

Abstrakt

Tato práce se zabývá popisem problematiky deterministického chaosu v programovém prostředí Matlab. Problematika deterministického chaosu přitahuje v poslední době stále více pozornost vědců a inženýrů na celém světě spolu s rozvojem výpočetní techniky, moderní teorie dynamických systémů a teorie řízení chaosu. V této práci byl v programovém prostředí Matlab proveden matematický popis deterministického chaosu. Popisovanými systémy byly jak jednodimenzionální mapy např.: logistická rovnice, kvadratická, sinusová a Gaussova mapa a další, tak i atraktory, jako např.: Lorenzův systém či Rosslerův systém. Dále byla vytvořena sada programů pro přehlednou grafickou vizualizaci deterministického chaosu.

Abstract

This work deals with the description of the problems of the deterministic chaos in the Matlab environment. The problems of deterministic chaos attract attention of the researchers and engineers all around the world together with the development of computers, modern theory of dynamical systems and theory of chaos control. The mathematic description of deterministic chaos in the Matlab environment was made in this work. The described systems were both one dimensional maps, for example: logistic equation, quadratic, sinus and Gauss map, and atractors, for example: Lorenz and Rossler. Further the set of programs for easy graphical visualization of deterministic chaos was made in this work.

Děkuji tímto panu Ing. Romanu Šenkeříkovi. za poskytnuté cenné praktické a teoretické rady, za odbornou pomoc a vedení při zpracování této diplomové práce

OBSAH

OBSAH	5
ÚVOD.....	6
I. TEORETICKÁ ČÁST	7
1 VZNIK CHAOSU – OSOBNOSTI V HISTORII CHAOSU.....	8
2 DETERMINISTICKÝ CHAOS – VÝSKYT	11
3 PROJEVY DETERMINISTICKÉHO CHAOSU	13
3.1 NEJEDNODUŠŠÍ MODEL DETERMINISTICKÉHO CHAOSU - LOGISTICKÁ ROVNICE	13
3.2 BIFURKAČNÍ DIAGRAM	13
4 STAVOVÝ PROSTOR.....	15
5 ATRAKTOR.....	16
II. PRAKTICKÁ ČÁST.....	18
6 VIZUALIZAČNÍ PROSTŘEDÍ MALTAB	19
6.1 PROČ PROSTŘEDÍ MALTAB?	19
6.2 POPIS PROGRAMU PRO VIZUALIZACI DETERMINISTICKÉHO CHAOSU.....	19
6.2.1 Hlavní okno programu	19
6.2.2 Jednodimenzionální (jedno-rozměrové) mapy.....	21
6.2.3 Atraktory	34
ZÁVĚR.....	38
SEZNAM POUŽITÉ LITERATURY.....	39
SEZNAM OBRÁZKŮ	40

ÚVOD

Na pojem chaos můžeme nahlížet z několika úhlů. Jako první se mnoha lidem při vyslovení tohoto slova vybaví chaos, jakožto stav neuspořádanosti, leckdy i temnoty, stav kde vládne nepředvídatelnost, neuchopitelnost jakýchkoli struktur, stav blízký nicotě, stav neexistence řádu.

Chaotické chování nás obklopuje celý život, avšak jen málo lidí ví, že toto leckdy zdánlivě chaotické chování (větrné turbulence, kouř cigarety) skrývá řád. Proto vznikla tato práce, která slouží k objasnění zdánlivě chaotického chování.

Chaos je disciplínou, která dostala své jméno až ve dvacátém století, nicméně jeho kořeny sahají až do 19. století, kdy bylo zjištěno, že už jednoduché problémy generují velmi složité a nepředpověditelné chování.

V poslední době vznikl v souvislosti s deterministickým chaosem směr zvaný „řízení deterministického chaosu“. Je to relativně nový vědecký interdisciplinární směr, v němž dochází k symbióze více oborů jako je např. fyzika, chemie, biologie či elektronika. To vše pak zastřešuje kybernetika, která využívá poznatků přispívajících oborů k syntéze toho, čemu se říká „řízení deterministického chaosu“.

Hlavním úkolem této práce je přiblížení problematiky deterministického chaosu čtenáři v programovém prostředí Matlab, kde může najít poměrně přehlednou vizualizaci matematického popisu chaotického chování. Ještě před samotným předváděním příkladů chaotického chování se v teoretické části práce zaměřuje na objasnění principů a závislostí chaotického chování včetně její historie. V teoretické části jsou jen obecně nastíněny některé náležitosti deterministického chaosu, jako například stavový prostor, logistická rovnice, bifurkační diagram, atraktory. Poměrně zajímavější pro čtenáře je praktická část této práce, kde jsou názorné ukázky deterministického chaosu uvedeny ve vizuální podobě.

V úvodu praktické části je popis programu pro vizualizaci deterministického chaosu v programovém prostředí Matlab. Tento popis je určen pro čtenáře, kteří by si chtěli vyzkoušet více příkladů chování deterministického chaosu v tomto programu.

V praktické části se práce zaměřuje hlavně na chování logistické rovnice. Jsou zde popsány různé průběhy a chování logistické rovnice pro rozličné počáteční podmínky a nastavení parametrů.

I. TEORETICKÁ ČÁST

1 VZNIK CHAOSU – OSOBNOSTI V HISTORII CHAOSU

Dvacáté století je století v němž prošlo lidstvo mnoha revolučními změnami a to jak v oblasti sociální, tak vědecké. Ve vědě došlo k mnoha dramatickým zvratům, které se promítly i do každodenního života běžného občana. Jedním z těchto zvratů je i vznik „nové vědy“ populárně zvané „Chaos“.

Jedním z lidí, kteří se podíleli na jejím vzniku byl ruský matematik A. M. Ljapunov. Jeho jméno je spojeno s chaosem přes tzv. Ljapunovovy exponenty. Tyto exponenty, obvykle označované jako „ λ “, vyjadřují u chaotických systémů divergenci nebo-li rozbíhavost blízkých trajektorií. Tato rozbíhavost je jedním z charakteristických rysů chaosu.

Dalším z řady významných lidí byl německý matematik George Cantor. I když jeho objevy patří spíše do oblasti fraktální geometrie je velmi často spojován s chaosem. G. Cantor je objevitel tzv. Cantorovy množiny, která byla poprvé publikována v roce 1883. Za otce chaosu se považuje až E. N. Lorenz, který se proslavil objevem tzv. "Motýlího efektu", který prezentoval v 60. letech. Motýlí efekt poukazuje na nemožnost dlouhodobé předpovědi počasí (max. 2-3 dny) a tím na jeho chaotičnost. Základní "myšlenkou" motýlího efektu je, zda nepatrné mávnutí motýlích křídel nad Tokiem může způsobit uragán nad New Yorkem. Lorenzův atraktor je spolu s Ljapunovovým exponentem, symbolem pro chaos. Díky jeho práci se zrodil termín "citlivost na počáteční podmínky". Lorenz tak poukázal na to, že mnohé systémy jsou citlivé na počáteční podmínky a že mohou díky tomu generovat to, čemu se dnes říká chaos.

Dalším z významných objevů byl Rosslerův atraktor. Je to umělý systém generující podivný atraktor, založený na nejjednodušších principech generace chaosu, a to transformaci "rozprostření a ohyb". Rosslerův atraktor (1976) vychází z atraktoru Lorenzova. Rosslerovo jméno začalo být spojováno s jednoduchým atraktorem ve tvaru stuhy, svinuté do věnečku se záhybem. Rossler si ale také představoval atraktory z vyšších dimenzí jako ohýbání a stlačování stavového prostoru, které se skutečně stalo klíčem ke konstrukci podivných atraktorů. Rossler zastával názor, že tyto tvary představují princip samoorganizace v přírodě (vznik disipativních struktur díky složitým a vzájemně souvisejícím procesům).

Van der Pola lze též považovat za jednoho z otců moderní experimentální dynamiky, kterou inicioval v období 1920 – 1930 svými pracemi na elektrických

obvodech jenž obsahovaly elektronky. Během těchto prací objevil stabilní oscilace, které jsou dnes nazývány „limitní cykly“. V roce 1927 Van der Pol spolu se svým kolegou Van den Markem v časopise Nature článek o jevu, jenž se zabýval šumem objevujícím se mezi jistými jasně definovanými frekvencemi. Jak z popisu jevu, tak z rekonstrukce jeho obvodu plyne, že za objevitele deterministického chaosu lze považovat už Van der Pola a ne E. N. Lorenze, jak to bývá často uváděno. Jejich článek je samozřejmě jeden z prvních, který pojednává o deterministickém chaosu.

Van der Pol postavil mnoho elektronických modelů lidského srdce za účelem studia srdeční dynamiky a stability. Jeho výzkum byl zaměřen na „řízení“ srdeční činnosti externím signálem, což je dnes záležitost kardio-stimulátorů. Jeho cílem bylo najít metodu jak stabilizovat srdeční arytmiie.

Lev Davidovič - jeho práce pokrývaly všechny oblasti tehdejší teoretické fyziky od mechaniky kapalin až po kvantovou teorii pole. L. D. Landau rovněž přispěl ke studiu chaosu kde se zabýval studiemi turbulence.

Jules Henri Poincare byl jednou z prvních osobností, která přispěla k teorii chaosu. I když žil ještě před oficiálním objevením deterministického chaosu, přispěl k jeho současnému studiu některými myšlenkami. Snad nejužívanějším výrazem je tzv. Poincarého plocha (rovina, řez, sekce), která slouží k vizualizaci chování mnohorozměrových dynamických systémů. J. H. Poincare se též velmi intenzivně zabýval studiem pohybu tří těles ve vzájemné gravitační interakci. Své poznatky získané v tomto směru zveřejnil v práci „On The Problem of Three Bodies and the Equations of Equilibrium“. Při práci na tomto problému objevil Poincare komplexitu chování tohoto, na první pohled jednoduchého systému. Problém tří těles je jedním ze základních demonstrací deterministického chaosu. S teorií chaosu dost úzce souvisí i tzv. teorie katastrof, která byla vyvinuta francouzským matematikem Rene Thomem v šedesátých letech 20. století.

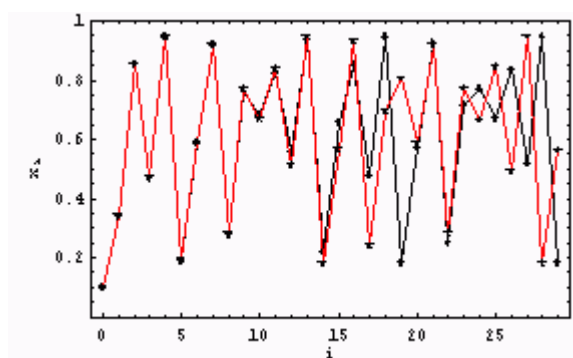
Michal Feigenbaum je další osobnost, která přispěla významnou měrou k teorii chaosu. Jeho největší přínos v teorii chaosu spočívá ve faktu, že objevil konstanty (nesoucí jeho jméno), které jsou nezávislé na generujícím systému a poukazují na jistou univerzálnost chaosu. Tyto konstanty jsou velmi dobře patrné z tzv. logistické rovnice, která je často používána k demonstraci koexistence druhů typu predátor-kořist.

Andrei Nikolajevič Kolmogorov je další z dlouhé řady významných osobností, jenž přispěly nejen k teorii chaosu. Zabýval se problémy jako je např. pohyb planet, turbulence,

dynamické systémy apod. V roce 1953-54 publikoval dva příspěvky z oblasti dynamických systémů, které se staly neoficiálně zakládajícími příspěvky tzv. KAM (Kolmogorov-Arnold-Moser) teorie z oblasti tzv. Hamiltoniánských systémů a samozřejmě i jejich chaotického chování. [1]

2 DETERMINISTICKÝ CHAOS – VÝSKYT

S deterministickým chaosem se lze setkat téměř na každém kroku. V podstatě se dá jednoduše říci, že stačí si jen najít jakýkoliv dynamický systém, který je velmi citlivý na počáteční podmínky – tzv. Motýlí efekt [2]. Tedy mávnutí křídylek motýla například ve Zlíně může způsobit po dostatečně dlouhé době (1 týden) úplně jiný vývoj počasí na jiném místě na zemi, než který by nastal, pokud by nemávnul. Citlivost na počáteční podmínky je názorně zobrazena na Obr. 1.



