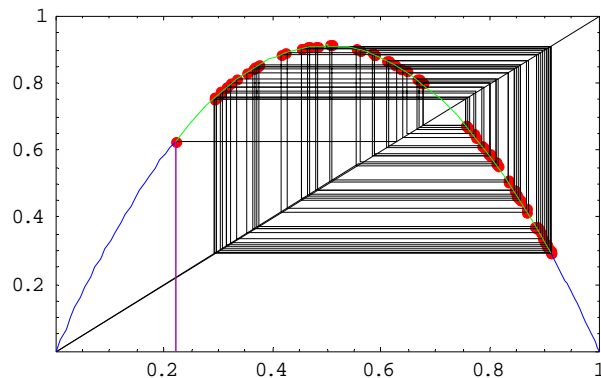


Obr. 14 Bifurkační diagram

Na obrázku (Obr. 12) je znázorněn bifurkační diagram. Jak vidíme, pro některé vstupní hodnoty je systém stabilizován, pro další hodnoty dochází k periodickému překmitávání mezi několika stavy a pro zbylé hodnoty se systém chová chaoticky.

### 2.9.2 WEB diagram

WEB diagram je další způsob, jak znázornit chaotické chování systémů. Vizualizuje charakter daného singulárního nebo-li pevného bodu. Lze ho smysluplně použít jen pro ty systémy, jejichž chování se mění na základě jednoho řídicího parametru. WEB diagram se skládá ze dvou částí.



Obr. 15 WEB diagram

První část tvoří přímka o rovnici  $x = y$ . Jedná se tedy o diagonálu daného grafu. Přímka směřuje z počátku a představuje množinu pevných bodů. Body mají charakter odpuzování nebo přitahování.

Pokud by byl takový např. přitahující bod dosazen do logistické rovnice, pak by funkční hodnota byla stejná jako argument. [20]

Druhou část grafu představuje vlastní křivka, která je daná logistickou rovnicí. Průnik křivky s diagonálou udává pozici singulárního bodu ve WEB diagramu a jeho kvalitu (zdroj, propad).

Kvalita singulárního nebo-li pevného bodu je dána derivací logistické rovnice v místě tohoto bodu. Je-li  $df/dx < 0$ , pak je daný pevný bod typu propad, jelikož všechny okolní trajektorie k sobě přitahuje. V druhém případě může nastat možnost, kdy  $df/dx > 0$ , potom

jsou všechny trajektorie odpuzovány, až uniknou k jinému singulárnímu bodu, nebo se ustálí v limitním cyklu.

Ve WEB diagramu jsou důležité svislé spojnice, vodorovné spojnice můžeme označit jako pomocné části grafu.

## 2.10 Feigenbaumova konstanta

Feigenbaumovi konstanty jsou dvě matematické konstanty pojmenované po matematikovi Mitchellu Feigenbaumovi. Obě vyjadřují poměr v bifurkačním diagramu a jsou transcendentní.

Transcendentní číslo je takové komplexní číslo, které není kořenem žádné algebraické rovnice s racionálními koeficienty.

### 2.10.1 Feigenbaumova konstanta delta

Feigenbaumova konstanta delta nese hodnotu:

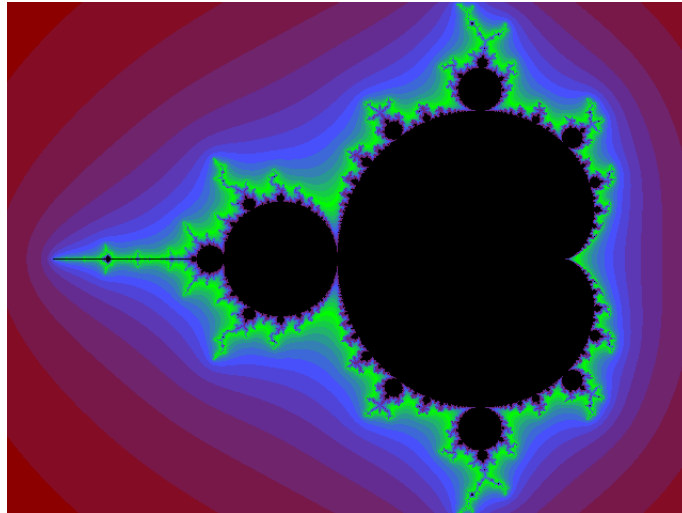
$$\delta = 4,66920160910299067185320382 \quad (4)$$

Feigenbaum původně objevil toto číslo v bifurkacích logických zobrazení, ale také ukázal, že toto číslo vzniká ve všech jednodimenzionálních bifurkačních zobrazeních s kvadratickým maximumem, vedoucích k chaosu. Feigenbaumovu konstantu lze použít k předpovědi, když v systému vznikne chaos.

Jedná se o poměr mezi následnými bifurkacemi nebo mezi průměry po sobě jdoucích kruhů na ose Madelbrotovy množiny.

Mandelbrotova množina je jeden z nejznámějších fraktálů. Celá množina leží uvnitř kruhu se středem v počátku souřadnic s poloměrem 2. Roku 1982 A. Douady a J. H. Hubbard

dokázali, že množina je jednoduše souvislá. Množina je kompaktní a sestává se ze spočetně nekonečného množství podobných kardioidám a kruhům, které se vzájemně dotýkají.



Obr. 16 Mandelbrotova množina

### 2.10.2 Feigenbaumova konstanta alfa

Feigenbaumova konstanta alfa nese hodnotu:

$$\alpha = 2,502907875095892822283902873218 \quad (5)$$

Jedná se o poměr šířky mezi dvěma hroty mezi dvěma bifurkačními body.

## 2.11 Teorie katastrof

Tato poměrně nová teorie se zabývá vlastnostmi množin parametrů dynamických systémů, při kterých dochází ke kvalitativním změnám stavu systému. Takové množiny se nazývají katastrofické a Thom dokázal, že za určitých předpokladů existuje pouze sedm elementárních typů těchto katastrofických množin. Teorie katastrof se tedy zabývá strukturní stabilitou dynamických systémů, zkoumá vznik, změnu nebo ztrátu stability.

Vlastní teorie katastrof pracuje s modely reálného světa a hledá tzv. katastrofickou množinu. Tato tak zvaná katastrofická množina nám říká, za jakých podmínek a kdy dojde v daném systému ke katastrofě.

Pomocí teorie katastrof můžeme objasnit případ, kdy při plynulé změně vhodných parametrů systému může čistě deterministický systém přejít přes sérii bifurkací až do chaotického režimu.

V případě deterministického chaosu se katastrofou rozumí tzv. bifurkace (změna počtu a charakteru singulárních bodů ve stavovém prostoru systému), při níž dochází ke kvalitativním změnám ve struktuře stavového prostoru daného systému. [20]

Nejjednodušší katastrofy jsou typu:

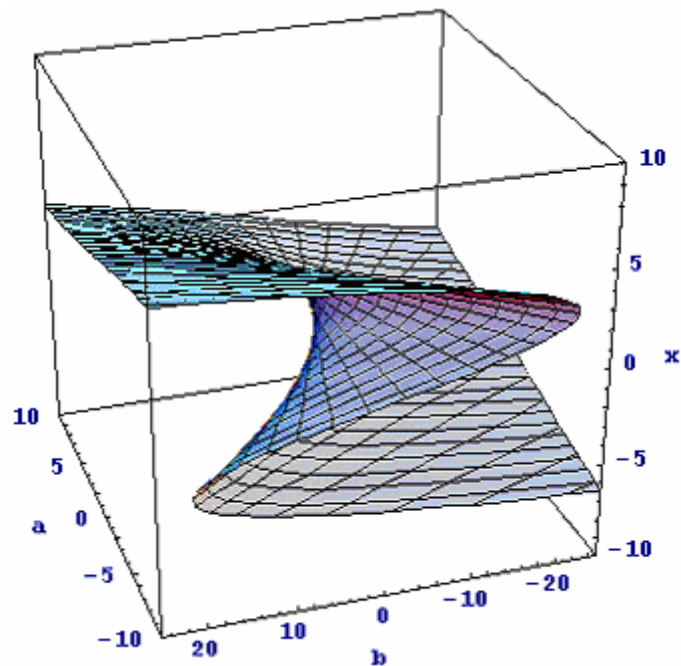
- záhyb
- pyramida

### 2.11.1 Katastrofa typu záhyb

Katastrofa typu záhyb umožňuje jednoduše demonstrovat ideu teorie katastrof. [20] Každý bod plochy je určen třemi čísly, a to parametry  $a$ ,  $b$ , které jsou parametry modelu (např. teplota, napětí), a závisle proměnnou  $x$ , která představuje například stav systému či mechanickou pevnost.

Mění-li se hodnoty parametrů, a to například  $a = \langle -10, 10 \rangle$ ,  $b = \langle -20, 20 \rangle$ , potom stav daného systému můžeme chápat jako bod pohybující se po ploše v závislosti na parametrech. Překročí-li parametr  $b$  hodnotu 20 a parametr  $a < 0$ , potom se stane následující. Pro tuto a větší hodnotu parametru  $b$  je závisle proměnná  $x$  definována mnohem výše než do-

posud, a to znamená, že systém skočí z jednoho stavu do druhého. Jinými slovy stane se katastrofa. Podobná interpretace platí i u složitějších systémů s více řídicími parametry.

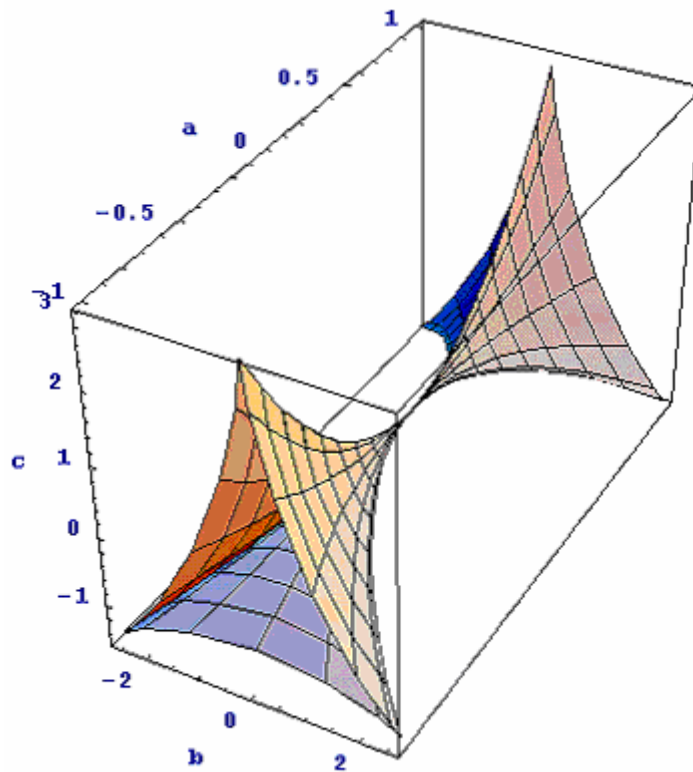


Obr. 17 Katastrofa typu záhyb

### 2.11.2 Katastrofa typu pyramida

Plocha tohoto tvaru je tvořena množinou bodů, které představují katastrofu. Jinými slovy, pokud se v daném dynamickém systému mění tři parametry, pak tzv. parametrická trajektorie v tomto parametrickém prostoru dynamického systému při průchodu plochou indikuje vznik katastrofy, která může znamenat např. právě změnu charakteru singulárního bodu systému – bifurkaci.[20]





Obr. 18 Katastrofa typu pyramida

## 2.12 Logistická rovnice

Logistická rovnice je jeden z velice jednoduchých modelů chaotického chování. Rovnice byla vytvořena jako odezva na potřebu simulace biologických systémů a v rámci těchto systémů popisuje chování druhu v jeho přirozeném prostředí.

Logistická rovnice byla vytvořena v 70. letech 20. století biologem R. M. Mayem na základě studia dynamiky životních cyklů mušek.

Tato iterativní funkce je dána vztahem:

$$x_{n+1} = Ax_n(1 - x_n) \quad (6)$$

Rovnice popisuje nelineární dynamiku počtu jedinců v po sobě následujících populacích. Při nárůstu parametru, který charakterizuje nelinearitu, dochází k bifurkacím, čili k postupnému zdvojení period a následně k chaotickému chování, které vykazuje výrazný fraktální charakter. Daná zdvojení se v přesném měřítku opakují.

Přes svou jednoduchost generuje logistická rovnice velmi komplikované chování, a to od chování deterministického až po chování chaotické.

### 2.13 Pevný bod

Pojem pevný bod nebo-li fixed point úzce souvisí s logistickou rovnicí. Aplikuje-li se logistická rovnice iterativním způsobem, obdržíme sekvenci hodnot  $x_1, x_2, \dots, x_n$ . Této sekvenci hodnot se říká trajektorie nebo orbita, a to z toho důvodu, že při vykreslení těchto hodnot získáme graf, který zobrazuje chování iterativní funkce.

Pokud se pozdější hodnoty této trajektorie ustálí na konstantní hodnotě, pak se této hodnotě říká pevný nebo-li singulární bod. [12] Singulární z toho důvodu, protože se jedná v podstatě o stejný typ objektu, který se vyskytuje u dynamických systémů. Ustálí-li se trajektorie v periodickém cyklu, pak je singulární bod mezi hodnotami cyklu.

Pro logistickou rovnici je singulární bod na pozici  $x = 0$ ,  $x = 1-1/A$ . Jestliže vývoj systému začne přesně na pozici singulárního bodu, pak v tomto bodu setrvá. Je-li však pertrubován, pak se trajektorie vyvíjí směrem k tomuto bodu nebo od něj, a to v závislosti na jeho typu.