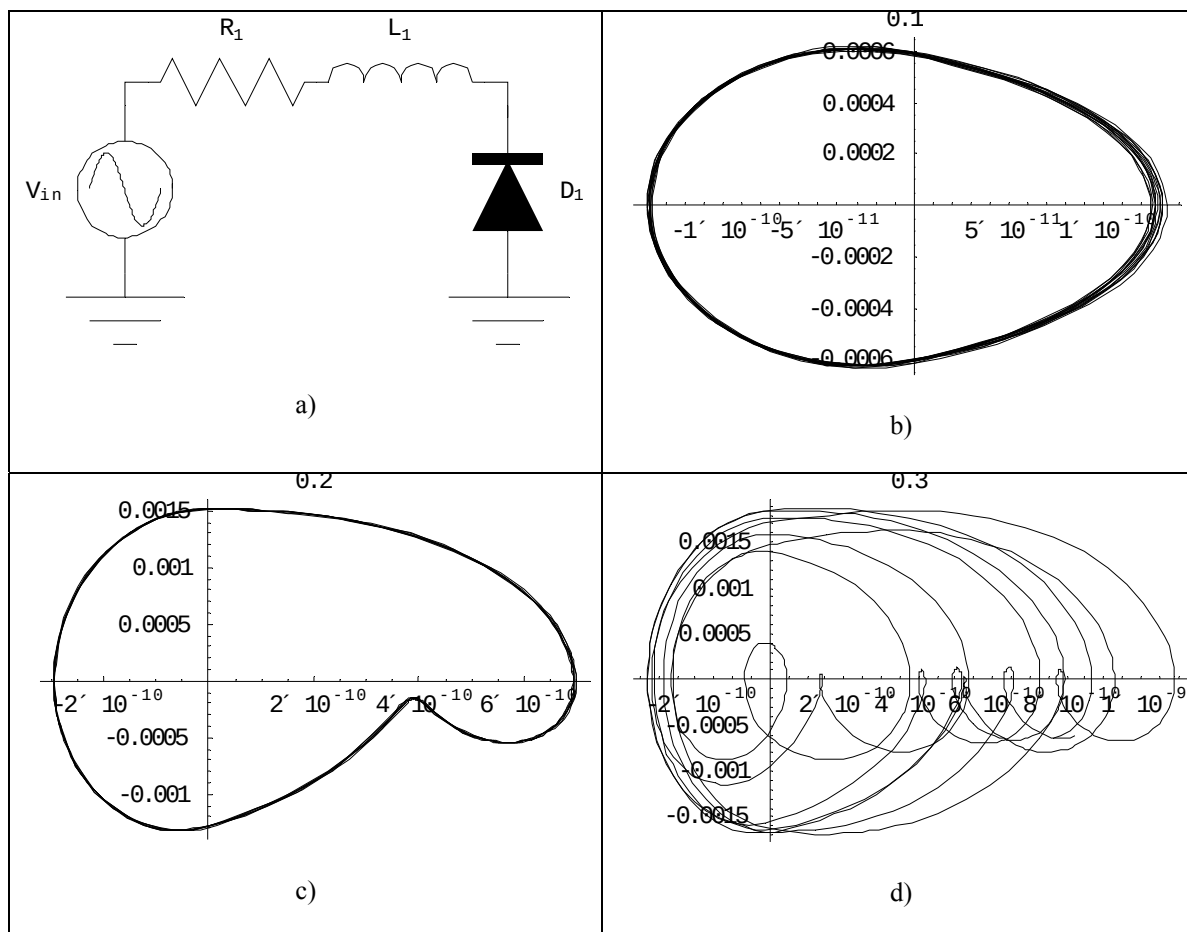
Obr. 1 Ukázka citlivosti na poč. podmínky.

Tedy typickým příkladem chaotického systému je počasí. Jeho základním typickým modelem je soustava tří diferenciálních rovnic (1), která generuje známý Lorenzův atraktor. [3]

$$\begin{aligned}
 \dot{x}(t) &= -3(x(t) - y(t)) \\
 \dot{y}(t) &= -x(t)z(t) + 26.5x(t) - y(t) \\
 \dot{z}(t) &= x(t)y(t) - z(t)
 \end{aligned}
 \tag{1}$$

Velmi často se pak lze s chaosem setkat v mnoha elektronických obvodech, kde ke vzniku chaosu stačí v podstatě jen pár součástek. Jeden z velmi jednoduchých chaotických obvodů je zobrazen na Obr. 2. V části a) je vlastní schéma a v části b)-d) je změna chování daného obvodu v závislosti na změnách řídicího napětí.

Dalšími příklady častého výskytu deterministického chaosu jsou plynové lasery, speciální mechanické soustavy a oscilátory [4].



Obr. 2 Jednoduchý elektronický obvod generující chaos.

Výskyt chaosu lze jednoduše očekávat a pozorovat 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, nebo že systém se nalézá v nevyhovujícím stavu, mnohdy je i využití deterministického chaosu prospěšné pro rychlou a energeticky nenáročnou změnu jeho stavu k lepšímu.

3 PROJEVY DETERMINISTICKÉHO CHAOSU

3.1 Nejjednodušší model deterministického chaosu - logistická rovnice

Jeden z velmi jednoduchých modelů chaotického chování je tzv. logistická rovnice (2). Tato rovnice vznikla jako odezva na potřebu simulace biologických systémů, v jejichž rámci popisuje chování druhu v jeho přirozeném prostředí. Tento popis byl založen na existenci nějakého druhu v "uzavřeném" prostředí (většina druhů má omezené migrační schopnosti), které mu poskytovalo obživu. Nejjednodušším příkladem jsou např. kapři a štiky v rybníku. Pokud se do rybníku nasadí určité množství kaprů a štik, pak samozřejmě dojde k jejich množení a růstu. Ten bude mít nároky na kapacitu potravy (kaprů) v rybníku. Při rostoucím počtu jedinců bude klesat množství potravy v rybníku, což se projeví zpomalením růstu počtu kaprů. Od jisté hranice pro nedostatek potravy začnou štiky vymírat hladu, zatímco díky jejich poklesu se množství kaprů začne zvyšovat. To od určité hranice množství kaprů způsobí jejich opětovný nárůst. Prostým citem tedy lze očekávat, že populace bude asi "periodicky" oscilovat, nebo se ustálí na nějaké hodnotě. Jak již jednoduché simulace ukázaly, může systém popsáný touto rovnicí vykazovat velmi komplikované chování od ustáleného přes periodické až po chaotické. Výzkum logistické rovnice prováděl biolog Robert May. Svůj výzkum vykonával May jen pomocí kalkulačky. Bylo to velmi zdlouhavé, ale i přesto se mu podařilo odhalit všechny důležité vlastnosti na vyvíjející se populaci hmyzu [5].

$$x_{n+1} = rx_n(1 - x_n) \quad (2)$$

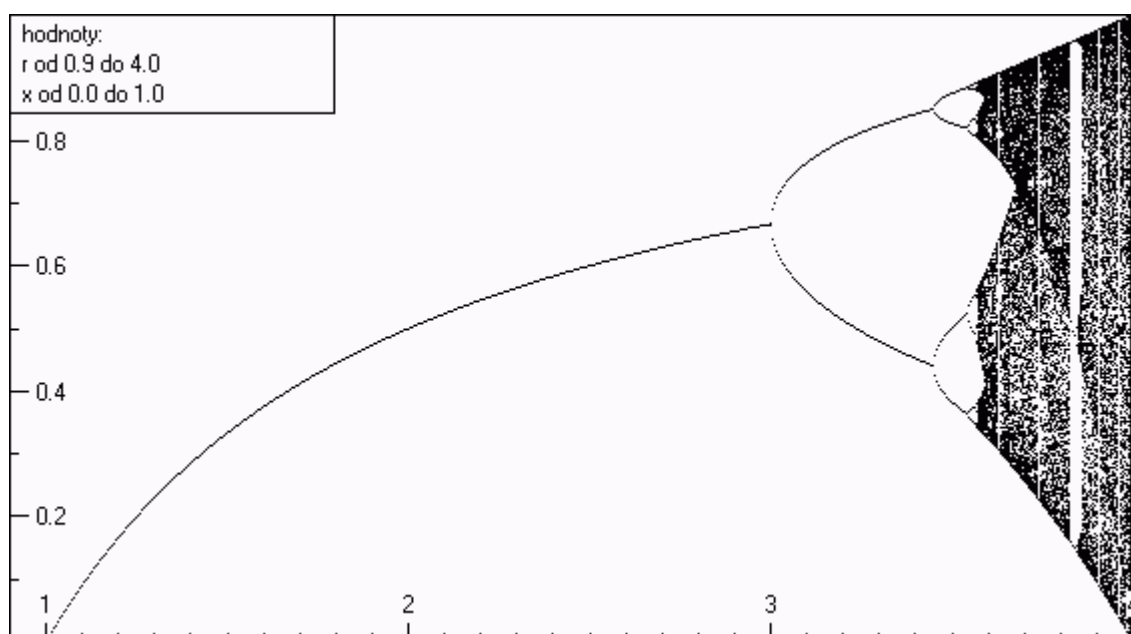
Kde x_n představuje stav populace v n -tém roku, x_{n+1} stav populace v $(n+1)$ roku a r představuje tempo nárůstu populace. Tento vzorec navrhl dávno před Mayem Verhulst, proto se nazývá „Verhulstův vzorec“.

3.2 Bifurkační diagram

Chování logistické rovnice se znázorňuje pomocí tzv. bifurkačního diagramu viz Obr. 3, což je graf, který ukazuje závislost chování modelu systému na řídicím parametru r . Každý bod bifurkačního diagramu křivky představuje ustálený stav systému s různou hodnotou. Soubor těchto hodnot pak v závislosti na parametru r vytváří hladkou křivku.

Při dalším nárůstu však může dojít k zajímavému jevu, a to je tzv. zdvojení periody, což znamená, že se stav populace vrací do původního stavu až po dalším stavu. Při dalším nárůstu pak dochází dalšímu zdvojení již zdvojené periody až vývoj přejde do chaotického průběhu. Ukázkou bifurkačního diagramu je možné vidět na Obr.2. Zajímavostí je, že daný průběh vykazuje fraktální charakter (Feigenbaumova konstanta) - daná zdvojení se v přesném měřítku opakují. A to i u jiných rovnic nežli je tato.

Podrobněji je o této problematice zmíněno v praktické části, kde je bifurkační diagram vykreslen.



Obr. 3 Bifurkační diagram logistické rovnice.

4 STAVOVÝ PROSTOR

Stavový prostor určuje, jakých hodnot může nabývat stavový vektor dynamického systému. Stavový vektor je tvořen množinou proměnných, které mohou nabývat hodnot z určitého intervalu. Interval všech těchto hodnot potom určuje celý stavový prostor.

Stavový prostor může být několika typů:

- konečný
- spočítatelný
- nekonečný

Konečný stavový prostor má omezené množství stavů. Například stavový prostor vyhozené koruny má pouze dva stavy - na minci bude při dopadu buďto *panna* nebo *orel*. Stavový prostor vržené kostky má šest stavů atd.

Spočítatelný stavový prostor má sice nekonečné množství stavů, ale tyto stavy jsou spočítatelné. To znamená, že každému stavu můžeme přiřadit nějaké přirozené číslo. Například stavový prostor nabitého kondenzátoru má nekonečné množství stavů (kondenzátor může být vybitý, nebo nabitý na teoreticky libovolně vysokou hodnotu), ale vzhledem k tomu, že celkový náboj je kvantován elementárním nábojem elektronu, je tento prostor spočítatelný. Celkový náboj tedy můžeme vyjádřit přirozeným číslem, které udává, kolik elektronů dohromady tvoří celkový náboj v kondenzátoru.

Nekonečný stavový prostor má jako stavové proměnné reálná čísla. To znamená, že je nejen nekonečný, ale není také spočítatelný. Neexistuje tedy takové mapování, aby každému stavu jednoznačně přiřadilo celé číslo. Nekonečný stavový prostor má například planeta obíhající okolo slunce. Stavové proměnné jsou zde souřadnice planety. Tyto proměnné nejsou ničím kvantovány, nejsou tedy spočítatelné.

5 ATRAKTOR

Dynamický systém sestává ze stavového prostoru, jehož souřadnice popisují stav systému v daném čase a z dynamických podmínek, které popisují změnu tohoto systému v čase. Stav systému je potom popsán vektorem, který celý leží ve stavovém prostoru. Dynamické podmínky jsou většinou zadány soustavou diferenciálních rovnic, které popisují změnu stavového vektoru v čase. Změna stavu dynamického systému se děje provedením těchto diferenciálních rovnic a nahrazením starého stavového vektoru vektorem novým.

Atraktor (anglicky *attractor*) dynamického systému je stav, do kterého systém směřuje. Je to tedy množina, ve které je stavový vektor, když je systém v nekonečném čase [6].

Atraktory rozdělujeme do několika tříd:

- atraktorem jsou pevné body
- atraktorem jsou periodické body
- atraktorem jsou kvaziperiodické body
- atraktor je chaotický
- podivný atraktor

Jsou-li *atraktorem* dynamického systému *pevné body*, jde o nejjednodušší případ. Systém se tedy v nekonečném čase ustálil v nějakém stabilním stavu a v podstatě už nejde o dynamický systém. Příkladem může být kyvadlo, které se vlivem odporu vzduchu a odporu ložisek zastaví v nejnižším bodě své dráhy.

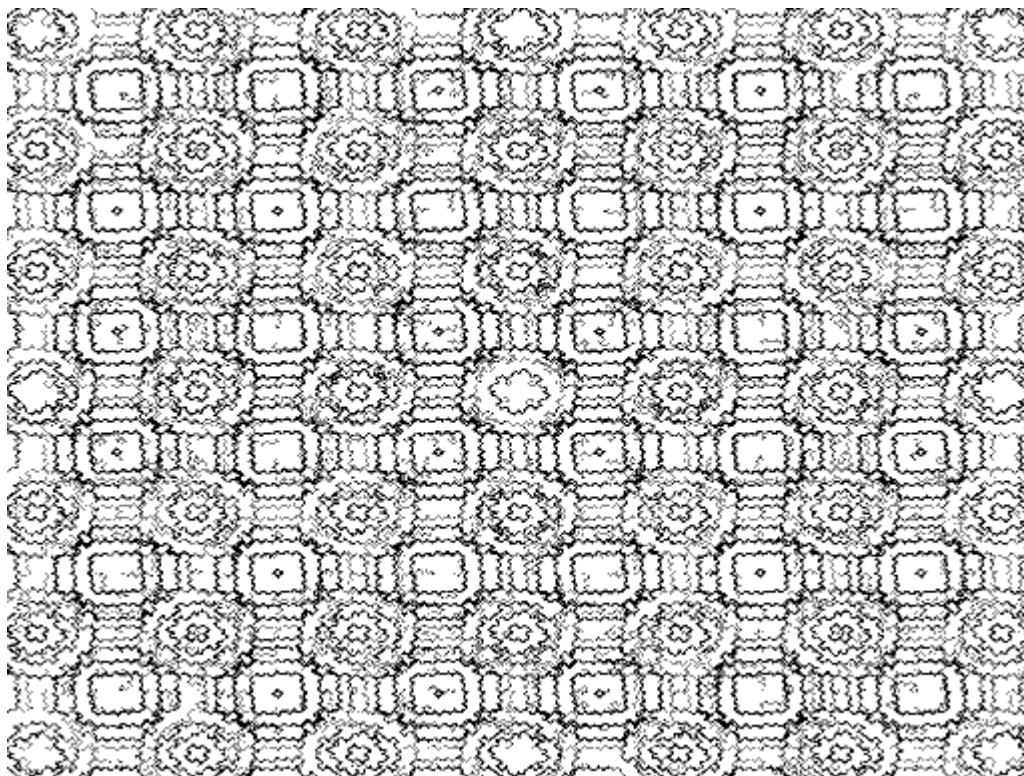
Jsou-li *atraktorem periodické* (resp. *kvaziperiodické*) *body*, jde také o jednoduchý případ. Systém se ustálil tak, že osciluje mezi několika stavy. Příkladem je těleso, které se na své cestě vesmírem dostane do blízkosti velmi hmotného tělesa. Po určitém čase se pohyb tohoto tělesa ustálí na eliptické dráze.

Je-li *atraktor chaotický*, znamená to, že výsledný atraktor nelze v podstatě nijak dopředu předpovědět. To je způsobeno tím, že je systém velmi *citlivý na počáteční podmínky*. Chaotičnost v tomto případě neznamená náhodnost, protože se bavíme o deterministických systémech. Příkladem může být koule postavená na vrcholku jehlanu.

Jakýkoliv vnější podnět způsobí, že koule tento stav opustí a dostane se do některého atraktoru (místo pod jehlanem). Tento atraktor nelze předpovědět, protože nemůžeme bez zásahu do měření zjistit počáteční podmínky. V kvantové fyzice existuje takzvaný *princip neurčitosti*, který má obdobný význam pro kvantové jevy.

Podivný atraktor (anglicky *strange attractor*) je nejzajímavějším případem atraktoru. Termín *strange attractor* poprvé zavedli ve své práci Ruelle a Takens v roce 1970. Tento typ atraktoru vzniká, je-li systém popsán minimálně třemi diferenciálními rovnicemi. Takový systém může mít velmi komplikovaný atraktor, který sice bude chaotický, ale přesto bude vykazovat určité pravidelnosti. Termín *podivný atraktor* není ještě přesně matematicky definován, ale považujeme za něj takový atraktor, který vykazuje stejné vlastnosti, jaké mají fraktály (podivný atraktor je tedy fraktálem).

První dynamický systém, který měl podivný atraktor, vytvořil Ed Lorenz v roce 1963. Šlo o jednoduchý systém se třemi diferenciálními rovnicemi(1), které však ve svém důsledku vytvořily při simulaci chaotický atraktor, který měl fraktální strukturu. Na těchto rovnicích bylo také vidět velkou citlivost na počátečních podmínkách. I při velmi malé změně počátečních podmínek byl výsledek diametrálně odlišný. Atraktory jsou dále probrány v praktické části této práce.



Obr. 4 Ukázka dynamického systému, který má podivný atraktor.

II. PRAKTICKÁ ČÁST

6 VIZUALIZAČNÍ PROSTŘEDÍ MALTAB

6.1 Proč prostředí Matlab?

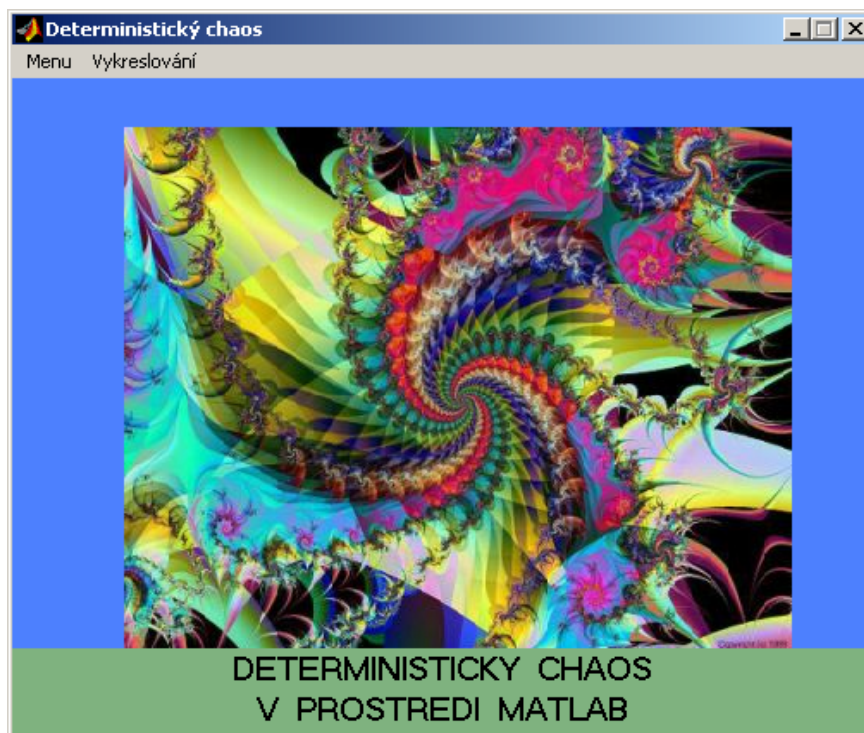
Matlab je interaktivní softwarový balík firmy MathWorks Inc. Matlab je zkratka z MATrix LABoratory, což přesně vystihuje jeho určení. Jeho základní stavebním prvkem je matice a taktéž všechny standardní operace jsou definovány pro matice. Skalár zde chápeme jako matici rozměru 1×1 . Matlab taktéž obsahuje mnoho algoritmu z vyšší matematiky, jako je LU rozklad, spektrální analýza, minimalizace, numerická kvadratura, Fourierova transformace atd. Matlab taktéž podporuje grafické vstup i výstup a to jak v 2D tak i ve 3D. Silnou stránkou Matlabu je taktéž programování pomocí tzv. M-files nebo v jazyce C či Fortran.

6.2 Popis programu pro vizualizaci deterministického chaosu

Pro jednoduchost je v další části používáno pro vizualizační program vytvořený v programovacím prostředí Matlab zkrácený název „program“.

6.2.1 Hlavní okno programu

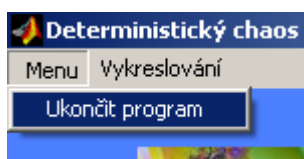
Cílem tohoto programu vytvořeném v programovém prostředí Matlab bylo zajištění jasného, snadného a vizuálně atraktivního popisu deterministického chaosu. Program popisuje deterministický chaos v jedno a více dimenzionálních mapách. Základní menu programu je na Obr. 5.



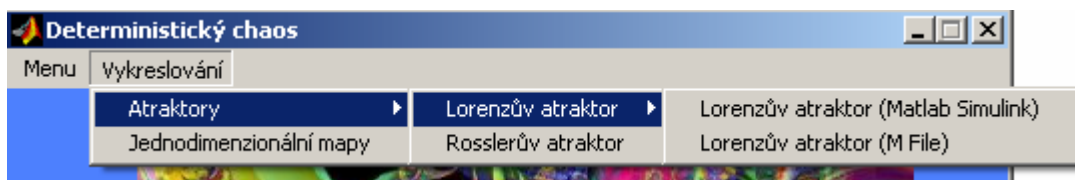
Obr. 5 Hlavní menu programu.

V hlavním menu programu je po kliknutí na políčko „Menu“ nabídnuta položka „Ukončit program“ viz. Obr. 6, která slouží k ukončení programu včetně prostředí Matlab.

Další položka „Vykreslování“ pod sebou skrývá nabídku viz. Obr. 7 .



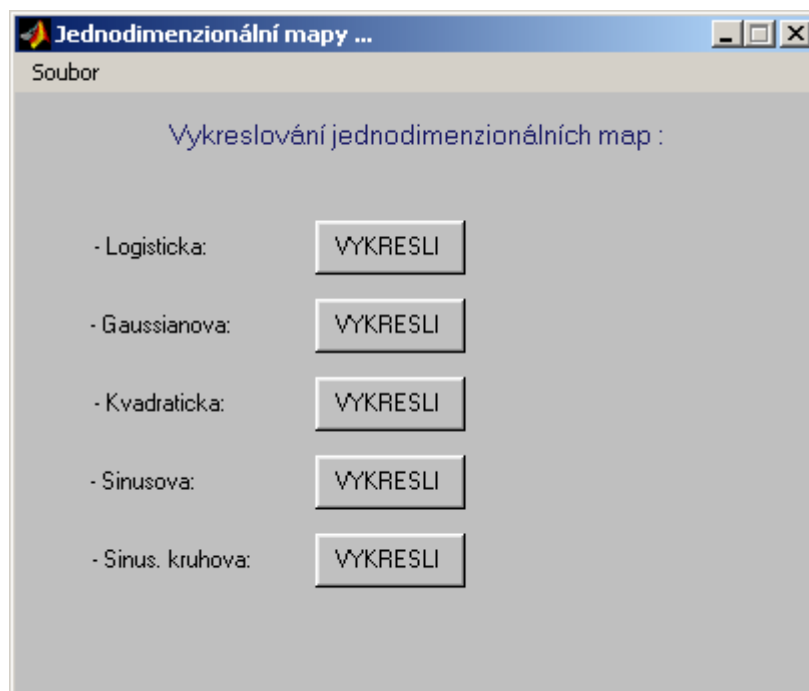
Obr. 6 Nabídka „Menu“.



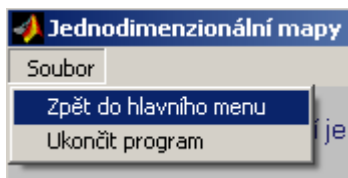
Obr. 7 Nabídka „Vykreslování“.