[12]

## 3 ŘÍZENÍ A SYNCHRONIZACE DETERMINISTICKÉHO CHAOSU

### 3.1 Úvod

Dnešní věda velice pokročila v oblasti nové vědy zvané chaos. Jejím důležitým úkolem je systematicky pracovat na tom, abychom naše znalosti v oblasti chaotického chování využili v aplikacích. Z tohoto důvodu je řízení chaosu velmi významné.

Řízení chaosu zahrnuje potlačení chaosu, a to tehdy, kdy je škodlivý, a produkování chaosu tehdy, kdy je užitečný (synchronizace chaosu) a kdy řízení systému vede do určitého cílového stavu vloženého do chaotických atraktorů. To jsou důvody, proč se oblast synchronizace a řízení chaosu stala žhavým tématem v několika málo letech.

Řízením deterministického chaosu se rozumí takové působení na daný chaotický systém tak, aby se z režimu chaotického dostal systém do režimu periodického či neperiodického ustáleného stavu. [20] Existují i opačné tendence, a to takové, že řízením systému tento systém přivedeme do chaotického režimu, jelikož některé děje mají optimální průběh právě při chaotickém chování. Jedná se například o chemické reakce.

V publikaci Controlling chaos přišli jako první s myšlenkou řízení deterministického chaosu Ott, Greboki a York. Jedná se o tzv. OGY algoritmus.

Principem řízení chaosu je ovlivňování aktuálního stavu systému vykazujícího chaos malými pertrubacemi za účelem jeho stabilizace do periodického chování či ustáleného stavu. Pertrubace může ovlivnit chaotické chování jen tehdy, pokud je aktuální stav systému dostatečně „blízko“ režimu, do kterého má být systém převeden. [20]

Stavová trajektorie „dostatečně hustě“ navštíví všechna místa stavového prostoru, díky ergodicitě a díky tomu, že systém je v chaotickém stavu. Počkáme, až se stavová trajektorie přiblíží k požadovanému stavu, ve kterém chceme systém udržet, a následně provedeme řídicí zásah. A to tím způsobem, že aktivujeme pertrubaci.

Pro řízení chaosu založeného na logistické rovnici je velikost pertrubace řídicího parametru dána rovnicí:

$$\Delta A_n = \frac{A_0(2x(i) - 1)(x_n - x(i))}{x(i)(1 - x(i))} \quad (7)$$

Principy OGY algoritmu jsou vhodné pro řízení chaotických systémů s nízkou dimenzí, obvykle méně jak 5, pro systémy s vyšší dimenzí tedy pro časoprostorové chaosity je obvykle tento algoritmus nevhodný.

Při řízení časoprostorového chaosu vzhledem k velkému počtu stupňů volnosti musí být příslušné řízení vhodně distribuováno. Princip je založen na aplikaci řídicího zásahu v jednom místě a ten je pak díky fyzikálním vlastnostem systému distribuován do ostatních míst.

Tento způsob řízení není však vhodný pro systémy typu systémů, jako jsou například chemické reakce, optické systémy či biologické systémy typu „srdce“. Z tohoto důvodu byla vypracována metoda využívající topologických defektů k řízení daného systému. Jedná se například o řízení časoprostorového chaosu, který lze modelovat komplexní Ginzburg-Landauovou rovnicí.

## 3.2 Časový chaos

Systémy časového chaosu demonstrují deterministický chaos v časové oblasti. Jinými slovy chaotické chování je možno pozorovat při časovém rozvoji chování systému.

Pro řízení časového chaosu nebo-li nízkodimenzionálního chaosu jsou vhodné principy OGY algoritmu. Tento nízkodimenzionální časový chaos je reprezentován takovými příklady, jakými jsou již zmíněný Lorenzův atraktor či logistická rovnice.

### 3.3 Časoprostorový chaos

#### 3.3.1 Systémy časoprostorového chaosu

Systémy, které vykazují časoprostorový chaos jsou modelovány parciální diferenciální rovnicí (partial differential equations – PDE), vázanou obyčejnou diferenciální rovnicí (coupled ordinary differential equations – CODE) a vázaným polem mřížky (coupled map lattices – CML). Jedná se o systémy s rozprostřenými parametry.

Chování těchto systémů už lze mnohdy definovat jako samoorganizaci, při níž se v chování systému objevuje nová kvalita (obvykle geometrické uspořádání), která nebyla do systému implementována.

Pro řízení systémů generujících časoprostorový chaos existují metody, které lze chápat jako rozšíření základní myšlenky o řízení chaosu. [20] Při jeho řízení musí být příslušné řízení vhodně distribuováno z důvodu velkého počtu stupňů volnosti. Můžeme říct, že stupeň volnosti značí, s kolika proměnnými můžeme ještě “hýbat”.

Příklad časoprostorových systémů můžeme nalézt v hydrodynamice, v chemii, jako jsou například chemické oscilace, dále v problematice laseru či supravodivosti.

#### 3.3.2 Logistické rovnice v paralelním spojení

Jako příklad tohoto typu chování můžeme uvést logistické rovnice v paralelním spojení, které je označováno jako CML nebo-li vzájemně vázaná pole mřížky. V tomto časoprostorově vázaném systému se jedná o vývoj  $n$  rovnic, které se navzájem ovlivňují přes vazební konstantu označovanou jako  $\epsilon$ . Na CML můžeme tedy pohlížet jako na pole “oscilátorů“, které se navzájem ovlivňují.

Matematický popis CML:

$$x_{n+1}(i) = (1 - \varepsilon)f(x_n(i)) + \frac{\varepsilon}{2}[f(x_n(i-1)) + f(x_n(i+1))] \quad (8)$$

Kde funkce  $f(\dots)$  představuje příslušnou iterační rovnici.

Okrajová podmínka CML může být buď otevřená, periodická nebo fixovaná v závislosti na problému.

Rovnice představuje symetrický CML, jelikož přes vazební konstantu  $\varepsilon$  působí příslušná k-tá rovnice na své sousední rovnice po obou stranách stejně.

### 3.3.3 Přenos informace

Vygenerujeme pomocí přesně stanoveného řídicího zásahu chaotickou sekvenci na vhodně zvoleném CML. Tuto sekvenci zdigitalizujeme a nuly nahradíme hodnotou  $-1$ . Máme-li k dispozici například 100 hodnot z vývoje CML nebo-li vázaného pole mřížky a každá tato hodnota bude zdigitalizována s přesností na 10 míst, pak výsledkem bude  $100 \cdot 10 = 1000$  bitů.

Vlastní zpráva je také reprezentována sekvencí hodnot  $-1$  a  $1$ . Tato sekvence je namodulována na zdigitalizovaný chaotický nosič, a to pomocí součinu:

$$G = M \cdot S, \quad (9)$$

Kde  $S$  je chaotická nosná složka.

Výsledkem je signál o stejné délce jako signál původní chaotický, ale nesoucí informaci, která je nečitelná pro nezasvěceného příjemce. [12] K přečtení zprávy musíme znát velice přesně onen stanovený řídicí zásah, můžeme ho nazvat klíčem. Pomocí tohoto řídicího zásahu se vygeneruje chaotická nosná složka  $S$  a za pomoci korelační analýzy se z chaotického signálu vyjme užitečný signál.

Pokud by neoprávněný příjemce odhadl klíč, pak i odchylka v řádu  $10^{-6}$  způsobí, že informace nebude dekodována. To je dáno faktem, že systémy deterministického chaosu jsou citlivé na počáteční podmínky. [12]

### 3.4 Samoorganizace

#### 3.4.1 Historie

Samoorganizace je jev, který je intenzivně studován až v minulém století. V roce 1900 Henri Bénard jako první zaznamenal proces samoorganizace, a to když studoval chování molekul v závislosti na rostoucí teplotě. Roku 1916 se o vysvětlení tohoto jevu pokusil lord Rayleigh. Z toho důvodu se tento jev nazývá Rayleigh – Bénardova hydrodynamická nestabilita.

V roce 1999 vědecký publicista Roger Lewin vydal knihu *Complexity: Life at Edge of Chaos*, která se zabývá vědeckým výzkumem samoorganizace v různých kontextech, jako jsou celulární automaty, náhodné Booleovské cykly, umělý život, sebemodifikující programy, embryologie, morfogeneze, ekosystémy a přírodní evoluce populací a druhů.

Lewin ve své knize odkazuje na hypotézu Stuarta Kauffmana z Institutu v Santa Fe, podle níž organizované systémy mají tendenci se vyvíjet k tomu, co označuje jako hrana chaosu. Zatímco řád je sterilní a chaos je destruktivně nestabilní, hrana chaosu je podle Stuarta Kauffmana oblast, kde složitost, zdatnost a výpočetní schopnost jsou optimální.

#### 3.4.2 Bělousov – Žabotinského reakce

Samoorganizace je po teoretické stránce výborně propracována v chemii, kde zapřičiňuje to, že se při vhodných podmínkách začnou miliony molekul chovat jako inteligentní jedinci.

Takovým příkladem je právě Bělousov – Žabotinského reakce. Při této reakci se začínou v tekutině cyklicky vytvářet okem viditelné barevné vzory s nádherným geometrickým motivem.



Obr. 19 Bělousov – Žabotinského reakce

Obrazce vznikají díky difuzi příslušných reaktantů v nejjednodušším uspořádání kyseliny citrónové, bromidu draselného, kyseliny sírové a iontů céru.

### 3.4.3 Entropie

Entropie vyjadřuje míru neuspořádanosti daného systému. Čím více je daný systém neuspořádaný, tím větší je entropie a naopak. Samoorganizace potom vzniká v systému, který se skládá z obrovského množství samostatných jednotek, které tvoří menší systémy, mezi kterými pak dochází k „tokům“ entropie.

Tímto způsobem dojde ke vzniku disipativních struktur, u kterých můžeme za určitých podmínek pozorovat chaotické chování.



V současné době se v biologii a přílehlých vědách objevuje stále více názor, že je to právě samoorganizace, které vděčíme za to, že na Zemi existuje život. [11]

#### **3.4.4 Teorie samoorganizace**

Teorie samoorganizace obvykle tvrdí, že "celek je více než jeho části", avšak současně tvrdí, že všechno je obsaženo v těchto částech a nic není přidáno z vnějšku. Tento paradox je pro teorii samoorganizace charakteristický, pokud se snažíme vysvětlit, jak se celek "samovolně" objevil s novými vlastnostmi, které jsou důsledkem vzájemných interakcí jednotlivých částí.

## 4 SOMA ALGORITHMUS

### 4.1 Samo-Organizující se Migrační Algoritmus

Zkratka SOMA znamená Samo-Organizující se Migrační Algoritmus. Algoritmus existuje od roku 1999 a jeho činnost je založena na geometrických principech. SOMA algoritmus, který pracuje s populací jedinců, byl vyvinut na principech, které lze odpozorovat v přírodě, kdy skupina inteligentních jedinců kooperuje na řešení společného problému, jako je například hledání zdroje potravy. Algoritmus je založen na kooperativním prohledáváním nebo-li migraci prostoru možných řešení daného problému. Smyslem je nalezení nových lepších řešení nebo-li pozic na dané hyperploše.

Vlastnost samoorganizace u algoritmu SOMA plyne z faktu, že se jedinci ovlivňují navzájem během hledání lepšího řešení, populace si tedy sama organizuje vzájemný pohyb jedinců. Algoritmus funguje ve smyslu nalezení globálního extrému na dané hyperploše.

### 4.2 Parametry

Běh algoritmu SOMA je ovlivňován skupinou parametrů, tyto parametry se dělí na řídicí a ukončovací. Řídicí parametry mají vliv na kvalitu běhu algoritmu, a to z hlediska hodnoty účelové funkce. Ukončovací parametry ukončují běh algoritmu, a to za předem nadefinovaných podmínek. Všechny tyto parametry musí být uživatelem zadány před během algoritmu.

#### 4.2.1 Mass

Parametr Mass patří do skupiny řídicích parametrů. Tento parametr je někdy nazýván Path length. Jeho hodnota leží v intervalu  $\langle 1, 1;3 \rangle$ . Parametr Mass značí, jak daleko se aktivní jedinec zastaví od vedoucího jedince.

Zadáme-li parametr  $Mass = 1$  aktivní jedinec se zastaví přesně na pozici vedoucího jedince. Bude-li se parametr  $Mass = 2$ , zastaví se za vedoucím jedincem ve vzdálenosti z jaké startoval. Pokud bychom nastavili parametr  $Mass < 1$ , došlo by k tomu, že aktivní jedinec se zastaví před vedoucím jedincem a budou nalezeny pouze lokální extrémy.

#### 4.2.2 Step

Parametr Step leží v intervalu  $\langle 0, 11; Mass \rangle$  a patří do řídicích parametrů. Tento parametr udává vzorkovací frekvenci nebo-li hustotu skoku, s jakou se prohledává prostor možných řešení.

Nastavíme-li parametr Step na nízkou hodnotu, prostor možných řešení se pak bude prohledávat podrobněji, a tím zvýšíme pravděpodobnost nalezení globálního extrému.

#### 4.2.3 PRT

Řídicí parametr PRT znamená pertrubaci a nachází se v intervalu  $\langle 0, 1 \rangle$ . Podle PRT se tvoří pertrubační vektor, který ovlivňuje to, jakým směrem se bude aktivní jedinec pohybovat. Znamená to, zda se aktivní jedinec bude pohybovat směrem k vedoucímu jedinci, či naopak.

Při pohybu jedinců skrz prostor možných řešení je jejich pohyb náhodou rušen, tzn. pertrubován. Jak silně bude pertrubace probíhat závisí na nastavení PRT parametru. Pro každého jedince je pomocí tohoto parametru generován pertrubační vektor, který je platný jen pro jeden aktuální běh aktivního jedince.

Pro každý parametr pertrubačního vektoru je generováno náhodné číslo z intervalu  $\langle 0, 1 \rangle$  a porovnává se PRT parametrem. Je-li číslo vygenerované z intervalu menší než PRT parametr, pak je danému prvku pertrubačního vektoru přiřazena 1, v opačném případě je přiřazena 0.

Pokud PRT vektor má všechny prvky rovny 1, pak aktivní jedinec putuje rovnou podle směrového vektoru  $k$  vedoucímu jedinci. Jsou-li některé prvky rovny 0, pak by se dalo říct, že dané souřadnice jsou zmražené, tedy se nemění.

#### 4.2.4 DIM

Parametr DIM někdy označován parametr D udává počet argumentů účelové funkce. Jinými slovy můžeme říct, že se jedná o rozměr jedince nebo-li značí počet prvků v jedinci. Velikost tohoto parametru je dána problémem.

#### 4.2.5 POPSIZE

Tento parametr je někdy značen zkratkou NP a určuje, kolik jedinců bude tvořit populaci. Doporučený rozsah tohoto řídicího parametru je <10, zadává uživatel>.

#### 4.2.6 Migrace

Tento ukončovací parametr udává, kolikrát se populace jedinců přeorganizuje. Doporučený rozsah parametru Migrace leží v <10, zadává uživatel>.

#### 4.2.7 MINDIV

Zkratka MINDIV znamená minimální diverzitu. Tento parametr je někdy také označován jako AcceptedError a definuje, jaký maximální rozdíl mezi nejhorším a nejlepším jedincem je v aktuální populaci povolen, a to ve smyslu účelové funkce. Je-li rozdíl menší jako hodnota MINDIV, pak je běh algoritmu ukončen.

#### 4.2.8 Specimen

Nadefinováním parametru Specimen vygenerujeme celou počáteční populaci. Používá se také ke korekci parametrů jedince při překročení hranic prohledávaného prostoru.

## 4.3 Princip algoritmu SOMA

### 4.3.1 Definice parametrů

Než-li začne běh algoritmu SOMA, je nezbytně nutné provést nadefinování parametrů, a to jak řídicích, tak i ukončovacích. Rovněž je nutné nadefinovat účelovou funkci, kterou budeme optimalizovat a která nám reprezentuje problém. Účelová funkce vrací skalár, který slouží jako měřítko kvality daného jedince.

### 4.3.2 Tvorba populace

V tomto kroku je vytvořena prvopočáteční populace pomocí generátoru náhodných čísel. [10] Pro každý parametr jedince je generováno náhodné číslo pomocí Specimenu a generátoru náhodných čísel. Typ a hodnota vygenerovaného náhodného čísla je dána rozsahem v odpovídajícím prvku Specimenu.

### 4.3.3 Migrační kola

SOMA algoritmus pracuje v cyklech zvaných migrační kola. Během migračních kol jsou přemísťováni do finální pozice pomocí sekvence pozic vypočítaných vzhledem k pozici Leadera. Jinými slovy jedinci migrují skrz prostor možných řešení.

Každý jedinec je ohodnocen účelovou funkcí a z nich je zvolen Leader pro následující kolo. Leaderem se stává jedinec s nejlepší hodnotou účelové funkce, a to ve smyslu minima či maxima.

V dalším kroku se jedinci začnou pohybovat pomocí skoků směrem k Leaderovi. Velikost skoků je dána parametrem Step. Po každém skoku si jedinec přepočítá hodnotu účelové funkce, a je-li lepší nežli předchozí hodnota, pak si ji zapamatuje. Jedinci skáčou tak dlou-

ho, dokud není dosažena pozice, která je dána parametrem Mass. Po skončení běhu se jedinec vrací na pozici, kde byla vypočítána nejlepší hodnota účelové funkce.

#### **4.3.4 Testování naplnění ukončovacích parametrů**

V tomto kroku je kontrolováno, zda byly vykonány migrační cykly v počtu, jenž je dán parametrem Migrace, a zda je rozdíl mezi nejhorším jedincem a Leaderem menší než hodnota parametru AcceptedError.

#### **4.3.5 Stop**

V posledním kroku zjistíme nejlepší nalezené řešení po posledním migračním kole nebo-li se jedná o návrat nejlepšího jedince.

## 5 DIFERENCIÁLNÍ EVOLUCE

Diferenciální evoluce je nový typ evolučního algoritmu vyvíjeného od roku 1995. Stejně jako ostatní evoluční algoritmy i diferenciální evoluce pracuje s populacemi a důležitou roli zde také hraje mutace. Rozdíl u diferenciální evoluce je ten, že potomek je tvořen ze čtyř rodičů a ne se dvou, jak tomu bývá obvykle. Proces křížení u diferenciální evoluce nastává až po mutaci, což je další zvláštností tohoto algoritmu. Cílem diferenciální evoluce je vyšlechtit v cyklech, které nazýváme generace, co nejlepší populaci jedinců, a to ve smyslu hodnot účelové funkce.

### 5.1 Parametry

#### 5.1.1 CR

Jedná se o práh křížení. Nabývá hodnot v intervalu  $\langle 0, 1 \rangle$ . Nastavíme-li hodnotu CR na 0, mutace se nedostane do zkušebního jedince, a to má za následek to, že zkušební jedinec bude přesnou kopií aktuálního rodiče. Vývoj evoluce se zastaví. Při nastavení CR na 1 se zkušební jedinec vytvoří jen ze tří náhodně vybraných rodičů.

#### 5.1.2 NP

Tento parametr udává velikost populace. Hodnoty parametru NP leží v intervalu  $\langle 2D, 100D \rangle$ , přičemž D je dimenze problému.

#### 5.1.3 F

Řídící parametr F je mutační konstanta nabývající hodnot  $\langle 0, 2 \rangle$ .

#### 5.1.4 Generations

Generations udává počet evolučních cyklů tzv. generací, během kterých se celá populace vyvíjí. Tento parametr zadává uživatel a musí být větší než 0.

### 5.1.5 D

D znamená dimenzi problému a jedná se o počet argumentů účelové funkce. Tento parametr je tedy dán problémem, který řešíme.

## 5.2 Princip diferenciální evoluce

### 5.2.1 Stanovení parametrů

V prvním kroku stanovíme již zmíněné parametry, které určují chod celé evoluce. Musíme také nadefinovat prototyp jedince zvaný Specimen, která říká, z jakých typů čísel se budou jedinci skládat.

### 5.2.2 Tvorba populace

Populace se vytvoří podle Specimenu vygenerováním množiny jedinců. Musíme počítat u každého jedince s jedním prvkem navíc, který bude představovat hodnotu účelové funkce.

### 5.2.3 Započítí cyklu

Během každé generace se provádí ještě cyklus, který zabezpečuje postupné evoluční šlechtění každého jedince z populace. [10]

### 5.2.4 Evoluční cyklus

V této fázi je prováděna mutace a křížení.

### 5.2.5 Testování naplnění ukončovacích parametrů

Diferenciální evoluce je ukončena pouze tehdy, provede-li se uživatelem zadaný počet generací. [10]



### 5.2.6 Vyhodnocení

Během každého cyklu nebo-li generace se ukládá hodnota účelové funkce nejlepšího jedince.

## **II. PRAKTICKÁ ČÁST**

## 6 SOFTWARE MATHEMATICA

Pro zpracování vybraného problému byl zvolen software Mathematica.

### 6.1 Historie

Stephen Wolfram založil v roce 1987 společnost Wolfram Research, Inc. A již o rok později spatřil světlo světa program Mathematica. Říká se, že vydáním software Mathematica začal nový věk zvaný technical computing.

### 6.2 Mathematica

Program Mathematica integruje nástroje pro numerickou a symbolickou matematiku, grafický a dokumentační systém a zajišťuje pokročilé propojení s dalšími aplikacemi.

Klíčovou myšlenkou, která toto všechno zajistila, bylo objevení nového symbolického jazyka. Symbolické programování umožňuje reprezentovat data, funkce, grafy, programy, a dokonce i celý dokument jednotným způsobem, a to jako symbolický výraz. [5]

Mathematica využívá nezávislého interakčního dokumentu zvaného notebook. Notebookové rozhraní kombinuje textový procesor s buňkami. Buňky vizuálně a funkčně oddělují text na vstupy, výstupy, grafiku a jiné.

Mathematica notebook zajišťuje kompletní technický dokumentační systém zahrnující sazbu matematických výrazů, formátovaného textu, zvuku, grafiky, animací a hypertextových odkazů. [5]

Mathematica poskytuje silné programové vývojové prostředí. Program obsahuje interaktivní nápovědu s plnohodnotnou dokumentací pro všechny funkce softwaru Mathematica.

## 7 SYNCHRONIZACE LORENZOVÝCH ATRAKTORŮ

Jako první příklad pro synchronizaci byly vybrány dva Lorenzovy atraktory.

### 7.1 Synchronizace

#### 7.1.1 Věta o synchronizaci

Dva identické systémy mají identickou synchronizaci, jestliže atraktor leží na hyperploše, jejíž dimenze je striktně menší než nejvyšší dimenze fázového prostoru. Tato věta dokonce platí, i když nemáme stejné množství proměnných v každém systému.

#### 7.1.2 Způsob synchronizace

Přístup k synchronizaci systémů je následující. Máme dva identické systémy, v našem případě se jedná o Lorenzovy atraktory. Kompletně nahradíme jednu z proměnných v jednom systému, který budeme nazývat synchronizovaný systém, jeho protějškem v dalším systému, nazývaného řídicí systém.

Jak jména systému napovídají, propojení je jednosměrné a tento jednocestný přístup budeme nazývat úplně nahrazení.

Konkrétně nahradíme proměnnou  $x$  v synchronizovaném systému proměnnou  $x_2$  z řídicího systému. Získáme pětirozměrný dynamický systém.

## 7.2 Popis programu

### 7.2.1 Proměnná LorenzAttractor

Proměnná LorenzAttractor obsahuje šest diferenciálních rovnic, z nichž první tři rovnice představují Lorenzův atraktor, který zde má funkci synchronizovaného systému. Poslední

tři rovnice představují také Lorenzův atraktor. Tento systém je řídicí a konkrétně proměnná  $x_2$  z tohoto řídicího systému nahradila proměnnou  $x$  v systému synchronizovaném.

V této proměnné jsou nadefinovány počáteční podmínky pro  $x$ ,  $y$ ,  $z$  a pro  $x_2$ ,  $y_2$ ,  $z_2$ . Dále jsou zde nastaveny konstanty, jež Lorenzův atraktor obsahuje, a to konstanty  $a$ ,  $b$  pro synchronizovaný systém a konstanty  $a_2$ ,  $b_2$  v systému řídicím.

### 7.2.2 Proměnná NLA

Tato proměnná v sobě zahrnuje příkaz `NDSolve`. `NDSolve` nalezne numerické řešení rovnic `LorenzAttractor` pro funkce  $x$ ,  $y$ ,  $z$ ,  $x_2$ ,  $y_2$ ,  $z_2$  s nezávisle proměnnou  $t$  v rozmezí  $t_{\min} = 0$  až po  $t_{\max} = 20$ .

Příkaz `NDSolve` obsahuje jeden z možných parametrů, a to parametr `MaxSteps`. Tento parametr určuje maximální počet kroků. V tomto případě je `MaxSteps` nastaven na hodnotu 3000.

### 7.2.3 ParametricPlot3D

`ParametricPlot3D` vykresluje trojrozměrnou křivku v rozmezí  $t_{\min} = 0$  až po  $t_{\max} = 20$ . Každý z Lorenzových atraktorů je vykreslen pomocí samostatného příkazu `ParametricPlot3D`. První `ParametricPlot3D` je pro funkce  $x$ ,  $y$ ,  $z$  a druhý pro funkce  $x_2$ ,  $y_2$ ,  $z_2$ . Znak `/.` zaměňuje za funkce  $x[t]$ ,  $y[t]$ ,  $z[t]$  a funkce  $x_2[t]$ ,  $y_2[t]$ ,  $z_2[t]$  hodnoty vypočítané v proměnné NLA. K ohodnocení funkcí použijeme příkaz `Evaluate`.

`ParametricPlot3D` obsahuje další parametry, které určují charakter grafu. `PlotPoints` je parametr pro vykreslování funkcí. Určuje, kolik bodů pro vykreslování použijeme.

Parametr `TextStyle` určuje styl textu grafu. V našem případě obsahuje `FontWeight`, `FontColor`, `FontSize`. `FontWeight` je obdoba toho, čemu se ve Microsoft Word říká řez písma a

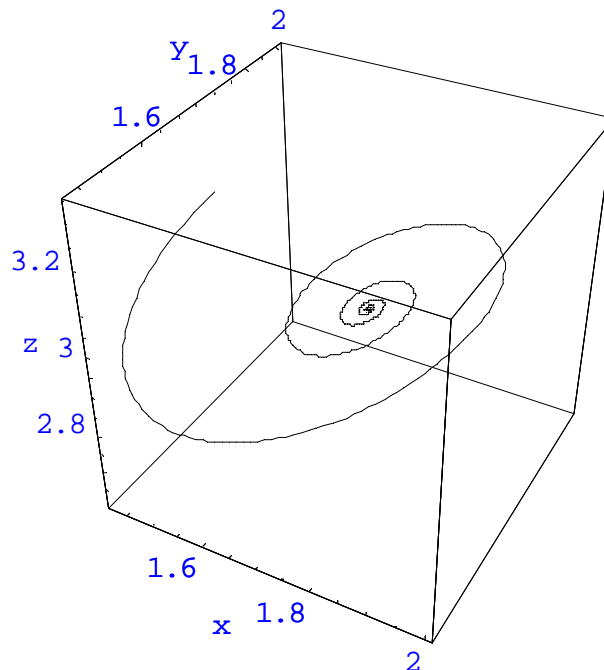
umožňuje například změnit obyčejné písmo na tučné. FontColor umožňuje na základně barevných modelů měnit barvu písma. V tomto případě je použit model RGBColor, který obsahuje složky red, green, blue. Každá z těchto složek nabývá hodnot 0-1 a optimálním nastavením a kombinací těchto složek dosáhneme požadované barvy písma. Software Mathematica nabízí k dispozici další barevné modely, a to Hue, GrayLevel, CMYKColor a ColorOutput. FontSize nám umožňuje měnit velikost písma.