První položka „Atraktory“ se dále dělí na vykreslování Lorenzova a Rosslerova atraktoru. Nabídka „Lorenzův traktor“ nabízí možnost vykreslení, jak přes Matlab Simulink, tak i

přes, v programu více používaný, M-file. Položka „Jednodimenzionální mapy“ nabízí vykreslení pěti jednodimenzionálních map viz. Obr. 8.



Obr. 8 Jednodimenzionální mapy.



Obr. 9 Menu Jednodimenzionální. Mapy.

V tomto okně se také nachází menu „Soubor“, který obsahuje nabídku viz. Obr. 9. Toto menu je ve stejné podobě obsaženo v téměř všech dalších oknech programu, a proto dále nebude zmiňováno.

6.2.2 Jednodimenzionální (jedno-rozměrové) mapy

Logistická rovnice

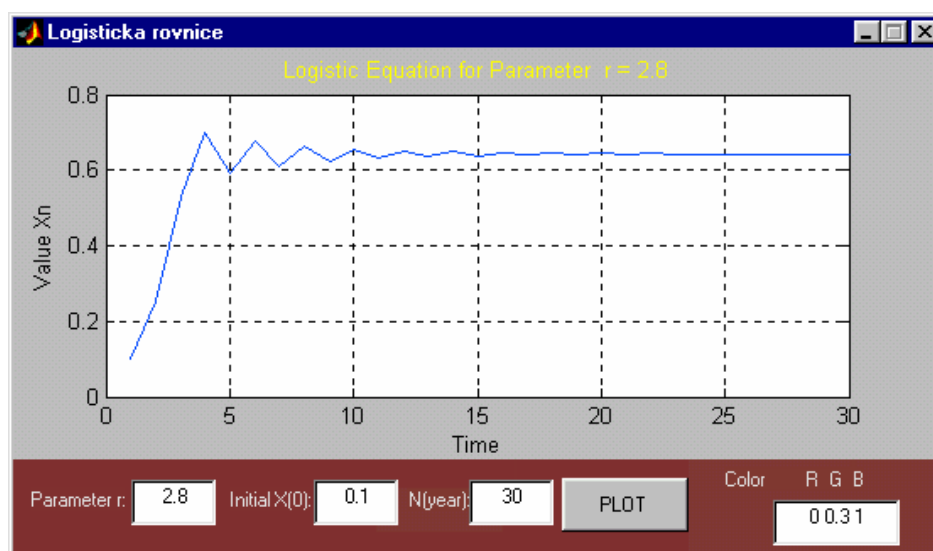
Logistická rovnice je nejznámější z uvedených map Obr. 8, je to možná proto, že její zdánlivě jednoduchý vzorec (2) vykazuje tak složité „chování“ viz. Obr. 15. Nebo dalším důvodem může být to, že právě tato rovnice byla jako první, která odhalila dosud nepředpovídané chaotické chování.

V další části bude názorně ukázáno, jak můžeme pomocí vzorce, který používal Robert May pro předpovídání vývoje populace. Již zmíněný vzorec logistické rovnice:

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

Tento vzorec slouží na to, že jakmile budeme znát populaci x_i v i -tém roce, můžeme pomocí něj vypočítat počet jedinců x_{i+1} v $(i+1)$ -tém roce.

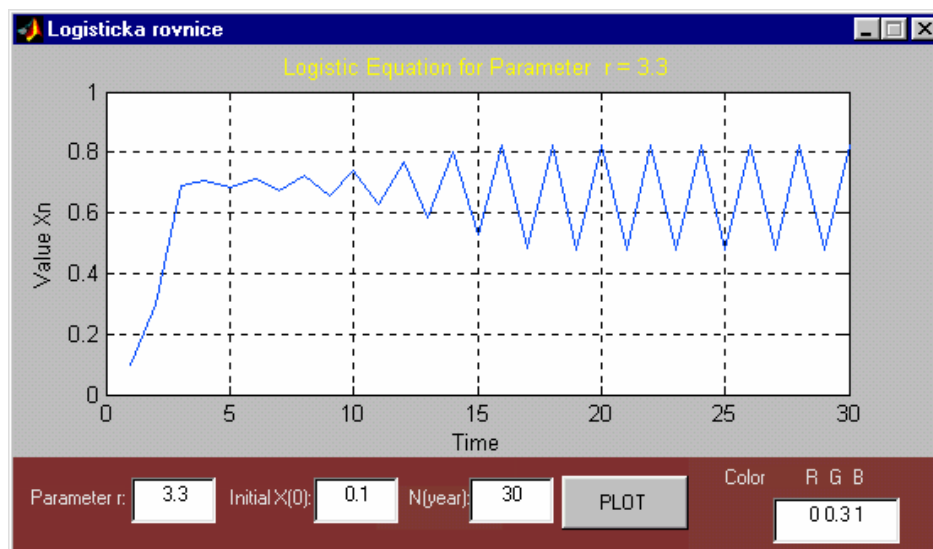
Nechť populace na začátku je $x_0 = 0,1$ a nechť koeficient r má hodnotu $r = 2,8$. Kdyby byla populace velmi malá (hodně menší jak 1), znamenala by takováto hodnota r nárůst populace v dalším roce o 180 % viz. Obr. 10.



Obr. 10 Populace při $X(0)=0.1$, $r=2.8$.

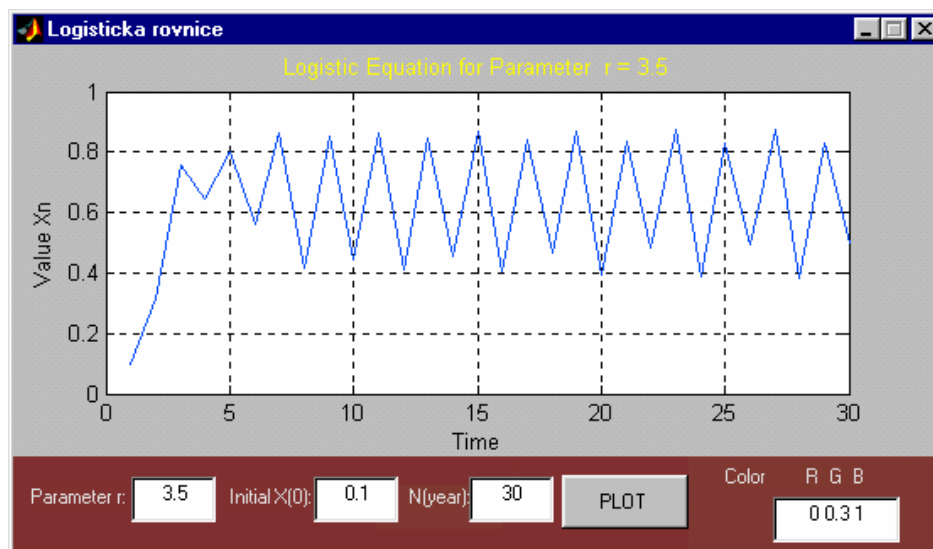
Z obrázku Obr. 10 je možné dobře vidět, že populace nejprve prudce stoupala až do té doby, než byla příliš vysoká a nezačal se projevovat hlad. To způsobilo pokles populace. Potom populace opět vzrostla a klesala, dokud se neustálila na konečné, stálé hodnotě.

Bude se populace chovat stále podobně, t.j. přejde vždy do ustáleného stavu? Pokusíme se na to teď odpovědět. Zvolíme si jinou hodnotu tempa růstu r . Volme teď větší hodnotu $r = 3,3$. Předpokládejme, že počáteční populace x_0 se znovu rovná 0,1. Na Obr. 11 je pak posloupnost populací v dalších třiceti letech vývoje.

Obr. 11 Vývoj pro $r = 3.3$.

Je vidět, že pro větší hodnotu r se po počátečním růstu populace dostala po několika rocích do stavu, kdy osciluje mezi dvěma hodnotami 0,5 a 0,81. I takovéto něco je možné očekávat od vývoje počtu jedinců populace. Vidíme, že v tomto případě není pevně určena žádná rovnovážná, konstantní populace.

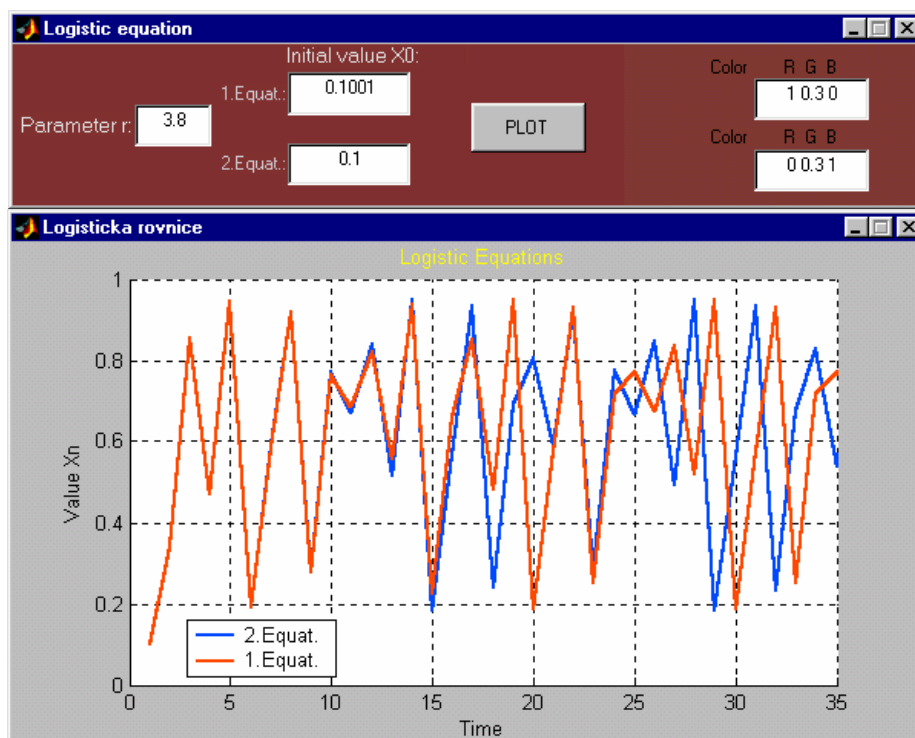
Ještě složitější vývoj je možné pozorovat, když zvolíme pro r hodnotu 3,5. Při stejné hodnotě x_0 jako předtím, dostaneme takovýto vývoj viz. Obr. 12:

Obr. 12 Vývoj pro $r=3.5$.

Populace pro $r=3.5$ nejprve naroste a potom postupně střídá za sebou jdoucí čtyři hodnoty. Pokud zvolíme za parametr r hodnotu 3,8, dostaneme průběh, ve kterém nevidíme žádné opakování, žádnou periodu. Vývoj vypadá jako naprosto nahodilý, avšak je dán vztahem

(2). Není v něm žádná náhodnost a právě takovému druhu chování se říká deterministický chaos.

Vývoj populace v chaotickém režimu velmi závisí na počátečních podmínkách, to vidíme z Obr. 13, kde rozdíl počátečních hodnot je na 4 desetinném místě a již po 10 letech vývoje se populace výrazně rozcházejí.



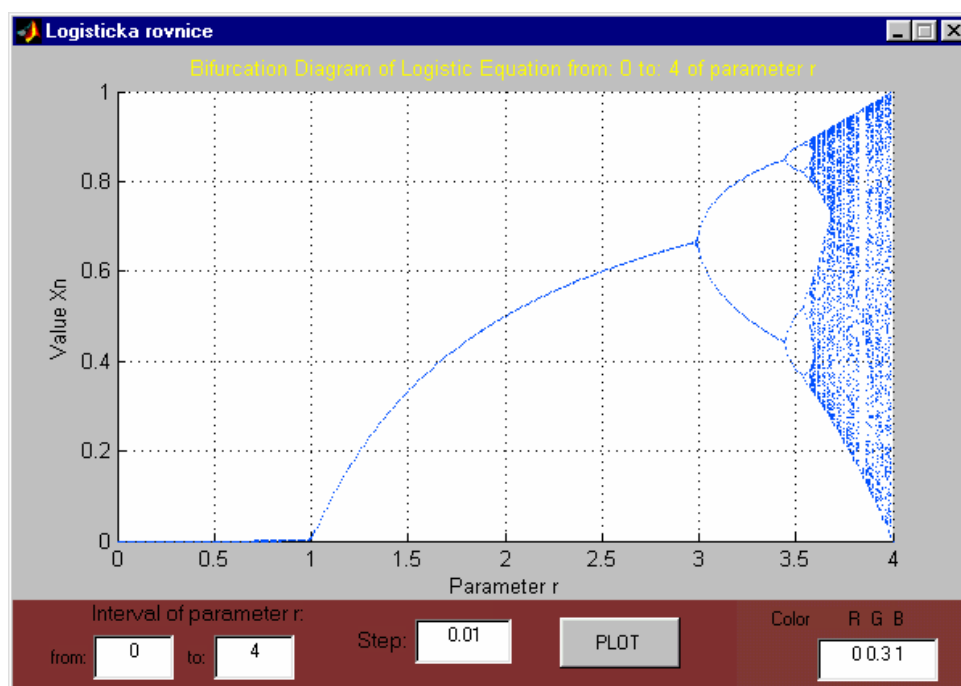
Obr. 13 Srovnání dvou logistických rovnic s „různou“ počáteční hodnotou $X(0)$.