ParametricPlot3D obsahuje i parametr AxesLabel, pomocí něhož nadefinujeme popisky os. Posledním parametrem v našem případě je BoxRatios. BoxRatios určuje poměr délek všech stran trojdimenzionálního grafu.

## 7.3 Zhodnocení výsledků

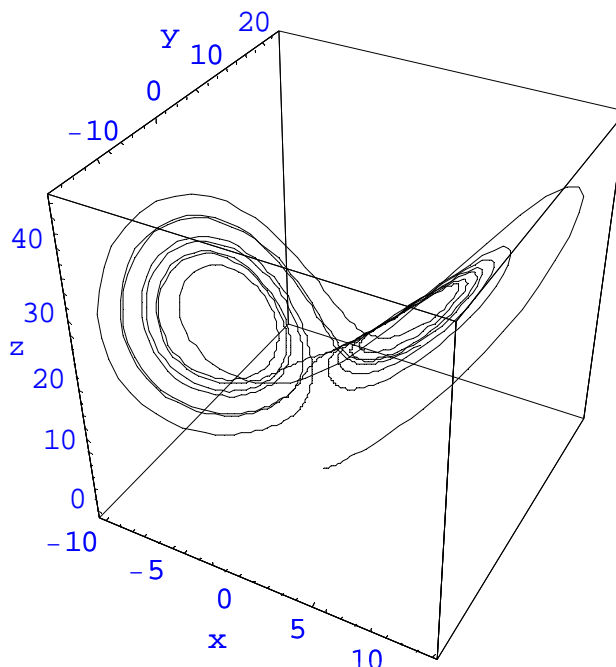
### 7.3.1 Systémy

Lorenzův atraktor pro konstanty  $a = 11$ ,  $b = 4$  generuje následující tvar atraktoru zobrazený na obrázku (Obr. 14). Nazvěme tento systém systémem A.



Obr. 20 Lorenzův atraktor pro  $a = 11$ ,  $b = 4$

Druhý Lorenzův systém generuje atraktor pro konstanty  $a = 3$ ,  $b = 26.5$ . Graf trajektorie vidíme na obrázku (Obr. 15). A tento systém budeme nazývat systémem B.

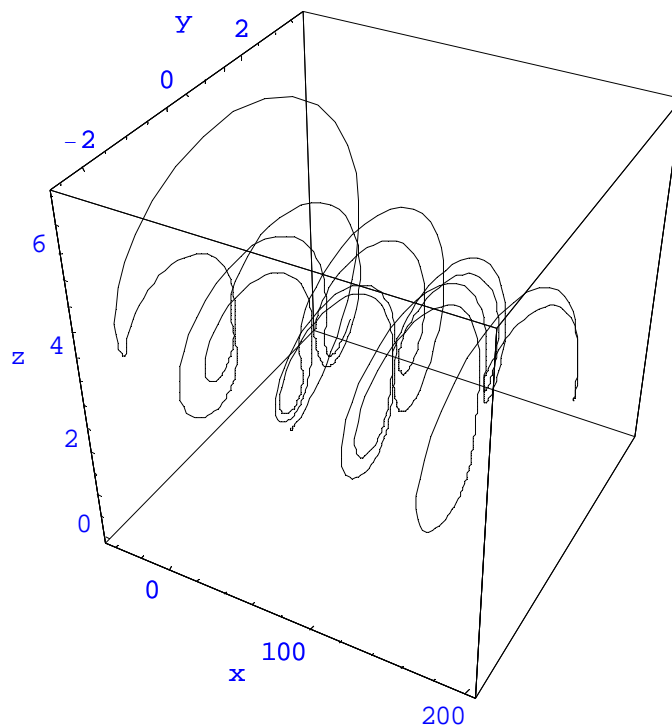


Obr. 21 Lorenzův atraktor pro  $a = 3$ ,  $b = 26,5$

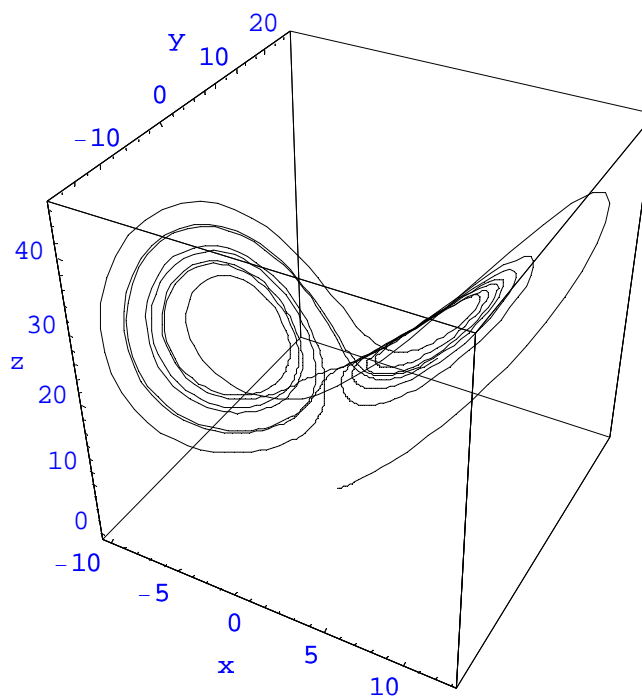
Nyní provedeme pomocí programu synchronizaci těchto dvou Lorenzových systémů.

### 7.3.2 Systém A synchronizovaný, systém B řídicí

Nejdříve nastavíme systém A jako synchronizovaný systém a systém B jako řídicí. Na obrázků (Obr. 16) vidíme jak se trajektorie systému A změnila a je značně ovlivněna řídicím systémem B. Zatímco řídicí systém zůstal nezměněn.



Obr. 22 Synchronizovaný systém A pro  $a = 11$ ,  
 $b = 4$

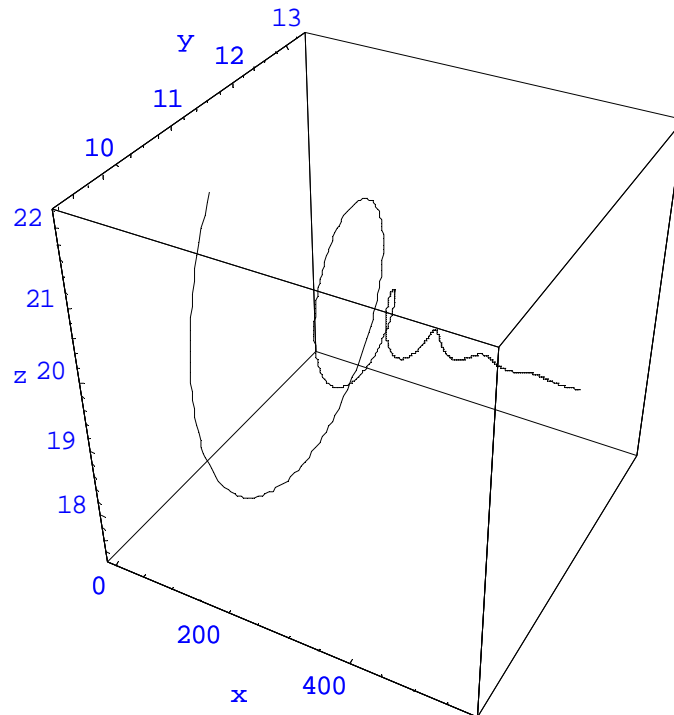


Obr. 23 Řídicí systém B pro  $a = 3$ ,  $b = 26,5$

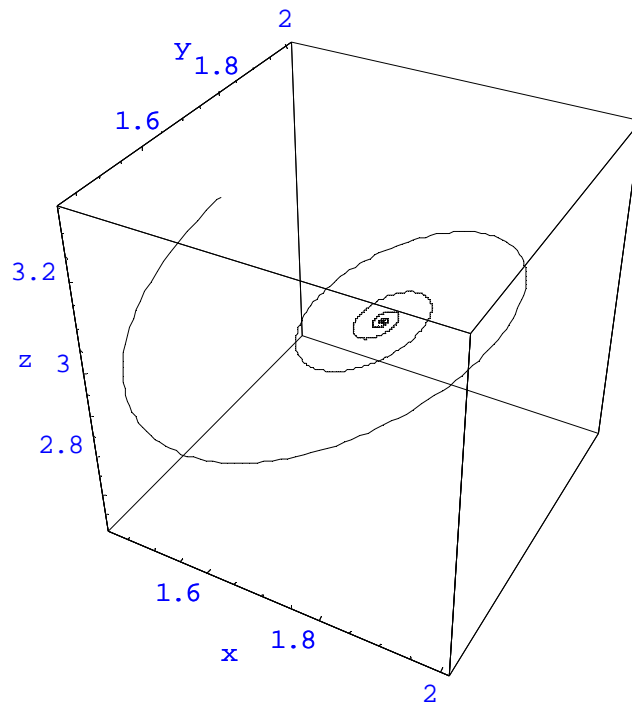


### 7.3.3 Systém A řídicí, systém B synchronizovaný

Nyní bude řídicím systémem systém A a jako synchronizovaný systém nám bude sloužit systém B.



Obr. 24 Synchronizovaný systém B pro  $a = 3$ ,  $b = 26,5$

Obr. 25 Řídicí systém A pro  $a = 11$ ,  $b = 4$ 

Jak vidíme na obrázku (Obr. 18) systém B, který nyní používáme jako systém synchronizovaný, je řízen trajektorií řídicího systému A. Řídicí systém, v tomto případě systém A, zůstal opět v nezměněné podobě.

## 8 SYNCHRONIZACE LORENZOVA A RÖSLEROVA ATRAKTORU

Druhá ukázka synchronizace bude provedena s Lorenzovým a Röslerovým atraktorem, kdy Röslerův atraktor bude řídit Lorenzův. Atraktory jsou řízeny přes parametr  $y$  s pomocí konstant. Pro nalezení optimálních hodnot konstant byly použity evoluční algoritmy SOMA a diferenciální evoluce.

Evoluční algoritmy byly použity z důvodu obtížnosti problému. Funkce vykazují nelinearitě a značné množství lokálních extrémů. Použitím evolučních algoritmů se tak vyhneme nepříjemnému faktu nalezení lokálního extrému. Tedy pomocí evolučních algoritmů nalezneme globální extrém.

### 8.1 Popis programu

#### 8.1.1 Proměnná LorenzAttractor

Tato proměnná obsahuje šestici rovnic. První tři rovnice představují Röslerův atraktor a zbylé tři rovnice znázorňují Lorenzův atraktor. Struktura rovnic Lorenzova systému má odlišný charakter, který je dán deterministickými postupy popsány v literatuře [21]. Přičemž parametr  $c$  je vazební konstanta.

#### 8.1.2 Proměnná NLA

Tato proměnná v sobě zahrnuje příkaz NDSolve. NDSolve nalezne numerické řešení rovnic obsažené v proměnné LorenzAttractor pro funkce  $x$ ,  $y$ ,  $z$ ,  $xx$ ,  $yy$ ,  $zz$  s nezávisle proměnnou  $t$  v rozmezí  $t_{\min} = 0$  až po  $t_{\max} = 200$ . Vazební konstantě  $c$  je přiřazena hodnota vypočítaná pomocí algoritmů SOMA a diferenciální evoluce. Tato hodnota je optimální.

Příkaz NDSolve obsahuje jeden z možných parametrů, a to parametr MaxSteps. Tento parametr určuje maximální počet kroků. V tomto případě je MaxSteps nastaven na hodnotu 30000.

### 8.1.3 ParametricPlot3D

ParametricPlot3D vykresluje trojrozměrnou křivku v rozmezí  $t_{\min} = 0$  až po  $t_{\max} = 200$ . Každý atraktor je vykreslen pomocí samostatného příkazu ParametricPlot3D. Přičemž první vykresluje Röslerův atraktor a druhý příkaz Lorenzův atraktor řízený Röslerovým atraktorem. První ParametricPlot3D je pro funkce  $x$ ,  $y$ ,  $z$  a druhý pro funkce  $xx$ ,  $yy$ ,  $zz$ . Znak  $/$  zaměňuje za funkce  $x[t]$ ,  $y[t]$ ,  $z[t]$  či za funkce  $xx[t]$ ,  $yy[t]$ ,  $zz[t]$  hodnoty vypočítané v proměnné NLA. K ohodnocení funkcí použijeme příkaz Evaluate.

ParametricPlot3D obsahuje další parametry, které určují charakter grafu. PlotPoints je parametr pro vykreslování funkcí. Určuje, kolik bodů pro vykreslování použijeme. V tomto případě je PlotPoints nastaven na hodnotu 10000.

Parametr TextStyle určuje styl textu grafu. V tomto případě obsahuje FontWeight, FontColor, FontSize, které jsou popsány v předchozí kapitole Synchronizace Lorenzových atraktorů.

ParametricPlot3D také obsahuje parametr AxesLabel, pomocí kterého nadefinujeme popisky os. Předposledním parametrem v tomto případě je BoxRatios, který určuje poměr délek všech stran trojdimenzionálního grafu. Poslední parametr je PlotRange a specifikuje, které body budou zakresleny do grafu. V našem případě je nastaven na stav All a jedná se tedy o všechny body.