Teď si pokusíme odpovědět na stejnou otázku, kterou si položil Robert May. Jak se bude populace chovat pro různé hodnoty r ? Přesněji: Na jakém typu chování se populace pro různé hodnoty parametru r nakonec ustálí?

Je potřeba si vybrat nějakou hodnotu r z povoleného intervalu hodnot od 0 do 4 (pro hodnoty větší jak 4 se populace vymyká z kontroly a diverguje). Po dostatečném počtu kroků, když vymizí přechodový proces a hodnot populace se ustálí, je třeba se podívat na to, o jaké chování vlastně jde. Tady jsou v principu jen 3 možnosti. Buď se populace ustálí na jediné hodnotě, a nebo bude pravidelně oscilovat mezi několika různými hodnotami. A nakonec, může se chovat i chaoticky, což znamená, že pohyb není periodický. Jakmile zjistíme, jak se populace chová, vybereme si jinou hodnotu parametru r a celý postup opakujeme. Pokud tedy budeme vynášet do grafu na vodorovnou osu hodnoty parametru r

a na svislou osu odpovídající hodnoty populace například od roku 5000 do roku 5300 vznikne graf z obrázku Obr. 14. Čím větší r , tím vyšší konečná hodnota populace. Pro hodnoty r z tohoto intervalu dostáváme na grafu křivku.

Pro hodnoty r nad hodnotou 3 dochází k zajímavému jevu – rozdělení, (bifurkaci; fork - anglicky vidlička), křivky z předcházejícího intervalu. Populace se už nemůže ustáli jen na jedné konečné hodnotě. Ustaluje se na dvou hodnotách, mezi kterými následně osciluje. Při hodnotě $r = 3,449$ dochází k novému rozdělení obou křivek. Populace teď osciluje mezi 4 hodnotami. Při zvětšování r ještě víc, dojde k celé řadě dalších bifurkací.



Obr. 14 Bifurkační diagram logistické rovnice pro parametr r v intervalu $(0,4)$.

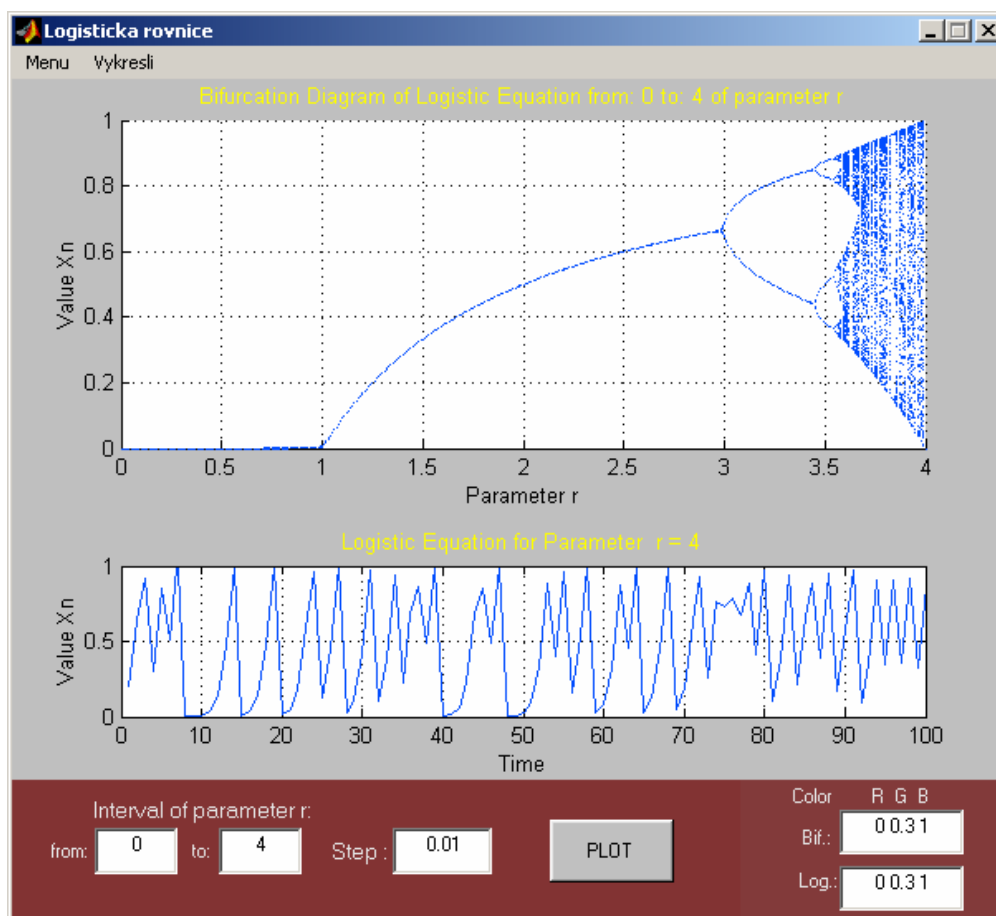
V programu je cyklus bifurkačního diagramu logistické rovnice popsán takto:

```

1 for A=Afrom:Astep:Ato
2     for i=2:1:cykly
3         pom(i)=A*pom(i-1)*(1-pom(i-1))
4     end
5     for i=odcyklu:1:cykly
6         x(j)=A;
7         y(j)=pom(i);
8         j=j+1;
9     end
10 end

```

Dále jsou popisovány a vizualizovány další mapy (rovnice) generující chaotické chování. Princip je analogicky odvozen od předcházející vizualizace logistické funkce.



Obr. 15 Logistická rovnice s jejím bifurkačním diagramem.



Obr. 16 Menu logistické rovnice.

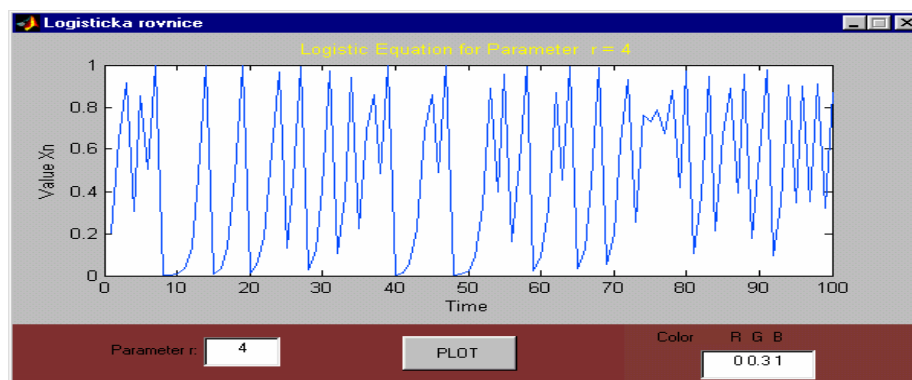
Na Obr. 15 je vidět, že v programu pro logistickou rovnici je možné měnit parametr r , krok („Step“) a barvy („Color“) v obou grafech. Měnit barvy je zde možné pomocí změny vzájemného poměru jednotlivých základních barev RGB. Vykreslování se provádí tlačítkem „PLOT“.

Další možnosti tohoto okna jsou uloženy v nabídce „Vykresli“ Obr. 16.

Nabídka okna „Vykreslit logistickou rovnici“ Obr. 17.

Nabídka okna „Srovnání dvou logistických rovnic“ Obr. 13.

Nabídka okna „Vykreslit bifurkační diagram“ Obr. 14.



Obr. 17 Graf logistické rovnice pro parametr $r = 4$.

Gaussova mapa

Je dána vztahem (3).

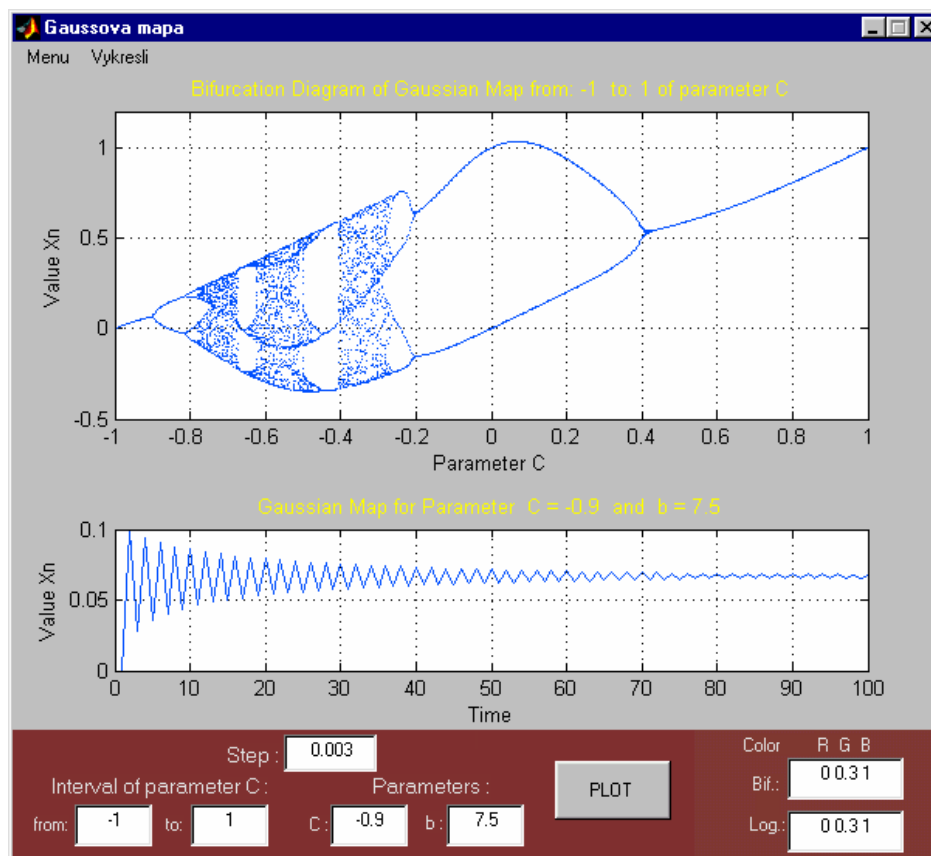
$$x_{n+1} = e^{-bx_n^2} + C \quad (3)$$

Ukázka hlavního cyklu programu pro vykreslení Gaussovy mapy (m-file):