### 8.1.4 Plot

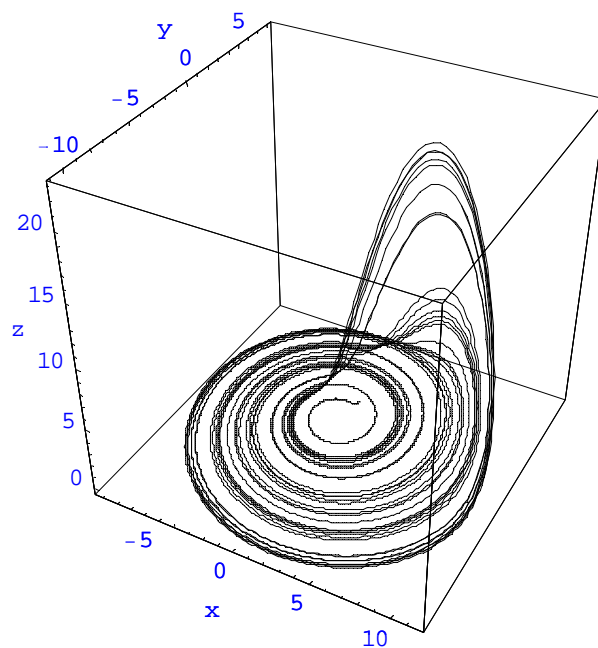
Parametr Plot vykresluje graf v 2D a znázorňuje synchronizaci dvou systémů podle jednoho parametru z každého systému. Plot obsahuje totožné parametry s ParametricPlot3D.

## 8.2 Pokus 1

V tomto pokusu byla vypočítána optimální hodnota pro vazební konstantu, a to pomocí algoritmu SOMA a diferenciální evoluce. Každý z algoritmů byl spuštěný stokrát. Hodnota vazební konstanty se pohybovala okolo 317. Zbylé konstanty byly nastaveny na hodnoty, které uvádí literatura [21] na hodnotu  $8/3$ .

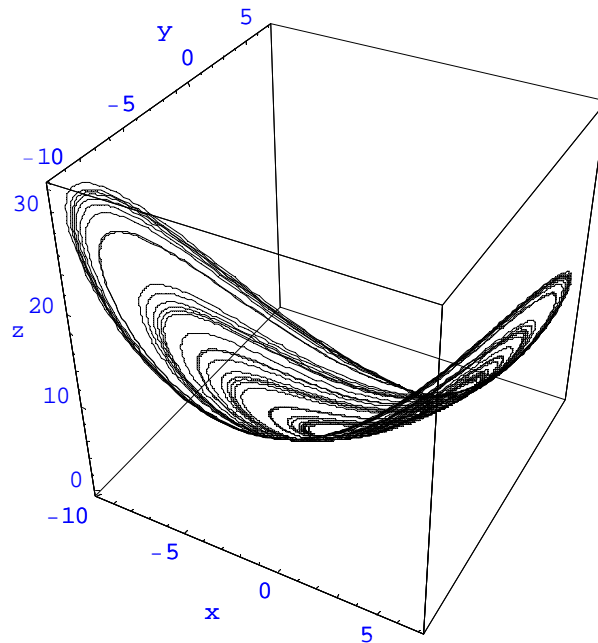
### 8.2.1 3D grafy

První ParametricPlot3D vykresluje Röslerův atraktor, který můžeme vidět na obrázku (Obr. 20).



Obr. 26 Röslerův atraktor

Na druhém vykresleném 3D grafu na obrázku (Obr. 21) je znázorněn Lorenzův atraktor, který je řízený Röslerovým atraktorem.

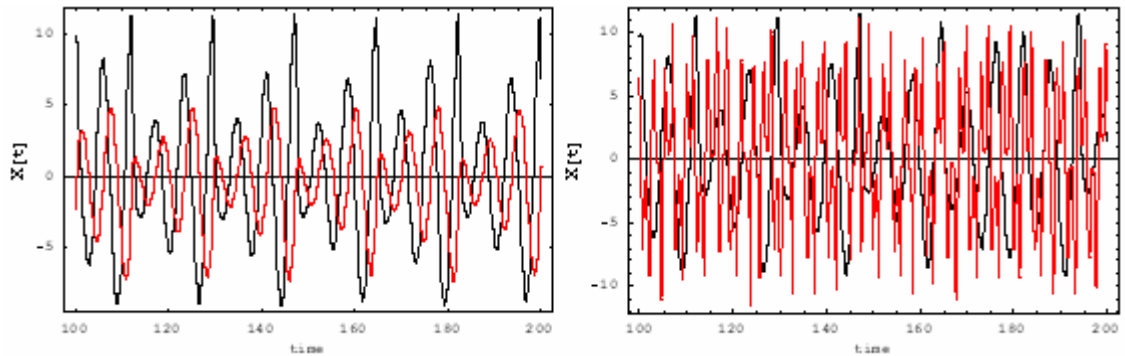


Obr. 27 Řízený Lorenzův atraktor

### 8.2.2 Grafy podle jedné proměnné systému

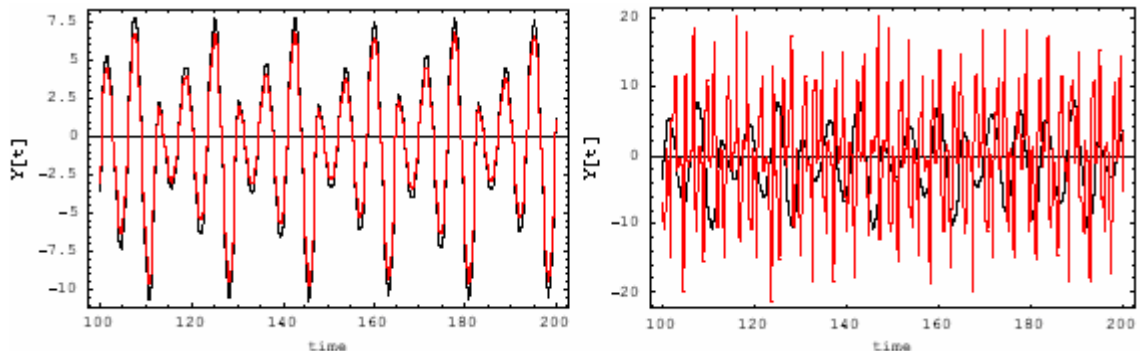
Na 2D grafech si ukážeme, jak probíhala synchronizace Röslerova atraktoru a Lorenzova atraktoru. Konkrétně Röslerův atraktor řídí Lorenzův. Dále se podíváme na to, jak vypadá průběh těchto systémů, když je nesynchronizujeme. Průběh bude vyjádřen jednou proměnnou z každého systému, tedy jednou proměnnou z Röslerova systému a jeho opakem v Lorenzově systému.

Na obrázku (Obr. 22) vlevo vidíme synchronizaci Lorenzova a Röslerova systému pro proměnnou  $x$  a  $xx$ . Křivka, která má červenou barvu, představuje Lorenzův systém. Druhá křivka černé barvy znázorňuje Röslerův systém. Křivky se vyznačují stejným průběhem, ale s menším fázovým posuvem, jelikož systémy jsou synchronizovány proměnnou  $y$ . Graf vpravo znázorňuje oba dva systémy, když nejsou synchronizované. Vidíme, že průběh každého systému je naprosto odlišný.



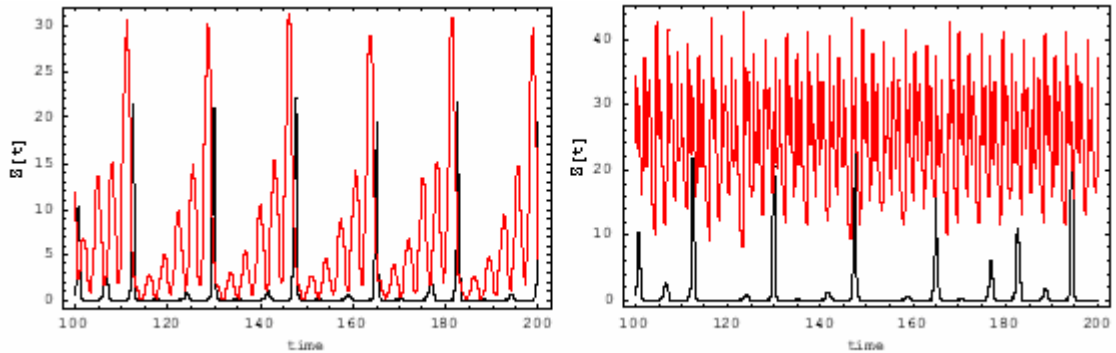
Obr. 28 Vlevo řízený systém pro  $x[t]$ ,  $xx[t]$ , vpravo neřízený systém pro  $x[t]$ ,  $xx[t]$

Obrázku (Obr. 23) vlevo je znázorněna synchronizace již zmiňovaných systémů, které jsou tentokrát sledované podle proměnných  $y$  a  $yy$ . Jelikož jsou systémy synchronizovány právě touto proměnnou, jejich průběh je naprosto totožný s nulovým fázovým posuvem. Vpravo jsou opět systémy nesynchronizované.



Obr. 29 Vlevo řízený systém pro  $y[t]$ ,  $yy[t]$ , vpravo neřízený systém pro  $y[t]$ ,  $yy[t]$

Na obrázku (Obr. 24) vlevo vidíme opět oba synchronizované systémy, tentokrát znázorněné podle  $z$  a  $zz$ . Z grafu vidíme, že průběh červené křivky, která představuje Lorenzův systém, se přibližuje průběhu Rösslerova systému. Naopak vpravo, kdy systémy nejsou synchronizované, se jejich průběh opět viditelně liší.



Obr. 30 Vlevo řízený systém pro  $z[t]$ ,  $zz[t]$ , vpravo neřízený systém pro  $z[t]$ ,  $zz[t]$

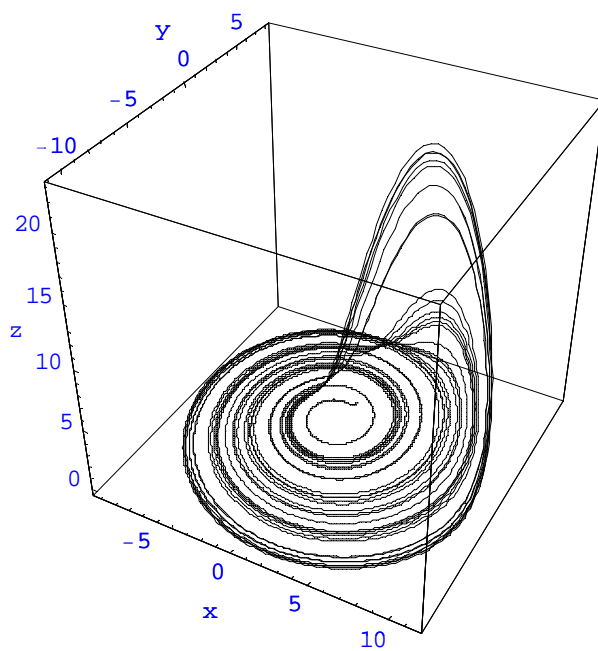
### 8.3 Pokus 2

V druhém pokusu opět řídil Röslerův atraktor Lorenzův atraktor. Nyní se provedly výpočty pro tři konstanty, a to pro konstantu  $a$ ,  $b$  a vazební konstantu  $c$ . Optimální hodnoty se hledaly podle algoritmu SOMA. Konkrétně byly nalezeny hodnoty pohybující se okolo těchto hodnot, pro  $a = 0,119$ ,  $b = 2,8$ ,  $c = 69,4$ . Každý z těchto dvou algoritmů byl spuštěn stokrát. Čtvrtá poslední konstanta byla nastavena podle údajů v literatuře [21] na hodnotu  $8/3$ .

#### 8.3.1 3D grafy

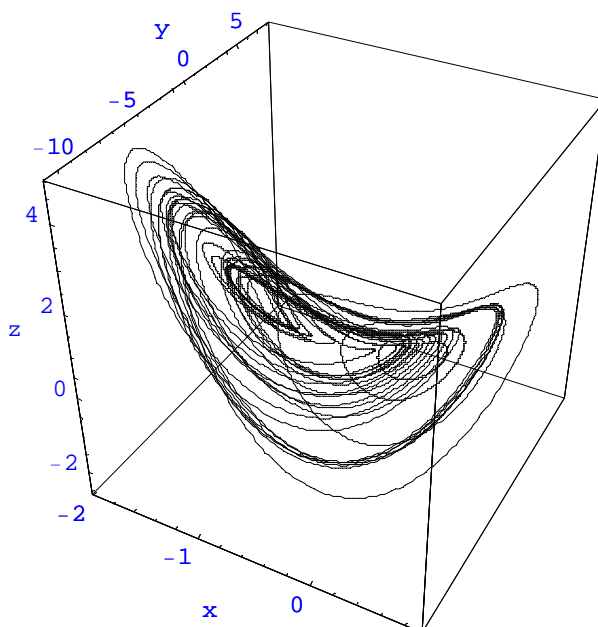
Na obrázku (Obr. 31) vidíme řídicí systém, a to Röslerův atraktor, kterým budeme řídit Lorenzův atraktor.





Obr. 31 Rösslerův atraktor

Řízený Lorenzův systém můžeme vidět na obrázku (Obr. 32).