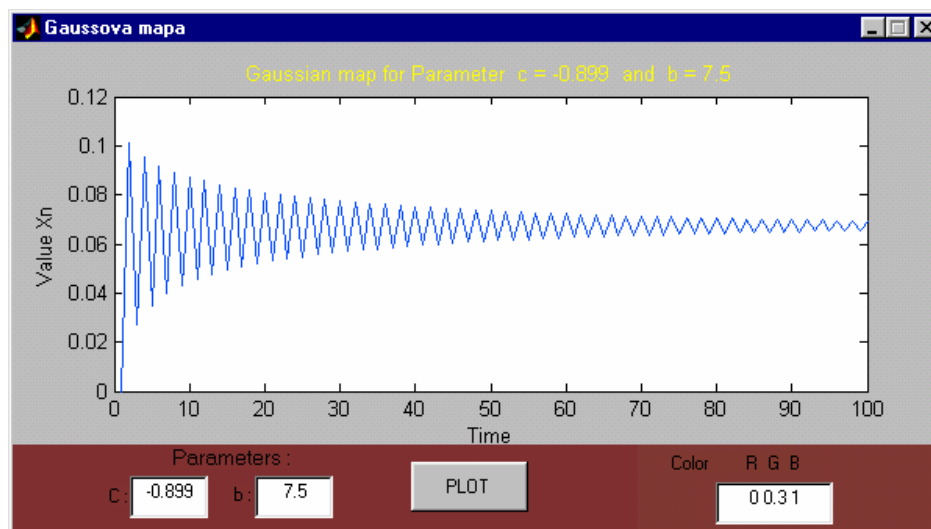
```

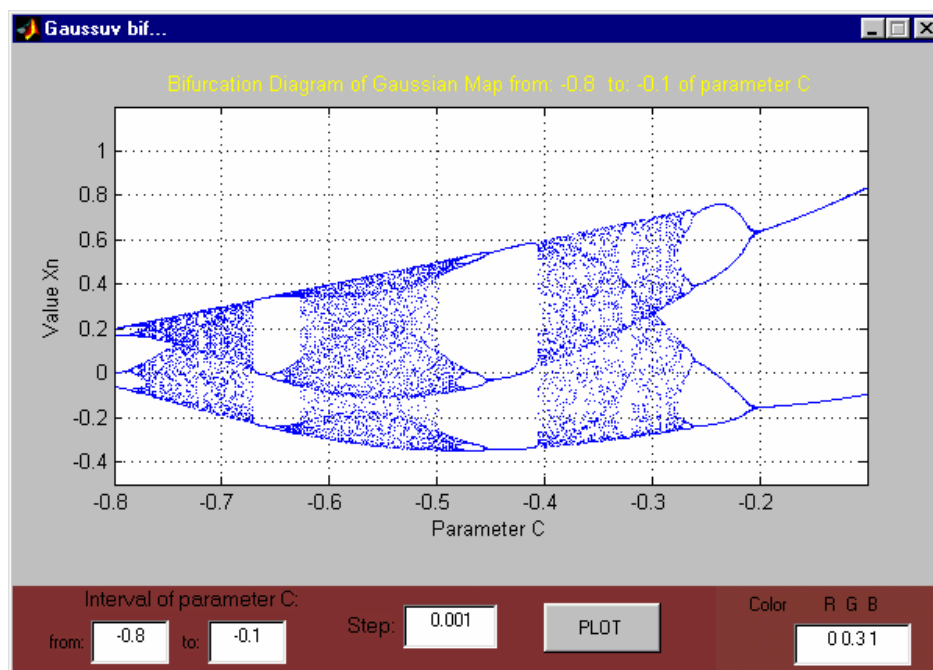
1 for C=Cfrom:Cstep:Cto;
2     for i=2:1:cykly;
3         pom(i)=exp(-b*pom(i-1)^2)+C;
4     end
5     for i=odcyklu:1:cykly;
6         x(j)=C;
7         y(j)=pom(i);
8         j=j+1;
9     end
10 end

```



Obr. 18 Gaussova mapa se svým bifurkačním diagramem.

Obr. 19 Gaussova mapa pro parametry $C=-0,898$ a $b=7.5$.



Obr. 20 Bifurkační diagram Gaussovy mapy.

Kvadratická mapa

Je dána vztahem (4).

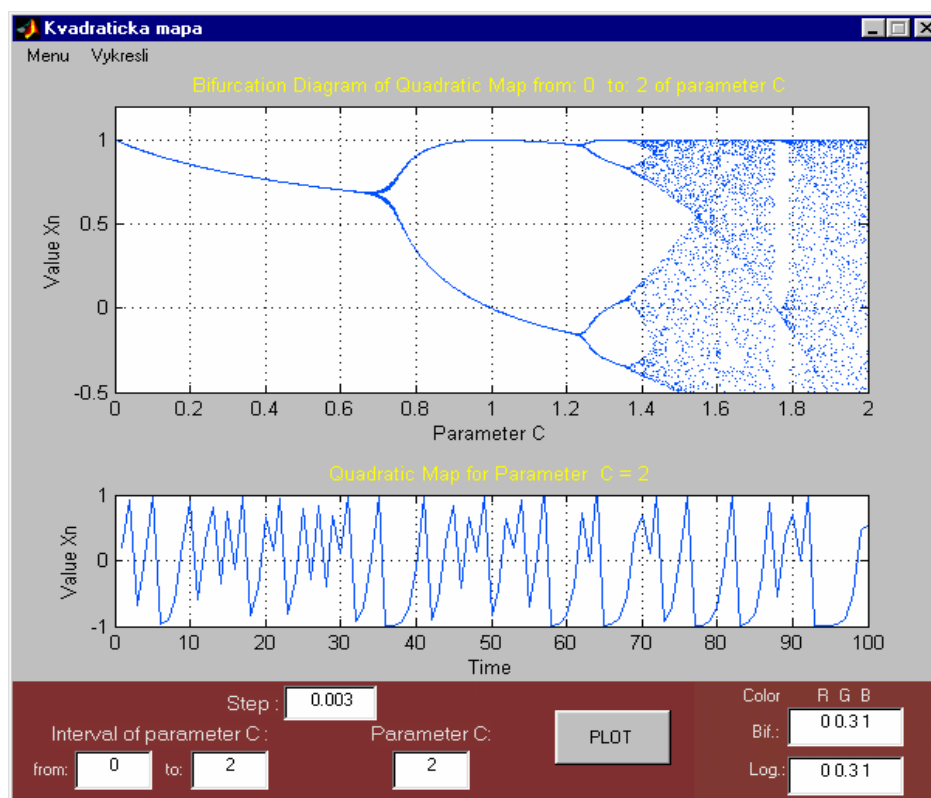
$$x_{n+1} = 1 - Cx_n^2 \quad (4)$$

Ukázka hlavního cyklu programu pro vykreslení kvadratické mapy (m-file):

```
1 for i=2:1:100;
2     x(i)=1-C*x(i-1)^2;
3 end
```

Ukázka hlavního cyklu programu pro vykreslení bifurkačního diagramu kvadratické mapy :

```
1 for C=Cfrom:Cstep:Cto;
2     for i=2:1:cykly;
3         pom(i)=1-C*pom(i-1)^2;
4     end
5     for i=odcyklu:1:cykly;
6         x(j)=C;
7         y(j)=pom(i);
8         j=j+1;
9     end
10 end
```

Obr. 21 Kvadratická mapa s bifurkačním diagramem pro parametr $C=2$.

Sinusová mapa

$$x_{n+1} = B \sin[\pi x_n] \quad (5)$$

Ukázka hlavního cyklu programu pro vykreslení sinusové mapy:

```

1 for i=2:1:50;
2     yy(i)=B*sin(pi*yy(i-1));
3 end

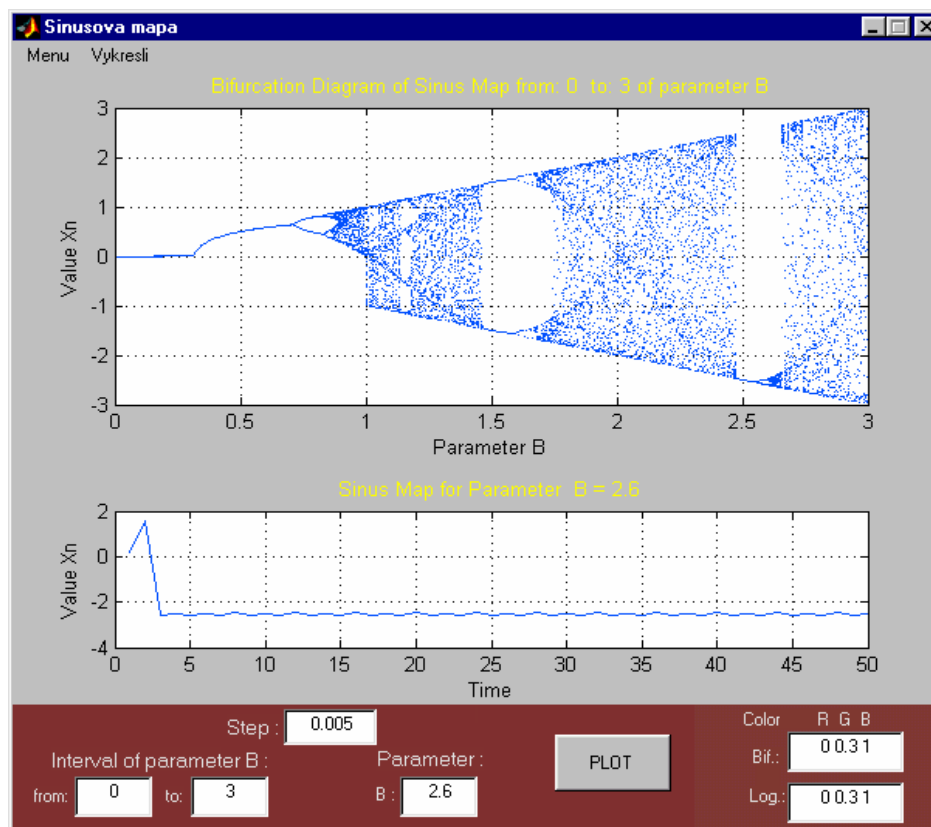
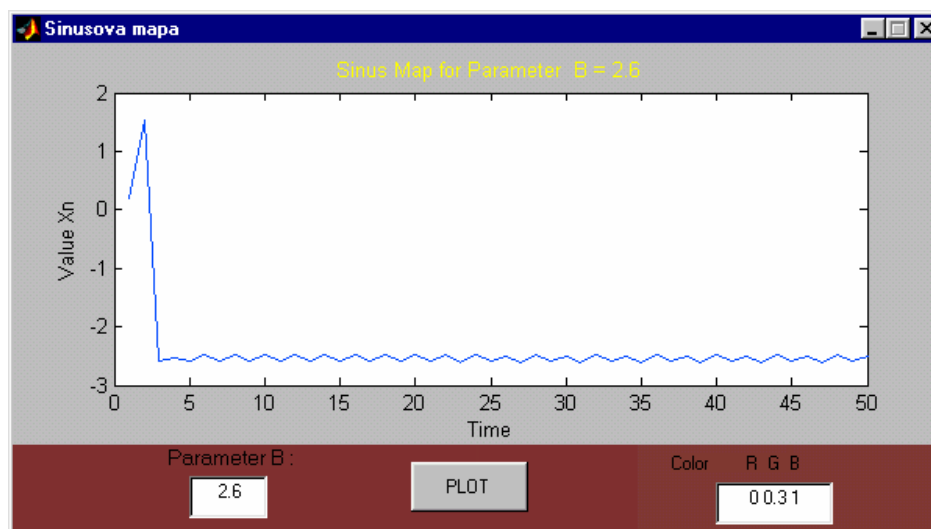
```

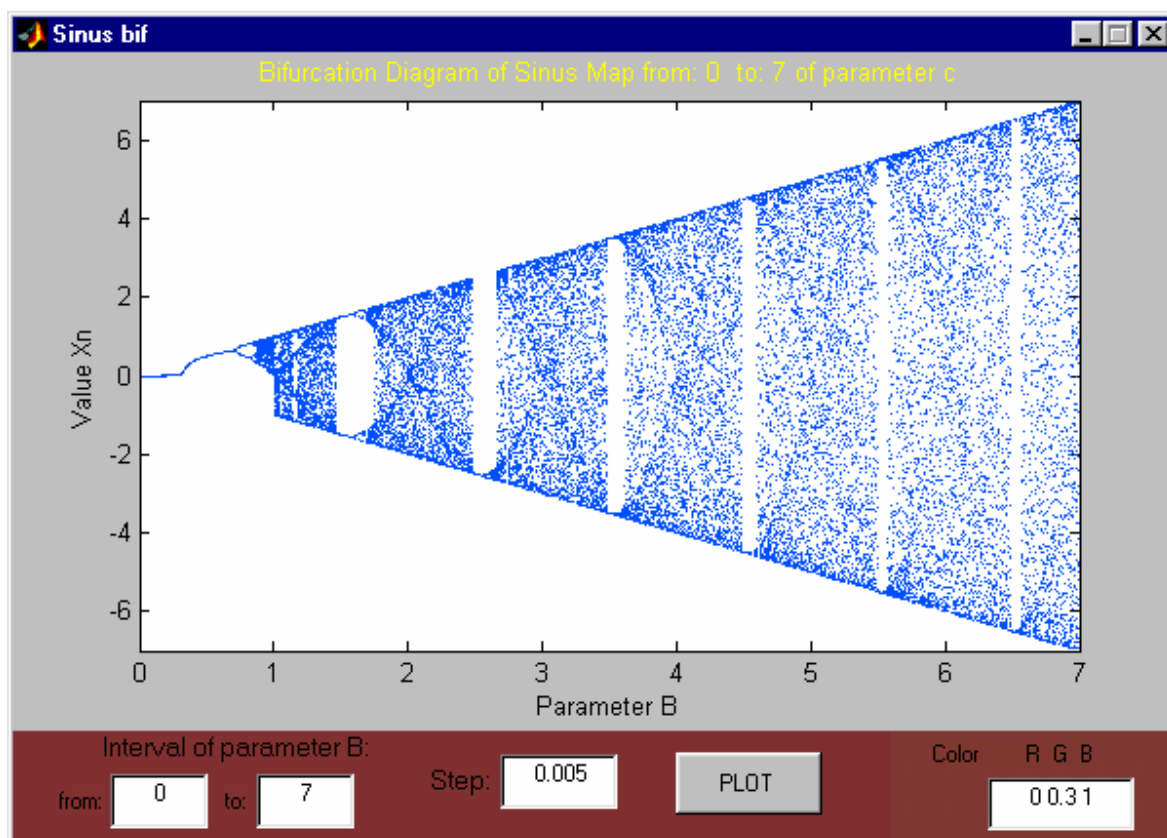
Ukázka hlavního cyklu programu pro vykreslení bifurkačního diagramu sinusové mapy:

```

1 for B=Bfrom:Bstep:Bto;
2     for i=2:1:cykly;
3         pom(i)=B*sin(pi*pom(i-1));
4     end
5     for i=odcyklu:1:cykly;
6         x(j)=B;
7         y(j)=pom(i);
8         j=j+1;
9     end
10 end

```

Obr. 22 Sinusová mapa s bifurkačním diagramem pro parametr $B \in (0,3)$.Obr. 23 Sinusová mapa pro parametr $B=2,6$.

Obr. 24 Bifurkační diagram sinusové mapy pro parametr $B \in (0, 7)$.***Sinusovo-kruhová mapa***

$$\phi_{n+1} = \phi_n + \Omega - \frac{K}{2\pi} \sin[2\pi\phi_n] \quad (6)$$

Ukázka hlavního cyklu programu pro vykreslení sinus-kruhové mapy (m-file):

```

1 for i=2:1:100;
2     sk(i)=sk(i-1)+omega-K/(2*pi)*sin(2*pi*sk(i-1));
3     if sk(i)>0;
4         pom=floor(sk(i));
5         sk(i)=sk(i)-pom;
6     elseif sk(i)<0;
7         pom=round(sk(i));
8         sk(i)=pom-sk(i);
9     end
10 end

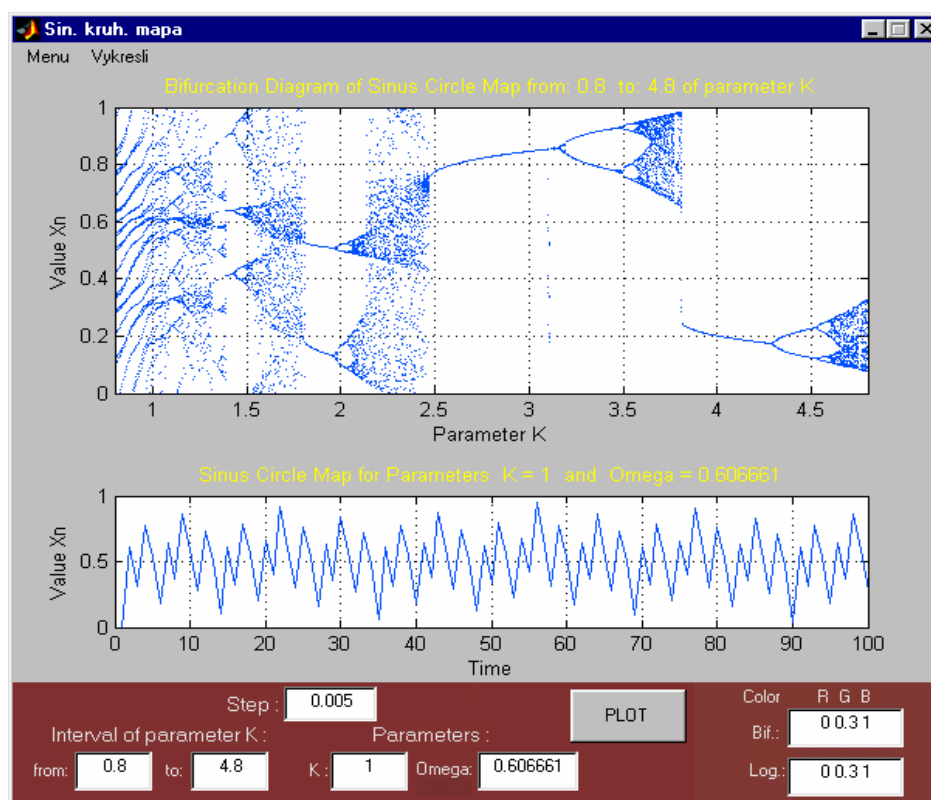
```

Ukázka hlavního cyklu programu pro vykreslení bifurkačního diagramu sinus-kruhové mapy (m-file):

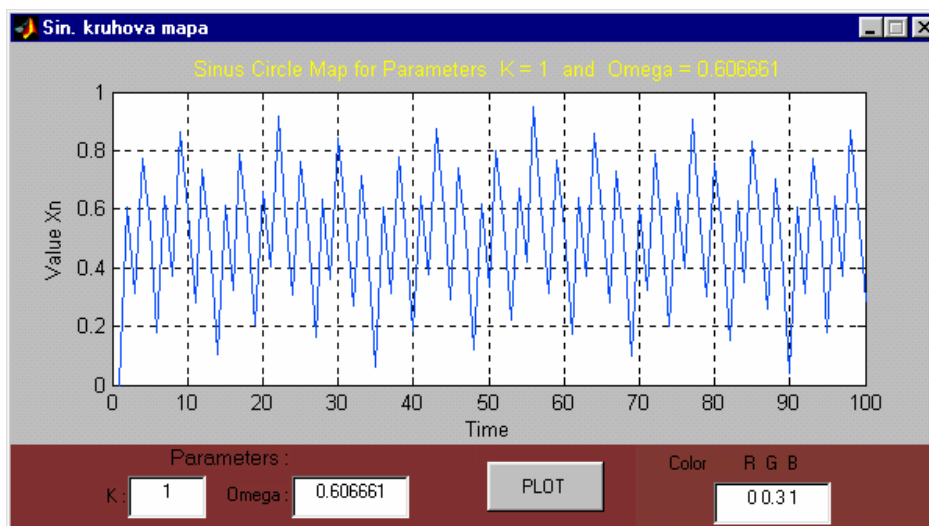
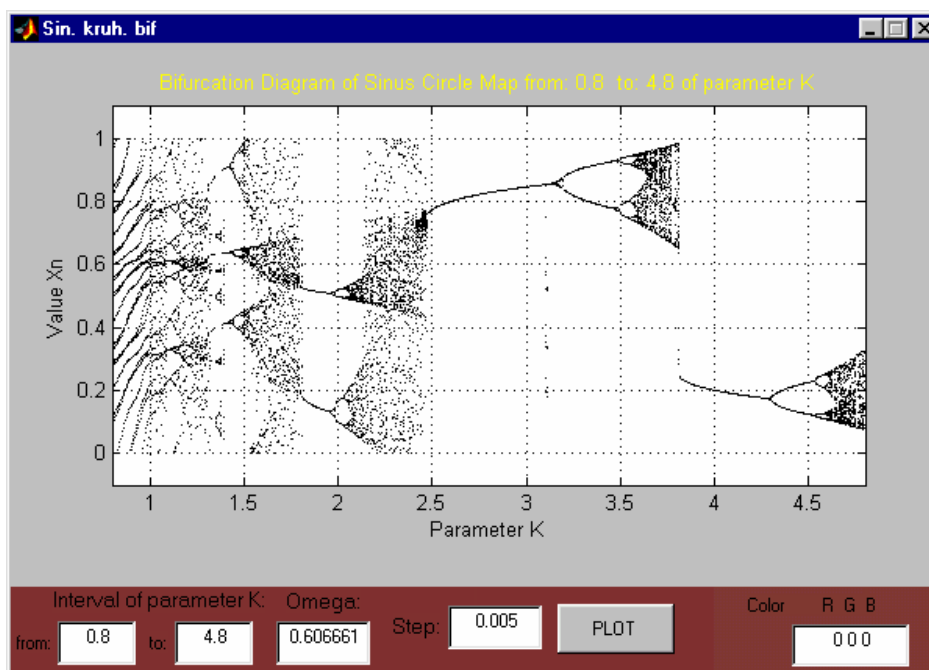
```

1 for K=Kfrom:Kstep:Kto;
2     for i=2:1:cykly;
3         pom(i)=pom(i-1)+omega-K/(2*pi)*sin(2*pi*pom(i-1));
4         if pom(i)>0;
5             pom2=floor(pom(i));
6             pom(i)=pom(i)-pom2;
7         elseif pom(i)<0;
8             pom2=round(pom(i));
9             pom(i)=pom2-pom(i);
10        end
11    end
12    for i=odcyklu:1:cykly
13        x(j)=K;
14        y(j)=pom(i);
15        j=j+1;
16    end
17 end

```



Obr. 25 Sinus-kruhová mapa s bifurkačním diagramem.

Obr. 26 Sinus-kruhová mapa s parametry $\Omega = 0,6066661$ a $K=1$.Obr. 27 Bifurkační diagram sinus-kruhové mapy s parametrem $\Omega = 0,606661$.

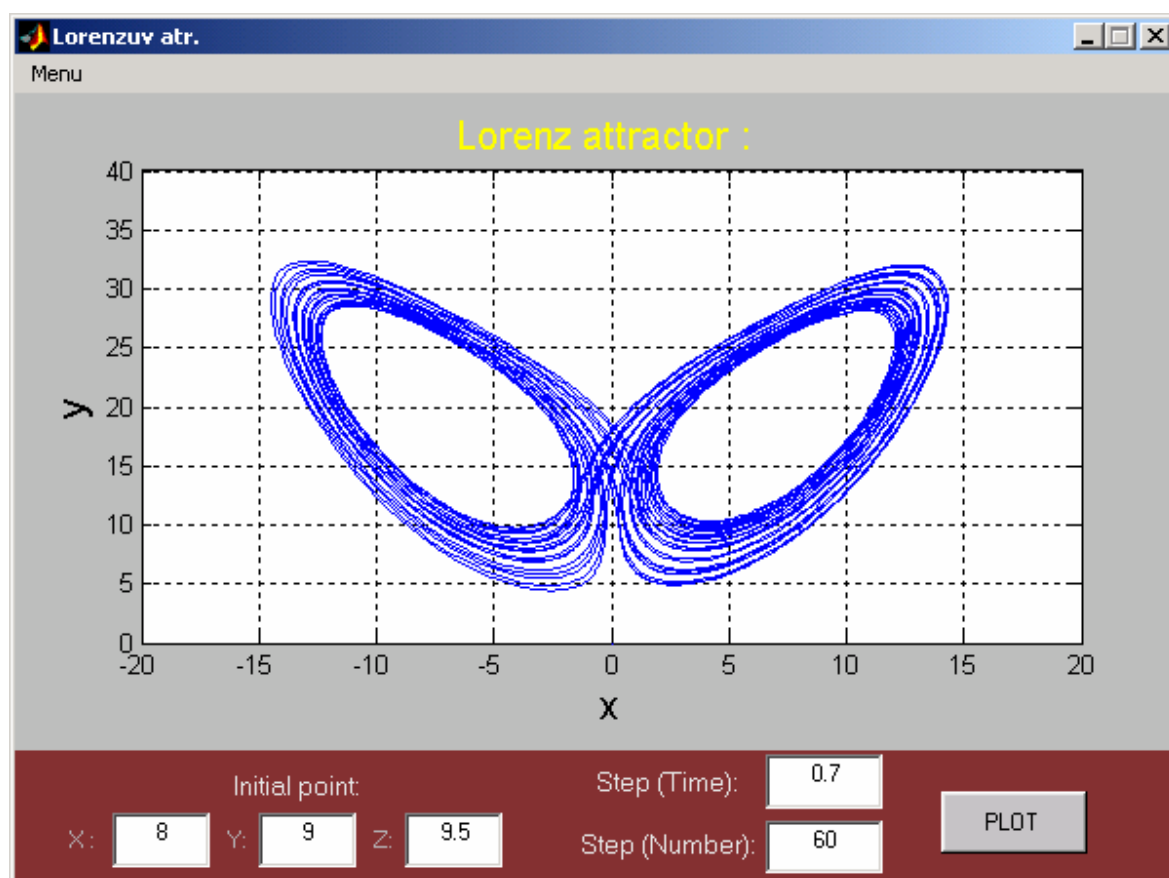
6.2.3 Atraktory

Dynamický systém má tendenci ustálit se po jistém čase na nějaké trajektorii ve fázovém prostoru. Tato konkrétní trajektorie jako kdyby přitahovala pohybující se stavový bod ve fázovém prostoru. Na ustálenou trajektorii se tento bod dostane po krátkém přechodovém jevu. Říkáme, že tato trajektorie ho "přitáhla". V angličtině se přiahovat řekne „to attract“. Proto se ustálené trajektorii, která "přitahuje" stavový bod říká atraktor.