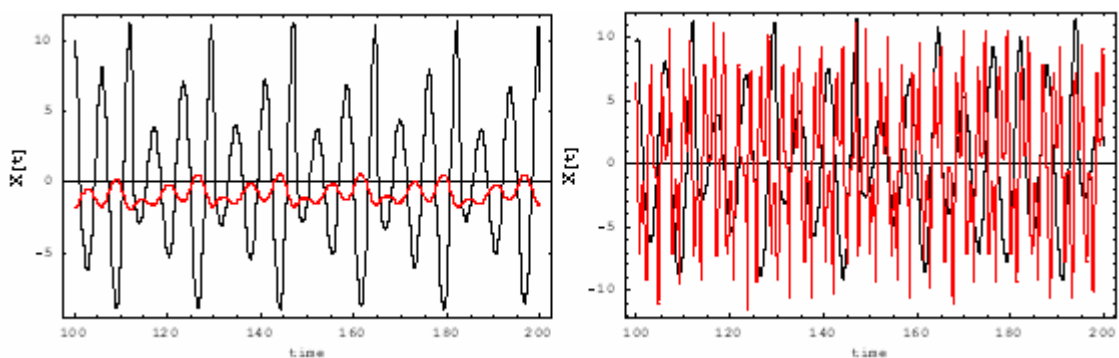


Obr. 32 Řízený Lorenzův systém

### 8.3.2 Grafy podle jedné proměnné systému

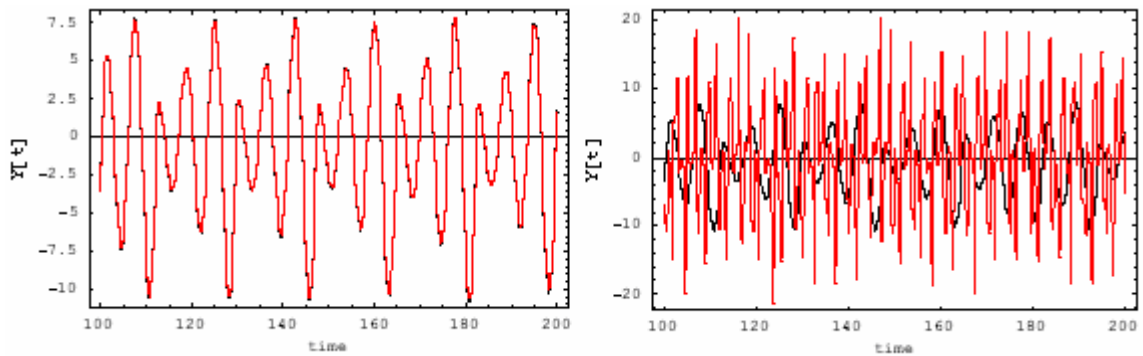
Nyní zhodnotíme průběh synchronizace podle jednoho parametru z každého systému, přičemž parametry představují svoje protějšky v systémech. Stejně tak se podíváme na to, jak vypadá průběh obou systémů, nejsou-li synchronizované. Jinými slovy, neřídí-li Röslerův systém Lorenzův.

Na obrázku (Obr. 33) vlevo vidíme průběh synchronizace podle parametru  $x$  z Röslerova systému, který je znázorněn křivkou černé barvy, a parametru  $xx$  z Lorenzova systému, jehož dráha vyvíjející se v čase je vykreslena křivkou barvy červené. Je patrné, že červená křivka představující Lorenzův synchronizovaný systém, se snaží přiblížit svým charakterem černé křivce patřící řídicímu systému. Graf vpravo představuje oba systémy ve stavu, kdy nejsou synchronizované. Z grafu vidíme, že trajektorie obou bodů jsou si svým průběhem značně vzdálené.



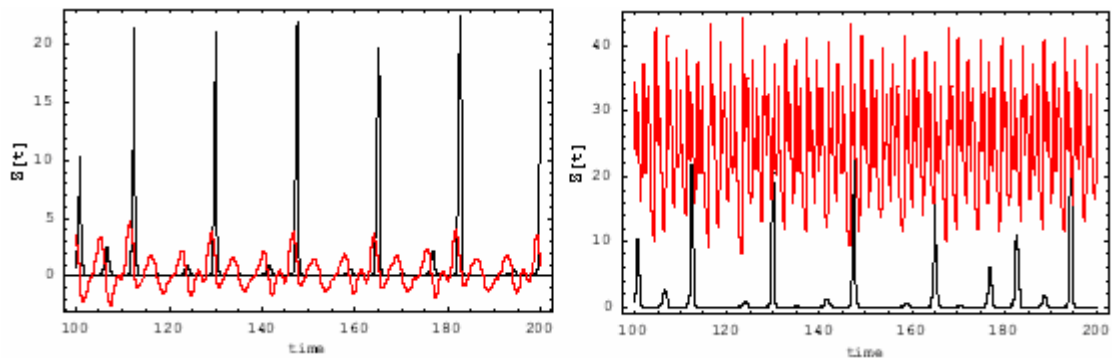
Obr. 33 Vlevo řízený systém pro  $x[t]$ ,  $xx[t]$ , vpravo neřízený systém pro  $x[t]$ ,  $xx[t]$

Obrázek (Obr. 34) vlevo představuje synchronizaci sledovanou podle parametru  $y$  Röslerova systému a parametru  $yy$  Lorenzova systému. Jelikož je Lorenzův systém řízen právě přes tento parametr, jsou křivky obou systémů naprosto totožné a nevykazují žádný fázový či amplitudový rozdíl. Graf vpravo opět znázorňuje oba systémy v nesynchronizovaném stavu.



Obr. 34 Vlevo řízený systém pro  $y[t]$ ,  $yy[t]$ , vpravo neřízený systém pro  $y[t]$ ,  $yy[t]$

Obrázek (Obr. 35) vpravo představuje synchronizaci systémů sledovanou pro parametr  $z$  a  $zz$ . Je patrné, že červená křivka synchronizovaného Lorenzova systému se snaží kopírovat trajektorii černé křivky, která představuje řídicí Röslerův systém. Vlevo opět vidíme oba systémy ve stavu, kdy nejsou synchronizované a jejich průběh není ničím ovlivňován.

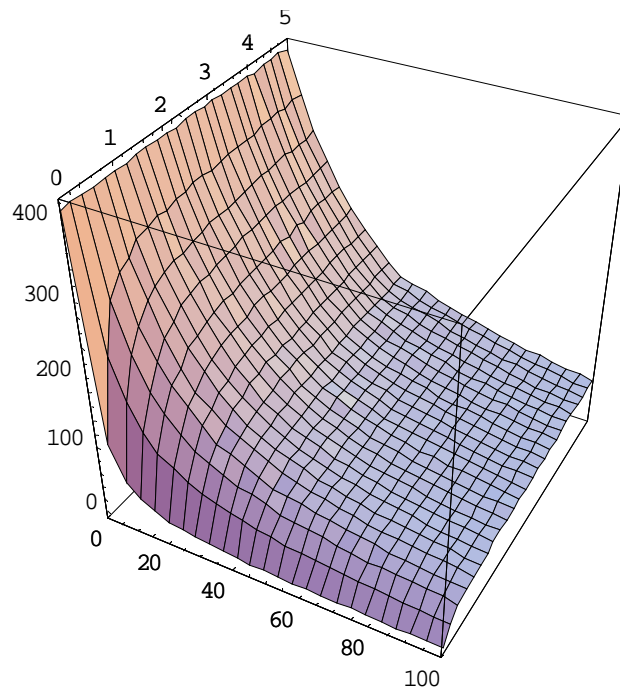


Obr. 35 Vlevo řízený systém pro  $z[t]$ ,  $zz[t]$ , vpravo neřízený systém pro  $z[t]$ ,  $zz[t]$

### 8.3.3 Účelová funkce pro jednotlivé řezy

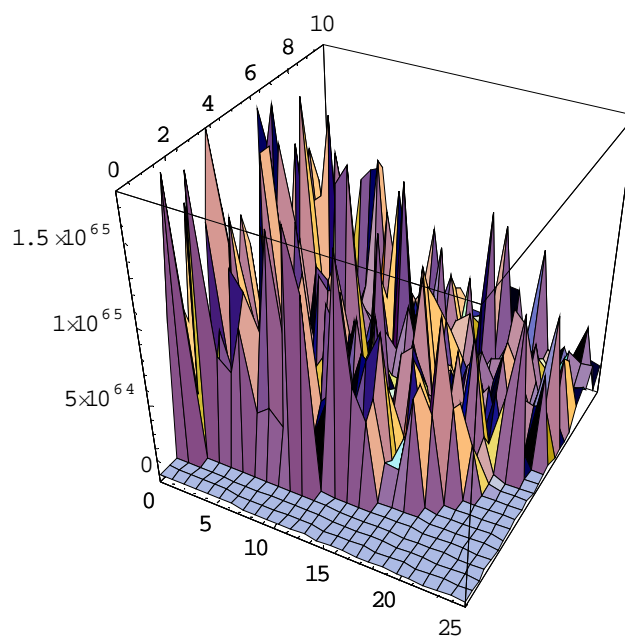
V této kapitole si ukážeme, jak vypadá účelová funkce pro jednotlivé řezy. Jelikož takovýto graf by byl čtyřrozměrný, použijeme tři třírozměrné grafy a pro každý takovýto graf bude jedna konstanta zmrazena nebo-li ji bude přiřazena pevná hodnota a průběh bude sledován podle dvou zbylých konstant.

Na obrázku (Obr. 36) sledujeme graf účelové funkce pro konstanty  $a$ ,  $c$ . Konstanta  $b$  je zmrazena. Tento trojrozměrný graf má charakter hladké plochy.



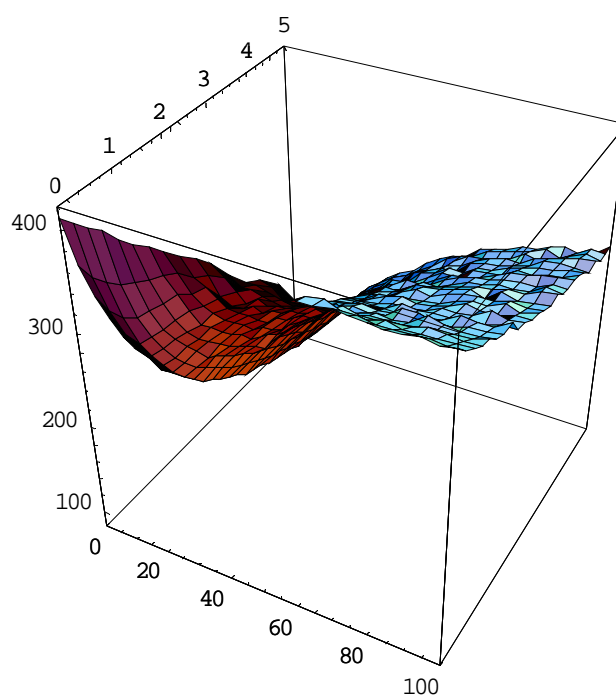
Obr. 36 Vývoj pro konstanty  $a$ ,  $c$

Obrázek (Obr. 37) znázorňuje graf pro konstanty  $a$ ,  $b$ . V tomto grafu je zmrazena konstanta  $c$ . Plocha svým charakterem připomíná pohoří a vykazuje značné množství lokálních extrémů.



Obr. 37 Vývoj pro konstanty a, b

Graf na obrázku (Obr. 38) znázorňuje účelovou funkci pro konstanty c, b. V tomto případě byla zmrazena konstanta a.



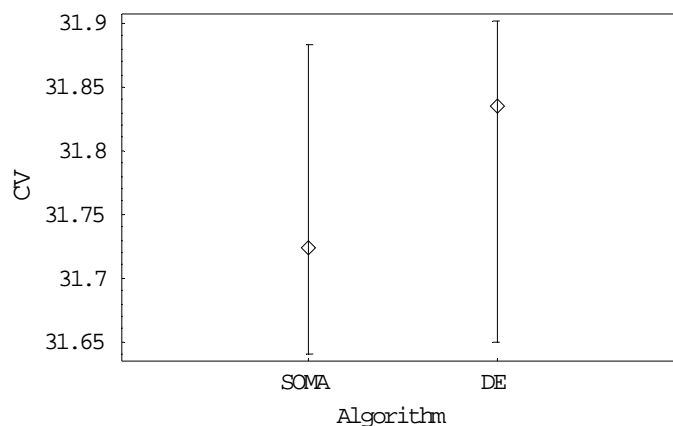
Obr. 38 Vývoj pro konstanty c, b

## 8.4 Zhodnocení vývoje evoluce

### 8.4.1 Pro pokus 1

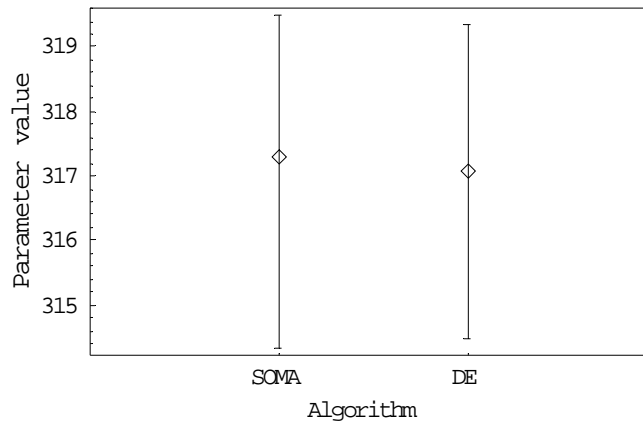
V pokusu 1 jsme řídili Lorenzův systém na základě Röslerova systému pomocí vazební konstanty  $c$  a přes parametr  $y$ . Optimální hodnotu vazební konstanty jsme našli pomocí evolučních algoritmů. Tyto algoritmy pracují ve smyslu ohodnocení účelové funkce, která představuje daný problém.

Na obrázku (Obr. 39) můžeme pozorovat vývoj evoluce z hlediska účelové funkce pro algoritmus SOMA a diferenciální evoluci. Okrajové body úseček představují dosažené minimum a maximum ve smyslu ohodnocení účelové funkce pro daný algoritmus. Můžeme zde pozorovat rozptyl hodnot, kterých účelová funkce dosáhla. Bod na úsečce, který je vyznačený prázdným kruhem, značí průměrnou hodnotu účelové funkce během evoluce.



Obr. 39 Vývoj evoluce z hlediska účelové funkce

Grafy na obrázku (Obr. 40) představují vývoj evoluce pro vazební konstantu  $c$  a pro příslušný algoritmus, pomocí kterého jsme nacházeli optimální hodnotu této konstanty. Krajní body úseček představují minimum a maximum pro konstantu  $c$ . Kruh ležící na úsečce představuje průměrnou hodnotu. Na úsečkách můžeme pozorovat rozptyl hodnot, které vazební konstanta dosáhla.



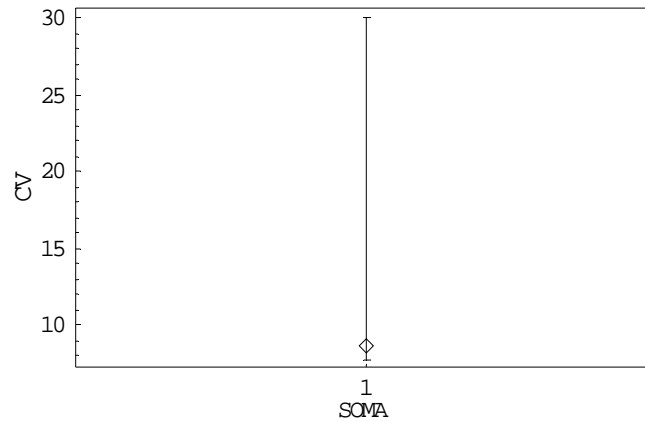
Obr. 40 Vývoj evoluce z hlediska vazební konstanty

c

#### 8.4.2 Pro pokus 2

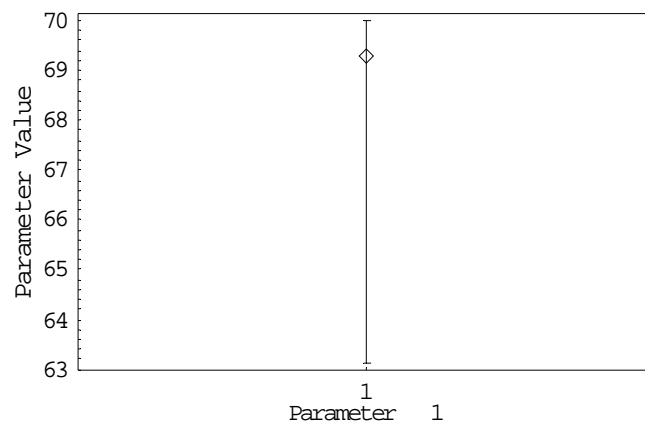
V pokusu s číslem dva jsme synchronizovali Lorenzův systém s řídicím Rösslerovým systémem. Synchronizace probíhala přes parametr  $y$  za pomoci tří konstant. Pro synchronizaci jsme použili algoritmus SOMA a hledali optimální hodnoty konstant. Tento algoritmus byl spuštěn stokrát a jeho vývoj můžeme sledovat na následujících grafem.

Na obrázku (Obr. 41) pozorujeme vývoj evoluce z hlediska hodnot účelové funkce pro evoluční algoritmus SOMA. Krajové body úsečky představují minimum a maximum, které účelová funkce dosáhla a kruh ležící na úsečce označuje průměrnou hodnotu účelové funkce, jež byla dosažena během sta evolucí.

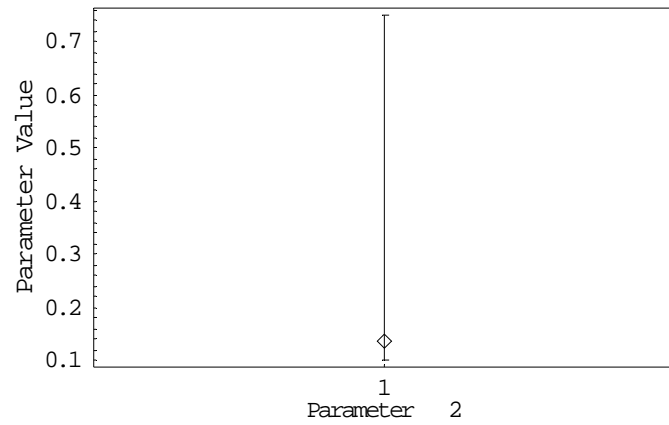


Obr. 41 Vývoj evoluce z hlediska účelové funkce

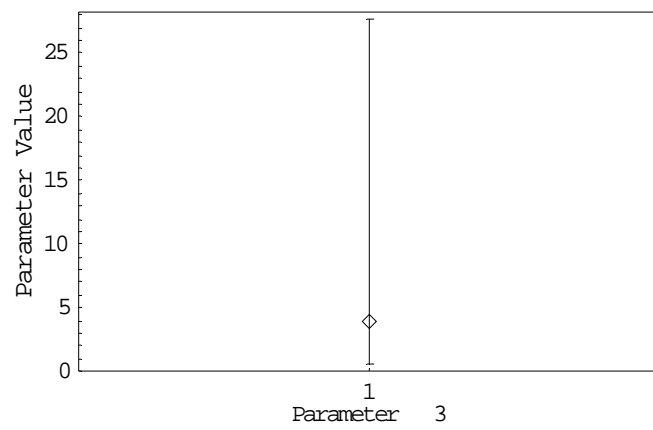
Na obrázcích (Obr. 42), (Obr. 43), (Obr. 44) pozorujeme vývoj evoluce z hlediska konstant. Konkrétně na obrázku (Obr. 42) sledujeme průběh pro konstantu  $c$ , obrázek (Obr. 43) patří konstantě  $a$ , a poslední obrázek (Obr. 44) vykresluje graf pro konstantu  $b$ . Okrajové body úseček představují minima a maxima pro dosažené hodnoty konstant a kruhy, které leží na úsečkách, jsou průměry dosažených hodnot během sta evolucí.

Obr. 42 Vývoj evoluce z hlediska konstanty  $c$





Obr. 43 Vývoj evoluce z hlediska konstanty a



Obr. 44 Vývoj evoluce z hlediska konstanty b

## 9 PŘENOS INFORMACE POMOCÍ ČASOPROSTOROVÉHO CHAOSU

Demonstrace přenosu informace pomocí časoprostorového chaosu je provedena v prostředí softwarového produktu Mathematica. Příloha (P I) obsahuje zdrojový kód včetně vygenerovaných grafů. Princip přenosu informace se odkazuje na literaturu [22]. Funkčnost programu je popsán v následujících řádcích

System CML, který je tvořený šedesáti nelineárně vázanými rovnicemi, generuje chaotickou sekvenci pomocí přesně stanoveného řídicího zásahu. Z této chaotické sekvence je vybrán každý  $n$ -tý kanál, v našem případě se jedná o každý třetí, tzn., že jsme získali 20 přenosových kanálů nebo-li nosičů informace.

Tento nosný chaotický signál je zdigitalizován. Po digitalizaci, tzn. po převodu do dvojkové soustavy, jsou nuly nahrazeny hodnotou -1. Na chaotický nosič je namodulována zdigitalizovaná informace, která je reprezentována jako sekvence -1 a 1. Modulace je realizována pomocí součinu  $G = M \cdot S$ , kde  $S$  je chaotická nosná složka a  $M$  představuje přenášenou informaci. Výsledkem je signál o stejné délce jako původní chaotický signál.

K přečtení zprávy je nutné znát velice přesně řídicí zásah (klíč), pomocí něhož jsme vygenerovali nosný signál  $S$ . Využitím korelační analýzy se z chaotického signálu vyjme užitečný signál nebo-li naše přenášená informace.

## ZÁVĚR

Oblast deterministického chaosu nabízí velice širokou a zajímavou škálu systémů pro synchronizaci a řízení, ať už se jedná o systémy se soustředěnými parametry, které vykazují časový chaos, nebo o systémy z rozprostřenými parametry, které produkují časoprostorový chaos.

Smysl řízení a synchronizace chaosu spočívá v usměrnění chaotického chování systému a jeho převedení do stabilní stavu. Nastává i opačný případ, kdy je žádoucí přivést systém do chaotického stavu.

Byly vybrány již řešené příklady a vypracováno alternativní řešení těchto příkladů v softwaru Mathematica. Použili jsme evoluční algoritmy pro nastavení optimálních hodnot konstant, které byly zapotřebí k synchronizaci systémů. Následně bylo provedeno zhodnocení dosažených výsledků. Z časových důvodů jsou v diplomové práci uvedeny pouze dva příklady synchronizovaných systémů a na nich provedeno několik pokusů.

Je pravděpodobné, že řízení a synchronizace chaosu naleznou v budoucnu větší uplatnění a bude se stále rozvíjet.

**SEZNAM POUŽITÉ LITERATURY**

Monografie:

- [1] ARROWSMITH, D. K., PLACE, C. M. *An Introduction to Dynamical Systems*. Cambridge University Press, 1900. ISBN 0-521-316-50
- [2] GLEICK, J. *Chaos*. Ando publishing, 1996. ISBN 80-86047-04-0
- [3] HILBORN, R.C. *Chaos and Nonlinear Dynamics*. Oxford University Press, 1994. ISBN 0-19-508816-8
- [4] HORÁK, J., KRLÍN, L. *Deterministický chaos*. Academia 1996. ISBN 80-200-0416-5, 1996
- [5] CHRAMCOV, Bronislav. *Základy práce v prostředí Mathematica*. Univerzita Tomáše Bati ve Zlíně, 2005. ISBN 80-7318-268-8
- [6] MAREK, M., SCHREIBER, I. *Chaotic Behaviour of Deterministic Dissipative Systems*. Academia. ISBN 80-200-0186-7
- [7] NICOLIS, G. *Introduction to Nonlinear Science*. Cambridge University Press, 1982. ISBN 0-521-46782-9
- [8] PALIS, J., TAKENS, F. *Hyperbolicity & sensitive chaotic dynamics at homoclinic bifurcations*. Cambridge University Press, 1993. ISBN 0-521-47572-4
- [9] PEITGEN, H.O., JURGENS, H., SAUPE, D. *Chaos and Fractals, New Frontiers of Science*. Springer-Verlag, 1992. ISBN 3-540-97903-4
- [10] ZELINKA, Ivan. *Umělá inteligence v problémech globální optimalizace*. Praha: BEN, 2003. ISBN 80-7300-068-7
- [11] ZELINKA, Ivan. *Umělá inteligence: Hrozba nebo naděje?*. Praha: BEN, 2002. ISBN 80-7300-069-5
- [12] ZELINKA, Ivan. *Základy informatiky*. Univerzita Tomáše Bati ve Zlíně, 2005. ISBN 80-7318-299-8

Internetové zdroje:

- [13] *Cantorova množina* [online]. [cit. 2006-15-03]. Dostupný z WWW:

[http://www.hajtmr.3web.cz/index.php?id\\_page=45](http://www.hajtmr.3web.cz/index.php?id_page=45)

- [14] *Edward Lorenz* [online]. [cit. 2006-05-02]. Dostupný z WWW:

[http://cs.wikipedia.org/wiki/Edward\\_Lorenz](http://cs.wikipedia.org/wiki/Edward_Lorenz)

- [15] *Mandelbrotova množina* [online]. [cit. 2006-30-04]. Dostupný z WWW:

[http://cs.wikipedia.org/wiki/Mandelbrotova\\_mno%C5%BEina](http://cs.wikipedia.org/wiki/Mandelbrotova_mno%C5%BEina)

- [16] *Lorenzův atraktor* [online]. [cit. 2006-04-03]. Dostupný z WWW:

[http://cs.wikipedia.org/wiki/Lorenz%C5%AFv\\_atraktor](http://cs.wikipedia.org/wiki/Lorenz%C5%AFv_atraktor)

- [17] TIŠNOVSKÝ, Pavel. *Fraktály v počítačové grafice III.* [online]. [cit. 2006-09-04].

Dostupný z WWW:

<http://www.root.cz/clanky/fraktaly-v-pocitacove-grafice-iii/>

- [18] VAŠÍČEK, Jiří. *Krátký úvod do teorie chaosu a jeho odraz v meteorologii* [online]. [cit. 2006-09-04]. Dostupný z WWW:

<http://www.pu.army.sk/view.php?cislocianku=2003110701>

- [19] ZELINKA, Ivan. *Historický nástin* [online]. [cit. 2006-12-03]. Dostupný z WWW:

<http://www.ft.utb.cz/people/zelinka/chaos/historie/historie.html>

- [20] ZELINKA, Ivan. *Řízení deterministického chaosu* [online]. [cit. 2006-21-04]. Dostupný z WWW:

<http://www.ft.utb.cz/people/zelinka/chaos/control/control.htm>

Část monografické publikace:

- [21] SCHUSTER, Heinz G. *Handbook of Chaos Control: Foundations and Applications (Hardcover)*. Wiley-VCH, 1999. Kapitola 10, Synchronization in Chaotic Systems, Concepts and Applications. ISBN 3527294368.
- [22] SCHUSTER, Heinz G. *Handbook of Chaos Control: Foundations and Applications (Hardcover)*. Wiley-VCH, 1999. Kapitola 3, Control of Patterns and Spatio-temporal Chaos and its Applications. ISBN 3527294368.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