Dynamický systém má teda po uplynutí jistého času tendenci usadit se na nějakém atraktoru.

Lorenzův atraktor

Na obrázku Obr. 28 je zobrazen nejznámější atraktor, a to Lorenzův.

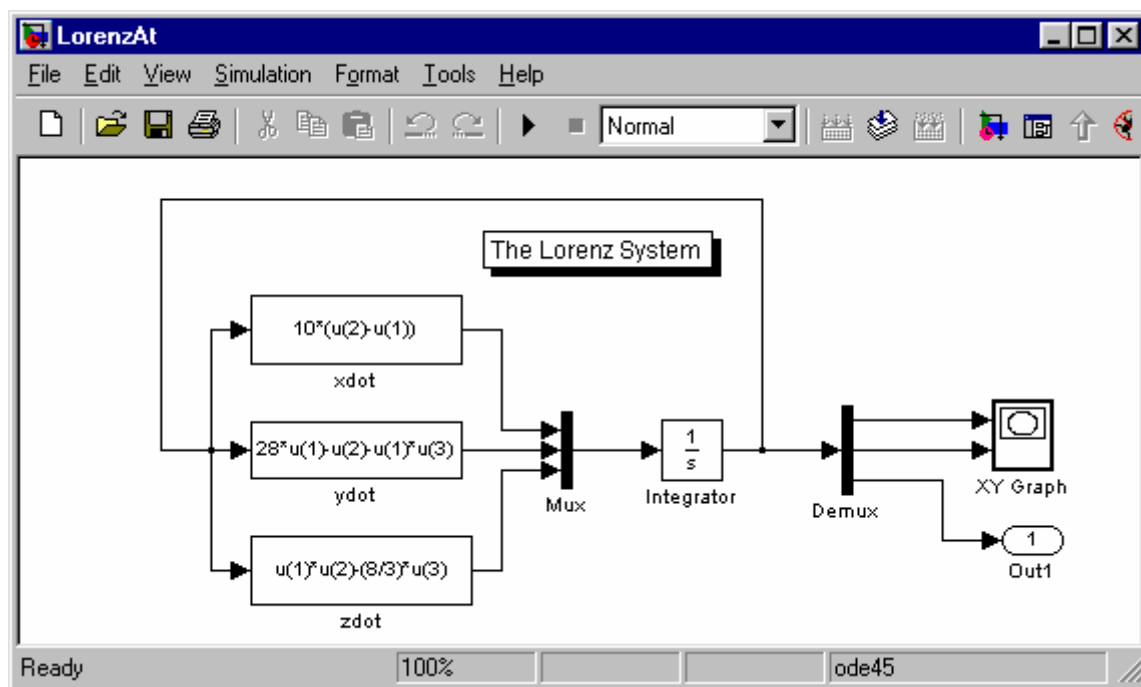


Obr. 28 Lorenzův atraktor.

V dialogovém okně programu je možné měnit jak počáteční body $X Y Z$, tak i krok v čase „Step(Time)“ a konečný počet kroků „Step(Number)“ (pro rychlejší vykreslení, volit krok „Step(Time)“ nad 0.5).

Stejný atraktor je taky možné vykreslit přes volbu položky „Lorenzův atraktor (Matlab Simulink)“ viz. Obr. 7. V tomto případě není potřeba novější verze Matlabu pro s podporou vykreslování diferenciálních rovnic, oproti případu vizualizace Lorenzova atraktoru při výběru položky „Lorenzův atraktor(M File) viz. Obr. 7.

Schéma ze Simulinku pro vykreslování Lorenzova atraktoru je na obrázku Obr. 29

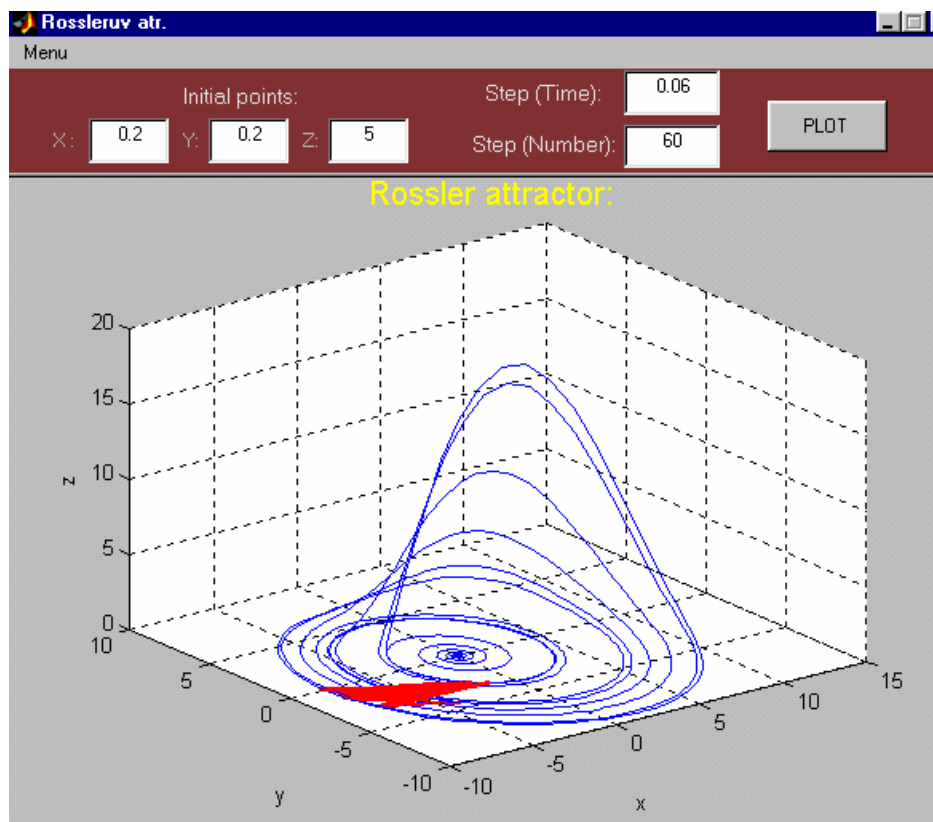


Obr. 29 Schéma ze Simulinku pro vykreslení Lorenzova atraktoru

Rosslerův atraktor

Rosslerův atraktor je umělý systém generující podivný atraktor, založený na nejjednodušších principech generace chaosu, a to transformaci "rozprostření a ohyb" (stretch and fold). Rosslerův atraktor (1976) vychází z atraktoru Lorenzova, který byl publikován 13 roků před Rosslerovým.

Po vybrání položky Rosslerův atraktor v hlavním menu Obr. 7, se objeví panel s nastavitelnými parametry – počáteční bod $X Y Z$, krok v čase „Step(Time)“ a celkový počet kroků „Step(Number)“ Obr. 30.



Obr. 30 Rosslerův atraktor

ZÁVĚR

Hlavním úkolem této práce bylo přiblížení problematiky deterministického chaosu čtenáři v programovém prostředí Matlab, kde může najít poměrně přehlednou vizualizaci matematického popisu chaotického chování. Ještě před samotným předváděním příkladů chaotického chování se v teoretické části práce zaměřuje na objasnění principů a závislostí chaotického chování včetně její historie. V teoretické části jsou jen obecně nastíněny některé náležitosti deterministického chaosu, jako například stavový prostor, logistická rovnice, bifurkační diagram, atraktory. Poměrně zajímavější pro čtenáře je praktická část této práce, kde jsou názorné ukázky deterministického chaosu uvedeny ve vizuální podobě.

V úvodu praktické části je popis programu pro vizualizaci deterministického chaosu v programovém prostředí Matlab. Tento popis je určen pro čtenáře, kteří by si chtěli vyzkoušet více příkladů chování deterministického chaosu v tomto programu.

Po prostudování této práce je možné si uvědomit jednoduchosti, které se za chaosem skrývají. Podobnými pokusy se vědcům podařilo pomocí jednoduchých pravidel "stvořit" rostliny, které botanici bez problému identifikovali. Pro co možná nejrealističtější počítačové modelování krajiny se používá právě těchto postupů.

Možná právě takto funguje příroda, kdy v pár bitech informací můžou být zachyceny větvičí se struktury. Díky fraktálovému charakteru většiny přírodních struktur je možné vytvořit velké množství tvarů. Možná právě tímto způsobem jsou kódovány genetické informace, které předepisují ony nesmírně složité struktury lidského organismu (cévní systém, plíce, ...). Možná právě takto vznikají tvary rostlin. Možná právě takto se rodí celá příroda. Možná právě takto se z jednoduchosti rodí zdánlivě chaotická skutečnost a my teď zpětně v této chaotické skutečnosti poznáváme onu podstatu, onu jednoduchost, onen řád.

SEZNAM POUŽITÉ LITERATURY

- [1] Zelinka, Ivan. *Chaos Page* [online]. [cit. 2005-04-15]. Dostupný z WWW:
< <http://www.ft.utb.cz/people/zelinka/chaos/index.html> >.
- [2] Hilborn R.C.1994, *Chaos and Nonlinear Dynamics*, Oxford University Press, ISBN 0-19-508816-8, 1994
- [3] Horák J., Krlín L.1996, *Deterministický chaos*, Academia, ISBN 80-200-0416-5,1996.
- [4] Andrievskii B. R. , Fradkov A. L, Control of chaos Methods and Applications II Applications. *Automation and remote control*.2003, Vol. 65, no. 4, p. 505 – 509.
- [5] *Teoria chaosu a fraktály* [online]. [cit. 2005-05-10]. Dostupný z WWW:
< <http://www.chaos.host.sk/> >.
- [6] Peitgen H.O., Jurgens H., Saupe D. 1992, *Chaos and Fractals*, New Frontiers of Science, Springer-Verlag 1992, ISBN 3-540-97903-4

SEZNAM OBRÁZKŮ

Obr. 1 Ukázka citlivosti na poč. podmínky.	11
Obr. 2 Jednoduchý elektronický obvod generující chaos.	12
Obr. 3 Bifurkační diagram logistické rovnice.	14
Obr. 4 Ukázka dynamického systému, který má podivný atraktor.	17
Obr. 5 Hlavní menu programu.	20
Obr. 6 Nabídka „Menu“	20
Obr. 7 Nabídka „Vykreslování“	20
Obr. 8 Jednodimenzionální mapy.	21
Obr. 9 Menu Jednodimenzionální. Mapy.	21
Obr. 10 Populace při $X(0)=0.1$, $r=2.8$	22
Obr. 11 Vývoj pro $r = 3.3$	23
Obr. 12 Vývoj pro $r=3.5$	23
Obr. 13 Srovnání dvou logistických rovnic s „různou“ počáteční. hodnotou $X(0)$	24
Obr. 14 Bifurkační diagram logistické rovnice pro parametr r v intervalu $(0,4)$	25
Obr. 15 Logistická rovnice s jejím bifurkačním diagramem.	26
Obr. 16 Menu logistické rovnice.	26
Obr. 17 Graf logistické rovnice pro parametr $r = 4$	27
Obr. 18 Gaussova mapa se svým bifurkačním diagramem.	28
Obr. 19 Gaussova mapa pro parametry $C=-0,898$ a $b=7.5$	28
Obr. 20 Bifurkační diagram Gaussovy mapy.	29
Obr. 21 Kvadratická mapa s bifurkačním diagramem pro parametr $C=2$	30
Obr. 22 Sinusová mapa s bifurkačním diagramem pro parametr $B \in (0,3)$	31
Obr. 23 Sinusová mapa pro parametr $B=2,6$	31
Obr. 24 Bifurkační diagram sinusové mapy pro parametr $B \in (0,7)$	32
Obr. 25 Sinus-kruhová mapa s bifurkačním diagramem.	33
Obr. 26 Sinus-kruhová mapa s parametry $\Omega = 0,6066661$ a $K=1$	34
Obr. 27 Bifurkační diagram sinus-kruhové mapy s parametrem $\Omega=0,606661$	34
Obr. 28 Lorenzův atraktor.	35
Obr. 29 Schéma ze Simulinku pro vykreslení Lorenzova atraktoru.	36
Obr. 30 Rosslerův atraktor.	37