CML coupled map lattices

CODE coupled ordinary differential equations

OGY Ott, Greboki, York

PDE partial differential equations

SOMA Samo-Organizující se Migrační Algoritmus

## SEZNAM OBRÁZKŮ

Obr. 1 Terry Pratchett .....	13
Obr. 2 Jules Henri Poincare .....	14
Obr. 3 Edward Norton Lorenz .....	15
Obr. 4 Lorenzův atraktor .....	16
Obr. 5 A. M. Ljapunov .....	17
Obr. 6 Georg F. L. P. Cantor .....	18
Obr. 7 Cantorova množina.....	19
Obr. 8 Oscar Rösler .....	20
Obr. 9 René Thom .....	22
Obr. 10 Feigenbaum .....	23
Obr. 11 Motýlí efekt .....	25
Obr. 12 Detail Mandelbrovy množiny, jednoho z nejznámějších fraktálů.....	31
Obr. 13 Graf trajektorie Lorenzova systému pro hodnoty $r = 28$ , $\sigma = 10$ , $b = 8/3$ .....	34
Obr. 14 Bifurkační diagram .....	35
Obr. 15 WEB diagram .....	36
Obr. 16 Mandelbrotova množina .....	38
Obr. 17 Katastrofa typu záhyb.....	40
Obr. 18 Katastrofa typu pyramida .....	41
Obr. 19 Bělousov – Žabotinského reakce.....	48
Obr. 20 Lorenzův atraktor pro $a = 11$ , $b = 4$ .....	62
Obr. 21 Lorenzův atraktor pro $a = 3$ , $b = 26,5$ .....	63
Obr. 22 Synchronizovaný systémém A pro $a = 11$ ,.....	64
Obr. 23 Řídící systémem B pro $a = 3$ , $b = 26,5$ .....	64
Obr. 24 Synchronizovaný systém B pro $a = 3$ , $b = 26,5$ .....	65
Obr. 25 Řídící systém A pro $a = 11$ , $b = 4$ .....	66
Obr. 26 Röslerův atraktor .....	69
Obr. 27 Řízený Lorenzův atraktor .....	70
Obr. 28 Vlevo řízený systém pro $x[t]$ , $xx[t]$ , vpravo neřízený systém pro $x[t]$ , $xx[t]$ .....	71
Obr. 29 Vlevo řízený systém pro $y[t]$ , $yy[t]$ , vpravo neřízený systém pro $y[t]$ , $yy[t]$ .....	71
Obr. 30 Vlevo řízený systém pro $z[t]$ , $zz[t]$ , vpravo neřízený systém pro $z[t]$ , $zz[t]$ .....	72
Obr. 31 Röslerův atraktor .....	73

Obr. 32 Řízený Lorenzův systém .....	73
Obr. 33 Vlevo řízený systém pro $x[t]$ , $xx[t]$ , vpravo neřízený systém pro $x[t]$ , $xx[t]$ .....	74
Obr. 34 Vlevo řízený systém pro $y[t]$ , $yy[t]$ , vpravo neřízený systém pro $y[t]$ , $yy[t]$ .....	75
Obr. 35 Vlevo řízený systém pro $z[t]$ , $zz[t]$ , vpravo neřízený systém pro $z[t]$ , $zz[t]$ .....	75
Obr. 36 Vývoj pro konstanty $a$ , $c$ .....	76
Obr. 37 Vývoj pro konstanty $a$ , $b$ .....	77
Obr. 38 Vývoj pro konstanty $c$ , $b$ .....	77
Obr. 39 Vývoj evoluce z hlediska účelové funkce .....	78
Obr. 40 Vývoj evoluce z hlediska vazební konstanty $c$ .....	79
Obr. 41 Vývoj evoluce z hlediska účelové funkce .....	80
Obr. 42 Vývoj evoluce z hlediska konstanty $c$ .....	80
Obr. 43 Vývoj evoluce z hlediska konstanty $a$ .....	81
Obr. 44 Vývoj evoluce z hlediska konstanty $b$ .....	81



## SEZNAM PŘÍLOH

- P 1: Zdrojový kód demonstrující přenos informace pomocí časoprostorového chaosu
- P 2: Přenosné médium CD-ROM

# PŘÍLOHA P I: ZDROJOVÝ KÓD DEMONSTRUJÍCÍ PŘENOS INFORMACE POMOCÍ ČASOPROSTOROVÉHO CHAOSU

## ■ parametry

`delkatransmise = 1000;`

`T = 50;`

`J = 10;`

`Signalu = 20;`

■ OCRML sn selvence

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

```
CompiledFunction[{x, A}, A x (1 - x), -CompiledCode-]
```

```
SPL = Compile[{{x, _Real, 1}, {ε, _Real}, {A, _Real}, {L, _Integer}},
```

```
  MapIndexed[
```

```
    If[#2[[1]] == 1, (1 - ε) Logistic[x[[#2[[1]]]], A] + ε Logistic[x[[L]], A],
```

```
      (1 - ε) Logistic[x[[#2[[1]]]], A] + ε Logistic[x[[#2[[1]] - 1]], A]
```

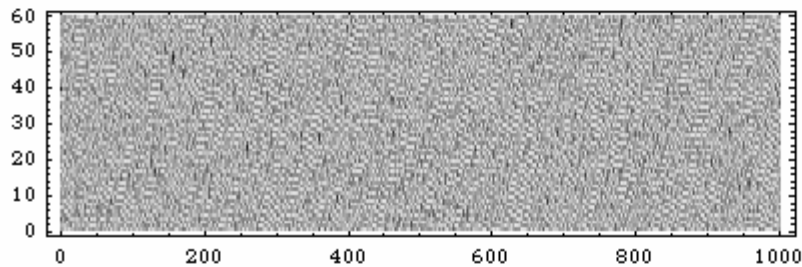
```
    ] &, x]
```

```
  ]:
```

```
xStart = Table[Random[], {i, 60}];
```

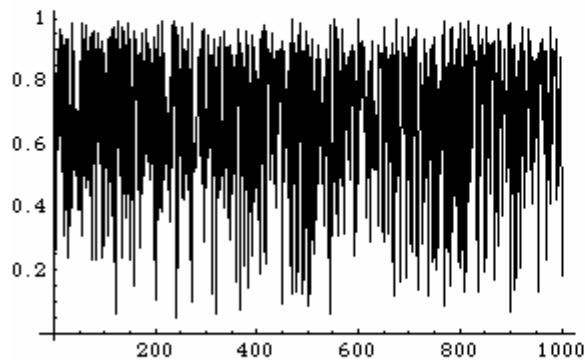
```
ocoml = NestList[SPL[#1, .8, 4, Dimensions[xStart][[1]]] &, xStart, delkatransmise];
```

```
ListDensityPlot[Transpose[ocoml], Mesh → False, AspectRatio → .3];
```



```
s_n = Take[Transpose[ocoml][[1]], delkatransmise];
```

```
ListPlot[s_n, PlotJoined → True]
```



- Graphics -

■ OCOML kanaly

```

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

SPL = Compile[{{x, _Real, 1}, {ε, _Real}, {A, _Real}, {L, _Integer}, {sn, _Real, 1},
  {idx, _Integer}},
  Flatten[{MapIndexed[
    If[#2[[1]] == 1, sn[[idx]],
      (1 - ε) Logistic[x[[#2[[1]]]], A] + ε Logistic[x[[#2[[1]] - 1]], A]
    ] &, x], idx + 1]
];

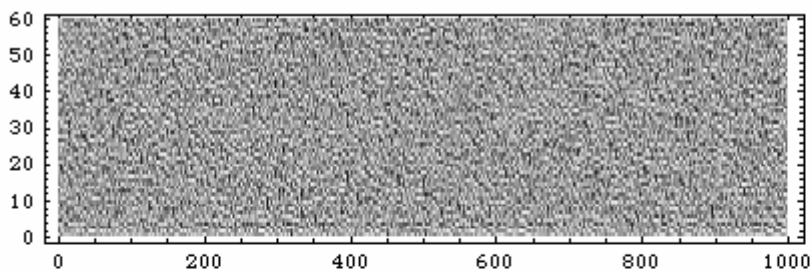
εε = .95;
AA = 4;

sites = 60;
x_start = Table[Random[], {i, sites}];

spch = NestList[SPL[Take[#1, 60], εε, AA, Dimensions[x_start][[1]], sn,
  Take[#1, -1]] &, Flatten[{x_start, 1}], delkatransmise];

ListDensityPlot[Take[Transpose[spch], 60, delkatransmise], Mesh → False,
  AspectRatio → .3];

```

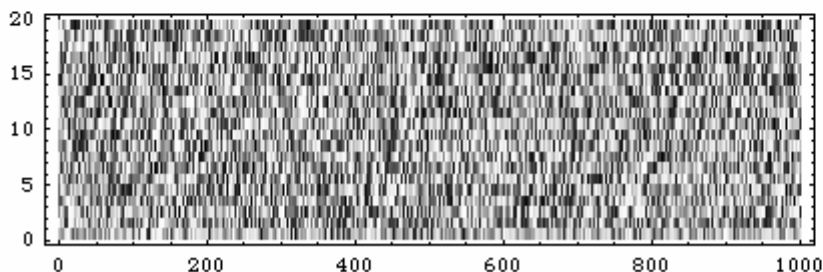


```

canal1 = Take[Transpose[spch], 60, delkatransmise];
canal2 = Table[canal1[[i]], {i, 1, 60, 3}];

ListDensityPlot[canal2, Mesh → False, AspectRatio → .3]

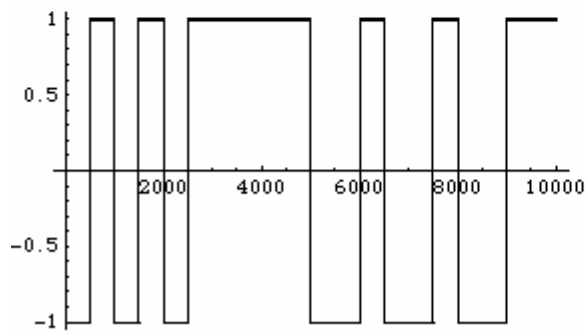
```



- DensityGraphics -

■ Signal

```
S = Flatten /@ ((RealDigits[#1, 2, J][[1]] & /@ #1 & /@ canal2) /. {0 → -1});  
MM = Table[Random[Integer, {0, 1}], {Signalu},  
  {i, Dimensions[S][[2]] / T / J} /. {0 → -1};  
M = Table[Flatten[Table[MM[[k, i]], {i, Dimensions[MM][[2]]}], {j, T J}],  
  {k, Signalu}];  
Dimensions /@ {M, S}  
{{20, 10000}, {20, 10000}}  
ListPlot[M[[1]], PlotJoined → True]
```

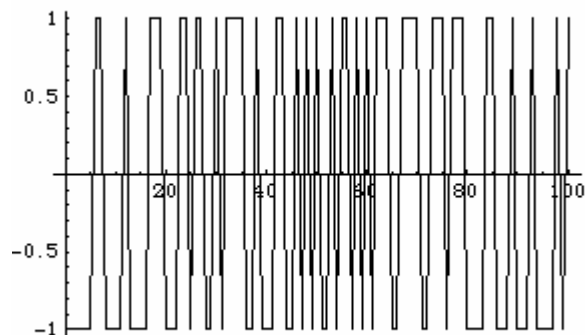


- Graphics -

■ Transmise 2D

G = M \* S;

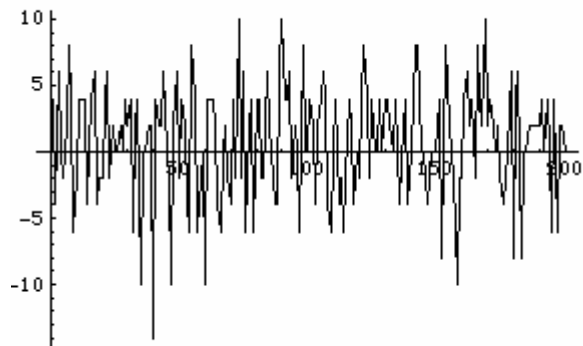
```
ListPlot[Take[G[[1]], 100], PlotJoined → True]
```



- Graphics -

Gtotal = Plus@@G;

```
ListPlot[Take[Gtotal, 200], PlotJoined → True]
```



- Graphics -

```
mx1 = Gtotal * #1 & /@ S;
```

```
Dimensions[#1] & /@ {S, Gtotal, mx1}
```

```
{(20, 10000), {10000}, {20, 10000}}
```

```
mpx = Partition[#1, J] & /@ mx1;
```

```
Dimensions[mpx]
```

```
{20, 1000, 10}
```

```
sumaj = ((Plus @@ #1 & /@ #1) & /@ mpx) // N;
```

```
Dimensions[sumaj]
```

```
{20, 1000}
```

```
sumaj1 = Partition[#1, T] & /@ sumaj;
```

```
Dimensions[sumaj1]
```

```
{20, 20, 50}
```

```
sumaN = ((Plus @@ #1 & /@ #1) & /@ sumaj1) / (T);
```

```
Mrek1 = Sign /@ sumaN;
```

```
Mrek1 - MM;
```

```
compare = Table[Mrek1[[i]] == MM[[i]], {i, 20}]
```

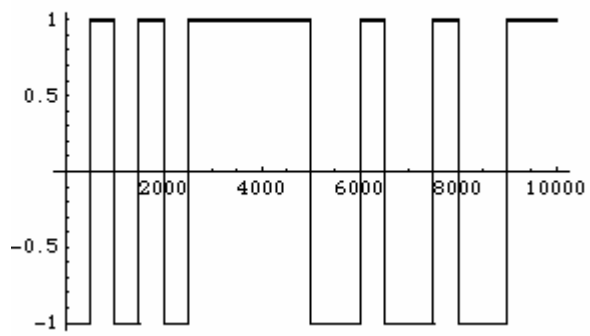
```
MMM = Table[Flatten[Table[Mrek1[[k, i]], {i, Dimensions[Mrek1][[2]]},  
{j, TJ}], {k, Signalu}];
```

```
Dimensions /@ {M, S}
```

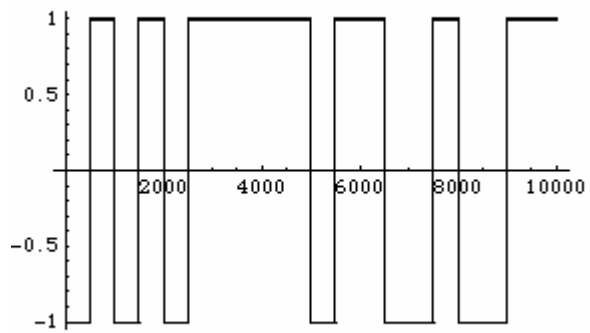
```
{{20, 10000}, {20, 10000}}
```

```
ListPlot[M[[1]], PlotJoined → True]
```

```
ListPlot[MMM[[1]], PlotJoined → True]
```



- Graphics -



- Graphics -

## **PŘÍLOHA P II: PŘENOSNÉ MÉDIUM CD-ROM**

Přenosné médium obsahuje diplomovou práci v pdf formátu a programy vytvořené v prostředí Mathematica.